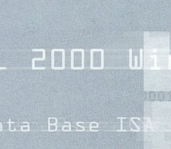
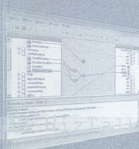


SQL SERVER 2000
BIZTALK SERVER
INTERNET SECURITY AND ACCELERATION SERVER
HOST INTEGRATION SERVER 2000
EXCHANGE 2000 SERVER
COMMERCE SERVER 2000
WINDOWS 2000 DATACENTER

tech.net

A MICROSOFT MAGYARORSZÁG SZAKMAI MAGAZINJA

KÜLÖNSZÁM



2000.10.



Jubiläum

A NetAcademia Kft. történelmében többszörös jubileumot ünnepelhetünk e kiadvánnyal. Először is túl vagyunk a Tech.Net szakmai magazin első számán, másodszor elérkezett a legelső különszám. A Microsoft háza táján is jubileumot ünnepelnek: 25 éve alakult a cég. Hiába no, minden nagy cégnél számontartják az eddig megtett utat :). Valószínűleg nagyobb hatást gyakorol a Microsoft az informatikai iparágra, mint mi, újságíróként tetszelgő rendszermérnökök. Ezért e különszám – bármilyen fájó – nem rólnak szót, hanem valami újról, valami forradalmiról. A technológiai elemek évek óta rendelkezésre álltak, de most állt össze az egységes jövőképe az Internetről: rivaldafényben a .NET stratégia. Döbbenetes élmény csak részt venni is e rohanásban. Azt hittük, ez az év már nem tartogat meglepetéseket, hisz alig fél éve, hogy megjelent a Windows 2000. Tévedtünk. A .NET nagyobb változásokat hoz, mint egy új operációs rendszer: a tét jelenleg a mindent átható Internet szabványainak kidolgozása, a jövő alapjainak lerakása. A Microsoft ezen alapok lerakásában segít átgondolt, egységes .NET kiszolgálótermék-családjának egyidejű megjelenítésével. Forradalmi változás ez, hisz minden termék teljes gőzzel, olykor 180 fokos fordulattal az Internet felé fordul. A Windows 2000 már eddig is bizonyította Internet-elkötelezettségét, de most minden egyes BackOffice termék előrébb lép e folyamatban. Az SQL Server 2000 teljeskörű XML támogatással és adatbányászati képességekkel (Data Mining) virít, az Exchange 2000 pedig csak annyira hasonlít az Exchange 5.5-re, mint a Windows 2000 a Windows 3.1-re - vagyis alig. Kizárólag kompatibilitási okokból tartották meg benne az eddigi üdvöskét, az X.400-at, a MAPI-t, 100.000%-ban újraírták az Outlook Web Accesst, az adatbázismotort, a fűrtámogatást, s mindehhez a hiba a tortán az XML.

Apropó XML. Aki még nem ismeri e régi-új leírónyelvet, legjobban teszi ha mihamarabb fejest ugrik bele: EZ a jelen évezred szabványos adatátviteli formátuma. Mindenre képes, még arra is, amire nem, hisz rugalmasan bővíthető, önelírő nyelvként arra hajlik, amerre éppen szükséges. Ez a csoda fog minket átvinni a papír- (és ember?) mentes iroda korszákába, ahol a vállalatok szoftíngépei szépen eltársalognak egymással, hisz végre értelmezni képesek egymás adatformátumát. Összetettebb esetben nélkülözhetetlen BackOffice komponenssé válik a BizTalk Server,

amelyet egy mondatban úgy tudnék jellemezni, mint XML átalakítót (trafó) és útválasztót, mely XML tolmácsként végre lehetővé teszi, hogy szót értsenek egymással SAP és egyéb, egyedi vállalati rendszereink. Az XML egyértelműen a harmadik évezred gazdasági fejlődésének motorja.

A robbanásszerű fejlődés előtt álló e-kereskedelem nem lehet meg olyan megoldás nélkül, amely a terhelés növekedésével szinte lineárisan növekedő teljesítményt nyújt. E téren két új Windows 2000 változat szolgáltatásait lehet igénybe venni: a DataCenter és AppCenter Servereket. Míg DataCenter Serverrel valószínűleg földi halandó nemigen fog találkozni, mert a termék csak olyan hardverekkel kerül majd forgalomba, melyek kiállták a négykilences próbát (99,99%-os rendelkezésre állás, azaz évente maximum 52 perc 33 másodpercnyi leállás !!), és áruk ennek megfelelően prohibítvó, addig az AppCenter Server majd minden végre fűszeres számára elérhető áron nyújt méretezhető megoldást kínzó növekedési problémájára. Biztonság, biztonság, biztonság. Mindaddig nem volt a Microsoft termékpalettáján olyan komponens, melyet a webre kített kiszolgálókat meg lehetett volna védeni a jóságos hackerek gonoszkodásaitól. A félelmetes nevű ISA Server 2000 valódi, bevizsgált tűzfal, mely csomag-szűrő és alkalmazás-tűzfal funkciókkal egyaránt rendelkezik, sőt képes néhány tipikus támadás észlelésére, és a megfelelő személyek riasztására.

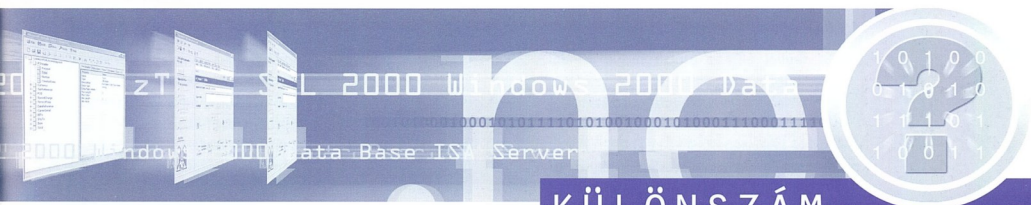
A jó öreg SNA Server-t is új változat váltja fel, itt a Host Integration Server 2000, mellyel olyan úgynevezett „legacy”, vagy szabad fordításban „Khm. Miért is nem fejlesztik?” rendszereket is be lehet vonni a vállalat információs vérkeringésébe, melyek pokoli sok pénzbe kerültek annak idején, eszméletlen sok adat van bennük, de tökéletesen reménytelen arra várni, hogy gyártójuk majd XML támogatással, vagy adatbányászati képességekkel egészíti ki őket. Mély tisztelet az elenyészően ritka kivételnek :)

A Microsoft elkötelezettségét mi sem jelzi jobban, hogy a .NET bejelentésével egyidőben a legfontosabb partnerek a felkészültség oly magas fokon állnak, hogy azonnal képesek voltak eleget tenni annak a felkérésnek, hogy mindegyikük ismertesse egy-egy cikkben a saját tevékenységi köréhez legközelebb álló .NET terméket – ennek kézzel fogható eredménye e különszám. Köszönet tehát a Protomix, a Synergon, a Novosys, a Minor, a HMS, az ICON és a Montana rendszermérnökeinek a kiváló munkáért!

Fóti Marcell
főszerkesztő

MCSE+I, MCDBA, MCT
marcellf@netacademia.net





tech.net

KÜLÖNSZÁM

Szerkesztőség
Főszerkesztő: **Fóti Marcell**
marcellf@netacademia.net
Főszerkesztő-helyettes: **Fülöp Miklós**
mick@netacademia.net
Szerkesztőség címe:
1105 Budapest, Ihász utca 13.
Tel.: 263-2732
technet@netacademia.net
Nyilvános levelezési lista:
technet@lyris.netacademia.net

Kiadja a **Microsoft Magyarország**
1031 Budapest, Graphisoft park 3.
Tel.: 437-2800
A kiadásért felel:
Arany Tóth László
va-laarto@microsoft.com

Terjeszti a **NetAcademia Kft.**
terjesztési előfizetési információ:
Tel.: 263-2732
terjesztes@netacademia.net
Megjelenik havonta, ára 899 Ft

Előfizethető megrendelőlevélben a szerkesztőségnél:
1105 Budapest, Ihász utca 13.
Fax: 261-7245
<http://technet.netacademia.net/subs>

Hirdetésfelvétel:
Bársonykalapács Marketing
Felelős: **Udvarev Rita**
Tel./Fax: 214-0923
velvethammer@elender.hu
1027 Budapest, Fő utca 67. V. 1.
Grafikai tervezés, kivitelezés,
nyomdai előkészítés:
Bársonykalapács Marketing
Művészeti vezető: **Balogh Zoltán**

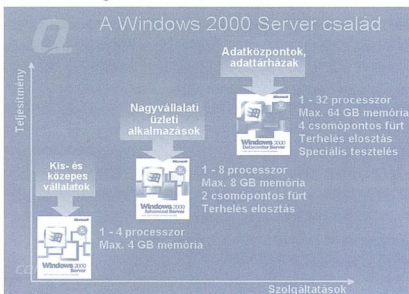
Nyomda:
Partner's 2000 Kft.
1124 Budapest, Sion lépcső 7.
Felelős nyomdász: **Galamboš Sándor**

ISSN 1586-5185

Windows 2000 DataCenter	3. oldal
Application Center	5. oldal
Exchange 2000 Server egykor és most	7. oldal
Exchange 2000 Conferencing Server	15. oldal
SQL Server 2000 új adminisztrációs eszközei	20. oldal
SQL – az XML támogatása	23. oldal
SQL Server 2000 újdonságai	29. oldal
Particionált nézetek az SQL Server 2000-ben	33. oldal
XML séma	36. oldal
OLAP	37. oldal
BizTalk Server	43. oldal
BizTalk Orchestration	47. oldal
ISA Server	49. oldal
HIS 2000 Server	55. oldal
HIS 2000 Server új lehetőségei	60. oldal
Commerce Server 2000	63. oldal



Idén február 17-én jelent meg a Windows 2000 Server család első két tagja. A család három tagból áll, az utolsó tag a Windows 2000 DataCenter Server most jelenik meg. A Windows 2000 megbízhatósága, rendelkezésreállása messze felülmúlja a Windows NT hasonló képességeit. Amikor operációs rendszert választunk, megbízhatóságon és rendelkezésreálláson túl a teljesítményparamétereket is kiemelten kezeljük. A Windows 2000 megjelenésével soha nem látott teljesítményadatokat produkált a rendszer. Hogy mást ne is említsünk, a TCP-C eredmények első tíz helyén Windows 2000 szerepel. A Windows 2000 Server család zászlóshajója a *Windows 2000 DataCenter Server* még tovább emeli a szintet megbízhatóság, teljesítmény, méretezhetőség tekintetében.



Windows 2000 választási lehetőségek

Rögtön le kell szögeznünk, hogy ez a Microsoft operációs rendszer jelentősen eltér az eddig megszokott operációs rendszerektől abban, hogy ezt csak és kizárólag a kiemelt hardvergyártóktól lehet beszerezni. A Windows 2000 DataCenter sokkal több, mint a telepítő CD-n lévő bitek összessége. Ez az operációs rendszer előre tesztelt és módosíthatatlan hardverkonfigurációt, folyamatosan elérhető szervizháttérrel és rendszerintegrációt jelent.

Windows 2000 DataCenter Server

Nézzük röviden a szoftver paramétereit. Köztudott dolog, hogy pusztán a processzorok számának növekedésével egy ponton túl már nem lehet a teljesítményt növelni. Ahhoz, hogy egy rendszerből ki tudjuk hozni a megfelelő teljesítményt, hatékony algoritmusokra van szükségünk. Windows NT alapokon maximum 8 processzoros rendszert tudunk építeni. A processzorszám növelésével a teljesítmény nemhogy nőne, inkább letörik.

A Windows 2000 DataCenter Server-rel 8, 16 sőt akár **32 processzoros** rendszert tudunk építeni úgy, hogy a rendszer teljesítménye a processzorok számának növelésével lineárisan növekszik.

Az olyan adatbáziskezelők, mint például az SQL 2000 akkor működnek a legnagyobb teljesítménnyel, ha feladataikat minél kevesebb lemezbizlettel, a memóriában

végezhetik el. A Windows 2000 DataCenter Server ideális platformja a nagyvállalati alkalmazásoknak, hiszen **64 gigabájt** memóriát tud kezelni.

Egy vállalat szempontjából kritikus alkalmazások rendelkezésreállítását úgy is tudjuk növelni, hogy az alkalmazást párba kapcsolt gépeken, úgynevezett fűrtökön futtatjuk. A Windows 2000 DataCenter Server-t használva **négy darab** fűrtbe kapcsolt számítógépet tudunk azonos feladatok elvégzésére beállítani. Ezek a számítógépek a Compaq üvegzászló hálózati technológiájának köszönhetően akár 10 kilométer távolságban is lehetnek egymástól.

A hardver

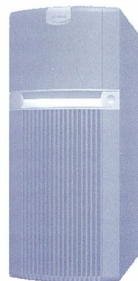
Az eddig elmondottakból tisztán körvonalazódik, hogy a Windows 2000 DataCenter esetében már régen nem PC kategóriájú gépekről beszélünk, hanem olyan számítógépekről, melyek működésükben, üzemeltetésükben, teljesítményükben a mainframe gépek kategóriájába esnek. A Compaq és a Microsoft közötti szoros együttműködésnek köszönhetően a Compaq elsőként jelenik meg olyan számítógépekkel, melyek megfelelnek a nagyon szigorú tesztelési feltételeknek. A minősítés során egy teljes rendszert 30 napon át folyamatosan, erős terhelésnek kitéve tesztelnek. A tesztelés során nemcsak magát a hardvert, hanem a működéséhez szükséges meghajtóprogramokat is tesztelik. Extrém terhelési situációkban, szűkös szabad memória környezetekben tesztelve a rendszert garantáltan kiszűrhető az esetleges meghajtóprogram-hiba. A meghajtóprogramban lévő hiba azért lehet végzetes a rendszer számára, mert a kernelszinten fut, ahol semmilyen kódellenőrzés nem zajlik. Ha tehát hibás, akkor az könnyen rendszerösszeomláshoz vezethet. A számítógép rendszerben bármilyen komponens megváltoztatása a teljes tesztelési folyamat újrajrkezését jelenti. A Microsoft által minősített hardverkomponenseket a HCL (*Hardware Compatibility List*) tartalmazza. Az a néhány rendszer, mely a Windows 2000 DataCenter Server futtatására jogosult, az úgynevezett Gold HCL-en található. Korábban már említettük, de fontossága miatt még egyszer hangsúlyozzuk, hogy a Windows 2000 DataCenter Server-t csak hardverrel együtt vásárolhatjuk meg.

A Compaq Proliant szervercsalád új zászlóshajója a **Proliant ML 770**, amely a DataCenter Server minden lehetőségét kihasználja. Maximális kiépítésben 32 darab Pentium III Xeon processzort és 64 Gbyte SDRAM memóriát tartalmazhat.

A Compaq Intel alapú zászlóshajója

Eddig nem beszéltünk róla, de tudnunk kell, hogy ez a kiszolgáló a perifériák kezelésében is új rekordot állít fel a maga 96 darab 64 bites PCI csatlakozójával. A szerver alapvetően két kiserelésben rendelhető:

- 16 processzor, 32GB RAM, 48 PCI csatlakozóval és redundás tápegységgel, valamint



Microsoft
BackOffice

☞ 32 processzor, 32 GB RAM, 96 PCI csatlakozóval és redundáns tápegységgel

Az ML 770 és szerver a következő komponensekből épül fel.

- ☞ 32 vagy 16 Pentium III Xeon 700 MHz processzor
- ☞ 2 MB L2 Cache processzoronként
- ☞ 32 GB memória (32P modell) vagy 16 GB memória (16P modell). A 64 GB memória a következő felévtől
- ☞ Két darab Compaq ProLiant DL360 szerver
- ☞ 24 portos Ethernet Switch
- ☞ StorageWorks BA35A-BA
- ☞ 96 bővíthető (64 bit/33 MHz Hot Plug PCI)
- ☞ Négy (32P Modell) illetve kettő (16P Modell) integrált NC3131 Fast Ethernet hálózati kártya 64 PCI Dual Port 10/100
- ☞ Négy (32P modell) vagy kettő (16P modell) integrált Ultra3 SCSI vezérlő
- ☞ 16 (hot plug redundant) tápegység (10 darab a 16P modellnél)
- ☞ Hat darab üzem közben cserélhető redundáns hűtőventillátor

A szerverek tároló diszkalrendszerei a hosszú ideje piacon lévő megbízhatóan működő, több terabájt adat kezelésére képes StorageWorks MA 8000 és MA12000.



Datacenter nincs adattárolás nélkül.

Az ML 770 működéséhez háromfázisú tápellátásra van szükség, ami azt jelenti hogy ez a szerver ritkán fog a légkondicionált szerverszobák falai közül kilépni. A tápegységek menet közben cserélhetők.

Ekkora elektromos teljesítmény felvétele jelentős hőterheléssel jár együtt. A rendszernek ezért fontos részét képezik a ventilációs modulok. Mondani sem kell, hogy ezek is reptében cserélhetők. Ha pedig a csere nem lehetséges, akkor az esetleg kiegészítő modul feladatát a többi ventilátor átveszi úgy, hogy automatikusan nagyobb fordulatszámra kapcsolnak.

Ha alaposan átolvassuk az ML 770-es paramétereit, biztos megakad a szemünk a felsorolásban negyedik helyen lévő „DL 360 szerver” tételén. Aki ismeri a Compaq termékskáláját tudja, hogy ezek a legkisebb, egész pontosan 1U magas szerverek. Ezeknek a szervereknek a feladatát ebben az esetben a nagy testvér (ML 770) felügyeletének ellátása. Nem mindennapi megoldás az, amikor egy nagy szerver házában egy kis szerver is működik. Ezek alapján állíthatjuk, hogy a

Compaq az ML 770-es piacra dobásával megalkotta azt a maximumot, amit ma Intel platformon meg lehet alkotni.

Egy új fogalom – particionálás

A particionálás fogalmát eddig a merevlemezrendszerek felosztása kapcsán használtuk. Az ML770-es particionálása lehetővé teszi, hogy egy fizikai gépen belül több, egymástól teljesen függetlenül működő logikai gépet alakítsunk ki. A particionálás során bizonyos számú processzort összefogunk (2x16P, 4x8P) vagy (1x16P, 2x8P) és így belőlük képezzük a virtuális számítógépeket. Egy particionált számítógép esetében annyiszor kell a Windows 2000 DataCenter Server-t telepíteniünk, ahány particiót kialakítottunk. Minden partició egymástól független életet él. Az egyiket futhat például az SQL 2000, a másikon pedig az Exchange 2000. Másik felhasználási lehetősége a particióknak a fűrtözés. Ebben az esetben úgy alakítottunk ki fűrtöket, hogy azok fizikailag egy gépen belül vannak.

Datacenter, de kisebb léptékben

Olvasván az ML770 szerveren lévő processzorok számát és a memória méretét mindenkinek megdobban a szíve. Ilyen teljesítményű gépekről eddig csak az egyetemi kutatószokra épített szuperszámítógépekkel kapcsolatban hallottunk, és ezek a gépek most részei lehetnek mindennapi életünknek. A józanodás pillanatai természetesen akkor következnek, amikor ezeknek a szervereknek az árcímekéjét meglátjuk.

A Compaq tisztában van azzal, hogy nem mindenkinek van szüksége akora teljesítményre, mint amekkorát ebből a gépből ki lehet hozni, ezért a ProLiant termékskalád eddigi csúcscategóriájú tagját, a ProLiant 8500-as gépeket is bevonta a Datacenter Server tesztekbe. Ennek eredménye, hogy az Industry Standard Architecture alapján kifejlesztett 8 utas ProLiant 8500 gépek egyes tesztelt modelljeire is telepíthetjük a Windows 2000 Datacenter-t. Ezzel a kombinációval igaz, hogy kisebb teljesítmény mellett, de megkapjuk a DataCenter Server többlétszolgáltatásait, illetve az oly fontos 99,99%-os garantált rendelkezésreállást.

Alkalmazási területek

Elsőszámú jelöltjei ennek a platformnak a nagy teljesítményt igénylő adatbázisszerverek, illetve az ezekre épülő vállalatirányítási rendszerek (SAP).

Az informatikai trendek figyelve a másik a felhasználási terület a szerverkonsolidáció.

Az Internet és az elektronikus kereskedelem komoly teljesítményigénye és a rendszerekkel szemben támasztott magas rendelkezésreállási igények szintén könnyedén kiszolgálhatók az ML770-el.

Tarsoly Balázs, balazst@microsoft.com



Microsoft Application Center

Hogyan üzemeltessünk szerverfarmot...

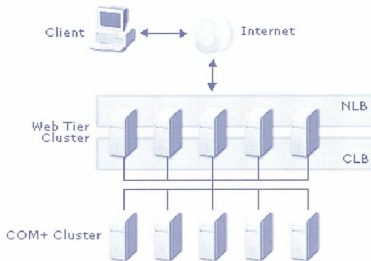
Gyorsuló világunkban lassan értelmét veszti klasszikus PC-s alkalmazásokat építeni: az egyépes megoldások ideje lejárt – ha valami nem képes kommunikálni, elavult. Ha a következő verzió nem internetes nyílt szabványokat használ, ne is dobjuk piacra... Az elosztott alkalmazások építése, üzemeltetése sok nyűggel jár: többek között nehezen megoldható a szabályozott telepítés, a monitorozás, a folyamatos fejlesztés, új verziók telepítése vagy a rutinszerű kapacitás-növelés. Ezen gondok megoldására született cikkünk tárgya, a Microsoft Application Center szoftver.

Írásunk elsősorban műszaki döntéshozóknak, döntéshozóknak szól. Célunk az, hogy elolvasás után alapszintű termékismeretre tegyenek szert. Ez alapján saját rendszereikben új, igazán merész elképzeléseiket is megvalósíthatják, és sokkal könnyebben és olcsóbban üzemeltethetnek.

Az Application Center környezete

Egy mai, egyszerű internetes alkalmazás architektúráját a következőképpen vázolhatjuk fel:

A böngészős kliensprogram HTTP szerverrel kommunikál, mely mögött alkalmazásszerveres üzleti logika működik. A háttérben adatbázisok, vagy egyéb háttéralkalmazások futnak. Persze, ha az alkalmazást üzleti környezetben, az ennek megfelelő rendelkezésreállási és adatbiztonsági elvárásokkal üzemeltetjük, nem hagyhatjuk ki a megfelelő redundáns elemeket sem:



Ekkor a böngészős kliensprogram már hálózati terhelésmegosztásban (*Network Load Balancing, NLB*) működő HTTP szerverekkel kommunikál, és az alkalmazás kiszolgálók is többszörözöten működnek, komponens szintű terhelésmegosztással (*Component Load Balancing, CLB*) osztva meg egymás között a feladatokat. A háttérben az adatbázisok vagy háttéralkalmazások fürtözött struktúrában működnek (*Cluster*).

Az éles megvalósításokat még inkább bonyolítja, hogy gyakran redundáns hálózatot, tűzfalakat és különböző biztonsági szintű zónákat is figyelembe kell vennünk. Szerencsére ezek a rendszerelemek az Application Center – illetve az alatta futó megfelelően konfigurált Windows 2000 Server-ek – számára transzparensnek, így a felépítésük után nem kell rájuk külön figyelmet fordítanunk.

A szerver működése

A működési környezet által meghatározott keretek között az Application Center vezérlésével a komponensek egységiesen jelennek meg a külvilág felé.

Felhasználói oldalról – Internetes protokollokat használó – böngészős kezelőfelülettel találkozunk. Ez állandóan rendelkezésre áll és meglepően nagyszámú párhuzamos kapcsolatot képes kiszolgálni. Az állandó rendelkezésreállás a redundáns gépelrendezésnek, a nagy kiszolgálókapacitás pedig az egyes gépek lineárisan összeadóó teljesítményének köszönhető. Üzemeltetői szemmel a rendszer rendezett, különböző szakfeladatokat ellátó szervercsoportokat látunk:

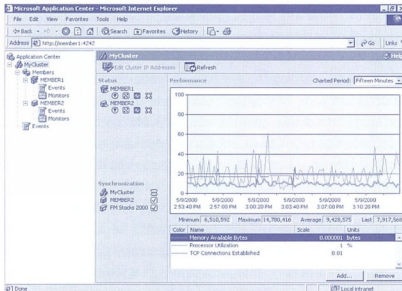
- ☞ Legnagyobb számban a kérésekre HTML lapokat, grafikákat, statikus webes tartalmat szolgáltató szervereket találunk. Ezek a gépek egy IP címen, hálózati terheléelosztással működnek, és az Application Center gondoskodik tartalmuk szinkronban tartásáról.
- ☞ Alkalmazástól függően a webszerverek mögött (*vagy magukon a webszervereken*) COM+ alkalmazásokat futtató szervereket találunk. Ezek a szerverek komponens szintű terheléelosztásban működnek, az alkalmazás futtató környezet egyenletesen osztja el a rendelkezésre álló hardvereken a terhelést. Itt fontos megjegyeznünk, hogy a webes alkalmazások kifejlesztésénél nem szükséges speciális szabályok betartása, a komponens szintű terheléelosztás szakszerű fejlesztés esetén teljesen transzparens a szoftverköd számára.
- ☞ A háttérben szintén nagy megbízhatóságú, fürtözött szerverek által működtetett adatbázisok, tárolórendszerek, vagy nagygépes alkalmazások állnak.



Microsoft
.net

Az egész parkot egy irányítófelületről kezeljük, melynek két alapvető szolgáltatását használjuk:

- Az adminisztrátori eszközökkel új gépeket, alkalmazáskomponenseket vehetünk fel, terhelésáthelyezéseket végezhetünk, vagy rendszert particionálhatunk.

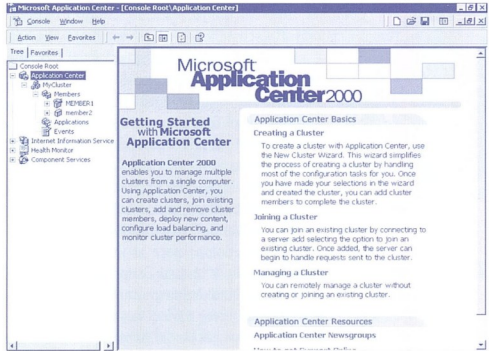


- A naplózó- és elemzőszökezzel való időben vagy utólag figyelhetjük rendszerünk működésére illetve az alkalmazásunkat használókra jellemző statisztikákat.

A háttérben két további szolgáltatás működik, mellyel közvetlenül foglalkozni nem kell, de nem szabad figyelmen kívül hagynunk: a folyamatos replikáció konszisztens tartja a szervereken tárolt redundáns adatokat, a belső névfeloldás pedig a gondoskodik arról, hogy az egész struktúra kívülről egységesen látszon, miközben belül a teljesítmény egyenletesen osztható el a szerverek között.



Ezekután például ha a szerverek számát növelni akarjuk, új, alapelemekkel feltöltött gépet kötünk a hálózatba, és regisztráljuk az Application Center alkalmazásban. A környezet automatikusan gondoskodik a szükséges komponensek telepítéséről és a szerver szoftverének feltöltéséről, konfigurálásáról, üzemeltetéséről.



