

TechNet

2006. AUGUSZTUS-SZEPTEMBER

MAGAZIN MÉLYVÍZ, CSAK ÚSZÓKNAK!



Office

Microsoft®

DEP ÉS ASLR

A Vista memória- és programvédelmének néhány vonatkozása

FESTŐK ÉS BÖDÖNÖK

A skálázhatóság és a párhuzamosság kérdései

SQL SERVER 2005 ANALYSIS SERVICES

Példák az üzleti elemzést támogató legfontosabb funkciókra

**INTEGRÁLT
INNOVÁCIÓ-
TÖBB MINT A
RÉSZEK EGÉSZE:**

**MITŐL
RENDSZER?**

**ÚJ TÁBLÁK
AZ IRODABAN**

**AMIKOR A RÉGI
TALÁLKOZIK
AZ ÚJJAL**

Csomagoljon és induljon el a karrier útján!

Kedvezményes árú, rugalmas időbeosztású
Microsoft mérnök képzéssorozatok

Válassza ki az Önnek megfelelő minősítést és képzési konstrukciót!

Windows Server 2003 rendszeradminisztrátor (MCSA) képzés
80 óra – 390.000 Ft helyett **már 299.000 Ft-tól**

Windows Server 2003 rendszermérnök (MCSE) képzés
160 óra – 760.000 Ft helyett **már 609.000 Ft-tól**

Öt éve Magyarország legsikeresebb tanfolyamsorozatai!

SQL Server 2005 technológiai szakértő (MCTS) képzés
72 óra – 465.000 Ft helyett **már 299.000 Ft-tól**

SQL Server 2005 adatbázis-adminisztrátor és fejlesztő (MCITP) képzés
144 óra – 930.000 Ft helyett **már 599.000 Ft-tól**

Visual Studio 2005 technológiai képzés
80 óra – 465.000 Ft helyett **már 329.000 Ft-tól**

Visual Studio 2005 Windows, webes és nagyvállalati fejlesztői sorozatok kezdőknek és haladóknak

Új sorozataink

Microsoft SA utalványokat beszámítunk, a képzések szakképzési hozzájárulásból támogathatók! A képzések „all inclusive”-ek, az árak a tananyagokat és az étkezést is tartalmazzák! A csomagtól függően további tanfolyami és vizsgakedvezmények igényelhetők!

Az Ön cége is HEFOP nyertes? A Microsoft képzések mellett több száz, nem csak informatikai képzés közül válogathat! Kérje ajánlatunkat, keresse munkatársainkat:

Telefon: 203-0304/4122 m., e-mail: simonf@szamalk.hu



Microsoft
GOLD CERTIFIED
Partner

Advanced Infrastructure Solutions
Networking Infrastructure Solutions
Learning Solutions

SZÁMALK Továbbképzés www.szamalk.hu/tisza/mernokkepzesek

a mi tudásunk
az Ön sikere

Egyes csomagok összetétele még időközben változhat.
A tanfolyamokkal és kedvezményekkel kapcsolatos további
tudnivalók weboldalunkon találhatóak, vagy kérjük, keresse
szervezőnket! A feltüntetett árak a 20% ÁFA-t nem tartalmazzák.

A RUHA IS TESZI A SZOFTVERT!



Sziebig Andrea

Sziebig Andrea
főszerkesztő

Jópofa fülekkel, szalagokkal és egy sor más – nem csupán – látványelemmel újul meg szinte valamennyi Office-alkalmazás.

Magazából nem egyszerű megérteni, hogy miért kell radikálisan újratervezni egy szoftver felhasználói felületét, amikor jó, ha 5–10 százalékban használjuk például a különféle Office-funkciókat. Ráadásul ezek többnyire olyan szolgáltatások, amelyeket már magunkévá tetünk a jó öreg Office 95-ben, s aztán rutinosan használtuk őket a későbbi Office-verziókban: a 97-ben, a 2000-ben és az XP-ben is.

Persze munkavégzés közben az embernek mindig támadnak különféle ötletei, hogy mit is szeretne gyorsan, ügyesen, hatékonyan megcsinálni az Office-ban. Ilyenkor aztán folyik a küzdelem az ismeretlennel: sehogy se jön rá, hogy is kéne tető alá hozni azt a dolgot. Holott tudja, hogy benne van az épp aktuális verzióban... De alig olvasunk felhasználói kézikönyvet, és csak keveseknek van idejük elmenni a különböző Office-tanfolyamokra. Zsigerből érezzük ugyanakkor, hogy jó lenne mélyebben ismerni az Office-t, mert ezáltal hatékonyabbak lehetnénk.

Vélhetően az ilyen felhasználói attitűdű emberekre is gondoltak a Microsoft fejlesztői, amikor megpróbálják megkönnyíteni a menüparancsokban, eszközsávokban és munkaablakokban eltévelyedő felhasználók életét. Épp ezért bújtatták új ruhába a legtöbb Office-alkalmazást. Vannak olyanok – a legfontosabb tartalom-előállító alkalmazások –, amelyek mindenestül megváltoztak. Más applikációk – mint például az Access – jelentős fazonírozáson estek át, de a többi Office-alkalmazást egyelőre elkerülte a nagy átszabási projekt.

S hogy milyen is valójában az Office 2007? Kiderül, ha elolvassák magazinunk címlapos összeállítását. És persze kipróbálják az Office 2007 interneten is elérhető tesztverzióját.

SZERKESZTŐSÉG
Főszerkesztő
 Sziebig Andrea – asziebig@vogelburda.hu
Szakmai lektor
 Budai Péter – pbudai@microsoft.com
Vezető szerkesztő
 Varga János – jvarga@vogelburda.hu
Nyomdai előkészítés
 Budakeszi Bejárati Kft.
Korrektor
 Matula Zsolt
Lapterv és címlap
 Emotion Bt.

Szerkesztőség és kiadó címe:
 Vogel Burda Communications Kft.
 1077 Budapest, Kéthly Anna tér 1.
 Tel.: 888-3400, fax: 888-3499

KIADÓ
A Microsoft Magyarország
megbízásából kiadja
a Vogel Burda Communications Kft.
A kiadásért felel
 Carsten Gerlach ügyvezető igazgató
cgerlach@vogelburda.hu
 Tel.: 888-3470, fax: 888-3499
Lapigazgató:
 Walitschek Csilla
cswalitschek@vogelburda.hu, tel.: 888-3450

A TechNetben közölt cikkek fordítása, utánnyomása, sokszorosítása és adatrendszerekben való tárolása kizárólag a kiadó engedélyével történhet. A megjelent cikkeket szabadalmi vagy más védettségre való tekintet nélkül használjuk fel.

MÉDIAREFERENSEK:

Harsányi Erika – eharsanyi@vogelburda.hu, tel.: 888-3452
 Németh Krisztina – knemeth@vogelburda.hu, tel.: 888-3468
 Rátóti Sarolta – sratoti@vogelburda.hu, tel.: 888-3453
 Szendrey Szilvia – szendrey@vogelburda.hu, tel.: 888-3455
 Fax: 888-3459

Marketing:

Gajdos Barna – bgajdos@vogelburda.hu, tel.: 888-3494

Hirdetési koordinátor:

Szöke Erika – eszoke@vogelburda.hu
 Tel.: 888-3411, fax: 888-3459

Nemzetközi hirdetésfelvétel:

Eric N. Wicha – ewicha@vogelburda.com
 Vogel Burda Holding

Pocistrasse 11, D-80336 München

Tel.: +49 89 74642-326, fax: +49 89 74642-325

A hirdetések körültekintő gondozását kötelességünknek érezzük, de tartalmukért felelősséget nem vállalunk.

TERJESZTÉS

Terjesztett példányszám: 3000

NYOMDA:

Pauker Nyomdaipari Kft.

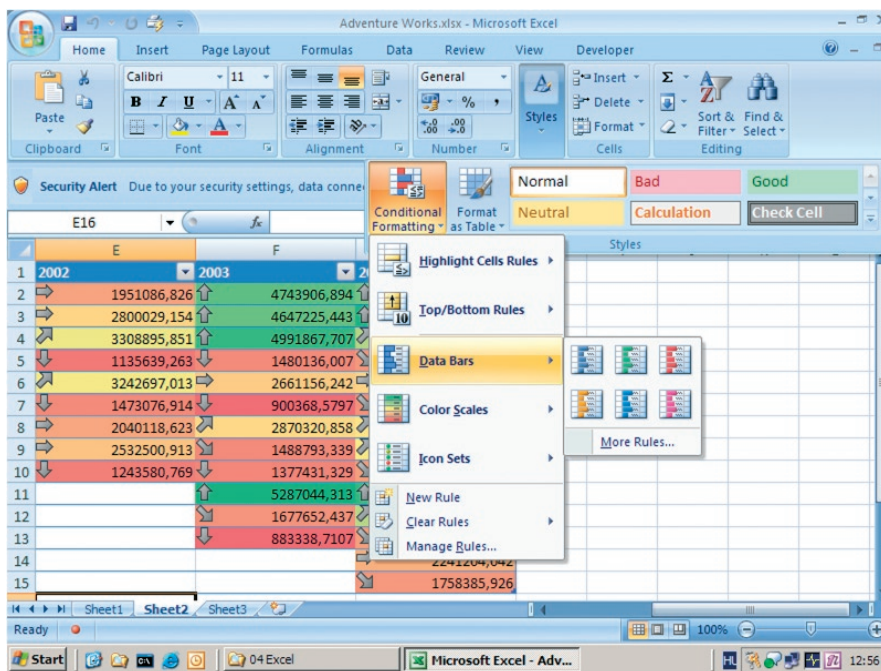
1047 Budapest, Baross utca 11-15.

Felelős vezető: Vértés Gábor ügyvezető igazgató

ISSN 1586-5185

Címlapon

A Microsoft termékfejlesztési filozófiájának fontos eleme az úgynevezett Integrated Innovation, ami magyarra lefordítva azt jelenti, hogy a cég szoftverei – miközben önmagukban is kitűnően használhatók – kifejezetten építenek egymás szolgáltatásaira, és együttes használatukkal a felhasználó többet kap, mint a részek összegét.



Mitől rendszer?

(König Tibor)

Webes technológiákat használó kiszolgálóalkalmazások és internetes szolgáltatások az írógéphez és telefonhoz szokott irodai munkatársaknak? Excel a böngészőben? Nyitott, XML fájlformátumok a Microsoft által birtokolt bináris formátumok helyett? Néhány éve még örült beszédnek tűnt mindez, de mára kiderült, hogy van benne rendszer. **6. oldal**

Új táblák az irodában

(König Tibor)

A 2007-es Office egyik szenzációszámba menő újítása a vadonatúj felhasználói felület (User Interface, UI). De vajon elegendő pluszt tartalmaz ez a felület ahhoz, hogy lemondhassunk az irodai alkalmazások sok-sok éves használatából származó tapasztalatokról? **9. oldal**

Amikor a régi találkozik az újjal

(König Tibor)

Milyen testre szabási lehetőségeket kínál a 2007-es Office, pontosabban az új felülettel ellátott alkalmazások? Egy példa: az extra szolgáltatások kitalálásától egészen a kibővített felületű alkalmazások teszteléséig. **12. oldal**

Biztonság

DEP és ASLR

(Kelemen László)

A Microsoft 2003-ban indította el Security Development Lifecycle (SDL) kezdeményezését. Az iniciatíva célja olyan szoftverek és megoldások fejlesztése, amelyek készítése során a kezdetektől a befejezésig fő szempont a biztonsági kockázatok csökkentése. **16. oldal**

Internet

Menedzsment WS-* szabványok segítségével

(Budai Péter)

A rendszergazdáknak mélyen ismerniük kell, hogyan kommunikálnak egymással azok a rendszerek, amelyeket fel kell építeniük, vagy üzemeltetniük kell. **20. oldal**

Infrastruktúra

Festők és bödönök

(Budai Péter)

Ha valaki új asztali számítógép, esetleg szerver vásárlásán gondolkodik, általában alaposan körülnéz, milyen lehetőségek vannak éppen a piacon. Fontos szempont, hogy az adott hardvert később mennyire lehet tovább bővíteni, illetve mennyire felel meg a futtatni tervezett szoftverek igényeinek. **24. oldal**

Kétszer hús érv II.

(Gál Tamás)

Folytatjuk az ISA Server 2004 Standard kulcsfontosságú jellemzőinek, előnyös vonásainak az előző lapszámban megkezdett ismertetését, kiegészítve a gyakorlatban felhalmozott tapasztalatokkal. **28. oldal**

Microsoft Virtual Server II.

(Lepénye Tamás)

A virtualizáció hatékony eszköz, de csak alapos tervezési munka után. Ezúttal abból adunk ízelítőt, milyen kérdéseket kell megválaszolnunk, amikor a hardvererőforrásokat tervezzük majdani rendszerünkhöz. **32. oldal**

Problémák a „Move Mailbox” után

(Somogyi Zoltán)

A jelenségre még nem érhető el hivatalos megoldás, de viszonylag egyszerűen elkerülhető, ha már eleve számítunk rá. **35. oldal**

Az Office 2003 bevezetése

(Moldova György)

Mielőtt elkezdenénk vadul telepítgetni, szánjunk egy kis időt a telepítési folyamat menetének és lehetőségeinek megismerésére! Az Office 2003 telepítésének folyamata szinte semmit sem változott az Office 2000-hez vagy az Office XP-hez képest. **36. oldal**

Alkalmazásplatform

Közelebbről az XML Paper Specification

(Garami Gábor)

Napjaink egyre fontosabb kérdése, hogy az elektronikus dokumentumok – és ezzel együtt a szellemi termékek: szerződések, jelentések, specifikációk és egyéb munkaanyagok jóváhagyott és hivatalos elektronikus példányai – milyen formátumban lássanak napvilágot. **38. oldal**

SQL Server 2005 Analysis Services

(Kovács Zoltán)

Az üzleti elemzést támogató legfontosabb funkciók működése – példákön keresztül. **41. oldal**

Folyjék a munka!

(Nagy Levente)

A .Net 3.0 egyik komponense a Windows Workflow Foundation (WF), ami a munkafolyamatok definiálását és kezelését teszi könnyűvé. Ebben a cikkben áttekintjük a WF felhasználásának egyik első területét, integrációját a 2007-es Office rendszerbe. **45. oldal**

Mentés és visszaállítás

(Pálós Máté)

A SharePoint-portálok és -webhelyek fő erőssége a rugalmasság: sokféle elemmel egészíthetők ki, képesek együttműködni más alkalmazásokkal, saját arculattal ruházhatók fel. **48. oldal**

Közösség

Figyelünk és tanulunk

(Budai Péter)

Alapvetően két gyakran ismétlődő megjegyzéssel találkoztunk: nem elég mély a tartalom, és drága a magazin. **50. oldal**

MITŐL RENDSZER?

Webes technológiákat használó kiszolgálóalkalmazások és internetes szolgáltatások az írógéphez és telefonhoz szokott irodai munkatársaknak? Excel a böngészőben?

Nyitott, XML fájlformátumok a Microsoft által birtokolt bináris formátumok helyett? Néhány éve még őrült beszédnek tűnt mindez, de mára kiderült, hogy van benne rendszer.

Úgy hívják: a 2007-es Microsoft Office rendszer.

A Microsoft termékfejlesztési filozófiájának fontos eleme az úgynevezett Integrated Innovation, ami magyarra lefordítva azt jelenti, hogy a cég szoftverei – miközben önmagukban is kitűnően használhatók – kifejezetten építenek egymás szolgáltatásaira, és együttes használatukkal a felhasználó többet kap, mint a részek összegét.

Egyenlőbb, mint a többi

Az integrált innovációt jól illusztrálja a Microsoft Exchange Server és a Microsoft Office Outlook kapcsolata. Az Exchange Server a vállalati levelezőrendszerek alapja, jól skálázható és menedzselhető e-mail-kiszolgáló.

Támogatja az elektronikus levelezésben használt ipari és bizottsági szabványok széles körét, így kiszolgálja (a teljesség igénye nélkül):

- a POP3/IMAP4 és SMTP internetes protokollokat használó levelezőprogramokat;
- a Microsoft MAPI protokolljának különböző változatait használó alkalmazásokat;
- az X.400-ra épülő klienseket;
- a hagyományos böngészőket (az Outlook Web Access révén);
- a mobiltelefonos böngészőket (az Outlook Mobile Access révén);
- az NNTP protokollt használó hírolvasó alkalmazásokat;
- más levelezőrendszerek, például a Lotus Notes klienseit (kapcsolószoftverek révén).

Az Outlook személyes információink, köztük elektronikus leveleink kezelésére való. E-mail-kliensként kapcsolódhat a következő szerverekhez:

- Exchange Server;
- internetes protokollokat használó levelező kiszolgálók;
- webes levelezőkiszolgálók (például az MSN Hotmail vagy a Live Mail);
- más kiszolgálók (kapcsolószoftverek révén).

Ha ez mind igaz, miért „egyenlőbb” az Outlook-Exchange-kettős, mint a többi lehetséges kombináció? Pontosan azért, mert – az integrált innováció révén – ez a páros egymás erősségeinek kihasználásával extra szolgáltatásokat is nyújt. Néhány ezek közül.

Automatikus beállítás. Mielőtt használatba vennénk egy levelezőprogramot, meg kell adnunk az e-mail-kiszolgáló címét, bejelentkezési adatainkat és más hasonló információkat. Az Outlook 2007-ben erre semmi szükség: automatikusan megkeresi az Active Directory címtárban az aktuális felhasználó postaládáját kezelő szerver adatait, elvégzi a beállításokat, és máris üzemkész.

Törölt elemek. Az Exchange Server számos szolgáltatással könnyíti meg az üzemeltetők munkáját. Ilyen például a törölt elemek megőrzése és a visszaállítás biztosítása a végfelhasználóknak. Ez a szolgáltatás csak az Outlook (és az Outlook Web Access) felületén keresztül érhető el.

Naptárkezelés. Az Outlook nemcsak levelezésre való, fontos képessége a naptárkezelés is. Ilyen szolgáltatással más alkalmazások is rendelkeznek, az Outlook azonban ismeri és ki is használja az Exchange Server foglaltsági (free/busy) adatbázisát, így könnyedén megjelenítheti a munkatársak naptárát. Arra is felhasználja az Exchange Servert, hogy a személyek mellett erőforrások (így például tárgyalók, vetítők, közös cégautók) naptárát is kezelhesse, és intelligens módon, automatikusan elfogadja vagy elutasítsa azok foglalását.

Információkezelő előírások. Az Exchange Server és az Outlook 2007-es verziója az információkezelési előírások (az úgynevezett compliance) megvalósításában is fontos szerepet játszik. Az Exchange-ben speciális információkezelő házirenddel szabályozott mappák definiálhatók. Az ezekbe kerülő levelek meghatározott idő után automatikusan archiválhatók, selejtezhethetők, megtekintésük, módosításuk naplózható. A mappák megjelenítése, kezelése az új Outlook feladata.

Nézzünk most egy valamivel összetettebb példát arra, hogyan valósul meg az integrált innováció az Office rendszerben!

Irodapélda

Egy vállalat munkatársai projekteknél dolgoznak: terveznek, feladatokat osztanak ki és hajtanak végre, dokumentumokat készítenek, adatokat gyűjtenek, összegeznek, publikálnak, és eközben persze gyakran kommunikálnak egymással és külsős személyekkel, például üzleti partnerekkel.

A fent leírtakhoz ma általában a következő informatikai megoldást veszik igénybe:

- A dokumentumok egy fájlkiszolgáló megosztott mappájában találhatóak. (Vessük el azt a változatot, amelyikben minden a projektvezető gépén lakik.) A csapat egy része ugyanabban a szervezeti egységben dolgozik, ők hozzáférnek a mappához. A más részlegekhez tartozók közül csak azok, akik jogot kértek rá, és akiknek a projektvezető nem felejtette el megadni azt. A többiek e-mailben kapják és küldik a különböző dokumentumváltozatokat.
- A szöveges információt általában Word-dokumentumokban rögzítik, a változáskövetést hol bekapcsolják, hol nem. A számszerű információkat és listákat Excelben tárolják.
- A dokumentumok konkurens szerkesztését úgy oldják meg, hogy a munkatársak telefonon megkérlik a másikat, hívja vissza őket, ha elengedte az éppen szerkesztett fájlt. A verziókövetés a fájlok átnevezésével történik, gyakran meglehetősen komoly elnevezési szabályok alapján.
- Az űrlapok Excelben és/vagy Wordben készülnek, az ezekkel rögzített adatok összesítése a legtöbbször manuális munkát igényel.
- A projekt céljáról, tagjairól, előrehaladásáról stb. e-mailben értesítik az érdekelteket – már amikor nem felejtik el.
- A kommunikáció kizárólagos eszközei: az e-mail és a telefon.

Az Office rendszer elemei, nevezetesen az ügyféloldali alkalmazások fellelhetők a fenti listában. Néhány további összetevő hozzáadásával áramvonalasabbá tehető a képzelt vállalat működése. Lássuk, hogyan!

Fájlserver++

A fájlservereket azért szeretik a felhasználók, mert könnyen elérhető a fájlrendszer-

ből és a különböző alkalmazásokból, meglehetősen gyorsak, és az IT menti a tartalmukat. Ennél többet azonban nem is igazán tudnak.

Ha a vállalat a Windows Server részét képező Windows SharePoint Services (WSS) webes alkalmazást használja, először is megkapja a fájlkiszolgáló előnyeit:

- a dokumentumtárak az URL-cím mellett UNC-névként, azaz \\kiszolgáló\megosztás\mappa formátumban is megcímezhető, így a Windows Intézőből, a különböző Commanderekből és a parancssorból egyaránt elérhető a tartalmuk;
 - a dokumentumelérés sebessége megközelíti a fájlkiszolgálóké;
 - az IT a WSS-webhelyeket is menti, egy-egy elem visszaállítását azonban a felhasználó is el tudja végezni.
- Aztán jönnek az extrák:
- a webhely a projekt portálja: feltüntethető rajta a cél, a résztvevők listája és szerepe, a fontosabb mérőszámok állása;
 - az egyes dokumentumokhoz tetszőleges leíróadatok (például a projekt azonosítója, a dokumentum típusa, készültése, felelőse) rendelhetők;
 - automatikus verziókövetés állítható be, fő- és alverziókkal – vége a manuális átnevezésnek;
 - a felhasználók előfizethetnek a dokumentumok megváltozásáról szóló értesítésekre – nem kell többé a telefonra várni;
 - a dokumentumok mellett listák is tárolhatók a SharePointban, szükségtevé az Excel alkalmazását; minden soruk külön-külön szerkeszthető, támogatják a verziókövetést, elemezhetők helyben, vagy exportálhatók Excelbe/Accessbe;
 - a csoportmunka-webhelyen nemcsak tárolhatjuk a dokumentumokat, hanem kioszthatjuk és követhetjük a hozzájuk kapcsolódó feladatokat is.

Ha a vállalat a WSS helyett a Microsoft Office SharePoint Server bevezetése mellett dönt, további előnyökhöz jut:

- a leíró információk külső társadattárak-

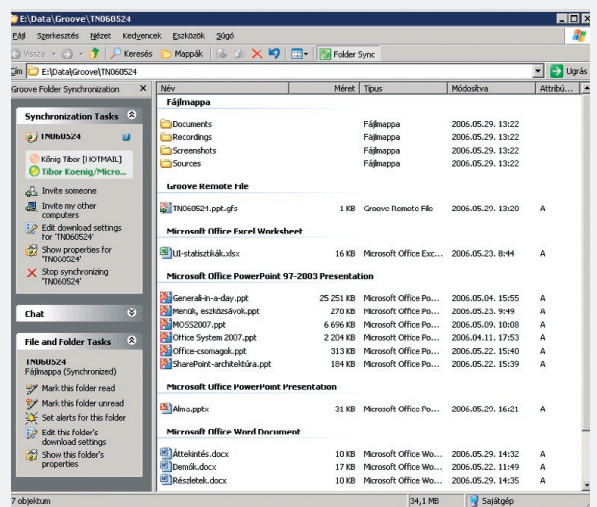
ből is származhatnak, a dokumentumok mellé odatehető például az ügyféltörzsben tárolt cégnev;

- beépített dokumentum-munkafolyamatok (véleményezés, jóváhagyás) vehetők igénybe, de ezen túlmenően saját munkafolyamatok is tervezhetők;
- a dokumentumokon végzett műveletek (olvasás, módosítás, törlés stb.) auditálhatók. És mindez egy rendkívül barátságos, az Office alkalmazásaiba is beépülő felületen. Kapcsoljuk ki a fájlservert!

Szinkronban

Az e-mail a fájlkiszolgálóhoz hasonlóan kedvelt megoldás, a címzett kiválasztásához, a levél megírásához és a mellékletek csatolásához az átlagfelhasználó is ért. A dokumentumok ide-oda küldözgetése legtöbbször azért szükséges, mert a külsős, illetve az eredeti tárolási helyet el nem érő felhasználók csak így kaphatják meg azokat. Ez a gyakorlat azonban problémákat is felvet, a különböző verziójú dokumentumokon végzett módosítások összefésülésétől kezdve a levelező-rendszerek fölösleges terheléséig.

Ha a cég az Office rendszer egy új komponensét, a Microsoft Office Groove alkalmazást használja, nem kell a dokumentumokat e-mailben továbbítani. A projektvezető elő-



Mappa megosztása a Groove-val

szőr létrehoz egy úgynevezett Groove-munkaterületet, ahová összegyűjti a projektben használatos dokumentumokat és egyéb állományokat, adatokat, információkat. A meglévő mapparendszer vagy SharePoint-webhely néhány kattintással munkaterületté alakítható.

Ezután felsorolja azokat a belső és külső munkatársakat, akik a projektben dolgoznak. Ezzel meghívja őket a munkaterületére. A meghívás e-mailben érkezik.

Ha a címzett gépén fut a Groove, csak rákattint a munkaterületet jelképező mellékletre. Ha nincs telepítve nála a szoftver, egy hivatkozás a Groove webhelyére kalauzolja, ahonnan letölthet egy harminc napig működő próbaverziót.

A meghívást elfogadott munkatársak gépén létrejön az eredeti munkaterület másolata. Dolgozhatnak a meglévő dokumentumok

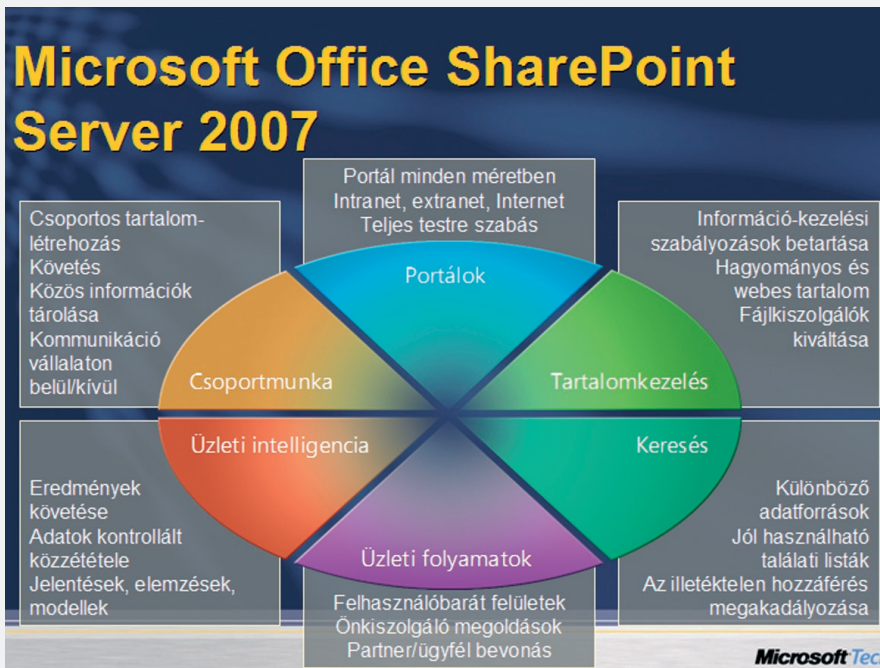
javít a dolgon, mert egyszerűbb benne az összegzés, de ha több Excel-állományt szeretnénk egyesíteni, manuális munkára lesz szükségünk.

Az Office rendszer űrlaptervező és -futató alkalmazása, a Microsoft Office InfoPath segít a dolgon:

- az összegyűjtött adatokat XML formátumban tárolja, s az XML tetszőleges sémára illeszkedhet;
- az adatokat mentheti fájlként (például egy fájlkiszolgálóra), közvetlenül adatbázisba, SharePoint-dokumentumtárba, vagy átad-

Office Live Communications Server többféleképpen is bővíti a kommunikációs lehetőségeket:

- A munkatársak jelenléte folyamatosan követhető. Ahol csak az Office rendszerben nevet látunk – például az e-mailek feladójánál, a dokumentumok szerzőjénél vagy utolsó módosítójánál –, azt is figyelemmel kísérhetjük, hogy az illető elérhető, megbeszélésen ül, vagy nincs a gépénél.
- Az elérhető projektagok azonnali szöveges, telefonos vagy audio- és videóüzeneteket válthatnak egymással.
- A munkatársak az alkalmazásmegosztás segítségével közösen, interaktívan dolgozhatnak egy dokumentumon. Az sem baj, ha a szükséges alkalmazás nincs minden résztvevő gépén telepítve.
- A vállalaton belüli rendszer összekapcsolható az internetes szolgáltatásokkal (MSN, Yahoo, AOL).



A SharePoint Server felhasználási területei

kon, újakat hozhatnak létre, és közben kommunikálhatnak is egymással. Minden változás automatikusan megjelenik valamennyi tag gépén, az ütközések automatikusan vagy manuálisan feloldhatók.

Mindez egy webszolgáltatás, a Microsoft Office Groove Enterprise Services segítségével történik, de nagyobb vállalatok választják saját Groove-kiszolgáló üzemeltetését is. A kommunikáció a 80-as portot használja, a tűzfalakon tehát nem kell lyukat fúrni.

Űrlapok

Az adatok rögzítésére szolgáló űrlapok azért készülnek Wordben, mert eredetileg kinyomtatásra tervezték őket. A velük összegyűjtött adatokat azonban komolyabb fejlesztés nélkül legfeljebb újragépelni lehet. Az Excel

hatja egy alkalmazásnak webszolgáltatáson keresztül, így egyszerű azok feldolgozása, összesítése, kimutatások, jelentések készítése;

- képes a Word és Excel alapú űrlapok importálásával új űrlapok létrehozására;
- a kitöltendő sablont közzé lehet tenni fájl és webszerveren, elküldhetjük e-mailben, de integrálható a Groove-munkaterületekbe is.

Kapcsolat

A telefon és az e-mail egyeduralgónak számít, a fax – legalábbis az informális projekteknél – kevésbé fontos. De a telefonhoz elérhetőnek kell lenni, az e-mailnél pedig sose tudhatjuk, mikor kapunk választ. Az Office rendszer egy újabb eleme, a Microsoft

Van benne rendszer

Az előző példából jól látszik az integrált innováció többszintű működése.

Már az is rengeteget segít, ha csupán az Office rendszer egy-egy összetevőjét alkalmazzuk. A SharePoint ötvözi a fájlkiszolgálótól elvárt szolgáltatásokat egy korszerű csoportmunka- és dokumentumkezelő alkalmazás képességeivel. A Groove feleslegessé teszi az e-mailben küldött dokumentumok időigényes összefűzését.

Az InfoPath strukturált adatgyűjtést biztosít, újragépelés és manuális egyesítés nélkül. A Live Communications Server a munkatársak elérhetőségének követésével hatékonyabba teszi a kapcsolatteremtést.

Elérhető munkatársak

Ha a komponenseket együtt használjuk, további előnyökhöz jutunk. A SharePointban tárolt dokumentumok a Groove segítségével olyanokhoz is eljuttathatók, akik nem férnek hozzá a vállalat intranetjéhez. A Live Communications Server pedig az Exchange Server free/busy adatbázisa alapján automatikusan is be tudja állítani a munkatársak elérhetőségét. Érdemes próbát tenni a fentiekkel! Az új Microsoft Office rendszer béta-változata letölthető a <http://www.microsoft.com/office/preview> címről.

König Tibor

(tibork@microsoft.com) Microsoft Magyarország

ÚJ TÁBLÁK AZ IRODÁBAN

A 2007-es Office egyik szenzációszámba menő újítása a vadonatúj felhasználói felület (User Interface, UI). De vajon elegendő pluszt tartalmaz ez a felület ahhoz, hogy lemondhassunk az irodai alkalmazások sok-sok éves használatából származó tapasztalatokról?

A következőkben röviden összefoglaljuk az új UI létrejöttének okait, a tervezési célokat, a fontosabb elemeket, és – kódrészletekkel illusztrálva – kitérünk arra is, hogyan bővíthető a felület meglévő és új elemekkel.

Könyvelés: első emelet, balra

A felhasználói felület olyan, mint egy hivatal recepcióján a tájékoztató tábla. Az a feladata, hogy elérhetővé tegye a szolgáltatásokat. Ha egy alkalmazás (vagy irodaház) kevés funkciót lát el, az útbaigazítás is egyszerű. A Microsoft Word for Windows 1.0 viszonylag egyszerű szövegszerkesztő volt, szolgáltatásait könnyedén lefedte körülbelül ötven menüparancs és két eszközsáv. A Microsoft Office Word 2003 funkcióinak száma sokszorosa az 1.0-ban lévőknek, ezek megjelenítéséhez azonban nagyon hasonló megoldásokat – menüket, eszközsávokat, munkablakokat – használtak a fejlesztők az első verzióban.

