

Microsoft®

100% technológia • 0% marketing

TechNet

New
New

A biztonság kulcsa

Windows XP Service Pack 2

Telepítési útmutató

A CRM szerver
és a kliens alkalmazások
Rendszermonitorozás
a .NET-ben

Újra az elektronikus
aláírásról

Mire jó az InfoPath?

ISSN 15865185



04

V./4. szám
2004. október

Maradjon a középpontban!

Microsoft
GOLD CERTIFIED
Partner

Learning Solutions

Microsoft CRM tanfolyamok

Ismerje meg a hatékony ügyfélkapcsolat-kezelés és értékesítés legjobb eszközeit!

MEGHIRDETETT CRM KÉPZÉSEK

Microsoft CRM 1.2 alkalmazások	november 15–17.
Microsoft CRM 1.2 telepítés, konfigurálás	november 18–19.
Microsoft CRM 1.2 testreszabás	december 13–14.
Microsoft CRM 1.2 megoldások fejlesztése	december 15–17.

OKTATÁSI CSOMAGOK

A tanfolyamok kedvezményes oktatási csomagokban is elérhetőek, így teljes mértékben lefedheti cége vagy partnere munkatársainak CRM képzését!

CRM TÁMOGATÁS

Kedvezményes CRM rendszerbevezetés-támogatás, konzultáció biztosítása kihelyezett formában a cég vagy – CRM viszonteladók esetén – a cég partnerének telephelyén.

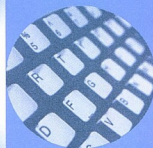
CRM KIEGÉSZÍTŐ KÉPZÉSEK (MCSA, MCSE)

Kedvezményes Microsoft **rendszeradminisztrátor**, **rendszermérnök és adatbázis-adminisztrátor** képzések a hatékony CRM bevezetéshez és menedzseléshez, **október 25-étől**.

További információ

Simon Ferenc, telefon: 203-0304/3050 mellék, e-mail: simonf@szamalk.hu

A SZÁMALK Továbbképzés a Microsoft első magyarországi aranyfokozatú partnere az oktatási megoldásokban



www.szamalk.hu/tiza/crm

TechNet Magazin

V. évfolyam, 4. szám
2004. október

Szerkesztőség és kiadó:

Microsoft Magyarország Kft.
1031 Budapest, Graphisoft park 3.

Felelős kiadó:

Székely Tamás marketingigazgató

Szerkesztő:

Takács Gitta (Epsilon Press)

Szaklektor:

Fóti Marcell (Netacademia)

Lapmenedzser:

Kolma Kornél
(Microsoft Magyarország)

Lapterv és nyomdai előkészítés:

Dobák Ildikó
Pataki Bernadett
(Ars Luna Bt.)

Borítógrafika:

Molnár Ferenc

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

A TechNet Magazinban közölt cikkek,

képek és illusztrációk csak a kiadóval történt előzetes egyeztetés után használhatók fel.

Adatvédelmi tájékoztató: Az Ön adatai

a Microsoft Magyarország adatbázisából származnak. Amennyiben nem kívánja, hogy a továbbiakban a TechNet Magazinmal vagy más ajánlatokkal keressük meg Önt, bármikor kérheti adatainak törlését a Microsoft Magyarország Kft. címére írott levélben vagy e-mailben.

Elkészült a magyar verzió is

MEGJELENT A WINDOWS XP SERVICE PACK 2 CSOMAG

Augusztusra gyártásra kész állapotba került a fejlett biztonsági technológiákat tartalmazó, ingyenes Windows XP Service Pack 2 (SP2) szervizcsomag, amely tartalmazza a Microsoft legújabb biztonsági frissítéseit és innovációit, erősebb védelmet nyújtó beállításokat tartalmaz meg, továbbá olyan új, biztonsági funkciókkal bővíti a rendszert, amelyek segítségével jobban megóvható a számítógép a hackerekkel, a vírusokkal és az egyéb biztonsági kockázatokkal szemben.

Az SP2 világméretű bevezetési folyamatának részeként a Microsoft 25 nyelvre – magyarra is – honosította illetve honosítja a szoftvert az elkövetkező hónapokban, és letöltéssel, kiskereskedelmi terjesztéssel, ingyenes CD-ken, valamint az új számítógépekre előtelepítve juttatja el a számítógépgyártókhoz, a vállalati ügyfelekhez és az egyéni felhasználókhoz. A már Windows XP rendszert használók úgy tudják a legegyszerűbben megoldani, hogy a kívánt nyelvű változat megjelenésekor megkapják az SP2 csomagot, hogy bekapcsolják a Windows XP Automatikusan frissítés funkcióját. A frissítéssel párhuzamosan egy új telepítőszoftver is letöltődik, amellyel optimál-

zálható az SP2 és a Windows XP jövőbeli frissítéseinek letöltése. Az, hogy valaki az SP2 csomagot az Automatikusan frissítés funkció segítségével mennyi idő múlva kapja meg, számos tényezőtől függ: ezek közé tartozik, hogy mennyit használja az internetet, hol él, milyen nyelvű rendszert használ, illetve, hogy mekkora igény jelentkezik az interneten az SP2 iránt. Az Automatikusan frissítés szolgáltatás az internet szabad sávszélességét használja a frissítések letöltésére, így nem lassítja a napi számítógéphasználatot. Akiknek nincs megbízható internetkapcsolatuk és CD-n szeretnének hozzájutni az SP2 csomaghoz, a Microsoft webhelyéről ingyenesen megrendelhetik a lemezt. A Microsoft a számítógépgyártókkal is kapcsolatban áll, hogy azok a lehető legmagasabb képesek legyenek olyan Windows XP alapú számítógépeket szállítani, amelyekre az SP2 csomag van előtelepítve.

A TechNet hasábjain természetesen ebben a számban is és a következőkben is részletesen foglalkozunk az SP2 kiértékelésével, a telepítésével kapcsolatos teendőkké, és a Microsoft honlapján is folyamatosan bővül a témával kapcsolatos ismeretek tárháza.

További információ:

<http://www.microsoft.com/hun/security/protect/>

Windows XP SP 2

ÚJDONSÁGOK, A TELEPÍTÉS VÁLTOZATAI, MÍGRÁCIÓ ÉS KOMPATIBILITÁS

Ebben a cikkben a Microsoft legújabb, nagy gondossággal és alaposággal fejlesztett illetve tesztelt, ingyenes javító-csomagjáról lesz szó, amely azonban lényegesen különbözik az eddigi szervizcsomagoktól. Mindenképpen előnyére.

ASP.NET 2.0 (Whidbey)

MÍ VÁRHATÓ A 2005. ÉVI ASP.NET-BEN?

II. rész: DetailsView, Data Componentek, ObjectDataSource

Sorozatunk előző részében áttekintettük az adatelérés alapjait az új 2.0-ás ASP.NET-ben, és láthattuk, hogy új adatelérő vezérlők és új megjelenítő vezérlők jelennek meg. Ebben a részben a DetailsView, a DataSet alapú adatelérés és az ObjectDataSource vezérlőkkel építünk fel egy webalkalmazást. Ezen példák már a VS.NET Beta 1 változatára épülnek.

Windows szolgáltatások 4. rész

A LOCALSYSTEM FIÓKOT HASZNÁLÓ SZOLGÁLTATÁSOK

Eddig a Windows Server 2003 mintegy 100 szolgáltatásának egyötödét „tudtuk le” ebben a sorozatban. Az eddig felsorolt szervizek közös tulajdonsága az volt, hogy nem a LocalSystem fiókot használták. Most felmérjük, miért ekkora sláger ez a fiók és melyek azok a szervizek, amelyek csak ezzel a hagyományos szolgáltatásfiókkal üzemelnek.

Rendszermonitorozás a .NET-ben 1. rész

RENDSZERFOLYAMATOK ÉS SZOLGÁLTATÁSOK MONITOROZÁSA

A .NET Framework számos olyan eszközt biztosít a fejlesztő számára, amelyek segítségével alkalmazásaik hozzáférhetnek az operációs rendszer folyamataihoz és szolgáltatásaihoz, megváltoztathatják azok állapotát, illetve lekérdezhetik tulajdonságaikat.

Kiszolgálópark üzemeltetése 3. rész

AZ ÉLETÚT ÉS AZ ÜZEMELTETÉSI FELADATOK

Már van hol és mivel, de hogyan? Hogyan tervezzem, telepítem, üzemeltessem? Hogyan dokumentáljam? Folytatjuk sorozatunkat a kiszolgálópark üzemeltetéséről.

Új törvény az elektronikus aláírásról

„RÁNCFELVARRÁS” 3 ÉV UTÁN

A 2001. évi elektronikus aláírásról szóló törvény megszületése óta nem változott sok minden az elektronikus aláírás elterjedését illetően, az áttörés, a tömeges alkalmazás továbbra is várat magára. A jogszabályi háttér megteremtéséért felelős kormányzat ezzel összefüggésben is az alig három éves jogszabály átdolgozásáról döntött.

Service Pack 1 szervizcsomag a SharePoint termékekhez

Jobban kezelhetők a nagyméretű dokumentumok és a frissítések, hatékonyabb lett a keresés és az adatbázisok visszaállítása.

Microsoft CRM v 1.2 – 2. rész

A CRM SZERVER ÉS A KLIENTS ALKALMAZÁSOK TELEPÍTÉSE

Ebben a részben megvizsgáljuk a CRM telepítési megoldásait, és ismertetjük, milyen előkészületeket és beállításokat kell elvégeznünk a rendszer bevezetéséhez.

Megjelent a Virtual Server 2005

A Virtual Server 2005, a Windows Server 2003 rendszerhez készített virtuálisgép-megoldás a „Dinamikus Rendszer Kezdeményezés” (Dynamic Systems Initiative Road Map) jegyében.

Ami a hivatalos Microsoft tanfolyamokból kimaradt...

MICROSOFT OFFICE 2003 - INFOPATH

Rovatunk mostani cikkében egy olyan termékről írunk, amely olyannyira „kimaradt”, hogy jelenleg egyetlen hivatalos tanfolyam sem foglalkozik vele. Az InfoPath egyike az egyre népszerűbb kis és középvállalatok részére kínált intranetes „csináld magad” megoldásoknak. Remekül integrálható a meglévő információportálokba, professzionális megoldásokat nyújtva mindenki meglegedésére.

Dr. Watson

ÉLRE TÖRÜNK A BÍZTONSÁG TERÜLETÉN: WINDOWS XP SP2

Megjelent a Windows XP soron következő javítócsomagja, a Service Pack 2.

Érdemes alaposabban megismerkedni vele, ugyanis viszonylag sok és lényeges változtatást tartalmaz a biztonsági beállítások területén.

Windows XP SP2

ÚJDONSÁGOK, A TELEPÍTÉS VÁLTOZATAI, MÍGRÁCIÓ ÉS KOMPATIBILITÁS

Ebben a cikkben a Microsoft legújabb, nagy gondossággal és alaposággal fejlesztett illetve tesztelt, ingyenes javító-csomagjáról lesz szó, amely azonban lényegesen különbözik az eddigi szervizcsomagoktól. Mindenképpen előnyére.

Bevezetés

Szinte napra pontosan három évvel a Windows XP RTM (Release to Manufacture) és másfél évvel a Service Pack 1 (2003. március) kiadása után 2004. augusztus 6-án megjelent az új szervizcsomag, az SP2. Eddig úgy tapasztaltam, hogy amikor elkészül egy, akár szerver-, akár kliens oldali SP Redmondban, akkor a rendszergazdák és más szakemberek viszonylag gyorsan tudomást szereztek róla (naná!) és árgus szemekkel tesztelték és figyelték mások tapasztalatait, hogy minél hamarabb telepíthessék ezeket. De most más lett a helyzet: a Microsoft az SP2 kapcsán jelentősen kibővítette a bevezetni kívánt újdonságok arányát, a tesztelés időtartamát és a tesztben résztvevők számát (több mint egymilliószor töltötték le az SP2 tesztverziót a Microsoft Technical Preview, Technical Beta és MSDN programja keretében), majd a Microsoft többször elhalasztotta a kiadás időpontját az alaposág miatt. Mindemellett szerzőgázón meg is hirdette előzetesen a csomag összes tervezett előnyét, illetve az esetleges velejáró (akár a kényelem rovására történő) szigorításokat is és szinte a Gyűrűk Ura/Mátrix premiereknél tapasztalható várakozás előzte meg a javítócsomag színelőadását, ezért most mindenki erőlről beszél, akár ért hozzá, akár nem.

Mit nyújt az SP2?

A SP2 csomagban található frissítések és fejlesztések három fő terület köré csoportosíthatók:

Szigorúbb biztonsági beállítások és eszközök

A Windows Firewall megújult és alapértelmezés szerint a „Kivételek nélküli aktiválás” módban (minden hálózati kapcsolatnál) bekapcsolt üzemmódba korrekelt megoldás a kéretlen hálózati forgalom ellen, akár alkalmazás, akár port szinten. Fontos megemlíteni, hogy az IPv4 és IPv6 tűzfal meghajtott egyaránt egy ún. „boot-time policy” alapján működnek a bekapcsoláskor, azaz lehetővé teszik a hálózati belépés előtti mindenképp szükséges DHCP és DNS forgalmat (de csak azt). Amikor pedig a Windows Firewall teljes mellszéliséggel működésbe lép, akkor ezt az (általunk egyébként nem szabályozható) házirendet kikapcsolja.

Az SP2 alapértelmezett biztonsági beállításai szigorúbbak, emellett olyan új szolgáltatásokat és megoldásokat (pl. az RPC objektumok csökkentett jogokkal történő futtatása, vagy

a DCOM infrastruktúrára vonatkozó megszorítások) tartalmaz, amelyekkel az operációs rendszer jobban megvédhető az illetéktelen behatolásokkal vagy a vírusokkal szemben.

Ne feledkezzünk meg a processzorgyártókkal közösen fejlesztett szoftver alapú memóriavédelemről sem (DEP = Data Execution Prevention), amely (ha a processzor is támogatja) megengedi az operációs rendszernek, hogy adott szakaszokat nem futtathatóként jelöljön meg a rendszermemóriában (kivéve, ha egyértelmű, hogy az adott memóriacím futtatható kódot tartalmaz). Így azok az alkalmazások (pl. vírusok), amelyek ezeket a részeket szeretnék jogosulatlanul használni, tehetetlenné válnak, buffer underrun hibákat sem tudnak generálni. A DEP rendszerszintű szabályozása a boot.ini-ből történhet, a `/noexecute` kapcsoló négy opciójával, ám bekapcsolt állapotában a System Properties / Advanced / DEP fül alatt is lehetséges befojósolni a működését. A szervizcsomag természetesen tartalmazza az operációs rendszer RTM kiadása óta megjelent biztonsági és egyéb frissítéseket is [1] [2].

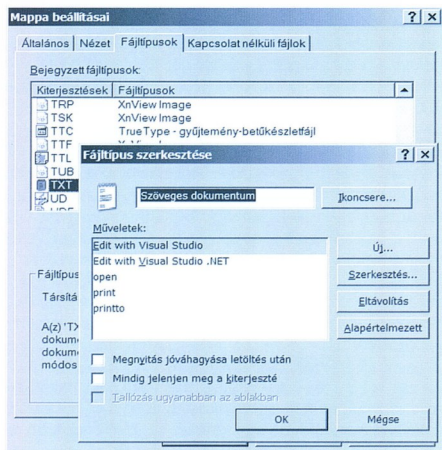
Jobb kezelés és felügyelet

A biztonságot üzemeltetést segítő hibajavítással és frissítéssel kapcsolatos területeken is történtek változások, ezek közül az elsőként megemlíthető a Security Center (Biztonsági központ), amely ellenőrzi, hogy futnak-e a következő programok: 1. egy tűzfal program; 2. egy naprakész vírusvédelmi program; 3. a frissítések automatikus letöltésére beállított *Automatikus frissítések* szolgáltatás. Ha ezek közül valamelyik nincs, vagy nem aktuálisan frissített, akkor erre figyelmezteti a felhasználót, sőt a probléma elhárításáig minden egyes bejelentkezőkor megjelenik egy további figyelmeztetés. Ellenkéntben az otthoni felhasználással, tartományi környezetben a Security Center alpból nem működik, de engedélyezése után a Csoportházi renddel részletesen szabályozható.

Az Automatic Updates kient is lecserélték, a jelenlegi változat továbbra is képes együttműködni a SUS-sal, de a SUS jelenleg még béta fázisban lévő, ám ígéretes utódjával a WUS-sal (Windows Update Services) is probléma nélkül dolgozik. Az új kliens viszont a fontos frissítéseken kívül képes az ún. update rollups-ok és a szervizcsomagok letöltésére és telepítésére is és talán az egyik legszimpatikusabb újdonságot is alkalmazza, a Shutdown Install-t, azaz a gép leállításakor el-

végzendő frissítést. Az ebbe a témakörbe passzoló BITS (Background Intelligent Transfer Service) 2.0-ás változata és a delta tömörítés használata nagytólódhat a kisebb sávszélességgel elvégzett frissítések tempóján.

A szervizcsomaggal érkező házirend sablon – például tartományi környezetben a Csoportházirendbe beágyazva – gyakorlatilag az összes új funkció központi felügyeletére alkalmasak és például a Windows Firewall esetében különösen hasznosak, hiszen attól is függővé tehető a beállítások segítségével a kiens oldali tűzfal működése, hogy tartomány vagy munkacsoportos környezetben működik a gép.



■ Megjelenik a Shutdown Install lehetőség

Az Internet Explorer is tartalmaz jónéhány a biztonsággal kapcsolatos újdonságot és az alap biztonsági beállítások is szigorúbbak. Megakadályozza például, hogy szkriptek futtatásával egy olyan ablak nyíljon meg, amely elrejt az Internet Explorer vezérlőelemeit, és ezzel „elfedi” a hátsó szándékú tevékenységet. Korlátozza a helyi számítógépen emelt szintű jogosultságokkal történő HTML-kód végrehajtását, és értesíti a felhasználót, ha potenciálisan veszélyes letöltést észlel. Az IE és az Outlook Express a letöltött illetve lementett állományokat egy ún. ZoneID-val megjelöli, majd ez az azonosító beépül a letöltött állományba (az NTFS, Additional Data Stream (ADS) funkcióját kihasználva). Ezután, akár hetek múlva is kiolvasható az állomány tulajdonságai közül és persze fel is használható pl. a forrás megbízhatóságának ellenőrzésekor, azaz akkor, amikor majd a felhasználó megnyitja a letöltött alkalmazást. Az IE egy kellemes bővítménykezelővel is rendelkezik, amelyben átlátható módon kezelhetjük (letilthatjuk) például az automatikusan beépülő elemeket.

További újdonságok

Az SP2 tartalmazza a fontosabb illesztőprogramok továbbfejlesztett verzióját, és a felülről technológiák (Wi-Fi, Bluetooth) szembetűnően átdolgozott támogatását is. Most már csak a felsorolás szintjén nézzük az ebbe a kategóriába tartozó újdonságokat:

- Internet Explorer előugró ablakok blokkolása
- Internet Explorer információs eszköztár
- DirectX 9.0c
- Windows Media Player 9 Series
- Windows XP Media Center Edition
- Windows XP Tablet PC Edition

Hogyan telepíthetjük szülő gépen?

Ha először egy gépen akarjuk kipróbálni, akkor az SP2-t a már említett módon, AU kliens segítségével is beszerezhetjük, de ez sokkal hosszadalmasabb, hogy települ is automatikusan (a beállításoktól függően). Az egész futtatható csomag (ún. network install, a cikk leadásakor egyelőre még csak angol nyelven) letöltését is megtehetjük (266 MB) [3], vagy akár rendelhetünk egy ingyenes CD-t [4]. Ha letöltöttük a network install csomagot, akkor rögzvet telepíthetjük is lokálisan az adott gépre (standalone telepítés), vagy kibontathatjuk a csomagot az „-x” kapcsolóval és felhasználhatjuk egy slipstream (azaz a szervizcsomaggal integrált, van erre a célra egy külön kapcsolója a telepítőnek: /integrate) változat készítésére [5], majd ügyesen boot CD-t is kreálhatunk belőle [6]. Így az új gépekre már rögtön egy SP2-vel ötvözött friss, ropogós Windows XP mehet. Ez az eljárás teljesen jogtiszt és külön licenct nem természetesen nem igényel.

A telepítés után használhatjuk az új Support Tools-t és a Deployment Tools-t is, ezek a hasznos kiegészítők is megtalálhatóak a CD-n, illetve letölthetőek [7] [8].

Ha szülő telepítést végzünk, akkor a telepítő legyárt egy naplóállományt, a %Systemroot% mappába, Svcpack.log néven (integrált telepítéskor Spsslpsrm.log az állomány neve). Ebben a következő információkat találhatjuk meg:

- annak a parancssori segédprogramnak a nevét, amely az update.exe-et beindítja,
- azoknak az állományoknak a nevét, amelyekre a telepítő felmásol a gépünkre,
- hibaüzeneteket (ha voltak problémák),
- azokat a parancssori programokat, amelyeket a telepítés során használt az update.exe

Amernyiben az ún. *Unattended* (válaszadás nélküli) telepítést választjuk, akkor használjuk az unattend.txt válaszállományt (csak az integrált telepítésnél). Ezt az állományt az ismert módon egy text editorral vagy a Setup Managerrel készíthetjük el. Gyakorlatilag ennél a telepítési változatnál nincs újdonság, minden információ és eszközt megtalálunk a Deploy.cab állományban.

A csomag, vagy a csomagból kibontott update.exe jónéhány kapcsolóval rendelkezik, a teljes listát nem sorolnám fel, csak kiemelnek egy-két érdekesebbet a teljesség abszolút igénye nélkül:

/integrate: ez a korábban már említett integrált változat készítéséhez szükséges, külön erre a célra fejlesztett kapcsoló újdonság

/passive: nincs felhasználói interakció, bár a telepítés folyamata nyomkövethető a GUI-n

/F: kérdés és mentés nélkül, agresszíven bezár minden futó alkalmazást a telepítés végén, hogy minél előbb újraindíthassa az operációs rendszert

/N: nem készül biztonsági másolat, azaz helytakarékos, bár hátránya, hogy nem lesz eltávolítható a csomag.

Hogyan telepíthetjük többgépes környezetben?

Egy Windows hálózatban, a nem túlságosan komplex szülő telepítés mellett több, az adott hálózat lehetőségeihez, az hardver/softver környezet vagy a szokásjogokhoz passzoló forgatókönyv (vagy ezek kombinációja) is „lejátszható”.

Központi helyről indított lokális telepítés

Ebben az esetben a számítógépek egy helyi háttérrel rendelkeznek a telepítő állományok tárolására, de a telepítő egy központi helyről indítható. Előnye, hogy anélkül megy le a frissítés, hogy egy hálózati megosztáshoz kellene kapcsolódnia folyamatosan a telepítőnek.

Központi telepítés

Készíthetünk egy megosztást egy központi helyen, amelybe a kibontott csomag tartalmát bemásoljuk és a kliensekről ehhez a megosztáshoz csatlakozva frissítünk.

Telepítés a SUS segítségével

Mivel a SUS SP1 óta van lehetőségünk a szervizcsomagok a biztonsági frissítésekhez hasonlóan egyszerű és központi (és akár automatikus) telepítésére, ezért augusztus 16-a óta (ekkor jelent meg a WU szervereken a csomag) az SP2-t is feltehetjük a kliens gépekre ezzel a módszerrel. Nincs semmilyen különbség, szinkronizálunk, engedélyezzük a csomagot (approve) és várunk. Pár körülményt azért tartunk szem előtt. pl. a letöltéssel járó esetleges sávszélesség problémát, illetve a „terítésrel” járó hálózati forgalmat. De ezeket összegezve (és beleértve a korábban említett AU kliens önálló működését is) is megéri a SUS-t használni egy tartományban az SP2 telepítésére, mert

- csak egyszer kell letölteni
- kizár(hat)ja a WU és AU féle telepítést
- a SUS segítségével az ún. „silent” azaz, beavatkozás nélküli telepítés hajtódik végre a klienseken [9].

Ez utóbbi lehetőség egy utólagos SUS korrekció (08.19.) miatt létezik, ti. az első SP2-es csomag (08.16.) SUS-on keresztüli telepítések még szükség volt felhasználói interakcióra a klienseken, azonban ezt korrigálták. Természetesen az új szinkronizálás után, az egész csomagot nem kell még egyszer letöltenünk, hanem csak egyetlen elemet (aurf.cab), amely kevesebb mint 1 MB méretű és amely tartalmazza a beavatkozás nélküli telepítéshez szükséges frissített telepítési paramétereket.

A SUS és az SP2 kapcsolatról, a tervezési és méretezési feladatokról szóló hosszú-hosszú dokumentumot megtekinthetjük itt [10].

Telepítés az SMS segítségével

Természetesen van lehetőség a legutóképebb nagyvállalati megoldás, az SMS segítségével is telepíteni a Windows XP SP2-t, de ennek részleteibe helyhiány miatt ezeken a hasabokon nem térünk ki. Viszont a következő oldalon bőseges információkat találhatunk erről a telepítési módról is [11].

Telepítés a Csoportházi renddel

Active Directory környezetben rendelkezésünkre áll a Szoftvertelepítés funkció, egyetemben a Windows Installer technológiával a szervizcsomag központi telepítéséhez. Ebben

az esetben az Update.msi csomag tartalmazza azokat az információkat, amelyek szükségesek a Windows Installer számára a telepítéshez (vagy akár az eltávolításhoz). Természetesen bármilyen a Csoportházi rend hatókörében belüli eszközben (telephelyek, tartományok vagy szervezeti egységek) beállíthatjuk a szoftvertelepítést, bár a csomag mérete miatt ügyeljünk a megfelelő célpont kiválasztására. A Windows Server 2003-ban van lehetőségünk WMI filterekkel behangolni a csomag célpontjait, egy olyan szervezeti egységben is, ahol nemcsak Windows XP kliens számítógép fiókok találhatók. Viszont az SP2 esetében csak a gépekhez való hozzárendelést (Assign) választhatjuk ki, mint telepítési módszert. A szokásos megosztás elkészítése (vagy DFS használata), illetve a telepítés tulajdonjogainak a csoportházi rendben való beállítása után, a kliens gépen történő következő hálózati belépéskor a szervizcsomag automatikusan települ.

A Windows Firewall beállítása a telepítés előtt/után

Bármelyik módszerrel is telepítjük, felmerülhet a probléma, hogy ha alapértelmezés szerint be van kapcsolva a tűzfal, akkor hogyan fogunk kapcsolatba lépni akár a telepítés alatt, akár később a klienssel? Hiszen egy sima pinget sem fog megengedni, és nincs olyan kerülő, intelligens megoldás, mint egy, az RDP-n keresztül ISA 2004 távtelepítés esetén. Van viszont több ajánlott manuálisan végrehajtható megoldás, itt most három különböző variációról szeretnék említést tenni.

Unattend.txt: Egy – esetlegesen – központi helyen tárolt, válaszadás nélküli telepítést szabályozó válaszállományban megtalálható egy szakasz és ezen belül több alszakasz is, ami a tűzfal beállításait teszi lehetővé, akár profilonként (Domain, Standard) különválasztva. Találunk benne beállításokat a tűzfal teljes kikapcsolására, a Remote Assistance, a Remote Desktop, a WebService komponensekre vonatkozóan, vagy akár az alkalmazásokra, portokra, szolgáltatásokra, vagy az ICMP parancsokra is. Egyszerűen, átfogó előzetes konfigurációs beállításokat határozhatunk meg.

Netfw.inf állomány: A tűzfal ezzel a módosítható INF állománnyal érkezik, amely a regisztrációs adatbázisban található alapértelmezett beállításokat tartalmazza. Mindkét fajta telepítés (standalone és integrated) és a későbbi működés során is ez az állomány használatos, ennek alapján „lővi be” a tűzfalat az operációs rendszer. A módosítása gyakorlatilag megfelel a tűzfal GUI-n történő konfigurálásához. A Netfw.inf első két szakaszát (verzió- és konfiguráció információ), nem célszerű módosítani, viszont jól láthatóan elkülönül benne a két profilra vonatkozó szakasz, amelybe bevezethetjük a szükséges beállításainkat. Az innen [12] letehető olvasnivalóban a bejegyzések konkrét formájához alapos segítséget és számos példát is kapunk. A módosítás előtt célszerű az alap INF állományról egy másolatot készíteni, majd az állomány készíttel kicserélni, az azonnali érvényesítést pedig a következő parancssorral kikényszeríteni: „netsh firewall reset”.