A teljes adminisztrátori környezet a már megszokott négy arcot mutatja:

- MMC-s, varázslós felülettel végezhetjük el a napi rutin-feladatokat
- parancssoros felülettel futhatunk végig a már igazán begyakorlott lépéseken
- weben keresztül avatkozhatunk be a rendkívüli helyzetekben
- a nyílt API-kon keresztül saját szoftvereket is illeszthetünk a felület mellé

Összefoglalás

Az Microsoft Application Center 2000 tehát egy olyan szoftverkönyezetet ad, mellyel nagyszámú gépből összeállított, internetes szabványokon alapuló, magas rendelkezésre állású szerverfarmot üzemeltethetünk. Ez az eszköz a Microsoft első olyan terméke, amely egyértelműen a .NET stratégia mellett kötelezte el magát: az Internetes piacnak szánt szolgáltatások megbízható, gazdaságos működtetéséhez elengedhetetlen eszköz, mely versenyelőnyre válthatja a professzionális felhasználók előtt.

Probojcssevity Ottó, otto@icon.hu





Amióta 1996-ban megkezdte pályafutását, azóta a Microsoft Exchange Server széles körben ismertté vált és kiszolgálóoldali (*back-end*) üzenetközvetítő terméként kerül felhasználásra. Számos, ha nem a legtöbb ügyfél gondolkodásában az Exchange is ügyfélprogramjai, ideértve a Microsoft Outlook-ot és Microsoft Outlook Express-t is, ma már egyenértékűnek számítanak az e-mail-lel. Az Exchange azonban több, mint pusztán levelező kiszolgáló.

Ha utánanézzünk az „Exchange” szónak, akkor azt a meghatározást találjuk, miszerint a csere „kölcsonös adás és vétel” ill. „olyan hely, ahol tárgyak vagy szolgáltatások kicserélésére kerül sor”. Mindkét meghatározásból kiviláglik, hogy a csere fogalma jóval többet takar üzenetek küldésénél és fogadásánál, még akkor is, ha ezek az üzenetek díszes formát ölthetnek vagy hozzájuk csatolt dokumentumokat, képeket és egyéb objektumokat foglalhatnak magukban. A szó értelmezése szerint a csere sokkal inkább hasonlít egy piactéren vagy városközi ülésen, mintsem postahivatalban zajló eseményre.

Lehet, hogy az eddigi bevezető üres fecsegésnek tűnik, lényegében azonban arról van szó, hogy az Exchange, mint termék mindig is megfelelt a neve mögött rejlő elképzelésnek – nemcsak úgy, hogy támogatja az elektronikus levelek „kölcsonös adását és vételét”, hanem azért is, hogy elősegíti az információk kereskedelmét olyan mechanizmusokon keresztül, amelyek köre a naptáraktól és hirdetőablaktól kezdve magán- és közönyvtárakon át egészen a testreszabott alkalmazásokig, a feladatkezelésig, dokumentumtárolásig és távoli elérésig terjed.

A szabadon kicserélt információk koncepciója márpedig sehol másutt nem nyilvánvalóbb, mint az Exchange megjelenés előtt álló új verziója esetében, amely a fejlesztés szakaszában a Platinum fedőnevet kapta. Ebben a legújabb változatban, amely hivatalosan Microsoft Exchange Server néven ismert, az Exchange nemcsak elektronikus levelek, hanem hangposta, („streaming”) videofolyam dokumentumok és webtartalom továbbítását is támogatja, és lehetővé teszi az azonnali üzenetküldést, videokonferenciát és tárgyalások lebonyolítását.

Röviden, az Exchange valamennyi verziójában mindig is az együttműködés támogatója volt, hajtőerőként szolgált mindazoknak a mindennél fontosabb (*habár gyakran bizonytalanul definiált*) tevékenységeknek a vonatkozásában, amelyeket együttesen a csoportmunkát támogató program kategóriájába soroltak. Az Exchange 2000 azért terjeszti ki a támogatást a csoportos tevékenységekre, hogy felőleje az olyan jól bevált, emeltszintű és újonnan kifejlesztett technológiákat, amelyek a korábbi változatoknál még ismeretlenek voltak.

A Microsoft Exchange Server a BackOffice csomag kommunikációs, üzenetküldő és csoportmunka-szolgáltatásokért felelős része. Egyesíti a hagyományos e-mail rendszerek továbbfejlesztett szolgáltatásait, csoportidőbeosztás-készítő, elektronikus úrlapkezelő és általános csoportmunka-alkalmazási lehetőségekkel. Tervezésekor célul tűzték ki, hogy az elektronikus levelezést könnyebben használhatóvá, könnyebben felügyelhetővé, megbízhatóbbá és még rugalmasab-

bá tegye a hálózat méretétől függetlenül. A beépített csoportmunka-alkalmazásoknak köszönhetően a kisebb és nagyobb szervezetek is javíthatják és automatizálhatják a kulcsfontosságú üzleti folyamataikat. A Microsoft Exchange 2000 Server-t – amely szorosan integrálódik a Microsoft Windows 2000 operációs rendszerbe – úgy tervezték, hogy megfeleljen bármilyen méretű vállalkozás levelezési és csoportos munkavégzési igényeinek – a kis cégektől a nagy, területileg széttagolt vállalkozásokig. Ügyfélszoftverével, a Microsoft Outlook 2000-rel együtt az Exchange rendkívül megbízható, méretezhető és könnyen kezelhető üzenetküldő és csoportos munkavégzési infrastruktúrát biztosít. Innovatív, új webes tárolórendszere az Exchange megbízhatóságát és méretezhetőségét a Web elérhetőségével és nyíltságával ötvözve kiterjedt adattárházat és üzleti alkalmazási platformot teremt. Az Exchange 2000 Conferencing Server (*konferencia szerver*) adat, audio és videó-konferencia szolgáltatási platformot biztosít, megteremtve az együttműködés új módszereinek alapjait.

Továbbfejlesztett üzenetkezelési és együttműködési platform

Az Exchange 2000 megbízható, méretezhető, és könnyen kezelhető infrastruktúra, amely a hét minden napján, huszonnégy órában elérhető üzenetkezelési és együttműködési szolgáltatást biztosít alacsony birtoklási költség (TCO) mellett. Az Exchange 2000 Server, fokozott megbízhatóságot, méretezhetőséget és megnövelt teljesítményt biztosít alacsonyabb birtoklási költség mellett, ami az üzenetkezelési, együttműködési és hálózati erőforrások egységes kezelésének köszönhető.

Integráció a Windows 2000-re

Active Directory integráció. Az Exchange 2000 teljes mértékben kihasználja a Windows 2000 Active Directory szolgáltatásának előnyeit, lehetővé téve a rendszergazdák számára egy vállalati címter létrehozását, egyetlen pontban biztosítva az összes felhasználó, felhasználó-csoport, jogosultság, konfigurációs adat, hálózati bejelentkezés, fájl és webszintű megosztás, stb. kezelését. Az Active Directory egy vállalati szintű címterrel való integráció, amely rendkívül méretezhető, szabványos Internet-technológiákból építkezik, és operációs rendszer szinten teljesen mértékben integrálva van az Exchange-dzsel. Az Active Directory lehetővé teszi az összes felhasználó, üzenetkezelő és hálózati erőforrások közös pontból történő kezelését – beleértve az e-mail és postafiók információkat, konfigurációs információkat, fájlokat, perifériaeszközöket, számítógépes kapcsolatokat, adatbázisokat, webhozzáféréseket, felhasználókat, egyéb tetszőleges objektumokat és szolgáltatásokat. Az összes Exchange 2000 címter-információt (*beleértve a felhasználókkal, postafiókokkal, szerverekkel, telephelyekkel, és egyedi címzettekkel kapcsolatos információkat is*) az Active Directory tárolja. Míg a rendszergazdák az egységes adminisztráció előnyeit élvezhetik, a tudásanyagok foglalkozó alkalmazottak, akik hozzászoktak az Exchange Server címter használatához (*amelyet Globális Címlistának hívnak*) nem tapasztalnak érzékelhető változást



a felhasználói felületben, és nem lesz szükségük átképzésre, amikor áttérnek az Active Directory használatára. Az Exchange Server és az Active Directory integrációjának jellemzői többek között a következők:

- ☞ Az Exchange Server és a Windows 2000 objektumok egységesített adminisztrációja lehetővé teszi, hogy az összes felhasználói adatot egyetlen helyen, egyetlen eszközzel kezeljük. A Delegation Wizard segítségével könnyedén elvégezhetjük a szokásos feladatokat.
- ☞ A Windows 2000 biztonsági csoportjai automatikusan felhasználhatók Exchange 2000 terjesztési listaként, szükségtelenül téve a párhuzamos terjesztési listák készítését minden egyes osztály vagy csoport számára.
- ☞ Az Active Directory adatbázis-leírásának bővíthetősége lehetővé teszi az elosztott információk rugalmas kezelését.
- ☞ A Lightweight Directory Access Protocol (LDAP), mint natív címtárinformáció elérési protokoll használata az e-mail könyvtár számára elérést biztosít az ügyfél szoftverek széles köréhez.

Active Directory Connector. Az Active Directory Connector (ADC), amely mind az Exchange 2000-ben, mind a Windows 2000-ben megtalálható, lehetővé teszi a címtárobjektumok hierarchiájának replikációját a Microsoft Exchange Server 5.5 címtára és a Windows 2000 Szerver Active Directory szolgáltatása között, s így alapvető eszközzé válik az Exchange 5.5-ről Exchange 2000-re áttáló cégek számára. Az Active Directory Connector a Windows NT-ről Windows 2000-re, valamint az Exchange Server 5.5-ről Exchange 2000-re történő frissítés és átállás rugalmas stratégiájának kulcsfontosságú része. Az Active Directory Connector feladatai többek között:

- ☞ Az Active Directory gyors, egy lépésben történő feltöltése az Exchange 5.5 címtárban tárolt címtárinformációkkal.
- ☞ Kétirányú másolás az Exchange és az Active Directory között. A címtárak között csak a változások kerülnek átmásolásra, így módon csökkentve a hálózati sávszélesség felhasználását.
- ☞ Tömeges importálás és exportálás az Active Directoryből, szöveges fájlformátum használatával.
- ☞ Rugalmas objektum-megfeleltetési szabályok az Exchange 5.5 címtár és az Active Directory objektumainak összekapcsolásához, ami által csökken az ütközések száma.
- ☞ Kapcsolat-egyeztetések alkalmazása, amelyek lehetővé teszik, hogy pontosítsuk a replikációs ütemtervet, a hitelesítési paramétereket, és a replikációs séma formátumát.

Az Internet Information Server (IIS) integrációja. Az Exchange 2000 szorosan integrálódik a Microsoft Internet Information Server-rel, hogy biztosítsa a nagyteljesítményű e-mail protokollok (Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), stb.) használatát, az Exchange webböngészővel történő elérését az Outlook Web Access segítségével.

Tárolási csoport (Storage Group)

Az Exchange 2000 egyik új koncepciója a tárolási csoport, amely olyan adatbázisok csoportosítása, amelyek egyetlen tranzakciós napló sorozaton osztoznak, s így módon egyetlen adminisztrációs, adatmentési és helyreállítási pontot használnak. Minden egyes tárolási csoport egy pár fűrtbe kapcsolt kiszolgáló közül átkapcsolhat a másikra, amennyiben azok nem válaszolnak többé. A tárolási csoportok létrehozása az Exchange Systems Manager MMC modul segítségével történik. Tárolási csoportonként több adatbázis is létrehozható, és több tárolási csoport definiálása is lehetséges. Ez azt jelenti, hogy számos különálló adatbázis létezhet egyetlen Exchange 2000 kiszolgálón (lásd a *Többszörös üzenet-adatbázisok című részt az alábbiakban*). A tárolási csoportok jelentik az adatmentések alapegységét, s így módon lehetővé nyílik arra, hogy úgy mentünk le egy teljes tárolási csoportot, hogy mindössze a tranzakciós napló-sorozat egyetlen példányát kelljen szalag-ára írunk. Ugyanakkor azonban minden egyes adatbázis külön visszaállítható, míg az összes többi adatbázis továbbra is on-line kapcsolatban marad, és kiszolgálja a többi felhasználót. Ez lehetővé teszi a lementett adatok gyors, a lehető legkevesebb felhasználót érintő visszaállítását.

Többszörös üzenet-adatbázisok

Az Exchange Server első megjelenése óta támogatja a tranzakció naplózást és a korábbi állapot szerinti helyreállítást, és mindig is rendelkezett beépített adatmentési és visszaállítási segédprogramokkal. Az Exchange 2000 erre az architektúrára épít és lehetővé teszi az üzenettároló particionálását külön kezelhető adatbázisokra, melyek mindegyike korlátlan méretű lehet. Ez az architektúra biztosítja mindkét megoldás legelőnyösebb vonásait: hatalmas méretű (egyenként akár 100 gigabájt méretet is meghaladó) adatbázisok kialakításának lehetőségét nagyvállalatok számára, valamint azt a lehetőséget, hogy egyetlen logikai adatbázis különálló fizikai adatbázisokra osztható. E technológia előnye abban a lehetőségben rejlik, hogy a szerverek összeszerelésével a hardverkötségek csökkenthetők, miközben a végfelhasználói rendelkezésre állás fokozódik. A további előnyök között megemlíthetjük többek között:

- ☞ A rendszer megbízhatóságát megnöveltek, hogy ha az egyik adatbázis elromlik, az ne befolyásolja a másik adatbázis felhasználóit.
- ☞ A fűrt-szervezés segítségével egyetlen logikai adatbázis számos kiszolgálóra menthető át hardver hiba esetén.
- ☞ Gyorsabb és rugalmasabb adatmentés ütemezésre van lehetőség, mivel az adatbázisok általában kisebb méretűek.
- ☞ Hardverhiba esetén a helyreállítási idő jelentősen csökken, mert az egyes fizikai adatbázisok külön-külön állíthatók helyre.

Míg a helyreállítás folyamatban van, a többi adatbázis on-line használatra továbbra is rendelkezésre áll.

Az Exchange 2000 adatbázis több gépen elosztva működik, ezért szükség esetén leállíthatjuk az adatbázis egy bizonyos példányát, és helyreállíthatjuk azt a kérdéses adatbázishoz tartozó tranzakciós naplóból még akkor is, ha közben a

többi adatbázis folyamatosan üzemel. Az adminisztráció logikai szinten történik, s így egységes képet nyújt az összes adatbázisról. Ez azt jelenti, hogy a szervizelés és helyreállítás csak azokat a felhasználókat érinti, akiknek adatai egy bizonyos gépen találhatóak. Ez a jellemző nagyobb rugalmasságot is biztosít az Exchange adatok adatmentési stratégiái esetében, lehetővé téve az adatmentési időtartamok lerövidülését. Egyszerre több adatbázis menthető le, ami csökkenti az adminisztrációs költségeket. Például egy kiszolgálón két adatbázist hozhatunk létre (például *egyét a felső vezetés menedzserei és egyét a marketing stáb számára*), és mivel az Exchange mindkét adatbázist ellenőrzi, ha az egyik adatbázis megsérül, az Exchange tovább kommunikál a másik adatbázissal. A sérült adatbázis dinamikusan leválasztható és adatmentésből visszaállítható, miközben a működő adatbázis továbbra is elérhető marad a felhasználók számára. Természetesen az Exchange 2000 egyetlen számítógépen egyetlen adatbázissal is használható, éppúgy, mint az Exchange 5.5 verzió.

Fürtözés

Az Exchange 2000-ben a korábban is ismert fürtözési támogatást továbbfejlesztették és az már tartalmazza az aktív/aktív fürtözési lehetőséget is, amely jelentős mértékben fokozza az Exchange infrastruktúra végfelhasználói rendelkezésre állását, miközben csökkenti a rendszer birtoklásának összköltségét (TCO). A Windows 2000 Advanced Server Microsoft Clustering Services szolgáltatására támaszkodva a multi-master fürtözés lehetővé teszi, hogy a fürtben használt összes kiszolgáló aktívan feldolgozza az üzenetkezelési kéréseket, amíg fel nem lép egy olyan hiba, amely kiváltja az átállásos rendszer-helyreállítást. Azáltal, hogy lehetővé teszi külön tranzakciós naplófájl-sorozatokat kialakítását a fürt egyes tagjai számára az Exchange 2000 még inkább kihasználja a Windows 2000 Clustering Services szolgáltatásait. Az a képesség, hogy az összes kiszolgáló bármilyen időpontban használható csökkenti a rendszer költségeit, miközben fokozza a megbízhatóságot, mivel nincs szükség külön erre a célra kijelölt „csak rendszer-helyreállításra szolgáló” gépekre.

Elosztott szolgáltatások (Frontend/Backend Server-ek)

Az Exchange 2000 lehetővé teszi, hogy az Exchange alrendszereket (protokoll, tárolás, könyvtár) különböző kiszolgálókon elhelyezve akár több tízmillió felhasználó-számot is lehetővé tevő méretezhetőséget érjünk el. Ez a konfigurációs rugalmasság új lehetőségeket teremt a vállalati ügyfelek számára, hogy rendszerarchitektúrájukat a vállalat belső működéséhez illetve a növekedés igényeinek kielégítéséhez igazítsák. Az Exchange 2000 egy sor előtér (front-end) és háttér (back-end) kiszolgálóként konfigurálható, ami lehetővé teszi a felhasználók számára, hogy virtuális Internet Protokollal (IP) címezhető előtér szerverekhez csatlakozzanak, míg az üzeneteket és az együttműködési adatokat külön háttérkiszolgálókon tárolják. Például egy sor kiszolgálót, amelyek a bejövő HTTP, SMTP, POP és Internet Message Access Protocol (IMAP) kéréseket kezelik, elhelyezhetünk az adatok tárolástól teljes egészében elkülönítve.

nített, RAID vezérlő nélküli gépeken. Ez csökkenti a rendszer költségét, és megakadályozza, hogy az SMTP hibák, és az szolgáltatás ellen irányuló támadások befolyásolják az üzenettárolót vagy címátárát.

Nagyteljesítményű internetes kapcsolat

Hibatűrő SMTP üzenetküldési útvonali-irányítás. Az Exchange 2000 natív támogatást tartalmaz az SMTP, POP, LDAP, IMAP, Hypertext Transfer Protocol (HTTP), Network News Transfer Protocol (NNTP), Secure/ Multipurpose Internet Mail Extensions (S/MIME), és X.509 version 3 protokollokhoz. Ezek a protokollok lehetővé teszik, hogy egy vállalat Internet gateway-eként működjön. A robosztus, nagy teljesítményű e-mail útvonali-irányítás az Exchange Server egyik megkülönböztető jellemzője. Az Exchange 2000 egyik legnagyobb újdonsága, hogy az eddig használt X.400-zal szemben az SMTP lett az alapértelmezett átviteli protokoll a szerverek közötti üzenetforgalomban, egyetlen telephelyen belül éppúgy, mint telephelyek között. Az SMTP használata fokozott teljesítményt eredményez és új lehetőségeket teremt az Internettel való integrációra a vállalatok közötti, illetve a vállalatok és a vásárlók közötti kommunikációhoz. Az Exchange 2000 üzenetküldési útvonali-irányítási algoritmusait jelentősen továbbfejlesztették, biztosítva a hibátűrő üzenetkézbesítést és a kiküszöbölve az üzenetek „visszadobását” még akkor is, ha több kiszolgáló vagy hálózati összeköttetés esik ki. Ez a szolgáltatás fokozza az üzenetátviteli teljesítményt, különösen az Internet felé irányuló forgalom esetében. Az X.400-as szabvány folytatódó támogatásával kombinálva (ehhez az Exchange 2000 Enterprise Server szükséges) az SMTP útvonali-irányítás az ügyfelek számára jelentős rugalmasságot biztosít, ha megbízható, nagy teljesítményű üzenetkezelő gerinchálózat kialakítását tervezik az Exchange 2000-rel.

Natív internetes üzenettartalom. Az Exchange 2000 jelentősen növeli az Internetes e-mail teljesítményt, mivel az e-mail kliensek számára lehetővé teszi, hogy a MIME tartalmat közvetlenül az Exchange 2000 adatbázisban tárolják, és onnan hívják le bármilyen tartalom-konverzió nélkül.

Adminisztráció

Egyszemélyes felügyelet a Microsoft Management Console-lal. Az Exchange 2000 egyszemélyes, grafikus adminisztrációt és felügyeletet tesz lehetővé a megszokott Windows 2000 Microsoft Management Console-lal (MMC) való integrációja révén. A Microsoft szoftverek, beleértve magát a Windows 2000-et, az SQL Server 7.0 verziót, az Exchange 2000-et és a System Management Server 2.0 verzióját, valamint sok, más cégek által gyártott alkalmazást, az MMC-t használják elsődleges adminisztrációs konzolként. Az Exchange 2000 rendszerkezelése három MMC snap-in használatával valósul meg, ezek az Exchange System Manager, az Active Directory Users and Computers, és az Internet Services Manager. Az MMC snap-in modulokat egyedi igények szerint módosította adminisztratív konzolok hozhatók létre, amelyek a nagyvállalatoknál jel-



lemzően elforduló különböző adminisztratív szerepkörökre specializáltak. Egyedi adminisztrációs konzol hozható létre a „szerzői üzemmód” segítségével, valamint különböző kiterjesztések hozzáadásával és kikapcsolásával. Például egy „csak e-mail” jellegű rendszerkezelő konzol hozható létre, hogy adminisztrációs elérését biztosítsan az Exchange System Manager snap-in minden szolgáltatásához, de ugyanakkor kizárja az összes közös mappákkal kapcsolatos adminisztrációs feladatot.

Bővíthető rendszerfelügyeleti felület. Az Exchange 2000 az egyedi felügyeleti fejlesztéséhez egy Collaboration Data Objects (CDO) for Exchange Management elnevezésű átfogó programozói felületet tartalmaz, amely a CDO programozói felület kiterjesztése. Ezekkel az eszközökkel más cégek által és egyedileg fejlesztett alkalmazások postaláda és közös mappakezelési feladatok széles skáláját végezhetik el, és egyedi rendszerkezelési felhasználásokhoz kibővíthetik az MMC snap-in eszközöket. Ez lehetővé teszi, hogy feladat-specifikus felületeket hozzunk létre, a vagy automatikus batch szkripteket írjunk az ismétlődő feladatokhoz. Például ilyen interfészek alkalmazásával programozott adminisztrációs ügynök kvóta-határokat állíthat fel a felhasználók számára, vagy ellenőrizheti a Microsoft Web Storage System kulcsfontosságú nyilvános mappáinak replikációs ütemezését.

Házirendalapú adminisztráció. Az Exchange 2000 hatékony és rugalmas házirendalapú adminisztrációs modell kínál, amely lehetővé teszi, hogy egyetlen művelettel változtassuk meg egy objektumsorozat (például a felhasználói postaládák, szervek és közös mappák) adminisztrációs opcióit. Egy házirend a meghatározás szerint konfigurációs beállítások olyan együttese, amely egy vagy több Exchange konfigurációs objektumra alkalmazható. Rendszergazdák objektumok százaira alkalmazható házirendeket definiálhatnak, és megvalósításuk után egyetlen művelettel változtathatják meg a konfigurációs opciókat vagy érvényesíthetnek bizonyos beállításokat az említett összes objektumra vonatkozóan. Az rendszergazdák ezen felül hatékony LDAP lekérdező interfész segítségével egyetlen művelettel definiálhatnak üzenetfogadási házirendet felhasználók, csoportok és kapcsolatok százeire számára a Windows 2000 Active Directory szolgáltatásában.

Rendszerfigyelés a Windows Management Instrumentation segítségével Az Exchange 2000 teljesben új rendszerfigyelési infrastruktúra és felhasználói interfész segítségével biztosítja az Exchange rendszerszintű felügyeletét, egyszerű, szűrhető, áttekinthető képet nyújtva az összes kiszolgálóról és kapcsolódási pontról a vállalatban belül. A kiszolgálóknál figyeli a lemezterület felhasználást, a működési állapotot és a sorbanállási trendeket. A felügyelet alapját a Windows Management Instrumentation (WMI) architektúra képezi, az eseménynaplók, a PerfMon adatok, a lemezadatok és a működési állapot eléréséhez pedig szabványos Windows 2000 WMI szolgáltatásokat használ. Ezen felül az Exchange-re jellemző szolgáltatók a kap-

csolódási állapotot közvetlenül az útvonalirányítási kapcsolatot állapottáblái és a QAPI-tól kapott várakozási soradatok alapján felügyelik. Az összes felügyeleti adat rendelkezésre áll egy MMC modulon keresztül, és az összes adat elérhető más cégek alkalmazásaival is.

Dokumentumkezeléssel és ismeretkezeléssel kapcsolatos szolgáltatások

Hozzáférés a Microsoft Web Storage System-hez. A Microsoft Web Storage System egy speciálisan az üzenetküldésre, együttműködésre dokumentum-tárolásra, és webes alkalmazásokra optimalizált adatbázis. A Microsoft Web Storage System-hez ügyfélszoftverek széles választékával lehet hozzáférni, többek között az alábbiakkal:

- ☞ A Microsoft Outlook Express és minden olyan e-mail vagy hírcsoport klienssel, amely a következőket támogatja: SMTP, POP, IMAP vagy NNTP
- ☞ FrontPage 2000 (lásd [lejjebb](#))
- ☞ Windows Explorer
- ☞ Web Folders (Web könyvtárak) (az Internet Explorer 5.0, Office 2000, és Windows 2000 egyik szolgáltatásaként szállítva)
- ☞ Minden webböngésző
- ☞ Minden 32 bites Windows alkalmazás
- ☞ MS-DOS rendszerablak

Installálaskor a Web Storage System úgy jelenik meg, mint a kiszolgálón található lokális meghajtó. Emiatt a használata egyrészt teljesen azonos lehet, mint meglévő Windows file rendszeré. Emellett (innen a „web” jelölés) közvetlenül, minden kiegészítés nélkül a teljes tartalom elérhető webfelületen is, a Microsoft IIS-en keresztül.

Továbbfejlesztett integráció az Office 2000-rel. Az Exchange 2000 jelentős mértékben növeli az integráció szintjét az Exchange és az Office 2000 között. A dokumentumokat az Office 2000 alkalmazásokból közvetlenül el lehet menteni a Microsoft Web Storage System-be a **File/Save As** párbeszédablak segítségével, könnyen vissza lehet keresni, és az Exchange 2000-ból közvetlenül el lehet hívni a **File/Open** párbeszédablak segítségével. Az Office felhasználók immár minden vonatkozó információt és e-mail üzenetet tárolni tudnak az Exchange kiszolgálón.

Telepíthető fájlrendszer. A Microsoft Web Storage System olyan telepíthető fájlrendszert biztosít, amely lehetővé teszi a zökkenőmentes integrációt az Exchange-alapú adatok és a Windows fájlrendszer között. Az Exchange Server-t futtató számítógépen lévő M: meghajtó közvetlen Win32 alkalmazásprogramozói felületen keresztül hozzáférést biztosít a Microsoft Web Storage System-ban tárolt valamennyi adathoz. Ezeket a tárolókat (stores) azután más gazdaszámítógépek közösen használhatják pont úgy, mint a normál fájlrendszer szerint létrehozott könyvtárakat. Minden olyan Win32 alkalmazás, amely a fájlokat közvetlenül lemeze

menti, előnyösen használhatja az említett hozzáférési mechanizmust abból a célból, hogy fájlrendszer lokális vagy távoli helyekről közvetlenül a Microsoft Web Storage-ban tárolja el.