A hagyományos felület azonban már nem volt képes a szolgáltatások hatékony, rendezett, jól használható megjelenítésére. A menüparancsok száma meghaladta a 270-et, és ez szükségessé tette a többszörösen lépcsőzetes – tehát beláthatatlan – menük kialakítását. Több mint harminc eszközsávból választhatott a felhasználó – ekkora készletnél nehéz eldönteni, mit jelenítsen meg, és mit ne. A munkablakok száma megközelítette a húszat – mindegyik önmagában is egy-egy újabb mini-alkalmazás.

Az irodaépület-analógiához visszatérve: gyakorlatilag lehetetlen egy igazán komplex szervezet elérhetőségét hagyományos tájékoztató táblákkal megadni. Aki látott már sokemeletes irodaház sokemeletes eligazító tábláját hosszú percekig böngésző, majd lemondóan az információs pulthoz ballagó állampolgárt (ügyfelet, beteget stb.), érti, miről is van szó. Az információ ott van, csak éppen az elérése nehéz.

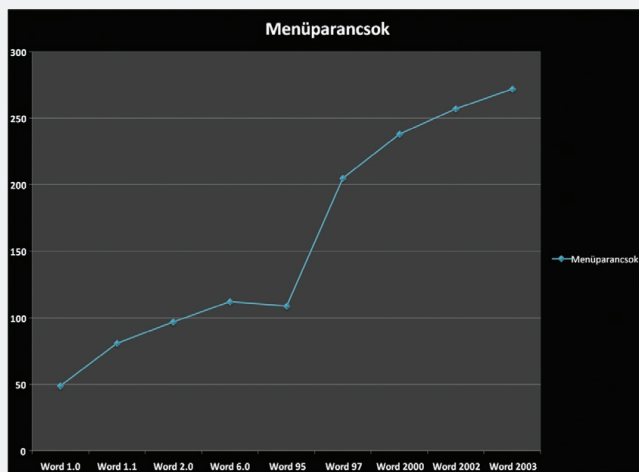
Ide kívánczik egy Douglas Adams-idézet a „Galaxis Útikalauz stoposoknak” című könyvből:

- De hiszen a tervek ki voltak állítva.
- Ki voltak állítva? Éppenséggel a pincébe kellett lemennem, hogy megtaláljam őket.
- Ott van a kiállítási részleg.
- Zseblámpával.
- Talán leszakadt a villanyvezeték.

– Meg a lépcső.
– Nézze, végül is megtalálta a feljegyzést, nem?

– Ó, hogyme – felelte Arthur –, megtaláltam. Egy használaton kívüli vécében elsuvasztott, bezárt inatszekrény fenekén volt kiállítva, az ajtón a következő felirattal: Vigyázz! A leopárd harap!

A fentiekkel (nem a leopárral) a Microsoft is szembesült. A felhasználók egyfelől azt hangoztatják, hogy az Office aktuális verziójának legfeljebb 10–15 százalékát használják ki, és minden, amire valaha is szükségük lehet, már az egyvel korábbi Office-ban is



A menüparancsok száma az égbre megy

megtalálható volt. Másfelől panaszkodnak, hogy azért akadna egy-két új funkció, ami-

vel könnyebben elvégezhetnék feladataikat. De vajon új képességekről van itt szó? Ennek sokszor épp az ellenkezője igaz. A Microsoft számos termékfejlesztési és -korszerűsítési javaslatot kap, és ezek olyan szolgáltatásokat reklámálnak, amelyek már benne vannak a termékben!

Tervezési célok

Az új felület megalkotásának céljai – ahogy azt *Jensen Harris*, a megoldás egyik „atyja” érzékletesen leírja kitűnő blogjában (<http://blogs.msdn.com/jensenh>) – a következők voltak:

- a már ismert szolgáltatások hatékonyabb felhasználásának elősegítése;
- a még nem használt (esetleg nem is ismert) szolgáltatások felfedezhetővé tétele;
- professzionális dokumentumok előállításának megkönnyítése olyan felhasználók számára (is), akik nem dizájnerek, és/vagy sem indítottak, sem igényük arra, hogy az oldal elrendezésének, a betűcsomagok kiválasztásának és a színek összehangolásának valamennyi lehetőségét kipróbálják, ugyanakkor végül pontosan azt kapják, amit akarnak.

Ami változott – a szalag

Az új Office – pontosabban az az öt (négy és fél) alkalmazás, amelyekbe az új felület bekevert – szakít a régi felület számos megoldásával. Eltűnik a hagyományos értelemben vett menük és eszközsávok nagy része, és egy új elem helyettesíti őket: a szalag (Ribbon).

A szalag azokat a funkciókat teszi elérhetővé, amelyek a dokumentumokban végzett munkát segítik. (A dokumentumokon, illetve az alkalmazáson végrehajtott műveletek az új Office-menübe kerültek.) A szalag az alkalmazás ablakának felső részét foglalja el hozzávetőleg két és fél eszközsávnyi helyen.

A korábbi főmenüket úgynevezett fülek (tabs) váltották fel, ezek a főbb munkafázisokat jelképezik. Az egyes fülekhez saját szalag



Professzionális formázási opciók széles választéka: galériák

tartozik. A szalagon csoportokba rendezetten vezérlők találhatók – ez olyan nagyon még nem különbözik a mai eszközsávoktól. Viszont egy adott szalagon minden pillanatban minden lényeges vezérlő látszik! Nem kell többszintes menühierarchiakon átverekednie magát a felhasználónak, hogy meglegye a kívánt parancsot.

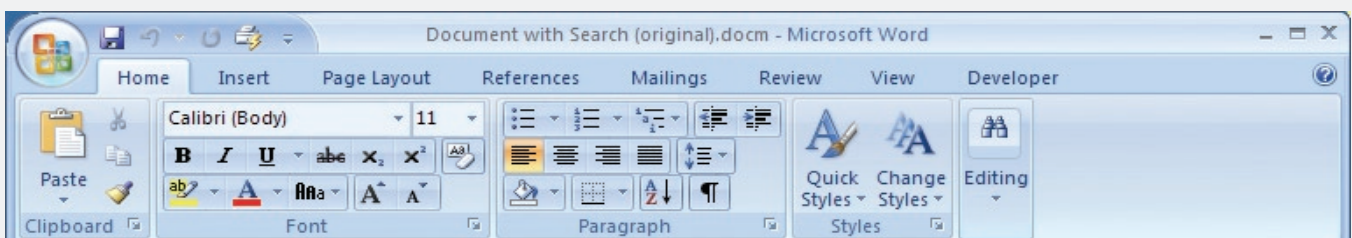
A szalagok skálázhatók, a vezérlőknek van nagy, normál és kisméretű változatuk, az ablakot összenyomva tehát továbbra is látható marad minden. Az egyes fülekre kétszer kattintva egyébként eltüntethetők (és vissza is hozhatók) a szalagok, a profik így helyet takaríthatnak meg.

Egyes szalagok csak adott kontextusban

jelennek meg. Ha a dokumentum táblázatot tartalmaz, és belekattintok, a táblázatformázáshoz szükséges fülek és szalagok válnak elérhetővé. Ha a táblázat helyett egy képet jelölök ki, képszerkesztő szalagot kapok.

A gombok mellett rengeteg egyéb vezérlő is megtalálható a szalagokon, a párbeszédpaneleken szokásos szövegmezőktől, listáktól, jelölőnégyzetektől kezdve egészen az úgynevezett galériáig. Ezek előre elkészített dokumentumrészletek, professzionális formázási opciók széles választékát kínálják – és meg is mutatják azokat, hogy egyszerűen választani lehessen közülük.

A galériák egy további szolgáltatást is kínálnak: ha az egeret egy formázási lehetőség



Egy új elem, a szalag helyettesíti a hagyományos menüket és eszközsávot

főle visszük, az Office-alkalmazás röptében átalakítja az éppen megnyitott dokumentum látható részét, azaz megmutatja, milyen lenne a „munkadarab”, ha az adott opciót választanánk. Nem tetszik, amit látunk? Csak odébb kell húzni az egeret! A módosítás nem következik be addig, amíg ténylegesen rá nem kattintunk egy formázási lehetőségre. Ez az élő előnézet (Live Preview) rendkívüli módon felgyorsítja a munkát.

Ami változott – az Office-menü

Ha az éppen megnyitott dokumentumban már befejeztük a munkát, valamit kezdeni akarunk a dokumentummal – az ehhez szükséges parancsok (a korábbi verziók Fájls és Eszközök menüjének nagy része) az új Office-menüben található.

Ez a menü a bal felső sarokban található kerek Office-logóra kattintva érhető el. Itt a következőket lehet elvégezni:

- új dokumentum létrehozása, meglévő dokumentum megnyitása (a korábban megnyitott dokumentumok közül a gyakran használtak egy rajszóggal rögzíthetők, így nem „kopnak ki” a listáról akkor sem, ha régóta nem nyúltunk hozzájuk);
- dokumentumok mentése, korábbi Office-verzióban vagy más formátumban (például XPS) is;
- nyomtatás;
- véglegesítés (dokumentumlajdonságok beállítása, dokumentum vizsgálata a nem kívánt tartalom – például megjegyzések, személyes információk – kiszűrésére, hozzáférés korlátozása

- munkafolyamat-kezelés (munkafolyamat indítása, állapotának megtekintése);
- az alkalmazás beállításai.

Ami változott – a keret

A dokumentumok megtekintésével, állapotával, fontosabb adatainak megjelenítésével kapcsolatos funkciók (azaz a Nézet és Ablak menük nagy része, illetve az állapotsáv) az új alkalmazáskeretbe kerültek. Ez az egységes keret teszi lehetővé

- a váltást a különböző nézetek között;
- a nagyítást/kicsinyítést egy jól használható csúszka segítségével;

maznak párbeszédpaneleket, ezek használata azonban csak a galériák bőséges lehetőségeinek kimerítése után szükséges, azaz az átlagfelhasználó várhatóan szinte sohasem találkozik velük.

A jobb egérgombbal ezután is előhívhatók tartalomérzékeny menük.

Egy speciális „lebegő” menü minden alkalommal megjelenik, amikor a felhasználó egerrel kijelöl egy szövegrészletet. Ez a félig átlátszó menü az eger föléhúzásakor válik teljesen láthatóvá, és a leggyakoribb formázási opciókat tartalmazza, így csökkenthető az egerrel bejárt út hossza.



A Fájls és az Eszközök menü az Office-menübe került

- a fontos információk (a Wordben az oldalak, bekezdések, szavak, karakterek, az Excelben a kijelölt cellák összege, átlaga,

Ez most komoly?

Mint említettük, nem az összes Office-alkalmazás kapott új ruhát. A szerencsések közé tartozik a Word, az Excel, a PowerPoint és az Outlook – pontosabban annak elemszerkesztő ablakai, az úgynevezett Inspectorok. A fentiek az Office fő tartalom-előállító alkalmazásai, radikális átforgatásuk tehát üzenet. Új felület jutott tovább az Accessnek, rendkívül kényelmessé téve az új generációs adatbázis-alkalmazások elkészítését. Mi több, az Office webes kiépítéseként funkcionáló, folyamatosan frissülő Office Online külseje is az új dizájnhoz illeszkedik.

Maradt a régi felületnél az Office-beltagok közül az InfoPath, a OneNote és a Publisher, illetve a kültagok (a csomagba

nem tartozó programok) mindegyike: a Project, a SharePoint Designer (az egykori FrontPage), a Visio és a Groove. Gyakran kérdezik,

A Nézet és Ablak menük, illetve az állapotsáv helyett: keret

- sa, digitális aláírás, a kompatibilitás ellenőrzése);
- küldés e-mailben, faxon;
- közzététel blogbejegyzésként, dokumentumkezelő kiszolgálóra vagy dokumentum-munkaterületre;
- kiszolgálóoldali feladatok (beadás/kivétel, korábbi verziók megtekintése);

számossága stb.) valós időben frissülő megjelenítését.

Ami nem változott

Az új felület nem teljes szakítás a múlttal, inkább megszüntetve megőrzés. Lássuk, mi maradt!

Az Office-alkalmazások továbbra is tartal-

van-e titkos billentyűkombináció, registry-beállítás, amittől az új felület visszaváltozik a régivé. Tudomásunk szerint nincs, a fejlesztők kihúzták a hajókat a partra, és elégettek őket, hogy ne lehessen visszatérni a régi módhoz.

Kőnig Tibor

(tibork@microsoft.com) Microsoft Magyarország

AMIKOR A RÉGI TALÁLKOZIK AZ ÚJJAL

Milyen testre szabási lehetőségeket kínál a 2007-es Office, pontosabban az új felülettel ellátott alkalmazások? Egy példán keresztül az extra szolgáltatások kitalálásától egészen a kibővített felületű alkalmazások teszteléséig.

Az Office menü gombjától jobbra egy rövidke eszközsáv, az úgynevezett Quick Access Toolbar található. Alapértelmezésben négy gombot tartalmaz, ezek: a Mentés, a Visszavonás, az Ismétlés és a Nyomtatás (a szokásos beállításokkal).

Ez az egyedüli eszközsáv, amely a régiekhez hasonlóan bővíthető. Bármelyik szalag bármelyik csoportja és vezérlője egy jobb- és egy balkattintással hozzáadható ehhez az eszközsávhoz, és eltávolítható onnan.

A Parancsok Kútja

A gyorsan elérhető eszközsáv részletes testre szabását lehetővé tevő egyszerű felépítésű, de az Office-alkalmazások teljes funkciógazdagságát megmutató párbeszédpanel – a fejlesztők szóhasználatában a Parancsok Kútja (Command Well) – minden parancsot és makrókat elérhetővé tesz, függetlenül attól, hogy az része-e valamelyik szalagnak vagy sem.

Gombot a szalagra

A legizgalmasabb természetesen az új felület kihasználása, azaz saját szolgáltatások elérhetővé tétele az új felület szalagjain. Az Office 2007-es verziója a korábbiakhoz hasonlóan támogatja az úgynevezett beépülő elemek (add-ins) használatát, ezeket kezelt és nem kezelt kód írásával, a Visual Studio Tools for Office segítségével lehet előállítani. Az így létrehozott és az adott rendszerben regisztrált beépülő elemek automatikusan megjelennek a Beépülő elemek (Add-ins) fülön, illetve a hozzá tartozó szalagon. De hogyan hozhatók létre saját szalagok az egyéni funkcionalitás megjelenítéséhez? Alapvetően kétféle módon:

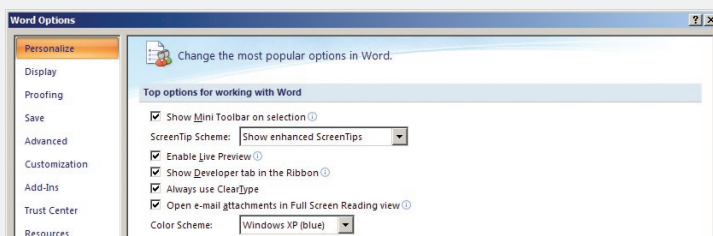
- olyan beépülő elem készítésével, amely a hagyományos Extensibility.IDTExtensibility2 felület mellett a Microsoft Office.Core névtérben található új IRibbonExtensibility felületet is implementálja;
- az XML-ben definiált felhasználófelület-bővítés hozzáadásával egy dokumentumhoz vagy sablonhoz, az új Open XML fájlformátum kihasználásával.

Egy használható példa

A fenti megoldások közül az első szétfeszítené a cikk kereteit, így csupán a második bemutatására szorítokozunk. Kibővítve az egyik Office-alkalmazás, a Word 2007 szolgáltatásait, a hozzáadott képességeket pedig egy új szalagon jelenítve meg.

Az első lépés az extra szolgáltatások kitalálása. Nem különösebben lelkesítőek azok a mintaprogramok, amelyek megfelelően illeszkednek ugyan az adott szakmai cikk témáját, de egyébként haszontalanok. Jellemző példa a gomb, amely megnyomása után kiírja: „Rám kattintottál!” Izgalmas, ugye? Nos, a gomb a következő példában is stimmel, megnyomása azonban egy webes keresést indít el, ahol a keresett szöveg az aktuális Word-dokumentumban éppen kijelölt szó vagy kifejezés. Igazából két gombot használunk, az MSN, illetve a Google keresőmotorjához.

A második lépés a szolgáltatások megvalósítása. Miután a Visual Basic for Applications (VBA) a 2007-es Office-ban is támogatott, a legegyszerűbb, ha ezt használjuk. Elindítjuk a Word 2007-et, majd bekapcsoljuk a Developer fület és a hozzá tartozó szalagot. Ezt az Office menü alján, a Word



Developer fül: a szolgáltatások megvalósításához be kell kapcsolni

Options gombra kattintva, majd a Show Developer tab in the Ribbon beikszelésével lehet megtenni.

A Developer fülön elindítjuk a Visual Basic Editort, hivatkozunk a Microsoft Internet Controlsra, majd kétszer kattintunk a ThisDocument objektumra, és beírjuk az 1. ábrán látható sorokat.

A fentiek nem igényelnek sok magyarázatot: először létrehozunk az Internet Explorer egy példányát, aztán megnyitjuk az MSN, illetve a Google keresőlapját úgy, hogy a megfelelő URL-t kiegészítjük a Wordben éppen kijelölt szöveggel, végül megjelenítjük a böngészőt és benne a találati listát.

A hivatásos programozók számos finomítási lehetőséget találhatnak a szándékosan leegyszerűsített kódban (objektum létrehozásának módja, hibakezelés hiánya) – ezeket ki-ki maga készítheti el. Házi feladat továbbá a Windows Live Search web-szolgáltatásának igénybe vétele (egy rövid összefoglaló található ezen a címen: <http://msdn.microsoft.com/live/gettingstarted/searchstart/default.aspx>).

Most már bezárhatjuk a Visual Basic Editort, és menthetjük – mondjuk az Asztalra – az új szolgáltatásokkal kibővített dokumentumot. Az alapértelmezett .docx kiterjesztés helyett a .docm-et használjuk, ahol az m betű arra utal, hogy a fájl makrókat is tartalmaz. Legyen a fájl neve Keresés.docm.

A harmadik lépés a szolgáltatások kivezetése a felhasználói felületre. Egy új, Search nevű fület – és a hozzá tartozó szalagot – jelenítünk majd meg; a szalagon egy Keresés a weben nevű csoport lesz, abban pedig két gomb, Keresés (MSN), illetve Keresés (Google) felirattal. A gombok részletes információkkal szolgálnak magukról.

Az új felhasználói felület-elemeket – minő meglepetés – egy XML fájl írja le. Kedvenc XML-szerkesztőnkkel, például a Visual Studio 2005-tel létrehozunk egy üres XML fájlt, és bemásoljuk a 2. ábra tartalmát.

```
Sub SearchWithMSN()
    Dim oIE As New InternetExplorer
    oIE.Navigate „http://search.msn.com/results.aspx?q=” & Selection.Text
    oIE.Visible = True
End Sub

Sub SearchWithGoogle()
    Dim oIE As New InternetExplorer
    oIE.Navigate „http://www.google.co.hu/search?q=” & Selection.Text
    oIE.Visible = True
End Sub
```

1. ábra. Szolgáltatások

```
<customUI xmlns=”http://schemas.microsoft.com/office/2006/01/customui” >
<ribbon startFromScratch=”false” >
  <tabs>
    <tab id=”tabCustomSearch” label=”Search”>
      <group id=”grpCustomSearch” label=”Keresés a weben” >
        <button id=”btnSearchWithMSN”
          label=”Keresés (MSN)”
          supertip=”A kijelölt szöveget keresi a weben az MSN
            Search segítségével.”
          size=”normal”
          onAction=”ThisDocument.SearchWithMSN” />
        <button id=”btnSearchWithGoogle”
          label=”Keresés (Google)”
          supertip=”A kijelölt szöveget keresi a weben a
            Google Search segítségével.”
          size=”normal”
          onAction=”ThisDocument.SearchWithGoogle” />
      </group >
    </tab>
  </tabs>
</ribbon>
</customUI>
```

2. ábra. Elemek

```
<?xml version=”1.0” encoding=”UTF-8” standalone=”yes”?>
<Relationships xmlns=
  ”http://schemas.openxmlformats.org/package/2006/relationships”>
  <Relationship Id=”rId3”
    Type=”http://schemas.openxmlformats.org/officeDocument/
      2006/relationships/extended-properties”
    Target=”docProps/app.xml”/>
  <Relationship Id=”rId2”
    Type=”http://schemas.openxmlformats.org/package/
      2006/relationships/metadata/core-properties”
    Target=”docProps/core.xml”/>
  <Relationship Id=”rId1”
    Type=”http://schemas.openxmlformats.org/officeDocument/
      2006/relationships/officeDocument”
    Target=”word/document.xml”/>
</Relationships>
```

3. ábra. Kapcsolatok

A szerkezet első látásra is érthető: ki-bővítjük a meglévő füleket/szalagokat (startFromScratch=”false”) úgy, hogy létrehozunk egy új fület, benne egy csoportot, abban pedig két gombot. A gombok tulajdonságai a következők:

- id: a gomb azonosítója;
- label: a gomb címkeje;
- supertip: az egeret a gomb fölé húzva megjelenő tájékoztató szöveg;
- size: a gomb mérete;
- onAction: a gomb megnyomásakor végrehajtandó eljárás.

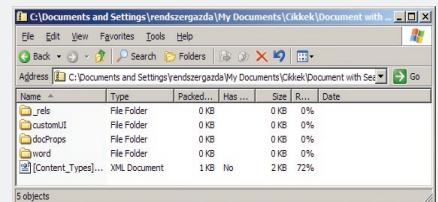
A gombok mellett természetesen további vezérlők is rendelkezésre állnak, az egyes vezérlők tulajdonságai sem korlátozódnak a fenti ötre.

A bővítést leíró XML-fájl ezzel elkészült: tetszés szerinti helyen, például az Asztalon létrehozunk egy customUI nevű mappát, és abba mentjük customUI.xml néven.

Most felvesszük a fenti fájlt a Word-dokumentumot alkotó többi állomány közé, és megadjuk azt is, hogy mire való.

Ehhez először is alkotóelemeire bontjuk a .docm kiterjesztésű dokumentumot. Ez egyszerű feladat: a 2007-es Office-ban használt Open XML fájlformátum valójában mappákba rendezett XML-fájlok együttese, ZIP technológiával összetömörítve.

Átnevezzük a Keresés.docm dokumentumot Keresés.zip-re. A Windows figyelmeztet a kiterjesztés megváltoztatásának veszélyeire, de hősiesen vállaljuk a kockázatot. Az átnevezett fájl megnyitáskor a következőket látjuk a Tallózóban:



A ZIP tartalma

Egy egérmozdulattal bemásoljuk a korábban létrehozott customUI mappát (és benne a customUI.xml fájlt) a megnyitott ZIP-be. Ezzel hozzáadtuk az új felhasználói felület leírását a dokumentumhoz, de még nem áruktuk el, mire is való.

A ZIP-ben lévő mappák egyike a _rels névre hallgat. Az egérrel átmásoljuk az Asztalra, és ott megnézzük a tartalmát. Egy .rels nevű

fájl találunk benne; trükkös kiterjesztése ellenére ez is XML. Megnyitjuk a kedvenc XML-szerkesztőnkkel, és a 3. ábrának megfelelő sorokat látjuk (az olvashatóság érdekében beillesztett sortörésekkel).

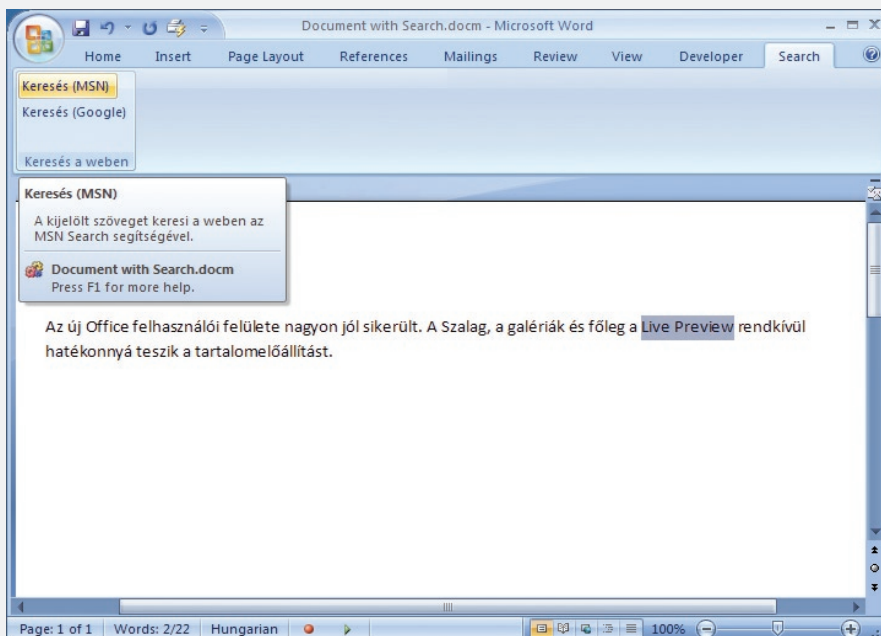
A .rels az Open XML formátum egyik legfontosabb eleme: megadja az egyes fájlok szerepét, azaz kapcsolatát a dokumentumhoz. Az egyes kapcsolatok három fő tulajdonsága:

- Id: egyedi azonosító;
 - Type: a fájl típusa, itt például maga a dokumentum, a beépített, valamint a kibővített dokumentumtulajdonságok (officeDocument, core-properties, extended-properties);
 - Target: a fájl relatív címe a ZIP-en belül.
- A felhasználói felület-bővítést leíró fájl az alábbi sorral jegyezzük be a többi elem közé:

```
<Relationship Id="rId4"
  Type="http://schemas.microsoft.com/office/2006/relationships/ui/extensibility"
  Target="customUI/customUI.xml"/>
```

A típusa tehát ui/extensibility, a címe pedig az előbb bemásolt customUI mappa, illetve azon belül a customUI.xml fájl.

Mentjük és bezárjuk a .rels fájlt, és a _rels mappával együtt visszamásoljuk az asztalról a ZIP-be. Bezárjuk a ZIP-et megjelenítő Tallózó ablakot (mentésre nincs szükség),

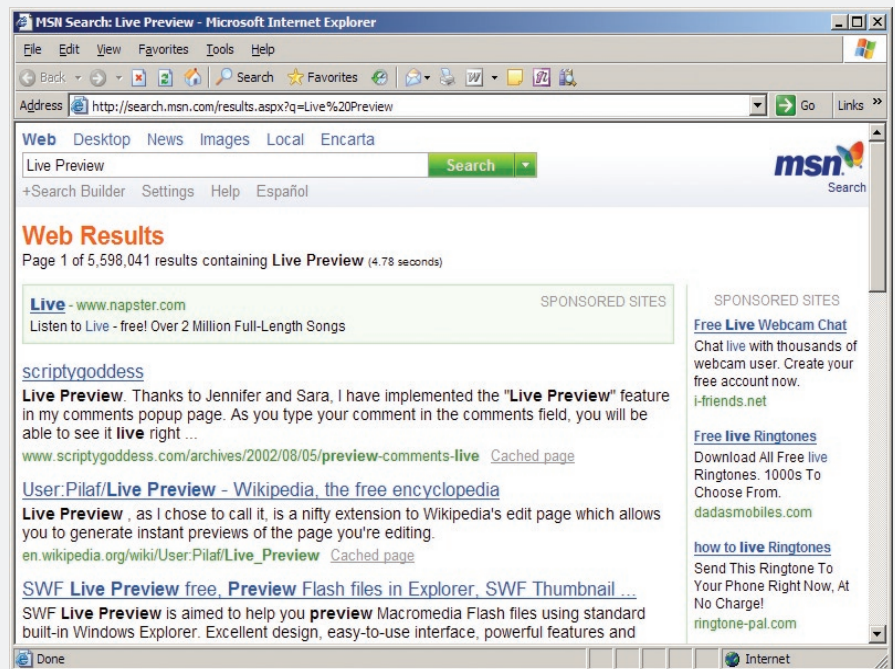


Keresés az egyszerű, áttekinthető felületen

visszanevezzük a Keresés.zip fájl Keresés.docm-re, és készen is vagyunk.

Az utolsó lépés a kibővített felületű alkalmazás tesztelése. Megnyitjuk a Word-dokumentumot, és a Developer fültől jobbra meg-

jelölt szöveg webes keresésének eredményével. Ez mind szép és jó, de vajon hol lehet utánanézni a fenti fájlokban használt tulajdonságok lehetséges értékeinek? Nos, a „Customizing the Office (2007) Ribbon



A Live Preview máris népszerű

jelenik a létrehozott Search fül. Beírunk néhány szót, és kijelöljük a szöveg egy részét.

Először a Search fülre, majd a kedvenc keresőt jelképező gombra kattintunk, és egy böngészőablakot kapunk, benne a ki-

User Interface for Developers” cikksorozat (első rész: <http://msdn2.microsoft.com/en-us/library/ms406046.aspx>, második rész: <http://msdn2.microsoft.com/en-us/library/ms406047.aspx>) sok hasznos információt nyújt, bár a tördelés nem igazán intuitív.

Fejlesztők, előre!

Az új Office felhasználói felülete radikális változáson ment át, hogy leegyszerűsítse az ismert funkciók használatát, megkönnyítse az új szolgáltatások felfedezését és igénybevitelét, és lehetővé tegye professzionális dokumentumok előállítását pilótavizsga és sokszori újrapróbálkozás nélkül is. Az Office-alkalmazások képességeinek kibővítése a korábbi verziókhoz hasonlóan egyszerű maradt, az új felhasználói felületre történő kivezetés pedig néhány XML-fájl készítésére/módosítására redukálódott. Fejlesztők, figyelmelem: a második bétaváltozat már elérhető, a felhasználók ismerkednek vele, itt a remek alkalom az Office-ra épülő saját megoldások vonzó köntösbe csomagolására!

König Tibor

(tibork@microsoft.com) Microsoft Magyarország

Ahol a technológia és a lehetőség találkozik

Mi már látjuk a jövő tehetséges és lendületes IT-szakembereit,

- akik a világ legnagyobb szoftvergyártójánál,
- Magyarország egyik legjobb munkahelyén dolgoznak,
- akiknek a szakmai és személyes fejlődés, a nemzetközi karrierlehetőség, a kihívást jelentő feladatok jelentik az értéket,
- akik a Microsoft technológiája iránti elkötelezettségüket igazán tűzközelből, inspiráló és dinamikus környezetben, egy remek csapat tagjaként élik meg.

Dinamikusan bővülő csapatunk az Ön számára is tartogathat szakmai és személyes kihívásokat!

Látogasson el a

www.microsoft.hu/allasajanlatok
oldalra!

Szakmai bemutatkozó anyagát aktuális ajánlatainkra és adatbázisunkba is várjuk, az **allas@microsoft.hu** címre!

Neked
lehetőség.
Nekünk kihívás.

www.microsoft.com/emea/careers

© 2006 Microsoft Corporation. Minden jog fenntartva. A Microsoft és a „Neked lehetőség. Nekünk kihívás.”™ logó a Microsoft Corporation bejegyzett védjegye vagy védjegyei az Egyesült Államokban és/vagy más országokban.

Microsoft®

Neked lehetőség. Nekünk kihívás.™

DEP és ASLR

A Vista memória- és programvédelmének néhány vonatkozása.

A Microsoft 2003-ban indította el Security Development Lifecycle (SDL) kezdeményezését. Az iniciatíva célja olyan szoftverek és megoldások fejlesztése, amelyek készítése során a kezdetektől a befejezésig fő szempont a biztonsági kockázatok csökkentése. Az SDL szigorú és formális tervezési, fejlesztési, tesztelési, bevizsgálási és visszajelzési szabályokat ír elő a sérülékenységek száma és a támadási felület nagyságának csökkentése érdekében.

Néhány Microsoft-terméken már most is látszik az SDL-filozófia, de valójában a Vista lesz az első olyan kliensplatform, amely teljes egészében az SDL-szempontok alapján készül. Az SDL meglehetősen szerteágazó, számos technológiában kikristályosodó fogalomkör. Írásunkban a Vista két olyan biztonsági megoldására koncentrálunk, amelyek célja a memória és a futó programok védelme – illetve az utóbbiak rosszindulatú változatai elleni védelem. A választás nem önkényes, mert az önmagukban is sarkalatos védelmi lehetőségek egymással kombinálva teljesítik ki igazán a képességeiket.

Ne futtasd!

A helytelen memóriahasználat azt jelenti, hogy valamilyen program egy nem futtatható kódot tartalmazó memóriaterületet erővel kódfuttatásra akar használni. Tipikusan ilyenek azok a rosszindulatú programok, amelyek egy esetleges puffertúlsordulásos sérülékenységet kihasználva tárolnak le futtatható gépi utasítások sorozataként értelmezhető adatokat (=”program”) verem-, adat-, heap- stb. területként kijelölt memóriába, majd az így betöltött kódnak adják át a vezérlést.

A Data Execution Prevention (adatfuttatás-megelőzés) vagy NoExecute (ne futtasd) leegyszerűsítve olyan biztonsági technológia, amely a programok memóriahasználatának helyességét felügyeli. Ha a DEP észreveszi, hogy valamilyen program vagy folyamat helytelenül próbálja használni a memóriát – azaz illegális helyen akar kódot futtatni –, kíméletlenül lezárja azt, és figyelmeztetést küld a felhasználónak.

A DEP a modern Windowsok részeként közel egy időben jelent meg az XP második szervizsomagjában, a Windows XP Tablet PC 2005-ben és a Windows Server 2003 SP1-ben. A védelmi lehetőséget természetesen a Vista és a Microsoft további operációs rendszerei is tartalmazzák.

A DEP a sérülékenységgel nem tud mit kezdeni, a puffertúlsordulást és a program adatainak eltárolását nem akadályozza meg, de akkor már közbe tud avatkozni, amikor a rossz helyre tárolt kód futtatására kerülne sor.