Csoportházi rend: Talán ez a legkényelmesebb megoldás, hiszen tartományi környezetben központi helyről, egyszerűen és teljeskörűen, ám minimális munkával szabályoz-

hatjuk a tűzfal beállításokat (is). Az új tűzfalra vonatkozó részt a Computer Configuration / Administrative Templates / Network / Network Connections szakaszban találjuk meg. Kétféle módon is frissülhet a Csoportházi rendünk az SP2 új beállításait tartalmazó sablonokkal (természetesen nemcsak a tűzfal sablonokra vonatkoznak ezek a módszerek, az SP2 rengeteg újítása miatt, a Csoportházi rendtől alaposan ki kell bontani). 1.; egy, már az SP2-vel működő tartományi kliensről az mmc.exe-vel megnyitjuk az adott GPO-t, 2.; a Csoportházi rendben kézzel beimportáljuk az új system.adm (1703 KB a korábbi legfrissebb w2k3-as 1474 KB-al szemben) mellett inetres.adm, a wmlplayer.adm illetve wuau.adm sablon állományokat. Csak a pontosság kedvéért: automatikusan csak azokba a GPO-kba kerülnek be az új sablonok, amit megnyitunk (tehát ha egy GPO-ba bekerült, az meg nem jelenti azt, hogy azonnal a többi is frissülni fog).

Fontos felhívni a figyelmet arra a körülményre, hogy az új sablonok kicsit más szerkezetűek, azaz nem teljesen kompatibilisek a jelenlegi GPO szerkesztőkkel. Ennek az a következménye, hogy alábból csak az XP SP2-es kliensekről tudjuk hibát nélkül a Csoportházi rendet konfigurálni. Ha pl. Windows 2000/2003 alatt próbáljuk megnyitni valamelyik GPO-t, akkor a következő (többszörös) hibaüzenettel szembesülünk:

```
"The following entry in the [strings] section is too long and has been truncated.."
```

A probléma a Microsoft előtt is ismert és az SP2 kiadása után rövest meg is voltak a javítások, amelyeket a következő helyekről le is tölthetünk, Windows 2000 és Windows 2003 Server-hez egyaránt [13] [14].

Segítség a migráláshoz, inkompatibilitás

A Microsoft kiadott egy részletes, több mint 100 oldalas, angol nyelvű migrációs útmutatót [15] azon rendszergazdák számára, akik át szeretnék térni a Windows XP Service Pack 2-es verziójára. Számos rémhír és dezinformáció keringett és kering ma is az interneten (nem kevés – akár más területeken egyébként magas színvonalú – magyar weboldalon is) az SP2-ről, ezért célszerű volt egy részletes és hiteles leírás kiadása.

Sokan attól tartanak, hogy az SP2 új biztonsági megoldásai és restriktói miatt jónéhány eddig használt szoftver nem fog korrek módon működni, az „Application Compatibility Testing and Mitigation Guide for Windows XP Service Pack 2” című dokumentumban viszont megtaláljuk, hogyan lehet kompatibilitási tesztekkel végezni, és az esetlegesen felmerülő hibákat elhárítani. Ezen kívül két lehetséges telepítési útmutatót is tartalmaz nagyvállalati rendszerek számára, valamint rengeteg példa szkriptet (Outlook, WF, DCOM, IE, RPC kategóriában) amelyekkel enyhíthetjük az SP2 által okozott „sebeket”.

A Microsoft persze tisztában van bizonyos inkompatibilitási esetekkel, ez a KB cikk [16] a tűzfaljal kapcsolatos esetleges problémákat és a hibakeresést vázolja. Az itt [17] található KB cikkben pedig egy lista található azokról az alkalmazásokról, amelyekkel gond lehet. Hasznos lehet az a gyűjtemény is, amelyet egy MVP kolléga állított össze és jelentős mennyiségű SP2-vel kapcsolatos KB cikket és egyéb forrást tartalmaz [18].

Hogyan ne telepítsük?

Gondolom kicsit meglepő ez a kérdés, épp ebben az újságban, de azért életszerű. A Microsoft szerint előfordulhat olyan eset, amikor a szükséges, ám esetenként kellemetlen megszorítások és új funkciók miatt (pl. vállalati szinten), alaposan fel kell készülni az SP2 bevezetésére. Előfordulhat viszont, hogy a kliensek a Windows Update weboldalon keresztül, vagy az Automatic Update kliens segítségével nem kívánatos módon mégis letöltik és telepítik a szervizcsomagot. Ezt elkerülendő a Microsoft publikált három megoldást, amelyekkel 120 napig (augusztus 16-tól kezdve) lehetséges késleltetni az SP2 alkalmazását. Az egyik módszer megint csak egy tartományban „élő” megoldás, a Csoportházi rendhez kiadtak egy új ADM sablont, amely blokkolja a WU/AU módszerrel végzett telepítést a Windows XP RTM és SP1-es változatán. A következő helyre kerül a Csoportházi rendben:

```
Computer Configuration / Administrative Templates / Windows Components / Windows Update
```

A szokásos módon lehetséges beimportálni, majd részletesen konfigurálni az egyetlen új opciót.

Amennyiben nem lehetséges a Csoportházi rend segítségével kezelni a kliens gépeket, akkor használhatjuk a Microsoft által digitális aláírással ellátott futtatható állományt (XPSP2Blocker.exe). Az alkalmazás amelyet a /B kapcsolóval állítunk tiltó üzemmódba, gyakorlatilag egy registry bejegyzést (DoNotAllowXPSP2) kreál a lent kiemelt helyre, amelynek az értékét 1-re állítja (az /U kapcsoló hatására pedig törli a bejegyzést).

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

Rendelkezésünkre áll még egy szkript is, amely gyakorlatilag ugyanezt teszi meg, de rugalmasabban parameterezhető. Az ezekhez a módszerekhez való kellék egy adagban, az ún. „Toolkit to Temporarily Block Delivery of Windows XP SP2 to a PC Through Automatic Updates and Windows” csomagban találhatóak meg, amelytől innen letölthető [19]. Természetesen az interaktív vagy a SUS/SMS/GP által végzett telepítésre ezeknek az eszközöknek a használata nincs befolyással.

GÁL TAMÁS
MCSE 2003
MVP (Software Distribution)
gtamas@tjszki.hu

A cikkben szereplő URL-ek:

- [1] <http://tinyurl.com/5uwhg>
- [2] <http://support.microsoft.com/?kbid=811113>
- [3] <http://tinyurl.com/5uqoe>
- [4] <http://tinyurl.com/6g6575>
- [5] <http://tinyurl.com/63abx>
- [6] <http://tinyurl.com/4tb2m>
- [7] <http://support.microsoft.com/?kbid=838080>
- [8] <http://support.microsoft.com/?kbid=838079>
- [9] <http://tinyurl.com/6wkcw>
- [10] <http://tinyurl.com/5she4>
- [11] <http://tinyurl.com/3vbnx>
- [12] <http://tinyurl.com/6d95a>
- [13] <http://tinyurl.com/4ze3y>
- [14] <http://tinyurl.com/4apzv>
- [15] <http://tinyurl.com/6l6wn>
- [16] <http://tinyurl.com/4uung>
- [17] <http://tinyurl.com/6eusa>
- [18] <http://jmf.mps.nvps.org/SP2.htm>
- [19] <http://tinyurl.com/4an3w>

ASP.NET 2.0 (Whidbey)

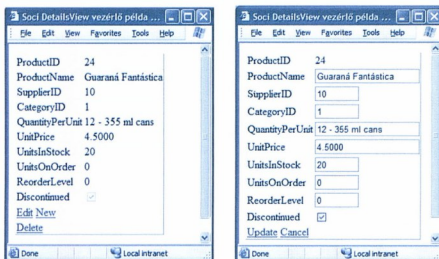
Mi várható a 2005. évi ASP.NET-ben?

II. RÉSZ: DETAILSVIEW, DATA COMPONENTEK, OBJECTDATASOURCE

Sorozatunk előző részében áttekintettük az adatelérés alapjait az új 2.0-ás ASP.NET-ben, és láthattuk, hogy új adatelérő vezérlők és új megjelenítő vezérlők jelennek meg. Ebben a részben a DetailsView, a DataSet alapú adatelérés és az ObjectDataSource vezérlőkkel építünk fel egy webalkalmazást. Ezen példák már a VS.NET Beta 1 változatára épülnek.

DetailsView

A GridView több rekordot jelenít meg, táblázatos formában, azaz a mezőket oszlopokban. Ezzel szemben a DetailsView egy rekordot mutat meg, minden elemét egy sorban kiírva. Az Edit gombra kattintva szerkeszthetővé válnak a sorok:



A DetailsView csak olvasható és szerkesztő módban

A fentieket generáló DetailsView deklaráció:

```
<asp:DetailsView
  ID="ProductDetails"
  Runat="server"
  DataKeyNames="ProductID"
  AutoGenerateRows="False"
  DataSourceID="ProductDataSource"
  AutoGenerateDeleteButton="True"
  AllowPaging="True">
  <Fields>
    <asp:BoundField
      InsertVisible="False"
      SortExpression="ProductID"
      DataField="ProductID"
      ReadOnly="True"
```

```
      HeaderText="ProductID" />
    <asp:BoundField
      SortExpression="ProductName"
      DataField="ProductName"
      HeaderText="ProductName" />
    <asp:BoundField
      SortExpression="SupplierID"
      DataField="SupplierID"
      HeaderText="SupplierID" />
    <asp:BoundField
      SortExpression="QuantityPerUnit"
      DataField="QuantityPerUnit"
      HeaderText="QuantityPerUnit" />
    <asp:BoundField
      SortExpression="UnitPrice"
      DataField="UnitPrice"
      HeaderText="UnitPrice" />
    <asp:CheckBoxField
      SortExpression="Discontinued"
      HeaderText="Discontinued"
      DataField="Discontinued" />
    <asp:CommandField
      ShowInsertButton="True"
      ShowEditButton="True" />
  </Fields>
</asp:DetailsView>
```

Érdemes megfigyelni az *InsertVisible="False"* sort a *ProductID BoundField*-ben. Ez azt mondja meg az adatforrásnak, hogy INSERT művelet esetén nem szeretnénk felhasználni az oszlop értékét, sőt beszúrás módban nem is látszik az oszlop mögötti *TextBox*. Miért? Mert a *ProductID* automatikusan adatbázisban generálódó érték, egy *IDENTITY* oszlop, amelyet normál körülmények közt nem írhatunk be az SQL INSERT parancsba.

Az SqlDataSource

Az adatforrás esetünkben az Sql Server natív adatbázis-osztályaira épít, ezért a @-os paraméternevek:

```

<asp:SqlDataSource
  ID="ProductDataSource"
  Runat="server"
  ProviderName="System.Data.SqlClient"
  SelectCommand="SELECT ProductID, ProductName,
  SupplierID, CategoryID,
  QuantityPerUnit, UnitPrice,
  UnitsInStock, UnitsOnOrder,
  ReorderLevel, Discontinued
  FROM dbo.Products
  WHERE ProductID = @ProductID"
  UpdateCommand="UPDATE Products
  SET
  ProductName = @ProductName,
  UnitPrice = @UnitPrice,
  SupplierID = @SupplierID,
  CategoryID = @CategoryID,
  QuantityPerUnit = @QuantityPerUnit,
  UnitsInStock = @UnitsInStock,
  UnitsOnOrder = @UnitsOnOrder,
  ReorderLevel = @ReorderLevel,
  Discontinued = @Discontinued
  WHERE ProductID = @original_ProductID"
  DeleteCommand="DELETE FROM Products
  WHERE
  ProductID = @original_ProductID"
  InsertCommand="INSERT INTO Products
  (ProductName, SupplierID,
  CategoryID, QuantityPerUnit,
  UnitPrice, UnitsInStock,
  UnitsOnOrder, ReorderLevel,
  Discontinued)
  VALUES (
  @ProductName, @SupplierID,
  @CategoryID, @QuantityPerUnit,
  @UnitPrice, @UnitsInStock,
  @UnitsOnOrder, @ReorderLevel,
  @Discontinued)"
  ConnectionString="Integrated Security=SSPI;
  Initial Catalog=Northwind;
  Data Source=(local)">
<InsertParameters>
  <asp:Parameter
    Name="UnitPrice"
    Type="decimal" />
</InsertParameters>
<SelectParameters>
  <asp:QueryStringParameter
    Name="ProductID"
    Type="Int32"
    QueryStringField="id" />
</SelectParameters>
</asp:SqlDataSource>

```

A megjelenítendő terméket az *id* nevű *QueryString* szolgáltatja, e paraméterátadási módról az előző részben olvashatnak bővebben.

A *ProviderName*="System.Data.SqlClient" a machine.configban definiált adatforrások közül választja ki az SQL Serverhez tartozó osztályokat. A machine.config idevágó részlete:

```

<configuration>
<system.data>
  <dbproviderfactories>
  <add name="Odbc Data Provider"
    invariant="System.Data.Odbc"
    support="BF"

```

```

    description="Data Provider for Odbc"
    type="System.Data.Odbc.OdbcFactory,
    System.Data" />
  <add name="SqlClient Data Provider"
    invariant="System.Data.SqlClient"
    support="FF"
    description="Data Provider for SqlServer"
    type="System.Data.SqlClient,
    SqlClientFactory, System.Data , />
  </dbproviderfactories>
</system.data>
</configuration>

```

A Factory osztályok egyszerűen *Connection*, *Command*, *DataAdapter*, stb. példányokat hoznak létre, mindegyik a megfelelő adatbázishoz való osztályhalmazból (GoF Abstract Factory Pattern) [2]. Ezzel a megoldással viszonylag egyszerűen tudnak majd a varázslók váltani a különböző adatforrások között, és mégsem kell ledegradálniuk a generált kódot a legnagyobb közös nevezőre, az OleDb-re. De miért kell foglalkozni külön a UnitPrice oszloppal INSERT művelet esetén? A UnitPrice *money* típusú az SQL táblában. Ha nem adunk meg típust az *asp:Parameter* vagy egyéb *Parameter* elemmel, akkor *nvarchar* típusként mennek át a beszűrődő paraméterek. Azonban az SQL Serverben az *nvarchar* és a *money* típus között nincs implicit konverzió, így a típus megjelölése nélkül a beszűrésünk hibát okozna. A fenti *SqlDataSource* felhasználásával egy új termék felvitela a következő SQL parancssal történik meg:

```

exec sp_executesql N'INSERT INTO Products
(ProductName, SupplierID,
CategoryID, QuantityPerUnit,
UnitPrice, UnitsInStock,
UnitsOnOrder, ReorderLevel,
Discontinued)
VALUES (
@ProductName, @SupplierID,
@CategoryID, @QuantityPerUnit,
@UnitPrice, @UnitsInStock,
@UnitsOnOrder, @ReorderLevel,
@Discontinued)',
N'@UnitPrice decimal(5,0),
@ProductName nvarchar(4),
@SupplierID nvarchar(1),
@CategoryID nvarchar(1),
@QuantityPerUnit nvarchar(1),
@UnitsInStock nvarchar(4),
@UnitsOnOrder nvarchar(5),
@ReorderLevel nvarchar(3),
@Discontinued bit', @UnitPrice = 14444,
@ProductName = N'alma', @SupplierID = N'1',
@CategoryID = N'1', @QuantityPerUnit = N'1',
@UnitsInStock = N'1222', @UnitsOnOrder =
N'11111', @ReorderLevel = N'111', @Discontinued = 0

```

Azért ez a kultúrától még elég távol van. A probléma szemmel láthatóan az, hogy szinte minden típus *nvarchar*-ként megy át, és az SQL Server végzi el a szükséges adatkonverziókat. A példában pont nem volt törtszám vagy dátum, de ugye jól tudjuk, hogy ezek szöveges ábrázolása kultúrafüggő, azaz könnyen lehet, hogy az SQL Server nem érti meg az ASP.NET által generált „2004. 12. 11.” formátumú magyar dátumot.

Mi a megoldás? Típusosan adjuk át a paramétereket! Ehhez viszont segíteni kell az *SqlDataSource* vezérlőnek:

```
<InsertParameters>
  <asp:Parameter Name="UnitPrice"
    Type="decimal" />
  <asp:Parameter Name="SupplierID"
    Type="int32" />
  <asp:Parameter Name="CategoryID"
    Type="int32" />
  <asp:Parameter Name="UnitsInStock"
    Type="int16" />
  <asp:Parameter Name="UnitsOnOrder"
    Type="int16" />
  <asp:Parameter Name="ReorderLevel"
    Type="int16" />
  <asp:Parameter Name="Discontinued"
    Type="Boolean" />
</InsertParameters>
```

Hasonlóan az *updateParameter*st is ki kell tölteni:

```
<UpdateParameters>
  <asp:Parameter
    Name="ProductID"
    Type="int32" />
  ... a többi mint az InsertParametersnél
</UpdateParameters>
```

Így már megfelelő SQL kimenetet kapunk (részlet az SQL Profiler által elkaptott parancsból):

```
...
N'@UnitPrice decimal(2,0).
@SupplierID int,@CategoryID int,
@UnitsInStock smallint,
@UnitsOnOrder smallint,
@ReorderLevel smallint,
@Discontinued bit,
@ProductName nvarchar(3),
@QuantityPerUnit nvarchar(1)',
@UnitPrice = 12, @SupplierID = 1, @CategoryID = 1,
@UnitsInStock = 123, @UnitsOnOrder = 1234,
@ReorderLevel = 12345, @Discontinued = 1,
@ProductName = N'bbb', @QuantityPerUnit = N'1'
```

Az *updateCommand* okozott némi meglepetést. Intuitíven a módosítandó sort leszűrő feltételt így írtam meg:

```
WHERE ProductID = @ProductID
```

Azonban ez a megoldás elszállt azzal a hibázenettel (az SQL Server problémázott), hogy a *@ProductID* paramétert nem adták át a parancsban. A Profiler kimenetet elemezve kiderült, hogy a paramétert *@originalProductID* néven várják el. Ezért az UPDATE WHERE feltételt helyesen:

```
WHERE ProductID = @originalProductID
```

Azért ez nem hagyott nyugodni. Honnan kellett volna ezt nekem tudnom? Mivel ilyen esetekben (főleg Béta termék esetén) a dokumentáció nem sokat segít elölvettem a .NET programozás kétségtelenül legtöbbet használt eszközét, a .NET Reflektort [1]. Ez egy típusbongésző és IL kód visszafejlesztő alkalmazás. A következőképpen gondolkodtam. Ki hajtja végre az adatbázis parancsokat? Az *SqlDataSource*

objektumom. A Reflectorban látszik, hogy van neki egy *Update()* metódusa. A doksi megerősít benne, hogy ezt hívják meg az adatbázisművelet végrehajtásához. A metódus így néz ki (a Reflector fejtette vissza):

```
public int Update() {
    return this.GetView().Update(null, null, null);
}
```

Azaz a *SqlDataSource* lekér egy referenciát egy mögöttes lapulós *SqlDataSourceView* objektumra, és annak szintén *Update* metódusára delegálja a feladatot. Abban:

```
public int Update(IDictionary keys, IDictionary
  values, IDictionary oldValues) {
    return this.ExecuteUpdate(
      keys, values, oldValues);
}
```

Hm, az *ExecuteUpdate* lesz a mi emberünk, aki megkapja a kulcsokat, az aktuális (módosított) és az eredeti értékeket. Ezeket a *DetailsView* adja át az adatforrásnak. Az *SqlDataSourceView.ExecuteUpdate(...)* belsejét nem másolom ide, mert hosszú, de van benne egy szembeötölő propertyhívás: *OldValuesParameterFormatString*. Ebbe belenézve fellelben a fátyol, és átnézünk a Mayán:

```
public string OldValuesParameterFormatString{
    get {
        ...
        return „original_{0}”;
    }
}
```

A paraméter neve elébiggyesztenek egy *original_* sztringet. Ezért kellett nekem *@originalProductID*-t írnom! Miután erre rájöttem, megnéztem mit ír a VS.NET doksi az *OldValuesParameterFormatString* jellemzőről:

```
... The default value is „orig_{0}”.
```

Ez az, ezt már szeretem. Féltreinformál minket a doksi, ez alapján okoskodhattam volna pár éjszakát. Remélem ez a kis kitérő mindenkit meggyőzött arról, hogy a doksi hasznos eszköz, de csak a Reflectornak szabad hinni. De miért is logikus az *original* szó? Update esetén előfordulhat, hogy magát az elsődleges kulcsot is módosítjuk. A sikeres SQL UPDATE parancsokhoz a szűrésben még az eredeti értékre kell hivatkozni, az *@original_MezőNévb*en ezt adja át nekünk a *DataSource* és a *DetailsView*. A dokumentáció szerint akkor használják az eredeti értéket, ha pesszimista konkurencia modellét állítottunk be az *SqlDataSource*-ban:

```
<asp:SqlDataSource
  ConflictDetection="CompareAllValues" ...
```

Ekkor a Profilerben valóban jól látható, hogy minden mezőt átadnak a módosított ÉS az eredeti állapotában is. Mi a parancsunkban a WHERE feltételben felhasználhatjuk az eredeti értékeket annak ellenőrzésére, hogy valaki más módosította-e a sort a mióta mi az letöltöttük. Ha *ConflictDetection="OverwriteChanges"*, akkor csak a módosított értékeket kapjuk meg *@mezőnév* paraméte-

rekben, DE a kulcsból mindig az eredeti értéket kapjuk meg teljesen logikusan, ezért kellett az *original_* prefixsel hivatkozni rá. A probléma másik megoldása az lenne, ha az *OldValuesParameterFormatString* értékét {}-ra állítjuk át, és a kulcsra továbbra is @ProductID néven hivatkozunk a tárolt eljárásban. Mint az előző részben is említettem ez a deklaratív programozási stílus nem feltétlen emberbarát, inkább kódgenerátor-barát. Hisz az adatbázis séma ismeretében és némi heurisztikát alkalmazva a fenti xml deklarációk zöme legenerálható. Így az előbbi összes problémával nem kell szembesülni, ha a VS.NET varázslóival hozzuk létre az adatforrást, pl. az adatbázis táblát a Server Explorerből rádobva a tervezőfelületre. Ekkor minden paraméter és SQL parancsot hibátlanul kitöltenek a varázslók.

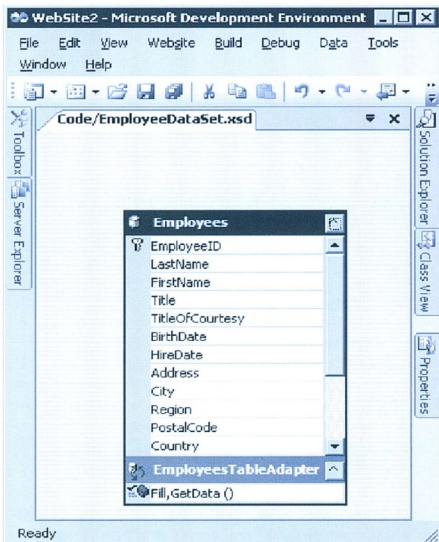
Adatelérés Data Componentsekkkel

A *DataSet*, *DataTable*, *DataRow* triumvirátus által felállított tábla alapú modell az adatokra koncentrált. Ami nem is baj, hisz általában az adatok a fontosak a számunkra. Ha nagyon sok üzleti logika (számítások, ellenőrzések, workflow, stb.) kapcsolódik az adatokhoz, akkor a *DataTable* alapú modell nehezen kezelhetővé válik. Az adatok és a rájuk értelmezett műveletek szétválnak egymástól, ami visszavezet minket a strukturált programozás idejébe, és emellett nehezen karbantartható kódot kapunk. Ha már van egy szög Objektumorientált keretrendszerünk, éljünk vele.

Első nekibuzdulásunkban azt gondolhatnánk, hogy ne dobjuk már ki a generált típusos *DataSet* által generált tehermérdék kódot, hisz abban legalább az adatábrázolás, módosítások követése, satöbbi legalább profi módon meg van oldva. A Code nevű speciális mappába felvehetünk egy új *DataComponent*-et, ez a Typed *DataSet* utóda. Ez nem csak egyszerűen egy leszármaztatott, „típusosított” *DataSet*, hanem *DataAdapterek* is hozzá vannak rendelve, azaz egy szinte egy teljes adatelérő réteget megírnak nekünk a varázslók. Például hozunk létre egy *EmployeeDataSet.xsd* nevű állományt (Project, Add New Item, *DataComponent*). A szokásos típusos *DataSet* varázslóhoz hasonló felületen kell végiggatgognunk, amiben azonban van egy érdekes ablak:

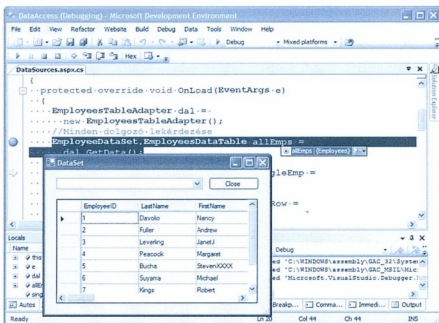
Létrehozunk adatelérő metódusokat is, nemcsak a *DataSet*-et, azaz *DataAdapterek* is készülnek a háttérben! Emlékeztetőül, a *DataComponent*-ünket a Code nevű speciális állománytába hoztuk létre, így a komponset leíró XSD alapján az ASP.NET Runtime automatikusan generálja a szükséges *DataSet* és egyéb osztályokat. Egyszerű példa a generált *Data Access Layer* használatára:

```
EmployeesTableAdapter dal =
new EmployeesTableAdapter ();
//Minden dolgot lekérdezésre
EmployeeDataSet EmployeeDataTable allEmps =
dal.GetData ();
```

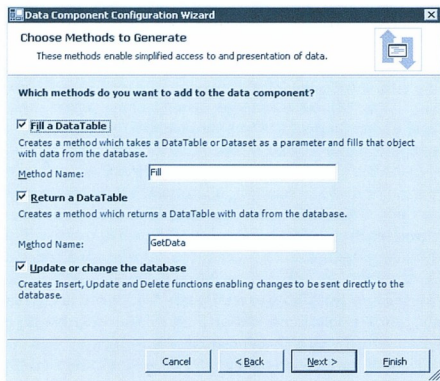


A varázsló által generált DataComponent

Az adatokat rögtön megnézhetjük a debuggerben az új *DataSet* Visualizer segítségével:



Új koncepció a debugolásban: Visualizerek

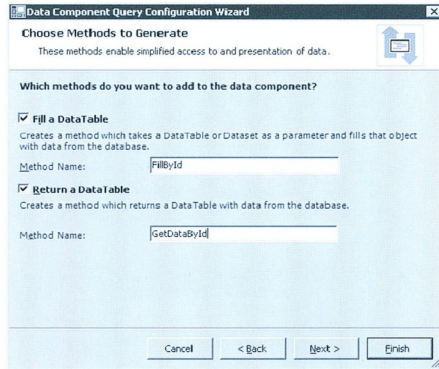


Adatelérő metódusok generálását paraméterező lépés

Ha szeretnénk egy olyan adatelőrt metódust, ami csak egy alkalmazottat ad vissza az azonosítója alapján, akkor DataComponent tervezőben az *EmployeesDataAdapter*-re kattintva felvehetünk új Queryt. A varázslónak paraméteres lekérdezést adunk meg:

```
SELECT *
FROM dbo.Employees
WHERE EmployeeID = @EmployeeID
```

Az adatelőrt metódusokat elnevezzük:



Új adatelőrt metódus felvétele a DataComponent DataAdapterébe

Használata:

```
EmployeeDataSet.EmployeesDataTable singleEmp =
    dal.GetDataById(4);
```

Ez idáig impozáns, ám ezen a ponton van egy kis szépség-hiba. Egy sort várunk vissza, ám az általános varázslógenerálta komponensek egy komplett DataTable-t adnak vissza. Kényelmetlen mindig table[0] formátummal hivatkozni az egy szem sorra.

Ha már a varázsló nem segít, segítségünk magunkon.

Tegyük fel, hogy szeretnénk kibővíteni a generált Adaptert olyan GetSingleEmployeeById metódussal, ami tényleg egy DataRow-t ad vissza. Egyik módszer a leszámraztatás volna, ám tudjuk, hogy a 2.0-ás C#-ban és VB.NET-ben van egy új lehetőség, a partial (részleges) osztály fogalom. Ez azt jelenti, hogy egy osztály definícióját több darabban is megírhatjuk. A generált kód már tartalmaz egy EmployeesDataAdapter-t, ezt kell kibővítenünk az új metódussal. A forráskódot most is Code mappába tesszük:

```
public partial class EmployeesTableAdapter
{
    public EmployeeDataSet.EmployeesRow
        GetSingleEmployeeById(int id)
    {
        EmployeeDataSet.EmployeesDataTable t;
        if ((t = GetDataById(id)) != null)
        {
            return t[0];
        }
    }
}
```

```
}
else
    return null;
}
}
```

Egyszerűen folytatjuk a tervező által generált kódot. Az új metódus így részévé vált az adapternek:

```
EmployeeDataSet.EmployeesRow singleEmpRow =
    dal.GetSingleEmployeeById(4);
```

Mint az 1.1-ben már megszokhattuk az oszlopokat típusosan, közvetlenül elérhetjük:

```
Label1.Text = singleEmpRow.Address;
```

Eddig örülünk, a varázslók elvégzik a munka dandárját, nekünk csak itt-ott finomítani kell e munkájukat.