Dokumentum-jellemzők (*Properties*)

A Microsoft Web Storage System lehetővé teszi a jellemzők (*properties* – *szerző*, *cím*, *lektor*, *munkafolyamat státusza*) eltárolását minden egyes elemhez (*item*) az adatbázisban, ezzel hatékony eszközzel szolgál a tájékozódást könnyítő nézetek (*view*) definiálására, illetve a hatékony visszakeresést szolgáló indexek építésére.

Minden egyes elemmel együtt bármennyi jellemző tárolható, és a jellemzők száma az egyes elemeknél más és más lehet. A relációs adatbázistól eltérően nem kell külön parancsot futtatni a jellemzők hozzáadása céljából, mert a jellemzőket egyszerűen hozzáadjuk a szóban forgó elemhez, és azzal együtt automatikusan eltárolja. A Microsoft Web Storage System a jellemzők sok fajtáját támogatja, többek között a következőket: sztring, dátum, Boolean, egész szám és pénzszám.

Ez a tulajdonsága akkor válik különösen fontosá, amikor az Office dokumentumokat Exchange 2000-ben tárolják, lehetővé téve az Office felhasználók számára, hogy az Outlook képességeit arra használják, hogy a dokumentumokat ugyanazokkal az eszközökkel nézzék meg, szortírozzák, keressék és kezeljék, mint amit jelenleg használnak e-mail-jelkeikhez, naptárukhoz és kapcsolataikhoz (*calendar*, *contacts*). Az Office dokumentumok és más dokumentumfajták jellemzői pl. MIME-kódolású üzenetek, automatikusan rendelkezésre állnak a Microsoft Web Storage System számára keresés, megtekintés és indexelés céljából. A dokumentum jellemzői a munkafolyamat-alkalmazásoknál is fontosak, mert módot adnak valamely munkafolyamat-elem állapotának vagy megfelelő helyre történő irányításának nyomon követésére és megjelenítésére.

Beépített tartalomindex és -kereső

A Microsoft Web Storage System beépített keresőt tartalmaz teljes szövegek gyors, pontos keresésére, mellyel a felhasználók a tartalomban gyorsan és könnyen kereshetnek. A tartalomindex lehetővé teszi a felhasználóknak, hogy valamely dokumentum tartalmában keressenek, mondjuk egy Microsoft Word dokumentumban, vagy hogy csak a dokumentum jellemzőit keressék meg, pl. a címet vagy a szerzőt. Az Outlook 2000 felhasználói a Microsoft Web Storage System-ben tárolt dokumentumokat megkereshetik, pont úgy, mintha e-mail üzeneteket keresték, ezzel a felhasználó produktivitása növekszik. A felhasználó ugyanazt az Outlook kereső felület használhatja amelyet már megszokott, de a lekérdezés jelentős mértékben felgyorsul, és a felhasználó a megadott kritériumoknak megfelelő dokumentumot megtalálhatja akár e-mail üzenethez van csatolva, akár önálló dokumentumként van eltárolva, tehát nemcsak az üzenetek között kereshet. A Microsoft Web Storage Systemben minden tartalom indexelve van, tehát az üzenetek, önálló dokumentumok, kapcsolatok, feladatok, naptárbejegyzések és együttműködésre vonatkozó adatok is. Az indexelés a Microsoft Web Storage

System saját tartalomindexkeresője végzi, ugyanannak a technológiának az alkalmazásával, mint amelyet az Internet Information Server (IIS) és az SQL Server 7.0 használ.

Webalapú elosztott dokumentumkezelési támogatás

WebDAV olyan új HTTP-alapú protokoll, amely lehetővé teszi, hogy a weben zökkenőmentesen közösen használjuk a dokumentumokat és azokkal dolgozzanak, és amely – függetlenül az egyes felhasználók szerzői eszközeitől – megengedi a felhasználóknak, hogy távolról a fájlokkal alapműveleteket (*másolás*, *átmozgatás*, *törlés*) végezzenek a weben keresztül. Az Exchange 2000 WebDAV támogatása lehetővé teszi, hogy az Office 2000 dokumentumokat közvetlenül az Exchange-be lehet elmenteni, megnyitni, módosítani a WebDAV az XML-t használja az adatalemek közvetítési formájaként.

Alkalmazásfejlesztés

Webalkalmazás platform. A Microsoft Web Storage System segítségével webhelyek Exchange 2000 gazdaszámítógépként működhet azon és munkafolyamat-alkalmazások számára, amelyek vagy az Outlook-ot vagy egy böngészőt használnak adatokhoz történő hozzáférésre, és amelyeket olyan szokásos alkalmazástervező eszközökkel lehet létrehozni mint amilyen a FrontPage 2000, Visual Basic, Visual Basic for Applications, és a Microsoft Visual Studio fejlesztőrendszer – vagy más egyéb standard webfejlesztő eszköz. A HTML lefordításával és ASP kódok közvetlenül az Exchange 2000 alapfolyamatában történő végrehajtásával, valamint az XML és más Internet-szabványok alkalmazásával, az Exchange 2000 jelentősen megnöveli azoknak a webes alkalmazásoknak a funkcionalitását és teljesítményét, amelyek hozzáférnek az Exchange 2000-ben tárolt közös adatokhoz és levelezéshez. Ez lehetővé teszi a tervezők számára, hogy szorosabban integrálják a Microsoft Web Storage System tartalmát – pl. tartalom, dokumentumok, megbeszélések, feladatok és munkafolyamat-feldolgozás – a saját webalkalmazásaikba.

CDO

A CDO 3.0 verziója Component Object Model (*COM*) alap objektumkészlet, amelyet a munkafolyamat-kezeléshez és más, a csoportmunkát támogató alkalmazásokhoz meg lehet határozni az üzleti logikát, webalapú alkalmazásokat lehet kifejlesztetni, és hozzá lehet férni az Active Directory szolgáltatáshoz. A CDO-t fel lehet használni olyan alkalmazások létrehozásához, amelyek az Exchange 2000 beépített lehetőségeit használják fel mint pl. levelezés, naptár, kapcsolat-kezelés, rendszerkezelés, Active Directory hozzáférés, valamint munkafolyamat kezelés (*workflow*). A CDO 3.0 az OLE DB-re épül, és nemcsak az Exchange Server adataihoz, hanem az Internet protokoll szolgáltatásokhoz is (pl. *LDAP lekérdezések és Multipurpose Internet Mail Extensions (MIME) üzenetek*) hanem hozzáférést biztosít. A CDO 3.0 alkalmazásával a további képességeket adhatunk hozzá a kiszolgálóhoz és az Outlook ügyfélhez, hogy az megfelelően a speciális elvárásoknak és üzleti igényeknek (pl. *üzenetek archiválása, vállalati szabályzatok betartatása, értesítések továbbítása személy-*



hívókra, és címlisták forgalmának kezelése).

HTTP és XML támogatása

Az Exchange 2000 a HTTP protokoll segítségével biztosítja a Microsoft Web Storage System-hez való hozzáférést. Az XML lehetővé teszi egyedi adatformátumok létrehozását speciális alkalmazások számára, és ideális formátuma a strukturált adatok kiszolgálók közötti átadásának. Az XML-t egyre szélesebb körben használják eszközként a strukturált adatok többretegű alkalmazásokban a közép-ső rétegtől az ügyfélig történő továbbításához. Az alkalmazásfejlesztők az Exchange 2000 XML támogatását felhasználva speciális új lehetőségeket építhetnek alkalmazásaikba. Ezen felül az Exchange 2000 az XML-t használja fel arra, hogy növelje az Outlook Web Access kliens teljesítményét és a felhasználói felület gazdagságát.

Webürlapok

A Microsoft Web Storage System ürlapjai böngészőalapú ürlapok, amelyeket a Microsoft Web Storage System tárol és HTTP-n keresztül az Exchange 2000-rel közvetlenül a böngészőhöz közvetíti. Pl. egy Microsoft Web Storage System könyvtár tartalmazhat költségjelentési ürlapot vagy megrendelés nyomtatványt, amely egyszerűen az URL-be begépelve vagy kedvenc (*favorite*) kiválasztásával lehet webböngészővel hozzáférni. A webürlapokat FrontPage 2000-rel lehet létrehozni, és arra használhatók, hogy a Microsoft Web Storage System könyvtár alatt elemeket lehessen létrehozni és módosítani egy olyan böngésző segítségével, amely tudja kezelni a HTML 3.2. verzióját. A Microsoft Web Storage System ürlapjai tovább gazdagítják a felhasználói felületet, többek között az egérral történő műveleteket (*drag and drop*). Ez a technológia olyan hatékony integrált eszközt ad az alkalmazástervezők kezébe, amellyel a webalkalmazásokat össze tudják kapcsolni az üzleti logikával, eseményekkel, továbbá a Microsoft Web Storage System meta-adat jellemzőivel.

Integrálás a Microsoft FrontPage 2000-rel

A FrontPage 2000 a Microsoft Web Storage System-en mint gazdagépen található webalkalmazások szerkesztésére és kezelésére szolgál. Például a FrontPage 2000-ben egyedi igényre szabott ürlapokat lehet létrehozni és a Microsoft Web Storage System-ben tárolni. A FrontPage-ben lévő eredeti párbeszédablak felhasználásával a fejlesztők közvetlenül a Microsoft Web Storage System-ben tudnak egy weboldalt kinyitni és azt megszerkeszteni az általuk jól ismert FrontPage eszközökkel, majd az alkalmazást közvetlenül a Microsoft Web Storage System-ből egy böngészőben futtatni.

OLE DB és ADO integrálása

A Microsoft Web Storage System az elemeknek félig-strukturált adatbázisa, amelyben a korábban említett fájlrendszerszerű és webalapú hozzáférés mellett az OLE DB és ADO interfészek által nyújtott, jól ismert SQL szintaxis felhasználásával is lekérdezhető, módosítható. Az OLE DB és az ADO 2.5 olyan programozó interfész, amely közös

eszközként szolgál az összes szervertől adatbázishoz való hozzáférésnél, függetlenül attól, hogy melyik kiszolgálón és milyen típusú adatbázisban tárolódnak az adatok. Az Exchange 2000 magában foglal két OLE DB szolgáltatót, egy távszolgáltatót a kliens-alkalmazások – mint Outlook 2000 vagy Office 2000 – szervertől történő hozzáférésehez, és egy helyi szolgáltatót amelyet magában az Exchange-ben valósítottak meg nagytejlesztményű COM-
hozzáférésre olyan alkalmazásokhoz mint pl. víruskereső programok és munkafolyamat kezelők.

Lehetővé teszi a kombinált SQL és Exchange adatkezelést is. Az alkalmazás-tervezők az ADO-t arra is felhasználhatják, hogy Exchange Server adatokban navigáljanak, azokat lekérdezzék, rendezzék és szűrjék. Ez lehetővé teszi az SQL alkalmazások fejlesztésében jártos fejlesztők számára, hogy ugyanazt a szakutadást és eszközt alkalmazva könnyen írhasanak olyan alkalmazásokat, amelyek a Microsoft Web Storage System-ben tárolt adatokat használják. Szintén könnyen létrehozhatók olyan alkalmazások, amelyek hozzáférnek a Microsoft Web Storage System és az SQL Server adataihoz.

Többcsatornós nyilvános mappafák

Az Exchange 2000 támogatja a többcsatornós nyilvános mappafákat, ezzel nagyobb rugalmasságot adva a rendszergazdának a vállalaton belül működő csoportmunkaalkalmazások bevezetésékor. A rendszergazdák most már el tudják különíteni a közösen használt adatbázisokat funkcionális, üzleti vagy földrajzi szempontok alapján. Pl. egy nyilvános könyvtárfa tartalmazhatja a HR osztály összes csoportmunka alkalmazását, egy másik a könyvelését, és egy harmadik a kutatás és fejlesztését.

Üzleti folyamatok automatizálása

Folyamatirányító alkalmazások kialakítása üzleti logikával. Az Exchange 2000 immár magas szinten képes kiszolgálni a munkafolyamatokat (*workflow*) nyomonkövet eljárásokat (*dokumentumok jóváhagyása, költségjelentés, beszerzés igénylés stb*). A CDO 3.0 és a Microsoft Web Storage System eseménymodell együttesen professzionális workflow motort alkot, amely lehetővé teszi a workflow állapotváltozások vezérlését, a társuló üzleti logikából következő lépések és az események egymáshoz rendelését. Az üzleti logikaelemeket Visual Basic-ek lehet kialakítani, a megsokotott eszközök használatával.

A workflow megoldások tervezése és kialakítása az Exchange 2000-ben nagyban leegyszerűsödik a Workflow Designer for Exchange 2000 használatával, mely a Microsoft Office 2000 Developer termék része.

Workflow Designer for Exchange 2000

A Workflow Designer for Exchange használatával a fejlesztők grafikus segédeszközök felhasználásával határozhatják meg az egyes folyamatlépések és az információáramlás szabályainak sablonját egy adott workflow megoldáson belül. A létrehozott workflow folyamat a Collaboration Data Object (*CDO*) objektumokat felhasználva „fordítja le”

tényleges működő alkalmazással a grafikusan meghatározott munkafolyamat modellt. Ezután természetesen a fejlesztők a workflow folyamatok egyes lépéseit testreszabhatják scriptek hozzáadásával. Ezek mellett a Workflow Designer lehetővé teszi az egyes lépésekben a felhasználók számára engedélyezett tevékenységek definícióját.

Kommunikáció bármikor és bárhol

A földrajzi, szervezeti és technológiai korlátokon átlépő információelérés kritikus fontosságú a szellemi tevékenységet végzők sikere szempontjából, akiknek képesnek kell lenniük arra, hogy bárholnán gyorsan és hatékonyan kommunikáljanak. Az olyan szolgáltatásokkal, mint az Instant Messaging és a valós idejű adat- és videókonferencia, az Exchange 2000 bármikor és bárholnán biztosítja azoknak az embereknek és információknak az elérését, melyekre szüksége van.

Kliens hozzáférés

Bővített Outlook Web Access. Az Exchange Web kliensét, melyet Outlook Web Access-nek neveznek, az Exchange 2000-ben jelentősen bővítették. Az Outlook Web Access, melyet távoli, otthoni, utazó és alkalmi felhasználók használnak, Web böngésző elérést biztosít az e-mail-hez, a határidőnaplóhoz (*beleértve a csoportos határidőnaplókat*), a címlistához (*kapcsolatokhoz*), és a Microsoft Web Storage System mappákban tárolt együttműködő (*közös*) információkhoz. Amikor az Internet Explorer 5.0-val használják, a Web kliens natív módon a DHTML-t használja, mely a szerverről érkezett XML adatokat dolgozza fel, demonstrálva a Microsoft Web Storage System gazdag lehetőségeit. Az Outlook Web Access bármilyen böngészővel használható, mely támogatja a frame-eket és scripteket. Az Exchange 2000-t különösen az alábbi területeken fejlesztették tovább:

- Inkább az Outlook-hoz hasonló megjelenés, üző menük, eszköztárak, valós hierarchiavezérlés és rich-text szerkesztés, amikor a Microsoft Internet Explorer 5.0-val használják.
- Az alapvető HTML megjelenítések mostantól nem script-alapúak, mely nagyságrendi teljesítményjavulást eredményez.
- Beágyazott üzenetek küldése és fogadása az e-mail üzenet belsejében.
- Frame-eken belüli navigálás a régi ügyfelek számára.
- Beágyazott OLE-objektumok megjelenítése.
- Automatikus névmeghatározás az Outlook kapcsolatok valamint az Exchange címek között.
- A felhasználói felületen keresztül további opciók érhetők el.
- Megnövelt funkciókészletű naptárszolgáltatások, ezen belül delegált elérés.

URL címzés

Az Exchange 2000 megkönnyíti a Microsoft Web Storage System információk letöltését a webböngésző használatával, mivel az Exchange 2000-ben tárolt minden adat (*üzenetek, dokumentumok és workflow elemek*) elérhető a webböngészőből egy felhasználóbarát URL-en keresztül.

Például a felhasználók könnyedén elérhetik postaládájukat webböngészőjükkel, egy olyan URL használatával, mint pl. <http://servername/exchange/username> és épp ilyen egyszerűen elérhetik az egyes üzeneteket, dokumentumokat és együttműködő elemeket.

Egységesített üzenetküldő platform

Sok egységesített üzenetküldő rendszerrel (*Unified Messaging*) rendelkező vezető vállalat már az Exchange Serverre építi ezen szolgáltatásait. Az Exchange-et platformként használják arra, hogy a minden „knowledge worker” részére telefon vagy egyéb eszköz segítségével bárhol és bármikor biztosítsák egy olyan univerzális „postaláda” elérését, mely e-maileket, hangüzeneteket, faxokat és személyhívó üzeneteket egységes rendszerben tárol. Több mint 50 vállalat, köztük olyan ágazatvezető vállalatok, mint a Lucent Technologies Inc., a Nortel Networks Corp. és az Active Voice Corp. az Exchange architektúrára építik egységesített vállalati üzenetküldő rendszerüket. Az Exchange 2000 az egységesített üzenetküldő rendszer platformszolgáltatásait az alábbi területeken bővítette:

- A VPIM szabvány támogatása, mely együttműködést tesz lehetővé különböző üzenetrögzítő rendszerek között
- A Windows 2000 Active Directory-ban lévő beépített mezők elérése a felhasználók hangüzenetekkel kapcsolatos adatai számára.
- Telepíthető Outlook űrlapok hangüzenetek rögzítésére és lejátszására.
- Webhozzáférés minden egységesített üzenetküldő adathoz az Outlook Web Access használatával

Instant üzenetküldés

Az instant üzenetküldés arra a képességre utal, hogy egy számítógépes hálózatban lévő másik felhasználónak azonnali szöveges üzenetet lehet küldeni. Az e-mail üzenetekkel ellentétben az instant üzeneteket azonnal postázzák a másik felhasználó képernyőjére, megteremtve ezzel a közös munka új formáinak alapjait. Az instant üzenetküldés széleskörű kommunikációs jelenségévé vált az Internet felhasználók körében és várhatóan jelentős szerepet fog játszani üzleti eszközökben mindenfajta méretű szervezet számára. Az Exchange 2000 biztonságos, szabványos, egységesített architektúrára épülő instant üzenetküldő szolgáltatást tartalmaz, mely ideális lehet a vállalatlan belüli, vagy az Interneten keresztül vállalatok közötti kommunikációban való bevezetésre. Az instant üzenetküldéshez tartozó kliens szoftver az Exchange 2000-ben az MSN Messenger Service instant üzenetküldő kliens.

Jelenléti információ

A jelenléti információ, mely szorosan kapcsolódik az instant üzenetküldéshez, lehetővé teszi az egyik számítógépes felhasználó számára, hogy megnézze, hogy egy másik felhasználó az adott pillanatban be van-e jelentkezve valamilyen hálózatra, a vállalati LAN-ra vagy az Internet-re. A jelenléti információt a felhasználó beállíthatja valamilyen konkrét státusz jelzésére (például



„telefonálok” vagy „kimentem ebédelni”), vagy ez az állapot bizonyos idejű inaktivitás után automatikusan beállhat. Az Exchange 2000 teljes körű támogatást nyújt a jelenléti információkhoz.

Chat szolgáltatások

A Chat az egyik legrégebbi és legszélesebb körben használt alkalmazás az Interneten. A Chat a két vagy több ember közötti valósidejű kommunikáció alacsony sávszélességű, könnyed módszerét biztosítja. A Chat általánosan elterjedt módszer olyan online felhasználói közösségek kiépítésére, melyek tagjait hasonló témák és kérdések megvitatása érdekli. Ez két felhasználó közötti beszélgetéseket, csoportos megbeszéléseket és egy szakértővel, hírességgel vagy kulcsfigurával (úgy mint pl. vállalati vezető) folytatott előadótérmi stílusú, közvetített beszélgetéseket foglal magában. Az Exchange 2000 tartalmaz egy chat szolgáltatást, mely az Internet Relay Chat-en (IRC) alapul, mely a szöveges chat rendszerek szabványos protokollja, és számos bővített szolgáltatással rendelkezik, melyek a következők:

- ☞ Bővített skálázhatóság 20.000 felhasználóig egyetlen szerveren.
- ☞ Előadótérmi üzemmód, különleges eseményekhez kapcsolódó chat támogatására, úgy mint pl. online chat egy vezetővel, hírességgel vagy szakértővel. Ez lehetővé teszi, hogy csak a műsorvezető és az előadó küldjön üzeneteket a chat-ben résztvevő összes személy számára (a hallgatóság). A hallgatóság közvetlenül a műsorvezetőnek küldhet kérdéseket, anélkül, hogy megszakítaná az elsődleges megbeszélést vagy előadást.
- ☞ Windows 2000 Active Directory integráció, annak érdekében, hogy egyetlen címteret biztosítson az állandó konfigurációs adatok tárolására, beleértve a csatornákat, a leltárásokat, az osztályokat és a hálózati konfigurációt.
- ☞ Windows 2000 Access Control listák használata a biztonság érdekében.
- ☞ Virtuális chat hálózatok egyetlen szervercsoporton.

Adatkonferencia

Az Exchange 2000 adatkonferencia lehetővé teszi az adatok és információk dinamikus, igény szerinti megosztását bármilyen T.120-kompatibilis kliens használatával (pl.: Microsoft NetMeeting). Ez képessé teszi a felhasználókat arra, hogy lássák egymás alkalmazásait, beszélgessenek (chat-eljenek) és multimédiás adatokat osszanak meg egymással. Az Exchange 2000 az adatkonferencia három kulcsterületét biztosítja:

- ☞ **Alkalmazások megosztása.** Egy program megosztása, úgy mint a Microsoft Word, mely a résztvevők számítógépén fut és képessé teszi a konferencia résztvevőit arra, hogy valós időben együtt hozzanak létre dokumentumokat.
- ☞ **Megbeszélés.** A konferencia szöveges vagy hangos komponense, mely lehetővé teszi a felhasználók közötti interakciót.
- ☞ **Fájlvitel.** Egy fájl küldése a másik felhasználónak, a

konferencia hátterében.

- ☞ **Whiteboard lehetőségek.** Több lapos, több felhasználós rajzoló alkalmazás, mely lehetővé teszi a felhasználók számára diagramok és szervezeti ábrák felvázolását és grafikus információ megjelenítését, melyet a résztvevők a konferencia ideje alatt meglehetnek.

Audio- és videokonferencia

Az Exchange 2000 konferencia szerver teljeskörű többoldalú audio és videó konferenciát kínál, a vállalati felhasználásra ideálisan alkalmas technológia segítségével. Az Exchange 2000 a Telephony API (TAPI) 3.0-t használja a Windows 2000 egyedülálló együttműködési szolgáltatásainak elérésére, úgy mint a Quality of Service vagy az IP-alapú multicaszt technológia, az audio és videó telekonferenciaszolgáltatások biztosításához. Az IP-alapú multicaszt technológia különösen alkalmas vállalati konferenciákhoz, mert lényegesen több szimulán felhasználó részvételét teszi lehetővé a videokonferenciákon, mint a ma használatos többrésztvevős videómegoldások.

Integráció az Outlook 2000-rel

Az Outlook 2000 beépített támogatással rendelkezik az online megbeszélések ütemezéséhez, mely kifinomult felhasználói felületet biztosít a valós időben együttműködő felhasználók számára. Az Outlook szabványos megbeszélés meghívás űrlapját használva a felhasználók be tudnak állítani egy adat- vagy videokonferenciát és az Exchange 2000 címterét használhatják a résztvevők meghívásához. A megbeszélések lehetnek magánjellegűek vagy nyilvánosak, a megbeszélés szervezője döntheti el, hogy a megbeszélés a meghívottakra korlátozódik vagy nyilvánosan hozzáférhető. Amikor megkezdődik a megbeszélés, a résztvevők egyetlen egérkattintással csatlakozhatnak. A felhasználók bönsgeszőjük segítségével is ütemezhetik és elérhetik a megbeszéléseket.

A Microsoft Exchange 2000 Server nem egyetlen termék, hanem egy termékcsalád, amely három verziót jelent:

- ☞ **Exchange 2000 Server.** Ez a kiadás azokat a szolgáltatásokat és azt a funkcionalitást biztosítja, amelyeket egy kis- vagy közepes méretű cég vagy fiókroda igényel üzenetküldési és csoportos munkavégzési feladatok ellátásához.
- ☞ **Exchange 2000 Enterprise Server.** Ez a kiadás az az elsődleges szerver, amellyel megbízható, méretezhető és könnyen kezelhető üzenetkezelő és együttműködési infrastruktúra hozható létre.
- ☞ **Exchange 2000 Conferencing Server.** Ez a kiadás lehetővé teszi a tudásanyaggal dolgozó alkalmazottakat, hogy az Intraneten és az Interneten adat-, hang- és videokonferenciákat szervezzenek és vezessenek, illetve hogy ilyen konferenciákon vegyenek részt.

Füzessy Tamás, ftamas@montana.hu



TECH.NET

Exchange 2000 Conferencing Server

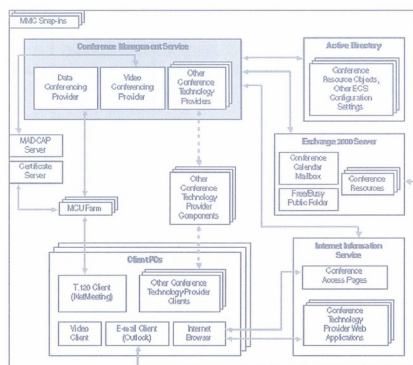


A Microsoft Exchange 2000 Conferencing Server nagyon jól skálázható megoldást kínál valósidejű konferenciák, megbeszélések számítógépes alapokra helyezésére. Előre beütemezhető megbeszélések szervezhető élő hang- és videókapcsolattal, amelyet számos egyéb eszköz egészít ki: közös információk tábla (*whiteboard*), szöveges beszélgetési lehetőség (*chat*), alkalmazásmegosztás (például *közösen kiemelezhető egy-egy dokumentum*) és fájlátvitel. Az így szervezett konferenciák lehetnek teljesen nyilvánosak vagy zártkörűek mindössze néhány résztvevővel. Az Exchange Conferencing Server által központosított foglalási és szervezési rendszer lehetővé teszi az ilyen jellegű események tervezését, ütemezését a Microsoft Outlook naptárába, ahonnan a felhasználók akár közvetlenül is csatlakozhatnak ezekhez. Azok a felhasználók, akik nem használnak Microsoft Outlook-ot, egy webböngésző segítségével vehetnek részt a konferenciákon.

Az Exchange Conferencing Server új lehetőségeket nyit meg a valósidejű hang-, kép- és adatátviteli terén. Megkönnyíti a felhasználók számára a csoportmunkát, új platformot jelenthet a különböző e-learning megoldások számára.

Az Exchange Conferencing Server komponensei

Az Exchange Conferencing Server felépítését a következő ábra mutatja be:



Az Exchange Conferencing Server komponensei és a hozzá kapcsolódó szolgáltatások

Az összetevők két fő csoportba sorolhatók: **belső és külső komponensek.**

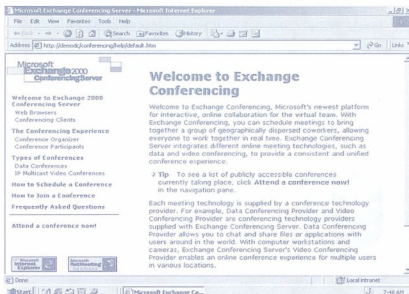
Belső komponensek

Az Exchange Conferencing Server belső alkotóelemei a Conference Management Service, a Data Conferencing Provider, és a Video Conferencing Provider képezik. Ez utóbbi két elemet technológiaszolgáltatóknak (*conference technology provider*) is nevezik.