Hardveres DEP

A DEP-nek két alapvető formája létezik: hardveres és szoftveres. A hardverrel érvényesített DEP olyan architektúrában lehetséges, amelyben a processzor támogatja azt.

A „jobb” processzorok (az összes 64 bites, valamint számos 32 bites AMD- és Intel CPU) a memória kezelésekor belsőleg használt lapozótábla-bejegyzések utolsó bitjét NX (No eXecute) mutatóként értelmezik: 0 esetén a hivatkozott területen lehetséges, 1 értéknél tilos a kód futtatása. (Meggjegyzendő, hogy az NX az eredeti AMD-s elnevezés, az Intel terminológiájában XD bitről – eXecute Disable – beszélünk.)

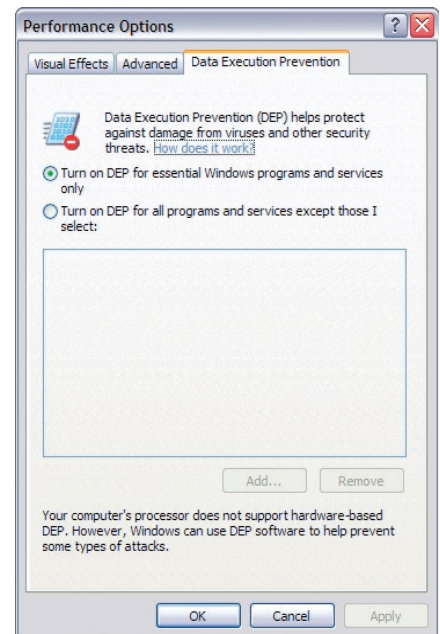
Előfordulhat az is, hogy legitim program futtatna adatterületen gépi kódot, de a DEP miatt nem lenne lehetséges. Az ilyen eseteken segít az a lehetőség, hogy a biztosan nem ártó szándé-

kú programokat meg lehet jelölni; fel lehet ruházni olyan jogosultsággal, hogy az „adatok” is futtassák.

A Windows XP, Server 2003 és Vista változatok elég okosak ahhoz, hogy a processzort felismerve kihasználják a hardveres DEP előnyeit, sőt még ennél is okosabbak, mert ha a processzor nem támogatja, akkor emulálják.

Szoftveres védelem

Az NX vagy XD bitről független szoftveres DEP, azaz Safe Structured Exception Handling (SafeSEH – biztonságos struktúrájú kivételkezelés) esetében kicsit bonyolultabban valósítható meg – de nem lehetetlen – a verem-, a heap vagy az adatszegmens rendszer-



A DEP állításának lehetősége Windows XP SP2 alatt (control panel – system – advanced fül – performance gomb – Data Execution Prevention fül)

szintű figyelése. A Windows esetében, mint azt a név is jelzi, a szoftveres DEP a kivételkezelő mechanizmusokon keresztül érvénye-

sül. A SafeSEH követelményeinek eleget tevő programoknak rendelkezniük kell ilyen funkciókkal, és futtatáskor regisztrálniuk kell a saját kivételkezelő eljárásukat.

Persze vannak olyan (régebbi) programok, amelyek nem a SafeSEH-megfelelőség szerint készültek. A szoftveres DEP ilyenkor úgy érvényesül, hogy a rendszer a kivételnek megfelelő, az azt kezelő funkció hívása előtt megvizsgálja: maga a kivételkezelő kód futtatható memóriaterületen van-e.

Végső soron a hardveres DEP is a kivételkezelésbe torkollik bele, mert az adatterületen történő kód futtatási kísérlet kivételt generál, amit az operációs rendszer kezel.

Összegezve: akár hardveres, akár szoftveres DEP-ről van szó, a rendszer nem javítja ki például a puffertúlcsordulásos sérülékenységeket, és nem akadályozza meg memóriaterületek felülírását. Amikor azonban a vezérlés adatterületre kerülne, akkor hardveres esetben könyörtelenül hardveres kivétel generálódik, amelynek értelmezésével le lehet fűlteni a goromba ágenseket. Szoftveres DEP-nél a kivételgenerálást az operációs rendszer szoftveresen végzi. Ennek a könyörtelensége, azaz a „jósága” attól függ, mennyire érzékenyen figyel a rendszer a helytelen memória-használatra.

A DEP másodlagos áldásait az alkalmazásokat és eszközmeghajtókat fejlesztők használhatják ki nagyszerűen, hiszen egyrészt kényszerítve vannak a memóriaterületek respektálására, és ez megfelelő programozói gyakorlat rögzüléséhez vezet. Másrészt a programozói hibából felülírt vezérlés nem „kékíti el a képernyőjüket” fejlesztés közben.

DEP-opciók

A „nem-futtatás” rendszerszintű beállításait az XP és a Server 2003 platformok esetében a boot.ini rendszerállományban használt kapcsolóval lehet megadni:

`/noexecute=szabály_szint`, ahol `szabály_szint` lehetséges értékei: **OptIn**, **OptOut**, **AlwaysOn**, **AlwaysOff**.

OptIn (alapbeállítás). A DEP csak a Windows saját (futtatható) rendszerállományaira és alkalmazásaira terjed ki.

OptOut. Ebben az esetben a DEP minden futó folyamatra érvényes lesz, de a felhasználók a Control Panel (Vezérlőpult) System

(Rendszer) appletjének segítségével létrehozhatnak kivétellistákat; az ezekben felsorolt programokra az operációs rendszer nem alkalmazza a memóriavédelmet. A fejlesztők az Application Compatibility Toolkit segítségével ugyancsak kivonhatják a programokat a DEP alól.

AlwaysOn. A DEP ennél az opciónál is minden folyamatra kiterjed, de a rendszer itt nem enged meg semmilyen felhasználói/fejlesztői kivételezést, azaz semmilyen listával vagy programozói fogással sem lehet kivonni semmilyen folyamatot a DEP védelme alól.

AlwaysOff. Az operációs rendszer semmilyen programot sem lát el DEP-védelemmel. A DEP ugyanúgy nem érvényesül a futtatható rendszerállományokon, ahogyan a felhasználói programokon sem.

A felsorolt kapcsolók mindegyike érvényesül hardveres és szoftveres DEP-nél egyaránt. Jegyezzük meg, hogy a rendszer vagy a szervizcsomag a boot.ini-be írt explicit `/noexecute=OptIn` kapcsolóval települ, illetve amennyiben a boot.ini-ből valamilyen okból eltávolítják a `/noexecute` kapcsolót, akkor a Windows úgy viselkedik, mint ha a `/noexecute=OptIn` lenne érvényben.

A boot.ini állomány könnyedén állítható valamilyen szkriptelő mechanizmussal, a bootcfg.exe vagy az msconfig.exe programokkal, illetve a rendszerbeállítások megfelelő szerkesztőeszközével.

A rendszerszintű beállítás mellett a Vezérlőpult Rendszer ikonjának segítségével a teljesítményre vonatkozó beállításoknál az adminisztrátorként bejelentkezett felhasználók kézzel is megadhatják, hogy milyen egyéb szabályok érvényesüljenek az OptIn és OptOut eseteknél:

- OptIn kondíció mellett itt lehet ki- vagy bekapcsolni az alapvető futtatható rendszerállományok védelmét;
- OptOut állapotban itt lehet felvenni a listába azokat a programokat és szolgáltatásokat, amelyekre ne terjedjen ki a DEP védelme.

Hogy a helyzet ne legyen túl egyszerű, az eddig leírtak kizárólag a 32 bites platformokra és a 64 bites rendszerek 32 bites folyamataira igazak. A 64 bites Windowsok összes 64 bites alkalmazása – legyen rendszer- vagy felhasználói program – axiómaszerűen DEP-konform. A 64 bites szoftverek nem futtathatnak kódot a visszatérési veremben vagy

a heapen és más adatterületeken, emellett futtatható folyamataik számára tárterület-foglalásakor kötelesek expliciten meghatározni annak futtathatóságát. Cserébe minden 64 bites programra kikapcsolhatatlanul kiterjed a DEP védelme. A 32 bites alkalmazásokra ugyanakkor 64 bites Windows alatt is mindenben érvényesek a 32 bites rendszer DEP-jéről leírtak.

Vista: csak kicsit más

A Windows Vistának nincs boot.ini állománya, ahol használni lehetne a `/noexecute` kapcsolót. A betöltés alatt beállítandó opciók bekerültek a BCD (Boot Configuration

Exception Handling

Csak emlékeztetőül: a kivételkezelők olyan mechanizmusok, amelyeket a program futásában bekövetkezett változások vagy előre meghatározott körülmények (kivételek) aktiválnak. Az Exception Handler megvizsgálja a kivételt, majd visszaadja a vezérlést, vagy anomália esetén ellenlépéseket tesz – a DEP esetében például mindenképpen leállítja a programot.

Példák kivételekre: osztás zérussal, memóriakezelési incidens, hardveres és szoftveres megszakítások stb.

Data) nevű struktúrába, ahol az adminisztrátorok más lehetőségek mellett a DEP-et, azaz a Vista esetében az „NX-házirendet” is állíthatják a bcdedit.exe parancssori eszközzel. (A Vistánál egyébként az eddig DEP-ként emlegetett technológiát szinte minden szakirodalmi forrás konzekvensen NX-ként említi.)

A paraméterek nélkül futtatott bcdedit parancs megmutatja az alapvető beállításokat. A „Boot Loader” szakaszban alapesetben ezt látjuk:

```
nx OptIn
```

Ez a 32 bites Vistánál azt jelenti, hogy alapesetben a korábbi platformokhoz hasonlóan csak a futtatható rendszerfájlokra érvényes a védelem. A 64 bites változatra ugyanaz igaz, mint amit az előző szakaszban leírtunk: minden 64 bites alkalmazás NX-védett, a 64 bites rendszeren futó 32 bites

programok védelme pedig megfelel a 32 bites platforménak.

A bcdedit kényelmes lehetőség a BCD paramétereinek állítására is. Az NX-házirend beállítására a következőképpen kell használni:

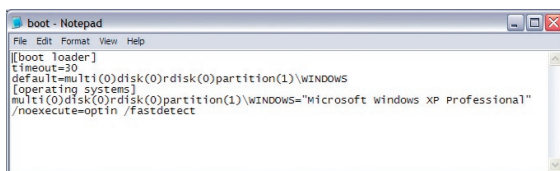
```
bcdedit /set nx szabály _szint
```

ahol *szabály_sztint* lehetséges értékei a korábbiakban leírtakhoz hasonlóan: OptIn, OptOut, AlwaysOn, AlwaysOff. Értelmezésük és működésük is megegyezik az előzőleg leírtakkal.

NX-es programok írása – nem csak programozóknak

Kódolásnál legtöbbször nem kell különösebben figyelni a kód NX-es vonatkozásaira; a 32 bites alkalmazások az érvényes beállítások szerint viselkednek majd, a 64 bitesek meg úgyis NX-védettek – erről a fordítóprogram és a Vista gondoskodik.

Bizonyos körülmények között elképzelhető, hogy alacsony szintű nyelven írt alkalmazásoknál az NX-megfelelőségre is figyelni kell. Ökölszabály, hogy a program ne futtasson kódot adatterületen. Ha az alkalmazás dinamikus kódot generál és futtat, akkor nem túl bonyolult programozási fogásokkal elkerülhetők a programok megfeneklésével járó kivételek.



A boot.ini és a benne lévő noexecute kapcsoló

A Microsoft Linker „/NXCOMPAT” kapcsolójának használatával gyakorlatilag kijelentjük az alkalmazásról, hogy NX-szempontról megfelelő, és ez le is van tesztelve. A Vista alá fejlesztett alkalmazásoknál általában célszerű használni ezt a kapcsolót; így a Vista is NX-védelemben futtatja majd a kódot. Ha speciálisan a Vista alá fejlesztett programoknál „/SUBSYSTEM:6.0” linkelési opciót használunk, akkor az „/NXCOMPAT”-ot nem kell külön megadni, mert automatikus a beállítása. Ha Vista alatt valamilyen okból ki akarunk vonni egy alkalmazást az NX-védelem alól, akkor ezt a „/NXCOMPAT:NO” opcióval érhetjük el.

Ha alkalmazásunk dinamikusan lefoglalt memóriaterületen akar kódot futtatni, akkor a VirtualAlloc() szabványos API-hívással kell lefoglalnia a tárterületet oly módon, hogy a Vista számára futtathatónak jelöli

don célszerű használni (más környezetekben a példának megfelelő behelyettesítésekkel):

```
pMem = VirtualAlloc(
    NULL,
    dwSize,
    MEM_COMMIT | MEM_RESERVE,
    PAGE_EXECUTE_READWRITE);
// futtatható és írható.
```

A fenti példában a memóriát írhatónak és futtathatónak jelöltük ki, ezért a lefoglalt terület de facto sérülékeny. Csak a lehető legrövidebb ideig célszerű fenntartani ezeket a viszonyokat, azaz addig, amíg a dinamikus generált kódot betöltjük a pMem által mutatott területre. Ezután kívánatos azonnal nem-írhatóvá nyilvánítani a VirtualProtect() API-hívással:

```
VirtualProtect(pMem,
    dwSize,
    PAGE_EXECUTE_READ,
    &dwOldProtect); // már nem írható, csak futtatható.
```

meg azt a memóriát, amire a foglalási igény vonatkozik.

A Vistában semmilyen más memóriafoglalási lehetőség – függvény vagy rendszerhívás, stb. – nem alkalmas futtatható tárterület lefoglalására! A VirtualAlloc() függvényt a következő beszédes kódrészlethez hasonló mó-

A leírtakkal megvalósítható egy zárt rendszer konzisztens védelme, de a külvilágból érkező alkalmazásokban ne bízunk meg; ezeknél tüzetes ellenőrzésre van szükség, mielőtt ráeresztjük őket a rendszerünkre.

ASLR

Az Address Space Layout Randomization vagy véletlenszerű címtérület-kiosztás azt jelenti, hogy a betöltött alkalmazás, program vagy folyamat kulcsterületei (futó kódja, könyvtárai, heap, verem stb.) véletlenszerű helyre kerülnek az erre szolgáló memóriában. Nem az összes kap egy konkrét véletlenszámmal megadott eltolást, hanem minden egyes modulhoz külön generálódik egy random érték. Ez a technika megnehezíti egy lehetséges külső támadó dolgát, mert egyrészt nem jósolható meg előre a konkrét memóriacím, másrészt a kiszámítása (helyesebben próbálgatása) is vesződéses.

A nyílt kódú ASLR-technológiát először Linux-rendszereken használták (OpenBSD, linuxos PaX és Exec Shield rendszerek); a Vista-tesztváltozatok közül a második bétánál látjuk először.

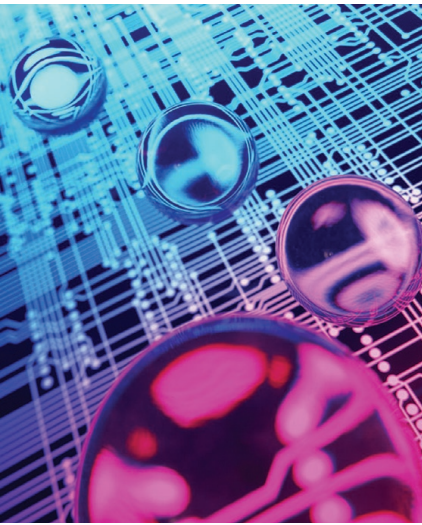
Heap-túlcsordulás

A program futtatása előtt „statikusan” (azért ez sem egészen egyszerű, de ezt most nagyvonalúan a processzor belügyének tekintjük) lefoglalt memóriaterületekkel ellentétben a heap (halom) olyan adatterület, amelyet futásidőben dinamikusan lehet lefoglalni, és abszolút változók helyett mutatókkal lehet rá hivatkozni. Ha egy rosszul megírt alkalmazás elmulasztja annak ellenőrzését, hogy a heapen lefoglalt változóba másolt adatok beférnek-e a dinamikusan lefoglalt területre, akkor egészen egyszerűen akkora adatmennyiséggel is „be lehet etetni”, amely a mérete miatt a dedikáltan lefoglalt területen túl elhelyezkedő változókat is felülírja.

Természetesen a hackernek tudnia kell, hogy mit ír felül, illetve mire és hogyan használhatja fel a sérülékenységet, valamint az is, hogy milyen adatokkal kell elvégezni a „beetétést”, de megfelelő ismeretek birtokában némi bűvészkedéssel gyakorlatilag tetszés szerinti helyre bármit beírhat, és átveheti az ellenőrzést a program futtatása felett.

A felülírás után az uralom átvétele a processzor PC (program counter, utasításmutató) regiszterének a megváltoztatásával lehetséges; erre is létezik egy sor trükk: lehet manipulálni a visszatérési vermet, az Exception Handlereket, és ki lehet használni az operációsrendszer-magok egyedi lehetőségeit (Windows esetében például az úgynevezett shatter attackkal).

Ha nagyon le akarunk egyszerűsíteni a dolgot, azt lehet mondani, a DEP halálpontosan arra figyel, hogy a processzor utasításmutatója ne vegyen fel meg nem engedett értéket.



Az ASLR jósága attól függ, hány bitet, az az mekkora entrópiafaktort használunk a memória randomizálásához; konkrétan egy folyamat kezdőcíme a kitalálási esélye $1/(2^b)$, ahol b a bitek száma.

Ha a támadónak esetleg több modulra is szüksége van, akkor tovább romlik a találati arány: $1/m \cdot (2^b)$, ahol m a célmodulok száma. Bizonyos ASLR-implementációknál a betöltés alatt még az egyes modulok betöltési sorrendjét is külön randomizálják (Library Load Order Randomization). Ebben az esetben a támadónak még a sorrendet is ki kell találnia, amire $1/n$ az esélye, ahol n a betöltendő modulok száma.

A Vista és az ASLR

A puffertúlsordulásokat kihasználó aknáktól az ASLR sem véd meg, de megnehezíti a támadó dolgát. Nyilván nem univerzális csodaszer, de más technológiákkal kombinálva hasznos védelmi eszköz, mert a támadó szempontjából a Vista „másképpen néz ki” – nem az, aminek látszik.

A Vista második bétaváltozatában nemcsak jelen van az ASLR, hanem alapértelmezésben be is van kapcsolva. Egy adott modul esetében az iménti szakasz releváns képletében $b=8$, azaz a Vistánál támadó $1/256$ találati aránnyal lehet sikeres a próbálkozásánál.

Az ASLR a következőkre terjed ki:

- visszatérési verem (a második béta utáni változatokban);
- heap- (vagy halom-) memória;
- az operációs rendszer részeként települő összes bináris állomány.

Minden más .exe vagy .dll állománynak

kifejezetten „kérnie” kell az ASLR-lehetőséget a megfelelő PE-fejlécmutató (Portable Executable Header) beállításával.

További védelmet jelent, hogy jelenlegi információink szerint az ASLR-oltalom alá eső folyamatok minden öket használó más alkalmazás vagy program esetén „újrarandomizálódnak”. (Ennek persze nyilván teljesítménykövetkezményei vannak.)

„Vermes” forgatókönyv

A b kis értéke miatt sokan támadják a Vista ASLR-jét. Ha belegondolunk, van is némi igazság abban, hogy egy laza brute-force rohammal gyorsan ki lehet találni a belépési pontokat – a mindössze 256 lehetőség miatt még a „nyers erő” kifejezés is túlzás.

Hangsúlyozzuk, hogy nem beszélhetünk tökéletes biztonságról, de a Vistán több olyan védelmi mechanizmus is működik, amelyek célja a támadások lehetséges hatásainak megelőzése vagy csökkentése. A technológiák együttese által nyújtott biztonság azonban nem egyenlő az egyes komponensek biztonságának az összegével, hanem annál több.

Példaként vegyünk egy olyan rendszerszolgáltatást, amelyet belezavartak egy visszatérési verem túlsordulásába. Eddig nem volt róla szó, de a C/C++-kódot Vista alá „/GS” opcióval kell fordítani, amely előírja a verem ellenőrzését, ezért „helyből” jó esély van rá, hogy a folyamat és így a támadó kódja is leáll. A visszatérési verem ráadásul a futó kód feletti memóriában van, ezért a túlsordulás miatt a program nem sérül, az alulcsordítás meg ritka. Mondjuk, hogy a támadó valahogyan mégis megtörte a verem védelmét. Ekkor jön az NX-falax, hogy elejét vegye az illegális terület futásának.

További információk

A futásidő védelem és a dinamikusan generált kód viszonyai – http://msdn.microsoft.com/security/productinfo/XPSP2/memoryprotection/exec_imp.aspx

DEP-FAQ – <http://windowshelp.microsoft.com/Windows/en-US/Help/186de3d0-01af-4d4c-981d-674637d2f4bf1033.msp>

Windows Vista Security (Technet) – <http://www.microsoft.com/technet/windowsvista/security/default.msp>

Tegyük fel, hogy a támadó még ezen is átverekszik magát. A legtöbb rosszindulatú programnak szüksége van olyan Windows-rendszerhívásokra, mint a socket() vagy a LoadLibrary(). Mivel ezek ASLR-védettek, ezért 256 lehetséges lokációt kellene végigzongorázni.

Összességében elég nehéznek látszik a támadó dolga, ráadásul a Vista második bétá-

ASLR kontra „return-to-libc”

A „return-to-libc” néven emlegetett támadásnál az „akna” olyan rendszerfunkciót hív meg, amelynek ismeri a memóriapozícióját. Az ASLR-nek az a dolga, hogy az ilyen funkciók belépési pontjait „szétszórja” a memóriában. Mivel nem lehet előre megjósolni a helyzetüket, ezért az akna nehezebben tud „korrektül” dolgozni.

jának ASLR-je még csak v1 változatú, ami azt jelenti, hogy lesz ez még (sokkal) jobb is.

„Halmos” forgatókönyv

Vegyünk most egy olyan esetet, amelyben egy szolgáltatás ismert heap-túlsordulási sérülékenységét próbálják kihasználni. A halom azonban erős fal mögött van, mert:

- a memóriablokkjaihoz ellenőrzőösszegek tartoznak;
- az adattartalma kódolt;
- a kiindulási címe „meg van ASLR-ezve”;
- a heapet kezelő funkciók belépési pontjai kódoltak;
- a Vista alkalmazásai problémára utaló kivételnél leállnak.

Ha a támadó magát és energiáit nem kímélve átverekszik magát ezen a szögesdróton, még mindig ott van egy csomó nehézség, amelyek – ha ebben a cikkben nem is szólunk róluk – a Vista erődítményének részét képezik: a „Service Hardening” és a „Service Control Manager” szigorú jogosultság-ellenőrzési mechanizmusai, illetve a rendszerszolgáltatásoknak a hálózati tűzfalal összehangolt házirendje...

És persze egy apróság: a tűzfal alapértelmezésben be van kapcsolva, ezért ártó kódunk csak azt látja a hálózathoz, amit a tűzfal nem takar ki előle. És persze vannak még más védelmi inyenccségek, amelyek a végső változatokban még jobbakk lesznek.

Kelemen László

MENEDZSMENT WS-* SZABVÁNYOK SEGÍTSÉGÉVEL

A rendszergazdáknak mélyen ismerniük kell, hogyan kommunikálnak egymással azok a rendszerek, amelyeket fel kell építeniük, vagy üzemeltetniük kell. Jelenleg ehhez leginkább a hálózati technológiák ismeretére van szükség, azonban ahogy a webszolgáltatások egyre inkább elterjednek, és már nemcsak webes alkalmazások, hanem akár a hardverek és más, üzleti szempontból kritikus alkalmazások is ezek segítségével kommunikálnak a külvilággal és egymással, szükség lesz ezeknek az új technológiáknak a megismerésére is.

Legutóbbi számunkban körüljártuk, hogyan is épülnek fel a webszolgáltatások, milyen rétegekre oszlanak, és miért lesznek egyre fontosabbak nemcsak a fejlesztők, hanem valamennyi informatikus számára. Ebben a cikkben mélyebbre ásunk, és megnézzük, hogyan működik a webszolgáltatások segítségével (különös tekintettel a WS-Managementre) a heterogén rendszerek és a hardverek felügyelete, menedzsentje az újgenerációs rendszereknél. Ez egyben arra is jó példa, hogy az elejétől a végéig lássuk, hogyan segítenek a WS-* szabványok az iparág megoldásainak egységesítésében.

Mit értünk rendszermenedzsenten?

A WS-Management specifikáció létrehozásának célja az volt, hogy olyan általános, SOAP alapú protokollt definiáljon, ami PC-k, szerverek, hardvereszközök, webszolgáltatások, alkalmazások és más menedzselhető entitások felügyeletéhez szükséges. A szoftver- és hardverrendszerek menedzsentjének általános problémáit vizsgálva arra jutottak a WS-Management specifikáció létrehozásában részt vevő vállalatok (Sun, Microsoft, BMC, Intel, AMD, Dell), hogy a legtöbb menedzsentmegoldás a következő feladatok elvégzésében vesz részt:

- a menedzselhető eszközök és erőforrások felfedezése, navigálás közöttük (Discover);
- az egyes erőforrások beállításainak és paramétereinek lekérdezése, módosítása, létrehozása, törlése és átnevezése (Get, Put, Create, Delete, Rename);

- nagyobb szekvenciák és listák megfelelő kezelése, például az eseménynaplóké (Enumerate);
- előfizetés adott események előfordulásának értesítésére (Subscribe);
- meghatározott menedzsentfeladatok elindításának lehetősége (Execute).

További információk

A WS-Management specifikációja – http://www.dmtf.org/standards/published_documents/DSPO226.pdf

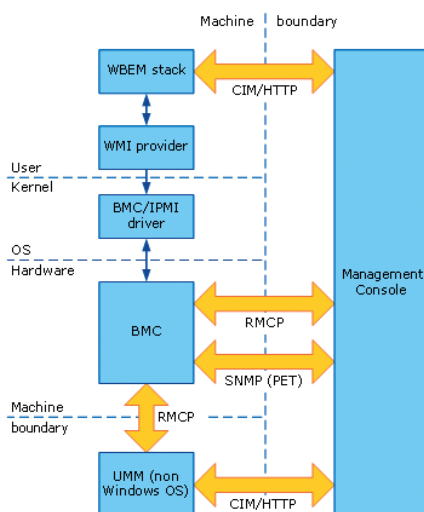
A WS-Management ezekre az általános feladatokra ad egységes ajánlást. A szabványt a Distributed Management Task Force

(DMTF) idén véglegesen elfogadta, így az mostantól valóban az alapját fogja képezni valamennyi menedzsementmegoldásnak. Mielőtt azonban elmélyednénk abban, hogy milyenek lesznek a WS-Managementre épülő megoldások, nézzük meg, milyen lehetőségeink vannak már ma, illetve milyen korábbi problémákra adott választ.

Hardvermenedzsement

Jóllehet a rendszermenedzsement nem csak a hardverre vonatkozik, mégis érdemes itt kezdeni az ismerkedést, hiszen a hardvermenedzsementnek nagyobb múltja van az informatikában. Elvégre minden szervernél fontos, hogy tudjunk annak állapotáról, és a hibákra mielőbb reagálni tudjunk, akár manu-

Proxy based architecture



A hardvermenedzsement hagyományos megvalósítása

lisan, akár automatizáltan. Természetesen az sem elhanyagolható szempont, hogy a hardverek felügyelete az egyik legmeghatározóbb szelete egy cég IT-kiadásainak.

Amikor egy márkás szervert vásárolunk (és nem mi barkácsoljuk össze), ahhoz általában kapunk az adott hardver felügyeletére alkalmas szoftvert is. A hardverek esetében fontos azonban, hogy ne csak akkor értesüljünk az eszközök állapotáról, amikor már fut az operációs rendszer rajta (ezt nevezzük in-band esetnek), hanem akkor is, amikor még csak éledezik, vagy amikor épp összeomlott valami hiba következtében (ez pedig az out-band). Ennek megvalósítására találták ki a Baseboard Management Controllert (BMC). A BMC folyamatosan figyeli a szerverek állapotát, és távoli hozzáférési lehetőséget biz-

tosít azokhoz, hogy az előforduló hibákról, illetve az eszköz aktuális állapotáról mielőbb értesülhessünk.

Több szabvány is született már a BMC felépítésének és kommunikációs protokolljainak definiálására, ezek közül a legnagyobb iparági támogatást az Intelligent Platform Management Interface (IPMI) kapta. Mégis, a rendszergazda szintjén ebből a szabványosítási erőfeszítésből mindeddig nem sok látott, ugyanis szinte minden gyártó szerverét eltérő szoftverekkel lehet távolról felügyelni. Ennek következtében még a legegyszerűbb menedzsementfeladatok elvégzéséhez is előfordul, hogy adott esetben 2-3 eltérő alkalmazást kell használnunk. Mi ennek az oka?

Egyfelől a BMC távoli elérésének protokollja, a speciálisan erre a célra kialakított, bináris Remote Management Control Protocol (RMCP), amihez valamennyi gyártó eltérő biztonsági mechanizmusokat alakított ki, és ezek persze nem kompatibilisek egymással. Ahhoz, hogy egy egységes felületről lehessen menedzselni a különböző gyártók hardvereit, a hagyományos proxy alapú megoldást alakították ki, vagyis a menedzsementfelületet alkotó szoftverhez érhetőek el bővítvények, és ezek a konkurens megoldásokat is képesek kezelni. Ugyancsak érdekes megoldásnak számított a HTTP alapú Web-Based Enterprise Management (WBEM) protokoll, azonban ez még nem a webszolgáltatások architektúrájára épült, és emiatt nem is tudott igazán elterjedni. Szintén a WBEM ellen szól, hogy azt egyik jelentősebb szerver-operációsrendszer sem támogatja, csak további, harmadik féltől származó szoftverek telepítése után működik.

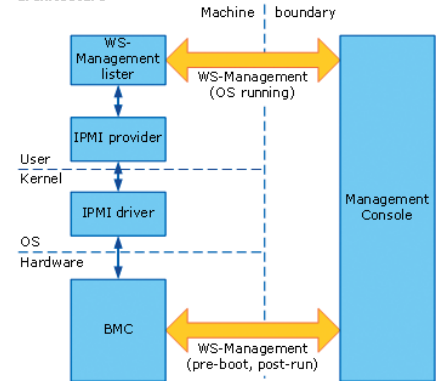
És ugyanez WS-Management segítségével?

A WS-Management most már iparágszerte elfogadott szabvánnyá vált, lehetővé téve, hogy valóban egységes rendszermenedzsement-megoldásokat lehessen kialakítani. A szabvány elfogadásában nagy szerepet játszott, hogy az további, már korábban elfogadott WS-* és kommunikációs specifikációkra épül, tisztázva a legfontosabb kommunikációs és biztonsági kérdéseket. (A WS-Management a HTTPS, SOAP over HTTP, SOAP 1.2, WS-Addressing, WS-Transfer, WS-Enumeration, WS-Eventing szabványokra épül.)

A WS-Management kialakításánál figyel-

tek arra is, hogy az egyformán hatékonyan tudjon működni az operációs rendszer és az egyes hardverelemek, illetve a BMC szintjén is, így nem kell különböző protokollokat kialakítani az in-band és az out-band menedzsementhez. Használatához már proxyra sincs szükség, hiszen mind a hardver, mind a menedzsementszoftver ugyanazt a nyelvet beszéli.

WS-Management based architecture



Hardvermenedzsement WS-Management segítségével

A Windows Server 2003 R2-ben találkozhattunk a WS-Management első implementációjával, ami elsősorban a WinRM (Windows Remote Management) parancssori eszköz révén érhető el. Az R2 ezenkívül további segítséget nyújt a hardvermenedzsement kapcsán az IPMI providerrel és az Event Collectorral.

Az IPMI provider képes arra, hogy a BMC és az operációs rendszer közötti kommunikációt megteremtse (WMI providerként). Ezt kiegészítve az Event Collector szolgáltatás képes előfizetni a WS-Managementen keresztül érkező eseményekre, és azokat tárolni a lokális eseménynaplóban. Ez a jelenlegi implementációval két esetben használható:

- ha fut a távoli szerveren az operációs rendszer (in-band), és az is egy R2;
- ha egy távoli szerver BMC-je támogatja a WS-Managementet, és annak az eseményeit naplózzuk (ez lehet in-band és out-band is).

Az Event Collectort a Windows Event Collector Utility (Wecutil.exe) parancssori alkalmazással lehet konfigurálni.

WinRM

A Microsoft első WS-Management alapon működő, távoli hardvermenedzsemen-

IT-BUSINESS

Infokommunikációs hetilap üzleti döntéshozóknak



Csak IT-BUSINESS előfizetőknek!

- a megjelenés előtt a magazin teljes tartalma elérhető a weboldalon
- teljes körű hozzáférés az IT-BUSINESS archívumhoz – cikkek 2003-tól

Tekintse meg mutatószámunkat weboldalunkon!

FIZESSEN ELŐ!