Lelkesedésünkben eszünkbe jut, hogy az objektumorientált elvek alapján ha egy *EmployeesRow* példány modellez egy alkalmazottat, akkor a fizetésemelést, amely például a Salary adatbázisoszlop, itt *DataRow* oszlop értékén munkálkodik az *EmployeesRow* osztályban kell implementálni (nincs ilyen oszlop a Northwind adatbázis *Employees* táblájában). Korábbi sikereink által feltűzelve nekilátunk kiegészíteni a generált osztályokat:

```
public partial class EmployeeDataSet {
    public partial class EmployeesDataTable { }
    public partial class EmployeesRow {
        public void RaiseSalary() { /*... */ }
    }
}
```

És itt kezdődnek a problémák. A fordító szerint:

```
6ef48caj.2.cs(1691,18): error CS0260: Missing
partial modifier on declaration of type
'EmployeeDataSet.EmployeesRow'; another partial
declaration of this type exists
```

Ajaj, a generált kódba nem rakták be a *partial* kulcsszót az *EmployeesRow* definíciójába:

```
public class EmployeesRow : System.Data.DataRow
```

Nos, ez baj. Nagy baj. Miért nem rakták ki a *partial*? Csak nem felejtették el, hisz a *DataTable* és *DataSet*-nél ott van. A *DataRow* sajnos huncutabb másik két társánál. Ő nem létezhet csak úgy, egymagában, neki mindig szüksége van egy *DataTable*-re, aki az oszlopaít (sémát) definiálja. Emiatt nincs is nyilvános és paraméter nélküli konstruktor. Próbálkozhatnánk a hagyományos leszámraztatással is. Ez azonban szintén véres vállalkozás lenne, mert a generált, típusos *DataTable* osztály nincs felkészítve leszámrazzott sorok előállítására, ehhez *Factory Method*okkal [2] ki kellett volna nyitni a *DataTable* osztályt. Aki szereti a vért és erőszakot, az feltétlen nézze meg ezt a könyvrészletet [3], itt leírják hogyan lehet mégiscsak leszámraztatni a *DataRow*-ból, ám a végén a szerző új *DataSet* generátort írt (szó szerint).

Szóval itt most elakadtunk. Valójában érzésem szerint eddig akarta a Microsoft hagyni, hogy testre szabjuk a *DataSet* családot. Ha valóban objektumalapú megközelítésre van szükségünk, akkor bármennyire is fáj otthagyni a kényelmes varázslókkal lakályossá tett világot ki kell mennünk a puszta, és kézzel kell megírunk szinte mindent.

Az objektumorientált megközelítés

Rendben, modellezzünk egy alkalmazottat egy teljesen közönséges osztállyal:

```
public class Employee {
    protected string lastName;
    protected string firstName;
    protected decimal salary;

    public string LastName {
        get { return lastName; }
        set { lastName = value; }
    }
    public string FirstName {
        get { return firstName; }
        set { firstName = value; }
    }
    public decimal Salary {
        get { return salary; }
    }
    public void RaiseSalary(decimal amount) {
        if (amount / salary > 0.1M)
        {
            throw new ArgumentException(
                "10 %-nál magasabb fizetésemelés
                nem adható!", "amount");
        }
        else {
            salary += amount;
        }
    }
}
```

Ehhez rettentően egyszerűen hozzáadhatjuk az üzleti logikát, plusz metódusok formájában, mint a példában is látható *RaiseSalary* metódus. Közönséges objektumorientált vezetés, nem rakétatudomány.

Viszont az adatok általában relációs adatbázisban laknak, amelyek nem tudnak az objektumainkról. Hogyan lesz egy *SELECT* által visszaadott *Employee* sorból egy adatokkal feltöltött *Employee* objektum? Nos, a problémakört *Object-Relational Mapping*nek hívják, és messze sokkal bonyolultabb téma, mint azt első ránézésre gondolnánk. Emiatt most nem megyek bele az illesztés kérdésébe, de a [4] címen elérhető példakódom illetve az [5] könyv segítséget nyújthat az érdeklődőknek.

Tegyük fel az *O-R Mapper*ünktől (amit pl. vásároltunk, mert a Microsoft *ObjectSpaces* terméke még sajnos várat magára) így kapnánk vissza egy *Employee* objektumokkal feltöltött *ArrayList*-et (a példában nem használtam ki a 2.0-talán legnagyobb erejét, a generikus programozást):

```
EmployeeMapper mapper = new EmployeeMapper();
ArrayList employees = mapper.FindAll();
```

Semmi SQL, erről szó az *O-R* leképezés. Ha az alkalmazottakat meg akarjuk jeleníteni, akkor fel kell dobni egy *ObjectDataSource* komponenst a tervezőre. Elindul egy varázsló, ami két fontos dolgot kérdez: ki állítja elő az üzleti objektumokat (*EmployeeMapper*), és melyik metódusát kell ennek meghívni az adatok lekérdezéséhez (*FindAll*). Persze az *Insert*, *Update* és *Delete* is implementálható és megadható. Az *ObjectDataSource* reflection segítségével éppúgy hívogatja az adatelérő komponens metódusait, mint az *SqlDataSource* a háttér *SqlDataAdapter*-ekét. Azaz az adatok megjelenítéséhez már csak fel kell dobni egy *DetailsView* vezérlőt a lapra, és megadni neki az *ObjectDataSource*-t adatforrásként. És láss csodát, a megjelenítés teljesen intelligens módon előveszi az *Employee* objektumunk nyilvános jellemzőit, és azokat megjeleníti, teljesen hasonlóan, mint az adatsorokkal tette korábban. Azaz ha az *O-R* mapping rész készen van, az üzleti objektumok megjelenítése *ASP.NET 2.0*-val gyerekjáték.

Zárszó

Az *ASP.NET 2.0* adatelérési stratégiája jelentősen fejlődött az 1.1-hez képest. Immáron kétirányú az adatoktól, hisz láttuk, hogy pl. a *DetailsView* vezérlő könnyedén visszairja az adatokat az adatforrásába.

Magasszintű vezérlőket kapunk az adatok megjelenítéséhez, amelyek képességei mesze sokkal többek, mint amit e cikk bemutatott, ezek felfedezését a kedves olvasókra bízom. Cikksorozatomban következő részében a felhasználók számára testreszabható weboldalak írásához támogatást nyújtó *Personalization*, *Membership* és *Role Manager* architektúrával fogok foglalkozni. Addig is mindenkit bátorítok, hogy a példákat [6] próbálja ki az egyelőre ingyen letölthető *Visual Studio Express* változatokkal [7].

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] www.aisto.com/roeder/dotnet/
- [2] www.netacademia.net/workshop/dp/
- [3] www.adoguy.com/content.aspx?id=SampleChapter/Chapter1
- [4] netacademia.net/tudastar/articlepage.aspx?upid=2481
- [5] <http://martinfowler.com/books.html#eaa>
- [6] netacademia.net/tudastar/articlepage.aspx?upid=3681
- [7] <http://lab.msdn.microsoft.com/express/>

Windows szolgáltatók 4. rész

A LOCALSYSTEM FIÓKOT HASZNÁLÓ SZOLGÁLTATÁSOK

Eddig a Windows Server 2003 mintegy 100 szolgáltatójának egyötödét „tudtuk le” ebben a sorozatban. Az eddig felsorolt szervizek közös tulajdonsága az volt, hogy nem a LocalSystem fiókot használták. Most felmérjük, miért ekkora sláger ez a fiók és melyek azok a szervizek, amelyek csak ezzel a hagyományos szolgáltatásfiókkal üzemelnek.

Miért hagyományos?

Azért, mert a Windows XP és a Windows Server 2003 előtt gyakorlatilag csak ez a szolgáltatásfiók létezett. A klasszikus NT-kben (ide sorolandó ebből a szempontból a Windows 2000 is) a háttérben futó szolgáltatások ennek a fióknak a „virtuálisan” határtalan jogosultságait használták. Biztonsági szempontból ez nem túl okos dolog, nyilván ezért lett néhány szolgáltatás „átállítva” a sorozat előző részeiben már tárgyalt két új – LocalService ill. NetworkService – fiókra, amelyeknek már korlátozott lehetőségeik vannak. Azonban ha jól megnézzük a szervizek listáját, azt láthatjuk, hogy túlnyomó többségük még mindig ezt a fiókot használja a Windows Server 2003-ban is, olyan alapvető fontosságúak is, mint például a DHCP Server/Client, a DNS Server, az összes IIS komponens, az RRAS, vagy akár az RPC. Szinte biztosak lehetünk abban is, hogyha telepítünk egy külső programot, amely a céljai eléréséhez rendelkezik egy szervizzel, akkor az is ezt a fiókot fogja használni. Ergo, előljáróban tekintsük át néhány fontos jellemzőjét és tulajdonságát eme fióknak, mielőtt a nevében futó szolgáltatásokkal elkezdünk foglalkozni.

A LocalSystem fiókról

Ez a fiók ugyanaz, mint amellyel a Windows 2000-től kezdve a user módú operációs rendszer komponensek futnak, beleértve a, Session Manager-t (Windows\System32\Smss.exe), a Win32 alrendszer processzt (Csrss.exe), és a helyi biztonsági alrendszert (Lsass.exe), és a Winlogon processzt (Winlogon.exe). Mint már volt róla szó, a LocalSystem fiók eléggé „izmos” jogosultsági szempontból, a lokális rendszert tekintve sokkal erősebb mint bármelyik helyi vagy tartományi felhasználói fiók. A következő fő jellemzői miatt igaz ez az állítás:

- A legtöbb állományhoz és regisztrációs adatbázis kulcsokhoz alaptól teljes körű hozzáférése van, ám ha történetesen nincs, akkor is van, mert az ezzel a fiókkal futó processzek „át tudják venni” (take-ownership) az objektumok tulajdonjogát, így rögtön elérhetővé válik a hiányzó jogosultság is.

- A fiókot használó folyamatok az alapértelmezett felhasználói profil alatt futnak (HKEY_USERS\DEFAULT), ezért nincs lehetőségük elérni azokat konfigurációs beállításokat (azaz nem is hatnak rájuk), amelyeket az operációs rendszer üzemeltetője a többi felhasználói profilra kötelezően előír.
- Ha a gép egy Windows 2000/2003 tartomány tagja, akkor ennek a fióknak további, a lokális gépen kívüli, más gépekre is kiható tulajdonságai is előtérbe kerülhetnek. Például az, hogy a fiók tartalmazza annak a gépnek a biztonsági azonosítóját (SID) amelyiken fut az adott szerviz, ezért automatikusan és probléma nélkül „hitelesítődik” ennek az azonosítónak a segítségével egy bármelyik másik gépen, amelyik ugyanabban az erdőben található.
- A helyi rendszergazda csoport tagja, ám jónéhány olyan privilégiuma is van, amely ezen csoport többi tagjainak alaptól nincs. Nézzük néhányat az e fiók által birtokolt jogosultságok közül, amelyek nem szokványos volta miatt szeretnek a szolgáltatások ezen fiók égisze alatt futni:

Act As Part Of The Operating System

Magyarul az operációs rendszer részeként való működés, azaz egy processz számára lehetőség arra, hogy a rendszerben lévő bármelyik felhasználó nevében azonosíthassa magát, így hozzáférve gyakorlatilag bármelyik lokális erőforráshoz.

Creae A Token Object

Egy hozzáférési token létrehozására szolgáló jog. Például az Active Directoryban a felhasználó hitelesítése után, a helyi biztonsági rendszer létrehoz a felhasználó számára egy tokenet, azaz egy olyan speciális „csomagot”, amely (többek között) tartalmazza a felhasználó nevét, SID-jét és a felhasználót tartalmazó csoportok SID-jeit is. Innenlő a rendszer használatának lehetőségei az ebben a tokenben tárolt információktól függenek. Ennek kreálásához megint csak kell ez az erős jogkör.

Log On As A Service

Jogosultságok megadása, abból a célból, hogy egy-egy felhasználói fiók egy szolgáltatás biztonsági kontextusának előnyeit élvezve léphessen be a rendszerbe. A LocalSystem fiók mindig így lép be (ezt nem is lehet megváltoztatni), azonban alapértelmezés szerint semmilyen fiók nem kapja meg ezt a lehetőséget, ezért a külön fiókot igénybe vevő alkalmazásnak kell erről a telepítés során gondoskodni, vagy ha például egy Exchange Servert saját fiókkal futtatunk, akkor nekünk kell megadni a fióknak ezt az engedélyt a vonatkozó csoportházi rendben, ezen a helyen:

```
Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment
```

Creare Permanent Shared Objects

Ez a jogosultság azt definiálja, hogy mely fiókok „gyárthatnak” címirt objektumot az Object Managerben. Alapértelmezés szerint erre csak azok a kernel módú komponensek képesek, amelyek a LocalSystem fiók biztonsági környezetét használják. Ezen kívül nincs is másik fiók, amely rendelkezne ezzel a joggal.

Generate Security Audits

Meghatározza azon fiókokat, amelyeket használó processzek képesek a biztonsági naplózó bejegyzéseket hozzáadni. Ez is egy igen fontos lehetőség, gondoljunk bele, mi minden múlhat ezen bejegyzések értelmezésén. Alapból ezt is csak a LocalSystem „tudja”.

Replace A Process Level Token

Ennek a jogosultságnak is kizárólagos tulajdonosa a LocalSystem fiók, csak azok ennek nevében lehetséges kezdeményezni egy olyan processzt, amely kicseréli egy másik futó processz meglévő hozzáférési tokenjét.

Néhány praktikum

Mivel ennek a fióknak az alapértelmezés szerinti jogosultsági köre sok esetben túlságosan is kiterjedt, ezért ha indokoltnak látjuk az adott szolgáltatásnál (pl. az Exchange Server szolgáltatások esetén), lecserélhetjük egy általunk kreált felhasználó fiókra. Viszont ebben az esetben ennek a fióknak meg kell adnunk azokat a jogosultságokat és házi-remény engedélyeket, amelyek szükségesek a korrekett működéshez. Egy másik megszorításról is esszen szó az ezzel a fiókkal futó szolgáltatások esetén, ez pedig az interaktivitás. Pontosabban a hiánya, ugyanis nem képes pl. párbeszéd-panelet vagy egyéb ablakokat kirakni a rendszert éppen használó szeme elé (néhány üzenet kivételével), azaz beavatkozásra sem kerülhet sor.

Ha odavagyunk a kihívásokért és szeretnénk megtekinteni a regisztrációs adatbázis azon részét, amelyet még a rendszergazdai fiókkal belépve sem láthatunk, akkor indítsuk el a következő paranccsal a szerkesztőt (ha éppen 11:10 a pontos idő):

```
at 11:11 /interactive regedt32.exe
```

Ennek hatására időzítve indul el a szerkesztő a LocalSystem fiók nevében, mivel a Task Scheduler szerviz ezzel a fiókkal fut. Egyfajta „Run as...” ez, és arra jó, hogy belekukantsunk pl. a HKLMSAM és a HKLMSECURITY ágakba

(Fóti Marcell tollából erről korábban már lehetett olvasni ebben az újságban [1]).

Az ebben a számban bemutatott LocalSystem fiókot használó szolgáltatások a következők lesznek:

- Application Management
- Automatic Updates
- Background Intelligent Transfer Service
- ClipBook
- COM+ Event System
- COM+ System Application
- Computer Browser

Application Management

(Alkalmazásvezérlés)

A szerviz rövid neve: AppMgmt

Az alkalmazás neve: appmgmts.dll (svchost.exe)

Függés: –

Függesztés: –

Porthaszálat: –

Alapértelmezett indítás: kézi, leállítva

Ennek a szerviznek a szoftvertelepítésben van szerepe. Valószínűbben tudják, hogy tartományi körülmények között a Csoportházi rend segítségével telepíthetünk és frissíthetünk központi helyről programokat. Ezeket a speciális telepítősomagokat hozzárendeltjük kötelezően a számítógépekhez és felhasználókhoz (Assign), vagy közzé tehetjük számukra az Add/Remove Programs alatt (Publish) a felhasználói részére. Amikor ez utóbbit tesszük és a felhasználó boldogan kattint az „Add...” gombra a fentebb említett ablakban, akkor kezd el működni ez a szolgáltatás és átadja a programnak az információkat a számára engedélyezett telepítősomagok hollétéről, státuszáról, stb. Akkor is dolgozik ez a szerviz, ha csak szimplán telepítünk vagy eltávolítunk valamilyen szoftvert a rendszerből, és még abban az esetben is, ha a felhasználói felületről egy telepítési kérés indul el egy kiterjesztés alapján egy alkalmazás felé, amely erre a hívásra aztán szépen települ is (ez szintén egy Csoportházi rendben szabályozható módszer). A szolgáltatás (mivel kézi indítású) igény szerint indul, de nem áll le, akkor sem, ha befejeződött a művelet, azaz „úgy marad” a következő számítógép újraindításig.

Ha letiltjuk, akkor a fent ismertetett tartományi telepítési lehetőségek természetesen nem működnek, az Add/Remove Programs ablakban az „Add New Programs” ikon alatt a „No programs are available on the network” felirat fog ékeletlenkedni, akkor is, ha lenne telepítésre váró alkalmazás. Ha viszont nincs szükségünk rá vagy nem is lehetséges ezt a szolgáltatást használnunk (pl. nem is tartományi tag a gépünk), akkor nyugodtan, sőt ajánlottan tiltsuk le.

Automatic Updates

(Automatikus frissítés)

A szerviz rövid neve: Wuauerv

Az alkalmazás neve: appmgmts.dll (svchost.exe)

Függés: –

Függesztés: –

Porthaszálat: TCP 80

Alapértelmezett indítás: automatikus

Ez egy közismert – a Windows 2000 SP3 és a Windows XP SP1 által telepített szerviz (a Windows Server 2003 már alapértelmezés szerint tartalmazta), amely lehetővé teszi, hogy

automatikusan frissítéseket, bővítéseket és meghajtó programokat tölthessünk le az internetről, pontosabban a Windows Update szerverekről egy kliens alkalmazás segítségével, a Windows Update weblap használatával, automatikusan, vagy a helyi SUS (Software Update Service) szerverről, szintén automatikusan, ám központilag szabályozva.

A félreértések elkerülése végett szét kell választanunk a frissítési típusokat. Abban az esetben ha a Windows Update weboldalt használjuk (amely egyébként mostanság alakul át, a Windows XP SP2-es gépek már a napokban debütált v5-ös változatot használják), manuális beavatkozásra is szükség van, azaz a böngészőnkön keresztül mi választunk, hogy mit szeretnénk telepíteni a biztonsági frissítésekből, a bővítvényekből vagy a eszközmeghajtók frissítéseiből. A fent említett első esetben az AU kliens csak a kritikus biztonsági frissítéseket telepíti, azokat azonban beavatkozás nélkül, akár helyileg szabályozva a System Properties panelen keresztül, akár pl. a Csoportházi rend alapján egy tartományban. Ez a kliens tarthatja a kapcsolatot a második pontban említett SUS szerverrel is, szintén a Csoportházi rendben központilag beállított tulajdonságok alapján vagy ennek hiányában a regisztrációs adatbázis „megpiszkálásával”. Az AU kliens rendelkezik még néhány érdekes és értékes tulajdonsággal. Például a kliensoldali beállításához rendszergazdai jogosultság kell, valamint a frissítések telepítése előtt ellenőrzi, hogy a csomagok rendelkeznek-e a Microsoft digitális aláírásával. Fontos az is, hogy az egyszerre több frissítés telepítésére kerül sor, és ezek közül az egyik a gép újraindítását igényelné, nem kell megszakítanunk a telepítést, csak az összes telepítés után lesz szükség az újraindításra.

A szerver leállítása vagy letiltása – közvetlenül – nem okoz semmilyen fennakadást az operációs rendszer működésében, és természetesen a böngészőből végzett manuális frissítést sem akadályozza meg, azonban önálló, nem védett hálózatok tagjaként működő, internet hozzáféréssel rendelkező gépeken mindenképpen javallott használni a biztonsági szint emelésére.

Background Intelligent Transfer Service

[Háttérben futó intelligens átviteli szolgáltatás]

A szerviz rövid neve: BITS

Az alkalmazás neve: qmgr.dll (svchost.exe)

Függés: Remote Procedure Call

Függésztés: -

Porthasználat: TCP 80

Alapértelmezett indítás: kézi, leállítva

A BITS egyidős az Automatic Update szolgáltatással (ugyanazokban a javítócsomagokban vdt először megtalálható), hiszen ez utóbbi megoldás érdekében született. Egy érdekes szervizről van szó, amelyet kifejezetten a frissítések letöltésének optimalizálására fejlesztettek ki. Aszinkron állományátvitelt valósít meg egy HTTP szerver és a kliens között, úgy hogy megpróbál takarékoskodni a sávszélességgel. Mindezt a felhasználó tevékenységétől függetlenül végzi a háttérben, ám a felhasználó internet használatának figyelembevételével. A hozzá érkező, letöltésekre vonatkozó kérések egy ún. queue managerbe kerülnek, majd a sávszélesség felszabadulásával végrehajdnak (alapértelme-

zés szerint maximum 90 napig tárolja a kéréseket). Amennyiben a felhasználó elkezdli igénybe venni a sávszélességet, a BITS lefejezi a letöltést. Ha az internetes hozzáférés megszűnik vagy a felhasználó kilépi illetve a gépet kikapcsolja, akkor a BITS elmenti a letöltéseket, azért, hogy a következő belépéskor folytathassa azokat. A queue manager működésébe a felhasználó is beleszólhat, azaz elindíthatja/leállíthatja a letöltéseket, fel- és beállíthatja a különböző szabályokat és prioritásokat a letöltési műveletekkel kapcsolatban, megnézheti a különböző letöltések státuszát, illetve beállítható az is, hogy mely folyamatokról kapjunk visszajelzéseket.

Egy, az előtérre (ez a legmagasabb prioritás) és három, a háttérre vonatkozó prioritási szintet határozhatunk meg a letöltéseket illetően. Így a magasabb prioritású kapott processzek vagy hálózati alkalmazások többet kapnak a sávszélességből (gyr persze előzik is a többit), az alacsonyabbak pedig várhatnak a sorukra. Ám van egy kis demokrácia is a rendszerben, ugyanis az azonos prioritású élvezők megosztják a hozzáférést és hogy ne előzzék be jelentősen egymást, egy round-robin elvet használó időzítő cserélgeti a hozzáférést számukra.

A szerviz a LocalSystem fiókot használja, azonban az alapértelmezés szerint a BITS folyamatok az aktív felhasználó biztonsági környezetében működnek, illetve beállítható a proxyra vonatkozó hitelesítési információ. A BITS támogatja a Basic, az NTLM, a Digest és a Passport hitelesítést és a protokollok közül a HTTP-t és a HTTPS-t egyaránt. A beállításokra a GUI-n működő alkalmazás híján a bitsadmin.exe [2] parancsori programot használhatjuk (igaz kicsit nehezebben, mert elsőre rémisztő a rengeteg opció). Ez az eszköz az adott operációs rendszer Support Tools csomagjában található meg (az RTM XP-ben az 1.0, a Windows Server 2003-ban az 1.5), és a SUS csomag is tartalmazza. Jó tudni, hogy az 1.5-ös verzió már a Csoportházi rendből is szabályozható, igaz csak szerényen (engedélyezés/tiltás illetve a 90 napos gyári inaktivitás érték csökkenthető), valamint már felfelé is tud tölteni, persze csak akkor, ha van IIS és telepítettük a hozzá való BITS Server Extension-t. A Windows XP SP2-vel megjelent a BITS 2.0 változata, amellyel egy új opció is megjelent a Csoportházi rendben – a maximális sávszélesség használat mértéke. A szolgáltatás igény szerint, hívásra indul, majd amikor az összes letöltés megtörténik, automatikusan le is áll. Ha leállítjuk vagy letiltjuk, akkor értelemszerűen pl. az Automatikus frissítés szolgáltatás vagy a MSN Explorer nem tudja kihasználni az automatikus letöltések optimalizálására, azaz ezek helyett marad a hagyományos letöltési mechanizmus, amelyet kézzel nekünk kell indítani, hiszen a BITS nem adja át pl. az Internet Explorernek a queue managerbe berakott feladatokat.

Clipbook [Vágókönyv]

A szerviz rövid neve: ClipSrv

Az alkalmazás neve: clipsrv.exe

Függés: Network DDE, Network DDE DSDM

Függésztés: -

Porthasználat: -

Alapértelmezett indítás: letiltva

Talán egy kevésbé izgalmas szolgáltatás következik, amely lehetővé teszi a Clipbook Viewer számára azt, hogy létrehozzon és megosszon „oldalakat” amelyeket egy távoli szá-

mítógépről ezek után megtekinthetők lesznek. Ezt kifejezetten a Távolsi asztal vagy egy terminálkapcsolat esetében használjuk a vágólap megosztására. A megosztott vágólap tartalmát a rendszer szinkronizálja a helyi vágólapéval, azonban átlományok és mappák átvitelére nem, csak szöveges és képi adatok kezelésére használhatjuk. A szerviz alapértelmezés szerint le van tiltva, ezért ekkor a vágólap is csak helyben használható. Ha nem muszáj, ne is változtassunk ezen az állapoton.

COM+ Event System (COM+ Eseményrendszer)

A szerviz rövid neve: *EventSystem*

Az alkalmazás neve: *es.dll (svchost.exe)*

Függés: *Remote Procedure Call*

Függésítés: *System Event Notification, Window Internet Name Service, DHCP Server, COM+ System Application*
Porthasználat: –

Alapértelmezett indítás: *kézi, elindítva*

Ez a szerviz a programozási oldalhoz áll közelebb inkább, és az üzenetkezelésben van komoly szerepe. A COM+ eseményszolgáltatói (publishers) által generált értesítéseket kapják meg az erre „feliratkozók” (subscribers) ennek a szolgáltatásnak a segítségével. A szerviz tárolja is ezeket a különböző eseményszolgáltatóktól érkező eseményinformációkat, abból a célból, hogy a subscriberek által lekérdezhetővé váljanak. Gyakorlatilag ez a szerviz egy COM+ alkalmazás, ami csak azért vált szolgáltatássá, hogy rezidensen elérhető legyen. Ha letiltjuk, akkor néhány alkalmazás (WINS, DHCP Server) biztosan nem működik tovább, de pl. a Volume Snapshot szolgáltatás, vagy a Media Player sem, illetve még olyan extrém szoftverekkel is lesz probléma, mint a Bootvis.exe. Valamint, ebben az esetben egy visszatérő DCOM hiba bejegyzéssel is számolhatunk az eseménynaplóban.

COM+ System Application (COM+ rendszeralkalmazás)

A szerviz rövid neve: *COMSysApp*

Az alkalmazás neve: *dllhost.exe*

Függés: *Remote Procedure Call, COM+ Event System*

Függésítés: –

Porthasználat: –

Alapértelmezett indítás: *kézi, leállítva*

A COM+ rendszeralkalmazás COM+ szolgáltatásokat tárol, valamint az alkalmazások konfigurációját és nyomkövetését menedzseli. Ha letiltjuk, ezek az alkalmazások nem működnek tovább, azaz az operációs rendszer számára lehetetlenné válik bármilyen COM+-t használó program vagy OLE regisztráció üzemeltetése.

Computer Browser (Számítógép tallózó)

A szerviz rövid neve: *Browser*

Az alkalmazás neve: *browser.dll (svchost.exe)*

Függés: *Server, Workstation*

Függésítés: –

Porthasználat: *TCP: 139, UDP: 137, 138*

Alapértelmezett indítás: *automatikus*

Egy sokszor előtérbe kerülő és sokat vitatott szolgáltatásról van szó, amelyet inkább negatív felhanggal szoktak emle-

getni (szerintem is joggal). A Windows 3.11 óta az egy hálózatban lévő tartományok, munkacsoportok és számítógépek listázásra/tallózására használható mind a felhasználó mind az alkalmazások (pl. a My Network Places, vagy a NET VIEW parancs) kiszolgálása érdekében. Azok a számítógépek, amelyek ezt a szerepet a speciális szabályozású „választás” (Election) folyamat során elnyerik, tallózó szerverekké válnak és töltik kérdezhető le ettől fogva a tallózó lista. A tallózó szervereknek többféle típusa van, közös jellemzőjük viszont, hogy simán és automatikusan (pl. akár egy kisebb leállás vagy működési hiba miatt) elveszthetik az adott kiemelt szerepüket, hiszen távollétükben nem szólhatnak bele az ad hoc választásba. A szerepek a következők lehetnek:

■ Domain Master Browser

Ez a legmagasabb egyben az egyetlen egyedi szerepkör az egész hálózatban. Alapértelmezés szerint egy tartományvezérlő birtokolja (a munkacsoportokban nem is lehet ilyen), amely összegyűjti az esetlegesen külön alhálózaton működő Master Browserok tallózó listáit.