Conference Management Service

Conference Management Service felelős az erőforrások és a megfelelő technológiaszolgáltatók irányításáért. Ez a szolgáltatás teszi lehetővé egy IIS segítségével publikált weboldalon keresztül a konferenciához való hozzáférést. Szintén a Conference Management Service feladata a konferencia erőforrások foglaltságának, illetve rendelkezésre állásának nyilvántartása az Exchange 2000 Serverben. Az Outlook ezeket az adatokat felhasználva jelzi a felhasználók számára az erőforrások elérhetőségét a különböző időpontokban. Ha egy konferencia kerül beütemezésre, akkor a Conference Management Service létrehoz számára egy új, egyedi URL-t. A résztvevők ezen az URL-n keresztül férhetnek hozzá a konferenciához. A Conference Management Service a konferenciaerőforrások alapján weboldalon keresztül jeleníti meg az éppen elérhető nyilvános konferenciákat, megbeszéléseket. Ezeket bármely felhasználó részt vehet.

A Conference Management Service a technológiaszolgáltatókkal közösen felügyeli a konferenciák teljes életciklusát. Meg kell győződni, hogy az adott konferencia a tervezettnél megfelelően kezdődött, illetve fejeződött-e be, vagy történt-e bármi egyéb változtatás, például hosszabbítás. Szintén a szolgáltatás feladata az egyes események naplózása, különböző statisztikák vezetése.



Az Exchange Conferencing Server saját ismertető oldala

Data Conferencing Provider

A Data Conferencing Provider egy technológiaszolgáltató, mely több, a résztvevők együttműködését, információcseréjét megkönnyítő eszközt is biztosít. Ilyen például a közös információs tábla, szöveges beszélgetési lehetőség, alkalmazásmegosztás és fájlok átvitele. Ezek a szolgáltatások az ITU T.120-as multimédia tartalmazó hálózati kommunikációra vonatkozó szabványosozatát figyelembe véve kerültek megvalósításra, lehetővé téve így az átviteli és a hálózati erőforrások optimalizálását, valamint a távoli vagy tűzfalon keresztül történő kapcsolódást is.

Egy ilyen konferencia során minden résztvevő egy T.120-as központi konferencia adatszolgáltató és irányító egységhez kapcsolódik (*multipoint control unit = MCU*). Ezeknek az egységeknek az a feladata, hogy összekapcsolják a résztvevőket, hozzájuk a konferencia adatait eljuttassák és



szinkronizálják a köztük lezajló eseményeket. Egy conferencing site-on belül több MCU is telepíthető, biztosítva így a terheléelosztást, valamint a nagyobb rendelkezésre állást. Amikor egy felhasználó csatlakozik egy konferenciához, a Data Conferencing Provider a következő szempontok alapján megpróbálja kiválasztani a legjobb MCU-t:

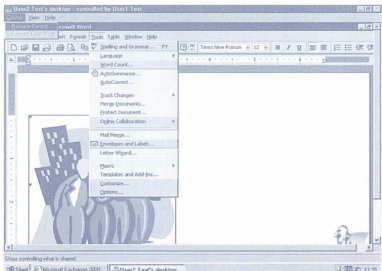
- ☞ Melyik MCU-k állnak rendelkezésre?
- ☞ Szolgál-e ki már konferenciát az adott MCU?
- ☞ Mekkora a terheltsége az MCU-nak?
- ☞ Van-e bármiféle adminisztratív korlátozás, hogy melyik MCU használható?
- ☞ Hol helyezkedik el a hálózatban a résztvevő?
- ☞ A résztvevő a helyi conferencing site-on belüli vagy kívüli, esetleg a szervezett teljes hálózatán kívüli?

A paraméterek alapján a Data Conferencing Provider csatlakoztatja a résztvevőt egy MCU-hoz, mely hozzá a legközelebb van a hálózatban és a legkevésbé terhelhető, így optimalizálva a továbbítandó adatmennyiséget. Ennek következtében az MCU-k közötti kapcsolatok dinamikusan változhatnak a résztvevők számának változásával együtt. Az MCU-k bridge-ként is szolgálhatnak egy WAN kapcsolat két végpontján. Ebben az esetben a lassú WAN kapcsolaton a konferencia adatai csak egy példányban mennek keresztül, akárhányan is csatlakoznak a végpontokhoz, feltéve, hogy mindenki a hozzá közelebb eső MCU-hoz kapcsolódik (ez adminisztratív korlátozásokkal biztosítható).

A T.120 MCU-k függetlenül telepíthetők a többi komponensnél. Minden MCU a Certificate Service-től kapott tanúsítvány segítségével kölcsönösen azonosíthatja magát a résztvevőkkel és a kommunikációban szereplő többi MCU-val szemben.

A Data Conferencing Provider a következő eszközöket biztosítja a résztvevők számára:

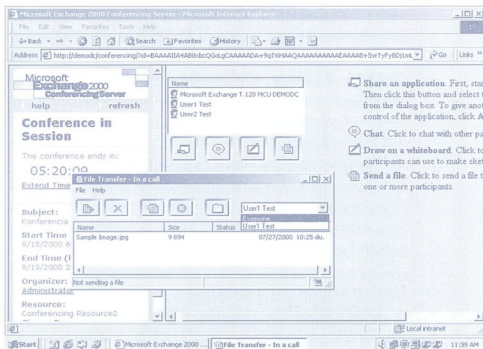
- ☞ Alkalmazásmegosztás. Tetszőleges program, amely a résztvevők valamelyikének a gépén fut, megosztható a többi résztvevővel. A résztvevők láthatják ugyanazokat az információkat, adatokat, a program futása során bekövetkező eseményeket úgy, hogy az alkalmazás nincs is telepítve a gépükön. A megosztást létrehozó résztvevő engedélyezheti a többi résztvevő aktív közreműködését is, így lehetővé téve a résztvevők számára a beavatkozást.



Alkalmazásmegosztás aktív közreműködéssel

A konferencia résztvevői is éppen úgy vezérelhetik az adott alkalmazást, mintha a saját gépükön dolgoznának. A NetMeeting 3.0-s változatától kezdve pedig lehetőség van a teljes desktopfelület megosztására is, lehetővé téve így a teljes gép felügyeletének átengedését más személyek számára.

- ☞ Közös vágólap. A résztvevők könnyen cserélhetnek adatot egyszínű másolás, kivágás, illetve beillesztés műveletek segítségével. Lehetőség van például a helyi adatok egy megosztott alkalmazásba történő bemásolására.
- ☞ Fájltvitel. A konferencia közben, a háttérben fájlokat küldhetünk át egy vagy több résztvevő számára. A fájltvitel a háttérben történik, a résztvevőknek nem kell megszakitaniuk addigi tevékenységüket.



Fájltvitel a konferencia résztvevői között

- ☞ Információs tábla. Különböző diagramok, előre elkészített ábrák jeleníthetők meg ezen a többlapos, több felhasználó által használt közös információs táblán. Az információs tábla objektumorientált (nem pixelalapú), lehetővé téve így az ábrák manipulálását. Emellett beépített mutatóeszközrel rendelkezik a fontos részekre való rámutatáshoz.
- ☞ Online beszélgetési lehetőség. Egyszerűen begépelhetjük a megbeszélés közben támadt ötleteinket, gondolatainkat. Az így elhangzott gondolat rögzíthetők is egy fájlba, megkönnyítve így a későbbi emlékeztető írását. A NetMeeting lehetőséget kínál elküldött privát beszélgetések lefolytatására is a konferencia egy bizonyos résztvevőjével, vagy azok egy csoportjával.

Video Conferencing Provider

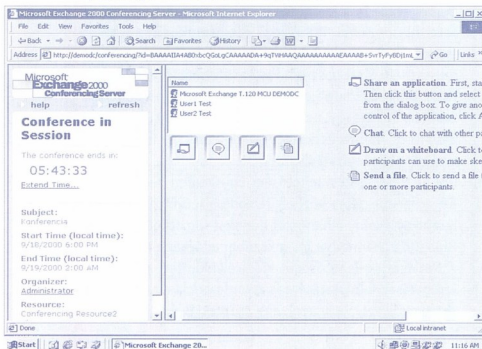
Video Conferencing Provider szintén egy technológia-szolgáltató, ami lehetővé teszi a két vagy több résztvevős élő hang és videokonferenciák lebonyolítását. A Video Conferencing Provider már nemcsak a NetMeeting 3.01 által kínált az ITU-T H.323-as szabvány sorozatában definiált unicast alapú pont-pont kapcsolat közötti videó- és hangátvitelt támogatja, hanem a több résztvevős multicastalapú videó- és hangátvitelt is. Így a konferencia résztvevői egyidejűleg láthatják és hallhatják egymást anélkül, hogy

szükség volna több párhuzamos telekonferenciára. A Video Conferencing Provider által kínált multicast alapú konferenciákhoz a H.323 szabványú kompatibilis unicast kliensek egy IP multicast/H.323 unicast bridge-n keresztül kapcsolódhatnak. Ezek a bridge-k MCU-k formájában telepíthetők az Exchange Conferencing Server részeként.

Egy videókonferencia létrejöttkor a Video Conferencing Server egy speciális multicast IP címet igényel a Multicast Address Dynamic Client Allocation Protocol (MADCAP) segítségével a Windows 2000 DHCP szerverétől, ami lefoglalásra kerül a konferencia számára. Egy ilyen cím egészen addig nem osztható ki újból, amíg a foglalási idő le nem jár. A foglalási idő a konferencia tervezett időtartamából és a maximálisan engedélyezett hosszabbítás időtartamából tevődik össze. A maximálisan engedélyezett hosszabbítás időtartama az az időtartam, amennyi idővel (percenként mérve) a konferencia szervezője meghosszabbíthatja a konferencia időtartamát. (Ez a paraméter a Conferencing Manager-ből állítható.) Egy tiszta multicast alapú videókonferencia esetén, szerverek nem vesznek részt a videó- és hangátvitelben, mert a kliensek közvetlenül a multicast címről veszik, illetve küldik az adatokat. Amennyiben routerek is vannak a kliensek között a hálózaton, akkor azok továbbítják a kliens hálózati szegmensére ezeket a multicast címmel rendelkező csomagokat.

Konferenciaelérési oldalak

A konferenciaelérési oldalak (conference access pages) segítségével érhetik el a felhasználók az aktuális konferenciákat, megbeszéléseket. Ezeket az oldalakat az IIS szerver publikálja a felhasználók számára.



A felhasználók internetböngészőjük segítségével érhetik el a különböző konferenciákat, megbeszéléseket

A konferenciákra szóló meghívó tartalmazza a konferencia elérését tartalmazó URL-t. Egy ilyen URL két fő részből tevődik össze: az első rész tartalmazza az alap elérési útvonalat (szervernév, mappa), a második rész pedig a konferencia azonosítóját.

Külső komponensek

Az Exchange Conferencing Server üzembehelyezéséhez, illetve az általa nyújtott szolgáltatások teljes körű kihasználásához számos hálózati szolgáltatás és ügyféloldali alkalmazás előzetes megléte szükséges a Windows 2000 infrastruktúrában:

Active Directory

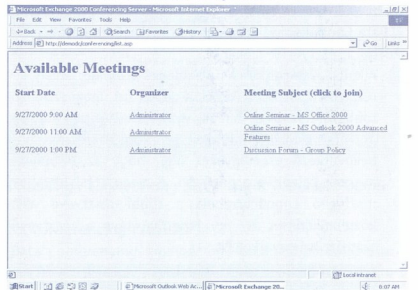
Az Exchange Conferencing Server több célra is használja az Active Directory címtárszolgáltatását. Innen veszi a számára szükséges hálózati topológiára vonatkozó információkat, valamint itt tárolja konfigurációs adatait.

Exchange 2000 Server

Az Exchange Conferencing Server az Exchange 2000 Server-t használja a konferenciák időpontjainak, erőforrásfoglalásainak nyilvántartására. Az Exchange Conferencing Server kétfajta postafiókot használ ezekre a célokra: konferencia-naptár postafiók (conference calendar mailbox) és konferenciaerőforrás postafiók (conference resource mailbox). A konferencia-naptár postafiókban tárolódnak a konferenciákra vonatkozó leíró és strukturális információk. A konferenciaerőforrás postafiók feladata pedig az, hogy a felhasználók rajta keresztül jelezzék, illetve biztosítsák a virtuális rendezvényekre vonatkozó erőforrásigényüket.

IIS

A Conferencing Server az Internet Information Server-t használja a konferenciaelérési oldalak (conference access pages) publikálására. A felhasználók ezeken az oldalakon szerezhetnek információkat a nyilvános konferenciákról.



Az aktuálisan elérhető konferenciák oldala

MADCAP szerviz

A hálózatnak támogatnia kell a Multicast Address Dynamic Allocation Protocol-t (MADCAP) valamint a routereknek a multicast alapú üzenettovábbítást ahhoz, hogy multicast-alapú videókonferenciák legyenek lebonyolíthatók. A Windows 2000 DHCP Server-e támogatja mind a multicast, mind a unicast címtartományokat egyaránt. A DHCP adminisztrátor segítségével könnyen konfigurálható a megfelelő IP címtartomány és hatókör.

PKI infrastruktúra

Zártkörű konferenciák, tárgyalások lebonyolítása során az MCU a Certificate Service szolgáltatásait használja. A Certificate Service által kiadott tanúsítványok alapján történik a konferenciák résztvevőinek, valamint az MCU-k egymás közötti azonosítása is. A konferencia résztvevői a saját tanúsítványaikkal igazolhatják, hogy jogosultak egy adott konferencián, megbeszélésen való részvételre.

Windows kliens

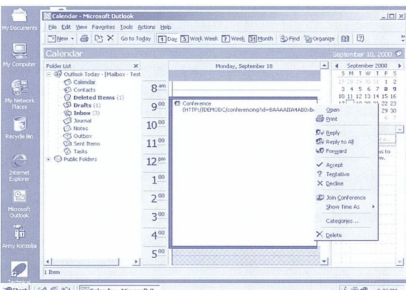
A Windows 2000 alapú ügyfelek résztvehetnek multicatalapú videó konferenciákban, miután a Windows 2000 teljes körű multicast támogatással rendelkezik. Emellett a Windows 2000 tartalmazza a Telephony Application Programming Interface (TAPI) 3.0 elemeit is. A konferencia szoftverek a multicatalapú videokonferenciákban való részvételhez.

Internetböngésző

A konferenciák elérési oldalainak megtekintéséhez legálább Internet Explorer 4.01 vagy annak újabb változata ajánlott, de bármilyen, a frame-et és a JavaScript-et támogató böngészővel elérhetőek ezek az oldalak.

Outlook 2000

A konferenciák, megbeszélések szervezői levelezőprogramjuk segítségével tudják ütemezni a rendezvényeket, illetve lefoglalni az ezekhez szükséges erőforrásokat. Elsődlegesen Outlook 2000 a javasolt levelező kliens, de az Outlook korábbi verziói is megfelelőek. Emellett az Outlook Web Access segítségével is szervezhetünk konferenciákat.



Csatlakozás egy konferenciához közvetlenül az Outlook 2000-ből

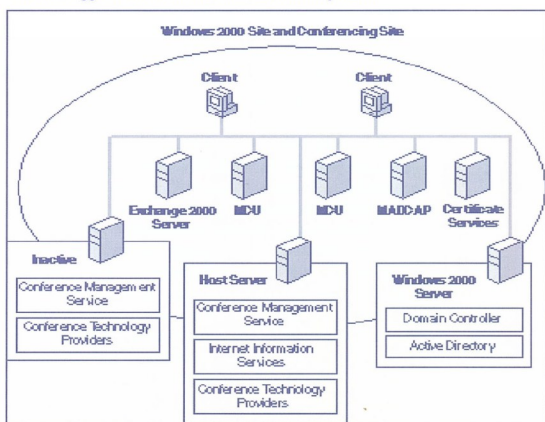
NetMeeting

NetMeeting 3.01 a javasolt T.120 kliens. A NetMeeting segítségével a felhasználók elérhetik az MCU-kat, lehetővé téve így számukra a különböző adatok küldését és vételét. Emellett bármilyen más, a T.120-as szabványsorozattal kompatibilis kliensalkalmazás használható. Amennyiben a NetMeeting nincs telepítve a ügyfélgépen, akkor a konferencia elérési oldalának letöltésekor a böngészőprogram kéri a használandó T.120 kliens alkalmazás megadását.

Exchange Conferencing Server környezet

Az Exchange Conferencing Server által nyújtott szolgáltatások hatóköre nemcsak magára a szolgáltatást nyújtó szerverre terjed ki, hanem az egész Windows 2000 site-ra, amelyben a Conferencing Server található. Ezért egy Windows 2000 site-on belül csak egy aktív Conference Management Service lehet. A site-on belül több példányban is telepíthető a Conference Management Service, viszont ezek a példányok inaktívak lehetnek csak, és terheléselosztásra nem használhatók. Esetleges meghibásodás esetén egy ilyen példány aktiválásával azonban továbbra is biztosíthatók a konferenciaszolgáltatások. A Conferencing Server terhelésének tervezésekor figyelembe kell venni, hogy a site-on belüli összes konferenciafoglalás és ügyfélkérés feldolgozását a Conference Management Service egyetlenegy aktív példánya végzi. A site-on belül a Conference Management Service első telepítésével létrejön és aktiválódik a conferencing site. Egy conferencing site-on belül több technológiaszolgáltató is telepíthető.

Az Exchange Conferencing Server jól skálázható megoldást kínál a konferenciaszolgáltatások kapacitása terén. Amennyiben a rendszer bővítésére igény merül fel (pl. az MCU-k túlzott leterheltsége miatt megnő az átlagos válaszidő) a Data Conferencing Provider lehetőséget biztosít a conferencing site-ba történő újabb MCU-k segítségével történő automatikus terheléselosztásra. Egy adott konferencia lebonyolítására mindig a legkevesebb leterhelt MCU kerül kijelölésre. Ezután a helyi résztvevők mindegyike ehhez az MCU-hoz kerül csatlakoztatásra, minimalizálva így a szükséges forgalmat. Ez mindaddig folytatódik, míg az MCU el nem éri a bekonfigurált maximális terheltséget, ekkor a következő résztvevő már egy másik, aktuálisan a legkevesebb leterheltséggel rendelkező MCU-hoz kerül átirányításra.



Minden conferencing site saját konferenciaelérési oldalakkal rendelkezik. A felhasználók ezeket az oldalakat használják a konferenciák elérésére minden site-ban. Egy-egy konferencia kiterjedhet több conferencing site-ra is.

Biztonsági szempontok

Az Exchange Conferencing Server különböző biztonsági szolgáltatásokat kínál a konferenciák elérésével és az azokban való részvétellel kapcsolatban. Egy adott konferencia lebonyolításáért felelős technológiaszolgáltató nemcsak a részvétel módját határozza meg, hanem annak biztonsági szintjét is. A konferenciák létrehozásához szükséges jogosultságok, valamint az elérésükhöz szükséges jogosultságok ellenőrzése viszont a Conference Management Service feladata.

A konferenciák elérésének ellenőrzése

Egy új konferencia megszervezésekor meghatározható, hogy az teljesen nyilvános vagy nyilvános, de jelszóval védett vagy privát legyen.

- ☞ Nyilvános konferenciák: ezek teljesen nyilvánosak, bármelek felhasználó, aki eléri a konferencia URL-t, részt vehet ezeken a rendezvényeken. Alapértelmezés szerint a konferenciák nyilvánosak. Valamennyi nyilvános konferencia felsorolása megtalálható és elérhető a konferenciaelérési oldalakon keresztül (http://<server_name>/conferencing/list.asp). Az Outlook 2000-nél korábbi Outlook verziókkal és az Outlook Web Access for Exchange Server 5.5-tel csak ilyen konferenciákat lehet szervezni.
- ☞ Jelszóval korlátozott nyilvános konferenciák: ebben az esetben csak azok a felhasználók csatlakozhatnak a konferenciához, akik birtokában vannak a megfelelő jelszónak. A technológiaszolgáltató ezt a jelszót használja fel a hozzáférések korlátozására. Ilyen konferenciák csak az Outlook 2000 segítségével szervezhetők az „Allow external attendees” jelölőnégyzet beiktatásával, és persze a meghívásban a jelszó elküldésével a résztvevők számára.
- ☞ Privát konferenciák: ezekhez a rendezvényekhez csak azok a felhasználók csatlakozhatnak, akik szerepelnek a meghívottak listáján. Ebben az esetben a Conference Management Service, vagy a technológia-szolgáltató, vagy mindkettő ellenőrzi a résztvevő jogosultságát a konferenciához való hozzáféréskor. Ha a Conference Management Service végzi ezt az ellenőrzést, akkor az IIS-t használja a felhasználó jogosultságainak ellenőrzésére. A technológiaszolgáltató további biztonsági szolgáltatásokat nyújt a privát konferenciák számára. Például a Data Conferencing Provider X.509 v3 formátumú tanúsítványokat használ a felhasználók autentikálására, illetve a konferencia adatainak titkosítására.

Telepítés

Az Exchange Conferencing Server Windows 2000 Server-re vagy Windows 2000 Advanced Server-re telepíthető. A telepítés feltételei a következők:

- ☞ Active Directory
- ☞ Microsoft Exchange 2000 Server a tartományban
- ☞ Microsoft Internet Information Server a site-ban
- ☞ Exchange System Manager a szerverten

A használni kívánt Exchange Conferencing Server szolgáltatásoktól függően szükséges lehet még:

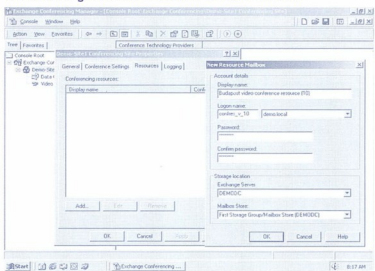
- ☞ Windows 2000 MADCAP Services
- ☞ Windows 2000 Certificate Services

Egy teszt-, illetve bemutatókörnyezet kialakítása során mind-ezek a szolgáltatások lehetnek egyetlennegy szerverten, amik aztán később folyamatosan kiterjeszthetők az igényeknek megfelelően.

A telepítés során választható teljes vagy egyedi telepítés. Teljes telepítés esetén mind a Conference Management Service, mind a technológiaszolgáltatók telepítésre kerülnek. Egyedi telepítés esetén kiválaszthatók a telepíteni kívánt komponensek. Ez utóbbi módszerrel adhatók újabb MCU-k a conferencing site-hoz.

Konfigurálás

Az installálást követően létre kell hozni a konferenciaerőforrás postafiókokat (*conference resource mailboxes*). Célszerű ezeket olyan névkonvenciót kidolgozni, amely megfelelően beszédes a felhasználók számára. Például „Veszprém videó konferencia (10 fő)”, ami egy maximum 10 résztvevős videótávtelre is lehetőséget biztosító veszprémi erőforrást jelöl a felhasználók számára. Az erőforrások jobb kihasználása végett javasolt az azonos erőforrásokból több különböző méretű (*résztevő számú*)t létrehozni. Ezeket a postafiókokat, valamint a konferenciánaptár postafiókot célszerű külön adatbázisban tárolni a többi postafióktól, hogy minél könnyebb és gyorsabb legyen egy esetleges hardvermeghibásodás esetén a visszaállítás.



Konferenciaerőforrás postafiók létrehozása

Kliensoldalon az Outlook 2000-ben engedélyezni kell a konferenciaerőforrások kezelését. Ez az alábbi registry kulcs létrehozásával történik:

HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Outlook\ExchangeConferencing

Ezen konfigurációk elvégzése után a felhasználók már használhatják a Microsoft Conferencing Server által nyújtott szolgáltatásokat. Fontos megjegyezni, hogy a konferenciák közvetlenül a szervertől nem érhetők el, csak ügyfél gépről.

Simon Csaba, Csaba.Simon@Synergon.hu

Az SQL Server 2000 új adminisztrációs eszközei

Az SQL Server 2000 fontos lépés a Microsoft .net kezdeményezése felé vezető úton. Ennek jeleit felismerhetjük azokból a legfontosabb változásokból, amelyek az SQL Server 2000 megbízhatóságát és teljesítményét jellemzik. Ebben a cikkben azokat a fő újdonságokat foglaljuk össze, amelyek a termék adminisztrációs képességeit jellemzik. Célunk nem a teljes mélységű technikai leírás, hanem a képességek felhasználási lehetőségeinek bemutatása:

- ☞ Újdonságok a telepítésben
- ☞ Adatbázisok kompatibilitása
- ☞ Kódlapok és rendezések
- ☞ Az új SQL Query Analyser
- ☞ Az új SQL profiler
- ☞ Log Shipping

Újdonságok a telepítésben

Az SQL Server 2000 az első SQL Server változat, amelyik több példányban is feltelepíthető egy számítógépre. Az első hallásra sokak számára meglepő gondolat mások számára a „végre!” felsóhajtást jelenti. Mi értelme van ennek? Ha valaki kevés adatbázist használ, jól meghatározott felhasználói körrel, akkor valószínűleg nem sok. Ha valaki SQL Server-en fejleszt, a dolognak már több értelme van: el lehet egymástól választani a fejlesztői, tesztes és éles üzemi környezeteket és még csak külön számítógép sem kell hozzá.

A legnagyobb változást azok számára jelentheti ez, akik alkalmazásgépként foglalkoznak: egy „jó nagy” számítógépre (amely Windows 2000 DataCenter esetében akár 32 CPU-t is kezel) több SQL Server-t is telepítve, kívülről ezeket több külön számítógépen lévő SQL Server-ként érzékelhetjük. Ez segíthet a különböző ügyfelek adatainak elkülönítésében, de akár a CPU sávszélesség definíálásában is (az SQL Serverrel az ún. „affinity mask” segítségével megmondhatjuk, hogy az SQL Server mely CPU-kat használhatja).

Amennyiben a telepítés pillanatában még nincs SQL Server-ünk telepítve, az elsőnek telepített szervert az ún. alapértelmezett példány lesz, neve a számítógép nevével fog megegyezni. A további példányok ún. új példányok („new instance”) lesznek és egy példánynéval kell őket ellátnunk. Ezekre a számítógép nevének és a példány nevének kombinációjával hivatkozhatunk.

Ha a szervertünk TESTSQL-nek hívják és az alapértelmezett példányok mellé még az SQL1 és SQL2 példányokat is telepítettük, akkor az elnevezések:

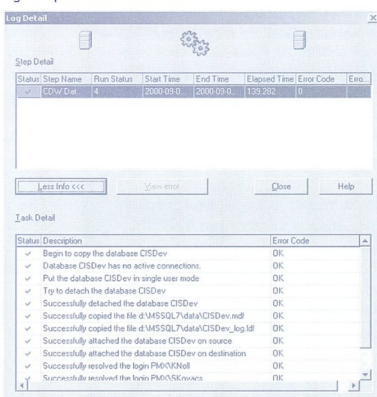
- ☞ TESTSQL
- ☞ TESTSQL\SQL1
- ☞ TESTSQL\SQL2

A többpéldányos telepítés során az SQL 2000 képes az SQL Server 6.5 és 7.0 mellé települni, ebben az esetben természetesen csak új példányok telepítésével. Egy számítógépet fel lehet telepíteni úgy, hogy egy SQL 6.5-ös, egy SQL 7-es és néhány SQL 2000-es szervert is tartalmazzon. Az SQL 6.5-ös és 7-es szerverek közül ugyan csak az egyik futhat (mindig az, amelynek környezetét éppen aktuálisnak állítottuk be a „Microsoft SQL Server Switch” menüben), de ezeket párhuzamosan az összes SQL 2000-es példány is futhat.