Az előfizetés 5 egyszerű módja:



Tel.: 06-1-888-3424



Fax: 06-1-888-3499



E-mail: terjesztes@vogelburda.hu



Internet: www.it-business.hu

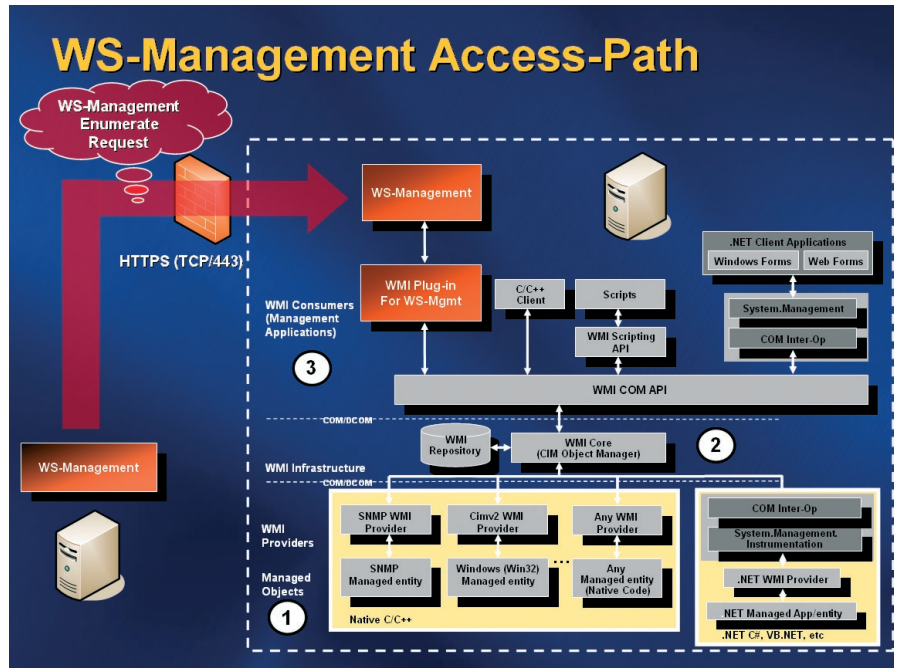


ITmédiabOLT: 1054 Budapest,
Bajcsy-Zsilinszky út 60.

Nyitva tartás: hétfő-péntek: 9-20, szombat: 10-18

tet támogató kliensalkalmazása a WinRM. Magáról a WinRM-ről mint technológia beszélünk, azonban a megvalósítása valójában egy VBScript (Winrm.cmd, ami a Winrm.

A WinRM használatához először is szükségünk lesz egy HTTP listener létrehozására, amit a legegyszerűbben a „winrm quickconfig” parancs kiadásával érhetünk el. Ezek



A WS-Management stack elhelyezkedése

vbs-t futtatja Cscript.exe-vel). Ez egyben azt is jelenti, hogy a WinRM scripting API segítségével testreszabhatjuk a működését.

A WinRM képes a kimenetet szövegesen vagy akár XML-ként is átadni a számunkra. A szöveges megjelenítés akkor lehet hasznos, ha kézzel kérdezzük le vele adatokat, az XML pedig inkább akkor, ha automatizálni szeretnénk a segítségével.

A WS-Management révén a WinRM egyben arra is alkalmas, hogy WMI-lekérdezéseket futassunk le távoli gépeken mindössze a HTTPS protokollt használva, anélkül, hogy DCOM-mal kellene kommunikálnia a két gépnek egymással. Ez a lehetőség különösen a tűzfalak miatt lehet kedvező.

Telepítés és használat

A Windows Server 2003 R2 itt felsorolt hardvermenedzsmet-képességei (és a WinRM, ami másra is jó lehet, például a remote WMI scripting kapcsán) alapbeállítás szerint nem találhatók meg egy újonnan telepített rendszeren, azt a Windows összetevőinél a Management and Monitoring tools alatt, a Hardware Management opciót választva tehetjük fel a rendszerre.

után már neki is állhatunk az első parancsok kiadásához, például:

A win32_operatingsystem lekérdezése:
winrm get cimv2/win32_operatingsystem -r:localhost

A futó folyamatok listájának lekérdezése:
winrm enumerate cimv2/win32_process -r:localhost

A -r: kapcsoló után természetesen távoli szerver címét is megadhatjuk, ekkor HTTPS-en keresztül történnek a lekérdezések. Ehhez szükség lehet a tanúsítványok megfelelő konfigurálására is.

Hogyan tovább?

A WS-Management szinte minden menedzsmetmegoldás részévé fog válni a következő évek során. A szabványra épülő eszközök és szoftverek felügyelete pedig ezek után lényegesen egyszerűbbé válik, és mindent képesek leszünk a számunkra leginkább szimpatikus felületről elérni. Természetesen erre nemcsak az itt bemutatott parancsori eszközökkel lesz lehetőség, hanem például a System Center Operations Manager 2007-tel is.

Budai Péter

(i-pbudai@microsoft.com) Microsoft Magyarország

Mi már látjuk,

hogyan védheti meg értékes információit.

A Microsoft® folyamatosan azon dolgozik, hogy termékei minél biztonságosabbak legyenek. Iparági partnereinkkel, kormányzati szervekkel együttműködve segítünk az adatvédelmi megoldások tökéletesítésében. Így biztosítjuk, hogy az üzlet a jövőben zökkenőmentesen fejlődjön.
www.microsoft.hu/lehetoseg



FESTŐK ÉS BÖDÖNÖK

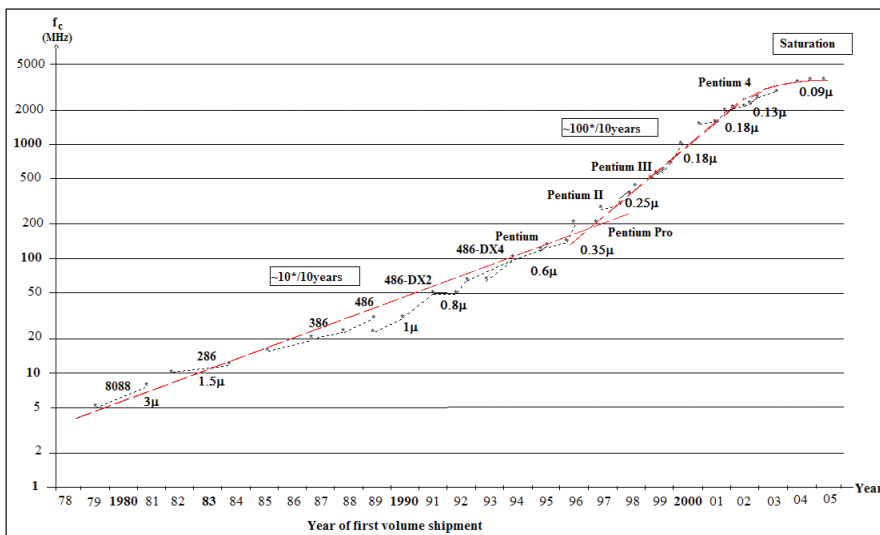
Ha valaki új asztali számítógép, esetleg szerver vásárlásán gondolkodik, általában alaposan körülnéz, milyen lehetőségek vannak éppen a piacon. Fontos szempont, hogy az adott hardvert később mennyire lehet tovább bővíteni, illetve mennyire felel meg a futtatni tervezett szoftverek igényeinek.

Az elmúlt pár évben észrevehető változás állt be a processzorok fejlődésében, ami egyben azzal is járt, hogy már egy gyártó termékénél sem lehet egyszerűen a frekvencia alapján eldönteni, melyik az erősebb.

A többprocesszoros rendszerek mellett teret hódítanak a többmagos architektúrák is (amelyek végeredményben több mikroprocesszort tartalmaznak egy szilíciumlapkán), és ezek a technológiák most már nemcsak a szerverek tervezésekor, hanem az asztali gépek építése esetében is megkerülhetetlenek lesznek. Nemcsak a legnagyobb szervereken futó szoftverek kapcsán lesz fontos kérdés az, hogy azok mennyire skálázódnak jól több processzormagon – egyetlen kliensalkalmazás sem lesz ugyanis jelentősen gyorsabb a legújabb processzorokon, hacsak azok nem képesek tökéletesen kihasználni a párhuzamos feldolgozás nyújtotta előnyöket.

A processzorok fejlődése

A problémakör megvilágítása érdekében kicsit mélyedjünk el abban, hogyan is zajlott le az elmúlt években a processzorok fejlődése. Adott egyrészt Moore törvénye, ami a tranzisztorok számának növekedését írja le: Moore állítása szerint minden második évben megduplázódik a lapkákra integrált tranzisztorok száma. De vajon ez azt jelenti-e, hogy a processzorok abszolút teljesítménye is hasonló arányban növekszik? Sajnos nem.



A processzorfrekvencia növekedésének üteme

A processzorok teljesítményének növekedése érdekében két fő irányt követtek a tervezők, ezek kivitelezéséhez folyamatosan szükség volt a tranzisztorok számának növelésére: az egyik sebességnövelő lehetőség a frekvencia emelése volt, a másik pedig a processzormag belső működésének optimalizálása. A 8088-as processzoroktól indulva egészen a Pentium Pro processzorok megjelenéséig elmondható volt, hogy mind a frekvencia növeléséből, mind a belső optimalizációból származó teljesítménynövekedés közel tízszeresére nőtt tíz év alatt, ezzel együtt mintegy százszorosára nőtt tízévente a processzorok abszolút sebessége is.

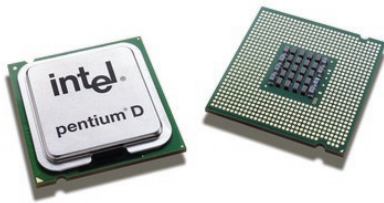
Azóta azonban szinte csak az órajel növekedett, ami a trendet tekintve tíz év alatt százszoros frekvencianövekedést jelentett volna, ugyanakkor a magok belső felépítése csak kicsi pozitív változásokat eredményezett. 2004 tájáig már erősen érezhető volt, hogy a hardvergyártók számára szinte lehetetlen kihívást jelent a 3,6–4 gigahertzes határ átlépése. Az Intel Pentium 4 alapjait alkotó NetBurst mikroarchitektúra, ami eredetileg 10 gigahertzig (később ezt a prognózist 5 gigahertzre módosította az Intel) képes lett volna skálázódni, technológiai korlátokba ütközött – túlságosan megnőtt ugyanis a frekvencia növelésével a processzor áramfelvétele, és hőkibocsátása is az egekbe szökött. Hasonló problémával találkozott az AMD mérnökei is.

Eddig a pontig, akár a frekvencia, akár a belső optimalizáció javult, új hardver vásárlásakor egyből minden szoftverünket gyor-

sabbnak érezhettük. Ez a korszak azonban lezárulni látszik, és a legújabb hardverek csak akkor tudnak majd újra jelentős sebességnövekedést elérni, ha abban a szoftverek is partnerek. Nem meglepő módon a régebbi programok (különösen a desktop-alkalmazások) egyáltalán nincsenek erre felkészítve. A szerverszoftvereket azonban már régebben is úgy tervezték, hogy több processzoron is hatékonyan fussanak, jól skálázódjanak fölfelé. Hamarosan azonban itt is találkozunk egy jelentős skálázhatósági problémával.

Többmagos processzorok

A processzorgyártók úgy döntöttek, hogy egy lapkán több processzormagot alakítanak ki, és ezzel növelik tovább a feldolgozás sebességét. Ez az új megoldás sokkal könnyebben kivitelezhető, hiszen a miniaturizáció még mindig fokozható, az egy lapkára integrált tranzisztorok száma is növelhető. Első körben azonban ez azzal járt, hogy az egyes magok frekvenciája alacsonyabb és belső felépi-



Az utolsó NetBurst architektúrájú Intel processzor, a kétmagos Pentium D

tésük is egyszerűbb lett (a fogyasztás csökkentése miatt), mint amilyen az egymagos processzorok esetében volt. Például az Intel kétmagos asztali processzoraiból kikerült a HyperThreading, és a frekvencia is 2,6–3,6 gigahertz között mozgott az egyes magok esetében. (Ugyanakkor az Intel NetBurst alapú kétmagos szerverprocesszorai továbbra is tartalmazzák a HyperThreadinget.)

A több mag kihasználása nagyjából ugyanazokat a programozói és tervezői technikákat igényli, mint amik a többprocesszoros rendszereken jól skálázó szoftverek esetében már régebb óta működnek. Ez azt jelenti, hogy a szoftvereknek képeseknek kell lenniük az általuk végzett feladatokat párhuzamosítani, hogy azokon egyszerre több processzor vagy processzormag is hatékonyan tudjon dolgozni. Fontos azonban kiemelni, hogy a jelenleg elterjedt szerverszoftverek többsége csak 8

processzorig skálázható kényelmesen, utána újabb processzorok hozzáadásával már nem nő számottevően a teljesítmény. Kivétel ez alól szinte valamennyi gyártó SQL alapú adatbázisszervere, ezek akár 64 processzonnal is jelentős teljesítménynövekedést képesek elérni. Az asztali szoftverek ilyen téren még hosszú fejlődés előtt állnak, hiszen még két processzort sem tudnak igazán kihasználni. Jelentős kihívást jelent majd ezeknek a skálázhatósági plafonoknak a feljebb tornászása.

Ez az áttérés meglehetősen komoly terhet ró a szoftverfejlesztőkre, különösen, ha figyelembe vesszük, hány mag lesz a következő évek processzoraiban. Az iparági elemzők szerint 2006 végéig már minden második új számítógép kétmagos processzorokkal lesz felszerelve, és közel két évente duplázódni fog a processzorba integrált magok száma. Ez egyben azt jelenti, hogy 2010-re már akár 16–32 magos processzorokkal is találkozhatunk, ami egy négyfoglatos rendszer esetében akár 128 magot is jelenthet majd. Ha ezekben lesz HyperThreading vagy ahhoz hasonló megoldás, akkor a logikai magok száma még ennek is a kétszerese lenne, vagyis 256. Jelenleg pedig nemigen akad olyan szoftver, amelyik képes lenne skálázódni ennyi párhuzamos magra.

Korábban, a processzorok belső optimalizációjának fokozása során azzal növelték a számítási és utasítás-végrehajtási teljesítményt, hogy a futó kódot, még akkor is, ha az egyetlen szálát használt (vagyis a programozó nem párhuzamosította a szoftvert), megpróbálták a processzoron belül párhuzamosá tenni. Erre többféle megoldás is született (ide kapcsolódnak a különböző generációs szuperskalár processzorok), azonban az egymást követő utasítások közötti adatfüggőségek miatt ezek hatékonyságát egy bizonyos szint után nem lehetett tovább növelni. Születtek még más optimalizáló megoldások is, például az utasítások előfeldolgozása (prefetch) vagy éppen az elágazások megbecslése (branch prediction) sorolható ide. A legtöbb megoldás közös vonása a Pentium Pro óta, hogy a kapott utasítássorozatokat más sorrendben hajtják végre, akár egymással párhuzamosan is, ezzel növelve az egy órajel alatt végrehajtható utasítások számát. Ezeket gyűjtőnéven OOO-nak, vagyis Out of Order végrehajtnak nevezzük.

Most már egyre inkább az a tendencia

látszik, hogy – biztosítandó a processzor fogyasztásának csökkentését miatt – ezek az optimalizációk szép lassan ki fognak kerülni a processzormagokból, és a processzor inkább párhuzamosan használható magokat, logikai vagy fizikai processzormagokat bocsát a programozók, a compilerek és az operációs rendszer rendelkezésére. Ezek megfelelő kihasználása azonban (gondoljunk csak vissza a mai szoftverek skálázhatóságára!) teljesen új szoftveres megoldásokat igényel.

Mit kell tudni a párhuzamosságról?

A párhuzamosság kihasználása által nyerhető maximális teljesítménynövekedés mértéke számos tényezőtől függ. A legfontosabb azonban az, hogy a lefuttatandó program kód milyen arányban oszlik soros, és milyen arányban párhuzamosítható részekre. A szoftver sebességének gyorsítását előidézheti, ha képesek vagyunk a párhuzamosítható részeket valóban egy időben futtatni, több szálon, akár több dedikált processzormagon.

Ez ugyanakkor önmagában kevés, hiszen a sebességgyorsítás maximumát minden esetben a soros komponens határozza meg, így ha valóban sok párhuzamos processzormagot vagy szálát feltételezünk, akkor csak úgy tudjuk jelentősen növelni a hatékonyságot, ha a soros tényezőt csökkentjük a párhuzamosíthatóak javára. Ezt konkrétan Amdahl törvénye fogalmazza meg, ami ellentétben Moore tapasztalati és spekulatív alapokon nyugvó megfigyelésével, valóban pontosan meghatározza, adott esetben meddig lehet növelni a teljesítményt.

Próbáljuk megvilágítani ezt a kérdéskört egy igencsak egyszerű, hétköznapi példa felhasználásával! Tegyük fel, hogy egy kerítést festünk. Az elképzelt kerítésünk pontosan 300 lécből áll, és szeretnénk ezt a lehető leggyorsabban lefesteni. A festés megkezdésekor szükség van 30 perc előkészítésre, a festés befejezésekor pedig ugyancsak 30 percre, hogy mindent elrámoljunk magunk után

Amdahl törvénye

$$\frac{1}{F + (1-F)/N}$$

F – szekvenciális tényező, N – a párhuzamos műveletvégzők száma. Az eredmény a maximális elérhető sebességnövekedés-szorzó.

(vegyük úgy, hogy ezek alkotják a folyamat soros komponensét). Mondjuk, hogy egy ember egy perc alatt egy lécfestésére képes. Ebből a következő összefüggések adódnak:

Festők száma	Szükséges idő	Sebesség-növekedés	Hatékonyság
1	360 = 30 + 300 + 30	1,0X	100,0%
2	210 = 30 + 150 + 30	1,7X	85,0%
10	90 = 30 + 30 + 30	4,0X	40,0%
100	63 = 30 + 3 + 30	5,7X	5,7%
Végtelen	60 = 30 + 0 + 30	6,0X	Nagyon alacsony

A valóság még bonyolultabb

Még ez alapján sem könnyű azonban eldönteni, hogy hány festő segítségét kérjük. Nyilvánvalóan látszik, hogy ha két ember dolgozik a festésen, akkor sem lesz kész kétszer gyorsabban a munka, hanem csak 150 percet lehet nyerni. Ha már 100 ember segítségét kérjük, akkor is csak 5,7-szeres sebességnövekedést értünk el. Ha a soros tényezőt képesek volnánk nullára redukálni, akkor valamennyi újabb festő bevonásával ugyanakkora arányban csökkenne a befejezéshez szükséges idő.

A valóságban természetesen ez még ennél is bonyolultabb, ugyanis a soros tényezőt gyakran növeli a párhuzamosan dolgozó emberek – vagy számítógép esetében a szálak – összefogása, szinkronizálása. Vegyük például csak azt az egyszerű esetet, hogy mindössze egyetlen festékesbödönt használ valamennyi munkásunk. Ebben az esetben a festőknek, mondjuk, minden ötödik ecsetvonás után meg kellene várniuk, míg hozzáférnek a festékesbödönhöz. Arra is fokozottan figyelni kell valamilyen folyamatrendszer kialakításával, hogy véletlenül se akarjon egyszerre kettő vagy több festő rávetődni a festékre, hiszen abból ki tudja, milyen balesetek történnének.

Most már talán érthető, miért is olyan nehéz a többszálú programok írása, azok megfelelő tesztelése, valamint annak elérése, hogy azok tetszőlegesen sok processzormagon is folyamatos sebességnövekedést produkáljanak. Hasonlóan nehéz feladat, hogy a párhuzamosságot elméletileg támogató szoftverek megfelelő hardvereken és helyes beállításokkal valóban képesek legyenek elérni a maximális sebességüket. Mindezekon felül valakinek döntést is kell hoznia adott esetben, hogy mi az a határ, ameddig még megéri

bővíteni az adott rendszert, vagyis meddig éri meg további értékes forintokat áldozni a teljesítmény egyre kisebb mértékű növekedésére.

Szálak egymás között

A számítógép az operációs rendszertől, de leginkább a kernelétől függően számos lehetőséget ad a párhuzamosság kezelésére. Szinte minden környezetben rendelkezésre állnak úgynevezett szálak (thread), amelyek látszólag párhuzamos futtatására még az egymagos processzorok is képesek, igaz ezek valójában csak egymást felváltva, időosztásos alapon futnak. Többmagos vagy többprocesszoros rendszerek esetében ezek a szálak akár ténylegesen is képesek egy időben futni a hardveren.

Csakúgy, mint az előbbi példában, a szálak közt is gyakori, hogy azok egy közös erőforrást használnak, például a memória adott részét. Ilyenkor garantálni kell, hogy az egyik szál ne tudja megzavarni a másik működését. Ez abban az esetben viszonylag egyszerű, ha mindkét szál csak olvasni szeretné az adott memóriaterületet, hiszen akkor általában nem kell várniuk egymásra (az előző példából kiindulva ez az az eset lenne, amikor minden festő csak rápillantana a festék dobozára, hogy van-e még benne egyáltalán, érdemes-e odamennie, hogy belemártsa újra az ecsetét). Ha azonban az egyik szál módosítani is akar a memória tartalmán (a példánál maradva: miután megállapította, hogy van még elég festék, csökkenti a festékesdoboz festéktartalmát), akkor már komolyabb szinkronizációs eszközökhöz kell nyúlni.

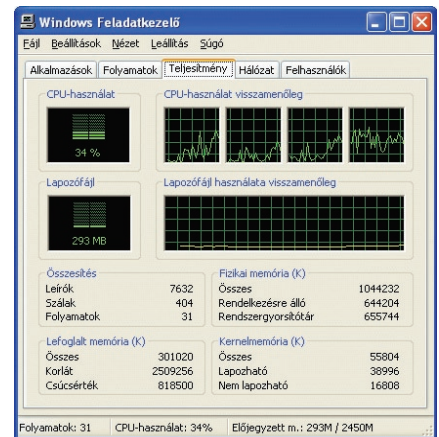
A legalapvetőbb megoldás erre az, hogy zároljuk (lock) a memória adott szeletét, és amíg ez a zárolás tart, addig valamennyi szálnak várakoznia kell a zárolást végzőre. Ha ezt nem tennénk, akkor több szál is megkísérelhetné egy időben csökkenteni az általa korábban kiolvasott memóriaterület értékét. Például egyszerre vagy éppen egymás után 20 szál kiolvasa, hogy most éppen 3 liter festék áll még rendelkezésre, és mind azt hiszi, hogy ő egyedül csökkenti azt 0,1 literrel.

Rosszul, lockolás nélkül megírt kód esetén ez azt eredményezheti, hogy a legutolsó szál is abban a hiszemben csökkenti a 3 literet 2,9-re, hogy senki más nem nyúlt előtte hozzá, pedig valójában ekkorra már 2 liter festék fogyott

el, és nem csak 1 deciliter. A legjobb megoldás ebben az esetben természetesen az, ha minden lockolással járó memóriairás után a következő szál, amelyik sikeresen hozzájut a közös erőforráshoz, maga is lockolja azt, kiolvasa annak aktuális értékét, és ennek megfelelően csökkenti, majd elengedi a lockot.

Szerencsére erre lényegesen kifinomultabb megoldások is rendelkezésünkre állnak többszálú alkalmazások programozásakor, azonban ha ilyenekkel dolgozunk, ehhez nagyban hasonló gondolkodásmódot kell alkalmaznunk, hogy minden lehetőségre felkészüljünk.

A szálkezelés és a szálak szinkronizálása azonban valóban növeli a program soros összetevőit, ami egyenesen arányos a közösen használt erőforrások és azok elérésének számával. Érdekes és szorosan ide kapcsolódó



Egy kétmagos, HyperThreadinget támogató processzor négy logikai CPU-nak látszik a feladatkezelőben

kérdés, hogy az operációs rendszer milyen módon osztja el a futó szálakat a processzormagok között azok prioritása és affinitása alapján, és ez milyen további idővesztéssel járhat a cache-ek és a memória másolása, illetve a kontextusváltás miatt.

Rendszergazdáknak

Ha az olvasót inkább a rendszerek kapacitásának tervezése, illetve a nagyobb, jól skálázható rendszerek építése és felügyelete érdekli, érdemes elsőként elolvasni a Windows Server 2003 Performance Guideot (Mark Friedman és a Microsoft Windows Performance Team közös munkája). Ez részletesen kifejti, hogyan lehet akár scale-up (az adott szerver teljesítményének növelésével), akár scale-out (több szerverszámítógép hozzá-

adásával) megoldásokkal növelni a Windows Server 2003 alapú szerverek és a rajtuk futó szolgáltatások (például fájlszerver, IIS-webszerver, saját .Net-alkalmazás) teljesítményét. Szintén ide kapcsolódik a Windows Internals negyedik kiadása (a szerzők: David Solomon és Mark Russinovich), ami pontosan bemutatja, hogyan működik a kernel és az operációs rendszer szintjén a szálak és az erőforrások kezelése, külön kiemelve a párhuzamos architektúrájú rendszereket is.

Ebben a cikkben leginkább a scale-up megoldásokkal kapcsolatos kérdéseket ecsegettük, azonban nem szabad azt sem elfelejteni, hogy immár a Microsoft is rendelkezik igazán univerzális scale-out megoldással, ez pedig nem más, mint a Windows Server Compute Cluster Edition, amellyel tetszőleges számú számítógépre terjeszthetjük ki a nagy számítási kapacitást igénylő feladatokat. Például az Office 2007 Excel Serverre képes lesz kihasználni ezt a tetszőlegesen skálázható architektúrát komolyabb (például banki, pénzügyi) számítások elvégzéséhez.

A scale-out megoldások azonban már régebb óta megtalálhatók a Microsoft szerveral-

kalmazásainak többségében, bár mindegyikben kicsit másképp kell implementálni, más trükkökre kell odafigyelni. Általánosságban elmondható, hogy ezek többsége a közös erőforrások replikációjára és a terhelésmegosztásra épül, azonban egyedül az SQL Serverben erre legalább ötféle különböző megoldás áll rendelkezésre, és mindegyiknek megvannak a maga előnyei és adott esetben hátrányai.

Fejlesztőknek

Ha azonban többszálú, skálázható alkalmazások írására adjuk a fejünket, akkor készülünk fel alaposan, nem kevés információn kell átrágni magunkat ahhoz, hogy készségszinten tudjuk alkalmazni az itt leírtakat valós helyzetekben. Kiindulásképpen érdemes áttanulmányozni az Microsoft SQL Server, illetve más ACID elven működő adatbázisszerverek tranzakció-kezelését, izolációs szintjeit és a mögötte húzódó logikát, valamint a szálkezelés és szinkronizálás alapjait akár .Net-en, akár hagyományos C++, Win32-es környezetben. Nem kevés információra – és nagy türelemre – lesz szüksége annak, aki valóban hatékony, többszálú al-

kalmazást szeretne készíteni, viszont az eredmény kárpótol a befektetett energiáért.

Minden esetben törekedjünk arra, hogy a kódunk először egymagos rendszeren működjön hibátlanul, de közben azért minimálisan figyeljünk oda a szálak szinkronizációjára. Ha már minden tökéletes az egymagos környezetben, csak akkor próbáljuk ki a szoftvert többmagos rendszeren, és ekkor javítsuk ki a szálszinkronizáció hibájából fakadó problémákat. Ha valaki egyből a párhuzamosságból adódó hibákkal találkozik, szinte esélye sem lesz kijavítani a sokkal egyszerűbb funkcionális hibákat.

A Windows Vista fejlesztői is mindent megtesznek annak érdekében, hogy maga a felhasználói felület valóban ki tudja majd használni a többmagos rendszerek előnyeit. A jelenlegi mérések szerint – amit IE7 böngészőkkel és Windows Media Playerrel, hátterben történő zenelejátszással teszteltek –, a duplamagos rendszerek mintegy 20–25 százalékos sebességnövekedést értek el a hasonló egymagos rendszerekkel szemben.

Budai Péter

(i-pbudai@microsoft.com) Microsoft Magyarország

Az informatikai biztonság és az információbiztonság legaktuálisabb kérdései

IT-SECURITY@2006

Az elmúlt 12 hónap legérdekesebb IT-biztonsági témái

Magyarország legjelentősebb informatikai biztonsági szakembereinek véleménye

Minden, amit az IT-biztonságról 2006-ban tudni érdemes

Bővebb információ:

www.it-business.hu

Kiemelt partnereink:

Fókusz Könyvruház, VII., Rákóczi út 14–16.

Szakkönyvruház, VI., Nagymező u. 43.

Gellért Könyvesbolt, XI., Bartók Béla u. 15.

ITmédiabOLT, XIII., Visegrádi utca 11.



Ára: 4000 Ft

Keresse a nagyobb könyvesboltokban, vagy rendelje meg kiadónktól!

06 (1) 888-3421, 22 06 (1) 888-3499 www.itmediabolt.hu terjesztes@vogelburda.hu

KÉTSZER HÚSZ ÉRV II.

Az üzemeltetők egy része az ISA Server 2004 Standarddal kapcsolatban a tervezés, illetve a bevezetés fárasztó óráit, napjait éli, és sokan vannak, akik még nem tértek át az ISA 2000-ről, de szeretnének.

Folytatjuk az ISA Server 2004 Standard kulcsfontosságú jellemzőinek, előnyös vonásainak az előző lapszámban megkezdett ismertetését, kiegészítve a gyakorlatban felhalmozott tapasztalatokkal.

21. Űrlap alapú hitelesítés

Új lehetőséggel bővült az ISA Server 2004 hitelesítési opcióinak listája, amely elsősorban az OWA- (Outlook Web Access) publikálás fontos eleme. Az FBA (Forms-Based Authentication) olyan, Asp.Net alapú hitelesítési típus, amely egy HTML űrlapon keresztül képes feldolgozni a belépési információt. Egyben lehetőséget ad arra is, hogy a felhasználó az OWA bejelentkező képernyőjén eldöntse, milyen biztonsági körülmények (publikus/privát) között fogja használni az Exchange Server HTTP-kliensét. Az üzemeltető viszont e választástól függően képes befolyásolni az OWA működését, azaz például az automatikus kiléptetés intervallumát vagy a csatlások megtekintésének tiltását. Ami még fontos: az ilyen belépés után a hitelesítési információ nem kerül semmilyen gyorsítótárba, azaz nem reprodukálható, a felhasználónak muszáj újra belépnie, ha elnavigál az oldalról, bezárja a böngészőt, illetve ha kilép. Az FBA-hitelesítést az Exchange Server 2003 önmagában is ismeri, ám az ISA 2004-gyel űrlap alapú hitelesítési megoldásának használata mellett két további fontos előny is szól:

- nem az Exchange-kiszolgálót terheljük a hitelesítéssel, a direkt vagy véletlenül „elrontott” bejelentkezési kísérletek nem kerülnek be a védett hálózatba;
- bármilyen Exchange-verziót használhatunk az ISA 2004 „mögött” (akár még az 5.5-öst is), az FBA a rendelkezésre áll.

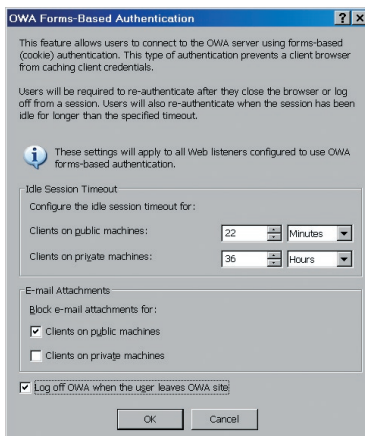
22. TS-elérés SSL VPN-nel

A Windows Server 2003 SP1 telepítése után a Terminal Server támogatja az „RDP over SSL”

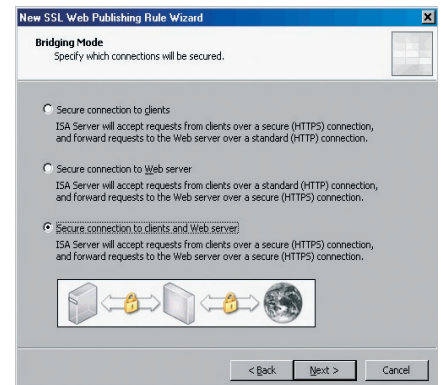
megoldást, azaz a terminálkapcsolatok használatát VPN-en keresztül. Ezt az ISA Server 2004 azzal egészíti ki, hogy lehetővé teszi a Terminal Server publikálását az SSL VPN-kapcsolat segítségével. Ez azt jelenti a gyakorlatban, hogy az ISA egy nem átlátszó TCP-kapcsolatot hoz létre a kliens és a szerver között, a hitelesítés és a titkosítás közvetlenül a két fél között zajlik, az ISA felügyelete és beavatkozása nélkül.

23. SSL Tunneling/Bridging

Az új webpublikálás varázsló értelemszerűen lehetővé teszi számunkra a webszerverek SSL-lel történő elérését akár az internet felől, akár a belső hálózatból. Újdonság viszont, hogy a szimpla „SSL Tunneling” (közvetlen továbbítási) módon kívül, egy spe-



Az FBA-opciók az OWA publikáló szabályának beállításai között



Az SSL-támogatás hatóköre

ciális üzemmódot is használhatunk. Ez az úgynevezett SSL Bridging, amelynek során az ISA 2004 megszakítja a webszerver felé irányuló forgalmat (nem „feltöri”, ahogyan néha tévesen elhangzik), „lehántja” róla a titkosítást, majd az integrált HTTP-filter segítségével megvizsgálja a tartalmat, kiszűri az ártalmas részeket, illetve az általunk beállított kritériumoknak megfelelően tisztítja meg a HTTP-folyamot. Ha ezzel elkészül, igény szerint újra hozzáteszi a csomaghoz a tanúsítványt, és továbbítja mindezt az ebből a műveletből persze semmit sem észlelő webszervernek. Abban pedig szabadon dönthetünk, hogy az SSL-támogatás a teljes, kliens/szerver-forgalomban működjön vagy csak a kliens és az ISA 2004 között, illetve csak az ISA 2004 és a webszerver között.