■ Master Browser

Ennek a funkciónak a birtokosa hajt a legtöbbet. Az adott alhálózaton összegyűjti a gépek adatait, aztán elküldi a hierarchiában alatta lévő Backup Browsereknek, valamint a felette lévő Domain Master Browserrel is tartja a kapcsolatot. Alhálózatonként egy-egy kell belőle, mivel a tallózó információk begyűjtése broadcast alapú, ezért a Master Browserok nem tudnak másik alhálózatban szorgoskodni.

■ Backup Browser

Nem sok teendőkük van, gyakorlatilag csak megőrzik a Master browserektől kapott tallózási listákat, de esetenként ki is szolgálhatják ezekkel az információkkal a hozzájuk forduló klienseket ezért ezek az ún. *Potential Browser* szerepet is betöltik.

■ Nonbrowser (Browser Client)

Mindegyik a hálózatban működő gép, amely nem birtokolja a fenti funkciókat és nem is léphet elő tallózóvá.

Az, hogy melyik szerepet ki nyeri el az adott hálózatban, az a kifejezetten erős broadcast forgalommal járó választási eseménytől függ, melynek még az is hátránya, hogy bármelyik számítógép kezdeményezheti! A választásnál előny (azaz több pontot jelent) a különböző már meglévő funkciók (master browser, PDC, WINS) a nagyobb verziószám, az újabb szervizcsomagok megléte, vagy a uptime mértéke de há! Istennek az általunk elhelyezett registry kulcsok is befolyásolhatják a döntést. A folyamat ennél lényegesen bonyolultabb [3], de egy szó mint száz, ezen kritériumok alapján egy pontverseny alakul ki a gépek (de azért elsősorban és optimális esetben a szerverek) között, és végül az nyer, amelyeknek a legtöbb pontja van.

Ha leállítjuk vagy letiltjuk ezt a szolgáltatást, akkor a tallózás lehetetlenné válik. Mondhatnánk azt, hogy ez nem baj, ezt az elavult rendszert ma már ne használjuk, de sajnos (főleg kompatibilitási okokból) még egy Windows Server 2003 alapú tartományban is szükség van, szükség lehet rá.

GÁL TAMÁS

MCSE 2003, MVP (Software Distribution)

gtamas@tjszki.hu

A cikkben szereplő URL-ek:

- [1] <http://www.netacademia.net/publikacio/cik/ doc/0110sid.doc>
- [2] <http://tinyurl.com/4thd5>
- [3] <http://tinyurl.com/3zp2k>

Rendszermonitorozás a .NET-ben I.rész

RENDSZERFOLYAMATOK ÉS SZOLGÁLTATÁSOK MONITOROZÁSA

A .NET Framework számos olyan eszközt biztosít a fejlesztők számára, amelyek segítségével alkalmazásaik hozzáférhetnek az operációs rendszer folyamataihoz és szolgáltatásaihoz, megváltoztathatják azok állapotát, illetve lekérdezhetik tulajdonságaikat.

Cikkünkben részletesen megvizsgáljuk azokat a komponenseket, amelyek segítségünkre lehetnek a folyamatok és szolgáltatások (Windows szervizek) kezelésében. Áttekintjük a Process és ServiceControler komponensek létrehozásának lépéseit, módszereit és tulajdonságait használatát.

A Process komponens segítségével a következő műveleteket fogjuk elvégezni:

- Folyamat elindítása és leállítása a helyi gépen
- Folyamat listázása a helyi gépen és távoli gépeken
- Adott folyamat által használt modulok listázása
- Adott folyamat által indított szálak listázása
- Futó folyamat tulajdonságainak lekérdezése a helyi gépen és távoli gépeken

A ServiceController komponens a következő műveleteket teszi lehetővé:

- A szolgáltatások és eszközközkezelők listázása a helyi gépen és távoli számítógépeken.
- Szolgáltatások és eszközközkezelők indítása leállítása, felfüggesztése
- Szolgáltatások és eszközközkezelők tulajdonságainak lekérdezése

Az [1] címről letölthető mintaprogram bemutatja a .NET rendszerfolyamatokkal és szolgáltatásokkal kapcsolatos képességeit, mintát ad valamennyi tárgyalt művelet végrehajtására. A program konzol alapú, menüszerkezete a következő:

```

c:\ Parancssor - procmon
D:\>procmon
=====
0 - Folyamatok listázása
1 - Folyamat tulajdonságai
2 - Folyamat indítása
3 - Folyamat leállítása
4 - Szolgáltatások listázása
5 - Eszközközkezelők listázása
6 - Szolgáltatás / eszközközkezelő tulajdonságai
7 - Szolgáltatás / eszközközkezelő indítása
8 - Szolgáltatás / eszközközkezelő leállítása
9 - Kilépés
|

```

■ A mintaprogram menüszerkezete

Természetesen egy GUI-t használó alkalmazás sokkal látványosabb, és könnyebben használható lenne, de szükség szerűen jelentősen több, a téma szempontjából teljesen fölösleges kódot kellene tartalmaznia. Céloom nem az áttekinthető és jól használható felület, hanem az áttekinthető és jól használható programkód létrehozása volt, ezért döntöttem a kb. 30 sorban megvalósítható menüszerkezetre, és a lehető legegyszerűbb hibakezelés mellett.

Process komponensek létrehozása

A Process komponensek segítségével könnyen és gyorsan elvégezhetjük a Windows rendszerfolyamataival kapcsolatos műveleteket. A helyi és távoli gépeken egyaránt lekérdezhetjük a futó folyamatok tulajdonságait, a helyi gépen pedig elindíthatunk, és le is állíthatunk rendszerfolyamatokat. Process komponens a System.Diagnostics névtérben található Process osztály példányosításával hozhatunk létre:

```

using System.Diagnostics;
Process proc = new Process();

```

A létrejövő komponens biztosítja a kapcsolatot az alkalmazás és a számítógépen futó folyamat között. Hogy Process komponensünk használható legyen, meg kell adnunk azt a fizikai folyamatot is, amellyel kapcsolatban fog állni. Alapvetően két lehetőség áll rendelkezésre; a komponens segítségével új rendszerfolyamatot indíthatunk (lásd később), illetve kiválaszthatjuk a rendszerben futó folyamatok közül a megfelelőt, és létrehozhatjuk a hozzá csatolt komponens. A csatolást a Process osztály statikus metódusainak segítségével végezhetjük el, a következő módon:

```

Process[] proc;
proc = Process.GetProcessesByName(processName);

```

Látható, hogy a metódus nem egyetlen Process objektumot, hanem ilyen objektumokból álló tömböt ad vissza. Ennek oka egyszerűen az, hogy a paraméterül megadott folyamatnév (ProcessName) nem egyedi tulajdonság, a rendszerben akárhány azonos nevű futó folyamat lehet. A ProcessName

tulajdonképpen a folyamathoz tartozó végrehajtható fájl nevével tartalmazza, kiterjesztés és útvonal nélkül. Az egyes folyamatokkal kapcsolatban álló Process objektumok a tömb elemei. A következő kódrészlettel például a rendszerben futó összes „Notepad” nevű folyamat PID-jét listázhatjuk ki:

```
Process[] proc;

//A Process osztály statikus metódusa Process
//objektumokból álló tömböt ad vissza. A metódus
//számítógépen futó folyamatok közül azokat
//válogatja ki, amelyek ProcessName tulajdonsága
//megegyezik a paraméterként kapott sztringgel.
proc = Process.GetProcessesByName("Notepad");

//A tömb elemein végiglépkedve lekérdezzük azok
//Id tulajdonságát.
foreach(Process notepad in proc) {
    Console.WriteLine(notepad.Id.ToString());
}
```

Ha egyetlen objektumot szeretnénk visszakapni, akkor egyedi tulajdonságot kell megadnunk, mégpedig az előbb említett folyamat-azonosítót (Id, Int32):

```
Process proc;
proc = Process.GetProcessesById(processId);
```

Minkét esetben második paraméterként megadhatjuk annak a gépnek a nevét vagy IP-címét, amelyen a megcélzott folyamat fut. Ha távoli számítógépet adunk meg, akkor a folyamatok listázására és tulajdonságok lekérésére van lehetőség (folyamatot indítani és leállítani nem lehet). Alapértelmezés szerint a második paraméter természetesen a helyi számítógép.

Folyamatok indítása és leállítása

Ha a Process komponens új folyamat indítására szeretnénk felhasználni, négy különböző Start() metódus közül is válogathatunk. A paraméterek nélküli Start() metódust akkor használhatjuk, ha korábban már létrehoztuk a Process komponens, és beállítottuk annak StartInfo tulajdonságát a megfelelő értékekre:

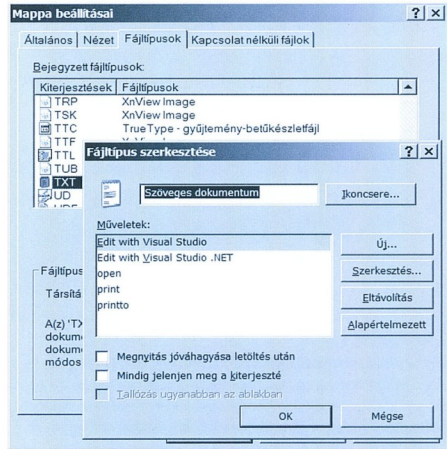
```
Process proc = new Process();
proc.StartInfo.FileName = "notepad";
proc.Start();
```

Hasonlóan a Windows Start menü Futtatás (Run) parancsához, a StartInfo.FileName tulajdonság értékeként nem csak végrehajtható fájl nevét adhatjuk meg. Bármilyen adatfájl neve megadható, amelynek kiterjesztése hozzá a rendeltve valamely programhoz, a fájl az adott program megfogja nyitni, sőt más műveletek is elvégezhetőek. A következő kóddal például a megadott dokumentumot nyomtathatjuk ki, a doc kiterjesztéshez regisztrált Word segítségével:

```
Process proc = new Process();
proc.StartInfo.FileName = "c:\marketing.doc";
proc.StartInfo.Verb = "Print";
proc.Start();
```

A megadható műveleteket a ProcessStartInfo objektum (a Process komponens StartInfo tulajdonsága) ilyet ad vissza) Verbs tulajdonsága tartalmazza, ez a kiterjesztéstől, és a számítógépre telepített programoktól is függ, például az én

gépemen doc kiterjesztés esetében Open, Print, Printto, és VirusScan. A fájlkiterjesztésekhez tartozó műveleteket megnevezhetjük, és újakat is felvehetünk a Fájlípusok Intéző (Explorer) ablakban:



Fájlkiterjesztésekhez tartozó műveletek szerkesztése

A Process osztály három statikus Start metódusa sztring vagy ProcessStartInfo objektum paraméterrel hívható, és Process objektumot ad vissza. A legegyszerűbb változat a következő:

```
Process proc = Process.Start("Notepad");
```

A Process komponens két metódust kínál a folyamatok bezárására. Ha a folyamat rendelkezik felhasználói felülettel, a CloseMainWindow() metódust használhatjuk, amelynek hatása megegyezik a folyamat ablakának bezárásával. Természetesen ilyenkor lefutnak a folyamat szokásos kilépési metódusai is, például rákérdezhet adatok elmentésére stb. Ha a folyamatnak nincsen felhasználói felülete, a Kill() metódus használatával zárhatjuk be, és ugyancsak a Kill() metódust kell használnunk, ha a folyamatot mindenképpen azonnal be szeretnénk zárni, tekintet nélkül a zárások elvégzendő műveletekre. A Kill() metódus akkor is bezárja a folyamatot, ha elmentetlen adatai vannak, vagy módálisan megnyitott ablak van fölötte. A következő kód bezárja az összes Notepad nevű folyamatot:

```
Process[] proc;
proc = Process.GetProcessesByName
("Notepad");
//van a megadott néven folyamat
if (proc.Length > 0)
    foreach(Process p in proc) {
        //ha sikerül szabályosan
        //bezárni, akkor OK
        p.CloseMainWindow();
        //ha nem, akkor Kill
        if (!p.HasExited)
            p.Kill();
    }
```

Folyamatok listázása

A rendszerben futó folyamatok listázásához a Process osztály GetProcesses() statikus metódusát kell felhasználnunk, amelynek visszatérési értéke a rendszerben futó folyamatokhoz tartozó Process komponenseket tartalmazó tömb. Az alábbi kódrészlet kiírja a folyamatok pid-jét és nevét is:

```
void processList(){
    //a processes nevű tömböt feltöltjük
    //a rendszerben futó folyamatokhoz csatolt
    //Processkomponensekkel
    Process[] processes = Process.GetProcesses();
    //végiglépkedünk a processes tömb elemein
    foreach(Process process in processes) {
        //pid
        String pid = process.Id.ToString();
        //a folyamat neve
        String name = process.ProcessName;
        System.Console.WriteLine(pid.PadRight(5) +
            % "\t" + name.PadRight(20));
    }
}
```

Folyamatok tulajdonságai

A Process objektum segítségével lehetőségek van a rendszerfolyamatok tulajdonságainak lekérdezésére. A legfontosabb lekérdezhető tulajdonságok a következők:

- **WorkingSet** – a folyamat által felhasznált fizikai memória bájtnban
- **VirtualMemorySize** – a folyamat által igényelt összes virtuális memória
- **PagedMemorySize** – a folyamat által felhasznált memória kilapoazható része
- **NonpagedSystemMemorySize** – a felhasznált memória nem kilapoazható része
- **PrivateMemorySize** – más folyamatokkal meg nem osztható memória mennyisége
- **UserProcessorTime** – a folyamat által felhasznált összes processzoridő azon része, amikor az alkalmazás saját kódja fut (nem az operációs rendszerhez tartozó kód)
- **PrivilegedProcessorTime** – a folyamat által felhasznált processzoridő azon része, amikor az operációs rendszer kódja fut
- **TotalProcessorTime** – a folyamat által felhasznált összes processzoridő (az előző két érték összege)
- **BasePriority** – ez a tulajdonság határozza meg, hogy az adott folyamat milyen prioritási szinttel indítja a létrehozott szálakat
- **Threads** – a tulajdonság a folyamat által futtatott szálakat adja vissza, ProcessThread objektumokból álló tömb képében
- **Modules** – a tulajdonság ProcessModule objektumokból álló tömböt ad vissza, amely tartalmazza a folyamat által használt függvénykönyvtárakat.

A következő kódrészlet kilistázza a megadott folyamat legfontosabb tulajdonságait:

```
void processProperties(String processName) {
    Process[] proc = Process.GetProcesses
        ByName(processName);
    if (proc.Length > 0) {
        //a lefoglalt memória
```

```
String mem = proc[0].WorkingSet/1024 + "";
//az összes igénybevett processzoridő
String procTime =
proc[0].TotalProcessorTime.ToString();
//a futtatott szálak száma
ProcessThreadCollection threads =
proc[0].Threads;
String threadCount = threads.Count+"";
//alapprioritás
int procPriority=proc[0].BasePriority;
//indítás időpontja
DateTime startTime = proc[0].StartTime;

Console.WriteLine("Lefoglalt memória      : "
+ mem + " K");
Console.WriteLine("Összes processzoridő : "
+ procTime);
Console.WriteLine("Szálak száma      : "
+ threadCount);
Console.WriteLine("Alapprioritás     : "
+ Enum.GetName(typeof(basePriority),
procPriority));
Console.WriteLine("Indítás időpontja : "
+ startTime.ToString());
Console.WriteLine("A folyamat által
használt modulok:");
//modulok listaja
ProcessModuleCollection modules =
proc[0].Modules;
foreach (ProcessModule module in modules)
    Console.WriteLine(module.ModuleName);
}
else
    Console.WriteLine("Nincs " + processName
+ " nevű folyamat a rendszerben.");
}
```

ServiceController komponensek létrehozása

A Windows szolgáltatásokhoz és eszközkezelőkhoz ServiceController komponensek segítségével férhetünk hozzá. A komponens lehetőséget ad az adott szolgáltatás elindítására, leállítására, szüneteltetésére, a szolgáltatás által támogatott más speciális műveletek elvégzésére, és a szolgáltatás tulajdonságainak lekérdezésére is.

ServiceController komponens a System.ServiceProcess névtérben található ServiceController osztály példányosításával hozhatunk létre:

```
using System.ServiceProcess;
ServiceController serv = new
    % ServiceController("IISAdmin");
```

A komponens használatához meg kell adnunk azt a szolgáltatást, amellyel kapcsolatban fog állni. Ezt megtehetjük mindjárt a konstruktorban, mint a fenti példában látható, vagy a ServiceName tulajdonság beállításával. A konstruktor második paraméterével, illetve a MachineName tulajdonság beállításával adhatjuk meg annak a számítógépnek a nevét, amelyen a szolgáltatás fut. Az alapértelmezés természetesen itt is a helyi számítógép.

```
using System.ServiceProcess;
ServiceController serv = new ServiceController();
serv.ServiceName = "IISAdmin";
serv.MachineName = "Tavoligeip";
```

Szolgáltatások indítása és leállítása

A ServiceController komponens megfelelő metódusainak meghívásával utasíthatjuk a csatlót szolgáltatást a kívánt művelet elvégzésére. A CanStop és CanPauseAndContinue tulajdonságok lekérdezésével megállapítható (illetve beállítható), hogy lehetséges-e az adott szolgáltatás leállítása vagy szüneteltetése. A következő kódrészlet egy adott szolgáltatás indítását, szüneteltetését és leállítást mutatja be:

```
void startService(String serviceName) {
    ServiceController service = new
        ServiceController(serviceName);
    //indítás
    //csak akkor lehet elindítani, ha áll
    if (service.Status.Equals
        (ServiceControllerStatus.Stopped)) {
        service.Start();
        Console.WriteLine("Indítás...");
        //várunk, amíg el nem indul, vagy max. 10
        //másodpercig
        service.WaitForStatus(ServiceController
            Status.
            Running, new TimeSpan(0,0,10));
        Console.WriteLine("A szolgáltatás
            állapot: " + service.Status);
    }
    //szüneteltetés
    if (service.CanPauseAndContinue) {
        service.Pause();
        service.WaitForStatus(ServiceControllerStatus.
            Paused, new TimeSpan(0,0,10));
        Console.WriteLine("A szolgáltatás
            állapot: " + service.Status);
    }
}

//újraindítás
service.Continue();
service.WaitForStatus(ServiceControllerStatus.
    Running, new TimeSpan(0,0,10));
Console.WriteLine("A szolgáltatás
    állapot: " + service.Status);

//leállítás
if (service.CanStop) {
    service.Stop();
    service.WaitForStatus(ServiceControllerStatus.
        Stopped, new TimeSpan(0,0,10));
    Console.WriteLine("A szolgáltatás
        állapot: " + service.Status);
}
}
```

Szolgáltatások listázása

A szolgáltatások és eszközezők listázását a ServiceController osztály két statikus metódusának segítségével végeztethetjük el. A GetDevices() metódus az eszközezőket, a GetServices() pedig a szolgáltatásokat adja vissza ServiceController komponensekből álló tömb képében. A következő kódrészlet ki listázza a gépen található szolgáltatások és eszközezők rövid és teljes nevét:

```
void List(){
    ServiceController[] services;
    ServiceController[] devices;

    //az eszközezőket tartalmazó tömb feltöltése
    devices = ServiceController.GetDevices();
    foreach (ServiceController device in devices) {
        String ShortName = device.ServiceName;
        String LongName = device.DisplayName;
        Console.WriteLine(ShortName + "\t"+
```

```
        LongName);
    }

    //a szolgáltatásokat tartalmazó tömb feltöltése
    services = ServiceController.GetServices();
    foreach (ServiceController service in
        services){
        String ShortName = service.ServiceName;
        String LongName = service.DisplayName;
        Console.WriteLine(ShortName + "\t"+
            LongName);
    }
}
```

Szolgáltatások tulajdonságai

A ServiceController komponens segítségével az adott szolgáltatás következő tulajdonságaihoz férhetünk hozzá:

- **Status** – az adott szolgáltatás állapotát tartalmazza (Running, Stopped, Paused, stb.), a visszaadott érték a ServiceControllerStatus enum valamelyik értéke
- **ServiceType** – a szolgáltatás típusát kérdeztethetjük le, a visszaadott érték a ServiceType enum valamelyik értéke (például KernelDriver, FileSystemDriver, Adapter stb.)
- **DependentServices** – az adott szolgáltatástól függő szolgáltatásokat adja vissza, a visszaadott érték ServiceController komponensekből álló tömb
- **ServicesDependedOn** – azokat a szolgáltatásokat adja vissza, amelyekről az adott szolgáltatás függ

A következő kódrészlet ki listázza a megadott szolgáltatás vagy eszközező legfontosabb tulajdonságait:

```
void serviceProperties(String serviceName) {
    ServiceController service = new
        ServiceController(serviceName);
    //állapot
    ServiceControllerStatus stat = service.Status;
    //típus
    ServiceType type = service.ServiceType;
    //függő szolgáltatások
    ServiceController[] dependent =
        service.DependentServices;
    //ezektől függ az adott szolgáltatás
    ServiceController[] dependedOn =
        service.ServicesDependedOn;
    //listák
    Console.WriteLine("A szolgáltatás
        állapot: " + stat);
    Console.WriteLine("A szolgáltatás
        típusa : " + type);
    Console.WriteLine("A szolgáltatástól függő
        szolgáltatások: ");
    foreach (ServiceController serv in dependent)
        Console.WriteLine(serv.DisplayName);
    Console.WriteLine("A szolgáltatás ezektől a
        szolgáltatásoktól függ: ");
    foreach (ServiceController serv in dependedOn)
        Console.WriteLine(serv.DisplayName);
}
```

A cikk következő részében a .NET Framework azon komponenseivel fogunk megismerkedni, amelyek lehetővé teszik a Windows teljesítményszámlálóihoz való hozzáférést, és miniprogramot készítünk, amely megjeleníti a kapott adatokat.

SZERÉNYI LÁSZLÓ
szerenyi.l@net.hu

A cikkben szereplő URL-ek:

- [1] http://store.netacademia.net/mshu/OTHER/technet_code/procmcon.zip

Kiszolgálópark üzemeltetése 3. rész

AZ ÉLETÚT ÉS AZ ÜZEMELTETÉSI FELADATOK

Már van hol és mivel, de hogyan? Hogyan tervezzem, telepítem, üzemeltessem?

Hogyan dokumentáljam? Folytatjuk sorozatunkat a kiszolgálópark üzemeltetéséről.

Bevezetés

A kiszolgálók telepítésével, használatával, hibajavításával kapcsolatban számtalan cikket olvastam, a kiszolgálók üzemeltetésével kapcsolatos információkat ellenben igen nehéz összeszedni. Ez a cikksorozat – melynek első két része a TechNet idei 1. és 2. számában jelent meg – ennek a hiánynak a pótlására született meg. A cikksorozatban görcső alá kerültek-kerülnek az üzemeltetéssel kapcsolatos kérdések, az elhelyezéstől a napi munkákig bezárólag. Célom az egyes témakörök áttekintő bemutatása. A harmadik részben az eszközök életútját kísérjük végig, illetve szó lesz az üzemeltetési feladatok áttekintéséről, szervezéséről.

Életút

Minden rendszernek, alkalmazásnak, hardvernek van életútja. Ha szemünket rajta tartjuk ezen az úton, akkor számos buktatót elkerülhetünk. Mindez az alábbi bontásban kerül kifejtésre:

1. Tervezés
2. Implementáció
3. Megvalósult állapot / Stabilizáció
4. Dokumentálás
5. Üzemeltetés
6. Érvényben levő üzemeltetéshez köthető szerződések követése
7. Licenckezelés
8. Üzletmenet folytonosság (pl. katasztrófa...)
9. Selejtezés, üzemen kívül helyezés
10. Változáskezelés

1. Tervezés

Az eszközök, alkalmazások kiválasztása során célszerű utána nézni az adott terméknek, legyen szó akár hardver, akár szoftverelemről:

Milyen régen jelent meg a piacon?

Egy régebben a piacon levő, bevezetett termék általában biztosabban működik, mint egy frissen megjelent, a legúj-

jabb technológiai újítások, új szolgáltatások viszont általában az új termékek sajátjai.

Hányadik generációs termék

Egy első generációs, frissen piacra dobott, új konstrukció a kínált újdonságokon felül számos gyerekbetegséggel rendelkezhet. Tartalmazhat tervezési, implementációs hiányosságokat, lehetnek benne újszerű, egyedi megoldások. Ez egyfajta kísérleti kockázatot jelent a választáskor, amelyet az újszerű megközelítés, az új funkciók ellensúlyozhatnak. Egy többedik generációs termék már tartalmazza az előző termékek fejlesztési, gyártási és támogatási tapasztalatait.

Milyen gyártó / fejlesztői csapat áll mögötte?

Tudja-e a gyártó / fejlesztő a megrendelő egyéni igényeit figyelembe venni? Mennyire szabható testre a termék? A felmerült hibákat, igényeket hogyan képes kezelni?

Milyen a helyi képviselő?

Egy Ulánbátorban található regionális képvisellel lényegesen nehezebb kapcsolatot tartani, mint egy, a helyi viszonyokat ismerő csapattal.

Ügyfélszolgálat, bibelbáritás, javítócsomagok, patcbek, verziókövetés?

A termék megvásárlása után milyen támogatást tud adni a gyártó? Melyek azok, amelyekre szükség lehet?

Jótállás, garancia?

Mire terjed ki a jótállás, mire vállal a szállító garanciát? Kellemetlen meglepetés érheti a felhasználót, ha nincs tisztában az adott termék garanciális feltételeivel. pl.: egy termékre a gyártó vállalja, hogy a következő munkanapon megkezdi a hibaelhárítást. Ez legrosszabb esetben azt jelenti, hogy a hiba bejelentése utáni napon valaki foglalkozni kezd az adott hibabejelentéssel. (dolgozunk rajta.) Így akár még hetekig is elhúzódhat a hiba kijavítása, ami meg-



Most a miként
van terítéken...

lehetősen kellemetlen lehet, ha az adott rendszerelem éppen fontos üzleti folyamat része.

Milyen szabványos / egyedi megoldásokat használ?

A szabványos megoldások használata sokféle előnnyel jár. Egyedi megoldásokat csak abban az esetben válasszunk, ha valamilyen lényeges indok ezt alátámasztja. Az egyedi megoldások használata túlságosan függővé teheti a rendszert az egyes megoldásokkal, szállítókkal. A fejlesztés, bővítés, átállítás során ezek gondot okozhatnak. Problémát jelenthetnek az egyedi megoldások más rendszerekkel való integráció során is.

Milyen integrációs kapcsolódási területei vannak?

Nyitott, vagy zárt-e a rendszer? Hogyan képes együttműködni más rendszerekkel? Hogyan kapcsolható össze más rendszerekkel?

Például egy külső diszrendszer milyen eszközökre kapcsolható rá? Milyen más eszközökkel működik együtt? Egy adott alkalmazás milyen formátumban tárolja az adatait? Ehhez egy másik alkalmazás hogyan férhet hozzá?

Hogyan működik együtt más rendszerekkel?

Hogyan integrálható bele a vállalat rendszerébe? Melyek az ismert inkompatibilitási esetek?

Biztonság

Az adott rendszerelemnek van-e biztonsági minősítése? Milyen biztonsági kockázatot jelent a rajta végzett munkafolyamatok szempontjából?

Milyen biztonsági kockázatot jelent a rendszer egésze szempontjából?

Milyen biztonsági / védelmi funkciókat valósítottak meg vele?

Mentés

Milyen követelményt támaszt a lefedni kívánt üzleti folyamat?

Ennek a követelménynek képes-e megfelelni az adott rendszerelem?

Hogyan illeszthető be az adott rendszerelem a mentési rendbe?

Mit, milyen gyakorisággal, hogyan, hová mentünk? Hogyan, mit lehet visszaállítani?

Vírusvédelem

Elsősorban szoftvereknél fontos megvizsgálni, hogyan védhetőek a vírustámadásokkal szemben.

Milyen vírusvédelmi rendszerekkel működik együtt.

Egyszeri bekerülési költség

Milyen költségekkel jár az adott rendszerelem megvásárlása, üzembe állítása? Milyen licenvfeltételekkel használható?

Célszerű tüzetesen áttanulmányozni a licenvfeltételeket, mert itt az egyes gyártók néha igen bonyolult, specifikus feltételeket támasztanak, melyek jogszzerű követése, betartása néha igen-igen nehézkes, illetve költséges lehet.

Rendszeres költségek

Milyen rendszeres díjak szükségesek az üzemeltetéséhez. Ezek lehetnek időalapú licenckonstrukciók (pl.: éves eszköz-, vagy szoftver bérleti díj).

Verziókövetés: ez időszak alatt az újabb verziókat, fejlesztéseket a gyártó a rendelkezésünkre bocsátja.

Támogatás: az adott időszak alatt valamilyen formában támogatást nyújt a termékkel kapcsolatban. Az, hogy pontosan milyen támogatást nyújt, a támogatási szerződésben kerül rögzítésre. Általában a gyártók többféle támogatási csomagot nyújtanak, változó tartalommal.

Szervizdíj: a garancia lejárta után milyen díjtételei vannak az adott rendszerelem szervizelésének?