Adatbázisok kompatibilitása

Akik végigjárták SQL 6.5-ben fejlesztett alkalmazások felfeljesztését SQL 7-ra, azok tudják, hogy az Upgrade Wizard nevű segédeszköz ehhez komoly segítséget nyújtott. Az SQL 6.5-ről az SQL 2000-re történő átállásban ugyanez az eszköz nyújt segítséget. De hogyan történik SQL Server 7-ről SQL Server 2000-re az adatbázisok átállítása? Igen egyszerűen, az ún. Copy Database Wizard segítségével. Ez a segédeszköz SQL 7-es, illetve SQL 2000-es adatbázisokat tud mozgatni, illetve másolni. A varázsló indítása után megadhatjuk a forrás és cél adatbázisokat (ezek természetesen különböző szervereken lehetnek), s azokhoz tartozó másolási opciókat (mit másoljunk még az adatbázissal együtt).

A varázsló egy ún. „package”-et hoz létre, amely futtatása elvégzi a kért adatbázismásolást. Az alábbi ábrán a „package” lépéseit mutatom be:



A Copy Database Wizard által végrehajtott „package”

Mit is csinál ez az eszköz? Először is meggyőződik arról, hogy senki nem használja az adatbázist, majd egyfelhasználós módba helyezi, hogy a másolás alatt se tudjon senki hozzáférni. Ezek után lemásolja az eredeti adatbázisfájlokat (s nem egyenként az adatbázisobjektumokat!), majd „élesíti” az SQL 2000 alatt a lemásolt fájlokat.

Az SQL Server 2000 tehát változatlan formában „megeszi” az SQL Server 7 fájljait, anélkül, hogy azokat bitről bitre átgymá! Az SQL Server 2000 fájlformátuma felülről kompatibilis az SQL Server 7-tel.

Azt, hogy az adatbázisunk hogyan viselkedjen az SQL Serveren (milyen SQL Server változatot legyen kompatibilis), az adatbázishoz kapcsolódó „Properties” dialógus „Options” lapján állíthatjuk be, illetve az sp_dbcmptlevel tárolt eljárással. Ha például az előzőleg átmásolt adatbázis kompatibilitási módját SQL Server 7-re állítjuk, nem tudunk felhasználói függvényeket létrehozni abban.



Kódlapok és rendezések

Az SQL Server korábbi változatainál a telepítés során meg kellett adnunk egy kódlapot és egy rendezési metódust, amelyet az SQL Server élete során használni kívántunk. Ha ezt nem jól választottuk meg, komoly tortúrát jelenthetett egy adott nyelvi beállításról áttérni egy másikra. Az SQL Server 2000 ezen a területen forradalmi változást hozott. A telepítéskor automatikusan felkínált beállítás *(amelyet természetesen megváltoztathatunk)* „csak” alapértelmezett beállítást, de azt megváltoztathatjuk, amikor egy adatbázist létrehozunk. Az új adatbázis létrehozásakor megjelenő dialógusablak „General” lapján található „Collation name” listából kiválaszthatjuk a kívánt nyelvi beállítást, amely meghatározza az adatbázisban használt Unicode adatok értelmezését, illetve a rendezési sorrendet. Például a „Hungarian_CI_AS” beállítást választva a magyar szótag szerinti kis- és nagybetűkre érzéketlen, de az ékezeteket figyelembe vevő rendezési sorrendet állíthatunk be.

Aki kíváncsi az összes támogatott nyelvi beállításra, az alábbi lekérdezéssel ezt megnézheti:

```
SELECT * FROM ::fn_helpcollations()
```

Ez a lista 753 sort tartalmaz! A rendezési sorrendet azonban nem csak az adatbázis szintjén lehet megadni, hanem egy tábla oszlopainak szintjén is, amint azt az alábbi példa is mutatja:

```
CREATE TABLE [dbo].[TestTable](
  [cID] char (5) NOT NULL,
  [cHunName] nvarchar(80)
    COLLATE SQL_Hungarian_CP1250_CI_AS NOT NULL,
  [cPolName] nvarchar(80)
    COLLATE SQL_Polish_CP1250_CI_AS NOT NULL
)
```

A tábla [cHunName] mezőjét magyar, a [cPolName] mezőjét pedig lengyel szótagi rendezéssel kívánjuk használni. Az SQL Server azonban még ennél is továbbmegy! Ha mégiscsak szükségünk lenne arra, hogy egy mező definiált sorrendjétől eltérő rendezést jelenítsünk meg, ezt is megtehetjük:

```
SELECT * FROM TestTable
ORDER BY cHunName
    COLLATE Hungarian_CS_AS
```

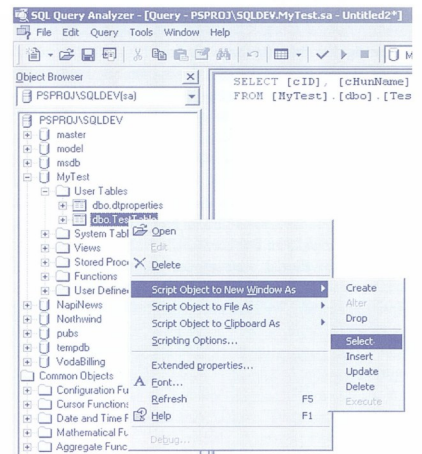
A COLLATE klauza segítségével az ORDER BY-ban is megadhatjuk a kívánt rendezési sorrendet.

Az új SQL Query Analyzer

Az SQL Server 2000 kliens oldali segédeszközei a korábbi változathoz hasonlóak maradtak. A telepítés után a „Configure SQL XML Support in IIS” eszköz az egyetlen új amelyik megjelenik *(ennek használatát az SQL Server 2000 XML támogatását tárgyaló cikk részletezi)*.

Az eszközök megjelenése és használata a korábbi változatoknak megfelelő maradt, kivéve az SQL Query Analyzer-t.

Ez az SQL Server 7-es változatában is már komoly fejlődést mutatott a grafikus végrehajtási tervek megjelenítésével és az indexanalízis lehetőségével, a legújabb változata azonban már szinte a fejlesztőeszköz magasságába emeli a Query Analyzert. Nézzük meg a legfontosabb újdonságokat! Indítás után a képernyő bal oldalán megjelenik az Object Browser, amelyet az SQL Server-en lévő rendszer és felhasználói objektumok között tallózhathatunk. Az objektumokra jobb egérgombbal kattintva helyzetérzékeny menü jön elő. Ezzel gyorsan és egyszerűen tudunk alapvető feladatokat végrehajtani az objektumokon. Az alábbi ábrán azt látjuk, hogyan lehet a helyzetérzékeny menü segítségével pár kattintással egy SELECT utasítást létrehozni egy táblához:



Az Object Browser és a helyzetérzékeny menük

Az objektumokat a fogd-és-vidd technikával is „fel lehet dobálni” az SQL lekérdezések összeállításá során. Bármely objektum nevét oda lehet „vonszolni”, de például egy táblához tartozó „Columns” mappát vonszolva a tábla mezőinek neve egymástól vészóvel elválasztva kerülnek bemásolásra a lekérdezés szövegébe.

Az új funkciók között találjuk az Object Search ablakot, amelyben lehetőségünk van az SQL Server objektumai között keresgélni. Egyszerre akár több ablakot is nyitva tarthatunk, több különböző feltétellel.

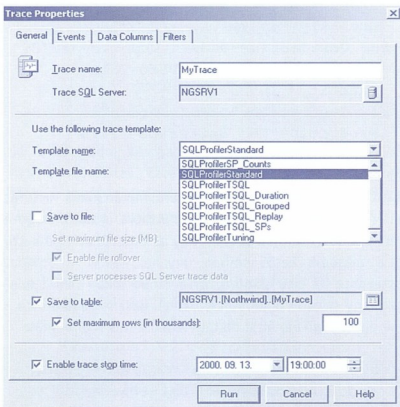
A Query Analyzer Tools menüjében található meg a Manage Indexes és Manage Statistics funkciókat. A Manage Indexes funkció megegyezik az SQL Enterprise Manager-ben található funkcióval *(ez használható indexek létrehozására, módosítására és törlésére)*. A Manage Statistics funkció segítségével statisztikákat tudunk definiálni *(mezők értékeinek eloszlási statisztikáit)*, amelyeket a későbbiekben meg is tekinthetünk. Az indexekhez tartozó eloszlási statisztikákat az SQL Server „query optimizer”-e használja a végrehajtási terv elkészítéséhez.

A Tools menü Customise menüfunkciójával tárolt eljárásokat rendelhetünk egy-egy nyomógomhoz. Azok lenyomása esetén a megfelelő tárolt eljárás végrehajtódik, s eredménye megjelenik. Ugyanezzel a funkcióval lehetőségünk nyílik arra, hogy a Tools menübe további eszközöket vegyünk fel. A Query Analyzer egy új funkciója az „Insert Template”, amellyel egy előre összeállított sablonyüteményből választhatjuk ki Transact-SQL parancssorozatokat. Ezek a sablonok paramétereiket is tartalmazhatják, amelyek aktuális értékeit az Edit menü Replace Template Parameters funkciójával állíthatjuk be, majd végrehajthatjuk a kitöltött sablonnak megfelelő parancsokat. Ezt a sablonyüteményt magunk is könnyen bővíthetjük egy-egy általunk gyakran használt Transact-SQL utasítással.

Az új SQL profiler

Az SQL Profiler egy olyan grafikus felülettel rendelkező eszköz, amely lehetővé teszi a szerveren történő események monitorozását. Az események – amelyek kiválasztását az SQL Server 2000-ben már sablonok is segítik – elmenthetők fájlba és táblába egyaránt. Az így elmentett események „visszajátszhatók”, vagyis eredeti sorrendben újra lefuttathatók a lekérdezések, tárolt eljárások stb. Az SQL Profiler jól használható:

- ☞ Teljesítmémmérésre
- ☞ Transact-SQL utasítások hibakeresésére
- ☞ Lassú lekérdezések beazonosítására és optimalizálására
- ☞ Működő rendszereken hibafeltárára az elmentett események teszrendszeren való „visszajátszásával”
- ☞ A szerveren végzett tevékenységek auditálására, biztonsági naplózásra



Trace jellemzők ablak

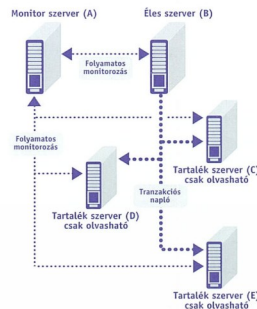
Az SQL Server 2000-ben megújult SQL Profilerrel találkozunk. Az egyik legfontosabb tulajdonsága, hogy olyan részletes eseményrögzítést, naplózást és auditálást tesz lehetővé, amelynek köszönhetően az SQL Server 2000 teljesíti az Egyesült Államokbeli C2-es biztonsági szintet. Rögzíthetők a

felhasználók és az adatbázis adminisztrátorok tevékenységei, ezen kívül biztonsági események (jogsorvitások megváltozása), szerver események (leállás, indulás), adminisztratív események (backup, BCP, DBCC, stb.) és az auditban bekövetkezett változások.

Az új Profilerben nem ekkora a monitorozható események száma is, ezen kívül az események kiválasztását sablonok teszik egyszerűbbé. A sablonokba nem csak a kiválasztott események, hanem oszlopok (mezők) és szűrők is beletartozhatnak. A képen a lenyílt listában az előre definiált sablonok találhatók. Ezeket kivül tetszőleges felhasználói sablonok is kiegészíthetjük. Szintén újdonság (a képen is látható), hogy az összegyűjtött adatok mennyisége és a futás ideje is beállítható.

Log Shipping

Az SQL Server 2000 a Microsoft Clustering Service-re alapuló fűrtözést támogat, amely hibatűrő architektúrák kiépítésére alkalmas. Emellett sok esetben kielégítő megoldást jelenthet (és költségkímélőbb is) az SQL Server 2000 Enterprise Editionben újdonságként megjelenő Log Shipping. A Log Shipping lényege, hogy az adatbázis tranzakciós naplója folyamatosan átmásolásra kerül a tartalékiszolgálón található másik adatbázisba. Ezáltal a két adatbázis szinkronizációja biztosított és az elsődleges szerver meghibásodása esetén a másolatot tartalmazó szerver átveheti a kérések kiszolgálását.



A Log Shipping működése

Az ábrán öt szerver található. A „B” szerver a forráskiszolgáló (source), a „C”, „D”, „E” szerverek a tartalékiszolgálók, vagy a tranzakciós napló másolása szempontjából a célszerverek (destination). Az „A” kiszolgáló a monitorozó szerepét tölti be, amely folyamatosan figyeli a backupok elmentéséhez és visszaállításához szükséges időt. Ha ez az idő egy előre beállított értéket meghalad, akkor a monitorozókiszolgálón az SQL Server Agent figyelmeztetést generál.

A Log Shipping konfigurálása egyszerű, a Database Maintenance Wizard segítségével könnyen elvégezhető.

Nóvák István, inovak@protomix.hu
Dömsödi Attila, adomsodi@protomix.hu



TECH.NET

Az XML támogatása az SQL Server 2000-ben



Az elektronikus kereskedelmi site fejlesztések kiemelt fontossága miatt a Microsoft SQL Server 2000 egyik legizgalmasabb újítása a beépített XML támogatás. Ennek köszönhetően az SQL Server 2000 XML alapú adatbáziskezelőként is működtethető – a hagyományos adatbázis-elérési módszerek mellett. Az XML támogatáshoz a következő újításokat találhatjuk meg a termékben:

- XML adatok közvetlen lekérdezése és feltöltése a szerverre
- XML Data Schema támogatás
- XPath lekérdezések használata a sémákra
- Az SQL Server elérése HTTP protokollon keresztül

Bevezetés

A teljes XML támogatáshoz nemcsak magát a szerveret, hanem a szerverhez kapcsolódó külső eszközöket is tovább kellett fejleszteni, így az OLE-DB meghajtó programok is támogatják az új funkciókat. Így vált lehetségessé, hogy lekérdezéseinket nemcsak SQL parancsként, hanem XML dokumentumként is megadhatjuk a szervernek, valamint a parancsok kimenete is lehet egy XML adatfolyam, szemben a hagyományos, bináris kimenettel (*Tabular Data Stream, TDS*).

A cikk az XML bemutatásával nem foglalkozik, annak legalább felületes ismeretét feltételezzük az olvasóról. A felhozott példák a Northwind adatbázisra vonatkozó alábbi lekérdezésen alapulnak:

```
SELECT oh.CustomerID, oh.OrderID,
od.ProductID, od.UnitPrice, od.Quantity
FROM Orders oh INNER JOIN [Order
Details] od ON oh.orderid = od.orderid
WHERE oh.CustomerID = 'GROSR' ORDER BY
oh.CustomerID, oh.OrderID, od.ProductID
```

A lekérdezés futtatásának eredménye:

CustomerID	OrderID	ProductID	UPrice	Quant.
GROSR	10268	29	99.0000	10
GROSR	10268	72	27.8000	4
GROSR	10785	10	31.0000	10
GROSR	10785	75	7.7500	10

Látható, hogy a lekérdezés hierarchikus eredményhalmazt adna vissza, azaz kilistázza a megrendelések közül azokat, amelyekben a vásárlói azonosító GROSR majd ezekhez hozzáveszi az Order Details táblából a részleteket. Ezt a hierarchiát azonban a hagyományos SQL eredményhalmaz nem tudja visszaadni, így csak „kiterítve” kapjuk meg az eredményeket. De nem így lesz ez az XML kimenetek esetén! Előjáróban fontos elmondanunk, hogy a példák eredményei sokszor nem teljesen készre megformázott XML dokumentumok, hanem csak töredékek. Például a legtöbb lekérdezés kimenete olyan XML adatfolyam, amelynek az első szintjén több mint egy kifejezés áll. Ennek két oka van. Az első,

hogy a kapott XML dokumentumtöredéket tovább szokás formálni, hogy egy teljesen szabályos, az üzleti igényeknek megfelelő XML dokumentumot kapjunk. Másodsorú így az UNION művelet is egyszerűen megvalósítható XML-ben, hisz csak egymás után kell felülni a dokumentumokat. Egyes programok – mint az Internet Explorer – nem tudja az SQL szerver kimeneteként előállt XML adatfolyamot közvetlenül megjeleníteni, emiatt a legtöbb helyen gyökérbejegyzést hozunk létre, és az alá helyezzük el az XML dokumentumot.

SELECT for XML

A SELECT SQL utasítás egy új záradékot kapott, melyek révén a lekérdezés eredményhalmaza közvetlenül XML formátumban jut vissza az ügyfél programhoz. Ez a FOR XML záradék. Formátuma:

```
FOR XML {mode} [,XMLDATA][, ELEMENTS][, BINARY
Base64]
```

A {mode} segítségével szabályozhatjuk az eredő XML dokumentum formátumát, melyek lehetséges értékei RAW, AUTO és EXPLICIT. A RAW mód az SQL lekérdezés minden egyes sorát közvetlenül áttranszformálja egy-egy XML sorrá, azaz minden sor, és annak minden adata egy szinten fog elhelyezkedni. Másképpen fogalmazva a forrás eredményhalmaz mezői XML attribútumokká fordulnak le.

```
<row CustomerID="GROSR" OrderID="10268"
ProductID="29" UnitPrice="99"
Quantity="10" />
<row CustomerID="GROSR" OrderID="10268"
ProductID="72" UnitPrice="27.8"
Quantity="4" />
<row CustomerID="GROSR" OrderID="10785"
ProductID="10" UnitPrice="31"
Quantity="10" />
<row CustomerID="GROSR" OrderID="10785"
ProductID="75" UnitPrice="7.75"
Quantity="10" />
```

Az AUTO mód segítségével egyszerű hierarchikus XML fát épít fel a szerver. A példánkat lefutattva látható, hogy a szerver a JOIN alapján tudja, hogy az OrderID az első kettő és a második kettő sorban ugyanaz, ezért a sorokat összevonta az első szinten, és a csoportokon belüli adatokat egy szinttel lejjebb helyezte el, azaz visszaállította a lekérdezés logikája által diktált természetes hierarchiát:

```
<oh CustomerID="GROSR" OrderID="10268">
<od ProductID="29" UnitPrice="99"
Quantity="10" />
<od ProductID="72" UnitPrice="27.8"
Quantity="4" />
</oh>
<oh CustomerID="GROSR" OrderID="10785">
<od ProductID="10" UnitPrice="31"
Quantity="10" />
<od ProductID="75" UnitPrice="7.75" />
```




```
Quantity="10" />
</oh>
```

A harmadik mód, az EXPLICIT egy kicsit bonyolultabb. A megértéséhez előbb nézzük meg, mit jelentenek a mód mögött található kiegészítések. Az XMLDATA azt jelenti, hogy az adatok elé még sémainformációt is beszúr az SQL Server. AUTO záradékkal együtt a példa lekérdezésünkre a következő séma információk generálódnak le:

```
<Schema name="Schema" xmlns="urn:schemas-
microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-
com:datatypes">
  <ElementType name="oh" content="eltOnly"
model="closed" order="many">
    <element type="od" maxOccurs="*" />
    <AttributeType name="CustomerID"
dt:type="string" />
    <AttributeType name="OrderID"
dt:type="i4" />
    <attribute type="CustomerID" />
    <attribute type="OrderID" />
  </ElementType>
  <ElementType name="od"
content="empty"
model="closed">
    <AttributeType name="ProductID"
dt:type="i4" />
    <attribute type="ProductID" />
    ...
  </ElementType>
</Schema>
<oh xmlns="x-schema:#Schema">
<CustomerID>GROSZR</CustomerID>
...
```

Ha AUTO módot használunk, akkor alkalmazható az ELEMENTS kiegészítés. Ezzel azt jelezzük az SQL Servernek, hogy az eredményhalmaz oszlopai ne a csomópontattribútumokként jelenjenek meg, hanem gyermekelemekként:

```
<oh>
  <CustomerID>GROSZR</CustomerID>
  <OrderID>10268</OrderID>
  <od>
    <ProductID>29</ProductID>
    <UnitPrice>99</UnitPrice>
    <Quantity>10</Quantity>
  </od>
  ...
</oh>
<oh>
  ...
</oh>
```

A BINARY Base 64 a bináris adatok (képek, hangminták, OLE objektumok) Base 64 kódolását írja elő. Nyilvánvaló, hogy másképp nem lehetne a bináris eredményeket XML dokumentummá transzformálni, mivel az XML dokumentum egy szöveges állomány, ami nem képes bináris adatok közvetlen tárolására.

Ennyi kitérő után térjünk vissza az EXPLICIT záradékra. Láthattuk, hogy a különböző kiegészítésekkel (ELEMENTS, XMLDATA) szabályozhattuk, hogy az eredményhalmaz milyen XML struktúrába képződjék le. Azaz mi szabhatjuk meg, hogy mely oszlopok képződjenek le XML attribútumokká, és melyekből legyenek gyermekcsomópontok. Az EXPLICIT segítségével teljesen a kezünkbe vehetjük a legenerálendő XML struktúra szerkezetét. Ebben az esetben a mi felelősségünk és a mi szabadságunk, hogy az így keletkező XML dokumentum helyesen legyen megformázva. A záradék taglalása oldalakat igényelne, ezért ebben a cikkben nem térünk ki rá. A Books Online részletesen tárgyalja a használatát.

```
SELECT 1 AS Tag, NULL AS Parent,
oh.CustomerID AS [Order!!CustomerID],
oh.OrderID AS [Order!!OrderID],
NULL AS [OrderDetail!2!ProductID!
element],
NULL AS [OrderDetail!2!UnitPrice!
element],
NULL AS [OrderDetail!2!Quantity!element]
FROM Orders oh
WHERE CustomerId = 'GROSZR'
```

UNION ALL

```
SELECT 2, 1,
oh.CustomerID, oh.OrderID, od.ProductID,
od.UnitPrice, od.Quantity
FROM Orders oh INNER JOIN [Order
Details] od
ON oh.orderid = od.orderid
WHERE CustomerId = 'GROSZR'
```

```
ORDER BY [Order!!CustomerID],
[Order!!OrderID],
[OrderDetail!2!ProductID!element]
FOR XML EXPLICIT
```

A Query eredménye:

```
<Order CustomerID="GROSZR"
OrderID="10268">
  <OrderDetail>
    <ProductID>29</ProductID>
    <UnitPrice>99.0000</UnitPrice>
    <Quantity>10</Quantity>
  </OrderDetail>
  <OrderDetail>
    ...
```

```

        </OrderDetail>
    </Order>
    <Order CustomerID="GROSR"
    OrderID="10785">
        ...
    </Order>
    
```

OpenXML

Az előző fejezetben arról volt szó, hogyan nyerhetünk ki az adatbázisból XML formátumú adatokat. Ebben a fejezetben azt járjuk körül, hogy ha van egy XML dokumentumunk, akkor annak tartalmát hogyan írhatjuk vissza az adatbázisba, illetve segítségével hogyan módosíthatjuk a sorok tartalmát. A megoldás kulcsa az OpenXML függvény. Ennek segítségével egy XML dokumentumot átkövetelhatunk rowset-ekké, amelyekre az SQL Server már közvetlenül fel tud használni új adatok beszerzésére, vagy meglévőket módosítására.

Az OpenXML függvény használatához először el kell készíteni az XML dokumentum belső megjelenítését, azaz el kell fordítani egy olyan formátumra, amit a függvény közvetlenül tud kezelni. Ezt az `sp_xml_preparedocument` tárolt eljárással tehetjük meg. Ennek hatására a dokumentum egy bináris fára képződik le, amiben sokkal hatékonyabban tud keresni a szervert. Mint minden objektum, ez a fá is erőforrásokat köt le, tehát az alkalmazása után fel kell szabadítani az általa használt memóriát az `sp_xml_removedocument` rendszer tárolt eljárással. Az áttörzsforgalmazott XML dokumentumra egy kezelőszámot (*handle*) kapunk vissza, amely az OpenXML bemenetétől fogunk felhasználni.

Az OpenXML függvény szintaktikája a következő:

```

OPENXML(
    idoc int [in],
    rowpattern nvarchar[in],
    [flags byte[in]]
) [WITH (SchemaDeclaration | TableName)]
    
```

Az első két paraméter kötelező, a többi szabadon választható. Az `idoc` paraméter a `sp_xml_preparedocument` által szolgáltatott kezelőszám. A `rowpattern` feladata az XML dokumentum azon csomópontjainak azonosítása, amelyeket fel szeretnénk dolgozni. Ezt XPath kifejezésként adhatjuk meg (lásd a példát alább). A `flags` paraméter a kiinduló XML dokumentum és a függvény kimeneteként elvárt rowset adatok közötti transzformációt szabályozza. Alapértelmezett módon attribútumközpontú transzformációt használ, a második üzemmódban pedig elemközpontú XML dokumentumra számít a függvény.

Nézzük is meg egy példán keresztül, hogyan lehet az előző fejezetben generált XML dokumentumtörödéket (FOR XML AUTO példa) feldolgoztatni az Open XML függvénnyel:

```

declare @idoc int
declare @doc varchar(1000)
set @doc = '
<ROOT>
<oh CustomerID="GROSR" OrderID="10268">
  <od ProductID="29" UnitPrice="99"
  Quantity="10" />
  <od ProductID="72" UnitPrice="27.8"
  Quantity="4" />
</oh>
<oh CustomerID="GROSR" OrderID="10785">
  <od ProductID="10" UnitPrice="31"
  Quantity="10" />
  <od ProductID="75" UnitPrice="7.75"
  Quantity="10" />
</oh>
</ROOT>'
    
```

```

exec sp_xml_preparedocument @idoc
OUTPUT, @doc

SELECT * FROM OPENXML (@idoc,
'/ROOT/oh/od',1)
    WITH (ProductID varchar(10),
    UnitPrice varchar(20),
    Quantity varchar(20))
    
```

```
exec sp_xml_removedocument @idoc
```

A kódot lefuttatva a következő kimenetet kapjuk:

ProductID	UnitPrice	Quantity
29	99	10
72	27.8	4
...		

Igaz ugyan, hogy egy hierarchikus XML dokumentumot dolgoztunk fel, de a `rowpattern` paraméter megadásával egyszerűen előírtuk, hogy az adatok a `root/oh/od` szinten vannak, ahol ezután lineárisan lépkedünk végig. A `WITH` után láthatunk a sémadeklarációs részt. Ebben írjuk elő, sorrendben az XML attribútumok milyen SQL adattípusokra és oszlopnevekre képződjenek le. A példában azért ilyen egyszerű a sémadeklaráció, mert a választott SQL oszlopnevek megegyeztek az XML dokumentumban használt attribútumnevekkel. Ha más oszlopneveket akarunk használni, akkor a sémában meg kell jelölni az adatforrás-attribútum nevét. Például, ha a `ProductID` attribútumot `ProdID` néven szeretnénk megkapni a `SELECT` parancs eredményeként, akkor azt a séma definícióban kell jelezni:

```
ProdID varchar(10) '@ProductID'
```

Raadásul lehet hivatkozni a hierarchia más pontjára is, azaz bevehetünk attribútumokat a szülő, illetve gyermek csomópontokból is. Így az előbbi példában a `SELECT` kifejezést kicsérélve erre:



```
SELECT * FROM OPENXML (@idoc,
'/ROOT/oh/od',1)
WITH (CusID varchar(10)
'../@CustomerID',
OrdId varchar(10) '../@OrderID',
ProdID varchar(10) '@ProductID',
UnitPrice varchar(20),
Quantity varchar(20) '@Quantity')
```

A következő kimenet generálódik:

CusID	OrdId	ProdID	UnitPrice	Quantity
GROS	10268	29	99	10
GROS	10268	72	27.8	4
...				