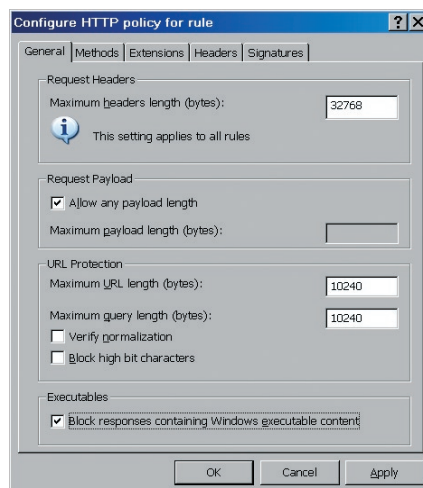
24. Kikényszerített titkosítás a biztonságos Exchange RPC-kapcsolatoknál

A távoli MAPI-kliensek (Outlook) ISA-kiszolgálón keresztüli kezelése esetén létrehozhatunk egy olyan RPC-házirendet, amely tiltani fogja a nem titkosított RPC-kapcsolatok használatát. Mindezt az 56 bites MD5 titkosítást végző RPC-alkalmazásfilter teszi lehetővé és egyszerűvé, hiszen gyakorlatilag ennek az opciónak az alkalmazása egy kattintással megoldható. Vegyük figyelembe ugyanakkor, hogy az ISA 2000 Feature Pack 1-ben debütáló Secure RPC megoldás sem az igazi, hiszen több veszélyes port (TCP 135 és egyéb dinamikus portok) kinyitásával jár együtt. Az igazi megoldás – VPN hiányában – az „RPC over HTTP”, amelyet az ISA 2004 természetesen szintén támogat, és amely szintén lehetővé teszi a külső MAPI-kapcsolatokat, ám csak a 80-as vagy a 443-as portok szükségesek hozzá. Ennek a megoldásnak a beüzemelése viszont csak számtalan előzetes kritérium teljesítése esetén kezdődhet meg, akár a kliens-, akár a szerveroldalt (Exchange) nézzük, valamint a beállítása sem egyszerű folyamat, viszont mindenképpen megéri.

25. Szabályonként alkalmazható HTTP-filter

Az ISA 2004 egyik legnagyobb „durranása” a HTTP-filter, amely a 80-as portot használó alkalmazások, illetve a HTTP-be csomagolt egyéb folyamatok alapos szűrését teszi

lehetővé. Ez a lehetőség megold egy halom problémát, hiszen a 80-as portot nem lehet tiltani, viszont így nem lehet felhasználni rosszindulatú vagy nagy sávszélességet igénylő alkalmazások rejtett futtatására sem.



A HTTP-filter General füle

Ebbe a körbe tartoznak például a fájlcsere-lők, a torrent-alkalmazások, a webes azonnali üzenetküldők (webes ICQ, MSN stb.). A HTTP-filter előnyeit számba véve az első fontos dolog, hogy a filter beállításai tűzfal-szabályonként különbözöek lehetnek, vagyis teljesen más konfigurációt használhatunk a publikálósabályoknál vagy a felhasználói internet-hozzáférés különböző változatainál. Egyetlen kivétel van a filter opciói között, ez pedig a „Maximum Headers Length”, azaz a fejlécek hosszát szabályozó opció, amely globális beállítás.

26. HTTP-szűrés minden típusú kapcsolatra

Az ISA 2000 Server is képes volt bizonyos tiltást végezni, de csak a webproxy-kliensek HTTP-forgalmában (MIME-típus alapján), illetve FTP-kapcsolatainál (kiterjesztés alapján). Az ISA 2004-ben nincs megszorítás, bármilyen HTTP-t használó kliens esetén működhet a tiltás.

27. A futtatható tartalom blokkolása

A HTTP-filter előnyei több szálon is megmutatkoznak. Egyszerűen megtehetjük például azt, hogy az összes futtatható – vagyis a HTTP-kéréssel, MZ sztringgel kezdődő – állomány letöltését blokkoljuk (Block responses containing Windows executable

content). Ne feledjük, ez az opció csak szabályonként, egyesével élesíthető.

28. Állományok letöltésének tiltása az állománytípus függvényében

Beállíthatjuk a HTTP-filteret úgy is (Extension fül), hogy az általunk megjelölt állománytípusokat ne tölthesse le a felhasználó – vagy éppen csak azokat tölthesse le, amelyeket megengedünk. A szűrő minden HTTP-kérést átvizsgál ilyenkor, és negligálja azokat, amelyek az általunk konfigurált kiterjesztéseket tartalmazza. Az ISA Server a vizsgálat során figyelembe veszi az URL utolsó pontja („.”) után következő kiterjesztést (például http://www.cim.hu/file.ext esetén az „ext” a vizsgálat tárgya), illetve azokat is, amelyek az esetleges „/” vagy a „.” után következnek a címben (ha nincs mögötte már több karakter). Abban az esetben viszont, ha több kiterjesztés is van az URL-ben, akkor csak az elsőt foglalkozik, például http://www.cim.hu/info.exe/html.asp esetén az „exe”-t hasonlítja össze a listánkkal, nem az „asp”-t.

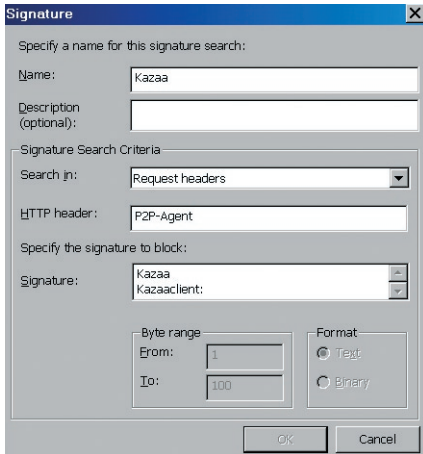
A filter beállításánál háromféle lehetőségünk van a szűrés hatókörére:

- Allow all extensions: nincs szűrés (alapértelmezett);
- Allow selected extensions: minden kérés blokkolva, csak a listában található elemek a kivételek;
- Block specified extensions (allow all others): minden, a listában található elem blokkolva, az összes többi engedélyezett.

29. Kulcsszavak, illetve szignatúrák alapján történő HTTP-tiltás

A HTTP-szignatúra (aláírás) egy, az adott alkalmazásra jellemző egyéni karaktersorozat, amely a HTTP-fejlécben vagy akár a törzsben is előfordulhat. Ahhoz, hogy blokkoljuk az ezt az aláírást hordozó csomagot, meg kell adnunk a szignatúra pontos formáját, illetve meg kell mondanunk a tartózkodási helyét. Minderre szintén a HTTP-filter ad lehetőséget a „Signatures” fül alatt. A kulcsszavak megadása szintén itt történhet meg, azaz tetsoleges szavakat is belevehetünk a listába, a szűrő pedig blokkolni fogja az ezt tartalmazó oldalakat. A folyamat azért ennél több figyelmet érdemel, hiszen gondoljunk bele, ha például a „Mozilla” szót óhajtjuk blokkolni, akkor a HTTP-oldalak jelentős százalékát megfogyjuk, mivel a Mozilla-kompatibilitást jel-

ző sztringet („User-Agent:...”) a weboldalak többsége a fejlécben hordozza. Visszatérve a szignatúrákra, általában ezt a lehetőséget használjuk a 80-as porton közlekedő nem engedélyezett alkalmazások – például a Kazaa fájlcsere – korlátozására.



Tiltás a szignatúra alapján

Az effajta szűrés feltétele, hogy a HTTP-kérések és -válaszok az UTF-8 enkódolásban közlekedjenek, ha nem ez az alapértelmezett forma, akkor a szignatúrablokkolás sajnos nem működik. Jó hír viszont, hogy egy – a cd-n mellékelt (\sdk\samples\admin\HttpFilterConfig.vbs) – szkript segítségével a HTTP-filternek az adott tűzfalszabályhoz tartozó beállításai importálhatók/exportálhatók.

Még egy megjegyzés a témához: a HTTP-policy készítésekor, ha a „request body” illetve a „response body” a kiválasztott keresőterület, akkor e panel alján megadhatjuk, hogy a célterület mekkora részében keressen. Alapértelmezés szerint ez az első 100 bajtót jelenti, de szabadon növelhető, persze azt is tudnunk kell, hogy mindez a teljesítmény rovására fog menni.

Maradt még egy kérdés: hogyan szerezhetjük meg ezeket a szignatúrákat? Nos, akár a Network Monitorral a csomagokba belelézve, vagy egy jóval egyszerűbb módszerrel, a <http://www.microsoft.com/technet/prodtechnol/isa/2004/plan/commonapplication-signatures.mspx> címen fellelhető táblázatból kiszemezgetve.

30. Szűrés a HTTP-metódusok alapján

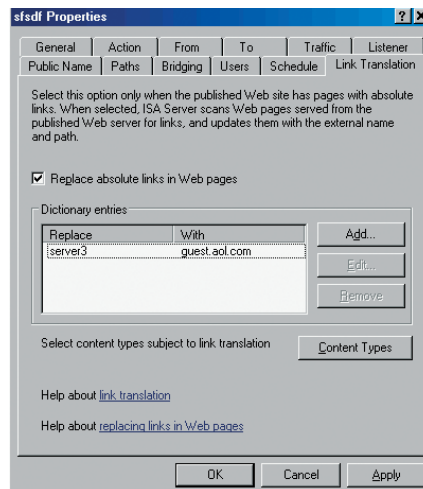
Az utolsó HTTP-filterrel kapcsolatos opciónk a szűrés a különböző HTTP-utasítások alap-

ján. A filter „Methods” fülén a már ismertett hatókörök szerint, egyszerűen csak be kell vinnünk a kérdéses metódust, amelyet alkalmazni szeretnénk az adott tűzfalszabályra. Konkrétan például a „GET” utasítás tiltása egyetlen weboldalt sem fog engedélyezni, vagy például a „PUT” tiltása esetén a távoli webszerveren tárolt őrlapok (egy kérdőív vagy egy regisztráció stb.) kitöltését akadályozzuk meg.

31. Link Translator

Előfordul, hogy a publikált weboldalaink HTML kódjában, a hivatkozások között bent maradnak a belső vagy lokális hivatkozások, amelyek az internet „felől” nyilvánvalóan feloldhatatlanok, ráadásul a szükségesnél több információt adnak az oldalainkat böngésző kezébe. A Link Translator szolgáltatás segítségével viszont az ISA 2004 képes arra, hogy ezeket a hivatkozásokat a publikus névre „fordítsa le”, azaz korrigálja a hibás neveket. Ez a megoldás egyaránt érvényesülhet a HTTP, illetve a HTTPS-publikálásban is. Az ISA 2004 több különböző szinten képes elvégezni ezt a „munkát”:

1. **Fejléckorrekció.** Alapértelmezésbe bekapcsolt állapotban van, és a fejlécben el-



A csere alapja a felvitt érték

helyezett hivatkozásokat vizsgálja, majd – ha szükséges – korrigálja.

2. **Fejléc + törzs korrekció.** A módszer ugyanaz, mint az előző esetben, különbség viszont, hogy valamennyi, az egész törzsben lévő hibás hivatkozás kijavítható.

Nem alapértelmezett, nekünk kell a publikálósabályt módosítani, azaz engedélyez-

ni a cseréltetést („Replace absolute links in Web pages” négyzet a publikálósabály Link Translation fülén).

3. **Korrekció egyéb weblapok esetén.** A Link Translation csak az adott tűzfalszabállyal publikált webszerverek esetén működik a fenti két változatban. Ahhoz, hogy bármely webszerverünk esetén használhassuk ezt a szolgáltatást, fel kell vennünk az előbb ismertett helyen egyesével a belső és az ezt helyettesítő, kifelé szánt sztringeket. Ezt az információt az ISA a „link dictionary”-ban tárolja majd.

32. Valós idejű monitorozás

A real-time naplónézegetés nagyon sokat segíthet az ISA-val kapcsolatos problémák felkutatásában. Mivel rettenetes mennyiségű kritérium alapján szűrhetünk, és mivel valóban élőben követhetjük az ISA MMC-ben a folyamatokat, azért könnyedén kideríthetők például az egyes kliensek működésével kapcsolatos hibák, különösen úgy, hogy mindig látszik az az információ is, hogy melyik tűzfalszabály engedélyezi vagy tiltja az adott forgalmat.

Tudnunk kell, hogy az online naplónézegetéshez be kell kapcsolnunk a telepítés közben az „Advanced Logging” szolgáltatást, valamint, hogy MSDE-adatbázisba kell naplónznunk.

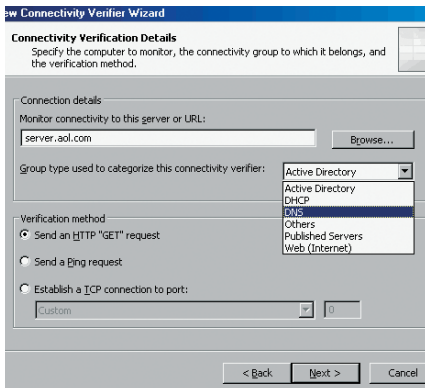
33. Eseménynapló-lekérdezés

A real-time naplónézegetés mellett van lehetőségünk az események utáni utólagos lekérdezésre is. Szűrhetünk kondíciókra (protokol, a kliens IP-címe, HTTP-utasítás stb.), időpontra, illetve intervallumra és értékre. Több, egyszerre zajló lekérdezés is lehet az alapja a vizsgálatnak, illetve a lekérdezések paramétereit elmenthetjük és vissza is tölthetjük.

34. Connection Verifiers

Ez egy különös komponens, amely kicsit ki- lóg a sorból – tudniillik nem sok köze van az ISA Server funkcióihoz, ám hasznos lehet akkor, ha ellenőrizni szeretnénk az ISA vagy bármelyik másik kiszolgálónk állapotát. A Connection Verifiers ugyanis az általunk felvett paraméterek alapján periodikusan képes az adott szerver funkciójától függően megvizsgálni az állapotát, majd hiba esetén (akár e-mailben) jelezni az üzemeltetőnek.

Lehetséges vele pinget, GET utasítást vagy tetszőleges porton TCP-kapcsolatot (pl. egy LDAP-lekérdezést) kezdeményezni, így tesz-



A CV beállítása

telve a DNS, AD, DHCP vagy más funkciót betöltő szerverünket.

35. Jelentések és publikálásuk

Látványos, HTML formátumú jelentéseket készíthetünk az ISA Server 2004 által gyűjtött adatokból. Lehetséges azonnali jelentést kérni, vagy időzítve gyűjtögetni az adatokat, majd összegzések formájában lekérni, persze mindezt teljesen automatikusan. A kategóriák a következők:

- internethasználat és a látogatott oldalak (felhasználók);
- a legtöbbet használt protokollok és alkalmazások;
- általános forgalmi adatok;
- a gyorsítótár találati és származási értékei.

36. Értesítések és publikálás a jelentésekkel kapcsolatban

Az elkészített jelentéseket automatikusan publikálhatjuk helyben vagy például egy megosztott hálózati mappába is. Ezt a mappát azután például egy webszerver „alá”, virtuális mappaként is elhelyezhetjük, így a böngészőből is tallózhatják a forgalmi adatokat az arra kiválasztott személyek. A jelentések elkészüléséről e-mailben is kérhetünk tájékoztatást.

37. Az összegző jelentés időpontja

Az ISA 2000-ben problematikus volt a „beégetett”, nem változtatható időpont (kötelezően 0.30 volt ez az érték) az összegző jelentések elkészülése kapcsán. Ez megszűnt, az ISA 2004-ben már tetszőleges és így életsze-

rű időpontban zárható összegző jelentéseket kérhetünk.

38. Naplózás

A naplózás fontosságát nem kell hangsúlyozni, azonban a működésén kívül nem mindegy az sem, hogy mennyire sikerül átlátható, használható és reális körülmények között tárolható naplóállományokat generálnunk. Az ISA 2004 háromféle formátumot ismer:

- Microsoft Data Engine (MSDE);
- SQL-adatbázis;
- hagyományos, szöveges állomány.

Az MSDE azért emelkedik ki a felsorolásból, mert ez az alapszükséglete például a real-time monitorozásnak, valamint azért is, mert ez a legegyszerűbb formátum, ami mégis sok mindenre képes. Az MSDE alapesetben korlátos, 2 gigabájt a méretkorlát, azonban az ISA automatikusan képes új állományt kreálni, ha elérjük ezt a limitet. Ez azért fontos, mert a naplónézegető csak az aktuális állományba képes betekinteni.

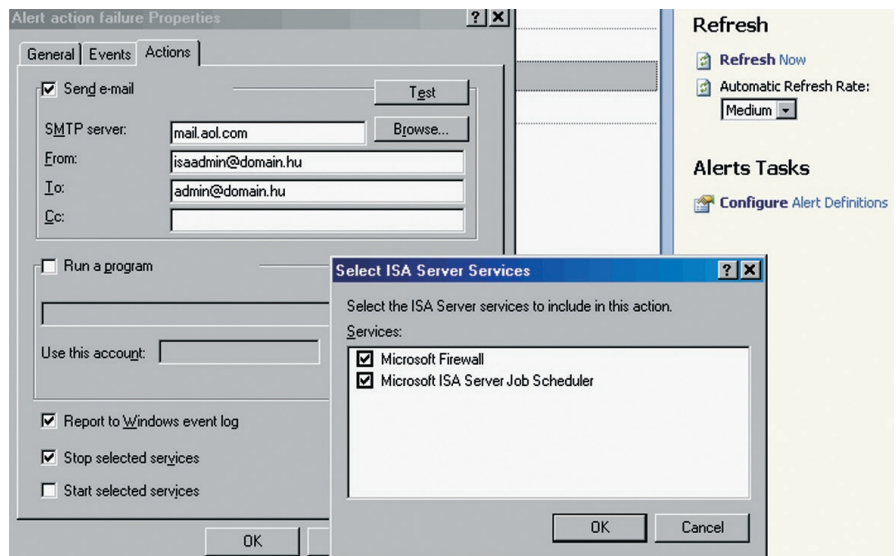
nens csak ebbe a formátumba képes dolgozni, MSDE-be nem.

39. Import/Export

Nagyon kellemes meglepetés, hogy az ISA 2004-ben igazán széleskörűen használhatjuk az export/import funkciókat. Akár tűzfal-szabályok esetén egyesével, vagy többet egyszerre, akár a különböző objektumokat, vagy akár az egész konfigurációt is exportálhatjuk, majd egy másik gépen gond nélkül importálhatjuk az .xml formátumba mentett beállításokat.

40. Lockdown-üzemmód

Ha valamilyen okból, akár véletlenül, akár szándékosan (ez utóbbit mi is elérhetjük például egy eseményre reagálva > Alerts fül a Monitoring szakaszban) leáll az ISA szerverünk valamelyik vagy esetleg az összes szervere, akkor az ISA egy speciális „Lockdown” módba kerül. Ilyenkor gyakorlatilag az ISA lezár mindent, kivéve például az ISA Admin



Itt állíthatjuk be az ISA-t egy riasztás apropóján

Szintén megoldás a naplózás az SQL-be, alapesetben viszont nem helyi SQL esetén tiltva van, vagyis a System Policyban engedélyeznünk kell az „Allow remote logging using NetBIOS transport to trusted servers” szabályt. A hagyományos állománytípus az ismert W3C vagy az ISA saját, szintén szöveges formátumába képes menteni.

Ha az SMTP-alkalmazásszűrésnél fontos Message Screenert használjuk, egy ilyen állományunk biztosan lesz, hiszen ez a kompo-

csoporthoz számukra a távoli elérés lehetőségét, vagy például a pingelést, illetve a DHCP-kliensként való működést. Ez egyúttal azt is jelenti, hogy az ISA 2004 hatásait ezzel a módszerrel nem fogjuk tudni semlegesíteni, sőt igazából erre csak az eltávolító telepítés nyújt megoldást. Vállalati, nagy hálózatokba tervezett, megbízható tűzfalként ez így is van rendjén.

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

MICROSOFT VIRTUAL SERVER II.

A virtualizáció hatékony eszköz, de csak alapos tervezési munka után. Ezúttal abból adunk ízelítőt, milyen kérdéseket kell megválaszolni, amikor a hardvererőforrásokat tervezzük majdani rendszerünkhöz.

Előjáróban el kell mondani, hogy a Microsoft Virtual Server R2 (VSR2) tesztcelű telepítése semmiféle komolyabb tervezést, előképzettséget nem igényel. Átlagos rendszergazdai ismeretekkel gyerekjáték a feladat, a próbálgatás azonnal kezdődhet, és ajánlott is. Ha azonban éles környezetben szeretnénk bevetni a VSR2-t, kellő megfontoltságra és előrelátásra van szükség. A továbbiakban kizárólag az „éles” környezetekkel foglalkozunk.

Megbízhatóság, amennyire csak lehetséges

A Virtual Server használata azt jelenti, hogy több kiszolgáló – bármilyen operációs rendszer fut is rajta – egyetlen fizikai hardveren osztozik majd. E hardver kiesése valamennyi vendégrendszer leállítását jelenti, tehát a VSR2 (és bármely más szerverparticionáló technológia) önmagában egyponstos meghibásodási lehetőséget épít a rendszerünkhöz.

Azoknál a szervezeteknél, ahol korábban sem használtak a kiesések elkerülését lehetővé tevő szolgáltatásokat (például redundáns tápegységet), ez nem túl jó hír. Nekik ugyanis legalább egyszer meg kell vásárolniuk ezeket a funkciókat. Ahol azonban eddig minden kiszolgáló tartalmazott némi védelmet, ez nem jelent majd problémát, sőt. A majdani kiszolgálót ugyanis csak egyszer kell megvásárolni, és egyszer kell ellátni védelmi megoldásokkal.

Nézzük, hogyan csökkenthetjük a szerverleállás valószínűségét! Nem nagy ördögösség megvizsgálni, hogy a leendő eszközünk képes-e több tápegység fogadására – tegyük ezt követelménnyé. Szintén megkettőzhetők a hálózati kártyák. A szerver saját lemezei – amennyiben ilyeneket használunk, és nem SAN-ról bootolunk – legyenek menet közben cserélhetőek, és legyen belőlük legalább kettő. Célszerű az SCSI használata, és érdemes őket RAID 1-be konfigurálni. Nagyon nagy teljesítményigény esetén választható akár négy lemez és RAID 0+1 tömbkonfiguráció is, ekkor azonban már a virtuális lemezeket valószínűleg nem ezen a tömbön tároljuk majd, hanem valamilyen SAN- vagy iSCSI-eszközön. A memóriamodulok vonatkozásában az ECC-változatok vásárlása javasolható. Ha Fibre Channelen vagy iSCSI-n keresztül SAN-tárolóhoz csatlakozunk, a kiszolgálók és a tárolóeszközök közötti adatútvonal is legyen kettőzött. Végezetül nem árt, ha a teljes távoli menedzsment a kiszolgáló hardver beépített eleme, vagy a gyártó által támogatott külön kártya ezt lehetővé teszi.

Senkit se riasszon el a fenti felsorolás! Egyrészt fontos hangsúlyozni a funkciók egyszeri – és nem hatszori, tízszeri – jelenlétét. „Rendes körülmények között” ezek a szolgáltatások amúgy is minden (vagy legalábbis a legtöbb) szerver részét képezik. Könnyebb eladni valamit egyszer, mint hatszor. Másrészt azt is el kell mondani, hogy a fenti kívánalmak csupán éssze-

rű biztonságot jelentenek, és nem mindenek feletti biztonságot. A fenti beépített funkciók a hétköznapi használat esetén „elég jó” – és nem „teljes” – megbízhatóságot adnak a rendszernek. Ugyanis az sem lehetetlen. Már a TechNet Magazinban is szó esett (a fűrtök kapcsán) a Stratus cég egyedi megoldásáról, amellyel lehetséges a kiszolgáló bármely elemét leállítás nélkül kicserélni, a Unisys ES7000-es családjáról nem is beszélve. A „hétköznapi” redundanciát alkalmazva tehát egyfajta arany középutat követünk. Ha a hüvelykujj-szabályok nem elegendőek, végezzünk megtérülési számításokat.

Miután a megfelelő biztonságról gondoskodtunk, nézzük, hogyan méretezzük az egyes hardverkomponenseket!

Processzorok

A processzorokról tudjuk, hogy a virtuális gépekbe csak egy virtuális „dughatunk”. A gazdagépben azonban jó, ha több is van. Megező rendszerek egyetlen hardverre konszolidálásánál végezhetünk előre méréseket a processzorok kihasználtságáról, ez némi támpontot adhat a majdani terhelésről. Vegyük figyelembe azonban, hogy a szokásos migráció esetén általában gyengébb processzorokról mozgatjuk át a rendszert kevesebb, de nagyobb teljesítményűre. Gyakori, hogy az egyes szerverek processzor kihasználtsága nagyon alacsony. Ha tíz szerver átlagos terhelé-

se 5 százalék, egyetlen processzort is csak 50 százalékban terhelnének. Az „átlagos” itt nagyon fontos szó, mert lehetséges, hogy a tizből háromnál a hó végi terhelés napközben esetleg átlagosan 50 százalék, így máris látzik, hogy egyetlen processzor nem elegendő. Az átlagos terhelés mellett legyünk tehát tisztában azzal is, hogy bizonyos időszakokban hogyan alakul a processzorkihasználtság, s ez alapján szükség esetén módosítsuk az eredeti elképzeléseinket.

A majdani processzor-terhelés elosztásában a Virtual Server is segítséget nyújt. Módunkban áll relatív súlyt megállapítani (a többi kiszolgálóhoz képest), és megadhatjuk, hogy legfeljebb vagy legalább mekkora processzoridőt adunk.

Amikor meghatározzuk a processzorok számát, a korábbi átlagos terheltség és a speciális időszakok mellett ne hanyagoljuk el a virtualizációs technológiából adódó többletterhelést és az esetleges terheléscsökkenést sem. Tíz gépünk helyett most tizenegy lesz – ha számítjuk a gazdagépünket is, és a szoftveresen megvalósított hardverelemek is mind-mind a processzort terhelik. A többlet-

rekéből, amelyeket most virtualizáltunk, a kiszolgálók közötti titkosítás kikapcsolásával komoly processzoridő-megtakarítást is elérhetünk, miközben a biztonságunk nem csökken.

A Hyper-threading (HT) technológiát korábban nem szerette a Virtual Server. Nagyobb terhelés mellett kifejezetten teljesítménycsökkenést okozott a használata, ezért még az adminisztrátori kézikönyvben is a kikapcsolását javasolták. Az R2 megjelenésével ez a tétel már nem érvényes, a HT nem jelent visszahúzó tényezőt. Ezt úgy érték el a fejlesztők, hogy megkülönböztették a logikai és a fizikai processzort (egészen pontosan a processzormagot), a virtuális gépet reprezentáló programkód-szálat (thread) pedig a Windows-gazdarendszer kernelének scheduler komponense sohasem osztja ki logikai processzorra.

A HT-hez képest a többmagos processzorok sohasem jelentettek problémát. Fontos tudni, hogy a Virtual Server 32 processzort támogat, ez processzorfoglatot jelent. Ad absurdum elképzelhető (és a piacon kapható is!) olyan megoldás, ahol 32 darab kétmagos

processzor bekapcsolt HT-vel 128 darab logikai processzor-ként jelenik meg a gazdagép feladatkezelőjében (ami persze ilyenkor már nagyon „mókásan” néz ki). Függetlenül a logikai és fizikai processzorok számától, az aktív virtuális gépek száma legfeljebb 64 lehet a Virtual Server R2-ben.

Említsük meg, hogy a 64 bites processzorok, egészen pontosan az x64 architektúra támogatott platform. Ez, bár a processzorok esetében is jelent teljesítménynövekedést, igazán a memóriahasználatnál nyújt sok előnyt. A tervezéskor vegyük figyelembe, hogy a VSR2 vendégrendszerei csak 32 bites alkalmazások lehetnek, függetlenül attól, hogy a gazdagép 32 bites vagy 64 bites változat-e. Végezetül említsük meg, hogy az Intel és az AMD által a processzorokban megvalósított virtualizációra szolgáló kódot (IVT, AMD-V) a VSR2 jelenleg még nem támogatja. Május 11. óta azonban letölthető a majdani SP1 csomag első bétaváltozata, és ez már tartalmazza a támogatást.

Látszólag sem a 64 bit, sem az IVT/AMD-V nem játszik komoly szerepet a tervezés-

kor, hiszen a rendszer 32 biten működik IVT/AMD-V nélkül is. Csakhogy a Virtual Server következő generációs változata már beépül a majdani Windows-szerverbe, és csak 64 biten, kizárólag IVT/AMD-V-támogatás megléte esetén fog futni. Ha tehát ma kell eszközt vásárolnunk, ezt a tényt mindenképp vegyük figyelembe!

A hosszas fejtegetésből látható, hogy a „hány virtuális gépet visz el egyetlen processzor” kérdés nem válaszolható meg autentikusan. A 4:1 arány lehet kiindulási alap, szélsőséges esetekben azonban az 1:1 és a 8:1 is elképzelhető.

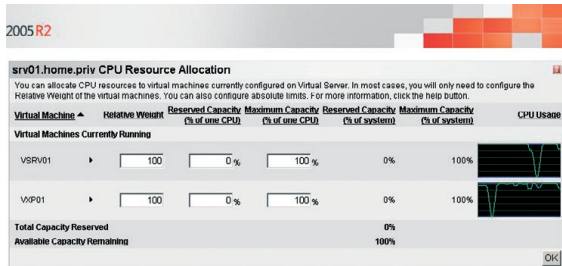
Memória

A processzorhoz hasonlóan a memória méretezésekor is több mindent meg kell fontolnunk.

Nem kétséges, hogy a Virtual Server – a vendégrendszereken keresztül – igen-igen memóriaiényes alkalmazás. Bár a gazdagép memóriáján osztoznak a vendéggépek, de egyébként induláskor minden virtuális gép valóban lefoglalja a neki megadott memóriamennyiséget, s ahhoz más virtuális gép nem férhet hozzá. A virtuális gépeknek megadott „fizikai” RAM nem lapozható ki, ezért aztán a futó virtuális gépek összes fizikai RAM-ja nem lehet több, mint a gazdagép által birtokolt fizikai RAM. Sőt: számolni kell minden virtuális gép esetén plusz 32 megabájttal és a gazdagép ki nem lapozható memóriájával is.

Ezeket a szabályokat, ha akarnánk sem tudnánk megszegni. Ha a Virtual Server azt tapasztalja, hogy egy elindítandó gép memóriája „túlfoglalná” a gazdagép tényleges memóriáját, akkor egyszerűen el sem indítja. (Szerencsénkre a történekről beszámol az eseménynaplóban.)

Mivel a virtuális gépek nem osztoznak a memórián, valódi memóriát kérnek és kapnak, egyáltalán nem mindegy, mennyit foglalunk nekik, továbbá, hogy a gazdagép hogyan kezeli a memóriát. A foglalást magunk határozhatjuk meg, de azért itt is meg van kötve a kezünk. Ha túl kevés fizikai memóriával rendelkezik egy vendégrendszer, akkor – akárcsak virtualizáció nélkül – a saját merevlemezét kezdi el használni, a lapozófájljához nyúl. Csakhogy virtualizációs környezetben ekkor már a közös IO-rendszer erőforrásait emésztí, ami azonnal visszahat a többi gép teljesítményére is. Mindebből az következik,



A processzor-erőforrás súlyozása

terhelés (virtualization overhead) nem hanyagolható el, de nem is katasztrofális: 5–15 százalék (persze a weben hitvallástól függően sok mindent lehet olvasni). Mindezzel óvatosan kell bánni. A virtuális hardver megvalósítása tényleg okoz némi többletterhelést, de érhetnek kellemes meglepetések is. Ha például intenzíven használjuk a belső, virtuális hálózatot, akkor az terheli ugyan a processzort – kvázi virtuális többletterhelést okoz –, ugyanakkor a szokásos Ethernet-hálózatnál jóval nagyobb sávszélességhez jutunk a kiszolgálók között, tehát a többlet-processzoridő hasznosult: sokkal gyorsabb hálózatot kapunk.

Ha korábban egy front-end-back-end szer-
verkapcsolatot hoztunk létre fizikai szerve-

hogy semmit sem – vagy legalábbis nem sokat – tudunk takarékoskodni a memóriával a fizikai környezetbe képest.

Térjünk vissza a gazdagép memóriakezeléséhez! Egy 32 bites környezetben, Windows Server Enterprise Edition R2-t feltételezve 4 gigabájt felett csak a /PAE kapcsolóval használhatunk memóriát. A 64 bites változathoz képest ez mindenképpen teljesítményvesztést jelent, ezért ha választhatunk, akkor 64 biten kezdjük építeni a VSR2-környezetünket, ott nem fogunk memóriakorlátokba ütközni, legalábbis a gazdagép oldaláról bizonyosan nem.

Sokprocesszoros és sok memóriát tartalmazó rendszerek esetén nézzük meg a vásárláskor, hogy az adott eszköz támogatja-e a NUMA szabványt. NUMA esetén több memória-alrendszer is működik ugyanabban a kiszolgálóban, ezzel biztosítva, hogy a teljesítménydegradáció jóval kisebb a processzorok számának növekedésével, vagyis erőteljesebb a teljes rendszer skálázhatósága.

IO-alrendszer, lemezek, hálózat

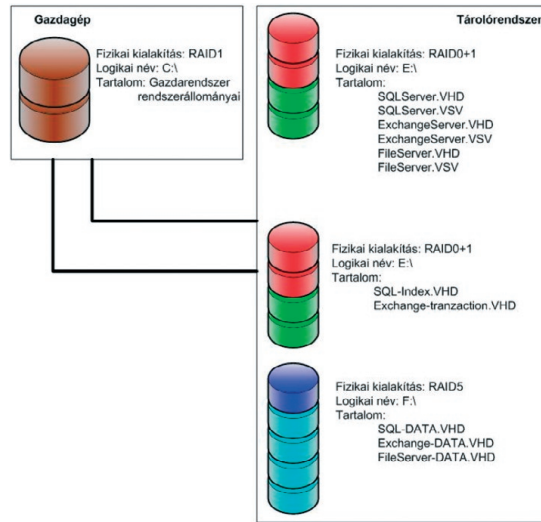
A processzor olyan erőforrás, amelyet egyúttal és elkülönülve is használhatnak a virtuális gépek. A memória teljesen szeparált komponens. Az IO-rendszer viszont teljes egészében közös a virtuális gépek számára. Ugyanazon a host bus adapteren vagy CD-ROM-on kell osztozniuk a virtuális gépeknek. Különösen fontos tehát, hogy jól becsüljük meg a várható IO-terhelést. Szerencsére ez sokkal jobban tervezhető, mint a processzorhasználat, mivel azonos a korábbi fizikai és a majdani virtuális IO-karakterisztika.