Eseti díjak: minden olyan díj, amely felmerülhet a rendszer használata során.

Kapcsolódó kiadások

Számításba kell venni minden olyan költséget, amely az adott rendszerelem üzemeltetéséhez szükséges.

A teljesség igénye nélkül néhány példa:

- Szoftverek ajánlott hardverkonfigurációja
- A Szoftverek alatt futó operációs rendszer
- Mentéshez szükséges ügynök
- Az alkalmazás külső adatbázis-kezelője
- Felügyeleti szoftver
- Kliensprogram licencre
- ...

Folyamatos üzemeltetési költség

Minden olyan költség, amely az üzemben tartáshoz szükséges.

A teljesség igénye nélkül néhány példa:

- Hardvereszközök energiaigénye
- Hardvereszközök hűtésigénye
- Az adminisztrációs személyzet költsége
- Kellékek
- ...

Referenciák, vélemények

Egy rendszer kiválasztása, tervezése során nagyon fontos a személyes tapasztalat megszerzése.

Célszerű hasonló üzleti folyamatot kiszolgáló rendszer megvizsgálása, az ottani üzemeltetéssel konzultáció.

Az így megszerzett információ tartalmaz a rendszer árnyoldalaival, buktatóival kapcsolatos ismereteket is.

Tesztelés / pilot

A tervezés során felmerült összes problémára és kérdésre nem lehet egyértelmű választ adni. Ezekre a kérdésekre a legegyszerűbben a tesztelés során kaphatunk választ.

A tesztelés nem egyszerűen az adott eszköz bekapcsolását, az adott alkalmazás felleltelepítését jelenti.

A tesztkörnyezet, illetve a tesztfolyamatokat gondosan meg kell tervezni. Az ad-hoc tesztek ritkán adnak felhasználható adatokat.

Először meg kell határozni, hogy mi az, amit tesztelni szeretnénk, milyen részeket, milyen funkciókat szeretnénk kipróbálni. (Már egy egyszerűbb rendszer esetén sem lehet mindent tesztelni.) Két nagy út áll előttünk: a gyakran használt funkciók tesztelése, illetve a nagyon extrém, nagyon ritka, esetleg hibás működés, adatbevitel tesztelése.

A tesztelési stratégiánk elkészítésekor arra is választ kell adnunk, hogy mik azok a pozitív, negatív, illetve kizáró szempontok, amelyekre fokozott figyelmet kell fordítani a tesztelés során.

A tesztelési feladatok meghatározásával egyidejűleg kell a tesztkörnyezetet is megtervezni. A tesztkörnyezetnek, legalább is a tesztelni kívánt funkciók tekintetében, minél inkább hasonlítania kell a valós környezethez. Ez igaz a hardver és szoftverelemekre. Sokszor előfordul, hogy a nem megfelelően átgondolt tesztek után váratlan nehézségekkel szembesülünk az implementáció során. Ezért fontos a kapcsolódó elemek integrációjának tesztelése is. Például a vírusvédelmi rendszer jelentősen megváltoztathatja a tesztek eredményét.

A megfelelő adattartalom, terhelés is fontos, hogy helyes eredményekhez juthassunk. Tipikus tesztelési hiba, egy több ezer felhasználóra tervezett alkalmazás tesztelése 3 gépes tesztkörnyezetben, 10 adattal feltöltött próbaadatbázissal súlyosbíva.

Az adatok kinyerhetőek az éles rendszerből, vagy generálhatóak hasonló mennyiségű és jellegű tesztadatok is. A terhelésre általában különféle jelenlét és forgalom szimulációs alkalmazások használhatók, a nagyobb rendszerek általában rendelkeznek ilyen segédprogramokkal.

A tesztek összeállítása során gondolat kísérleteket is kell végeznünk, meg kell adnunk, hogy az egyes tesztekre milyen eredményeket várunk, mi számít optimális, megfelelő eredménynek.

Ezek után nincs más hátra, mit a tesztkörnyezet implementációja, majd a tesztek végrehajtása, a tesztelési napló kitöltése.

A tesztek összeállítása és a tesztelés során sokat segíthet, ha segítségül hívjuk a leendő felhasználókat, akik a technikai részeken túl a konkrét üzleti folyamatok tesztelését véggezhetik.

A legfontosabb része a tesztelésnek a kiértékelés. Ekkor megvizsgáljuk a kapott eredményeket, összehasonlítjuk a várt eredményekkel.

Az eredményeket figyelembe véve véglegesíthetjük a terveket, illetve újabb tesztek tervezhetünk, amíg marad nyitott kérdés.

Az igazán alapos tesztelés idő- és erőforrás-igényes feladat.

2. Implementáció

A megfelelően megtervezett rendszer megvalósítása az implementációs terv alapján már viszonylag egyszerűen történik. Tapasztalatom szerint szinte minden extra nehézség, probléma, implementációs hiba tervezési, illetve tesztelési hibára vezethető vissza.

Éppen ezért nem győzőm hangsúlyozni az implementációs terv készítésének fontosságát. Egy éles adatokat tartalmazó valós környezetben a kísérletezés, az ad-hoc megoldások alkalmazása, a váratlanul felbukkanó, nem várt ablakok lezárása, vagy véletlenül kitöltése életveszélyes buherálásához vezet, számos hiba, rejtélyes, nehezen felderíthető hiba melegegyává válhat.

Visszaállítás (Rollback)

Az implementációs terv fontos része a visszaállási terv. Azokat a lépéseket tartalmazza, amelyek révén a megadott feltételek teljesülése, vagy nem teljesülése esetén a rend-

szert visszaállítható az implementáció megkezdése előtti állapotba.

Nagyon fontos, hogy az implementáció folyamán folyamatosan vezessük az implementációs naplót, melyből a megvalósulási dokumentáció is készül. Később, esetleg napok múlva szinte lehetetlen rekonstruálni, hogy pontosan milyen lépések, milyen sorrendben történtek.

3. Megvalósult állapot / stabilizáció

A sikeres implementáció után a rendszer új állapotba. Megadott sikerkritériumok teljesítése esetén tekinthetjük a rendszer implementációt sikeresnek. Ezt nevezzük megvalósult állapotnak.

Azonban egy-egy új rendszer(elem) beállítása után rendszeren számítathatunk különféle transziens jelenségekre. A rendszer éles környezetben történő használatára során előkerülnek olyan problémák is, amelyekre a tesztelés során nem derült fény. A stabilizációs fázis során ezeknek a problémáknak a feltérképezése, lehetőség szerint elhárítása a fő feladat. (A „szállítók” egyik kedvenc mondanca: „Mi ezt teszteltük, az otthon három gépes rendszerünkön, ahol mindenki rendszergazda, teljesen jól ment, nem értem, hogy itt mi lehet a baj”)

4. Dokumentálás

A jól dokumentált rendszert lehet igazán jól kezelni, üzemeltetni. A dokumentációval a legfontosabb tennivaló a naprakészen tartás. Egy elavult, csak részben helyes információkat tartalmazó dokumentáció nagyon megnehezítheti a munkát. Aztán lassan leszoknak a használatáról, frissítéséről, egyre inkább nyuggék, mellőzötté válik, az benne levő információ elértéktelenedik.

A dokumentumoknak célszerű egységes megjelenést biztosítani. Így könnyebben rendszerezhető, áttekinthetőbb, rendezettebb dokumentációt készíthetünk.

Kisebb rendszerek dokumentációja általában csak néhány dokumentumból áll, nagyobb rendszerek dokumentációja során célszerű a dokumentációt strukturáltan, áttekinthető állományokban tárolni.

A különféle rajzok, ábrák, egyéb dokumentumok mellett célszerű feltüntetni, hogy milyen programok, alkalmazások szükségesek az adott dokumentum megnyitásához, szerkesztéséhez.

Miért van szükség dokumentálásra?

Elsősorban kisebb rendszeremmel töltöthet fel magukban ezt a kérdést.

„Az én rendszerem nagyon egyszerű felépítésű, fejben tartom a dokumentációt...”

- Ha beüt a krach, akkor nem mindig elérhető az a bizonyos fejt. (Hétvégén, éjjel, éppen karbantartás közben kellenének az internet szolgáltató beállításai, adatai, amelyek éppen a szférben vannak, a széf kulcsát csak a titkárnő éri el, aki éppen ki van kapcsolva...)
- Ha hirtelen, gyorsan, stresszhelyzetben van szükség az adatokra, nem biztos, hogy a pontos adatok kerülnek elő.
- Az adatokat pontosan kell tudni. (Egy elgépelt név, rosszul megjegyzett szám sok galibát okozhat.)

- Harmadik személynek kell betekintést nyernie a rendszerbe valamilyen okból. Ilyenkor sokat segíthet az átlátható dokumentáció. (Az időt rabló vége-hossza nincs beszélgetés a rendszerrel! félreértésekre ad lehetőséget, hibás munkavégzéshez vezethet.) A betekintésre számos okot lehet találni:
- fejlesztés
- szervizelés, hibaelhárítás
- audit
- ellenőrzés
- stb...

Nyilván nem életszerű egy három géppel rendelkező kis cégnek több száz oldalas rendszer-dokumentációval rendelkeznie. Meg kell találni az arany középút.

Az igazán használható dokumentáció olyan részletességgel, felbontással, és éppen azokat az információkat tartalmazza, amelyekre szükség lehet.

Egy-egy rajz, áttekintő ábra (akár kézzel rajzolt is) a megfelelő adatokkal sok oldalas dokumentációnál hasznosabb lehet, sőt a naprakészen tartása is jóval egyszerűbb feladat.

Mit tartalmazzon egy dokumentum?

Az alábbi szempontok szerint megszerkesztett dokumentumokkal igen könnyen lehet dolgozni. Természetesen egyéni igényeknek megfelelően bővíthető (pl.: lehet vállalati fejlesztés elektronikus sablont használni), illetve zúdítható a szempontrendszer:

- elnevezés: a dokumentum neve, amely utal a tartalmára
- tartalomjegyzék: segít tájékozódni a dokumentumon belül
- állomány név, elérési út: hol található meg a dokumentum elektronikus formában?
- készítő: ki készítette a dokumentumot?
- verzió: hol tart most a dokumentum?
- módosítások / utolsó módosítás: mikor és ki módosította a dokumentumot
- utolsó módosítás dátuma
- összefoglaló: rövid általános, lehetőleg nem technikai nyelven megfogalmazott szöveg, melyben a dokumentum elkészítésének célja, irányelvei, tartalma, használatának módja megtalálható
- Leíró rész: a tulajdonképpeni dokumentáció. Legyünk olyan pontosak, amennyire csak lehetséges. Használjunk, a magyarázatok mellett ha szükséges képernyőképeket, ábrákat, táblázatokat.
- A dokumentum változásait követésére a Word változások követése funkciója is használható.

Hogyan szervezzük a dokumentációnkat?

- Rendszer-dokumentáció
- Mentési dokumentáció
- Üzemeltetői dokumentáció

Rendszer-dokumentáció

A rendszer és elemeinek leírása.

Készülhet többféle megközelítés szerint:

- hardverelemek: az egyes hardverelemek és a hozzájuk tartozó, rajtuk futó szoftverelemek.
- szoftverelemek: az egyes szoftverek, illetve az alattuk található infrastruktúra
- szolgáltatók: az egyes szolgáltatások és ezeknek a biztosításához szükséges hardver- és szoftverelemek
- infrastruktúra: az egyes infrastruktúrák elemei, illetve a kapcsolódó hardver és szoftverkomponensek
- áttekintő, topológiai, funkcionális, strukturális ábrák, rendszerrajzok.

Általában nagyságrendekkel nehezebb egy már meglévő rendszert felmérni és utólag dokumentálni, mint már a tervezés során elkezdeni a rendszerelemek dokumentálását.

A sok helyen meglévő hiányosságok fokozatos pótlása is a gördülékenyebb üzletmenetet segíti elő.

A dokumentáció alapja a rendszer későbbi fejlesztésének, üzemeltetésének, auditjának.

Mentési dokumentáció

Fontossága miatt is kritikusi része a teljes dokumentációnak.

A mentési stratégia alapján dől el, hogy melyik rendszerelemet, hogyan, milyen feltételekkel kell menteni.

A mentési stratégia ismeretében dolgozható ki az egyes rendszerelemek mentési terve. A mentési terv alapján kell a mentési munkafolyamatokat létrehozni, beállítani. Ezek alapján készül el a mentési ütemterv, amely a konkrét mentési munkák időbeosztását tartalmazza. A mentések lefutásának menetét a mentési napló tartalmazza.

Természetesen a mentési dokumentáció része a visszaállítási dokumentáció is. Csak annak a mentésnek van gyakorlati értéke, amelyből ténylegesen vissza lehet állítani részben, vagy teljes egészében az adott rendszer elemeket.

Üzemeltetői dokumentáció

Az üzleti folyamatok támogatására nyújtott szolgáltatások szerződésben rögzített teljesítményének, minőségének, rendelkezésre állásának biztosítására készített eljárások és tevékenységek leírása.

MEGYESI BARNABÁS
megyesi.barnabas@meb.hu
üzemeltetés vezető

MCSE, HP ASE, dromedárszakértő

Források:

1. Microsoft Windows2000 kiszolgáló üzemeltetési kézikönyvek
2. Microsoft Windows2003 kiszolgáló üzemeltetési kézikönyvek
3. Microsoft Operations Framework

A cikkben szereplő URL-ek:

- [1] <http://www.microsoft.com/technet/prodtechnol/windows2000serv/maintain/opsguide/default.mspx>
- [2] <http://www.microsoft.com/technet/prodtechnol/windowsserver2003/operations/default.mspx>
- [3] <http://www.microsoft.com/mof>

Új törvény az elektronikus aláírásról

„RÁNCFELVARRÁS” 3 ÉV UTÁN

A 2001. évi elektronikus aláírásról szóló törvény megszületése óta nem változott sok minden az elektronikus aláírás elterjedését illetően, az áttörés, a tömeges alkalmazás továbbra is várat magára. A jogszabályi háttér megteremtéséért felelős kormányzat ezzel összefüggésben is az alig három éves jogszabály átdolgozásáról döntött.

Az elektronikus aláírás jogszabályi háttere

Az információs társadalom megteremtéséhez szükséges jogszabályi háttér egyik első és kiemelkedő fontosságú eleme az elektronikus aláírásról szóló 2001. évi XXXV. törvény (Eat.) volt. Hatályba lépése óta számos további informatikai tárgyú jogszabály született, melyek közül a 2001. évi CVIII. törvényt (az elektronikus kereskedelmi szolgáltatások, valamint az információs társadalommal összefüggő szolgáltatások egyes kérdéseiről), a 2003. évi LXXXI. törvényt (az elektronikus cégeljárásról és a cégiratok elektronikus úton történő megismeréséről), a 2004. évi XXXV. törvényt (az elektronikus pénzt kibocsátó szakosított hitelintézetéről) és a 20/2004. (IV. 21.) PM rendeletet (az elektronikus számláról) érdemes emlékeztetőül kiemelni. Az elektronikus aláírás felhasználásának lehetősége ezen kívül számos további jogszabály kiegészítésével vált lehetségessé a köznapi és hivatalos élet megannyi területén. Példa erre a 2003. évi XCII. törvény az adózás rendjéről, a 1991. évi XLI. törvény a közjegyzőkről, a 2000. évi C. törvény a számvitelről és a 1992. évi LXXIV. törvény az általános forgalmi adóról. E korántsem teljes listából látszódik, hogy az elmúlt években sok minden történt az információs társadalom szabályozása területén. Nem változott viszont sok az elektronikus aláírás elterjedését illetően, ahol az áttörés, a tömeges elterjedés továbbra is várat magára. Az okok mélyebb elemzése egy másik cikk tárgyának kívánkozik, így most csak annyit említenék meg, hogy minden szereplő saját háza táján is kereshetné ezeket. Így tett a jogszabályi háttér megteremtéséért felelős kormányzat is, amikor az alig három éves jogszabály átdolgozásáról döntött. A kezdeti elképzelésekben egy nagygenerál szerepelt, ami azonban politikai okok miatt egy rendes felújítással szelődött, s a 2004. évi LV. törvény „az elektronikus aláírásról szóló 2001. évi XXXV. törvény módosításáról” címet kapta. Így is van azonban változtatás elég, melyek megérdemlik, hogy áttekintsük őket.

A törvényt módosítás indokai

A törvényt módosítás hivatalos indoklása szerint a változtatás egyik oka az elektronikus aláírás használatának hazai elterjedésének elősegítési szándéka, melyet a szabályozási környezet egyszerűsítésével kíván elérni. Ezen egyszerűsítés túl messzire végül is nem ment, legfeljebb egy-két fogalom összehasonlásig jutott. A változtatás másik – valószínűleg erősebb – indoka az volt, hogy még a 2003 során EU szakértői jelentés készült, amely a magyar szabályozás intézményrendszerének hibáira hívta fel a figyelmet (a törvény az Európai Parlament és Tanács az elektronikus aláírások közösségi programjáról szóló 1999/93/EK irányelve alapján készült). A módosítás figyelembe vette azokat a folyamatokat is, „amelyek a Magyar Információs Társadalom Stratégiában kitűzött célokkal összhangban 2003. év végén kezdetüket vették, s melyek – úgy az elektronikus kormányzati, mint a gazdasági kapcsolatok viszonyrendszerében – az elektronikus aláírási szolgáltatások hazai helyzetében, az elektronikus aláírás társadalmi szintű elterjesztésében döntő fordulatot jelentenek”. Az indoklás eme részlete csak elrettentő célzatú dekoráció, mivel értelmének megfejtesében nem tudom támogatni az olvasókat.

Főbb változtatások

A bővebb magyarázatra szoruló idézet megfejtesét segítheti, hogy az említett folyamatokat egy új szolgáltatási forma, az elektronikus archiválás bevezetésével és a kapcsolódó jogintézményi rendszer megteremtésével kívánja támogatni az új törvény. A módosítások az EU Irányelv által használt terminológiának megfelelően pontosítják a törvény fogalomhasználatát. Az egyéb változtatások közül kiemelendő a szolgáltató adatszolgáltatási kötelezettségének előírása (6. §), az önkéntes akkreditáció lehetővé tétele (8. §), az időbélyegzés-szolgáltatás nyújtásával kapcsolatos szabályok törvényi szinten való megállapítása (9. §), az elektronikus archiválási szolgáltatásra, az archiválási szol-

gáltató adatkezelésére, az archiválási szolgáltatást igénybe vevő kötelezettségeire, az archiválási szolgáltató kártérítési felelősségére és az archiválási szolgáltatás nyújtásának befejezésére vonatkozó előírások (10. §), valamint a technológiakövetés követelményével összefüggésben a hatóság speciális feladatainak megállapítása (12. §).

Fogalmi változtatások

Az Eat. 2. §-a nemcsak az e törvényben használt egyes fogalmakat határozza meg, hanem értelmezésük minden olyan jogszabályra nézve kötelezően irányadó, melyek azokat említik. Az eredetileg 21 pontból álló fogalomkészlet egyes elemeit a módosítás – az egyéb változásokkal összhangban – hatályon kívül helyezi, másokat részben vagy egészen új fogalom-meghatározásokkal vált fel, valamint új fogalmakat is bevezet. Rövidebb a nem változó fogalmak listája, melyek a következők:

- aláírás-létrehozó adat,
- aláírás-ellenőrző adat,
- aláírás-létrehozó eszköz,
- biztonságos aláírás-létrehozó eszköz,
- elektronikus aláírás,
- elektronikus aláírás ellenőrzése,
- elektronikus aláírás felhasználása,
- elektronikus aláírás-hitelesítési szolgáltató,
- elektronikus aláírás-történet.

Olyan alapvető fogalmak változnak viszont, mint az aláíró, mely az eredeti fogalom-használat szerint az a természetes személy, akihez az elektronikus aláírás hitelesítés-szolgáltató által közölt aláírás-ellenőrző adatok jegyzéke szerint az aláírás-ellenőrző adat kapcsolódik. A módosítás e fogalmat jelentősen egyszerűsíti, nem az aláírásra való jog keletkezéséhez, hanem – vélelmezve azok jogszerű meglétét – az aláíró megillető alapvető jogosítványokhoz (birtoklás és használat) köti. Így az aláíró az a természetes személy lett, aki az aláírás-létrehozó eszközt birtokolja és a saját vagy más személy nevében aláírásra jogosult.

Az elektronikus aláírási termék eredeti meghatározása szerint olyan szoftver vagy hardver, amely elektronikus aláírással kapcsolatos szolgáltatások nyújtásához, így különösen elektronikus aláírások, illetőleg időbélyegző készítéséhez vagy ellenőrzéséhez használható. Az új fogalom-meghatározás egyrészt utal arra, hogy a szolgáltatások nyújtásához a hardver és szoftver mellett egyéb összetevők is igénybe vehetők, valamint az elektronikus aláírással kapcsolatos szolgáltatásokon túl a termék felhasználási lehetőségét kiterjeszti az elektronikus aláírások, illetőleg időbélyegző készítéséhez vagy ellenőrzéséhez történő felhasználhatósággal is. Vagyis azok a szoftverek és hardverek (pl. chipkártyák), melyeket magunk egyszerű emberként használunk, ezen túl szintén elektronikus aláírási terméknek számítanak.

Az Eat. eredeti 2. §-ának 12. pontja határozza meg az elektronikus dokumentum, 13. pontja az elektronikus írat, 14. pontja az elektronikus okirat fogalmát. E megkülönböztetés helyett a módosítás az elektronikus írat és okirat fogalmi megszüntetése mellett új, egységes elektronikus dokumentum-definiációt vezet be, mely szerint az „elektronikus eszköz útján értelmezhető adategyüttes”.

Az „elektronikus aláírás érvényesítése” és „érvényességi lánc” új fogalomként jelennek meg a jogszabályban. Az előbbi annak tanúsítását jelenti minősített elektronikus aláírás vagy e szolgáltatás tekintetében minősített szolgáltató által kibocsátott időbélyegző elhelyezésével, hogy az elektronikus dokumentumon elhelyezett elektronikus aláírás vagy időbélyegző illetve az azokhoz kapcsolódó tanúsítvány az időbélyegző elhelyezésének időpontjában érvényes volt. Bár a fogalom meghatározásán láthatóan sokat dolgoztak, azt a törvényben a későbbiekben nem használják. Talán majd később, egy alacsonyabb rendű jogszabályban jól fog jönni. Mint bizonyultságából sejthető, a kifejezés nem a hétköznapi aláírás-ellenőrzést takarja (az, mint fogalom továbbra is fennmarad), hanem azt a tevékenységet, amikor egy szolgáltató (pl. archiválási vagy érvényességi szolgáltató) egy aláírás ellenőrzése után annak helyes voltát egy másik aláírás-szal igazolja. Első hallásra ez elég furcsán hangozhat, s a végtelen rekurzió gondolatát is felvetheti, mégis van értelme. Ilyen eset lehet például az elavult határához érkezett aláírások és időbélyegzők felülbélyegzése. Az érvényességi lánc fogalmát a szakemberek korábban is használták, most az elektronikus archiválási szolgáltatás kapcsán a törvényhozók is kulcsfontosságúnak tartották, s bevezették. Az általa meghatározott jogszerűségi, hitellességi követelmények azonban nem lettek kiterjesztve egyéb esetekre.

Az új Eat.-ban megjelenik az igénybe vevő fogalma. Erre – definíció nélkül – az eredeti törvénybe is esett utalás, mostantól azonban egységes és kiterjesztett értelemben történik a használata. Az igénybe vevő nem csak az aláíró lehet, de az aláírás ellenőrző is, illetve egyéb szolgáltatások esetében a szolgáltatást igénybe vevő természetes személy, jogi személy vagy jogi személyiség nélküli szervezet. Ezen fogalom bevezetése megszünteti az Eat. eredeti szabályozási rendszerének aszimmetrikus, elsődlegesen a szolgáltatói kötelezettségekre, jogosítványokra összpontosító voltát, s a jogbiztonság magasabb fokán teszi lehetővé az ügyféljogok meghatározását és érvényesíthetőségét. Hasonló célból valamint a szolgáltatások bővülő köre miatt definiálja az új törvény önálló fogalomként a minősített szolgáltatót és a szolgáltatót is: a specifikus meghatározások az általános szolgáltató-fogalomhoz képest a törvény más rendelkezéseivel össz-

Az elmúlt években sok minden történt az információs társadalom szabályozásával kapcsolatban. Az elektronikus aláírás egyre több területen használható jogi bizonyító erővel, a hagyományos aláírás alternatívájaként. Az új törvény a felhasználási lehetőségeket sorát egy igen fontos területtel bővíti.

hangban határoznak meg többletkövetelményeket. Az archiválási szolgáltatás összefüggésében a törvény két további fogalommal egészíti ki az Eat-ot: az archiválás szolgáltatása valamint – az elvárható biztonságossági szintre is tekintettel lévő technológiai követelményrendszer meghatározásával – a lenyomat definíciójával. A fokozott biztonságú elektronikus aláírás, az időbélyegző, a minősített elektronikus aláírás, a minősített hitelesítés-szolgáltatás és a tanúsítvány fogalom-meghatározásának módosítására az új fogalmakkal való összhang megteremtése érdekében került sor, tartalmilag ezek esetében nem történt jelentős változás.

Az Eat. 2. §-ának 20. pontja változatlan formában határozza meg az elektronikus aláírással kapcsolatos szolgáltatásokat nyújtó szolgáltatók tevékenységével kapcsolatos részletes eljárási és egyéb működési szabályokat tartalmazó szolgáltatási szabályzat definícióját. Új fogalomként jelenik meg ugyanakkor az ún. hitelesítési rend, valamint az időbélyegzési rend, mely az alsóbb szintű jogszabályokban korábban is szerepelt s számos félreértelmű okot adó tanúsítványtípus fogalmat váltja fel. E rendek alkalmazását elő is írja a jogszabály a szolgáltatók részére (akik jellemzően már korábban is kibocsátottak ilyet).

Az új Eat. 2. § 1. bekezdése az aláírás-létrehozó adat az aláírás-létrehozó eszközön való elhelyezése, az időbélyegzés elhelyezése és az elektronikus dokumentum archiválás szolgáltatások minősített szintjére vonatkozóan állapít meg vélelmeket, melyekhez hasonló az elektronikus aláírás hitelesítés-szolgáltatásra már korábban is létezett. Ezek szerint a szolgáltatónak mindaddig igazna van, amíg valaki azt meg nem cáfolja. Vagyis a bizonyítási kényszer a kétségekkel rendelkező félé.

Elektronikus archiválás szolgáltatás

Az archiválási szolgáltatás, mint új elektronikus aláírással kapcsolatos szolgáltatás megjelenése átszövi az új Eat.-ot. A fogalmi meghatározások után az új törvény az Eat. 6. §-ának szolgáltatástípusok-felsorolását egészíti ki az elektronikus archiválás szolgáltatással. A felsoroltakon kívül más elektronikus aláírással kapcsolatos szolgáltatás is nyújtható, de ezeket a jogszabály nem nevešti, s a velük kapcsolatos különös követelményeket sem tárgyalja. Ezek így vagy nem lesznek szabályozva, vagy más törvények, alsóbb szintű jogszabály tárgyalhatják őket. A 6. § (4) bekezdésében kerül összefoglaló meghatározásra az elektronikus archiválási szolgáltatás keretében nyújtott szolgáltatások tartalma. E szerint az archiválás szolgáltatónak a részére átadott dokumentumok letagadhatatlanságát, hiteles megőrzését, sérthetetlenségét, s az aláírások hosszú távú ellenőrizhetőségét kell biztosítania, s kérelem esetén igazolást kell kibocsátania a dokumentummal kapcsolatban, illetve áll kell adnia az igénybe vevő részére. Az értelemzhetőség/olvashatóság fenntartására a szolgáltató csak az általa vállalt dokumentum-típusok esetében köteles, hiszen az ehhez szükségeses hardver- és szoftverkönyezet fenntartása igen költséges lehet.

Erősen megkérdőjelezhető azonban, hogy mi keresni valójában az archiválási szolgáltatásnak az elektronikus aláírási szolgáltatások között, és így az új elektronikus aláírási törvényben. Noha a törvény szövege elektronikus dokumentumok archiválásáról szól, amely mint már tudjuk „elektronikus eszköz útján értelmezhető adategység”, a szövegből kitűnik, hogy a jogalkotók végig elektronikus aláírt dokumen-

tumok vagy elektronikus aláírások archiválásában gondolkodtak (melyek önálló megőrzésének végképp semmi értelme). Ennek okát a törvény indoklása homályban hagyja, s bármennyire erőlködöm, nem is tudom kitalálni. Ez az indoklatlan szűkítés még sok zavart okozhat a jogszabály alkalmazásában.