Azaz a ../@CustomerID segítségével hivatkoztunk az egy szinttel feljebb található oh csomópont attribútumaira. A rowpattern paraméterben közvetlenül is hivatkoztunk egy konkrét attribútumra, így ezután arra egy ponttal hivatkozhatunk, a többi attribútumra pedig a korábban látott módon:

```
SELECT * FROM OPENXML (@idoc,
'/ROOT/oh/@CustomerID',1)
WITH (CusID varchar(10) '..',
OrdId varchar(10)
'../@OrderID',
UnitPrice varchar(20)
'../od/@UnitPrice')
```

Kimenet:

CusID	OrdId	UnitPrice
GROS	10268	99
GROS	10785	31

Figyeljünk arra, hogy az attribútum konkrét megadása miatt a szintek egyfeljebb csúsztak! Végülis hogyan alkalmazhatjuk az eddigiekben leírtakat adatok betételére? Az OPENXML kimenete egy szabványos SQL rowset, amelyet bármely olyan helyen használhatunk, ahol egyébként egy tábla vagy egy view volna, és emiatt minden egyéb műveletet is elvégezhetünk rajta, például szűrhetünk is rá a WHERE kifejezéssel. Vegyünk egy táblát, amibe az XML forrásunkból sorokat szeretnénk beszúrni:

```
CREATE TABLE [XMLOrders] (
[CustomerID] [varchar] (10),
[OrderID] [varchar] (10) )
```

A korábban is használt XML adatokkal a beszúrás így néz ki:

```
INSERT INTO XMLOrders
(CustomerID, OrderID)
SELECT * FROM OPENXML (@idoc,
'/ROOT/oh',1)
WITH (CustomerID varchar(10), OrderID
varchar(10))
```

XML nézetek

Transact SQL-ben akkor használtunk nézeteket (*view*), amikor egy tábla tartalmát függőlegesen vagy vízszintesen megsűrűve akartunk megmutatni a kívüllagnak, azaz egyes oszlopokat vagy sorokat el akartunk rejtetni a nyilvánosság elől, (vagy több táblából összegyűjtött sorokat akartunk megjeleníteni). Hasonló módon létrehozhatunk olyan XML nézetet, amely egy tábla vagy egy bonyolultabb lekérdezés eredményét jeleníti meg egy megformázott XML adatfolyamként. Az ilyen nézetek neve Annotated XDR Schema. Az XDR az XML-Data Reduced rövidítése, mely egy széleskörűen elfogadott XML szabvány.

Az „annotation”-ók szolgálnak arra, hogy a virtuális XML dokumentum attribútumai és az SQL szerver virtuális táblái illetve a táblák oszlopai közötti összerendelést kijelöljék. Lehetséges az így létrehozott virtuális XML nézetek között relációk létrehozása is, hasonlóan a fizikai SQL adatbázis táblákhoz. Az XDR leírnyelv és az XDR sémák részletesebb ismeretése túlmutat cikkünk keretein, így csak egy példa XML nézetet mutatunk meg, amelyet az XPath lekérdezések tárgyalásánál fogunk felhasználni. Látható benne, hogy létrehozunk két nézetet, az egyik az Orders SQL táblát képezi le egy Order nevű XML nézetre, míg a második az Order Details nevű táblából készít egy OrderDetail nevű nézetet. Az Order táblán belül az sqlrelationship segítségével deklaráljuk a relációt a két tábla között:

```
<?xml version="1.0" ?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes"
xmlns:sql="urn:schemas-microsoft-com:xml-sql">

<ElementType name="Order"
sql:relation="Orders"
sql:key-fields="OrderID">
<AttributeType name="OrderID"
dt:type="id" sql:id-prefix="Ord-"
/>
<AttributeType name="CustomerID"
/>
<attribute type="OrderID" />
<attribute type="CustomerID" />

<element type="OrderDetail">
<sql:relationship
key-relation="Orders"
```

```

        key="OrderID"
        foreign-relation=" [Order
Details]"
        foreign-key="OrderID" />
    </element>
</ElementType>

<ElementType name="OrderDetail"
sql:relation=" [Order Details]"
sql:key-fields="OrderID ProductID">
<AttributeType name="ProductID"
dt:type="idref"
sql:id-prefix="Prod-" />
<AttributeType name="UnitPrice" />
...
</ElementType>
</Schema>
    
```

URL és HTTP elérés

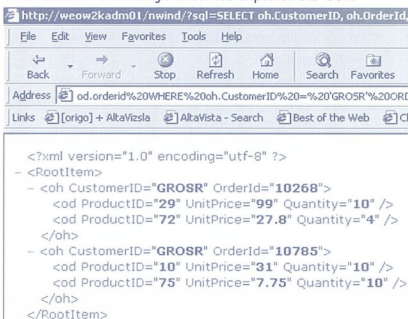
Az SQL Server 2000 XML támogatásának egyik fontos eleme a HTTP protokollon keresztüli adatelérés biztosítása. Ehhez az SQL Server együttműködik az Internet Information Server-rel. Ekkor egy ISAPI alkalmazás biztosítja a kapcsolatot az SQL Server és a webszerver között. Segítségével virtual root-okat lehet létrehozni a webszerveren, melyeket a Configure SQL XML Support in IIS nevű adminisztrációs programmal kezelhetünk. *(Természetesen programokból is lehet konfigurálni, amihez megfelelő adminisztrációs objektumkönyvtárak adnak segítséget.)*

A HTTP lekérések használatához létre kell hozni egy virtual root-ot a lekérdezendő adatbázisra. Ennek részletes leírása a szerverhez mellékelt Books Online-ban található meg. A „Creating the nwind Virtual Directory” című cikk a Northwind adatbázisra mutatja meg a szükséges lépéseket. A példák az ott leírt beállításokra alapoznak. Nézzük meg, hogy a korábban használt lekérézésünket milyen URL-el fogalmazhatjuk meg:

```

http://weow2kadm01/nwind/?sql=Az erede-
deti lekérézésünk FOR XML
AUTO&root=RootItem
    
```

A lekérézés eredménye Internet Explorer 5.5-ben:



A root=RootItem paraméter az URL végén arra utasítja a kérés feldolgozó ISAPI alkalmazást, hogy egy RootItem nevű csomópontot helyezzen el a dokumentum legelső szintjére. Mint korábban említettük, ez azért szükséges, mert alapértelmezésben az IE csak olyan XML adatokat tud megjeleníteni, amelyeknek legfelső szintjén csak egy csomópont található. Ennek az adatelérési módszernek URL Query a neve. Tesztelési célokra nagyon jól használható, azonban produkciós környezetben nem szabad engedélyezni, mert a teljes adatbázist elérhetővé teszi, azaz nem biztonságos. Az adatbázist közvetlenül lekérdező SQL parancsokat XML template-ek segítségével rejthetjük el a külvilágtól. Természetesen gyakorlati haszna csak akkor van egy ilyen template-nek, ha azt paraméterezni is lehet. Nézzünk egy egyszerű példasablont a már jól megszokott lekérézésünkhöz. A példa kilistázta a paraméterként megadott vásárló által megrendelt termékeket:

```

<?xml version='1.0' encoding='UTF-8'?>
<ROOT xmlns:sql='urn:schemas-microsoft-
com:xml-sql'>
<sql:header>
  <sql:param
name='CustName'></sql:param>
</sql:header>
<sql:query>
  A kiinduló példa lekérézésünk
  FOR XML AUTO
</sql:query>
</ROOT>
    
```

Ezt az XML template fájlt mentjük el a Northwind adatbázishoz létrehozott template könyvtárba orders.xml néven, így máris futtatható az alábbi URL:

```

http://weow2kadm01/nwind/templates/ord-
ers.xml?CustName=GROSR
    
```



A sablon által legenerált XML kimenet lényegében azonos az előző példával.

Az utolsó, HTTP-t használó elérési mód az XPath Query, melynek alapjait a korábban említett XML nézetek adják. Ott említettük, hogy XML nézet segítségével (*XDR Annotated Schema*) egy SQL adatforrásból XML formátumú adatokat nyerhetünk ki. Az XPath lekérdezések a forrásadatokat egy XML nézetből nyerik.

Az XPath lekérdezést kétféle módon lehet végrehajtani: közvetlenül, az XML nézetre mutató URL segítségével, vagy template fájlon keresztül. Az első esetben az XML nézetet be kell másolni a schema-ként kijelölt virtuális könyvtárba és onnan futtatni (*lásd a fejezet elején a virtual root létrehozása és konfigurálása rész*). A második esetben létre kell hozni egy template fájlt, ami meghívja az XML nézetet a megfelelő paraméterekkel. Ehhez a template-et a virtual root konfigurálásakor templates-ként kijelölt virtuális könyvtárba kell másolni. Az utóbbira mutatunk egy példát, mivel a template fájlok a gyakorlatban sokkal praktikusabbak a közvetlen URL-es elérésnél, hiszen egységbe zárják és korlátozzák az alkalmazható paramétereket. Az alább látható template fájl (*ordersxpath.xml*) az XML nézeteknél példaként használt orderschema.xml XDR sémát használja:

```
<?xml version='1.0' encoding='UTF-8'?>
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-sql">
<sql:header>
  <sql:param
name='CustName'></sql:param>
</sql:header>
<sql:xpath-query mapping-schema="
orderschema.xml">
  Order[@CustomerID=$CustName]
</sql:xpath-query>
</ROOT>
```

Ebben a template-ben a tulajdonképpeni XPath Query az „Order[@CustomerID=\$CustName]” sor. Ha a template-et paraméterezve meghívjuk:

<http://weow2kadm01/nwind/templates/ordersxpath.xml?CustName=GROSS>

Akkor a következő kimenetet kapjuk:

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-sql">
  <Order OrderID="Ord-10268"
    CustomerID="GROSS">
    <OrderDetail ProductID="Prod-29"
      UnitPrice="99" Quantity="10" />
    <OrderDetail ProductID="Prod-72"
      UnitPrice="27.8" Quantity="4" />
  </Order>
  <Order OrderID="Ord-10785"
    CustomerID="GROSS">
    ..
  </Order>
</ROOT>
```

Érdemes összevetni a schema XML leírást és a kapott eredményt, különös tekintettel arra, hogy a Prod és az Ord szócska hogyan kerül bele a legenerált XML kódba.

Zárásó

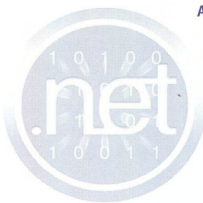
A termékek ismerkedőknek a fenti információk is kicsit töménynek tűnhetnek, pedig sok olyan hasznos dolog van még, amit terjedelmi okok miatt nem érinthetünk: adatok lekérdezése html oldalakról ASP vagy CGI nélkül, XML dokumentumok megformázása XSL segítségével az egyedi igényeknek megfelelően, adatbázisban tárolt képek elérése, a böngészőbe letöltött XML adatok off-line szerkesztése, majd visszatöltése a szerverre, stb. Ezer alkalmazási lehetőség, és egy a válasz: XML. A technológia adott, csak élni kell vele!

Soczó Zsolt, ZsSoczó@protomix.hu



TECH.NET

Az SQL Server 2000 újdonságai



Az SQL Server 2000 újdonságai adatbázis-fejlesztőknek

Az SQL Server 2000 sok olyan újdonságot tartalmaz a fejlesztőknek, amelyeket azok már régóta hiányoltak az SQL Server, illetve a Transact SQL kínálatából. Ebben a cikkben ezeket igyekezzünk részletesen bemutatni, a legfontosabb újításokat, ahol lehet, a technológiai hátteret is teljes mélységében tárgyalva:

- ↳ Új adattípusok
- ↳ Felhasználói függvények
- ↳ „Kaszkádtolt” integritás kritériumok
- ↳ Változások a triggerek használatában

A fejlesztők érdeklődésének középpontjában áll még az elosztott particionált nézetek kezelése éppen úgy, mint az SQL Server 2000 XML támogatási képességei. Ez utóbbi témákat külön-külön is olyan fontosnak gondoltuk, hogy külön cikkek keretében tárgyaljuk őket.

Új adattípusok

Új adatbáziskezelővel együtt jár új adattípusok megjelenése is. Bár az SQL Server 2000 mindössze három új adattípust vezet be, ezek segítségével azonban egyszerűbbé válik robusztus alkalmazásokról megírása.

Az első új típusa a *bigint*, amely – mint azt neve is jelzi – a jelenlegi *int* típus „nagytestvére”, 64 bites (8 bájtos) egész számok tárolására alkalmas. A -2^{63} -tól 2^{63} -ig terjedő számtartomány a jelenlegi *int* típus mellett ún. identitási mezőként is használható, jelentősen megnövelve annak értékészletét. A *bigint* típus mindenhol használható, ahol eddig *int* típust használtunk, de fontos tudni, hogy egy függvény vagy kifejezés értéke csak akkor lesz *bigint*, ha azt explicit módon jelezzük. Az SQL Server önállóan nem konvertálja a többi egész adattípust (*tinyint*, *smallint*, *int*) *bigint* típusúvá.

A *bigint* típus könnyebb kezelése érdekében két új beépített függvény is megjelent az SQL 2000-ben: a *BIG_COUNT* az eddigi *COUNT* aggregátum függvény, a *ROWCOUNT_BIG* pedig az eddigi *@@ROWCOUNT* globális változó „nagytestvére”, ugyanazzal a szemantikai jelentéssel. A „rég” *COUNT* és *@@ROWCOUNT* ugyanúgy használható, mint a korábbi változatokban (*int* értéket adnak vissza), míg a két új függvény *bigint* értéket.

A második új adattípus az *sql_variant*. Ez a Visual Basic „Variant” adattípusához hasonlóan több adattípust tud dinamikusan magába ágyazni, lehetővé teszi a ténylegesen tárolt típusok futásidőben történő lekérdezését. A *sql_variant* a *text*, *ntext*, *image*, *timestamp* és *sql_variant* típusokon kívül minden mászt képes kezelni. A tárolás során a tárolt érték mellett annak adattípusát, maximális méretét, skálázását és pontosságát is képes tárolni, szövegek esetén akár a rendezési jellemzőket is (erről a cikk további részeiben még lesz szó).

Az *sql_variant* típus használata előnyös lehet

olyan alkalmazásoknál, amelyek saját maguk végzik az adattípusok futásidejű kezelését. Például egy olyan ASP oldal, amely felhasználói információkat gyűjt, minden begyűjtött információt letárolhat egy *sql_variant* mezőbe, még a pontos típusellenőrzés előtt. Az *sql_variant* típus használattal azonban kellően körültekintően kell végezni. Ebben segít a *CAST* operátor (*hívhatjuk függvénynek is*), amely segítségével explicit típusúvá tudjuk tenni az *sql_variant*-ban tárolt értéket:

DECLARE

```
@samplevar sql_variant,  
@charvar varchar(20),  
@decvar decimal(10,5)
```

```
SET @samplevar = 12.3456
```

```
SET @charvar = CAST(@samplevar AS  
varchar(20))
```

```
SET @decvar = CAST(@samplevar,  
decimal(10,5))
```

Az *sql_variant* adattípus támogatására az SQL Server 2000-ben bevezették az *SQL_VARIANT_PROPERTY* függvényt, amely célja, hogy típusjellemzőket adjon vissza egy *sql_variant* típusú változó aktuális értékéről. A függvény formátuma az alábbi:

SQL_VARIANT_PROPERTY(változó, tulajdonság)

A *tulajdonság* paraméter a „BaseType”, „Precision”, „Scale”, „TotalBytes”, „MaxLength” és „Collation” értékeket veheti fel, és az *sql_variant* névnek megfelelő jellemzőjét adja vissza. Például az alábbi programrészlet az itt látható eredményt adja vissza:

DECLARE

```
@samplestr sql_variant,  
@sampledec sql_variant
```

```
SET @samplestr = '1.234'
```

```
SET @sampledec = 1.234
```

SELECT

```
SQL_VARIANT_PROPERTY(@samplestr, VarBaseType')
```

```
AS VarBaseType,
```

```
SQL_VARIANT_PROPERTY(@samplestr, VarTotalBytes')
```

```
AS VarTotalBytes
```

UNION

SELECT

```
SQL_VARIANT_PROPERTY(@sampledec, 'VarBaseType'),
```

```
SQL_VARIANT_PROPERTY(@sampledec, 'VarTotalBytes')
```

```
VarBaseType VarTotalBytes
```

```
varchar 13
```

```
decimal 5
```



A harmadik új típus a *table*, amely mint neve is jelzi, táblák tartalmának ideiglenes tárolására szolgál. Természetesen, az új típust az előző kettővel ellentétben nem használhatjuk egy adatbázistábla mezőjeként. A table típus az eddig használt, tempdb adatbázisban tárolódó ideiglenes táblák helyett használható. Mivel ez a típus a memóriában kezelődik, nem tárolódik a merevlemezben, lényegesen jobb teljesítményt tesz lehetővé, mint a korábbi SQL Server változatok ideiglenes táblái.

Felhasználói függvények

Az hiszem, hozzáám hasonlóan sok fejlesztő várta már „SQL verziók óta”, hogy végre tudjunk felhasználói függvényeket definiálni. Ezt a vágyukat szerény mértékben a tárolt eljárásokkal is kielégíthettük, de minden helyen használható függvények sokáig csak álomnak látszottak. Nos, az SQL 2000-ben megvalósított felhasználói függvények korábbi vágyainknál is messzebb elmennek.

Az SQL 2000 felhasználói függvényei a tárolt eljárásokhoz hasonló módon hajthatók végre, illetve olyan módon, akár a többi beépített függvény. Akárcsak a tárolt eljárások lehetnek paraméter nélküliek, de akár több paraméterük is lehet. Visszatérési értékük a tárolt eljárásoktól eltérően nem csak int lehet, hanem bármilyen skaláris adat, vagy akár *sql_variant* vagy *table*!

A felhasználói függvényeket a tárolt eljárásokhoz hasonlóan definiálhatjuk. Fontos különbség, hogy egy felhasználói függvénynek a RETURNS kulcsszóval definiálnia kell a visszatérési érték típusát és egy annak megfelelő típusú értéket vissza kell adnia a RETURN utasítással.

A table típus használatában komoly lehetőségek rejlenek: egy felhasználói függvény megteheti, hogy belsejében definiál egy table típusú változót, azt feltölti értékekkel, majd a változó értékével visszatér. Az ilyen függvényeket „rowset” függvényeknek nevezzük (*amíg „meg nem magyarul” én ezt az elnevezést fogom használni*). Hab a torta tetején, hogy a rowset függvényeket akár egy SELECT utasítás FROM klauzájában is használhatjuk (*s így akár nézetet is definiálhatunk vele!*).

Nézzünk néhány példát!

Az alábbi felhasználói függvény a négyzetre emelését valósítja meg:

```
CREATE FUNCTION dbo.Square (@x int)
RETURNS bigint AS
BEGIN
    RETURN CAST(@x*@x AS bigint)
END
```

A következő függvény table típusú változó segítségével egy táblát ad vissza:

```
CREATE FUNCTION dbo.ProductCat
(@CategoryID int)
RETURNS TABLE
AS
RETURN
```

```
SELECT ProductID, ProductName
FROM Products
WHERE CategoryID = @CategoryID
```

Természetesen a table típusú változó használata nem korlátozódik egyszerű SELECT utasításokra, akár több lépésben is előállíthatjuk a kívánt eredményt. Az így definiált függvényt felhasználhatjuk egy SELECT utasításban:

```
SELECT SUM(od.Quantity) FROM
dbo.[Order Details] od INNER JOIN
dbo.ProductCat(5) pr
ON od.ProductID = pr.ProductID
```

Kaskádolt integritás kritériumok

Az SQL Server már régóta támogatja a integritás kritériumok használatát, amelyekkel táblák közötti logikai kapcsolatot írhatunk le idegen kulcsok és elsődleges kulcsok közötti összefüggések definiálásával. Ha egy táblára vonatkozó módosító műveletnél ezt az integritáskritériumot meg próbáljuk sérteni (*például egy olyan rekordot törölni, amely elsődleges kulcsára más rekordok hivatkoznak*), az SQL Server ezt nem engedi.

A Microsoft Access már régóta támogatja az ún. kaskád UPDATE és DELETE műveletet a táblák közötti relációk konzisztenciájának fenntartására.

Az SQL 2000-ben újdonság a kaskád műveletek támogatása, amelyet az ANSI definíciónak megfelelően a FOREIGN KEY integritáskritériumok létrehozásánál írhatunk le. Vizsgáljuk meg a következő definíciókat, amelyeket a NorthWind adatbázisból ollóztunk ki:

```
ALTER TABLE dbo.[Order Details] ADD
CONSTRAINT FK_Order_Details_Orders FOREIGN KEY
(OrderID) REFERENCES dbo.Orders(OrderID)
ON UPDATE NO ACTION
ON DELETE CASCADE,
CONSTRAINT FK_Order_Details_Products FOREIGN KEY
(ProductID) REFERENCES dbo.Products(ProductID)
ON UPDATE CASCADE
ON DELETE NO ACTION
```

Ez a definíció két integritáskritériumot definiál.

Az első a rendelés [Order] és rendeléstételek [Order Details] között definiálja a kapcsolatot: egy rendeléshez 0, 1 vagy több tétel tartozhat. Az „ON UPDATE NO ACTION” az eddigi viselkedésnek felel meg: nem engedi meg egy rendelés azonosítójának átirását, amíg annak léteznek tételei. Az „ON DELETE CASCADE” új az SQL 2000-ben, jelentése: ha egy rendelést törölünk, akkor azzal egyidejűleg a hozzátartozó összes tétel is automatikusan törölrésre kerül.

A második definíció a rendeléstételek [Order Details] és a cikkek [Products] között definiálja a kapcsolatot: egy cikkre 0, 1 vagy több rendeléstétel hivatkozhat. Az „ON DELETE NO ACTION” az SQL Server ismert viselkedésének felel meg: nem törölhetünk ki addig egyetlen cikket sem az

adatbázisból, amíg az adott cikk szerepel valamelyik rendeltetésben. Az „ON UPDATE CASCADE” jelentése: ha egy cikk azonosítóját megváltoztatjuk, azt egyidejűleg automatikusan változtatassuk meg a rendeltéstelekekben is. Az SQL Server korábbi változataiban az ilyen jellegű működést triggerek írásával lehetett megoldani. A triggerek használata logikailag ugyanolyan jó megoldás, de új adatbázisoknál előnyösebb a „CASCADE” kritériumok használata, ugyanis ezeket sokkal egyszerűbb leírni, s már a táblák definíciójából (*szkripjből*) látszik a fejlesztők által definiált viselkedés. Legálább ilyen fontos, hogy a „CASCADE” kritériumok sokkal jobb teljesítménnyel futnak, mint a triggeres megoldások.

Változások a triggerek használatában

Amióta csak léteznek triggerek a relációs adatbázis-kezelőkben, a programozók előszeretettel használják azokat. Segítségükkel a kiszolgálóoldalon lehet olyan feladatokat megoldani (*s ráadásul akár utólag!*), amelyek miatt korábban a ügyféloldali programokat kellett módosítani. Ilyen feladatok például: táblák adatváltózáseinak naplózása, készletek kifogyása esetén utánrendelés elindítása, az adatbázis logikai-szemantikai egységét sértő műveletek kiszűrése, stb. Az SQL Server 7-es változata a korábbiakhoz hasonlóan támogatta triggerek definiálását az INSERT, UPDATE és DELETE műveletekre. Egy trigger akár több művelethez is csatolható, sőt egy művelethez több triggerrel is lehet csatolni. Maga az SQL Server is saját triggereket hoz létre például az adatreplikációs feladatok ellátására. Egy trigger működése az alábbi módon néz ki:

Az adatmódosító művelet tranzakció indítását váltja ki, majd a tranzakció életben tartása mellett az adatmódosító művelet lefut. Ha eközben hiba lép fel, a tranzakció – és ezzel együtt az adatmódosítás hatása is – visszagördül. Ha az adatmódosítás sikeresen befejeződött és létezik olyan trigger, amely a módosított tábla megfelelő adatmódosító tevékenységére vonatkozik, az adott trigger elindul, és a művelet elején indított tranzakció keretében végrehajtásra kerül. Sikereség esetén a tranzakció lezárul – és ezzel a módosítások végérvényesen megtörténnék. Sikertelenség (*hiba*) esetén az egész tranzakció visszagördül.

Ezt a képet tovább bonyolítja, hogy több trigger is létezhet az adott tábla adott műveletéhez. Ekkor ezek végrehajtási sorrendje indeterminisztikus, vagyis egyik trigger sem használhatja ki, hogy hányadikként fut. Valójában a triggerek a fizikai létrehozás sorrendjében futnak, de ennek a sorrendnek a kiderítése nem egyszerű feladat (*mincs például definiált katalóguseljárás erre*).

A triggerek által módosított táblákon további triggerek létezhetnek. Ha az adatbázis „recursive triggers” opciója be van kapcsolva, a triggerekben módosított táblák triggerrei is elindulnak, ellenkező esetben nem.

Nézünk meg, milyen változásokot hoz a fentiekhez képest az SQL 2000. Minden, amit a fentiekben leírtunk, továbbra is hasonlóan működik az SQL Server 2000 alatt, azonban két kiegészítés jelentősen könnyíthet mindennapi életünkön a fenti „triggerdeszungenben”:

Először is, lehetőségünk van az triggerek futási sorrendjének definiálására az `sp_settriggerorder` tárolt eljárás segítségével. Másodszor, a jelenlegi triggerstruktúra mellett lehetőségünk van ún. INSTEAD OF triggerek definiálására. Fent már említettük, hogy a triggerek csak az adatmódosító művelet végrehajtása után indulnak el – feltéve, hogy az adatmódosító művelet sikeres volt. Az SQL Server 2000 lehetővé teszi, hogy magát az adatmódosító műveletet is teljes egészében átdefiniálhassuk az ún. INSTEAD OF triggerek segítségével. Ha például az Order táblának létezik az INSERT műveletére INSTEAD OF trigger, az INSERT utasítás esetén maga a trigger kerül végrehajtásra. Az SQL Server 2000 terminológiájában a korábbi változatok triggerreit AFTER triggernek nevezi, utalva arra, hogy azok csak az adatmódosító műveletek végrehajtása után indulnak el.

Mire lehetnek jók az INSTEAD OF triggerek? Csak néhány lehetőséget sorolok fel, amelyekből az olvasó ötleteket kaphat:

- ☞ Ha egy INSERT műveletre INSTEAD OF triggeret használunk, lehetőség van a táblában már létező rekordok előzetes törlésére, vagy helyben fellírására, elkerülve az elsődleges kulcsok ütközését a többszörös beszúráskor.

- ☞ Egy DELETE művelet INSTEAD OF triggerrel a törölt rekordot automatikusan átmozgathatja egy archiv táblába. Azok a fejlesztők, akik gyakran használnak nézeteket (*VIEW*), tudják, hogy egy nézet adattartalmának módosítására rengeteg megkötés van, ami a gyakorlatban sokszor azt jelenti, hogy nehéz módosítható nézeteket létrehozni. Az INSTEAD OF triggerek segítségével az eddig lehetetlennek tűnő dolog is megoldható: akár olyan nézeteket is módosíthatunk, amelyek számított mezőt tartalmaznak!

Nézünk meg, hogyan is lehetséges ez. Hozzuk létre az alábbi nézetet:

```
CREATE VIEW dbo.EmployeeName AS
SELECT EmployeeID, FirstName + ' ' +
LastName AS EmployeeName
FROM Employees
```

Milyen jó lenne, ha ezen a nézetten futtatni tudnánk például az alábbi UPDATE utasítást:

```
UPDATE EmployeeName
SET EmployeeName = 'Anne
Dodsforth-Smith'
WHERE EmployeeID = 9
```

Az SQL Server 2000 nemcsak táblákon, hanem nézeteken is képes INSTEAD OF triggereket kezelni. A fenti UPDATE utasítás nem álom, hanem valóság, ha az alábbi INSTEAD OF triggerrel használjuk az EmployeeName nézetet:

```
CREATE TRIGGER
dbo.trg_employee_name_update ON
dbo.EmployeeName
INSTEAD OF UPDATE AS
BEGIN
UPDATE Empolyees SET
```




Microsoft
.net

```

FirstName = SUBSTRING(i.EmployeeName, 1,
  (CHARINDEX(' ', i.EmployeeName) - 1)),
LastName = SUBSTRING(i.EmployeeName,
  (CHARINDEX(' ', i.EmployeeName) + 1),
  DATALength(i.EmployeeName))
FROM Employees INNER JOIN inserted I
ON Employees.EmployeeID = i.EmployeeID
END

```

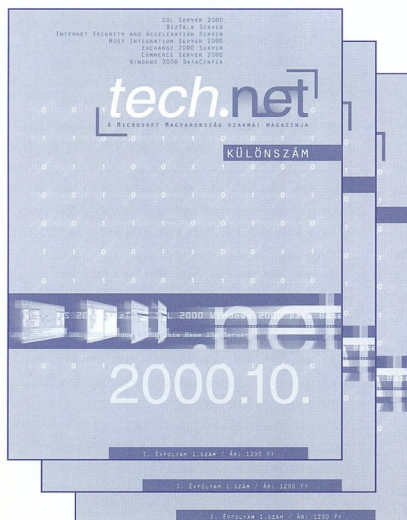
A trigger egyetlen SQL utasításból áll, amely két táblát csatol össze. Az inserted tábla egy „ál-tábla”, amelynek szerkezete pontosan megegyezik az Employee táblával és azokat a rekordokat tartalmazza, amelyeket az adtmódosító művelet során módosítani kell. A két tábla csatolására azért van szükség, hogy az Employee táblában ténylegesen csak az EmployeeName nézetben módosítani kívánt rekordok változásait vezessük vissza. Az UPDATE utasítás az inserted tábla EmployeeName mezőjét vágja szét a képzési szabálynak megfelelően vezeték és keresztnévre.

Összegzés

Az SQL 2000 sok újdonságot tartalmaz az adatbázis fejlesztők számára, amelyekkel nem csak megismerkedni érdemes, hanem használni is azokat. Az új adattípusok közül a *bigint* és az *sql_variant* egyszerűbbé teszi a Web-es alkalmazások készítését. A *table* típus és a *felhasználói függvények* együttes használata a korábbiaknál egyszerűbben leírható, olvasható kódot eredményezhet. A kaszkádolt integritás kritériumok egyszerűsíthetik életünket, a kaszkád módosításokat elvégző triggerek megírásától szabadítanak meg bennünket, s ráadásul a módosítások teljesítményét is növelik.

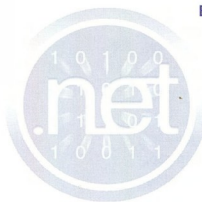
Az INSTEAD OF triggerok növelik a procedurális integritás implementációjának rugalmasságát azért, hogy „átdefiniálhatjuk” a táblákhoz és nézetekhez tartozó adtmódosító műveleteket. Segítségükkel olyan nézeteken is értelmezhetjük az adtmódosítást, amelyeken az SQL Server korábbi változatainál ez nem volt lehetséges.

Novák István, inovak@protomix.hu



Lapunk előfizethető!

Ha szeretné, hogy a **tech.net** legfrissebb példánya minden hónapban biztosan eljusson Önhöz, fizessen elő! Érdeklődjön az előfizetési lehetőségről a <http://tech.net.netacademia.net> oldalon.



Elosztott particionált nézetek az SQL Server 2000-ben

A Microsoft és az Oracle között éles verseny folyik az adatbázis sebességi rekordok javításában. A Microsoft ehhez nagy mértékben felhasználta az oldalirányú skálázás (*scaling out*) technológiáját, amely elosztott particionált nézeteket alkalmaz. Ez a cikk egy példán keresztül bemutatja az elosztott particionált nézetek:

- ☞ fogalmát
- ☞ definiálásának módját
- ☞ használatát

A cikk érinti az indexelt nézetek témakörét is. Ez ugyancsak a nézetekhez kapcsolódik, mint a lekérdezések sebesség-növelésének másik eszköze.

Mi az elosztott particionált nézet?

Azokban az alkalmazásokban, amelyek nagy rekordszámú táblákon végeznek műveleteket, az adatbázis növekedésével megnőhet az adatkezelő műveletek válaszüzeje. Ilyenkor az adatbázis adminisztrátora megpróbálhatja a válaszidő csökkentését a szerver „feljavításával” elérni (*Például CPU, memória hozzáadása*). Ezt nevezik „felfelé skálázásnak” (*scaling up*). Ennek lehetőségei azonban egy határ után kimerülnek. A teljesítmény javításának további módja lehet az „oldalirányú” skálázás (*scaling out*).

Ehhez az elosztott particionált nézetekben a nagy táblákat vízszintesen feldaraboljuk kisebb táblákra, vagyis úgy, hogy a kisebb táblák szerkezete megegyezik az eredeti tábla szerkezetével, és a rekordokat osztjuk szét a kisebb táblákba valamilyen kritérium szerint. Ezeket a táblákat külön szerverekre helyezük el úgy, hogy biztosítjuk a szerverek együttműködését (*federated cluster*).

Az összes szerverről el tudjuk érni az összes adatot olyan módon, hogy minden szerveren ugyanazon a néven definiálunk egy nézetet. Ezáltal elfedjük azt a tényt, hogy az adatok több szerveren vannak elosztva. A felhasználó illetve alkalmazás bármelyik szerverre csatlakozva ugyanúgy kiadhatja az adatkezelő utasításokat (*SELECT, INSERT, UPDATE, DELETE*) a nézetre, mintha egy táblával dolgozna. A nézet a táblák rekordjait UNION operátorral egyesíti. A particionált nézetek ilyen definiálására az SQL Server előző verzióiban is volt lehetőség, azonban az ilyen nézeteken keresztül nem lehetett UPDATE műveletet végrehajtani. Az SQL Server 2000 megszünteti ezt a korlátot, és a particionált nézetek feldolgozására új optimalizációs technikákat is alkalmaz.

A elosztott particionált nézetek létrehozása három lépésből áll:

- ☞ particionált táblák definiálása
- ☞ kapcsolt szerverek létrehozása
- ☞ particionált nézetek definiálása

Első lépés: Particionált táblák definiálása

Hozzuk létre a TestDB adatbázist a szervereken. Mindegyik szerveren ugyanazt a táblastruktúrát definiáljuk. A táblák csak abban fognak különbözni, hogy milyen kritérium szerint osztjuk el közöttük a rekordokat. A tábla tartalmaz egy oszlopot, a particionáló oszlopot, amelynek értéke meghatároz-

za, melyik rekord melyik táblába kerül. A feltételeknek kölcsönösen kizáróknak kell lenniük, hogy általuk minden rekord egy és csak egy táblához tartozzon. A leghatékonyabb particionáló feltétel olyan, amelyik a rekordokat a lehető legegyszerűbben osztja el, és amennyiben lehetséges, valószínűsíti, hogy egy adott szerverhez csatlakozó felhasználók a lokális tábla rekordjait akarják elérni. A rekordokat szétoszthatjuk például egy karakteres azonosító kezdőbetűje, egy numerikus azonosító intervallumai, földrajzi körzet vagy egyéb csoportosítás szerint. A feltételt minden táblában a particionáló oszlophoz tartozó CHECK constraint biztosítja.

Példánkban az SQL minta, a NorthWind adatbázis Customers tábláját particionáljuk. A particionáló oszlop a CustomerID oszlop. Az 1. szerveren levő CustomersAM tábla azokat a rekordokat tartalmazza, ahol a vevő kódjának kezdőbetűje A és M közé esik.

```
CREATE TABLE CustomersAM (
    CustomerID nchar(5) NOT NULL,
    CompanyName nvarchar(40) NOT NULL,
    ContactName nvarchar(30) NULL,
    ContactTitle nvarchar(30) NULL,
    Address nvarchar(60) NULL,
    City nvarchar(15) NULL,
    Region nvarchar(15) NULL,
    PostalCode nvarchar(10) NULL,
    Country nvarchar(15) NULL,
    Phone nvarchar(24) NULL,
    Fax nvarchar(24) NULL,
    CONSTRAINT PK_CustomersAM PRIMARY KEY
    CLUSTERED (CustomerID),
    CONSTRAINT CHK_CustomersAM CHECK
    (CustomerID BETWEEN 'A ' AND 'MZZZZ')
```

A 2. szerveren hasonlóan definiáljuk a CustomersNZ táblát, úgy, hogy az az N és Z közé eső vevőkódokat tartalmazza:

```
CREATE TABLE CustomersNZ (
    CustomerID nchar(5) NOT NULL,
    CompanyName nvarchar(40) NOT NULL,
    ContactName nvarchar(30) NULL,
    ContactTitle nvarchar(30) NULL,
    Address nvarchar(60) NULL,
    City nvarchar(15) NULL,
    Region nvarchar(15) NULL,
    PostalCode nvarchar(10) NULL,
    Country nvarchar(15) NULL,
    Phone nvarchar(24) NULL,
    Fax nvarchar(24) NULL,
    CONSTRAINT PK_CustomersAM PRIMARY KEY
    CLUSTERED (CustomerID),
    CONSTRAINT CHK_CustomersNZ CHECK
    (CustomerID BETWEEN 'N ' AND 'ZZZZZ')
```



A táblának szigorú feltételeket kell teljesíteni, hogy a nézetre később lehessen alkalmazni az UPDATE utasítást:

- ☞ Az elsődleges kulcsnak tartalmaznia kell a partícionáló oszlopot.
- ☞ A partícionáló oszlopra megadott CHECK constraint csak az alábbi operátorokat tartalmazhatja: BETWEEN, AND, OR, <, <=, >, >=, =
- ☞ A partícionáló oszlopra csak egy constraint lehet definiálva.
- ☞ A partícionáló oszlop nem lehet számított oszlop.
- ☞ A táblának nem lehet IDENTITY illetve TimeStamp mezője.

A fenti feltételek azt is jelentik, hogy a tábla kulcsa nem lehet a gyakran használt, IDENTITY mezőként definiált generált kulcs. Ha nem akarunk természetes kulcsot használni, a kulcsot generálhatjuk tárolt eljárással.

Második lépés: Kapcsolt szerverek létrehozása

Mivel az elosztott táblák több szerveren elhelyezkednek el, mindegyik szervernek el kell érnie az összes többit. Mindegyik szerveren az összes többit kapcsolt szerverként kell konfigurálni. Ezt meg lehet tenni az Enterprise Manager segítségével vagy az sp_addlinkedserver tárolt eljárás használatával. Például az 1. szerveren a 2. szerver elérését szolgáló kapcsolatot így definiálhatjuk:

```
sp_addlinkedserver
    @server = 'KAPCSOLAT2',
    @srvproduct = '',
    @provider = 'SQLEADB',
    @datasrc = 'SERVER2\SQLSERVER1'
EXEC sp_serveroption 'KAPCSOLAT2',
    'lazy schema validation', 'true'
```

A @server paraméter azt a logikai nevet adja meg, ahogy a kapcsolatra hivatkozni akarunk. A @datasrc paraméter adja meg annak a szervernek a nevét, amelyet el akarunk érni. A példában megadott szervernév azt a lehetőséget demonstrálja, hogy az SQL Server 2000 biztosítja egy szerveren több SQL szerver példány definiálását is. Ilyenkor a szerver nevét és az SQL szerver példány nevét „\” választja el. Tehát elosztott partícionált nézetek definiálását akár egy szerveren is kipróbálhatjuk, ha több SQL szerver példányt telepítünk rá. A „lazy schema validation” opció megadásával optimalizálhatjuk a teljesítményt azáltal, hogy Query Processor a kapcsolt táblák metaadatait csak akkor fogja lekérni, amikor azokra ténylegesen szükség lesz.

Harmadik lépés: Partícionált nézetek definiálása

Most már csak össze kell rakni a táblákat egy nézetben. Mindegyik szerveren definiáljunk egy nézetet, ugyanazzal a névvel. Mindegyik nézetben egy lokális és egy távoli tábla szerepel, az UNION ALL operátorral összekapcsolva. A nézet definíciója az 1. szerveren az alábbi:

```
CREATE VIEW Customers
AS
SELECT * FROM CustomersAM
UNION ALL
SELECT * FROM
KAPCSOLAT2.TestDB.dbo.CustomersNZ
```

A 2. szerveren pedig a következő:

```
CREATE VIEW Customers
AS
SELECT * FROM
KAPCSOLAT1.TestDB.dbo.CustomersAM
UNION ALL
SELECT * FROM CustomersNZ
```

A lokális táblákra elég csak a tábla nevével hivatkozni. A távoli táblára a négyrészes táblanév-hivatkozást kell használni, amelynek formája:

Kapcsolat.Adatbázis.Tulajdonos.Táblanév

A nézetekre a táblákhoz hasonlóan szigorú feltételeknek kell teljesülnie ahhoz, hogy a nézetre alkalmazható legyen az UPDATE utasítás:

- ☞ A nézet minden táblára és oszlopra csak egyszer hivatkozhat.
- ☞ Minden SELECT listának az összes benne levő tábla elsődleges kulcsának összes mezőjét tartalmaznia kell.
- ☞ Az összes nézet SELECT listájának ugyanazon a helyen teljesen azonos típusú mezőt kell tartalmaznia, és a partícionáló oszlopnak is ugyanott kell lennie.
- ☞ Ha egy mező szerepel az alaptáblában, de nem szerepel a nézetben, akkor meg kell engednie a NULL értéket.

A elosztott partícionált nézetek használata

Az elosztott partícionált nézetekre használhatjuk a DELETE, UPDATE, INSERT utasításokat. Esetünkben például töltjük fel a létrehozott táblákat a NorthWind adatbázis Customers táblájából. Az XACT_ABORT értéket azért kell ON-ra állítani, hogy futási hiba esetén a teljes tranzakció visszagörgetése (rollback) megtörténjen.

```
SET XACT_ABORT ON
GO
INSERT INTO Customers
SELECT * FROM
Northwind.dbo.Customers
```

Gondoskodjunk arról, hogy a Microsoft Distributed Transaction Coordinator (MSDTC) fusson, mivel az adatműveletek elosztott tranzakcióit fognak eredményezni. Az SQL Server gondoskodik arról, hogy a különböző CHECK constraint-eknek megfelelően a rekordokat elossa a táblák között.

Az elosztott táblákra nem feltétlenül kellene elosztott partícionált nézetet definiálni. Az UPDATE művelet végrehajthatóságát az SQL Server 2000-ben INSTEAD OF triggerekkel is lehet biztosítani. A elosztott partícionált nézetek előnye azonban nemcsak abban mutatkozik meg, hogy módosítani lehet rajtuk keresztül az adatokat. A Query Processor olyan intelligens, dinamikus végrehajtási tervet készít, amelyik optimalizálja az adatok kiolvasását az adatbázisból. A távoli táblák elérése olyan módon történik, hogy csak azokat a rekordokat kelljen kiolvasni, amelyek feltétlenül szükségesek. Például ha a partícionáló oszlop szerint keresünk, nem az összes táblában keres, hanem csak abban, amelybe a rekordnak a partícionáló feltételek szerint esnie kell. A vízszintesen partícionált adatok ilyen módú használata jelentős skálázási lehetőségeket jelent. Kezdetben az adatokat elég egy vagy két szerverre telepíteni, és az adatmennyiség növekedésével további szervereket hozzáadni, az adatokat újraosztani a partícionáló feltételek újradefiniálásával.

Javaslatok

Amennyiben több táblát is partícionálunk, és azok gyakran szerepelnek összekapcsolva SELECT kifejezésekben, célszerű a partícionáló feltételeket olyan módon összehangolni, hogy az összekapcsolandó rekordok egy szerveren legyenek. Például ha a vevőket a fenti módon a kód kezdetűje szerint osztjuk szét, akkor a hozzájuk tartozó megrendelések elsődleges kulcsába be kell venni a vevő kódját, és a megrendelésrekordokat a vevőkkel megegyező módon kell szétosztani.

Azokat a táblákat érdemes partícionálni, amelyekre gyakoriak a módosító műveletek. Amennyiben a partícionált táblákhoz gyakran kapcsolódik olyan tábla, amelynek rekordjai csak ritkán változnak, azokat érdemes a szervereken egyformán tárolni, és a másolatok között az egyezőséget például replikációval biztosítani.

Az elosztott partícionált nézetek használatát nem javasoljuk minden esetben, csak kifejezetten nagy terhelés esetén, amely az alkalmazásoknak csak nagyon kis százalékára áll fenn. Eltakarják ugyanis az adatok elosztottságát a rekordmódosító műveletekben, de nem takarják el azt az adatok kezelhetőségében.

Gondoljunk például arra, hogy ha valamelyik szerveren megsérül az adatbázis, és egy előző mentésből kell visszaöltetni az adatokat, akkor a visszatöltés utáni állapotnak összehangoltnak kell lennie. A visszatöltésnek mindegyik szerveren ugyanaddig a logikai pontig kell megtörténnie. Az SQL Server 2000 ezt úgy biztosítja, hogy az adatok visszaöltését is lehet végezni egy megjelölt tranzakcióig. Mindenesetre az adatbázis adminisztrátorok dolga ezáltal bonyolultabbá válik. A partícionálás miatt át kell tervezni az adatbázist, és alkalmazások módosítására is szükség lehet. Először mindenképpen a felfelé skálázás lehetőségeit érdemes kihasználni, mindaddig, amíg ki nem merülnek, illetve túl drágává nem válnak.

Indexelt nézetek

A nézeteket virtuális tábláknak lehet tekinteni. A nézet eredményeképpen előálló adatok nincsenek az adatbázisban tárolva, hanem azokat mindig az alapjukat képező táblák adataiból kell előállítani. Ez sok időt vehet igénybe például nagy összegzések esetén. Az indexelt nézet használatával a nézet eredménye is tárolásra kerül az adatbázisban. Ez jelentősen lerövidítheti a lekérdezés futási idejét. Ne felejtjük el ugyanakkor, hogy a nézet alapját képező valamely tábla módosítása a nézet adatainak azonnali módosítását is eredményezi, aminek következménye az adatmódosító műveletek némi lassulása.

Példánkat vegyük megint a Northwind adatbázisból. Összegezzük a tételeket rendelésenként. Az indexelt nézet definiálása megköveteli néhány opció beállítását. A SCHEMABINDING opció megakadályozza a nézet alapját képező táblák szerkezetének módosítását a nézet újradefiniálása nélkül.

```
SET NUMERIC_ROUNDABORT OFF
GO
SET ANSI_PADDING, ANSI_WARNINGS,
CONCAT_NULL_YIELDS_NULL, ARITHABORT,
QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
CREATE VIEW dbo.OrderSubTotals
WITH SCHEMABINDING
AS
    SELECT OrderID,
           SUM(CONVERT(MONEY,
           (UnitPrice*Quantity* (1-
           Discount)/100))*100) AS SubTotal,
           COUNT_BIG(*) AS DetailCount
    FROM dbo.[Order Details]
    GROUP BY OrderID
GO
```

A csoportosításban szereplő összes oszlopnak szerepelnie kell a SELECT listában, mivel ezek szerepelhetnek majd az indexben. Az index definiálását a következő sor jelenti:

```
CREATE UNIQUE CLUSTERED INDEX
IDX_CV_ORDERSUBTOTAL_OI_ST
ON OrderSubTotals (OrderID)
```

A nézetre elsőként definiált index csak „UNIQUE CLUSTERED” lehet.

Az indexelt nézet definiálása nem csak az adott nézet futtatását érinti. A Query Optimizer más lekérdezések optimalizálásában is felhasználhatja, ha úgy látja jónak.

Kovács Sándor, skovacs@protomix.hu





Kís e-commerce nyelvtan

Az XML ideális adatformátum a különböző médiumokon terjesztett, közvetlenül megjelenített strukturált szöveg számára. Az XML dokumentum speciális utasításokat, jelölő elemeket (*ún. tag-eket*) tartalmaz, amelyek a dokumentum egy azonosítható részét határolják.

A száraz definíció után vegyünk egy egyszerű XML dokumentumot:

```
<?xml version="1.0" ?>
<PEOPLE xmlns="x-schema:PEOPLE.DTD">
  <PERSON>
    <NAME>Mark Wilson</NAME>
    <AGE>34</AGE>
    <TEL>(+612) 12345</TEL>
    <ADDRESS>911 Somewhere Circle, Canberra, Australia</ADDRESS>
    <EMAIL>Mark.Wilson@somewhere.com</EMAIL>
  </PERSON>
</PEOPLE>
```

Aki már látta egy weboldal forrását, annak ismerősnek tűnhet a HTML-re hasonlító szintaktika. Mégis mi a különbség? Az első, amit észrevehetünk, hogy érthetőek a dokumentum által leírt értékek és kapcsolatok (*ez az XML egyik óriási előnye*). Egy Mark Wilson nevű, 34 éves ausztrál embert írunk le, elérhetőségi adataival együtt. Egyetlen „problémás” sor akadhat, a (2) sorszámú. Ez az ún. root tag, ami egy mutató a sémadokumentumra.

Miért van szükség sémára? Ha akár minimális tapasztalattal is rendelkezünk ilyen jelölőelemekkel határolt dokumentumok (*pl. HTML*) készítésében, végigtekintve a dokumentumon érezzük, hogy helyesen formázott. Tudjuk, hogy a <PEOPLE> részben több <PERSON> is szerepelhet, viszont egy PERSON-nak csak egy neve lehet. Megfelelő viszonyítási alap nélkül erre egy gép képtelen. A séma elkészítése éppen a „viszonyítási alap” megteremtése a számítógép számára, azaz a használható értékek és alkalmazható kapcsolatok definiálása. A sémáknak épp ezért van különleges szerepe az e-commerceben, hiszen segítségükkel dönthető el, hogy egy üzenet megfelel-e az előírt szabályoknak. Az előbbi üzenetet leíró séma egy lehetséges megvalósítása az alábbiakban látható.

```
<?xml version="1.0"?>
<Schema name="people.dtd"
  xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="PEOPLE" content="eltOnly">
    <element type="PERSON" minOccurs="1" maxOccurs="*" />
  </ElementType>
  <ElementType name="PERSON" content="eltOnly">
    <element type="NAME" />
    <element type="ADDRESS" />
    <element type="TEL" />
    <element type="AGE" />
    <element type="EMAIL" />
  </ElementType>
  <ElementType name="ADDRESS" dt:type="string" />
  <ElementType name="EMAIL" dt:type="string" />
```