A „legközösebb” és „leglassabb” címkéket egyaránt begyűjtő alrendszer tervezésekor mindenképp igyekezzünk „főlélni”. Válasszuk külön a gazdagép rendszerállományait és a vendégrendszerek rendszerdiszkeket reprezentáló fájlokat. Használjunk gyors RAID 0+1 tömböt az utóbbiak számára. Ha SAN-unk van, akár iSCSI, akár FC, ajánlatos egy nagyobb RAID 0+1 azoknak a virtuális lemezeknek, amelyek sebességet igényelnek (indexállományok, tranzakció-logok), valamint RAID 5-öt a kevésbé intenzív használatú lemezekhez.

A virtuális lemezek elhelyezése mellett a tervek készítésekor gondot kell fordítani a lemezek kapacitásának méretezésére is. Látszólag itt nincs is teendő: új rendszerrel

ezt ugyanúgy kell elvégezni, mint a fizikai szervereknél, migrációnál pedig a migrált rendszereket kell összeadni. Nos, ez tévedés. Számoljunk még a következő igényekkel:

- minden virtuális gép számára alokált memóriamennyiségnek megfelelő tárhely az állapotmentésekhez (Save state);
- tárhely az UNDO és differenciális lemez-fájlok számára (ezek olyan állományok,



Egy lehetséges lemezkiosztás

amelyek kikapcsoláskor eldobhatók vagy összemásolhatók az eredeti lemezzel, kezelésükről később még külön szólnunk);

- többletkapacitás az úgynevezett mester-rendszerpéldányokhoz (ezek előre elkészített, „kipreparált” operációs rendszerek, hogy a virtuális gépek létrehozása csupán néhány fájl másolásából álljon);
- többletkapacitás ISO image-állományok számára (a mindennapi használat során tapasztalni fogjuk, hogy a tényleges CD-k – DVD-k – helyett gyorsabb és kényelmesebb ISO image-eket használni).

Hálózat

A cikk elején említettük, hogy érdemes redundáns hálózati kártyákban gondolkodni. A teljes igazsághoz tartozik, hogy az úgynevezett Network teaminget a VSR2 jelenleg nem támogatja, mert a kliens-operációsrendszerek hálózati kapcsolatainak elvesztése gyakran ehhez a funkcióhoz kötődik. Hosszabb távon azonban biztosan működni fog ez a funkció, érdemes már most tervezni.

Az nem kérdés, hogy a hálózati kártyáknak célszerűen gigabites sávszélességgel kell

rendelkezniük. Sokkal inkább kérdés, hogy vajon hány fizikai kártyát használunk majd. Ez attól függ, hány olyan fizikai subnetünk van, amelyekhez egy vagy több virtuális gépeknek kell csatlakoznia. Ha például back-to-back tűzfalkonfigurációt tervezünk, akkor célszerű legalább két kártyát használni, de ha a DMZ kiszolgálói valóságos gépek, akkor szükségünk lesz egy harmadikra is.

A hálózat tervezésekor vegyük figyelembe, hogy a belső, virtuális gépek közötti forgalom nem érinti a valóságos hálózatot, ezért az nem Ethernet, hanem processzor kérdése.

Kritikus probléma lehet a mentés tervezése. Mentenünk kell a virtuális gépek „belsejét” hálózaton keresztül. Az adatmennyiséget meg kell becsülnünk, ki kell számítanunk a másoláshoz szükséges időt. Úgy ütemezzük a mentéseket, hogy azok lehetőleg ne egyszerre induljanak, így a hálózat mellett az IO-rendszert is kíméljük.

Emellett, ha a host nem rendelkezik saját mentőegységgel,

és az sem iSCSI-n, sem pedig FC-n nem érhető el, akkor a gazdagép mentése, szintén hálózaton keresztül zajlik majd. Ez még kisebb implementációk esetén is hosszadalmas lehet, végezzünk tehát becsléseket a mentési ablak, valamint az elvárt visszaállítási idő ismeretében.

Monitorozás, monitorozás, monitorozás

Ha nem is sikerül(hetet) valamennyi kérdést feltárni, az összegyűjtött ismeretek alapján egy fontos következtetést bátran lezűrhetünk: jól működő virtualizáció birtokába csak akkor kerülünk, ha az alapos tervezés mellett mind az eredeti (ha volt ilyen), mind pedig a virtualizált rendszerünk teljesítménymutatóit ismerjük, méghozzá alaposan. Ha ilyen ismeretekkel nem rendelkezünk, akkor vagy sok pénzt költünk el a feleslegesen fölé méretezett hardverre, vagy erőteljes teljesítménycsökkenést fogunk tapasztalni anélkül, hogy a pontos okokkal tisztában lennénk.

Jó tervezetést!

Lepénye Tamás
(lepenyet@mal.hu) rendszermérnök (MCSE)

PROBLÉMÁK A „MOVE MAILBOX” UTÁN

Az Exchange Server 2003 SP2 esetében meglehetősen gyakran találtak a Microsoft terméktámogató mérnökei az alábbi problémával az elmúlt pár hónap folyamán. A jelenségre még nem érhető el hivatalos megoldás, de viszonylag egyszerűen elkerülhető, ha már eleve számítunk rá.

Ha mailboxokat mozgatunk egy olyan Exchange 2003 szerverre, amelyre korábban már installáltuk az SP2-t, kellemetlen meglepetések érhetnek minket: a felhasználók egy része nem fog tudni bejelentkezni a mailboxába Outlookon keresztül. A hibáról mindössze egy pop-up ablak ad értesítést, azzal az üzenettel, hogy „stack overflow”.

Szimptómák

Sorozatos tesztelesek tapasztalatai szerint ha több száz mailboxot mozgatunk, előfordul, hogy egyetlen postafiók sem károsodik, azonban az a jellemző, hogy 10 százalékuk a mozgatást követően a következő szimptómák valamelyikét mutatja:

- Outlookkal nem lehet bejelentkezni a mailboxba, csak OWA-n keresztül;
- a postafiókon belül más felhasználóknak adott jogosultságok érvényüket veszítik;
- Outlookban duplán jelennek meg az alapmappák (Inbox, Sent items stb.);
- a felhasználók problémákat észlelnek a naptár (Calendar) használatkor: nem képesek új találkozót (appointment) létrehozni, nem tudják más felhasználók F/B-információit lekérdezni;
- problémák merülnek fel – pop-up ablakok által tolmácsolva – az Offline Address Book használatakor.

A jelenlegi állapot szerint a Microsoft izolálta a probléma forrását, és a post-SP2 fix (Exchange 2003-hoz) már készen áll, e cikk nyomdába adásakor az utolsó tesztelesek folynak. A tesztek befejezéséig az alábbi workaroundtal lehet elkerülni a problémát.

Kikapcsolható funkcionális

A hiba forrása a GRM (Guid-Replicid Mapping) cache, amely az SP2-ben – az SP1-gyel szemben – alapértelmezésben be van kapcsolva. Feladata a „move mailbox” során felmerülő teljesítményproblémák elkerülése (e teljesítményproblémák kialakulásának lehetősége azonban nagyon valószínűtlen).

A HKLM\System\CurrentControlSet\Services\MSEExchangeIS registry-kulcs segítségével – amelyen keresztül a GRM cache-t állíthatjuk – mi magunk is kikapcsolhatjuk az említett funkcionális. A registry-kulcs alatt található egy, a szerver nevét viselő alkulcs, és ez alatt a szerveren található valamennyi Store-hoz találunk egy újabb kulcsot.

Minden Store-nál, ahol ki akarjuk kapcsolni a GRM cache-t, vagyis amely Store-ba szeretnénk mozgatni mailboxokat, létre kell hozni egy bejegyzést DWORD adattípussal méghozzá,

„Guid-Replicid Caching” néven és nullát adva értékül.

A fenti lépések után mountoljuk újra a Store-t, vagy indítsuk újra a Store szolgáltatást. A cache kikapcsolásával folytatni lehet a migrálást a probléma ismételt felbukkanásának kockázata nélkül. Megjegyzés: a cache kikapcsolása nem lesz pozitív hatással azokra a mailboxokra, amelyeket már érintett az itt ismertetett probléma; segítségével csak a későbbi mailbox-mozgatások során tudjuk elkerülni a probléma ismételt felbukkanását. A már érintett postafiókok javítására nincs univerzális megoldás, hiszen még a szimptómák sem mindig azonosak.

Ismételt létrehozás

A sérült mailboxok nagy része ismét használhatóvá tehető, ha újra-mountoljuk a Store-t, vagy újraindítjuk a Store szolgáltatást, illetve egy ismételt mailbox-mozgatással egy másik Store-ba mozgatjuk őket (utána persze ajánlott visszamozgatni őket az eredeti Store-ba).

Ha ez nem segít, akkor – javítandó a Store logikai hibát – futtassuk le az „isinteg -s <servername> -fix -test alltests” parancsot.

Ha ezután még mindig van olyan mailbox, amelybe Outlookkal nem tudunk bejelentkezni, nagy valószínűség szerint csak a postaláda ismételt létrehozása segíthet.

Somogyi Zoltán

(zsomogyi@microsoft.com) Microsoft Magyarország

AZ OFFICE 2003 BEVEZETÉSE

A TechNet Magazin már eddig is sokat foglalkozott az asztali operációs rendszerek és szervertermékek bevezetésével, azonban nem esett szó a vállalatoknál ugyancsak gyakori Office-csomagok bevezetését megkönnyítő technológiákról.

Avállalatok számítógépeire az Office általában az alábbi módok valamelyikén került:
1. A rendszergazda feltelepítette az Office-csomagot cd-ről vagy megosztásról, végigment a varázsló lépésein, majd elvégezte az első alkalommal szükséges beállításokat; ennek a műveletnek a kimenete egy képfájl volt, amelyet ráhúztak az ügyfélgépre.

2. Az Office MSI csomagját a csoportházirendek segítségével távolról telepítették a kliensekre, a felhasználó pedig elvégezte az első alkalommal szükséges beállításokat.

Mind a két lehetőségnek megvannak az előnyei és hátrányai is. Az első verziónál (majdnem) mindent beállíthatunk, ahogy tetszik, ám a konfiguráció hardverfüggő – ahogy az az image-eknél lenni szokott –, míg a második lehetőségénél semmilyen módon sem tudjuk befolyásolni az Office-termékek működését. (Gondoljunk akár az Outlook-Exchange Server beállításra vagy a vállalati dokumentumsablonok telepítésére.)

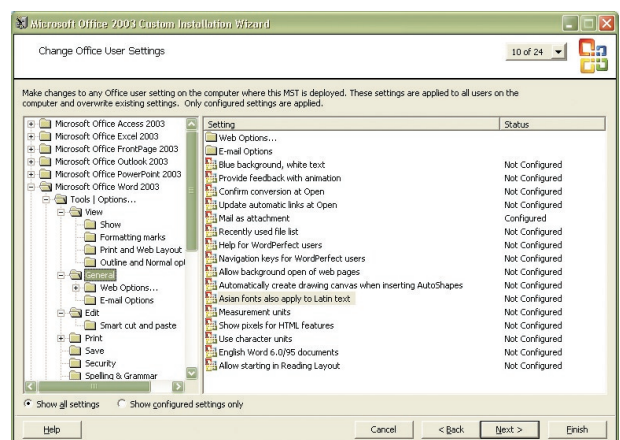
Most nézzük meg, hogyan tudjuk ezeket a lépéseket automatizálni, illetve hatékonyabbá tenni!

Az Office telepítőjének működése

Mielőtt elkezdenénk vadul telepígetni, szánjunk egy kis időt a telepítési folyamat menetének és lehetőségeinek megismerésére. Az Office 2003 telepítésének folyamata szinte semmit sem változott az Office 2000-hez vagy az Office XP-hez képest. A termék azóta is egy vagy több Windows Installer-csomagként települ, és ezekre különböző transformokat „ráhúzva” tudjuk befolyásolni a telepítés menetét. Sok egyéb bevezetési eszköz mellett az Office Resource Kitben található Custom Installation Wizard segít a transformfájlok létrehozásában. Hálózati környezetben a leghasznosabb, ha a telepítő /a kapcsolójának segítségével létrehozunk egy úgynevezett adminisztratív telepítési pontot: ez egy olyan megosztott mappa, ahol az Office-csomagok telepítőkészletén és az esetleges frissítéseken kívül a transformfájlok is megtalálhatók. Ennek a módszernek az előnye egyrészt (DFS közbeiktatásával) a centralizált telepítés, másrészt a patchek telepítésének egyszerűsítése. (Gondoljunk csak bele, mennyivel könnyebb a WSUS dolga – hálózati forgalmat is tekintve –, ha nem egy SP0-s, patchmentes telepítést kell frissítenie.)

Az Office 2003 telepítőjének figyelemre méltó újdonsága egy olyan alapértelmezett gyorsítótár a %SYSTEMROOT%\MSOCache alatt, amelyik a telepítőkészlet másolatát tárolja.

Ez egyrészt segít a felhasználónak, ha egy olyan szolgáltatást (például idegen nyelvű helyesírás-ellenőrzőt) telepít, ami addig nem volt aktív, másrészt a WSUS-t is támogatja patchek telepítésénél, mivel nem terheli feleslegesen a hálózatot. Amennyiben egyes ügyfélgépeinken nincs „felesleges” 400 megabájtnak, természetesen kikapcsolhatjuk



A legapróbb beállításokat is módosíthatjuk

a cache létrehozását már az adminisztratív telepítési pont elkészítésénél, de akár a transform is befolyásolhatja ezt a működést.

A Resource Kit eszközei

Vágyunk bele a gyakorlati részbe!

A <http://www.microsoft.com/office/orarchive/2003ddl.htm> címről töltsük le az „Of-

Office 2003 Editions Resource Kit"-et. Ez a csomag a következőket telepíti a gépünkre:

Custom Maintenance Wizard (CMW). Ezzel az eszközzel úgynevezett CMW-transzformokat hozhatunk létre. A CMW-transzformfájlok a már telepített és meglévő Office-csomagok „átplasztikázására” használhatók, segítségével lehet egységesíteni az Office-beállításokat minden ügyfélgépen.

CMW File Viewer. A már elkészített CMW-fájlok tartalmának megtekintésére használjuk.

Custom Installation Wizard (CIW). MSI-transzformok létrehozására szolgál. Ezt akkor használjuk, ha az új telepítéseket szeretnénk testreszabni.

MST File Viewer. A CIW által készített transzformok megtekintése.

Profile Wizard (OPS). Egy már telepített Office-ról készít snapshotot, majd ezt transzformként a CIW vagy a CMW használatával húzhatjuk rá a kliensekre.

OPS File Viewer. A Profile Wizard kimenetének megtekintése.

Mit, mikor?	Új számítógép	Létező számítógép
„Tiszta lap”	Custom Installation Wizard	Custom Maintenance Wizard
Már létező beállítások átvétele	OPS + CIW	OPS + CMW

Tegyük fel, hogy egy vállalatnál most vezetjük be az Office-t, ennek megfelelően az Office 2003 Professional „tisztá” telepítését szeretnénk testreszabni. A fentiek alapján indítsuk el a Custom Installation Wizardot. Elsőként meg kell majd adnunk, hogy melyik Office-csomagot szeretnénk testreszabni, irányítsuk a varázslót a pro11.msi fájlra. Ezután a varázsló kiolvassa, hogy miket lehet „beállítani” a csomagon. (Például ha egy FrontPage-telepítőkészletet akarunk testreszabni, akkor logikus, hogy, mondjuk, az Outlook e-mail-beállításait nem tudjuk szabályozni.)

Mínezek után egy 24 lépéses varázslóban a létező legapróbb részletekig mindent beállíthatunk, de persze semmit sem kötelező, bármelyik beállításra mondhatjuk, hogy maradjon az alapértelmezett. (Felhasználó és cégnév, alkalmazásbeállítások, e-mail-fiókok, MSOCache be- és kikapcsolása, Volume License-termékkulcs stb.)

Ha végiglépeltünk a varázslón, a kimenetünk egy darab .mst kiterjesztésű fájl lesz. Telepítés során ez a fájl utasítja a Windows Installert a beállítások végrehajtására. A CIW-varázslót annyiszor kell igénybe venni, ahány különböző konfigurációt szeretnénk használni (más-más beállítások például szervezeti egységenként.)

Hogyan tovább?

A transzformos telepítéshez a telepítőt a következőképpen kell futtatni:

```
„setup.exe TRANSFORMS=“\\Szerver\AIPShare\Transform-Name.MST” /qb-“
```

Ettől a ponttól kezdve a módosított telepítőt akár az SMS, akár a szoftvertelepítési Group Policy is meghívhatja és használhatja.

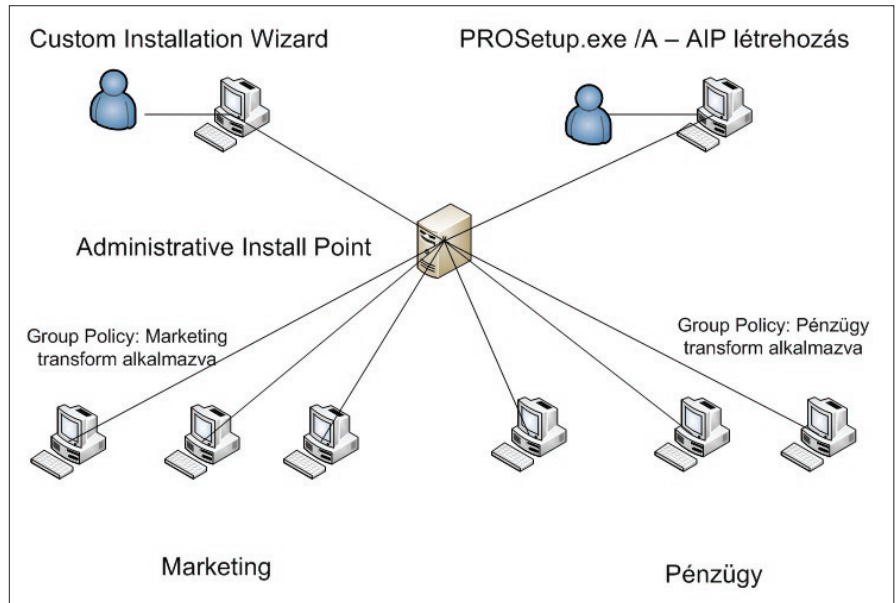
Ezzel azonban nem értek véget a feladataink, létre kell hoznunk az adminisztratív telepítési pontot. Ehhez egy mennyiségi li-

fussanak a hálózatról. (Figyeljünk a hálózati forgalomra, nehogy a fontos funkciókat helyezzük ki a szerverre.)

Ha létrehoztuk az adminisztratív telepítési pontot, nincs más dolgunk, mint a fenti parancssorral elindítani a telepítéseket.

Szegény ember (is) vízzel főz?!

Ha nem használunk SMS-szervert (vagy használunk, de mégis...), problémát okozhat, hogy az Office telepítése rendszergazdai jogkörhöz kötött. Az Office 2003 nyújt „csontváz-telepítési” lehetőséget is, ekkor egy rendszergazda jogkörű felhasználó az Office telepítési adatait bejuttatja a Windows Installer szolgáltatás adatbázisába, és a felhasználó számára elérhetőek lesznek az Office-alkalmazások parancsikonjai. Telepítésre azonban egészen addig nem kerül sor, amíg a felhasználó első alkalommal rá nem kattint valamelyikre. Ekkor a rendszergazdai jogkörrel futó Windows Installer szolgáltatás felte-



Office-bevezetési folyamat

cences telepítőt kell lefuttatnunk a /a kapcsolóval, amely a kért megosztott mappába másolja a fájlokat. Ezek után nemes egyszerűséggel ezt a telepítőkészletet is lehetőségünk lesz patchelni, így az új ügyfelek már a patchelt verziót telepítik gépükre. Ugyancsak előny, hogy amennyiben ügyfélgépeink nem rendelkeznek megfelelő tárterülettel, megtehetjük, hogy a CIW használatával létrehozott transzformban kikötjük: egyes funkciók (például többnyelvű elválasztás a Wordben)

lepti az elindított összetevőt az adminisztratív telepítési pontról, ezzel is sok lemezhelyet takarítva meg számunkra. A csontváz-telepítéshez így kell futtatni a telepítőt:

```
“\\szervernev\AIPShare\setup.exe /jm pro11.msi /t office.mst”
```

(Utóbbi csak akkor, ha a CIW segítségével létrehoztunk egy transzformot.)

Moldova György
(v-gyomol@microsoft.com) Microsoft Magyarország

KÖZELEBBRŐL AZ XML PAPER SPECIFICATION

Napjaink egyre fontosabb kérdése, hogy az elektronikus dokumentumok – és ezzel együtt a szellemi termékek: szerződések, jelentések, specifikációk és egyéb munkaanyagok jóváhagyott és hivatalos elektronikus példányai – milyen formátumban lássanak napvilágot.

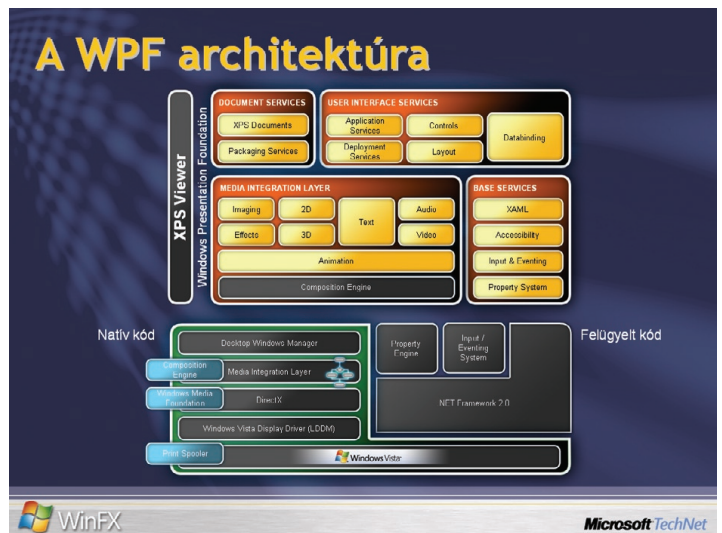
Számtalan elektronikus formátum és informatikai folyamatrendszer létezik már az elektronikus adatcsere megvalósítására, azonban egyik sem tökéletes. A mai szoftverekkel még gyakran találkozhatunk olyan helyzetekkel, hogy a dokumentumok az egyik gépen másképp jelennek meg, mint a másikon, kinyomtatva pedig szintén lényegesen eltérő formátumot kapunk. Mindez függ például az adott gépre telepített driverektől, betűtípusoktól, nyomtatóktól. Egyes formátumok (például a PDF) már kínálnak erre megoldást. Egy másik jelentős felhasználási terület a dokumentumtartalom automatizált kezelésének igénye, ebben a legnagyobb segítséget az XML alapú formátumok adják, például az Office XML dokumentumformátumai vagy éppen a nyílt forrású ODT formátum.

Az XPS bemutatása

Az XPS (XML Paper Specification) – hasonlóan a Windows Vistával érkező új generációs platform valamennyi eleméhez – egyetlen új programozási modellben igyekszik egyesíteni a meglévő megoldások előnyeit: jelen esetben az XPS dokumentumformátumban. Az XPS-dokumentumok képesek megőrizni saját fizikai megjelenésüket, így a környezettől függetlenül mindig pontosan ugyanazt az eredményt kapjuk akár monitoron, akár nyomtatón. A formátum képes részt venni munkafolyamatokban is, valamint XML alapokon tárolt tartalma tetszőlegesen felhasználható, manipulálható az általunk írt programok segítségével is. Az XPS mindemellett nyitott és platformfüggetlen.

Az XML alapú felhasználást azért érdemes kiemelni, mert – a megoldást a klasszikus, többretegű alkalmazásfejlesztéshez hasonlítva – ebben a dokumentumformátumban az adat és a hozzá tartozó megjelenítési réteg teljes mértékben el van választva egymástól, a dokumentumban lévő adatok egyszerűen, XML alapokon kezelhetők, újrahasználhatók, illetve hozhatók létre.

Az XPS az újgenerációs megjelenítési architektúra, a Windows Presentation Foundation (WPF) egy komponense. A WPF egyesíti a különböző megjelenítési megoldások és formátumok teljes skáláját: a felhasználói felületet (User Interface Services), a különbö-



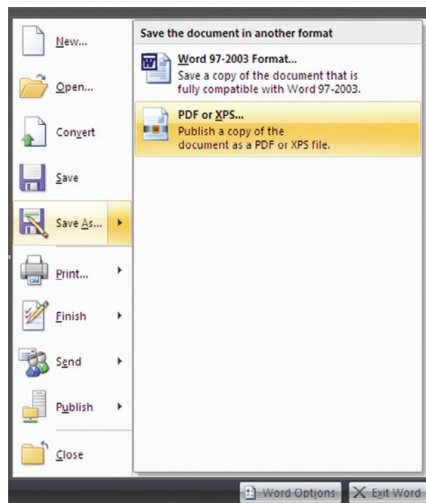
A Windows Presentation Foundation architektúrája

ző médiaformátumokat (Media Integration Layer) és a dokumentumkezelést (Document Services). Az XPS-dokumentumok kezelését

a Microsoft .Net keretrendszer és a WinFX (amit nemrég a .Net keretrendszer harmadik verziójának neveztek el) maximálisan támogatja, és ennek révén igen magas szinten kezelhető az XPS-struktúra minden eleme és funkciója.

XPS-támogatás Microsoft Office 2007 és Microsoft Windows Vista környezetekben

Az XPS formátumú dokumentumok előállítására a felhasználók számára igencsak egyszerű feladat lesz például a Microsoft Office 2007-ben is. Egy publikálható dokumentum esetén egyszerűen kiválasztjuk a „Mentés másként” funkciót, majd kimeneti formátumként az XPS formátumot. A létrejött dokumentumot ezek után megnyithatjuk az XPS-nézetével (XML Paper Specification Viewer), amely az Internet Explorer böngészőbe is beépül (csakúgy, mint például az Adobe Reader PDF-ek esetében). Ez a meg-

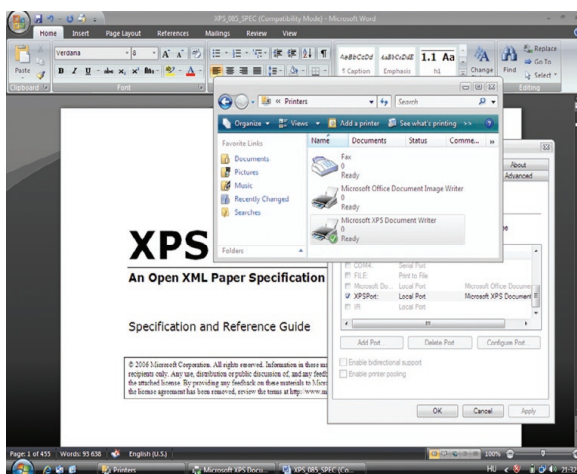


XPS formátum mentése Word 2007-ből, avagy mentés másként

jelenítésre szolgáló alkalmazás a WinFX részeként automatikusan települ, de külön is letölthetjük, használhatjuk magát az XPS Viewert, még akkor is, ha nincs az adott gépre telepítve a WinFX.

Az XPS-dokumentumok előállításának egy másik egyszerű módja, ha „rányomtunk” egy helyi XPS-nyomtatóra (Microsoft

XPS Document Writer). Ez gyakorlatilag ugyanazt eredményezi, mint ha fájlba nyomtatnánk. Hasonló megoldást már láthattunk



Nyomatás az XPS-nyomtatóra, XPS Port

a Microsoft Office 2003 használata esetén, amikor a nyomtatható és hordozható dokumentumok előállításra létrehozott lokális nyomtató dolgozta fel a nyomtatási sort, és állított elő egy MDI állományt – ez volt a Microsoft Office Document Image Writer. Itt azonban egy teljes értékű XPS-dokumentumot kapunk, ami nemcsak könnyen szállítható, de rendelkezik a formátum valamennyi itt ismertetett előnyével is.

Az XPS felépítése

Egy XPS-dokumentum tárolása „csomagokban” (package) történik, ami tulajdonképpen nem más, mint egy olyan tömörített ZIP

```
<FixedPage xmlns="http://schemas.microsoft.com/xps/2005/06" xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key" xml:lang="en-us" Width="800" Height="600">
  <Canvas>
    <Glyphs OriginX="1" OriginY="100"
      UnicodeString="Sample text" FontUri="Fonts/Tahoma.ttf" FontRenderingEmSize="20"
      Fill="#FF000000">
    </Glyphs>
  </Canvas>
</FixedPage>
```

1. ábra. Egyszerű XAML-szövegstruktúra – Glyphs elem használata

állomány, mely megfelel az Open Packaging Conventions szabályainak. E szabály definiálja a strukturáltan tárolt leíróadatok, adatkapcsolatok, fájlformátumok, digitális

alíráások tárolásának módját, és ez alapján jön létre az XPS-dokumentum logikai és fizikai struktúrája. A logikai szerkezet leírja az egyes oldalak tartalmát – XML dokumentum alapú oldalak –, míg a fizikai struktúra tárolja a megjelenéshez tartozó különböző komponenseket, amelyeket közös erőforrásként a logikai struktúrák használhatnak fel. Ennek a különleges felépítésnek köszönhetően az XPS-dokumentum a megjelenítéshez szükséges valamennyi komponenst képes eltárolni magában – például képeket, betűtípusokat, XML állományokat –, maximálisan támogatva a dokumentum hordozhatóságát.

Az XPS-dokumentumok tartalmának leírására szolgáló nyelv a WPF által használt XAML egy kisebb részhalmlaza. Maga az XPS specifikáció ennek segítségével pontosan ismerteti definíciós szinten a dokumentum megjelenítési elemkészletét – például polinomok, alakzatok, kitöltési módok, átlátszóság, minták, maszkolás, karakterek –, ezek

További információk

<http://msdn.microsoft.com/winfx/>
<http://www.microsoft.com/xps/>
<http://msdn.microsoft.com/msdnmag/issues/06/01/XMLPaperSpecification/>

segítségével egy olyan nyitott formátumot kapunk, amely heterogén rendszerek között is lehetővé teszi, hogy a dokumentumkép megegyező legyen. Arra is lehetőség van, hogy egy WPF-alkalmazás felületéből XPS-dokumentumot hozzunk létre, ilyenkor ki kerülnek a létrejött dokumentumból azok az XAML-elemek, amelyek nem részei az XPS oldalleíró nyelvnek. A fix oldalkép megtartása miatt az XAML lehetséges oldalelrendezési módjai közül csak a FixedDocument használható egy XPS-dokumentum esetében.

.Net-támogatás – XPS-dokumentum kezelése kódból

Microsoft Visual Studio 2005 használatával pillanatok alatt olvashatunk, kezelhetünk és hozhatunk létre XPS-dokumentumokat. A 2. ábrából látható, hogy magas szinten hogyan hozható létre egy dokumentum logikai struktúrája, és hogyan kapcsolhatók hozzá a különböző fizikai erőforrások (például képek). Gyakorlatilag az XPS-specifikáció alap-

ján létrehozott XML oldalakat mint lapokat egymás után fűzzük, és amennyiben tartalmaz valamilyen erőforrást, például képet, akkor az erőforrást is a dokumentumhoz fűzzük, majd létrehozuk az adott oldalon a

```
// új dokumentum-példány létrehozása
XpsDocument document =
    new XpsDocument(destFileName, FileAccess.
        ReadWrite);

// új FixedDocumentSequence példány létrehozása a do-
kumentumban
IXpsFixedDocumentSequenceWriter docSeqWriter =
    document.AddFixedDocumentSequence();

// új FixedDocument létrehozása ezen belül
IXpsFixedDocumentWriter docWriter =
    docSeqWriter.AddFixedDocument();

// az első FixedPage oldal létrehozása a FixedDocument-ben
IXpsFixedPageWriter pageWriter =
    docWriter.AddFixedPage();

//TODO: Itt kell hozzáadni a dokumentum tartalmát, és
a tartalmazott állományokat

// elmentjük az elkészült oldalt és a dokumentumot
pageWriter.Commit();
docWriter.Commit();
docSeqWriter.Commit();
document.Close();
```

2. ábra. Új XPS-dokumentum létrehozása

logikai hivatkozást rá. Ennek a megoldásnak megvan az az előnye is, hogy egy képet – például céges háttérképet – csak egyszer kell elhelyezni a dokumentumban, és minden oldal hivatkozhat rá.

Az XPS-dokumentumok programozására az XPS Document API ad lehetőséget, ami a System.Windows.XPS névtér használatá-

```
void XamlFileToXps(string srcXamlFile, string destXpsFile)
{
    using(Stream fileStream = File.
        OpenRead(srcXamlFile))
    {
        ParserContext context = new ParserContext();
        context.BaseUri = new Uri(Directory.
            GetCurrentDirectory() + "/");
        XamlStreamToXps(fileStream, context, destXpsFile);
    }
}
```

3. ábra. XPS-dokumentum létrehozása XAML-ből

val vehető igénybe. Ahhoz, hogy felügyelt kódból mi magunk hozzunk létre XPS-dokumentumokat és -oldalakat, a System.

```
void SignDocument(string srcXpsDocument,
    string certFilename)
{
    XpsDocument document =
        new XpsDocument(srcXpsDocument, FileAccess.
            ReadWrite);

    X509Certificate certificate =
        X509Certificate.CreateFromCertFile(certFilename);
    document.SignDigitally(certificate, true,
        XpsDigSigPartAlteringRestrictions.None);

    document.Close();
}
```

4. ábra. XPS-dokumentum aláírása

Windows.Xps.Serialization névtérre lesz szükségünk. A már meglévő XAML oldalakat pedig az XamlStreamToXps metódus hívásával rögtön XPS-nek megfelelő nyelvjárássá fordíthatjuk.