Az ennek kapcsán megjelenő hosszú távú elektronikus aláírás érvényesítés némi bővebb magyarázatot kíván. Hosszú távon azt az időtávot érti a szakma, amely alatt az irat hitelesítésekor használt kriptográfiai algoritmus elavul, vagy a hordozó média ill. a megjelenítő szoftver kereskedelmi forgalomban már nem kapható. Az European Electronic Signature Standardization Initiative (Európai Elektronikus Aláírások Szabványosítási Kezdeményezése) a digitális aláírások érvényesítésének három esetét: a rövid-, a közép- és a hosszú távú érvényesítést különbözteti meg, s a három esetben fellelhető archiválási feladatokat az ETSI vonatkozó szabványban meghatározott aláírás-formátumok alkalmazásához kapcsolva írja le (lásd „Biztonságos aláírás-kezelő alkalmazások” cikksorozat TechNet 2003. június-október). Meghatározása szerint középtávnak számít az az időtáv, amelyben az aláírás elhelyezésekor rendelkezésre álló információk (pl. visszavonási listák (CRL-ek) már nem elérhetőek). A hosszú távú archiválás során a szolgáltatónak az algoritmus elavulással feldakod feladatok ellátása mellett – mivel ebben az időtávban az elektronikus dokumentum keletkezésekor használt hardver- és szoftvereszközök is nehezen hozzáférhetővé válhatnak – az általa meghatározott formátumok tekintetében folyamatosan biztosítania kell az adathordozó olvashatóságát, illetve az elektronikus dokumentum megjeleníthetőségét is. A közép- és hosszú távú megőrzéssel kapcsolatos követelmények teljesítése speciális szakértelmet és nagy anyagi erőforrásokat igényel (technológiakövetés, olvashatóság és megjeleníthetőség biztosítása, stb.). E szolgáltatások során speciális felelősségi kérdések merülnek fel, illetve a szolgáltatás megbízhatatlansága esetén helyrehozhatatlan károk keletkezhetnek, ezért a hosszú távú hiteles (minősített) elektronikus archiválás megbízható harmadik személy által nyújtott szolgáltatásként került szabályozásra. Az Eat. megteremtíti a hosszú távú archiválás, mint szolgáltatás nyújtásának lehetőségét, a szolgáltatás részletes szabályai azonban külön IHM-rendeletben kerülnek meghatározásra, amire még szintén várni kell. Meg kell említeni, hogy a törvény csak megteremtíti az archiválási szolgáltatás nyújtásának feltételeit, de nem ír elő kötelezettséget arra, hogy valamilyen szervezet vagy ágazat a szolgáltatásra támaszkodva végezze el a megőrzést. Ezek a kötelezettségek külön jogszabályokban jelenhetnek meg a jövőben.

Az archiválási szolgáltató adatvédelmi jogi helyzete oly módon kerül szabályozásra, hogy a szolgáltatási szerződésnek, illetve általában az igénybe vevő és a szolgáltató közötti viszonyra meg kell felelnie az adatkezelő-adatfeldolgozó viszonyra az adatvédelmi törvény által előírt követelményeknek. Ezzel a megoldással a viszonyban is érvényesülnének egyes, a magyar adatvédelmi jog által meghatározott olyan követelmények, mint pl. az adatfeldolgozó adatfeldolgozó általi igénybevételek tilalma (vagyis a szolgáltató nem helyezheti ki a szolgáltatást egy másik szolgáltató részére).

Hatalmas hiányossága a szabályozásnak, hogy csak az elektronikus dokumentumok archiválását taglalja, a papír alapú dokumentumokét nem. A hiányosság egy további külön

jogszabályban kerül majd rendezésre, melyre a törvény is utal: „Ha a papíralapú dokumentumról külön jogszabályban meghatározott módon elektronikus másolatot készítenek, a másolatot tartalmazó elektronikus dokumentumhoz is fűződnék a papíralapú dokumentumhoz jogszabály által fűzött joghatások.” Sajnos azonban a hiteles másolatkészítésről szóló jogszabály több hónapos késésben van, így arra továbbra is várniuk kell a már fellelkesült érdekelteknek.

A szolgáltatások nyújtásának feltételei

Az új törvény az Európai Bizottsági szakértői jelentés alapján módosítja a szolgáltatások nyújtásának feltételrendszerét. Az új szabályok szerint nem változik a szolgáltatások nyújtására jogosultak köre, valamint a bejelentésére szolgáló határidő. Az irányelvet követve azonban szemléletbeli megújulás történt, melyet a paragrafusok címének változása is jelez: az eredeti „A szolgáltatások nyújtásának feltételei” cím „Bejelentés”-re módosult. A bejelentés időpontjában azonban minden, a nyilvántartásban majdan közzétételre kerülő, a szolgáltatóra és az általa nyújtani kívánt szolgáltatásra vonatkozó információknak rendelkezésére kell állni, ami miatt bővül a bejelentéshez kötelezően csatolandó mellékletek köre, s a szolgáltatónak nyilatkoznia kell arról is, hogy tevékenységét minősített szolgáltatóként kívánja-e folytatni. Az új szabályozás szerint a „minősített” jelzőt elektronikus aláírással kapcsolatos szolgáltatáshoz kapcsolódóan hirdetésében, hivatalos levelezése során, az általa kibocsátott tanúsítványban vagy bármely egyéb módon csak az e törvény szabályai szerint nyilvántartásba vett minősített szolgáltató használhatja.

Még értekelhetőbb a jogszabály filozófiájának változása ezen, minősített szolgáltatást nyújtani kívánó szolgáltatók tekintetében. Az eredeti Eat. előzetes hatósági minősítés kötelezettségét írja elő. Az új szabályozás szerint ez megszűnik, s a törvény az Irányelv szellemének megfelelően mindössze azt írja elő, hogy a bejelentés alapján minősített szolgáltatási tevékenységet folytatni kívánó szolgáltatónak működése során milyen feltételeknek kell megfelelnie. A minősítési eljárás megszűnésével párhuzamosan a minősített szolgáltatók ellenőrzését erősíti az új szabályozás. A garanciák erősítését indokolja az elektronikus aláírás elterjedéséhez szükséges fogyasztói bizalom megerősítése, valamint, hogy az Eat. az említett vélelmeket jövő ezen szolgáltatásokhoz. Ennek érdekében a szolgáltatás nyújtásának megkezdésétől számított 30 napon belül a Hatóságnak el kell végeznie a minősített szolgáltató ellenőrzését, valamint évente egyszer átfogó helyszíni ellenőrzést is tartania kell. Nem minősített szolgáltatók esetében a Hatóság jogosult, de nem köteles a működés ellenőrzésére.

A törvény szankciórendszere a minősítési eljárás törlesztése és a minősített szolgáltatás új értelmezésére tekintettel kiegészül azzal, hogy a Hatóság törölheti a nyilvántartásból azt a tényt, hogy a szolgáltató szolgáltatását minősítettként végzi, és megtilthatja az ezzel kapcsolatos jelző használatát, amennyiben a törvény, egyéb jogszabályok, a szolgáltatási szabályzat, illetve a Felügyelet határozatainak megszegését tapasztalja.

A szolgáltatók a minősítésen túl önkéntes akkreditációs rendszer keretében tanúsíthatják szervezetük, rendszerük, valamint a szolgáltatás során alkalmazott termékeiknek és hálózatuknak külön jogszabály szerinti informatikai biz-

tonsági követelményeknek, vagy más önként vállalt követelményeknek való megfelelést is. Ez olyan, kifejezetten a hitelesítés-szolgáltatásokra vonatkozó jogokat és kötelezettségeket meghatározó engedélyt jelent, amelyet az érintett hitelesítés-szolgáltató kérelmére az az állami vagy magánszervezet ítélt oda, amely az ilyen jogok és kötelezettségek megállapítását, felügyeletét végzi. Erősen kérdéses, hogy ezzel a lehetőséggel melyik hazai szolgáltató fog élni, illetve ilyen hazai akkreditációs rendszer ki fog e alakulni, hiszen ilyen rendszerek kialakítása, fenntartása, fejlesztése, a megfelelőségi vizsgálatokra való felkészülés hatalmas összegeket emészt fel, melyek messze nincsenek összhangban a piac jelenlegi jövedelmezőségével.

Aláírás-létrehozó adat elhelyezése és időbélyegző szolgáltatás

Az eredeti Eat. csak a hitelesítés-szolgáltatás elemeit és feltételeit határozta meg egyéb szolgáltatásokat csak megemlítt. Az új Eat. egyensúlyba hozza az egyes szolgáltatásokat azáltal, hogy az aláírás-létrehozó eszközön az aláírás létrehozó adat elhelyezése és az időbélyegző szolgáltatásokra vonatkozóan is megállapítja a szolgáltató tevékenységének főbb összetevőit. Az előbbi esetében a szolgáltató a szolgáltatás keretében vállalja, hogy az igénybe vevő aláírás-létrehozó adatát az aláírás-létrehozó eszközön elhelyezi, illetve elvégzi az aláírás-létrehozó eszköz megszemélyesítését. A szolgáltatónak gondoskodnia kell az aláírás-létrehozó és az aláírás-ellenőrző adatok biztonságos létrehozásáról, s köteles biztosítani az igénybe vevő aláírás-létrehozó adatának titkosságát, valamint az aláírás-ellenőrző adat sértetlenségét. Az utóbbi esetében szolgáltatók oly módon kell az időbélyegző létrehozni, hogy az összekösse az aktuális időt (az időadatot), egyedi sorszámot és az időbélyegzővel ellátni kívánt elektronikus dokumentumot vagy annak lenyomatát. A szolgáltató az időbélyegző elhelyezése során az igénybe vevő által számára átadott elektronikus dokumentum vagy lenyomat tartalmát csak az időbélyegző elhelyezéséhez szükséges mértékben módosíthatja.

További törvények és rendeletek

A törvény módosítja a cégnyilvántartásról, a cégnyilvánosságról és a bírósági cégeljárásról szóló 1997. évi CXLV. törvényt, valamint a közjegyzőkről szóló 1991. évi XLI. törvényt. Az előbbi az elektronikus cégeljárásról és a cégratok elektronikus úton történő megismeréséről szóló 2003. évi LXXXI. törvény rendelkezéseire tekintettel kiegészíti a cégratok elektronikus archiválásának és az elektronikus aláírás közjegyzők általi felhasználásának szabályait. A törvény 23. §-a az elektronikus közbeszerzéssel összefüggésben módosítja a közbeszerzésekről szóló 2003. évi CXXIX. törvényt is.

Az Eat. módosítását követően megtörtént a hozzá kapcsolódó kiegészítő rendeletek módosítása is: a Nemzeti Kereskedési Hatóságnak az elektronikus aláírással kapcsolatos feladat- és hatásköréről, valamint eljárásának részletes szabályairól szóló kormányrendelet; a Nemzeti Kereskedési Hatóságnak az elektronikus aláírással összefüggő nyilvántartással kapcsolatos tevékenységéről fizetendő díjakról szóló IHM rendelet; az elektronikus aláírás hitelesítés-szolgáltatásra, időbélyegző-szolgáltatásra és aláírás-létrehozó adaton aláírás elhelyezés szolgáltatásra és ezek szolgáltatóira vonatkozó részletes követelményekről szóló IHM rendelet és az elektronikus

alírási termékek tanúsítását végző szervezetekről, illetve a kijelölésükre vonatkozó szabályokról szóló IHM rendelet.

Eredeti jogszabályok:

151/2001. (IX. 1.) Korm. rendelet
15/2001. (VIII. 27.) MeHVM rendelet
16/2001. (IX. 1.) MeHVM rendelet
20/2001. (XI. 15.) MeHVM rendelet

Az ebbe a jogszabálysáládba tartozó 2/2002. MeHVM irányelv a minősített elektronikus aláírással kapcsolatos szolgáltatásokra és ezek szolgáltatóira vonatkozó biztonsági követelményekről; valamint a 7/2002. MeHVM rendelet az elektronikus aláírással kapcsolatos szolgáltatási szakértő nyilvántartásba vételéről, esetleges változtatásáról nincs tudomásom.

Az előkészületben lévő kapcsolódó jogszabályok közül érdemes még megemlíteni a „központi közigazgatási szervek és a helyi önkormányzati közigazgatási szervek által készített elektronikus dokumentumokhoz felhasznált elektronikus aláírásokkal és hozzájuk tartozó tanúsítványokkal, valamint egyes egyéb, közigazgatásban felhasznált tanúsítványokkal illetve az azokat hitelesítő szolgáltatókkal kapcsolatos követelményekről” szóló kormányrendeletet.

Az elektronikus aláírás technológiájának hazai fejlődését követő szakembereket érdekelheti még, hogy több mint fél

év után megszűletett a részletes követelményspecifikáció az elektronikus aláírás és intelligens kártya használatához a közigazgatás informatikai biztonságának érdekében. Ez a hatalmas (több mint 1000 oldalas) munka, komoly eredménynek számít, mivel alapja lesz a technológia közigazgatásban történő elterjedésének. Az elképzelések megmértetése azonban csak most következik, nálunk a Magyar Elektronikus Alírási Szövetségben (MELASZ) és a szélesebb szakmai-üzleti körökben egyaránt. Olvasóimnak, kollégáimnak azt javaslom, tanulmányozzák az új törvénnyel együtt ezt az anyagot is.

ALMÁSI JÁNOS
info@sansserif.hu
A szerző a MELASZ elnöke
www.melasz.hu

A cikkhez kapcsolódó URL-ek:

<http://www.ihm.hu/jogszabalyok/>
<http://www.nhh.hu/>
<http://www.melasz.hu/>
<http://sansserif.hu/ealairas/>
<http://www.fogalomtar.hu/>
<http://www.mavinfomatika.hu/ca/>
<http://www.netlock.net/>
<http://www.eszigno.matev.hu/>
http://www.ictsb.org/EESS1_home.htm

Service Pack 1 szervizcsomag a SharePoint termékekhez

Jobban kezelhetők a nagyméretű dokumentumok és a frissítések, hatékonyabb lett a keresés és az adatbázisok visszaállítás.

A Microsoft kibocsátott két új szervizcsomagot a Microsoft SharePoint termékekhez és technológiákhoz, amelyek megnövelik a Microsoft Office SharePoint Portal Server 2003 és a Microsoft Windows SharePoint Services teljesítményét. Az egymástól függetlenül letölthető két szervizcsomag segítségével hatékony és megbízható virtuális munkaterületek készíthetők intranetes és extranetes csoportok számára. A stabilitás és a biztonság növelése mellett a Windows SharePoint Services SP1 a következő előnyöket is nyújtja:

- **Nagyobb fájlok támogatása.** Az SP1 segítségével akár 2 GB méretű fájlok is menthetők és megoszthatók.
- **Egyszerű frissítés.** A frissítések révén lényegesen egyszerűbbé válik az esetleges hibák kijavítása.

A SharePoint Portal Server 2003 használoinak előbb telepíteniük kell a Windows SharePoint Services SP1 szervizcsomagot, és csak ezután tölthetik le és telepíthetik a SharePoint Portal Server 2003 SP1 szervizcsomagot, amely az alábbi területeken nyújt továbbfejlesztéseket:

- **Jobb keresési eredmények.** A SharePoint Portal Server 2003 SP1 számos területen javítja a keresést, például javult a terjesztés, az indexelés megbízhatósága, a kulcszavas és a tipp alapján történő keresés, a figyelmeztetési feltételek megadása, a nyelvi precizitás, és robusztusabb lett az XML-szűrés.
- **Content Watson funkció.** Javult a termék minősége, mivel a felhasználó hálózati számítógépe és a Microsoft fejlesztőcsoportja között egyszerűbben továbbíthatók a hibajelentések.

Az SP1 ingyenesen letölthető.

Windows SharePoint Services SP1:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=875DA47E-89D5-4621-A319-A1F5BFEDF497&displaylang=en>

SharePoint Portal Server SP1:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=FD3AB75D-C622-4488-BD06-8F5D8347E3D2&displaylang=en>

További információ:

<http://www.microsoft.com/hun/office>

Microsoft CRM v 1.2 – 2. rész

A CRM SZERVER ÉS A KLIENS ALKALMAZÁSOK TELEPÍTÉSE

Ebben a részben megvizsgáljuk a CRM telepítési megoldásait, és ismertetjük, milyen előkészületeket és beállításokat kell elvégeznünk a rendszer bevezetéséhez.

1. Igényfelmérés, hardverszükséglet

Mielőtt belevágnánk a CRM telepítésébe, fontos felmérnünk a vállalat jelenlegi igényeit, és azt, hogy a közeljövőben várhatóan milyen bővítéssel számolhatunk (ez elsősorban a felhasználók számát jelenti). Ez határozza meg ugyanis, hogy milyen hardver- illetve szoftverigényei lesznek CRM alkalmazásunknak:

- 40 felhasználóig (25 ajánlott): „minden egy dobozban”, azaz egy gépre telepítjük a CRM szervert és minden egyéb komponenst, amire a CRM-nek szüksége van - jelen esetben az operációs rendszerek közül a Windows 2003 SBS Server Premium lesz a gazdaságos megoldás. Itt a minimális hardverigény: 700 MHz Xeon III processzor, 512 MB RAM, és lehetőleg három merevlemez, RAID 5 konfigurációban. Ha erre nincs módunk, természetesen megteszi a tükör is, de mindenképpen valamilyen hibátűrő megoldást válasszunk. Tesztelés során mi egy P4 2,4 GHz-es processzorral és 1 GB RAM-mal szerelt asztali gépet használtunk, ezen már szépen működik a CRM, pedig az SBS 2003 szerver már önmagában is meglehetősen erőforrás-igényes szeret.
- 100 felhasználóig (50 ajánlott): „három dobozos” megoldás javasolható. Egy gépen van a CRM szerver és az SQL Server 2000. A hardverigény ugyanaz, mint az előbbinél ismertetett esetben. Külön gépeken foglal helyet az Active Directory tartományvezérlő (DC) és az Exchange Server. A hardverigényekre itt most külön nem térünk ki, elég annyit, hogy az újonnan vásárolt gépek valószínűleg többszörösen túl fogják ezeket teljesíteni, már persze amennyiben a 100 helyett nem 500 felhasználót akarunk ráerőltetni.
- 300 felhasználóig (150 ajánlott): ez már legalább négy dobozt jelent. Az előzőhöz képest a különbség annyi, hogy külön gépre telepítjük a CRM szervert és az SQL-t. Ha ennél több felhasználót kell kiszolgálnunk, akkor egyéb „trükkökhöz” kell folyamodnunk: az SQL-nél és az Exchange-nél megszokott módon a CRM szervert is elhelyezhetjük fűrtökben, ettől kezdve pedig már csak az határozza meg az ellátható felhasználók számát, hogy mekkora a szerverfarmunk.

2. Szoftverigények

Nézzük először az operációs rendszert! A Windows 2003 SBS-ről már esett szó, kisvállalati környezetben ezt használhatjuk az „egy dobozos” megoldásnál. A Windows 2003 Server sorozatból használható még a Standard, az Enterprise, illetve a Web Edition, a friss SP-k telepítésével. Szóba jöhet még természetesen a Windows 2000 Server illetve Advanced Server is, de használatukat már nem javasoljuk. Az operációs rendszeren kívül a következőkre lesz még szükségünk:

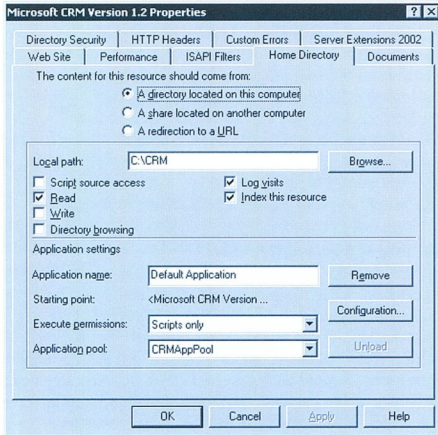
- Microsoft Message Queuing (MSMQ)
- Indexing Service
- Microsoft Data Access Components (MDAC): 2.71a (Win2K Server esetén), 2.8 (Win2K3 Server esetén)
- Internet Information Services (IIS): 5.0 (Win2K), 6.0 (Win2K3)
- Microsoft Core XML Services (MSXML): 4.0-ás verzió SP2-vel. A CRM szerver telepítésekor automatikusan telepítésre kerül, ha nincs fenn.
- Microsoft .NET Framework 1.1. Szintén telepítésre kerül a CRM-mel együtt.
- SQL Server 2000 a legújabb SP-vel és javításokkal
- Exchange Server 2000 vagy 2003 (opcionális, ha az e-mail router funkciókat is használni szeretnénk)

3. Teendőink a telepítés előtt

Ha a fentiekben már túljutottunk, következhet a szoftverkörnyezet előkészítése a CRM „fogadására”. Ehhez a következőket kell tennünk:

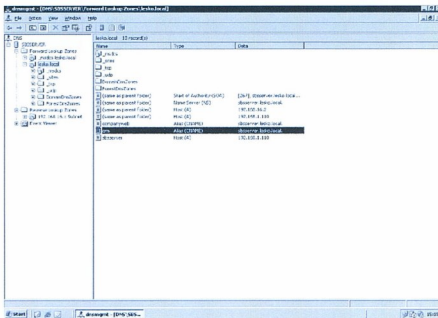
- Active Directory előkészítés: először is mindenki megnyugtatására, a CRM telepítés nem fog sémamódosítást végezni. Bizonyosodjunk meg róla, hogy az AD natív vagy Windows 2003 módban működik. Ezután mindössze annyit kell tennünk, hogy létrehozunk a címteremben egy szervezeti egységet, ami „alá” majd a CRM üzleti szervezeti egységei (*Business Unit*) kerülnek. Emellett ajánlott egy tartományi rendszergazda jogosultságú CRM rendszergazda fiók létrehozása is, akinek a nevében majd telepíteni fogunk. Ez a felhasználó lesz a CRM rendszer *rendszergazda* (*System Administrator*) szerepkörrel felruházott első felhasználója (CRM Admin).

- IIS konfiguráció: CRM-ünk helye soha ne az alapértelmezett (Default Web Site) legyen, bármennyire is szeretnénk! Hozzunk létre egy külön site-t, majd rendeljük hozzá CRM szerverünk IP címét, illetve a 80-as TCP portot. Host Header-ként írjuk be, milyen URL-en szeretnénk elérni az alkalmazást. Adjuk meg a Home Directory helyét!



■ Home Directory helye és jogosultságok az IIS-ben

- A DNS konfigurálása: természetesen ahhoz, hogy a kliensek megtalálják a CRM webhelyet, be kell jegeznünk azt a DNS-be is. Ehhez elég a megfelelő Forward Lookup zónában egy Alias (CNAME) rekord létrehozni, amely a Host Header-ként megadott DNS nevet tartalmazza, és szerverünk Host rekordjára mutat. Ez persze csak akkor igaz, ha a CRM szerverünk Host rekordja már megvan. Ha nincs, először azt kell bejegyeznünk. Megjegyezzük, hogy ez legfeljebb akkor fordulhat elő, ha elfelejtettük engedélyezni a dinamikus regisztrálást a CRM szerver TCP/IP DNS tulajdonságai között.



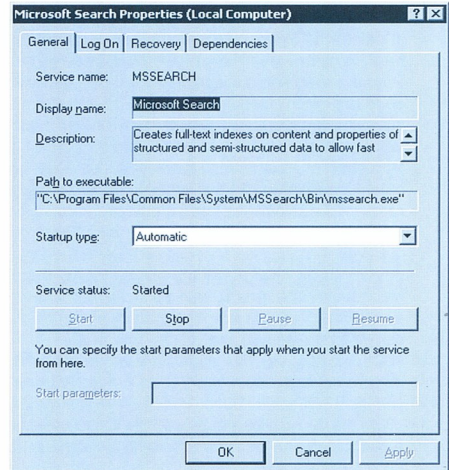
■ A CRM alias (CNAME) rekordja a DNS-ben

- SQL Server: az SQL Server Enterprise Management konzolon a CRM-et kiszolgáló SQL szerver felhasználói fiókjaihoz adjuk hozzá a CRM rendszergazdát, és válasszuk hozzá Rendszergazdák (System Administrators) szerepkört.
- Végül ellenőrizzük, hogy a következő szolgáltatások futnak-e, és állítsuk be őket automatikus indításra: SQL Full Text Search, SQL Server Service, SQL Server Agent, Windows Indexing Service, Microsoft Message Queuing Service (MSMQ), World Wide Web Publishing, IIS Admin Service

4. A CRM szerver telepítése

Ha az előzőeken sikeresen átküzdöttük magunkat, akkor a CRM szerver telepítése innen már gyerekjáték lesz.

- Jelenkezünk be az általunk létrehozott CRM rendszergazda felhasználóval! A telepítő CD behelyezése után a telepítő automatikusan elindul, itt válasszuk az *Install Microsoft CRM Server* pontot. A licenccfeltételek elfogadása után meg kell adnunk a vállalat nevét és licenccsolt. A többi Microsoft termékektől eltérően a kulcs itt kis- és nagybetű érzékeny, illetve a vállalat nevéhez hozzárendel, azaz pontosan kell megadnunk mindkettőt! Több licenccsolt is megadhatunk;

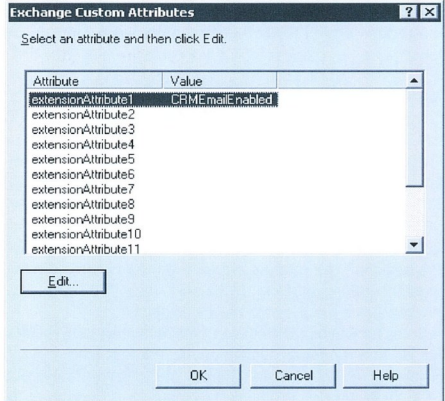


■ Full Text Search beállítása automatikus indításra

a licenck típusát, illetve a felhasználók számát az ablak alsó részén követhetjük nyomon.

- Ezután telepítésre kerülnek a szükséges egyéb komponensek, feltéve ha nincsenek már meg: az MSXML 4.0, a Microsoft .NET Framework 1.1 és a Win2K Hotfix Q320211.
- A következő ablakban (*Select SQL Server* ablak) válasszuk ki, hogy melyik SQL szervert, használja a CRM, majd a *Set Up Organization* ablakban adjuk meg, hogy az AD melyik szervezeti egysége lesz a CRM szülő szervezeti egysége (Parent OU).

- A következő a *Select Installation Locations* ablak itt válasszuk ki azt a webhelyet, amit a CRM-nek szánunk.
- A következő ablak (*Specify Security Account* ablak) már kicsit érdekesebb lehetőségeket kínál fel. Itt adhatjuk meg ugyanis, hogy melyik szolgáltatás milyen felhasználó nevében fusson. A három szolgáltatáshoz (Microsoft CRM, Crystal Enterprise, ASP.NET) akár külön-külön felhasználói fiókokat is rendelhetünk. Bár alapértelmezésben mindhárom a Local System nevében fut, biztonsági okokból javasoljuk, hogy külön-külön tartományi felhasználókat rendeljünk hozzá az egyes szolgáltatásokhoz. Fontos, hogy ezeknek a felhasználói fiókoknak ne legyenek rendszergazdai jogosultságai; elég, ha egyszerű tartományi felhasználók.
- Már csak annyi teendőnk maradt, hogy nyomjuk meg az *Install* gombot, és várjuk türelmesen, míg a CRM szerver települ, majd gépünk újraindul. Jó lenne, ha az újraindítás nem lenne szükséges egy szerver esetében, de sajnos itt elkerülhetetlen...

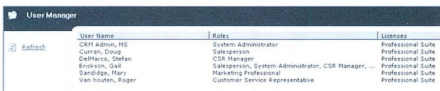


■ CRM e-mail kezelés beállítása (Exchange)

lónál be kell állítanunk, tehát bizonyos felhasználószám felett megéri írni hozzá egy szkriptet. Ezek után a router át fogja küldeni a beérkező leveleket a CRM rendszernek is, természetesen az ottani beállításoknak megfelelően, azaz vagy mindent, vagy csak az azonosítóval ellátottakat. Ezt a CRM-ben a *Tools* → *Options* → *Activities* menüben tehetjük meg, ahol két lehetőség közül választhatunk: *All incoming e-mail* (azaz mindent beengedünk) vagy *Only e-mail about existing Microsoft CRM records* (csak azoknak a leveleknek a beengedése, amelyek CRM rendszerből indított e-mail tevékenységekre érkező válaszok).

5. Teendőink a telepítés után

- Jelentkezünk be újra a CRM Admin felhasználókkal, a szolgáltatások között keressük meg a *CRM Security Service*-t, és indítsuk el. Elvileg elindul magától, de a gyakorlatban sajnos 10-ből 9-szer nem...
- CRM Admin felhasználónk még sok mindent nem tud csinálni, amíg hozzá nem rendelünk egy felhasználói licenct. Ezt már a CRM kezelői felületéről tehetjük meg: *CRM Home Page* → *Business Units Settings* →



■ CRM felhasználók és a hozzájuk rendelt licenck

Users és itt adjunk egy licenct a rendszergazdánk-nak. Megtehetjük ezt a *Deployment Manager*-ből is.