Mind a Microsoft Office 2007, mind az XPS-megjelenítő alkalmazás képes a dokumentumhoz tartozó jogosultságokat és digitális aláírásokat kezelni. Az elkészült dokumentumhoz tartományi vagy Microsoft Passport felhasználói jogosultság alapján különböző megszorításokat lehet alkalmazni (olvasás, másolás, nyomtatás, aláírás, használhatósági idő szabályozása).

Az XML alapú kialakításnak köszönhetően a dokumentum teljes tartalmát egyszerűen tudjuk kezelni, dokumentum, oldal, il-

```
void ListSignatures(string srcXpsDocument)
{
    XpsDocument document =
        new XpsDocument(srcXpsDocument, FileAccess.
            Read);
    foreach (XpsDigitalSignature digitalSignature in
        document.Signatures)
    {
        Console.WriteLine(digitalSignature.SigningTime.
            ToString() + " " +
            digitalSignature.SignerCertificate.Issuer +
            " " +
            digitalSignature.SignerCertificate.Subject);
    }
    document.Close();
}
```

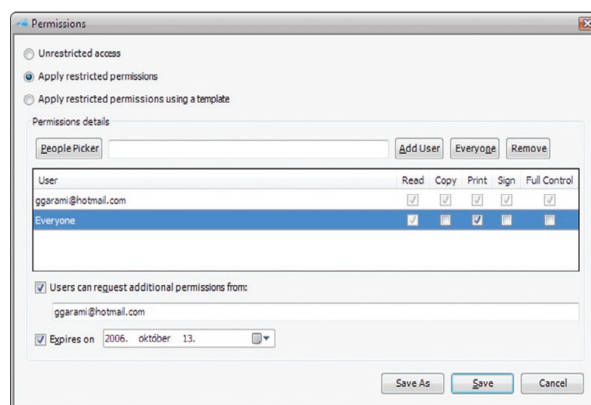
5. ábra. Egy dokumentum tanúsítványainak listázása

letve az oldalon található struktúra szintjén – mindezt a dokumentumok tartalmának teljes körű újrafelhasználása mellett tehetjük meg. Hasonlóan az alábbi példához – ahol a dokumentum tanúsítványait listázzuk –, könnyen írhatunk kódot például a képek vagy más dokumentumelemek feldolgozására is.

Jelenleg fejlesztés alatt állnak azok a konverterek (például xps2ps) és megjelenítők, amelyek segítségével nem Windows alapú rendszereken is fel tudjuk dolgozni, meg tudjuk jeleníteni az XPS-dokumentumokat – amelyre természetesen a specifikáció nyitottsága ad lehetőséget.

Összefoglalásul az XPS főbb előnyei:

- ingyenes, nyílt, hordozható formátum;
- garantáltan azonos megjelenés minden eszközön;



RMS használata XPS-dokumentum esetén

- opcionálisan tömörített adatstruktúra (ZIP);
- XML alapú tárolás, elszeparált adattárolás és megjelenítés;
- beágyazott erőforrások kezelése (betűkészletek, képek);
- egyszerű programozhatóság (létrehozás, feldolgozás, transzformálás);
- RMS kompatibilitás;
- digitális aláírások kezelése.

Jól látszik, hogy az XPS a programozók, valamint az elektronikus dokumentumokkal foglalkozó szakemberek számára is meglehetősen sok lehetőséget kínál. Érdemes továbbgondolni, hogy az XPS hogyan tud majd beilleszkedni a meglévő dokumentumkezelési rendszerekbe, és mennyiben fogja egyszerűsíteni azokat – különösen az új Office rendszer lehetőségeinek felhasználásával.

Garami Gábor

(Gabor.Garami@egroup.hu) fejlesztési vezető, E-Group Magyarország

SQL SERVER 2005 ANALYSIS SERVICES

Cikkünkben az üzleti elemzést támogató legfontosabb funkciókra koncentrálunk, példákon keresztül mutatva be ezek működését.

Az SQL Server 2005 Analysis Services számos olyan funkciót tartalmaz, amely jelentősen megkönnyíti az üzleti alkalmazások elemző, tervező, jelentéskészítési lehetőségeit. Az OLAP alapú rendszerek egyik fontos feladata, hogy az üzleti adatokat különböző szempontok szerint összesítve, rendszerezve mutassák meg a felhasználóknak. Az üzleti adatok elemzése során azonban gyakran találkozunk olyan adatokkal, amelyeket nem lehet minden szempont szerint összesíteni. A probléma leggyakrabban az időbeni összesítéseknel jelentkezik. A szemi-additív mennyiségek segítségével egyszerűen kezelhetünk bizonyos időbeni összesítési problémákat. Az Analysis Servicesben szemi-additív mennyiségeknek nevezzük azokat a mennyiségeket, amelyek az alábbi összesítő függvények valamelyikét használják:

ByAccount. Használatával speciálisan összegezhethetjük a főkönyvi számlák adatait. Ezzel a lehetőséggel részletesen foglalkozunk a továbbiakban. Használatára idő (Time) és számla (Account) típusú dimenziók definiálását igényli.

Ide húzhatja a Szűrőnek szánt mezőket

Év	Negyedév	Hónap	Termékcsoport				Végösszeg
			Érték	Érték	Érték	Érték	
2006	1. negyedév	Január	500	110	9	119	619
		Február	400	100	9	109	509
		Március	300	90	9	99	399
		Összes	1200	300	27	327	1527
	2. negyedév	350	130	25	155	505	
	3. negyedév	950	1960	10	1970	2920	
	Összes	2500	2390	62	2452	4952	
	Végösszeg	2500	2390	62	2452	4952	

1. ábra. Helytelen időbeni összesítés

Ide húzhatja a Szűrőnek szánt mezőket

Év	Negyedév	Hónap	Termékcsoport				Végösszeg
			Érték	Érték	Érték	Érték	
2006	1. negyedév	Január	500	110	9	119	619
		Február	400	100	9	109	509
		Március	300	90	9	99	399
		Összes	300	90	9	99	399
	2. negyedév	50	20	7	27	77	
	3. negyedév	450	960	3	963	1413	
	Összes	450	960	3	963	1413	
	Végösszeg	450	960	3	963	1413	

2. ábra. Helyes időbeni összesítés a LastNonEmpty összesítő függvénnyel

AverageOfChildren. Leegyszerűsíti az idődimenzióra vonatkozó átlagszámítást. Az előbbi függvényekhez hasonlóan az idődimenzióban átlagolja a mennyiséget, míg a többi dimenzióban összegez. Ha például az előző példa alapján az időbeni átlagos készletértékre vagyunk kíváncsiak, akkor az AverageOfChildren a helyes értéket adja az idő- és a termékdimenzióban is: időben átlagolnunk kell, az egyes termékek készletértékét pedig összegezzük.

Vigyázzunk, az AverageOfChildren nem használható a hagyományos átlag – mint például az átlagár – kiszámítására, mivel csak az idődimenzióban végzi el az átlagolást. Ha tehát olyan számításra van szükségünk, amely minden dimenzióban átlagként viselkedik, akkor azt számított mennyiségként kell létrehozunk.

Az 1. ábrán egy példát látunk az időbeni helytelen összesítésre. A hibát korrigálhatjuk, ha az „Érték” és a „Mennyiség” definíciójában a LastNonEmpty nevű összesítő függvényt alkalmazzuk.

A 2. ábrán a javított definíciók alapján készült összesítések mellett az átlagos készletértéket is kiszámoljuk az AverageOfChildren használatával.

Nem kis segítség az MDX-kifejezések írásakor, hogy a fenti szemi-additív összesítő függvények jól működnek az Aggregate függvényben, így lehetőség van az időbeni helyes

```
WITH MEMBER [Measures].[Kék érték] AS
'Aggregate (
  {[Raktar].[Raktar Nev].[Kék raktár]},
  [Measures].[Ertek]
)'
SELECT
  ([Termek].[Termekcsoport].[Könyvek])*
  ([Measures].[Kék érték]) ON COLUMNS,
  [Ido].[NegyedevNev].Members ON ROWS
FROM
  [Pelda]
```

Messages Results

	Könyvek
All	960
1. negyedév	90
2. negyedév	20
3. negyedév	960

3. ábra. Az Aggregate függvény használata

aggregáció kiszámítására egy tetszőleges hal-mazon is.

A 3. ábrán látható példa a „Kék raktár”-ban lévő könyvek készletértékét számolja ki.

Számlaintelligencia

A beépített számlaintelligencia lehetővé teszi, hogy a főkönyvi számlák összegzési szabályai szerint végezzük el adataink összesítését a kockánkban. A számlaintelligencia az előbbiekben megismert szemi-additív mennyiségekre, valamint az unáris operátorokra és egyedi összesítésekre épül.

Nézzük meg néhány példán keresztül, hogy milyen megoldandó feladatokkal kell szembenéznünk, ha alkalmazásunkba számla alapú elemző funkcionalitást szeretnénk beépíteni.

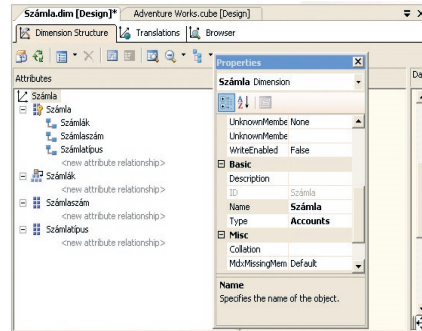
A számladimenzió megvalósítása. A számladimenzió tipikusan szülő-gyerekk hierarchia segítségével írható le, amelynek három kitüntetett eleme van: a számla száma, a főlérendelt számla száma és a számla elnevezése. Az Analysis Servicesben a számladimenziót ezzel a három attribútummal definiálhatjuk, a számlaszámot kulcsként („Key”), míg a főlérendelt számlát szülőként („Parent”) definiálva.

A számladimenzió alapján elemzendő mennyiségi adatok fontos jellemzője, hogy szorosan összefüggenek a számladimenzióval, azaz a számladimenzió aktuális tagjának értéke minőségi-leg is meghatározza a mennyiségi adatot (például „Nettó bevétel”, „Befektetett eszközök”), ellentétben más dimenziókkal, amelyek inkább leíró jellegű információt hordoznak (például „Terméknév”, „Raktárkód”).

Általában a számladimenzióknak együttműködő mennyiségeket nem különítjük el számlánként, mivel ez nagyon sok mennyiségi adat definiálását jelentené, hanem egy mennyiségi adatot definiálunk (például „érték”, „amount”), és ebbe töltjük be a különböző számlákhoz tartozó adatokat.

A számla jellemzésére használhatjuk a számlatípust (például bevételi számla, kiadási számla, egyenleg) mint attribútumot, amely a számlán végzett összesítések definiálásához szolgáltat fontos információt, ahogyan ezt a későbbiekben látni fogjuk.

Tipp: a számladimenzió definiálásánál ne felejtjük el a dimenzió „Type” tulajdonságát „Accounts”-ra, a számlatípus-attribútum „Type” tulajdonságát pedig „Account Type”-ra állítani, mert a számla szerinti összesítés csak ebben az esetben működik!



4. ábra. A számladimenzió definiálása

A számladimenzióhoz gyakran kapcsolódik esetszintű dimenzió, a terv-, tényadatok és tervváltozatok megkülönböztetéséhez.

Unáris operátorok és egyedi összesítések. Az első megoldandó feladat a számlák helyes összesítése a számladimenzióban valamennyi dimenzióra nézve, hiszen például a kiadást nem hozzáadnunk kell a bevételhez, amikor az eredményt számoljuk, hanem le kell vonnunk belőle.

Ezt a problémát a számladimenzióra alkalmazott unáris operátorokkal (Unary

Current level:	Számldim szint 02	Számldim szint 01	Unary Operator	Custom Rollup
[-] Balance Sheet				
[-] Assets	Assets		~	
[-] Liabilities and Owners Equity	Liabilities		-	
[-] Net Income	Revenue		+	
[-] Operating Profit	Revenue		+	
[-] Other Income and Expense	Revenue		+	
[-] Taxes	Expenditures		-	
[-] Statistical Accounts	Statistical			
[-] Headcount	Balances		~	
[-] Units	Flow			
[-] Average Unit Price	Balances		~	
[-] Square Footage	Balances		~	

5. ábra. Főkönyvi számlák helytelen összesítése

Operators) oldhatjuk meg. Az unáris operátorokkal összegezhettük („+”), kivonhatjuk („-”), szorozhatjuk („*”) és oszthatjuk („/”) egymással a dimenzió egyes tagjainak értékét, megváltoztatva ezzel az összesítések kiszámításának módját. Lehetőségünk van arra is, hogy az összesítésből kihagyjunk („~”) bizonyos tagokat.

Az unáris operátorok megvalósítása a dimenziótábla új oszloppal, az „unáris operátor oszloppal” való bővítésével, az unáris operátorok megfelelő megadásával és a dimenzió-

attribútum „UnaryOperatorColumn” tulajdonságának beállításával érhető el.

Amennyiben az unáris operátorokkal nem tudjuk megfelelően definiálni az adott számla összesítését, célszerű lehet egyedi összesítést alkalmazni a CustomRollupColumn tulajdonság használatával. Az egyedi összesítést a dimenziótáblához hozzáadott oszlop segítségével definiálhatjuk, amelyben egy MDX-kifejezéssel írhatjuk le a számítást. Az átlagárát például a „Nettó bevétel” és az „Eladott mennyiség” hányadosaként számolhatjuk ki, ezt azonban nem tudjuk unáris operátorokkal megtenni, mert a két mennyiség különböző helyen helyezkedik el a számlahierarchiában.

Ide húzhatja a Szűrőnek szánt mezőket			Ide húzhatja az		
Számla szint 01	Számla szint 02	Amount	Számla szint 01	Számla szint 02	Amount
[-] Balance Sheet	[-] Assets	13740731	[-] Balance Sheet	[-] Liabilities and Owners Equity	13740731
	[-] Liabilities and Owners Equity	13740731		Összes	27481462
[-] Net Income	[-] Operating Profit	119412203.5	[-] Net Income	[-] Other Income and Expense	359517.5
	[-] Other Income and Expense	359517.5		[-] Taxes	4168749
	[-] Taxes	4168749		Összes	123940470
[-] Statistical Accounts	[-] Headcount	188	[-] Statistical Accounts	[-] Units	5,305
	[-] Units	5,305		[-] Square Footage	276,500
	[-] Square Footage	276,500		Összes	NA

6. ábra. Unáris operátorok és egyedi összesítések definiálása

Tipp: ha a számladimenziókat írhatóvá tesszük, akkor az unáris operátorokat és az egyedi összesítéseket a BI Development Studióban is szerkeszteni lehet.

A 7. ábrán az unáris operátorok és egyedi összesítések alapján képzett helyes értékeket láthatjuk.

Ide húzhatja a Szűrőnek szánt mezőket			Ide húzhatja az Oszlop		
Számla szint 01	Számla szint 02	Amount	Számla szint 01	Számla szint 02	Amount
[-] Balance Sheet	[-] Assets	13740731	[-] Balance Sheet	[-] Liabilities and Owners Equity	13740731
	[-] Liabilities and Owners Equity	13740731		Összes	0
[-] Net Income	[-] Operating Profit	16728234.5	[-] Net Income	[-] Other Income and Expense	50017.5
	[-] Other Income and Expense	50017.5		[-] Taxes	4168749
	[-] Taxes	4168749		Összes	12609503
[-] Statistical Accounts	[-] Headcount	188	[-] Statistical Accounts	[-] Average Unit Price	12182.1805843544
	[-] Units	5,305		[-] Square Footage	276,500
	[-] Square Footage	276,500		Összes	NA

7. ábra. Főkönyvi számlák összesítése unáris operátorokkal és egyedi összesítésekkel

Számlák szerinti összesítés – a ByAccount összesítő függvény használata. Az unáris operátorok definiálásával azonban még nem oldottunk meg minden feladatot, mivel az unáris operátorok nem veszik figyelembe az időt. A probléma lényege az, hogy a számladimenzióknak egyes tagjait, mivel más típusú mennyiségeket írnak le, időben más-

képpen kell összegeznünk! A FirstChild, LastChild, FirstNonEmpty, LastNonEmpty, AverageOfChildren összesítő függvények itt nem alkalmazhatók, mivel ezek csak az idődimenziót veszik figyelembe, és nem foglalkoznak a számladimenzió aktuális értékével.

A megoldás a ByAccount összesítő függvény használata, amely lehetővé teszi, hogy az adatokat a számladimenzió és az idő függvényében megfelelően összesítsük.

Ide húzhatja a Szűrőnek szánt mezőket

Számla szint 02		Assets	Liabilities
Calendar Quarter	Month	Amount	Amount
Q3 CY 2001	July 2001	2960514	2960514
	August 2001	6601885	6601885
	September 2001	6339054	6339054
	Összes	15901453	15901453
Q4 CY 2001		25258913	25258913
Q1 CY 2002		23444943	23444943
Q2 CY 2002		23866269	23866269
Q3 CY 2002		40522145	40522145
Q4 CY 2002		36416398	36416398
Q1 CY 2003		28552806	28552806
Q2 CY 2003		32865952	32865952
Q3 CY 2003		45778286	45778286
Q4 CY 2003		40805093	40805093
Q1 CY 2004		33018641	33018641
Q2 CY 2004		37823368	37823368
Végösszeg		394254267	394254267

8. ábra. Helytelen összesítések a számla- és az idődimenzió metszetében

A ByAccount használatának legfontosabb lépése az összesítő függvények és a számlatípusok egymáshoz rendelése. Ezt az „adatbázisnév.database” forrásállományban tehetjük meg, de a BI Development Studióban vizuálisan is beállíthatjuk a Database/Edit Database menüpontban. Az egymáshoz rendelést a számladimenzió „Account Type” attribútumának lehetséges értékei, és az ezekhez tartozó megfelelő összesítő függvények megadásával végezzük el.

A számladimenzió megfelelő definiálása, az unáris operátorok és egyedi összesítések definiálása, valamint az összesítő függvények

Account Type Mapping

Name	Alias	Aggregation Function
Statistical	Statistical	LastNonEmpty
Liability	Liabilities	LastNonEmpty
Asset	Assets	LastNonEmpty
Balance	Balances	LastNonEmpty
Flow	Flow	Sum
Expense	Expenditures	Sum
Income	Revenue	Sum
<Add new account type>		

9. ábra. Az egyes számlatípusok összesítő függvényeinek beállítása

és számlatípusok egymáshoz rendelése után már csak egy feladatunk marad: a számladimenziót használó mennyiségeink összesítő függvényét ByAccountként definiáljuk.

Fő teljesítménymutatók

A fő teljesítménymutatók (angolul Key Performance Indicators – KPI) a vállalat mű-

ködésének legfontosabb jellemzőit írják le, a különböző adatokat a döntéshozók, elemzők számára egy-egy fontos mutatószámba „sűrítve”.

Ide húzhatja a Szűrőnek szánt mezőket

Számla szint 02		Assets	Liabilities and Ow
Calendar Quarter	Month	Amount	Amount
Q3 CY 2001	July 2001	2960514	2960514
	August 2001	6601885	6601885
	September 2001	6339054	6339054
	Összes	6339054	6339054
Q4 CY 2001		9068151	9068151
Q1 CY 2002		7570883	7570883
Q2 CY 2002		7116989	7116989
Q3 CY 2002		14313024	14313024
Q4 CY 2002		11121862	11121862
Q1 CY 2003		8248291	8248291
Q2 CY 2003		10428844	10428844
Q3 CY 2003		16338041	16338041
Q4 CY 2003		13399865	13399865
Q1 CY 2004		11371044	11371044
Q2 CY 2004		13740731	13740731
Végösszeg		13740731	13740731

10. ábra. Helyes időbeni összesítések a ByAccount összesítő függvény használatával

Az Analysis Services kiemelten támogatja a teljesítménymutatók definiálását, megjelenítését és a rajtuk végzett műveleteket a KPI-k segítségével.

A teljesítménymutatókat az Analysis Servicesben négy számított mennyiség definiálásával adhatjuk meg:

KPI-érték. MDX-kifejezés, a teljesítménymutató értéke. Egyszerű esetben lehet egy mennyiségi adat, leggyakrabban egy arányszám.

KPI-cél. MDX-kifejezés, a teljesítménymutató célértéke.

KPI-státusz. MDX-kifejezés, a teljesítménymutató aktuális értéke a célhoz viszonyítva. Értékét -1 és 1 közé normalizáljuk: értéke 1, ha túltejesítettük a célt, 0, ha a célnál vagyunk és -1, ha nem értük el a célt. -1

és 1 között tetszőleges számú közbülső értéket is felvehet.

KPI-trend. MDX-kifejezés, a teljesítménymutató időbeni változása. Értéke 1, ha a trend erősen pozitív, 0, ha változatlan az időbeni érték, és -1, ha a trend nagyon negatív. -1 és 1 között tetszőleges számú közbülső értéket is felvehet, így jelölve a trend irányát és relatív erősségét.

A fentiek mellett lehetőségünk van a státusz és trend alapértelmezett grafikus megjelenítését is előírni, így támogathatjuk például az olyan megjelenítő alkalmazásokat, mint az Excel 2007, a Report Builder vagy a Business Scorecard Manager.

A teljesítménymutatókat hierarchiába is szervezhetjük és súlyozhatjuk, így lehetőségünk van összetett teljesítménymutatók definiálására is. Összetett teljesítménymutatókat

a szülő (KPI parent) és a szülőn belüli súly (KPI weight) értékének megadásával képezhetünk.

Amennyiben a szülő teljesítménymutatójának definíciójában használjuk az alárendelt teljesítménymutatókat, a súlyok alkalmazásával kisebb vagy nagyobb mértékben vehetjük figyelembe az egyes részmutatók eredményeit.

Megkönnyíti az együttműködést az idődimenzióval, hogy a KPI számára az idődimenzió alapértelmezett értékétől eltérő értéket is megadhatunk, így a megjelenítésnél nem kell ezt külön definiálnunk.

A teljesítménymutatókkal végzett munkát segítik az MDX KPI-k kezelésére szolgáló

Name: Operating Profit

Associated measure group: Financial Reporting

Value Expression

```
// Account 48 = Operating Profit, Scenario 1 = Actual
([Account].[Account].<[48], [Scenario].[Scenario].<[1], [Measures].[Amount] )
```

Goal Expression

```
// Account 48 = Operating Profit, Scenario 2 = Budget
([Account].[Account].<[48], [Scenario].[Scenario].<[2], [Measures].[Amount] )
```

Status

Status indicator: Traffic light

Status expression:

```
When KpiValue( "Operating Profit" ) / KpiGoal( "Operating Profit" ) >= .95
Then 1
When KpiValue( "Operating Profit" ) / KpiGoal( "Operating Profit" ) < .95
And
KpiValue( "Operating Profit" ) / KpiGoal( "Operating Profit" ) >= .85
Then 0
```

Trend

11. ábra. KPI-definíció

függvényei: KPIValue, KPIGoal, KPIStatus, KPIWeight, KPICurrentTimeMember, amelyeket lekérdezésekben, MDX-kifejezésekben használhatunk a KPI-kre épülő számítások definiálásában, valamint a KPI-k megjelenítésében.

Pénznemkonverziók

Gyakran felmerülő feladat, hogy a különböző üzleti adatokat – mint amilyenek például a bevételek, a kiadások, a fizetendő adók, az eladási adatok – különböző pénznemekben jelenítsük meg. Az Analysis Services támogatja a pénznem konverzióját is, kihasznál-

va a több-több dimenziókapcsolat nyújtotta lehetőséget.

Az alapelv a következő: a táblában egy pénznemben tároljuk az adatokat, az alkalmazandó árfolyamokat, átváltási arányokat

```
SELECT
    (KPIValue("Operating Profit"),KPIGoal("Operating Profit"),
    KPIStatus("Operating Profit"),KPI Trend("Operating Profit")) ON COLUMNS,
    [Date].[Calendar].[Calendar Quarter].Members ON ROWS
FROM
    [Adventure Works]
```

	Operating Profit Value	Operating Profit Goal	Operating Profit Status	Operating Profit Trend
Q3 CY 2001	565965	750290	-1	1
Q4 CY 2001	1823445	2250190	-1	1
Q1 CY 2002	1061064	1434300	-1	1
Q2 CY 2002	814604.5	1149120	-1	1

12. ábra. KPI-függvények alkalmazása

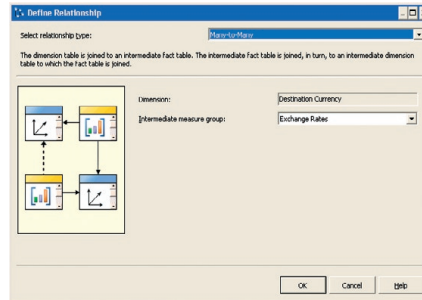
pedig egy külön táblában tároljuk pénznemenként és naponként. A pénznemeket tartalmazó dimenziókat az árfolyamokat, átváltási arányokat pénznemenként tartalmazó táblán keresztül kapcsoljuk az eredeti táblánkhoz. Mindezek következtében egy több-több kapcsolatot hozunk létre az eredeti adataink és a pénznemek között, miközben az átváltási árfolyamhoz is hozzáférünk a kapcsolatot megvalósító köztes táblában.

A megoldáshoz most már csak egy a-

róság hiányzik: a megfelelő átváltási árfolyam alkalmazása a mennyiségeinkre. Ezt a legegyszerűbben a mennyiségi adatunk „MeasureExpression” tulajdonságának beállításával érhetjük el: a mennyiségünket elosztjuk az átváltási aránnyal.

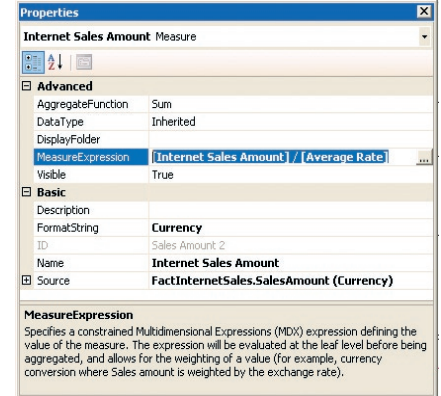
Összefoglalás

A fenti technikákat alkalmazva az Analysis Services segítségével egyszerűbben és gyorsabban építhetünk komplex elemző, tervező rendszereket, és támogatjuk az adatok megjelenítéséhez használt eszközök funkcionalitásának jobb kihasználását. Számos



13. ábra. Pénznemkonverzió megvalósítása több-több dimenziókapcsolat alkalmazásával

olyan eszköz, technika van még, amelyet a cikk terjedelmi korlátai miatt nem tudunk bemutatni – így például a nyelvi lehetőségek bemutatása, az egyes dimenziókapcsolatok



14. ábra. Az átváltási arány alkalmazása a mennyiségre

működése, a visszairás megvalósítása, az alternatív csoportosítások, hierarchiák kezelése. Terveink szerint a következő számok valamelyikében még visszatérünk ezeknek a lehetőségeknek a bemutatására is.

Kovács Zoltán

(kovacs@szamalk.hu) vezető oktató, Számalk

IT-BUSINESS PRODUCTS

INFOKOMMUNIKÁCIÓS TERMÉKEK ÜZLETI DÖNTÉSHOZÓKNAK

- informatikai döntéshozóknak és technológiai szakembereknek
- az elmúlt 168 óra legfontosabb hazai és külföldi, ICT-piacca kapcsolatos termékbejelentések, hardvereszközök és termékújdságok
- heti ingyenes elektronikus hírlevél

Regisztráljon!

www.it-business.hu/hirlevel



FOLYJÉK A MUNKA!

Workflow a 2007-es Office rendszerben.

Az új TechNet magazin első számában volt egy áttekintő cikk a Windows Vista idején megjelenő fejlesztői technológiákról. Az egyik legfontosabb fejlemény, hogy ezeket a technológiákat ma már WinFX helyett a .Net keretrendszer 3.0-s változatának nevezük. És ez az elnevezés talán még jobban mutatja, hogy ezek a technológiák a platform részei lesznek, és más alkalmazások majd ezekre építve tudják bővíteni saját szolgáltatásaikat. A .Net 3.0 egyik komponense a Windows Workflow Foundation (WF), ami a munkafolyamatok definiálását és kezelését teszi könnyűvé. Ebben a cikkben áttekintjük a WF felhasználásának egyik első területét, integrációját a 2007-es Office rendszerbe.

Mire jók a munkafolyamatok?

Mindennapi életünk során állandóan folyamatokkal találjuk szembe magunkat, még pontosabban: folyamatokban találjuk magunkat. Hivatali ügyintézés vagy egy szerződés jóváhagyása, sőt akár az autónk működése is – mind folyamatok. Esetünkben a folyamat definíciója: elemi műveletek szabályos egymásutánja. A szoftveralkalmazásokban ma is számos munkafolyamatot használunk, de ezek el vannak ásva az if-else-ek és switchek mélyére. Ennek két hátránya is van: egyrészt nehezen tekinthetők át, másrészt – ami az előzőből következik – nehezen lehet módosítani őket. A WF révén az alkalmazás folyamatlemeit és azok tulajdonságait függetleníti tudjuk a programkódtól, el tudjuk választani egymástól a „tartalmat és a formát”. A folyamat elemeit mint Activityket definiáljuk, ezek az alapvető műveletek, illetve az ezeket rendbe soroló elemek, mint például a feltételes elágazások vagy a tranzakciók.

Ezekből összerakunk egy workflow-típust, és a workflow-runtime ezeket a workflow-típusokat futtatja, menedzseli számunkra. Ennek előnye, hogy az így kivont formát, a munkafolyamat szerkezetét külön kezelve akár a felhasználók kezébe is adhatjuk a folyamat meghatározását, megkönnyítve ezzel a kommunikációt. A workflow-runtime annyira rugalmas, hogy a futó munkafolyamatokat akár futás közben is átalakíthatjuk, ha éppen arra van szükség. A folyamatokat kód mellett definiálhatjuk deklaratív módon az eXtensible Application Markup Language nyelven is (ez az XML egy nyelvjárása, aminek különleges szerepe van a .Net 3.0-ban), ami lehetővé teszi azt is, hogy egy grafikus tervező segítségével diagram formájában jelenítsük meg a folyamatot.

Workflow és Office

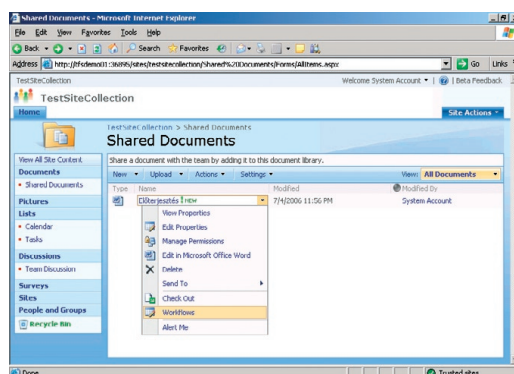
Miért hasznos a workflow fogalma az Office-alkalmazások számára? A Microsoft Office fejlődése során egyre nagyobb szerepet játszik a szervezetek információkezelésében. A csomagból rendszerré válása is ennek szellemében érthető meg a leginkább, hiszen a fejleszhető interfészekkel és az egyre komplexebb funkciókkal lehetővé válik, hogy ne csak a dokumentumok létrejöttét, hanem létük későbbi fázisát is támogassa. A cél az, hogy végig lehessen követni a dokumentumok és a bennük rejlő információ létét a létrehozástól a különböző folyamatokon át (szerkesztés, megosztás, jóváhagyás stb.) a publikációig (nyomtatás, web stb.) és utána akár az archiválásig.

Ebben az életciklusban több olyan szakasz is van, amelyeket jól lehet leírni ilyen .Net-munkafolyamatokkal, és itt hasznosulhatnak azok a platformfunkciók, amelyekről már szó volt (perzisztens tárolás, átkonfigurálhatóság, felhasználó általi megértés és módosítás stb.). A workflow több ponton is érintkezik az Office rendszer víziójával. A legfontosabb talán az, hogy az egyes

dokumentumokhoz lehet munkafolyamatokat rendelni, például egy jóváhagyást vagy közös szerkesztést. A másik lehetőség, hogy megoldást építünk fel, ahol a munka lényegét a munkafolyamat képezi, és ebbe néha-néha információt adunk be egy-egy InfoPath-úrlap segítségével. Ilyen lehet mondjuk egy hivatali ügyintézés informatikai támogatása.

A workflow típusai

A platform szemszögéből a munkafolyamatok többféle csoportosítása is létezik. Egy workflow lehet szekvenciális (sequential workflow), vagy állapotgép (state machine workflow), illetve szabály alapú. Technológiailag lehet tisztán kód alapú, kód + markup alapú vagy tisztán markupban definiált. Az utóbbi felosztásnak a megjeleníthetőség és a programozók, illetve a felhasználók közötti szereposztásban van szerepe, az előbbi pedig meghatározza, hogy mit tudunk vagy mit nem tudunk leképezni vele: a szekvenciális



Bármely SharePoint-dokumentumhoz tartozhat munkafolyamat

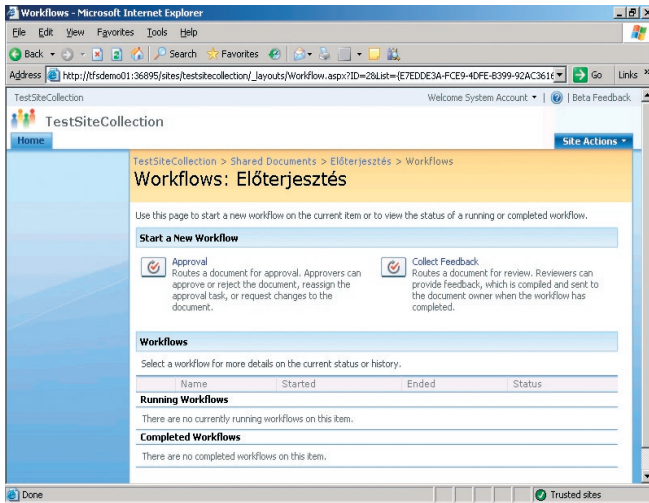
workflow egy előre meghatározott folyamat, amiben egyszerű elágazások még lehetnek, de például nem ugrálhatunk benne.

Az állapotgép-workflow lényege, hogy állapotokat definiálunk, ezek között átmenetek vannak, és így képezhetjük le a folyamatunkat, ami akár indeterminisztikus is lehet.

Például az egyes állapotok lehetnek egy hivatal ügyosztályai, amelyek tetszőleges vagy korlátozott sorrendben, de általában tetszőleges számú lépésben adogathatják egymásnak az „ügy” állapotát. (Igen, most minden leendő Office-fejlesztőnek elkezdi vizsketni a billentyűzete, és elképzeli, hogy mekko-

ni saját eszközeinkkel, hogy testreszabjuk a SharePoint lehetőségeit. Ehhez a SharePoint Designert fogjuk használni, ami a FrontPage szerepét veszi át a SharePoint- oldalak és -alkalmazások szerkesztésében. Miután kinyílt az alkalmazás, nyissuk meg a SharePoint-site-unkat! A SharePoint Designer képes a

mentsük a workflow-t. Közben láthatjuk, hogy a SharePoint Designer egy XOML mark-up-fájlt készít az általunk definiált folyamatból. Ha most frissítjük a korábbi Workflows- oldalt a böngészőben, akkor megjelenik egy újabb lehetőség, a mi folyamatunk, elindítása után pár pillanattal már láthatjuk is az álla-



Egy dokumentum munkafolyamatainak részletei

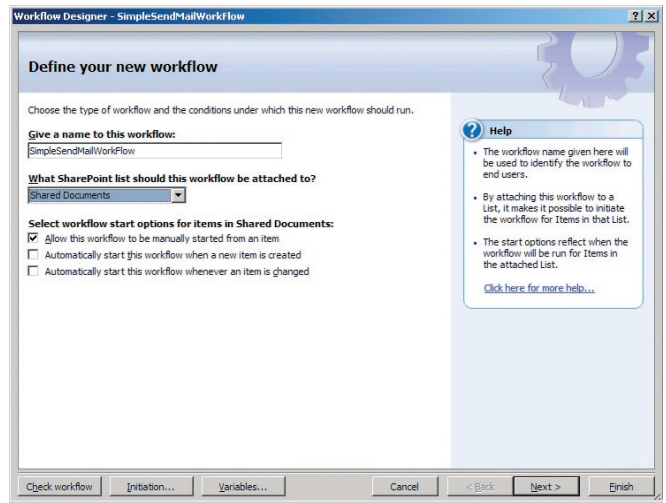
ra MaxNumberOfIterations konstantt fog beépíteni a rendszerbe...) A szabály alapú workflow lehetővé teszi, hogy különválasszuk a folyamat stabil és változékony elemeit: általában a folyamatok jellege ritkán változik, de az egyes döntési pontok paraméterei gyakrabban: például egy jóváhagyási folyamat formája ritkán változik, de az, hogy milyen összegig ki hagy jóvá, lényegesen gyakrabban. A szabály alapú folyamatok segítségével alkalmazkodhatunk az ilyen folyamatokhoz is.

Egyszerű Office Workflow

A Microsoft Office SharePoint Server (MOSS) már telepítés után tartalmaz előre definiált workflow-kat és workflow-elemeket. Ha egy dokumentumműveletet menüjét, akkor láthatunk ott egy Workflows pontot, mögötte pedig a két alapvető folyamatot (Approval és Collect Feedback). Innen könnyen el is indíthatjuk, majd megkapjuk a kezdő InfoPath-formot (igen, a böngészőben, ugyanis a MOSS egyik új lehetősége az InfoPath-űrlapok webesítésére, illetve szimmetrikus űrlapok létrehozására, amelyek működnek a weben is és az Office-alkalmazásokban is), ahol megadhatjuk a kezdő paramétereket. Ehelyett mi most egy egyszerű workflow-t szeretnénk definiál-

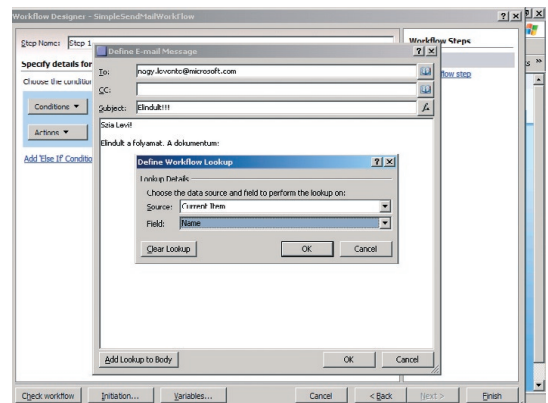
saját lehetőségeit korlátozni aszerint, hogy milyen szerephez tartozó felhasználó futtatja éppen. Ahhoz, hogy a Beta2-es változatban lássuk a Workflow-hoz kapcsolódó funkciókat, először ki kell kapcsolni ezt a funkciót: a Site menü Contributor settings ablakában válasszuk az Advanced gombot, vegyük ki az „Inherit ...” pipát, és utána nyomjuk meg a „Disable Contributor Settings” gombot! Az OK gombra kattintva visszatérhetünk az alapnézetbe, és máris nekiláthatunk egy egyszerű workflow összerakásának.

Nyissuk meg a File/New/Workflow ablakot, és nevezzük el új folyamatunkat SimpleSendMailWorkFlow-nak! Válasszuk ki, hogy a Shared Documents mappa a célunk, és a pipákat hagyjuk érintetlenül, mi most egy manuálisan indítandó folyamatot hozunk létre. A Next után definiálhatjuk a folyamat lépéseit. Új elem hozzáadásakor válogathatunk a SharePoint által előre telepített folyamat-elemek közül, például kiválaszthatjuk a Send-Email nevűt. Ezután beállíthatjuk a levél részleteit, ahol akár a dokumentum tulajdonságaiból is beilleszthetünk a szövegbe, majd



Új munkafolyamat létrehozásának lépései

potról szóló oszlopban, hogy lefutott. A beépített folyamat-elemeket böngészve rájöhettünk, hogy a SharePoint Designer segítségé-



A dokumentum tulajdonságait is használhatjuk paraméterként

vel egész bonyolult folyamatokat is le tudunk képezni, sőt, akár egyszerű if-else elágazások is definiálhatunk.

Workflow-tervezés Visual Studióval

Mit tehetünk, ha kinőttük a SharePoint Designer kereteit? Például olyan aktivitásokat is szeretnénk felhasználni, amelyek nincsenek alpból a SharePointban (mondjuk szeretnénk XML webszolgáltatásokat meghívni), vagy komplexebb logikát szeretnénk

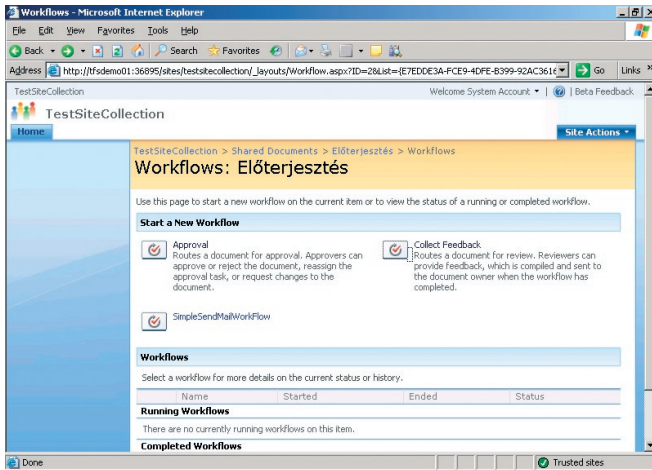
megvalósítani, amit már nem lehet egy szkevenciális folyamattal leképezni, és szeretnénk használni a WF adta állapotgép-lehetőséget? Ekkor kell a Visual Studióhoz nyúlni. A WF runtime mellett (ami települ a .Net 3.0-val) szükségünk lesz a Visual Studio Extensions for Windows Workflow Foundation megfelelő verziójára. Ez biztosítja számunkra a workflow-tervezés lehetőségét a Visual Studióban, illetve hasznos letölteni

SharePoint alapú workflow-k fejlesztéséhez. Ha ez megvan, akkor az új projekt ablakában egy új mappa keletkezik, ahonnan létre tudunk hozni SharePoint Workflow-projekteket. Ahhoz, hogy igazán boldogok legyünk, még egy lépésre szükség lesz: a Toolboxba importálni kell a SharePoint által már definiált folyamatemeleket, hiszen például a korábban használt e-mail-küldési funkció alpból nem biztosított a WF-ben, azt a SharePoint

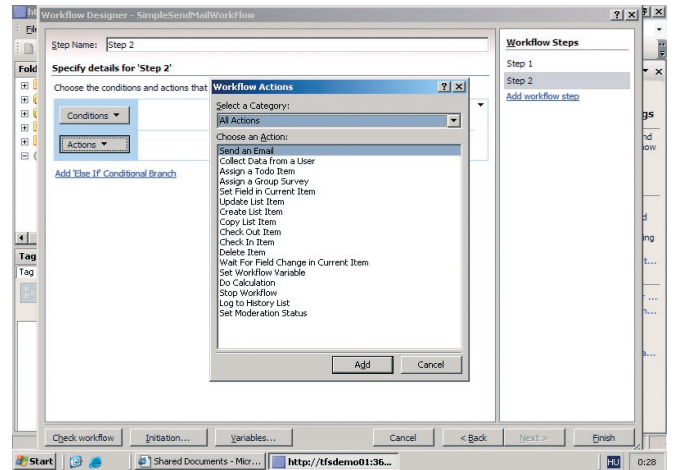
Point-listát szabtuk testre, hanem egy általános workflow-megoldást készítettünk el. A megoldást be kell csomagolni egy SharePoint feature-be, és azzal lehet majd telepíteni.

További lehetőségek

Most csak a felszínt tudtuk megkapargatni, hiszen nem volt még idő foglalkozni azzal, ha az InfoPath-t is bevonjuk a folyamatokba (weben vagy kliensben), vagy akár XML



Egy már létrehozott munkafolyamat-típus egy kattintással elindítható



A munkafolyamatok során felhasználható lépések típusai

a Workflow Developer Starter Kit for Windows SharePoint Services V3 nevű csomagot, ami alap-projektsablonokat tartalmaz a

adta nekünk, de szeretnénk itt is használni. Ezt megtehetjük a <http://msdn2.microsoft.com/en-us/library/ms460227.aspx> címen írt módon.

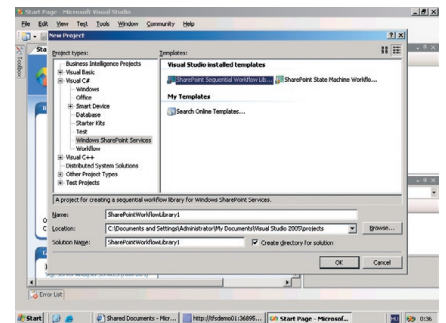
A frissen létrehozott folyamatunk első eleme már ki is van téve, ez kötelezően az on-WorkflowActivated elem. Ez inicializálja azokat az értékeket, amelyeket a SharePointban hostolt workflow-motorból kap a folyamatunk. Ez azért fontos, mert egy folyamatpusnak több példánya is futhat egy szerveren, és ezeket egymáshoz képest azonosítani kell. Ha szeretnénk például a korábbi e-mail-küldő folyamatunkat rekonstruálni, akkor elhelyezhetünk egy SendMail típusú elemet a folyamatunkban, aminél azonnal értéket is kell adni a correlation tokennek és a WorkflowId-nek, mindkettőt a kötelező elemből véve (az első workflowToken a másodikat pedig a ... gomb megnyomásával hozzá tudjuk bindolni az első activity WorkflowId tulajdonságához).

Így a beépített aktivitások és a saját magunk által írt aktivitások felhasználásával tetszőleges folyamatot tudunk implementálni. Ezeket a folyamatokat már bonyolultabb telepíteni is, hiszen itt nem egy konkrét Share-

webszolgáltatásokat is, illetve nem használtuk még ki az új fájlformátum adta lehetőségeket – például a tartalomtípusok definíciójának lehetőségét. A fantáziánk beindításához álljon itt most egy példa: az új formátum és a SharePoint együttműködése lehetővé

További információk

- Office rendszer honlap – <http://www.microsoft.com/office/preview>
- Office-fejlesztői oldalak – <http://msdn.microsoft.com/office>
- Workflows in Windows SharePoint Services – <http://msdn2.microsoft.com/en-us/library/ms416312.aspx>
- Workflows in Office SharePoint Server 2007 – <http://msdn2.microsoft.com/en-us/library/ms549489.aspx>
- Office Developer Screencasts – <http://msdn.microsoft.com/office/learn/screencasts/>
- How to: Manually Configure a Workflow Project – <http://msdn2.microsoft.com/en-us/library/ms460227.aspx>
- Workflow Development Tools Comparison – <http://msdn2.microsoft.com/en-us/library/ms461944.aspx>
- Contributor Settings in SharePoint Designer – <http://weblogs.asp.net/jan/archive/2006/06/02/Contributor-Settings-in-SharePoint-Designer.aspx>



Saját munkafolyamat Visual Studio 2005-tel

szí, hogy ne csak dokumentumokkal, hanem „szerződésekkkel” és „előterjesztésekkel” foglalkozunk, és a SharePointben majd ilyen tartalomtípusokhoz is köthetünk folyamatmodelleket. Látható: egész komplex rendszereket lehet majd építeni az Office rendszer hátára.

Nagy Levente
(nagy.levente@microsoft.com) Microsoft Magyarország

MENTÉS ÉS VISSZAÁLLÍTÁS

A SharePoint-portálok és -webhelyek fő erőssége a rugalmasság: sokféle elemmel egészíthetők ki, képesek együttműködni más alkalmazásokkal, saját arculattal ruházhatók fel.

A SharePoint-portálok rugalmasságának hátterében egy összetett rendszer áll, és az üzemeltetők sokszor csak egy katasztrófa utáni visszaállítás alkalmával szembesülnek először ezzel az összetettséggel.

Természetesen javarészt el lehet kerülni a bonyodalmakat, ha a lényeges információk rendelkezésre állnak. Ebben a szellemben néhány olyan tájékozási pontot szeretnénk megmutatni, amelyek nem egyértelműek, vagy kevésbé ismertek, de nagyban hozzájárulhatnak a vészhelyzetek sikeres megoldásához vagy az azokra való felkészüléshez. Nem az tehát a célunk, hogy a nyilvánosan elérhető SharePoint-leírásokban már ismertetett mentési módszereket összegezzük vagy részletezzük – ilyen igényekhez a „Resource Kit” című könyv „Disaster Recovery” fejezete kitűnő áttekintést nyújt.

A SharePoint-rendszer részei és mentései

Közhely, hogy nem sokat ér az a mentés, amelyből nem lehet visszaállítani a rendszert. Ha valaki próbált már SharePoint-portált vagy -webhelyet minden hozzá tartozó egyéni átalakítással együtt visszaállítani, az tudja, hogy ennek a tesztelésére érdemes időt szánni. Egy elveszett SharePoint-szerver visszaállításához az alkalmazás teljes újraterelítése után számos elemet kell helyretenni: a nyelvi sabloncsomagokat; a saját fejlesztésű webkijelzőket; a saját készítésű lista- és webhelysablonokat; egyéb, a külsínt illető módosításokat (képeket, stíluslapokat); a tartalmat (magukat az adatokat); portál esetén még az indexeket.

Nagyon fontos tudnivaló, hogy a SharePoint mentési eszközei a tartalom és portál indexeinek mentésére és visszaállítására szolgálnak. A nyelvi sabloncsomagok, saját webkijelzők és sablonok hiányában azonban az adattartalom sikeres visszaállítása nem elég ahhoz, hogy el lehessen érni a SharePoint weboldalait! Mivel nagyon sok esetben nem ugyanaz a személy – vagy társaság – végzi el a testreszabást, mint aki katasztrófa esetén a visszaállításért felelős, a módosításokat érdemes dokumentáltatni az üzemeltetés részére a bevezetést végző szakemberrel vagy csapattal. Lehetőség szerint a rendszer átvételekor el kell végezni egy visszaállítási próbát; ilyenkor az üzemeltető maga telepíti a testreszabásokat, és ellenőrzi a dokumentációs hibákat, hiszen majd egy éles visszaállítás alkalmával is az elejétől kezdve végig kell vinnie a frissen telepített SharePoint-rendszeren az összes lépést.

A portál mentéseinek tartalma

Az egyéni kialakítású hozzáadott részek visszatételénél tehát az üzemeltető a fejlesztők közreműködésére van utalva, és a szakirodalom csak korlátozott segítséget tud nyújtani; az adatok kezelése viszont minden SharePoint Portal Server rendszeren megegyezik.

A SharePoint a portálok, webhelyek teljes tartalmát SQL-adatbázisokban tárolja: amíg az adatbázisokból rendelkezésre áll friss mentés, nem kell tartani adatvesztéstől. Ha megnézzük egy átlagos portál – az SPS saját mentési eszközeivel, az spsbakup.exe-vel készült – mentésének a célkönyvtárát, öt fájlt találhatunk benne:

- bckpname.XML
- bckpname-Portal_SERV-Portal.SPB
- bckpname-SQL-Portal_PROF.SPB
- bckpname-SQL-Portal_SERV.SPB
- bckpname-SQL-Portal_SITE.SPB

Az XML-fájl nem túl nagy, a mentéshez kapcsolódó leíró információkat tartalmazza. A bckpname-Portal_SERV-Portal.SPB fájlban a portál indexei vannak. Ez általában nem szokott életbevágó fontosságú lenni, mert a legtöbb cégnél csak a portál, illetve az alá beillesztett webhelyek tartalmát járja be a keresőmotor, ami néhány órányi indexeléssel maradéktalanul pótolható.

Az SQL-szerver nevét tartalmazó három SPB-fájllal pedig el is értünk a lényeghez, mert ezekben található maga az adattartalom, SQL-adatbázismentések formájában. Ha kérészei támadnak az üzemeltetőnek, hogy sér-

További információk

SharePoint Resource Kit – Disaster Recovery
<http://www.microsoft.com/technet/prodtechnol/spt/reskit/c2861881x.mspx>

retlenek a fájlok, minden további nélkül visszaállíthatja azokat az Enterprise Managerben. (Így meggyőződhet róla, hogy az adatbázisok SQL-szinten rendben vannak, de ez nem azonos a portál visszaállításával.)

Mentési hibák

Maguk a mentések nem szoktak különösebb nehézséget okozni; amire oda kell figyelni, az a célmappához való hozzáférés beállítása. A SharePointnak és az SQL-szervernek nem kötelező egy gépen tartózkodnia (általában külön vannak), a mentésnek egy megosztott mappába kell irányulnia, és ezt a helyet mindkettőnek el kell tudnia érni. Az XML-fájlt és az indexet a SharePoint-szerver nevében futó folyamat teszi be ide, ellenben az SQL-mentéseket maga az SQL-fiók. Ha valamelyiküknek nincs jogosultsága írni a mappába (jellemzően az SQL fiókja szokott kifelejtődni), akkor nem megy végbe a mentés.

Hibanaplók

Hol kezdjük a hibakeresést, ha sikertelen a mentés vagy a visszaállítás? Természetesen a megfelelő logfájlban. Az SPS alapértelmezésben a „C:\Program Files\Sharepoint Portal Server\Logs” könyvtárban tartja a hibanaplóit. Az „SPS_..._SPSBACKUP.LOG” fájlok a mentés és a visszaállítás lépéseit mutatják, az „SPS_SiteCreation_...log” állományban pedig a portál visszaállítását követhetjük nyomon, némileg átfedésben az „SPSBACKUP” log tartalmával. Az spsbackup.exe eszköz minden használata alkalmával új hibanaplót nyit, míg a „SiteCreation” log egy-egy portál (webhely) teljes történetét (létrehozás és visszaállítások) tartalmazza, így baj esetén a végéről felfelé olvasva lehet a legkönnyebben megtalálni benne a hibát leíró részt.

Visszaállítási hibák

A SharePoint hotfixeinek és szervicsomagjainak telepítésével nemcsak a programfájlok, hanem az adatbázisok is frissülnek, változnak. A rendszer kényes arra, hogy az adatbázis és a programfájlok verziói megegyezzenek az eredeti szerveren (ahol a mentés készült) és a visszaállítás helyén.

A Windows SharePoint Services (WSS) és a SharePoint Portal Server (SPS) nem egymást felülír, hanem egymást kiegészítő komponensként működik (az SPS a WSS-re épül), de mindkettő saját verziószinttel rendelkezik,

a kettőt párhuzamosan kell frissíteni, illetve visszaállításkor mindkét verzióknak a megfelelő szinten kell állnia. A verziók körüli nehézségeket gyakran az tetézi, hogy az adatbázisok különböző okokból nem módosulnak a SharePoint-frissítések telepítésekor. Ilyenkor általában a rendszer mindenféle kóros tünet nélkül működik tovább, csak a katasztrófa-helyzetben jön hidegzuhanyként a felismerés: a visszaállítás nem akar menni, mert a célszer-

fájl és az adatbázis verziója). A táblázatban a SharePoint nyilvánosan letölthető javítócsomagjaihoz tartozó verziószámok láthatók. („A termék verziója” rovatban a Programok Telepítése/Eltávolítása felület „Click here for support information” ablakából vett érték szerepel, a KB-cikk pedig értelemszerűen a javítás részleteit és a letöltési linket tartalmazó forrás.) Ha a verziókülönbségek miatti hibával végződik a visszaállítási kísérlet, akkor a

Windows SharePoint Services

Javitás	A termék verziója	Az OWSSVR.dll verziója	System-Version	A virtuális szerver verziója	KB-cikk
Nincs	11.0.5608.0	11.0.5530.0	6.0.2.5530	6.0.2.5530	
SP1	11.0.6361.0	11.0.6361.0	6.0.2.6361	6.0.2.6361	841876
MS05-006	11.0.6361.0	11.0.6411.0	6.0.2.6411	6.0.2.6411	887981
SP2	11.0.7969.0	11.0.6568.0	6.0.2.6568	6.0.2.6568	887624

A WSS verzióját jellemző fájl:
 „C:\Program Files\Common Files\Microsoft Shared\web server extensions\60\NSAPI\OWSSVR.dll”
 Verzióadatok az adatbázisban:
 <WSS_tartalomadatb.>.SystemVersion;
 <portálnév>_SITE.SystemVersion.
 Az SP2 utáni hotfixcsomagok verziói a négyjegyű szám csoportját növelik.

SharePoint Portal Server 2003

Javitás	A termék verziója	Az mssrch.dll verziója	Schema-Version	Build-Version	KB-cikk
Nincs	11.0.5704.0	11.0.5704.0	2.0.40.0	11.0.5704.0	
SP1	11.0.6715.0	11.0.6715.0	2.0.40.0	11.0.5704.0	841883
SP2	11.0.8216.0	11.0.8216.aa	2.0.40.1	11.0.8216.aa	887623

Az SPS verzióját jellemző fájl: „C:\Program Files\Sharepoint Portal Server\Bin\mssrch.dll”
 Verzióadatok az adatbázisban:
 <portalname>_PROF.PortalSchemaVersion;
 <portalname>_SERV.PortalSchemaVersion;
 <portalname>_SITE.PortalSchemaVersion.
 Az SP2 utáni hotfixcsomagok az alverziószámokban látszanak, ami a négyjegyű számcsoport után következik (11.0.XXXX.aa). Az SP2-től kezdve a „SchemaVersion” utolsó számjegye is változik, de ez nem egyezik meg a már említett alverziószámmal.

veren (újra)telepített WSS és SPS verziói nem egyeznek a mentés adatbázisainak verzióival.

Akkor hogyan és hol lehet meggyőződni róla, hogy ezek rendben vannak-e? Mindig egyezni kell az OWSSVR.dll fájl verziójának és a SystemVersion tábla tartalmának (ez a WSS-komponenshez tartozik), illetve az mssrch.dll verziójának és a PortalSchemaVersion táblában található BuildVersion mező értékének (ezzel SPS-részek verzióját ellenőrizhetjük; kivétel az SPS SP1 állapota, itt nem egyezik a

táblázatból megállapítható, milyen frissítések kellenek (ha kellenek), hogy az újonnan telepített SharePoint-program fájljai összhangban legyenek a mentés adatbázisaival.

A verziókon kívül a SharePoint-telepítés nyelvi változatának is meg kell egyeznie az eredeti és a visszaállított gépeken: ha ez nem teljesül, akkor látszólag sikerülhet a visszaállítás, de az eredmény nem lesz használható.

Pölös Máté

(matepol@microsoft.com) Microsoft Magyarország

FIGYELÜNK ÉS TANULUNK

Hallgatunk a beérkező véleményekre.

Meglehetősen sok visszajelzést kaptunk a megújult TechNet Magazin kapcsolatban személyesen, illetve e-mailben, és a TechNet-eseményeken résztvevők is elláttak minket az első benyomásaikkal a leadott értékelőlapokon keresztül. Alapvetően két gyakran ismétlődő megjegyzéssel találkoztunk: nem elég mély a tartalom, és drága a magazin.

A tartalom

A visszajelzések hatására újra átlapoztuk az előző évek TechNet Magazinjait, és igyekeztünk összehasonlítani azokat a megújult lappal tartalmi szempontból. A korábbi lapszámokban többünket leginkább a sokrészes, végérhetetlen cikksorozatok zavartak. Az még nem is baj, ha van egy-két ilyen, de ha az összes cikk erre a sémára épül, akkor az nem biztos, hogy jó hosszú távon. Szinte pontosan lehetett tudni egy éven keresztül, hogy minden egyes számban találunk egy újabb cikket ugyanarról a témáról, mint az előzőben, és ez nem kedvezett a változatosságnak. (Egy TechNetKlub-tag kolléga szerint: a Magazin számára mindig is olyan volt, mint egy darabokban kiadott könyv.) Az új Magazinban ezzel szakítva új tartalmi koncepciót valósítottunk meg, amiben szintén elfér egy-két cikksorozat, de nem akkora hangsúllyal, mint korábban.

Nagyon sokat gondolkodtunk, hogyan is lehet megtalálni az arany középút. Az újdonságokkal foglalkozunk, vagy a jelenlegi technológiák használatával? Mennyire profi szakembereknek szóljanak a cikkek? Mennyit foglalkozunk fejlesztői, és mennyit üzemeltetői témákkal? Természetesen nem biztos, hogy ezzel a második számmal sikerül megtalálnunk a legjobb arányokat, de

igyekszünk figyelembe venni minden visszajelzést. Ami talán már többeknek szemet szúrhatott, hogy ebből a számból már teljesen hiányzik a Hírháttér rovat, és az újdonságokat áttekintő cikkek is kisebb számban szerepelnek, vagy kellően elmélyülnek az adott területen. Ahol kell, alapozunk, ahol lehet, elmélyülünk. Igyekszünk aktuális üzemeleti hibákról is írni, ebben a Microsoft terméktámogató mérnökei és konzulensei is a segítségünkre vannak.

Röviden összefoglalva: nem szeretnénk olyan cikket látni a TechNet Magazinban, amelyik bármely más újságban is megjelenhetne.

Ismét ingyenes a Magazin!

A lap árával kapcsolatban csak jó hírrel szolgálhatok, ugyanis úgy döntöttünk, hogy a TechNet Magazint továbbra is ingyenesen fogjuk megjelentetni. Nagyon köszönjük azoknak a bizalmat, akik rögtön előfizettek már az első lapszám után, ők ezentúl is biztosan meg fogják kapni a lapot, és természetesen megfelelő kárpótlásban is részesülnek. Igyekszünk kialakítani egy jól átlátható folyamatot arra, hogy hogyan tudnak azok is Magazinhoz jutni, akik nincsenek a korábbi „előfizetők” között, de szeretnének azá válni.

Tanuljuk az új technológiákat

Igaz, hogy még eléggé benne járunk a nyárban, de a Microsoft munkatársai, az MVP-k és az oktatóközpontok egyész nyáron készültek arra, hogy a lehető leghamarabb képben legyenek az új technológiákkal kapcsolatban. Szükség is lesz rá, hiszen ők fogják majd átadni a most megszerzett tudást és tapasztalatot az ősszel induló TechNet-eseményeken.

Az őszi TechNet-programokat úgy fogjuk összeállítani, hogy aki komolyan foglalkozik majd a szakmai fejlődésével, az év végére minden szempontból értékes tudást szerezzen, és azt munkája során maximálisan fel is tudja majd használni. Folytatjuk az idén megkezdett online előadásokat, webcastokat, és több energiát fordítunk arra is, hogy még több szakmai információ legyen elérhető a magyar TechNet Portálon, nem csak blog formájában. Terveink szerint őszi végére teljesen megújul a TechNetKlub.hu is; új külsővel, kereshető archívummal, de továbbra is levelezőlista alapokon fog működni. Megjelennek majd új listák is (az első a sorban már elkészült, ez a Microsoft Dynamics CRM-mel kapcsolatos problémák megoldására szakosodott), illetve van néhány régebbi lista, amire már ráférne egy kis vérfrissítés.

Továbbra is arra biztatunk mindenkit, hogy ha kérdése, észrevétele, javaslata van a TechNettel kapcsolatban, vagy tanácsstalan, hogy kihez is fordulhatna Microsofttal kap-

TechNet
MAGAZIN MÉLYVÍZ, CSAK ÚSZÓKNAK!

Microsoft®
Office

DEP ÉS ASLR
A Vista memória- és programvédelmének néhány vonatkozása

INTEGRÁLT INNOVÁCIÓ-TÖBB MINT A RÉSZEK EGÉSZÉ:

FESTŐK ÉS BÖDÖNÖK
A skálázhatóság és a párhuzamosság kérdései

MITŐL RENDSZER?

ÚJ TÁBLÁK AZ IRODÁBAN

SQL SERVER 2005 ANALYSIS SERVICES
Példák az üzleti elemzést támogató legfontosabb funkciókra

AMIKOR A RÉGI TALALKOZIK AZ ÚJJAL

Microsoft TechNet

csolatos szakmai kérdésekkel, az írjon egy e-mailt a techneth@microsoft.com címre. Terméktámogatást ezen a címen nem vállalunk, de minden másban örömmel segítünk.

Ha pedig valakiben megvan a szükséges kitartás, kedv és szakmai tudás, az jelentkezzen a TechNet-cikkírók vagy -előadók csapatába, szívesen látjuk!

Budai Péter

(i-pbudai@microsoft.com) Microsoft Magyarország



IT-BUSINESS

FIZESSEN ELŐ!

Az előfizetés 5 egyszerű módja:



Tel.: 06-1-888-3424



Fax: 06-1-888-3499



E-mail: terjesztes@vogelburda.hu



Internet: www.it-business.hu



ITmédiaBOLT: 1054 Budapest,
Bajcsy-Zsilinszky út 60.

Nyitva tartás: hétfő-péntek: 9-20, szombat: 10-16

Csak IT-BUSINESS előfizetőknek!

- a megjelenés előtt a magazin teljes tartalma elérhető a weboldalon
- teljes körű hozzáférés az IT-BUSINESS archívumhoz – cikkek 2003-tól

Tekintse meg mutatószámunkat weboldalunkon!

Igen, megrendelem az IT-BUSINESS-t egy évre - 20 %-os kedvezménnyel - 16 668 Ft-ért!

Igen, megrendelem az IT-BUSINESS-t fél évre - 15 %-os kedvezménnyel - 8 856 Ft-ért!

Kézbesítési adatok:

Cégnév:

Megrendelő neve:

Beosztás:

Irányítószám:

Település:

Utca, házszám:

Telefon:

E-mail:

Számlázási adatok:

Cégnév:

Irányítószám:

Település:

Utca, házszám:

IT-B-ICT

Kiadónk az Ön személyes adatait az 1995. évi CXIX. adatvédelmi törvény szerint kezeli. Adatairól Kiadónknál, a következő címen érdeklődhet: Vogel Burda Communications Kft. (1077 Budapest, Kéthly Anna tér 1.). Amennyiben nem járul hozzá, hogy ajánlatainkkal megkeressük, kérjük itt jelezze:

Töltse ki és faxolja el a 06-1-888-3499-es faxszámmal!



Hol tölti a novembert?

2006. november 14-17. Barcelona, Spanyolország

Négynapos konferencia mély, szakmai témákról, kifejezetten IT szakembereknek - a Microsoft hamarosan megjelenő szoftvereiről. Ismerje meg elsőként az új technológiákat!

Spóroljon meg 300 Eurót – Regisztráljon 2006. szeptember 29. előtt!
Az akciós részvételi díj €1,395 + 16% ÁFA (A regisztrációs kód: EEA11340)

További részletekért látogasson el az esemény weboldalára:

www.microsoft.com/europe/teched-itforum/



Microsoft
Tech·Ed IT Forum

Microsoft TechNet