6. Az Exchange E-mail Router telepítése

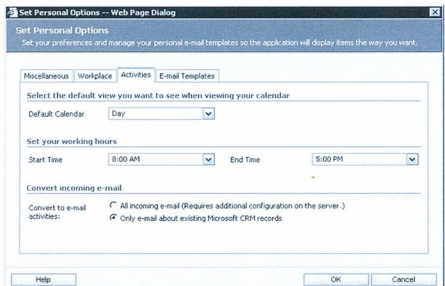
Ezt a komponenst az Exchange Serverre kell telepítenünk, melynek verziója 2000 vagy 2003 lehet. A telepítést végző felhasználó mindenképpen tartományi rendszergazda (Domain Administrator) és helyi rendszergazda (Local Administrator) jogosultságokkal kell hogy rendelkezzen. Maga a telepítés a CRM Server CD-ről automatikus, mindössze a CRM szerver nevét, a CRM site URL-jét és a CRM Admin felhasználó nevét illetve jelszavát kell megadnunk. Ezt követően kell konfigurálnunk a routert:

- Amelyik felhasználónál engedélyezni akarjuk a router használatát – azaz hogy levelei bekerüljenek a CRM rendszerbe –, a felhasználói fiók tulajdonság alapján a következő beállításokat kell elvégeznünk: az *Exchange Advanced* → *Custom Attributes*-ban adjunk a felhasználónak *Associated external account* jogosultságot, majd *Exchange Advanced* → *Custom Attributes*-ban válasszuk ki egy még szabad attribútumot (például *extensionAttribute1*) és írjuk át a következőre: *CRMEmailEnabled*. Ezt egyenként minden felhasználónál

7. A Crystal Enterprise Administrator jelszavának megadása

A CRM szerver telepítésekor a Crystal Enterprise Administrator-nak még nincs jelszava. Jelen esetben nem egy AD-ban is létező rendszergazdáról van szó, hanem ő lesz az a Crystal Enterprise felhasználó, aki a jelentéseket módosítani tudja, illetve új jelentéseket készíthet. Ennek a biztonsági résnek a betöméséhez a következő programot kell futtatnunk:

```
<InstallationRoot>\Program Files\Microsoft CRM
\Tools\CrystalAdmin.exe
```



■ E-mail fogadás tulajdonságainak beállítása

A régi jelszó helyét üresen hagyjuk (mivel nincs), és megadunk egy új jelszót. Gyorsan csináljuk meg, amint lehet!

8. A kliens szoftverek telepítése

Természetesen, ha böngészőn keresztül akarjuk elérni a CRM-et, sok teendőnk nincs, hiszen Internet Explorer van a kliens operációs rendszereken. Csak a minimális rendszerkövetelményekre kell odafigyelnünk: PIII 500 MHz 64 MB RAM-mal, az operációs rendszer pedig lehet Windows 98, 98 SE, Millenium Edition, 2000 (Professional vagy Server), 2003 (Web, Standard, Enterprise) vagy XP (Home vagy Professional). Az Internet Explorer 6.0-ás verziójú legyen, legalább SP1-gyel. Nem árt az sem, ha a CRM site-unkat hozzáadjuk a megbízható helyek listájához.

A Sales for Outlook Client telepítése már kicsit több munkával jár. Itt a rendszerkövetelmények: PIII 800 MHz 256 MB RAM-mal, operációs rendszerből pedig a Windows 2000 SP4, XP Professional SP1 jöhet számításba. Vigyázat: az XP Home Edition nem, ugyanis ott nincs IIS! A kliensgépek esetében mindenképpen figyelembe kell venni, hogy telepítéskor egy teljes funkcionális webkiszolgáló kerül a felhasználók gépére, így az ezzel kapcsolatosan felmerülő biztonsági kérdésekkel is foglalkoznunk kell. Legalább a beépített személyi tűzfalat használjuk, és gondoskodjunk a vírusvédelemről is. Szükségünk van még egy Office-ra is, ez lehet a 2003 bármely kiadása, Office XP SP2, vagy 2000 (amennyiben van hozzá SP3, SR1, illetve Microsoft Outlook 2000 Post SP3 Hotfix 228512), illetve az indexelő szolgáltatás. A telepítéskor felkerül még a kliensgépre az MSDE, az MSXML 4.0, a Microsoft .NET Framework 1.1 és a Windows Hotfix Q320211.

Fontos, hogy a telepítés előtt az Outlook-ot legalább egyszer el kell indítanunk, hogy létrehozzon legalább egy, alapértelmezett profilt!

Ezt követően már csak el kell indítanunk a telepítőt, és a licencceltételek elfogadása után a Sales for Outlook Client felkerül gépünkre: az Outlook indítása után tapasztalhatjuk, hogy a CRM beépülő moduljai megjelennek a felhasználói felületen.

9. CRM és Windows XP Service Pack 2

Amennyiben az operációs rendszerünk XP, és telepítjük az SP2-t, azt fogjuk tapasztalni, hogy a felhasználók nem érik el a CRM-et sem a böngészőn, sem az Outlook-on keresztül. Ennek számtalan, leginkább az új biztonsági elemek működéséből adódó oka van, mint például az előugró ablakok (pop-ups) blokkolása vagy a biztonsági tűzfal beállításai. Ezeket – illetve megoldásukat – helyszűke miatt itt részletesen nem ismertetjük. Egyik lehetőségünk a hibák elhárítására a 839939 számú frissítés telepítése a kliensgépre és a szerverre, az egyéb, elkerülő megoldásokról pedig itt olvashatunk:

<http://support.microsoft.com/default.aspx?kbid=870635>.

KOVÁCS LÁSZLÓ
(MCSE+S, MCT, CRM vezető oktató)
kovacs@sbuilders.hu
KOVÁCS ZOLTÁN
(MCSE, MCDBA, MCSD, MCT)
kovacsz@sbuilders.hu
A szerzők a Számalk Oktatási Rt.
További képzés bivatalt Microsofti oktatói.

Megjelent a Virtual Server 2005

A Virtual Server 2005, a Windows Server 2003 rendszerhez készített virtuálisgép-megoldás a „Dinamikus Rendszer Kezdeményezés” (Dynamic Systems Initiative Road Map) jegyében.

A Microsoft Magyarország a szeptemberi Fókuszpont partnerkonferencián jelentette be a Microsoft Virtual Server 2005 megjelenését és hazai elérhetőségét. A Standard és Enterprise kiadásban is megvásárolható Virtual Server 2005 segítségével csökkenthetők a hardverköltések, és növelhető a működési hatékonyság a következő három kiemelt területen: a szoftvertesztelés és -fejlesztő környezetek automatizálása, a régi alkalmazások áthelyezése, valamint a kiszolgálói terhelések – mint például a hálózatkezelés, a címár-infrastruktúra és a részlegszintű alkalmazások – összevonása terén. Az elosztott rendszerek megtervezését, bevezetését és üzemeltetését egyszerűsítő és automatizáló „Dinamikus Rendszer Kezdeményezés” egyik alapkövető jelentő Virtual Server 2005 révén rugalmasabban szabályozható az adatközponti erőforrások elérhetővé tétele. A Virtual Server 2005 korai bevezetői 50–95 szá-

zálékkal tudták csökkenteni a kiszolgálók üzembe állítására fordított időt, míg a rendszerrel megvalósított konszolidáció révén 50–80 százalékkal tudták csökkenteni a fizikai kiszolgálók számát. A Microsoft Virtual Server 2005 lehetővé teszi a virtuális gépek egyszerű telepítését és kényelmes, webes felügyeletét, emellett robusztus adattárolási és hálózati szolgáltatásokat is biztosít. A Microsoft és más gyártók továbbfejlesztett szerverfelügyeleti eszközeivel egyszerűen felügyelhetők a több virtuális kiszolgálót futtató szerverek. Például a Virtual Server 2005 Management Pack for Microsoft Operations Manager 2005 segítségével egy pontból figyelhető és kezelhető több virtuális és fizikai kiszolgáló.

További információ:

<http://www.microsoft.com/hun/windowsserversystem/virtualserver/bemutato.mspx>
<http://www.microsoft.com/hun/windowsserversystem/>.

Ami a hivatalos Microsoft tanfolyamokból kimaradt...

MICROSOFT OFFICE 2003 – INFOPATH

Rovatunk mostani cikkében egy olyan termékről írunk, amely olyannyira „kimaradt”, hogy jelenleg egyetlen hivatalos tanfolyam sem foglalkozik vele. Az InfoPath egyike az egyre népszerűbb kis és középvállalatok részére kínált intranetes „csináld magad” megoldásoknak. Remekül integrálható a meglévő információs portálokba, professzionális megoldásokat nyújtva mindenki megalgására.

Az egyik „csináld magad” termék tehát az InfoPath, melyet a Microsoft a 2003-as verzióval fémjelzett Office termékcsomagjában jelentetett meg elsőként. Az InfoPath segítségével űrlapokat készíthetünk, és azokat publikálhatjuk meglévő SharePoint Portal szerverünkbe. Űrlapot tudunk eddig az Excelben is, Wordben is, meg akár az Outlookban is készíteni kényünkre-kezdünkre (a FrontPage-ről nem is beszélve), viszont ahány probléma, annyi féle kinézetű és adattartalmú megoldás született. Az InfoPath ezt a szerteágazó problémahalmazt hivatott rendezni intranetes életünkben, kihasználva az XML szabvány minden előnyét. Nézzük...

A feladat

Egyszóval van ez a jó kis új terméke a Microsoftnak. A sablongyűjteményéből ki is nyomtattam pár kimutatást, amit kézzel kitöltve le is fűztem, a többi soha meg nem nézendő dokumentum mellé az irattárban. Elgondolkodtam (de tényleg ☺)! Biztosan nem ez a csúcsra járatott felhasználási módja ennek a terméknek. Szóval elgondolkodtam, és elővettem az egyik legalapvetőbb céges problémát, mely szinte minden kis- és középvállalatnál komoly elvárás az IT csapat tagjaitól.

Szabadságigénylés elektronikusan

Kezdjük is neki (ügyesebbek ilyenkor kiadják az egészet egy gyakoronknak, és kipipálják a feladatot a listában). Gyűjtsük össze milyen adatokra lesz szükségünk a szabadságigénylés automatizálásához. Mindenekelőtt legyen automatikus a dátum kezelése, mert ha kézzel nem datálható vissza az igénylés, akkor már elégedetten mosolyogni láthatjuk pár pillanatra főnökünket.

Szükséges továbbá az igénylést kitöltő alkalmazotról egy jó pár adat, melyek szintén automatikus kitöltésével biztosan azonosítani lehet az igénylő személyét minden kétséget kizáróan. Ugye, ha már visszamenőlegesen nem lehet, akkor más helyett se lehessen. Ilyen adatok:

- a felhasználó neve;
- e-mail címe (ez egyedileg azonosítja a felhasználót);
- munkaköre;
- csoportja / osztálya /szervezete;
- info mezője (egyedi ez-meg-az).

Most jöhet a szabadságigénylés maga. Ezt egy szintén automatizált táblázatban képzelem el, melynek sorait tetszés szerint adagolhatjuk. Sok helyen a főnök ragaszkodik hozzá, hogy ha a kedvenc alkalmazottja elmegy szabadságra, lehessen tudni, kit kell letorkolni, ha nem haladnak az átadott projektek. Tehát számoljunk helyettesítési információkkal is. Persze a szabadságigénylőnek meg kell adni a lehetőséget, hogy szabadságával kapcsolatos (soha figyelembe nem vett) megjegyzéseit örökkül hagyhassa az utóknak.

Tulajdonképpen végére érve a feladat megtervezésének, már csak az aláírások vannak hátra. A digitális jövő hajnalán kézenfekvő, hogy digitális aláírásokat kezeljünk. Vagy mégse? A jogszabályok mai állása szerint egy ilyen dokumentumon lehet alkalmazni ugyan a digitális aláírást, de mindenképpen le kell fűzni belőle egy példányt az irattárba, ha hivatalosnak minősítette a dokumentumot, a cég belső szabályzata. Ha már ennyit szenvedtünk, akkor legyen hát, és irassuk alá mindenkivel kézzel, akivel csak lehet.

Vegyünk egy kis levegőt!

Tegyük azt, hogy minden részfeladatot különböző InfoPath-os megoldással oldjunk meg.

- Az alapadatokat (dátum és az igénylő neve) az ActiveDirectory-ból vegyük automatikusan, annak tükrében, hogy ki van bejelentkezve a Windows-ba.
- Az elérhető személy (helyettes) adatait és a megjegyzés rovatot csak akkor kelljen kitölteni, ha ténylegesen tartalmaz adatokat, egyébként meg se jelenjen a nyomtatásban.
- Az aláírás szakasz pedig mindig annyi sort hozzon egymás alatt automatikusan, ahány felettesünk van a céges ranglétrán.

Csajunk a lecsóba!

Minden reklám és leírás attól hangos, hogy az Office 2003 – és ezáltal az InfoPath – teljesen integrált megoldást nyújt a vállalat intranethez. Tehát az első lépés, miszerint vegyük ki a szükséges adatokat az AD-ból, gyerekjáték kell, hogy legyen. Mert ugye a vállalati felhasználóinkat a SharePoint Portal Server is az AD-ben kezeli. Nyomkodom jobbról, nyomkodom balról az InfoPath-t, de sehol egy menü vagy gomb, ami csak úgy bedobná az AD-ban elérhető felhasználók adatait. MZ/X-et idézve: ahogy azt Móríca elképzelel! Persze csodát vártam, hisz ha ez így lenne, lényegében bárki kényére-kedvére nyűstölthetne direktben a címtárat, ami nem biztos, hogy ideális lenne. Viszont az adat importálásánál ott a választási lehetőség: a Webservice.

A Webservice

Tehát mielőtt bármihez is hozzányúlnánk, és végérvényesen elrontanánk az InfoPath doksinkat, el kell készítenünk egy teljesen átlagos, és egyszerű Webservice-t. Ennek a Service-nek olyan helyet kell találnunk IIS szerverünkön, ami már egy létező webes alkalmazás elérési helye is legyen, és a felhasználó kezelése „integrated Windows authentication”.

A Webservice két fájlból áll mindösszen. Egy service1.asmx fájlból, és dll-ből amit a kiszemelt webhely BIN könyvtárában kell elhelyezni. Az asmx fájl nem tartalmaz mást, mint ami tulajdonképpen megmutatja, hogy hol található a dll, és meg is hívja azt.

```
http://SERVER/service.asmx
<%@ WebService Language="c#"
Codebehind="Service1.asmx.cs"
Class="MOKService.Service1"
```

Tehát a fentiekből következtetve a http://SERVER/bin elérésen kell lennie a dll fájlunknak (pl.: MOKService.dll).

Mivel nem akarunk visszafelé tuszmakolni adatot a szerver felé mostani feladatunkban, ezért a Webservice-nek csupán egyetlen egy metódust kell kiszolgáltatnia. Így az is elégséges, ha ez a metódus csak meghívható, és generál számunkra egy XML-t. Ha ez sikerül, jók vagyunk, mert akkor az InfoPath már tud mit kezdeni az xml adathalmazunkkal (©).

Térjünk vissza pár sorig még a dll-re, mert mondhatnánk ez egy rutin ügy, de ne nézzük magunkat zseninek. Szóval először is össze kell szednünk, hogy a fenti feladatmegfo-

galmazásból milyen AD objektumok, mely adatait szeretnénk elérni. Bevetettem csodafegyveremet, és megkértem fejlesztői képességekkel megadott barátomat, hogy készítse már el ezt a dll-t csuklóból. Miközben Toci (Tóth László) készítette a dll-t, nagy lelkesedéssel magyarázta, mit is csinál, én meg próbáltam értelmes arcot vágni hozzá, nehogy kedvét vegyem.

A kulcs az egészben, a System.DirectoryService névtér. Itt lehet manipulálni az AD objektumait, és azok paramétereit. Aki az AD-ben kicsit járatosabb, annak nem okoz nehézséget az alábbi lista önálló összeállítás.

```
textArray2[1] = "mail";
textArray2[4] = "displayName";
textArray2[5] = "department";
textArray2[6] = "objectSID";
textArray2[7] = "manager";
```

Még nagyon fontos az is, hogy az adattartalmakat, miként kérdezzük le, tekintettel arra, hogy csak lekérdezzük, de nem írjuk vissza azokat. Az alábbi minta a „DisplayName”-et kéri el az AD-től. Mintájára el tudjuk készíteni a többi adatkérést is könnyedén.

```
if ((result1.Properties["displayName"] != null) && (result1.Properties["displayName"] Count > 0))
{
    data1.DisplayName = result1.Properties["displayName"][0].ToString();
}
```

A Toci lelkerre kötötte, miszerint nagyon fontos, hogy a „szálat” hol veszi fel az ember a dll-ben. Mindenesetre a Webservice az asmx, és a dll fájljal együtt letölthető a [http://www.ijqb.hu/webhelyről tömörített állományban](http://www.ijqb.hu/webhelyrol/tomrített_állományban).

Az űrlap

Eljuttottunk végre ide is. Mintha hazaértünk volna egy nagy tűráról. Ami hátravan, az már gyerekjáték.

Miután megnyitottuk az alkalmazásunkat, kezdjük úgy, hogy bizzunk az Office varázslójában és kérjük az új űrlap adatait adatforrásból.



A varázsló első kérdése tehát, hogy honnét vegye az adatot. Alapértelmezésben az XML formátumot erőlteti, de mi ne döljünk be neki, hisz nem ezért dolgoztunk idáig. Válasszuk a „Webszolgáltatás”-t.

Adja meg az adatforrás típusát az űrlaphoz:

- XML-séma vagy XML-adatfájl
- Adatbázis (csak Microsoft SQL Server vagy Microsoft Office Access)
- Webszolgáltatás

Mint a feladat definiálásánál tárgyaltuk, csak lekérdezi szeretnénk adatainkat a Webservice-től, visszafelé nem tudsz mákolunk semmit, egyelőre. Tehát válasszuk az „Adatok fogadása” lehetőségét.

- Adatok fogadása és küldése
- Adatok küldése
- Adatok fogadása

No, a következő kérdésnél meg kell adnunk a Webservice elérési helyét pontos útvonalal. A trükk (ha lehet egyáltalán annak nevezni, mivel oda van írva a példába is) abból áll, hogy ki kell egészíteni a sort egy „wsdl” kóddal. Ezen betűnégyes alapján a .NET generálja a Webservice-ünk technikai leírását az InfoPath számára.

Adja meg az adatforrásként használni kívánt webszolgáltatás helyét:

<http://SERVER/Service1.asmx?wsdl>

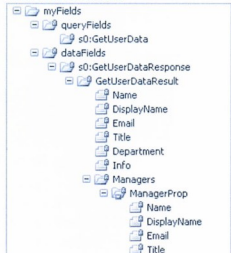
Példa: <http://www.contoso.com/Service.asmx?WSDL>

A folyamat megindul, az InfoPath felveszi a kapcsolatot az általunk megadott Webservice szolgáltatásával, és létrejönnek a környezeti leírások. Ugye csak adatkérés definiáltunk, ezért a pár pillanatig tartó kapcsolatfelvétel után választhatunk a felkínált lehetőségek közül. Jelen esetben az egyféle adatelérésből kiválaszthatjuk az egyfélélt (GetUserData).

Végezzünk a varázslóval! Persze ne véglegesen, kellhet még mások is.

Előnk táru lrlapunk teljesen „üres” képe. Ne szontyolodjunk el, ez már a sikeres adatelérést jelenti. Az adatforrások között feltűnnek a dll-ben nagy viszontagságok árán definiált AD mezőink. Tehát elérjük végre az ActiveDirectoryban tárolt objektumaink adatait, értékeit. Nézzünk végig alaposan az adatmezőkön és láthatjuk, ha jól készítettük el a Webservice-ünk dll fájlját, akkor szépen strukturáltan rendeződnek egymás alatti csoportokban a mezők. Nincs más dolgunk, mint az adatokat rádobálni a dokumentumtérre. Itt jól el lehet bibelődni. Tehát mielőtt ezt elkezdenénk, kukázzuk ki a „Lekérdező nézet”-et, hisz ezt a műveletet automata szeretnénk elvégezni.

Adatforrás használata:



Most már tényleg kezdhetjük az adatmezők feldobását. Először is egy „szakasz”-t kell definiálnunk, amibe az azonos tartalmú adatok kerülnek. Általában ilyen szakaszokba érdemes rendezni az InfoPath tartalmát a jó áttekinthető-

ség érdekében. Egyszóval az első „szakasz”-ba dobjuk bele a „RequestDate” adatelérést, a másodikba pedig a felhasználóra vonatkozókat. Kicsit rázzuk pofára (azaz formázzuk meg a külalakot is némileg).

Szabadság kérelem

Igénylés dátuma:

Szabadság

Alkalmazott adatai

Név:

E-mail:

Munkahely:

Csoport:

Info:

Az általános szakaszok egyik pozitív tulajdonsága, hogy mindig látszanak. Ez nem tűnik olyan nagy durranásnak, de emlékezzünk: a feladat megfogalmazásakor tettünk olyan bátor kijelentést, hogy a helyettes személy adatai, és a megjegyzés mezőt csak akkor jelenítsük meg, ha ténylegesen szükség van rájuk.

Kicsit jobban sajtóelnézve az eszközök között rábukkanunk közvetlenül a „szakasz” mellett a „választható szakasz” segédeszközre. Ki hitte volna? Dobjuk hát csak be! Nagyságrendileg pont azt értük el vele, amit szerettünk volna: az ilyen keretekben lévő adatok csak akkor jelennek meg a nyomtatásban, ha tartalmaznak információt, ellenkező esetben nem. Ugye arról nem feledkezzünk meg, hogy az itteni adatok azok bizony kitöltendő adatok, tehát nem kell az AD-ból ide semmi. Használjuk az alapértelmezett eszközeit az InfoPath-nak.

Távollétkor elérhető személy

Név:

Telefon:

E-mail:

Választható szakasz

Hátra van még többek között az egyik lényeges rész, maga a szabadságok igénylése. Ezt egy „Ismétlődő táblázat”-ban tudjuk kényelmesen megoldani, melyet beágyazunk egy teljesen normális szakaszba. Készítünk egy táblázatot ugye, aminek lesz egy fejléce, egy tartalmi sora (ez ismétlődik) és egy összegző, záró sora. A szépérezésem úgy kívánja, hogy a dátumokat kiválasztással adhassuk meg. Sajnos az ünnepnapokat és az egyéb munkaszüneti napokat nem tudja megfelelő biztonsággal kezelni a Windows, ezért a kivett szabadságok számát nem bízhatjuk egy algoritmusra, de az összegzést igen. No, már csak fel kell venni a táblában egy kiválasztó menübe a szabadságok fájljait (pl.: tanulmányi, beteg, stb.).

Távollét részletei

Kezdő dátum	Záró dátum	Típus	Napok száma
Válassz...	Válassz...	Válassz...	0
Kérnapok összesen			

Szabadság

Elérkezünk a lap aljára. Az aláírásoké a főszerep. Azt hátróztuk el a feladat megfogalmazásakor, hogy minden felesleges kapjon egy vonalat a lapra. A megoldás megint az ismétlődő táblázatnál keresendő, meg az AD adatokban. Addig ismétli önmagát, míg az AD lekérdezésből kapunk valós eredményt. Tehát megvizsgálja az „ismétlődő táblázat”, ki van e töltve a AD-ben a „manager” mező. Ha igen, készít egy új sort a táblában, és aláírja a személy nevét, majd folytatja a ciklust. Azaz megvizsgálja, hogy a felesleges szinten ki van e töltve a „manager” mezője az AD-ben. Ha igen, akkor megint húz egy vonalat és aláírja a felesleges feleslegesnek a nevét (és így tovább.) A ciklus akkor ér véget, ha nincs adat a vizsgált személynek (feleltesnek) a „manager” mezőjében.

Minden adat a helyére került az űrlapunkon. Még mindig nem egészen automata, mert ott virít a lekérdezés, illetve az adatfrissítés gombja. Dobjuk ki, ha eddig nem tettük meg. Keressük meg az Eszközök menüben a Parancsfájl almenüt, és választuk a „Betöltéskor...”-t, azaz editáljuk a betöltéskor automatikusan elinduló script-et.

Rá kell venni ezt a script-et, hogy csinálja meg a dátum, illetve az összes AD-ből lekért adat betöltését automatikusan a dokumentum megnyitásakor. Ennél tehát mi sem egyszerűbb. Nézzük meg a dátum, és az idő frissítési esetét:

```
function XDocument::OnLoad(eventObj)
{
  XDocument.Query();
  var now = new Date();
  setNodeValue("dfs:myFields/my:Request/my:Request
  ate",.getDateString(now));
  setNodeValue("dfs:myFields/my:Request/my:Request
  ime",.getTimeString(now));
}
```

Ha megvagyunk az összes lekérdezés megbütykölésével, nincs más hátra, mint publikálni az egészet a portálra. Nagyon fontos, és ezért nem szabad feledni, hogy az elkészült űrlapunk helyhez kötött! Azaz csak úgy, nem hurbicbálható, mint egy Word dokumentum. Tervezési módban kell publi-

kálnunk a megfelelő helyre, ami a SharePoint Portal 2003 Server esetében némi előkészítélet igényel. Létre kell hozni egy olyan kijelzőt, ami egy dokumentumtár is egyben, és InfoPath az alapértelmezett sablon formátuma. Ha nincs ilyenünk, sebj, ne dobjunk el mindent a kezünkön, hisz a közzététel során is elkészíthetjük a megfelelő portálhelyet.

Mire megyünk

Láthattuk az elején, hogy az adatokat nemcsak lekérni lehet, hanem egy kicsit bővebb WebService segítségével, vissza is tölthetjük azokat a portálba. De közvetlenül az adatbázisban az erre kialakított helyre is betölthetjük azokat, melyből a későbbiekben akár összesített lekérdezéseket is elereszthetünk (pl.: eddig ki, milyen, és hány nap szabadságot vett ki). Ha kicsit jobban belegondolunk, csak néhány apró kiegészítés, és akár komplett HelpDesk-et hozhatunk létre intranetünkön, vagy szervezési tudásbázist, stb. A mentett adatok igen kis helyet foglalnak portálunkon, és XML szabványú formátuma miatt az adattartalom más alkalmazásokkal kiválóan összehangolható. Azt hiszem – bátran mondhatom – hiába első verziós az InfoPath, igen jól sikerült megcélözni a Microsoft Office csapatának a termék felhasználási körét. A portálok és az automatizmusok kikerülhetetlenné válnak ma-holnap az intraneteken. Az InfoPath és az integrált XML technológia gyors alkalmazkodási képességet és megoldásokat nyújt erre.

FARKAS VIKTOR

*IQSOFT – John Bryce Oktatóközpont
farkas@kszki.obuda.hu
MCSE, MCT, HP ASE*

A cikkben szereplő állományok
letöltési helye:

http://www.iqjhb.com/UserFiles/tn_infopath.zip

Windows 2003 tanfolyamhoz Exchange 2003 kedvezmény

Hivatalos Windows Server 2003 tanfolyamokhoz most speciális Exchange Server 2003 kedvezményeket kínálunk: ingyen, „Exchange Server 2003 újdonságok” vagy kedvezményes Exchange 2003 rendszergazda tanfolyamot adunk. Részletek a honlapunkon.

Új hivatalos tanfolyamok!

ISA Server 2004, Microsoft Systems Management Server 2003.

Microsoft SA oktatási kuponok beválthatók

Nálunk beválthatja a Microsoft Software Assurance licenc vásárlása után kapott oktatási kuponjait! **Minden kupon 1 ingyenes tanfolyami napot jelent.**

Ha nem tudja, hogy mi ez a kupon, hívja munkatársainkat!

IQSOFT – John Bryce
OKTATÓKÖZPONT

IQSOFT – JOHN BRYCE
OKTATÓKÖZPONT KFT.

Cím: 1135 Budapest

Csata u. 8.

Web: www.iqjhb.com

Telefon: 236-6197, -8

E-mail: tanfolyam@iqjhb.hu


Microsoft
CERTIFIED
Partner

Learning Solutions

Dr. Watson

ÉLRE TÖRÜNK A BÍZTONSÁG TERÜLETÉN: WINDOWS XP SP2

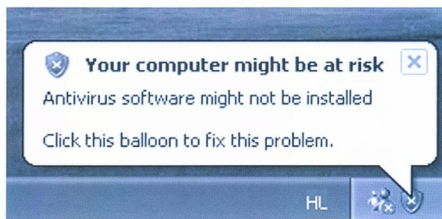
Megjelent a Windows XP soron következő javítócsomagja, a Service Pack 2. Érdemes alaposabban megismerkedni vele, ugyanis viszonylag sok és lényeges változtatást tartalmaz a biztonsági beállítások területén.

A tekintélyes méretű (256 MB) letölthető csomag általános bevezetése előtt érdemes tesztkörnyezetben játszódni vele egy picikét, mivel a biztonsági beállítások szorosabbra vétele miatt kompatibilitási problémák üthetik fel a fejüket. Mindenkinék azt javaslom, hogy inkább a régi ronsz szoftverét próbálja felhúzni arra a biztonsági szintre, amit az SP2 követel, mintsem hogy az SP2 lebutításával, a tűzfal kikapcsolásával és egyéb ármányok bevetésével bírja működésre azt. Most lássuk részletesen az XP SP2 új szolgáltatásait:

Windows Security Center

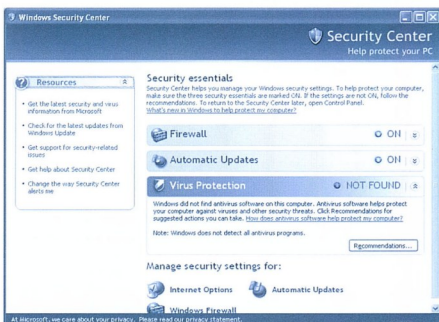
A Vezérlőpulton keresztül elérhető közös konzol a tűzfal, az Automatic Updates és az Internetbeállítások felé. Ez a komponens csak önálló (nem tartományi tag) számítógépeken vizsgálhatja a biztonsági beállításokat, és küld figyelmeztetéseket, mert a hozzá tartozó service, (ami figyelgeti, hogy minden rendben van-e a felsorolt komponensekkel, valamint a víruskergetőnkkel) tartományi tag gépeken nem indul el.

Ha azonban önálló gépünk van, így figyelmeztet például arra, ha nincs megfelelő víruskereső telepítve:



Hiányzik a víruskereső!

Ha a léggömbre (balloon) kattintunk, feltáru a Security Center ablaka (benne a jelzett problémával):



Tűzfal és Automatic Updates OK, de víruskereső nincs!

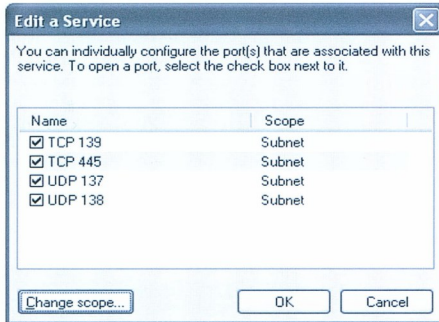
Kár, hogy a figyelmeztetést nem lehet (vagyis egyelőre nem tudjuk, hogyan lehet) finoman, kézzel ki/be kapcsolni, hanem a tartományi tagságtól (és a Security Center szolgáltatás futásától) függ.

Windows Firewall

A „védd magad” koncepció jegyében mostantól minden munkaállomáson bekapcsolt állapotban lesz a tűzfal. Ez már nem az Internetkapcsolat tűzfala, hanem egy, a többi szolgáltatással összeintegrált alkotmány, ami alapértelmezésben mindent zár. Ezt tessék szó szerint venni! A frissen telepített SP2-t még pingelni sem lehet (hacsak nem volt rajta korábban néhány megosztott könyvtár, lásd később), illetve mostantól mind a rendszertöltés (bootolás), mind a leállítás (shutdown) zárt ajtók (portok) mögött megy végbe. Ez egyébként egy óriási fegyvertény, még az ISA Server korábbi változata (ISA Server 2000) sem volt képes rá, aminek következtében távoli újraindítás esetén remeghetett a térdünk addig a 40-50 másodpercig, amíg a kiszolgálónk védelem volt. Vagy maradt az UTP-kábel kihúzása a bootolás idejére – de ehhez nyilván oda kell buszozni/metrózni az adott géphez. (Az ISA Server 2004 már bootproof.) De visszatérve eredeti témánkhoz, az SP2 tűzfala tartogat egy-két meglepetést. Sokakat zavarba fog hozni például, hogy az operációs rendszer többi szolgáltatásait tartott szoros kapcsolatának köszönhe-

tően automatikusan ki tud nyitni portokat, ha erre egy szolgáltatásnak szüksége van. Konkrét példaként említem a fájlmegosztást. Amíg nincs megosztva semmi, mind a NetBIOS-on keresztül, mind a natív (445-ös port, CIFS) megosztás-elérés tilva van. Hiába fut a Server szolgáltatás, és hiába lát-szik úgy a NETSTAT paranccsal, hogy a 445-ös port nyitva van, mert a tűzfal útját állja a behatólónknak.

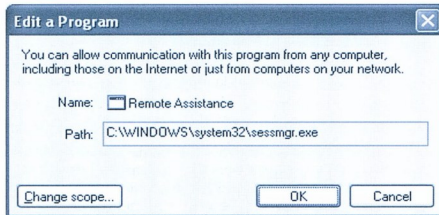
Ha azonban megosztunk egy könyvtárat, a megfelelő portok kinyílnak. De hogyan! Nem a nagyvilág számára, hanem csak a helyi hálózatra! Ha valaki egy másik alhálózaton található számítógépről próbálna idekapcsolódni, nem járna sikerrel. Meg is mutatom ezt egy képen:



Fájlmegosztás nyitva – de csak a saját hálózatunkra

Ha valaki mégis más hálózatról próbál belépni hozzánk, „tá-gasabbra” kell állítanunk a Scope (hatáskör??) opciót az alábbi lehetőségek valamelyikével: bárki bejöhet, illetve egy adott IP-cím tartomány gépeit fogadjuk el.

Érdekes megoldás a programok publikálása. Ha egy szolgáltatásról nem tudjuk eldönteni (mert nem dokumentált), hogy hányas porton fut, megpróbálkozhatunk az EXE fájl publikálásával. A gyári Remote Assistance is így érhető el távolról:



Alkalmazás elérésének engedélyezése

Ennek a megoldásnak külön előnye, hogy ha egy program mondjuk 1000 portot nyit, nem kell mind az ezret egyesével felvennünk, mert a Process ID (PID) alapján ezt megteszi a Windows tűzfala.

Fontos további opciók:

- Portszabályok átmeneti letiltása (repülőtéren ideális, gondoljunk a saját alhálózatra megnyitott portokra, ami adott esetben egy sereg idegen ember felé nyitja ki a gépet)

- Alapértelmezett állapot visszahozása (szétkonfigurálás után...)
- ICMP csomagok engedélyezése (PING be, Destination Unreachable ki stb.)



Download protection for Internet Explorer

Ez a komponens figyelmetztet, ha az Internetről veszélyes tartalmú fájlokat próbálunk letölteni.



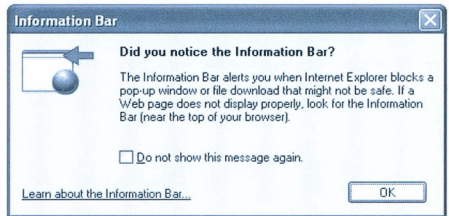
Attachment Manager

Csatolt fájlok „képeben” közeledő vírusok és férgék bejutásának megakadályozására szolgáló komponens.



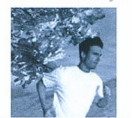
Pop-up Blocker in Internet Explorer

Vége a Microsoft Internet Explorer is tudja, amit a Mozilla már rég: ha egy hirdetésekkel zsúfolt weblap szeretné teleszemetelni a képernyőt újabb ablakkal, ezt most már meg lehet akadályozni az új komponenssel. Alapértelmezésben közepes (medium) erősséggel irtja a felugró (popup) ablakokat, ami azt jelenti, hogy ha egy ablak nem magától, hanem egy linkre vagy gombra kattintva jelenne meg, azt behozza, de az „önjáró” ablakokat nem. Érdemes kipróbálni akár melyik nagyobb hírportálon, milyen szépen megtisztítja a terepet! Első alkalommal figyelmetztet, hogy mit is művelt:



Rendőrség dönt a velencei magyar kiállításról?

Alig egy hét múlva kezdődik Velencében az Építészeti Biennálé, de az Index információi szerint a magyar kiállításon egyes műtárgyakat valószínűleg letakarna állítanak ki. Janesch Péter építész-kurátor ugyanis azt akarta bemutatni, hogy a szépség milyen sokféle lehet. Ehéhez a Szép Házak magazin fotói is felhasználta volna. A magazin főszerkesztője fejlejtést tett a rendőrségen és nem engedte bemutatni a fotókat.



Észrevetted, mit tettem? (Pop-up Blocker)

Ezen a kis ikonon kattintva lehetőségünk van a letiltott ablakok átmeneti engedélyezésére, a Pop-up Blocker beállításainak módosítására stb.

A többi komponensről már más cikkekben korábban volt szó, azokat most nem elemzem. De vannak lelkes fiatalok, akik ezt is kifejtik részletesen a mostani és a következő lapszámokban.

FÓTI MARCELL

marcell@netacademia.net

A szerző a NetAcademia vezető oktatója

MCE, MCT, MCDDB, MZ/X

Tisztelt Olvasónk!



Szeretettel meghívjuk az őszi TechNet szeminárium-sorozat során következő előadásaira. Reméljük, hogy a program elnyeri tetszését és előadásainkon üdvözölhetjük.

Biztonságos Windows (1. rész): A Windows XP biztonsága

IDŐPONT: 2004. szeptember 29.
HELYSZIN: Lurdy Ház, Hollywood Multiplex Mozi, (1097 Budapest, Könyves Kálmán krt. 12-14.)
ELŐADÓ: Fülöp Miklós

Az őszi TechNet előadások első három része egy sorozatot alkot, melynek témája a Windows alapú hálózatok biztonsága. Az első részben a vállalati munkaadások biztonságával kapcsolatos technológiákat ismertetjük. Ezek közül első helyen természetesen a fontos biztonsági újításokat tartalmazó Windows XP SP2 szervizcsomag áll, benne a Windows új tűzfalával, az Internet Explorer továbbfejlesztett biztonsági funkcióival, valamint a számítógép általános biztonsági állapotát felügyelő Biztonsági Központra. Az előadás további részében összefoglaljuk a biztonsági javítások kezelését támogató megoldásokat (az SMS 2003 illetve a MOM 2005 segítségével), valamint szó lesz a Windows XP helyi fájlrendszereinek titkosítását végző titkosított fájlrendszerrel, az EFS-ről.

Biztonságos Windows (2. rész): A Windows alapú balozatok biztonsága

IDŐPONT: 2004. október 13.
HELYSZIN: Lurdy Ház, Hollywood Multiplex Mozi, (1097 Budapest, Könyves Kálmán krt. 12-14.)
ELŐADÓ: Szalontay Zoltán

A Biztonságos Windows című sorozatunk második része a belső hálózatok biztonságát növelő technológiákat foglalja össze. Elsőként a hálózati felhasználók azonosításáról lesz szó, majd bemutatjuk, hogy a belső hálózatot alkotó Windows alapú gépek biztonságát beállítási hogyan írhatók elő és tarthatók be központiilag. A továbbiakban összefoglaljuk, hogy a Windows Server 2003 és Windows XP operációs rendszerek milyen újításokat tartalmaznak a nyilvános kulcsú infrastruktúra (PKI) terén. Ezt kö-

vetően áttérünk a belső hálózati adatforgalomra, melynek titkosítását és integritását az IPSec protokollal érhetjük el. Végül az előadás utolsó részében megmutatjuk, hogy a Windows Tartalomvédelmi Szolgáltatása (RMS) hogyan képes megakadályozni a belső vállalati információk kiszivárogtatását.

Biztonságos Windows (3. rész): Windows alapú internetes szolgáltatások biztonsága

IDŐPONT: 2004. október 27.
ELŐADÓ: Fülöp Miklós

A biztonságos Windows-ról szóló sorozat harmadik részében az interneten keresztül is elérhető vállalati szolgáltatások védelméről, valamint a belső vállalati hálózat interneten keresztül történő biztonságos eléréséről lesz szó. Az előadás során bemutatjuk, hogy az internetre csatlakozott felhasználók hogyan érhetik el a vállalati demilitarizált zónájában elhelyezett Exchange kiszolgálót böngészővel, vagy az Outlook ügyféllel. Szó lesz a vállalati internetes csatlakozási pontján történő vírus- illetve spamszűrési lehetőségekről, valamint bemutatjuk hogy az ISA Server 2004 segítségével hogyan alakíthatunk ki karanténnal védett VPN szolgáltatást.

Professionális üzleti diagramok és űrlapok a Microsoft Office Systemben

IDŐPONT: 2004. november 10.
HELYSZIN: Lurdy Ház, Hollywood Multiplex Mozi, (1097 Budapest, Könyves Kálmán krt. 12-14.)
ELŐADÓ: König Tibor

Régi adósságot teljesítünk, amikor az előadás első felében bemutatjuk a professzionális üzleti diagramok létrehozásának eszközeit, a Microsoft Office Visio 2003-at. Először az általános célú üzleti diagramok létrehozásáról, adatbázishoz kapcsolásáról és közzétételéről lesz szó. Ezután kitérünk a Visio informatikai területen való felhasználására (hálózatok, adatbázisok, fej-

lesztés), és bemutatunk néhány, a terméket kiegészítő külső alkalmazást is. Végül sorra vesszük a Visiora épülő fejlesztési lehetőségeket (saját alkalmazatok, stencilék, sablonok, Shape Studio, Visio ActiveX-vezérlő, XML-támogatás).

Az augusztus végén kiadott Office 2003 Service Pack 1 a legtöbb termék esetében csupán a szokásos biztonsági és egyéb javítócsomagokat tartalmazza, az üzleti űrlapok fejlesztésére és kitöltésére szolgáló Microsoft Office InfoPath 2003 azonban jelentős új képességekkel is bővült. Az előadás második felében élő példák segítségével, részletesen ismertetjük a felhasználói szerepek, a programozást kiváltó szabályok, a digitális aláírások, a kibővült SharePoint- és BizTalk-integráció, valamint a .NET-támogatás előnyeit.

A Microsoft Virtual Server gép technológiája: Virtual Server 2005 és Virtual PC 2004

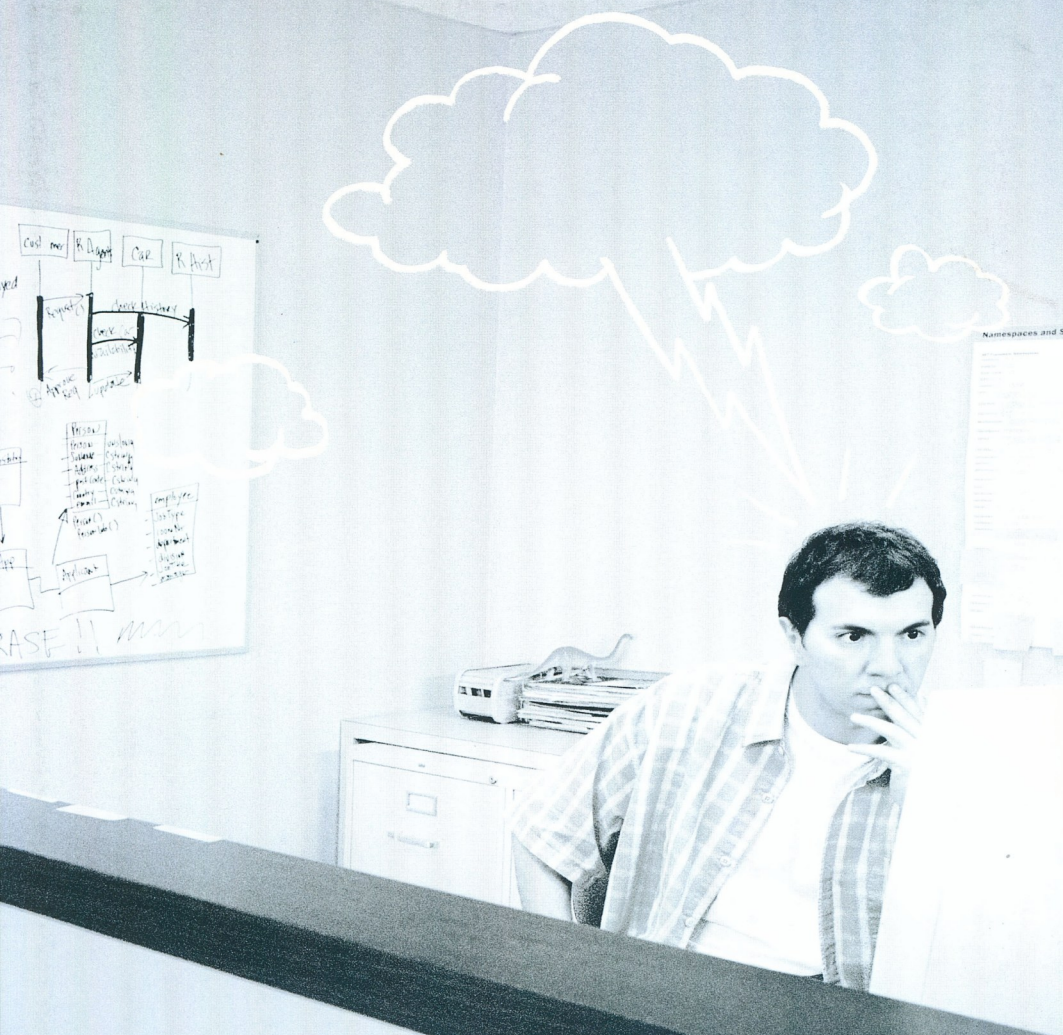
IDŐPONT: 2004. november 24.
HELYSZIN: Lurdy Ház, Hollywood Multiplex Mozi, (1097 Budapest, Könyves Kálmán krt. 12-14.)
ELŐADÓ: Szalontay Zoltán

Az őszi sorozat utolsó részében a tavaly óta már kapható Microsoft Virtual PC 2004 és az idén nyáron elkészült Microsoft Virtual Server 2005 termékszaladot fogjuk ismertetni és összehasonlítani. Az előadás első részében megismerkedünk a virtuális gép technológia alapjaival és főbb felhasználási területeivel (teszt rendszerek, fejlesztési gépek, szerver konszolidáció stb.). Ezt követően több virtuális gép példáján keresztül megismerhetjük a Virtual PC és Virtual Server komponenseit, valamint adminisztrációs felületeit. Szó lesz a virtuális gépek által használt lemezek típusairól és jellemzőiről, majd áttérünk a hálózati konfigurációra. Ismertetjük, hogy több virtuális gép futtatása esetén hogyan optimalizálhatjuk a host- és a virtuális gépek teljesítményét és végül példákon keresztül megmutatjuk a virtuális gépek automatizált üzemeltetését megkönnyítő programozási lehetőségeket.

Regisztráció és online közösség:

www.microsoft.com/hun/technet/

Az online jelentkezési lapot az adott esemény megrendezési időpontja előtt két héttel nyitjuk meg, amelyről e-mailben értesítést küldünk.



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.™

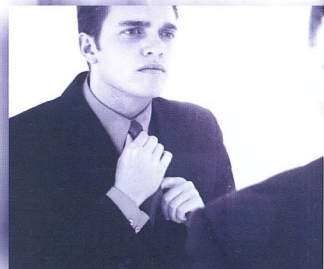
Microsoft

Arccal az MCSA felé!

Akciónk keretében a Microsoft Windows 2003 rendszeradminisztrátor minősítés (MCSA) megszerzéséhez szükséges 3 + 1 tanfolyamot megrendelők részére cégünk a **vizsgákat ingyenesen** biztosítja.

Az MCSA minősítés előnyei:

- Az Ön szakmai felkészültségének elismerése az iparágon belül.
- Közvetlen hozzáférés az MCP tagok weboldalán keresztül a Microsoft legfrissebb technikai és termékinformációihoz.
- Az adott minősítést igazoló hivatalos tárgyak (kitűző, névkártya, oklevél).
- Logók elérhetősége és használata (szabályozott kereteken belül).
- Ingyenes előfizetés a Microsoft MCP Online Magazine-ra valamint egyéb minősítés-specifikus számítástechnikai kiadványokhoz.
- Kedvezményes előfizetés a Windows & .NET Magazinokra.
- Illetve néhány közvetett előny (például karrierlehetőségek itthon és külföldön).



Az MCSA-képesítés a következő feladatok elvégzésére készít fel:

- Microsoft Windows Server 2003 környezetek menedzselése és hibaelhárítása.
- Active Directory kialakítása és üzemeltetése.
- Group Policy Objektumok létrehozása és konfigurálása. Felhasználói környezet kezelése csoportos házirenddel.
- Erőforrások hozzáféréseinek kezelése.
- A kiszolgáló teljesítményének monitorozása.
- Katasztrófa előtti állapot visszaállításának kezelése (Disaster Recovery).
- Szoftverek karbantartása Software Update Services használatával.
- TCP/IP alapú hálózati környezetek kialakítása.
- Biztonságos hálózat kialakítása.
- Munkaállomások telepítése és konfigurálása.
- Az új ISA Server 2004 konfigurálása és üzemeltetése.

A részvétel feltételei:

- A résztvevőknek **egy éven belül** el kell végezniük mind a négy MCSA tanfolyamot.
- Az ingyenes vizsgabónt mindig a következő tanfolyam megrendelésekor adjuk ki (tehát az első a második tanfolyam megrendelése után, a 2. bónt a 3. tanfolyam után stb.).
- Az „Arccal az MCSA felé!” akcióra 2004. december 31-ig fogadunk el jelentkezéseket!
- Az akció kedvezményei más kedvezménnyel nem vonhatók össze!

Tanfolyamok az MCSA-képesítés **kötelező** vizsgáihoz (mind a három vizsgát le kell tenni az MCSA-hoz)

2285– Installing, Configuring, and Administering Microsoft Windows XP Professional

A tanfolyamot Windows 2000 vagy 2003-as rendszergazdáknak ajánljuk, akik átfogóan szeretnék megismerni a Windows XP Professional újjdonságait, különböző telepítési módszereit, konfigurálását, üzemeltetését és támogatását Windows 2003 alapú hálózatokban.

Kapcsolódó vizsga : 70-270 – Installing, Configuring, and Administering Microsoft Windows XP Professional

2276/2277 - Implementing, Managing and Maintaining a Windows 2003 Server Network Infrastructure

Ez egy remek ötnapos tanfolyam arról, hogyan kell TCP/IP alapon felépíteni egy Windows 2003-hálózatot. A TCP/IP nélkülözhetetlen protokoll a Windows számára, ezért érdemes alaposan, tetőtől talpig megismerkedni vele. A tanfolyam komplex hálózati infrastruktúra kialakítását célozza meg. Ezen a tanfolyamon végigmegyünk a TCP/IP protokollcsalád elemein, és sorra megvizsgáljuk, melyiknek mit köszönhet a Windows. Gondolták volna, ha nincs ARP, nem létezhetnének fürtök (cluster) sem?

Főbb témakörök: RAS: betárcsázás, virtuális magánhálózatok kialakítása, útválasztási protokollok (RIP, OSPF). DHCP: IP-tartományok (scope) kialakítása, DHCP-opciók, foglalás (reservation), Relay Agent. Névfeloldás: DNS, zóna, zónatranszfer, DNS-rekordok típusai (NS, SOA, MX, A, SRV). Dinamikus DNS. Adatfolyam-titkosítás: IPSEC.

Kapcsolódó vizsga: 70-291 – Implementing, Managing, and Maintaining a Microsoft Windows Server 2003 Network Infrastructure

2273 – Managing and Maintaining a Microsoft Windows Server 2003 Environment

Ezen az ötnapos tanfolyamon felhasználók, csoportok, és Windows-erőforrások kezelését sajátítjuk el hálózatos (tehát címtáras) és hálózatmentes környezetben. Ez a tanfolyam szolgálhat kiindulási pontként a többi Windows 2003 tanfolyam elvégzéséhez.

Főbb témakörök: Felhasználók létrehozása, csoportba foglalása, szervezeti egységbe sorolása, kizárása, engedélyezése, jogosultságok állítása. Erőforrások (könyvtár, nyomtató) megosztása. A szervezeti egységek használata rendszerfelügyeletre: a csoportos házirendek. Erőforrások használatának naplózása (audit).

Kapcsolódó vizsga: 70-290 – Managing and Maintaining a Microsoft Windows Server 2003 Environment

Tanfolyamok az MCSA-képesítés szabadon választható vizsgáihoz (a három témakör közül egyet kell választani)

2824 – Implementing Microsoft Internet Security and Acceleration (ISA) Server 2004

Ezen a négynapos tanfolyamon az ISA Server főbb funkcióit vesszük sorra: a kimenő illetve bejövő hálózati kapcsolatok felügyeletét (tűzfal – Security), a hálózati kommunikáció gyorsítását, kiszolgáló-közzétételt (webkiszolgáló esetén fordított gyorsítótár - reverse proxy - funkcióval), behatolásérzékelést, alkalmazásérzékeny csomagszűrőket, például az áthaladó e-mailek ellenőrzése az esetleg veszélyes tartalmak kiszűrése céljából. A tűzfalfunkciót kiegészíti a hálózati címfordítás (NAT), illetve a Windows 2003 RRAS kiszolgálójával együttműködő VPN-támogatás. A teljesítmény fokozása érdekében az ISA Server nagyvállalati változata „fürtözhető”, a kiszolgálók ilyenkor gyorsítótárai tartalmát intelligensen megosztják egymás között.

Kapcsolódó vizsga: 70-350 – Implementing Microsoft Internet Security and Acceleration (ISA) Server 2004

2400– Implementing and Managing Microsoft Exchange Server 2003

Ez az ötnapos tanfolyam az Exchange 2003 üzemeltetéséhez szükséges feladatokat és tudnivalókat tartalmazza. Kiemelt hangsúlyt kap természetesen az SMTP-protokoll. Bár a tanfolyam az ügyfélprogramok (Outlook, Outlook Express) mindennapi használatára nem tér ki, az Exchange Serverrel való kapcsolattartás fontos témakör. Hogyan érjük el az Exchange Servert MAPI, POP3, IMAP4 és a többi protokoll segítségével? És ugyanezt tűzfalon keresztül?

A beérkezett levelek tárolásának megszervezése szintén jelentős feladat, alaposan megnézzük az Exchange Store-ok képességeit. És ne feledkezzünk meg az Active Directory-integrációról sem. Az Exchange System Manager felhasználói felülete javarészt nem más, mint ablak az Active Directory Configuration partíciójára. Mi minden kerül be az AD-ba az Exchange-nek köszönhetően?

Kapcsolódó vizsga: 70-284 – Implementing and Managing Microsoft Exchange Server 2003

2072 – Administering a Microsoft SQL Server 2000 Database

Az SQL Serverrel végezhető feladatok két részre oszlanak: alkalmazásfejlesztésre és üzemeltetésre. Ez a tanfolyam az üzemeltetéssel foglalkozik, amely meglepően sok lehetőséget tartalmaz annak dacára, hogy a kész adatbázisalkalmazást általában nem babráljuk. De a sebességén nagyot lendíthet a megfelelő indexturmix, a megfelelően elosztott adatbázisfájl és tranzakciós naplók stb. Ne feledkezzünk meg a mentésekről, a különböző mentési stratégiákról, s máris eljutunk a feladatok automatizálásáig: ha éjszaka kell menteni, ezzel ne nekünk kelljen foglalkozni! A feladatautomatizálás kapcsán előtérbe kerülnek a jobok és tárolt eljárások, és hirtelen rádöbbenünk, hogy mintegy 300 eljárást tartalmaz az SQL Server, ami hasznunkra lehet. És így tovább. Igen izgalmas kis tanfolyam.

Kapcsolódó vizsga: 70-228 – System Administration for Microsoft SQL Server 2000

Ha még elégedettebb arcot szeretne látni a tükörben...

Az MCSA minősítés megszerzéséhez szükséges vizsgák után a következő tanfolyamokkal szerezheti meg a Windows 2003 rendszermérnök minősítést (MCSE):

2278 – Planning and Maintaining a Microsoft Windows Server 2003 Network Infrastructure (kötelező)

Kapcsolódó vizsga: 70-293 - Planning and Maintaining a Microsoft Windows Server 2003 Network Infrastructure

2279 – Planning, Implementing and Maintaining a Windows Server 2003 Active Directory Infrastructure (kötelező)

Kapcsolódó vizsga: 70-294 – Planning, Implementing, and Maintaining a Microsoft Windows Server 2003 Active Directory Infrastructure

2810+2830 – Designing Security for Microsoft Networks (kötelező)

Kapcsolódó vizsga: 70-298 – Designing Security for a Microsoft Windows Server 2003 Network

2071 – Planning and Maintaining a Microsoft Windows Server 2003 Network Infrastructure (választható)

Kapcsolódó vizsga: 70-229 – Designing and Implementing Databases with Microsoft SQL Server 2000 Enterprise Edition

2072 – Administering a Microsoft SQL Server 2000 Database (választható)

Kapcsolódó vizsga: 70-228 – System Administration for Microsoft SQL Server 2000

Amennyiben kérdése van, vagy jelentkezni szeretne, keressen bennünket a lent látható elérhetőségek egyikén.

netACADEMIA
A LEGJÓBARÁT TANÍTJUK.

NetAcademia Kft.

Cím: 1062 Budapest Andrásy út 62.

Web: <http://www.netacademia.net>

E-mail: info@netacademia.net

Tel.: +36 1 4721214

Fax: +36 1 4721215

Irodai mobil: +36 20 3696947