```
<ElementType name="AGE"
  dt:type="int"/>
<ElementType name="NAME"
  dt:type="string"/>
<ElementType name="TEL"
  dt:type="string"/>
</Schema>
```

Az első érdekesség az, hogy egy XML sémát XML-ben írunk le, azaz az emberi nyelvhez hasonlóan rekurzívan találkozzunk.

A 2-4. sorban adjuk meg a séma nevét, ill. a névet. A fent megadott névtér lehetővé teszi a Microsoft beépített adattípusainak és a későbbiekben használt dt:type prefix használatát.

A következő 3 sor alkotja a dokumentum legfelső szintű deklarációs részét. Ebben határozzuk meg, hogy az üzenet milyen elemekből fog állni. Ezek szerint üzenetünk minimum 1 (*minOccurs attribútum*), maximum pedig tetszőleges számú PERSON elemből állhat (*maxOccurs attribútum*).

A legfelső szint után, ami általában egy vagy több absztrakt típus meghatározásából áll, rekurzívan haladunk lefelé az adatok definiálásával, amíg az összes általunk definiált típust elemi, a számítógép által már értelmezhető adattípusra nem bontunk (*8-19. sor*). Például a PERSON elem NAME, ADDRESS, TEL, AGE tag-eket tartalmazhat. Ezek pedig az egész típusú AGE kivételével mind stringek.

A fenti dokumentum és sémája nagyon egyszerű. Természetesen egy „igazi” üzenet ennél sokkal komplexebb is lehet, és lényegesen több lehetőség van a sémák tulajdonságainak kialakítására. Akit komolyabban érdekel a téma, az alábbi forrásokból bőveges információhoz juthat.

Irodalom

Neil Bradley: Az XML-kézikönyv
www.forrasok:

<http://www.biztalk.org/>
<http://msdn.microsoft.com/xml/default.asp>
<http://www.microsoft.com/biztalk/>

Dániel Kolos, kdaniel@icon.hu





Napjaink vállalatai eljutottak arra a szintre, ahol a hagyományos technológiák kínálta elemzési lehetőségek elégtelennek bizonyultak a növekvő verseny mellett kialakult új piaci feltételek mellett. Évről évre növekvő mennyiségű és egyre részletesebb alapadatokkal állunk szemben, így az információ kinyerésének sebessége folyamatosan csökken. Ennek következtében diverzifikálódott az adatbáziskezelők piaca.

Megjelent egy új adatmodell, amely már képes volt a felhasználók (döntéshozók) igényét megfelelő gyorsasággal és könnyű kezelhetőséggel kielégíteni. Ezt az új adatmodellt az OLAP betűszóval illették. Az adatmodell és annak tárolási stratégiája azonban nem támogatja a napi működés közben keletkezett információk tárolását és feldolgozását, így az továbbra is a hagyományos relációs adatbáziskezelők (operatív, működtető rendszerek) feladata maradt, az OLAP-ot pedig ráültették ezekre a rendszerekre. Tehát az OLAP adatbázisok nem helyettesítik, nem is helyettesíthetik a működtető rendszereket, illetve azok feladatait, csak mintegy kiegészítőjükként működhetnek.

Így általánosan elmondható, hogy az OLAP rendszerek a működtető rendszerek adataiból táplálkoznak, periodikusan áttöltve azok adatainak egy részhalmozát.

Az OLAP követelményei:

- Adatbázisa csak az elemzéshez szükséges adatokat tartalmazza.
- Többdimenziós nézet, amely kielégíti az elemzők igényeit.
- Adatbázisa felösszegezt és elemi információkat egyaránt tartalmaz.
- Az adatbázisban tárolt információ lekérdezésének sebessége nagyságrendekkel nagyobb, mint a hagyományos módon tárolt adatbázisok lekérdezésének sebessége.
- Felhasználóbarát, a ügyféloldali lekérdezés összeállítására drag and drop módszerrel történik.

Azokat a rendszereket, amelyek kielégítik a fent vázolt követelményeket, összefoglaló néven OLAP rendszereknek nevezzük. Az OLAP szinonimáiként gyakran használatosak a MIS (Management Information System), ennek magyar megfelelője a VIR vagy Vezetői Információs Rendszer, a DSS (Decision Support Systems), a BI (Business Intelligence), valamint az EIS (Executive Information Systems) kifejezések.

Data Transformation Services (DTS)

Ahhoz, hogy kielégítsük az OLAP követelményeit (pl.: gyorsaság), a működtető rendszerek adatainak azon részét, amelyek az elemzéshez szükségesek, át kell emelnünk az OLAP adatbázisába. Ez azt jelenti, hogy a vállalati adatoknak lesz olyan részhalmozása, amely duplikáltan lesz tárolva, elérhető OLAP-ból illetve az alaprendszerből is. Az SQL Server 2000 a relációs és többdimenziós tárolómotorok mellett tartalmaz egy adatbetöltési funkciót ellátó programot is, amely grafikus programozási felület segítségével tölti be a transzformált forrásadatokat a céladatbázisba.

AZ OLAP adatbázisa

Az adatokcák témaorientáltan tartalmazzák az elemzéshez szükséges adatokat, melyek adatbázissá szerveződnek. (Hasonlóan a relációs adatbázisoknál használt táblához). Bár kockákról beszélünk, melyek 3 dimenziós testek, egy MS OLAP adatokcának akár 128 dimenziója is lehet. A dimenziók segítségével határozhatjuk meg az adatokca egy pontjának helyét és tartalmát, mint ahogy egy n dimenziós tér egy pontjának meghatározásához n darab koordináta szükséges. Ezeket a koordinátákat nevezzük a dimenziók tagjainak vagy elemeinek. Amikor tehát lekérdezzük egy adatokcában tárolt adatot, az adatokca összes dimenziójának meghatározuk egy-egy elemét és az ezen koordináták által meghatározott síkot megjelenítjük a képernyőn. (Vesszük az n dimenziós kocka egy kétdimenziós szeletét.) Tegyük fel, hogy adott egy Idő, Cikk, Vevő dimenziókkal rendelkező 3 dimenziós adatokcánk, amely pontjai árbevétel adatokat tartalmaznak. (kinek, mikor, mit és mennyiért értékesítettünk). Ennek meghatározásához tehát, hogy egy konkrét vevő egy konkrét cikkből egy adott időszak alatt mennyit vásárolt, a vevő, cikk, idő dimenziók egy-egy elemének kijelölésével juthatunk. A dimenziók tagjait hierarchiákba szervezhetjük, azaz a dimenzió elemeit összegző és részletező csoportokba sorolhatjuk. (Pl.: év, negyedév, hónap, nap) Mivel az összegzőelemek is a dimenzió tagjai, ezért ezek kijelölésével a fenti példa alapján meghatározható az árbevétel értékének éves összege is. (Miatán összegzőelemet vettünk fel a dimenzióba, az ahhoz tartozó értéket kiszámítottuk a gyerekek értékének összegeként és letároltuk az adatokca megfelelő pontjaiba) Az OLAP adatbázisa tehát úgy van megszerkesztve, hogy az említett koordinátákhoz tartozó adatokat minél gyorsabban megtalálja.

MDX

Az MDX (Multidimensional Expressions) a MS OLAP adatbázisok lekérdező nyelve, amely egyúttal az OLE DB for OLAP szabvány része. Ezért az MDX más gyártók által forgalmazott OLAP adatbázisok lekérdezésére is használható, ha azok csatlakoztak a szabványhoz. (pl.: TM1, SAS,...) Az MDX lekérdezés kicsit hasonlít az SQL SELECT-hez. Például a

```
SELECT
    Time.members ON COLUMNS,
    Cikk.members ON ROWS
FROM Ertekesites
WHERE Vevo.Total
```

lekérdezés eredményeül az Ertekesites adatokca olyan két-dimenziós szeletét kapjuk, amelynek oszlopán az összes időpont, során a cikk dimenzió összes eleme található. A where feltételrel határozuk meg az adatokca két-dimenziós szeletének kijelöléséhez szükséges lapozódimenziók elemeit (a hiányzó koordinátákat).

Az adatkocka koordinátáira a dimenzióelemek egyedi nevével (*Unique name*) hivatkozunk. Ezt az egyedi nevet az Analysis Services belsőleg határozza meg az adott elem kódja (*vagy megnevezése*) és hierarchiabeli helye szerint.

Adatkockák

Az adatkockák olyan adatbázisobjektumok, amelyek az elemzéshez szükséges adatokat úgy tárolják, hogy azok egy tetszőleges szeletének lekérdezése a leggyorsabban megvalósuljon. Legtöbbször valamilyen OLTP rendszer-elemzéshez szükséges adatait és azok összesítését tartalmazza. Lekérdezésük során a kliens oldalon – rendszerint egérkattintásokkal – jelöljük ki a koordinátákat, amely alapján az adatbáziskiszolgáló visszaadja a lekérdezés eredményeként meghatározott sík adatait. OLAP adatkockák tárolhatóak relációs vagy többdimenziós adatbázisokban egyaránt. Mielőtt tehát létrehozunk egy adatkockát, meg kell határozunk annak tárolási módját.

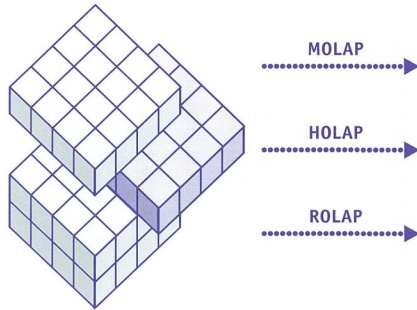
MOLAP, ROLAP, HOLAP...

Az Analysis Services képes az adatkockákat három különböző stratégia szerint tárolni. (Csak többdimenziós, csak relációs és egyik részét a többdimenziós, másik részét a relációs adatbázisokban tárolva.) Arra a kérdésre, hogy mikor, melyik tárolási stratégiát alkalmazzuk, nincs egyértelműen meghatározható szabály, de ökölszabályként használható, hogy ha a lekérdezés sebességének maximalizálása a cél, akkor tároljuk adatainkat a többdimenziós oldalon, ha pedig a tárkapacitás minimalizálása a cél, akkor tároljuk adatainkat a relációs oldalon.

MOLAP	ROLAP	HOLAP
Adatok tárolása többdimenziós adatbázisban	Adatok tárolása relációs oldalon	Összegzett adatok tárolása a többdimenziós, elemi adatok tárolása relációs oldalon
Leggyorsabb lekérdezés	Legkisebb tárkapacitás	A kettő előnyének ötvözése

OLAP technológiák összehasonlítása

A szűk keresztmetszettel függően tehát szabadon választhatunk a tárolási módok között. Sőt, lehetőségünk van akár egy adatkockán belül több tárolási mód választására is. Például 5 évvel ezelőtti ritkán használt adatainkat tárolhatjuk a tárkapacitás szempontjából leghatékonyabb ROLAP partíció, míg az 5 év belüli adatokat a lekérdezés sebességére optimalizált MOLAP partíció. Természetesen a felhasználó nem látja, hogy az adatkocka egyes részei külön partícióban vannak tárolva, legfeljebb a lekérdezés sebességéből sejtetheti, hogy éppen egy ROLAP partíciót kérdez le. A partíciókat sem szükséges egy szerveren tárolni, hanem megoldható, hogy egy központi Analysis kiszolgálóról regisztrálva a partíciókat külön-külön, több gépen helyezzük el.



Distributed Partitioned Cube

Az adatkockák particionálása természetesen megfordítható folyamat, azaz a partíciókat bármikor összevonhatjuk egy adatkockába.

Offline OLAP

Az adatkockák adatbáziskiszolgálón történő tárolása helyett használhatunk úgynevezett kockafájlokat (**.cube*). Ezek a hordozható adatfájlok egy létező adatkockából akár az Excel 2000 segítségével is előállíthatók, melyek szerveren vagy asztali gépeken tárolva is lekérdezhetőek. Így kockafájlok segítségével a vezetők munkahelyüktől távol, a központi gép elérése nélkül elemezhetik a számukra fontos információkat.

Virtuális adatkockák

A virtuális adatkocka több, létező adatkockából előállított logikai adatkocka, amely adatokat nem, csak az adatok címeit tartalmazza. A virtuális adatkockát úgy állítjuk elő, hogy a forrás adatkockák közös részhalmozásából kiválasztjuk a szükséges dimenziókat és azok elemeit. (Hasonlóan a view-hoz.) Miután elkészült a virtuális adatkocka, a felhasználó ugyanúgy használhatja azt, mint egy tárolt adatkockát. Gyakran használunk virtuális adatkockát több kocka adatainak számszerű összehasonlítására, de arra is, hogy bizonyos felhasználók elől elrejtjük a nem rá tartozó adatokat. (Egyetlenegy adatkockából több virtuális kocka is képezhető.)

Dimenziók

A dimenziók olyan minőségi, területi, stb. ismérvek, amelyek leírják az adatkocka adatait. Anyféle szempont szerint vizsgálhatjuk adatainkat, ahány dimenziót az adatkockához rendelünk. Az adatkocka dimenzióinak számát a kocka létrehozásakor mi határozzuk meg, figyelembe véve a felhasználók igényeit és a rendelkezésre álló adatok elérhetőségét. A dimenziók meghatározásánál mindenképpen szem előtt kell tartanunk, hogy a dimenziók elemeinek ne álljon elő olyan kombinációja, melynek az elemzési



terület szempontjából nincs értelme. *(Az OLAP adatkockák-nak nincs kitüntetett sor vagy oszlop dimenziója, azaz ez a két dimenzió tetszőlegesen választható ki az adatkocka összes dimenziója közül.)*

ROLAP dimenziók

Az Analysis Services újdonsága, hogy dimenziót tárolhatunk a többdimenziós adatbázis helyett a relációs adatbázisban is. A dimenzió relációs oldalú tárolásával elérhető, hogy elemeinek száma meghaladja a tízmilliót, és a rájuk épített hierarchiák akár fűzés közben is változtathatók legyenek. A ROLAP dimenziók robusztusságát használhatjuk ki „click-stream” elemzések készítésekor, így például amikor a weblapunkra látogatók adatait és szokásait idősorosan és összegveve elemezzük.

Hierarchiák

A hierarchiák a dimenzióelemek közötti kapcsolatot írják le. Ezek rendszerint fastruktúrájúak. A hierarchia mentén a gyökérponttól a levélpont felé haladva egyre részletesebb adatokhoz jutunk. Egy dimenzió adataiból akár több hierarchiát is építhetünk, például az idő dimenzión létrehozhatjuk az év, negyedév, hónap, nap és az év, hét, nap bontású hierarchiát is.

Kiegyensúlyozott és kiegyensúlyozatlan hierarchiák

Egy hierarchia kiegyensúlyozott, ha a gyökérponttól egy tetszőleges levélpontig tartó útvonalon a csomópontok száma állandó. Minden más esetben a hierarchia kiegyensúlyozatlan.

Szakadozott hierarchiák

Egy hierarchiát szakadozottnak nevezünk, ha legalább egy elemének logikailag legközelebbi szülője nem közvetlenül az elem fölött található szinten helyezkedik el. Például az ország-megye-város hierarchiában Budapest szülője nem a megye, hanem az ország szinten helyezkedik el. Egy hierarchia csak akkor lehet kiegyensúlyozatlan és szakadozott, ha az szülő-gyerekek típusú.

Szülő-gyerekek hierarchiák

A szülő-gyerekek hierarchiák forrástáblája két oszlopot tartalmaz, amelyek együttesen írják le a dimenzió elemei közti kapcsolatot. A gyerek kód oszlop azonosítja a dimenzió összes elemét, míg a szülő kód oszlop az összes elem szülőjét. Ebből következik, hogy egy szülő-gyerekek dimenzió összegző szinten elhelyezkedő eleméhez is tartozhat érték a vele dimenzionált adatkocka forrástáblájában. Ez felösszegzések során problémához vezethet, hiszen definíció szerint a szülő értékét a gyerekek értékéből számítjuk. Esetünkben azonban az adott szülőhöz tartozik egy olyan érték, amely a gyerekek adataitól független. A probléma megoldására felveszünk az adott szülő alá egy új gyerekelemet, amelyhez hozzárendeljük forrástáblában hozzátartozó értéket. Ezt a speciális elemet data member-nek nevezzük, és a hozzá tartozó adatot kétféleképpen jeleníthetjük meg:

- ☞ felösszegzés során nem vesszük figyelembe, hogy a szülőelemhez tartozik érték a forrástáblában
- ☞ a szülőelem gyerekei között szerepeltetjük a data member-t, amelyet a felösszegzések során ugyanúgy figyelembe veszünk, mint egy közönséges gyereket. *(PL.: egy vezető teljesítménye megegyezik önmaga és beosztottai által végzett teljesítmények összegével.)*

A data member egy belsőleg létrehozott és karbantartott dimenzióelem.

Gyakran változó dimenziók (changing dimensions)

Gyakran változónak nevezünk egy dimenziót, ha annak szerkezete *(tagjainak relatív helyzete a hierarchiában)* a rendszeres adatfeltöltések során gyakran változik. Ez egy logikai tulajdonság, amely minden dimenziótípusnál *(kiegyensúlyozott, szakadozott...)* beállítható. Ha egy dimenziót gyakran változóvá teszünk, akkor strukturális változás esetén elég inkrementálisan felösszegezni a változoniál adatkockákat. A lehetőség kihasználásával gyorsabbá tehetjük az adatkockák felösszegzését, de ez sajnos a lekérdezés sebességének csökkenését eredményezi. Így ha egy dimenzió szerkezete ritkán változik, akkor nem célszerű beállítani a fent említett tulajdonságot. A virtuális, ROLAP, és Szülő-gyerekek dimenziók mindig gyakran változó dimenziók.

Dimenziók szerkezetének futás közbeni megváltoztatása

Az OLAP-pal szemben támasztott követelmények között gyakran szerepel olyan igény, hogy az eszköz ne csak elemzésre, hanem modellezésre is alkalmas legyen. A modellezés során egy speciális, kliens oldalról szerkeszthető szülő-gyerekek hierarchiát hozunk létre, amelyek elemei mutatószámok. Ezek a szerkeszthető *(Write-enabled)* dimenziók lehetőséget teremtenek a felhasználóknak, hogy futás közben szerkesszék az említett mutatószámokat, illetve meghatározák hierarchiabeli helyzetüket. Természetesen lehetőségük van mutatószámok felvételére, törlésre, tulajdonságaik megváltoztatására is. A szerkeszthető dimenziók kitágítják az elemzés határait, hiszen velük a felhasználók komplex mutatószámrendszereket, testre szabott hierarchiákat alakíthatnak ki. Mivel az Analysis Services bármely felhasználó által eszközölt változást ugyanabba a táblába *(a dimenzió forrástáblájába)* írja vissza, ezért egy dimenzió szerkeszthetővé tétele előtt pontosan meg kell határozni azon felhasználók körét, akik a dimenzió szerkezetét megváltoztathatják.

Számított mezők (Calculated members)

A számított mezők olyan logikai dimenzióelemek, amelyek értékei más elemekből MDX képletek segítségével származtathatók *(pl. átlagár)*. A számított mezők az eredményt nem, csak annak kiszámításához szükséges képleteket tárolják, és ha szükséges az eredményt a lekérdezés során kiszámítják. A számított mezők hozzáadásával lehetőségünk van az adatkocka információszoftárgáló képességét úgy növelni, hogy a kocka mérete ne növekedjen. Ezáltal csak származtatott információt nyerhetünk, de a több mint



szám előre definiált függvény és az általunk regisztrálható függvények (UDF) segítségével a komplex elemzések könnyebben elkészíthetők.

Dimenzióelemek tulajdonságai (Member Properties)

A dimenzióelemek tulajdonságai olyan plusz információt hordoznak a dimenzióelemeiről, amelyek csoportosítási szempontnak nem minősülnek és az elemzés szempontjából e jellemzők összegzése nem ad új információt, de mint minőségi ismérv fontos tulajdonság. Ilyen lehet például a termék csomagolása, térfogata, színe. Lekérdezés során lehetőségünk van a dimenzióelemek közül azok kiválasztására, amelyek megfelelnek egy adott tulajdonságnak. Egy másik alkalmazási terület lehet a többnyelvű adatbázisok létrehozása. A dimenzióelemek tulajdonságai között letárolva azok idegennyelvű megnevezéseit, az Analysis Services automatikusan a ügyféloldali regionális beállításnak megfelelő nyelvű megnevezéseket adja vissza.

Felösszegzés

Ahogy azt már korábban említettük az előre kiszámított, felösszegzett adatok nagymértékben növelik az OLAP rendszer lekérdezési sebességét. A periodikusan ismétlődő adatbetöltés és felösszegzés során azonban az OLAP adatbázisok mérete exponenciálisan növekszik. Az Analysis Services segítségével szabályozni tudjuk a felösszegzett adatok letárolását. Meghatározhatjuk azt az arányt, hogy az aggregált adatok hány százaléka legyen letárolva, és hány százalékát számítsa ki az Analysis Services a lekérdezések alkalmával. A tárolt adatok térbeli elhelyezkedéséről azonban az Analysis Services matematikai algoritmusra dönt. Például egy 40%-os aggregációtárolás esetén az Analysis Services határozza meg azt, hogy a 40 százaléknyi aggregált adat a 100 százaléknyi aggregációs lehetőség közül hol álljon elő. Ez az algoritmus jó közelítéssel optimális megoldást ad. Az adminisztrátoroknak lehetőségük van a rendszer további finomhangolására, mivel az Analysis Services tárolja azokat a lekérdezéseket, amelyeket a felhasználók futtattak a szerveren. Ezen információk felhasználásával az aggregációk térbeli elhelyezése tovább finomítható.

Az OLAP és OLTP rendszerek közti egyik különbség az, hogy míg az OLTP rendszerekbe rövid, addig az OLAP rendszerekbe hosszabb ideig tart az adatok betöltése és felösszegzése. Nagyon nagy kockák esetében ez akár órákig is eltarthat. Ezért szintén nagyon fontos kérdés az adatok felösszegzésének sebessége. Az Analysis Services háromfajta felösszegzési típust támogat. Full process felösszegzési típus esetén törli az adatok struktúráját és tartalmát, majd újra létrehozva felösszegzi azt. Ezt az eljárást legtöbbször csak a fejlesztés/tesztelés periódusában használjuk, a napi működés esetén az incremental update és refresh data felösszegzési típusok közül választhatunk. E két utóbbi típus nem változtatja meg az adatok struktúráját, csak a benne tárolt adatokat cseréli, vagy új adatokat ad hozzá a meglévőkhöz. Incremental update esetén az Analysis Services létrehoz

egy partíciót, amelyen az új adatokat tárolja, és a felösszegzés közben az adatokca és az új adatok partíciót összevonja, adatait összeadja.

Amennyiben az üzlet természetéről rendelkezünk információval és meg tudjuk határozni azon dimenziók körét, melyek elemeinek bizonyos kombinációjában soha nem szerepelhet érték, akkor a két dimenzió függőségi kapcsolatát meghatározva (Dependent dimensions) a felösszegzés alatt az Analysis Services figyelmen kívül hagyja az említett kombinációk kiszámítását – ezáltal csökkenve az aggregációk számát, és a felösszegzés sebessége tovább gyorsítható. Az üres cellák tárolása szintén az OLAP adatbázisok méretének exponenciális növekedését okozhatja. A legtöbb gyártó ad valamely megoldást a problémára, melyek során például be lehet állítani olyan dimenzióelem kombinációkat, amelyekbe biztos nem kerülhet érték. (például Miskolcon nem értékesítünk gumimatracot), vagy az adatbázist úgy kell megszervezni, hogy az üres cellák kerüljenek egymás mellé, amit aztán a rendszer, ha egy lapnyi üres cellát talál, nem tárol le. Ezzel szemben az Analysis Services egyáltalán nem tárol üres cellákat.

Real-Time OLAP

Az OLAP az On-Line Analytical Processing szavak kezdőbetűjének összetételéből származik, és nem on-line adatelérést jelent a működött rendszerek adataihoz. Az OLAP adatbázisa és a működött rendszerek közötti kapcsolatot az OLAP felé egyirányú, és csak periodikusan, a betöltések alkalmával jön létre. A gyakorlati életben viszont előfordul, hogy az OLAP adatokkában látni szeretnénk a két adatbetöltés közötti változásokat is, azaz az élő adatokat. A Real-time OLAP az Analysis Services olyan szolgáltatása, amely lehetővé teszi, hogy az adatokca forrástáblájának megváltozásával összhangban megváltozzanak a Real-time adatokkában tárolt adatok is. (A Real-time kocka szinkronizálva van a működött rendszerekkel.) A Real-time adatokca nem más, mint egy speciális, ROLAP partícióban tárolt adatokca 0%-os felösszegzéssel. Amennyiben a dimenzió szerkezete is real-time változik, úgy ROLAP dimenziót használva annak hierarchiáját is szinkronizálhatjuk az alaprendszerekhez.

Adatokcák adatainak módosítása

Az Analysis Services lehetőséget teremt a felhasználóknak az adatokcák adatainak módosítására. Az adatokat módosíthatjuk elemi szinten (a hierarchia legalsó szintjén) és valamelyik összegző szinten. Vizsgáljuk meg először az egyszerűbb esetet, amikor csak az elemi szintű adatokat módosítjuk. (Csak olyan adatokca módosítható, amelynek minden hierarchiabeli szülője a gyerekek összegzeként áll elő.) Az Analysis Services a módosításokat sem az adatokkába, sem annak forrástáblájába nem írja vissza, hanem létrehoz relációs oldalon egy külön writeback táblát, amelyben a módosított adatok és az alapadatok különbségét tárolja. Például ha a felhasználó egy cella értékét 90-ről 100-ra növeli, akkor a writeback táblában az Analysis Services letárol +10 értéket a módosítás időpontjá-

val és a módosító felhasználó nevével. Később, az adatokca lekérdezésekor az Analysis Services a módosított célárhoz és annak összes szülő értékéhez hozzáadja a writeback tábla kapcsolódó értékét. (A módosítás az adatokca adatait nem érintette.) A módosított adatokat visszírhatjuk a kockába, ha a writeback tábla adatait partícióvá konvertáljuk, majd az alapadatokat és a writeback adatokat tartalmazó partíciókat összevonjuk. (Hasonló folyamat játszódik le incremental update esetén is.)

Az adatok összegzőszinten történő módosítása esetén a módosított értéket szét kell osztani (allokálni) az elemi szinten elhelyezkedő elemekre, majd az allokált adatok writeback táblába írásával folytatódik a folyamat a fent leírtak szerint. Az allokáció végrehajtását az Analysis Services a következő négy eljárással támogatja:

Allokációs módszerek	Leírás
Egyenlő allokáció	Elemi cellák értéke = Módosított érték / utolsó leszármazottak száma
Egyenlő növekmény	Elemi cellák értéke = Elemi cellák értéke + (Módosított érték - eredeti érték) / utolsó leszármazottak száma
Súlyozott allokáció	Elemi cellák értéke = Módosított érték * súlyozó tényező
Súlyozott növekmény	Elemi cellák értéke = Elemi cellák értéke + (Módosított érték - eredeti érték) * súlyozó tényező

Allokációs módszerek ismertetése

A súlyozótényezők rendszerint 0 és 1 közé esnek és összegük 1-gyel egyenlő. A súlyozó tényező meg nem határozása esetén az Analysis Services az elemi cella és a módosított cella eredeti értékének arányával számol.

Egyedi felösszegzési lehetőségek

Az Analysis Services az adatokcákat csak saját felösszegzési eljárásai szerint tudja felösszegezni. (Sum, Count, min, max, DistinctCount) Gyakorlati problémák megoldása során azonban előfordul, hogy a rendszer felösszegzési algoritmusai helyett valamilyen egyedi felösszegzési eljárást kell alkalmaznunk. Az Analysis Services Custom Rollup/member formuláival az ilyen jellegű problémák is megoldhatóak.

Custom Rollup formula

Tegyük fel, hogy zárókészletek változását szeretnénk hetente vizsgálni. Ha napi készletértékeket (pl.: 5,4,5,6,5) aggregálnánk heti szintre, akkor akkora zárókészletet (25) kapnánk, amely talán sohasem volt raktáron. Ezért a heti készletérték meghatározásához célszerű inkább az átlagos, vagy az utolsó napi készletértéket figyelembe venni. Az Analysis Services nem tartalmaz beépített átlagos vagy „utolsó gyerek” felösszegzési metódust, ezért ezt nekünk kell megvalósítanunk. A megoldás során tetszőleges hierarchia

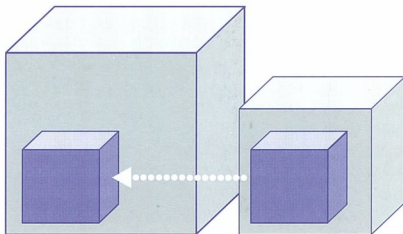
szintekhez hozzárendelünk egy MDX-ben írt képletet (pl.: *Hét=hét.UtolsoGyerek*), amelyet az Analysis Services a felösszegzés során nem vesz figyelembe, hanem az adatokca lekérdezésekor számítja ki. Amennyiben olyan felösszegzési eljárást szeretnénk használni, amely egy hierarchiaszinten nem az összes elemre érvényes, Custom Member Formulát célszerű használnunk.

Custom Member Formula

Hasonlóan működik a custom rollup formulához azzal a különbséggel, hogy nem szinthez, hanem dimenzió-elemhez rendel képleteket, melyeket az Analysis Services a dimenzió forrástáblájában tárol. A képlet nem hivatkozhat olyan dimenzióelemre, amelyet nem tartalmaz dimenziótábla (pl.: *számított mezők*). A Custom member formulát legtöbbször mutatószámrendszerek, gazdasági modellek kialakításakor használjuk.

Számított cellák (Calculated Cells)

Ez az új MDX funkció az Analysis Services olyan lehetősége, amely segítségével több különböző adatokca adatainak részhalmozát tudjuk egy másik adatokca azonos részhalmozában megjeleníteni. Ezáltal tovább finomítható az adatbázis szerkezete, több kisebb és tömörebb kockát létrehozva tudunk összetett elemzési igényeket hatékonyan kielégíteni.



Számított cellák

A számított cellák lehetőséget teremtenek, hogy a számított mezők, custom rollup/member formulák funkcionalitását alkalmazzuk az adatokca egy (jellemzően többdimenziós) részhalmozára vagy akár csak egy cellára is. Legtöbbször olyan esetekben használunk számított cellákat, amikor az adatokca egy részkockájának adatait már más adatokca tartalmazza. (Így lekérdezés során elég az adatokat onnan átvenni.)

Jogosultságkezelés

Az Analysis Services segítségével kétféle lehetőségünk van a jogosultságok beállítására: Cellaszinten vagy dimenzióelem szinten. Mindkét eljárással megoldható egy adott cella adatainak elrejtése, de kétféle eredménnyel. Cellaszintű jogosultság kezelés esetén a felhasználó láthatja az adatokca teljes struktúráját, azaz a dimenzió összes elemét, de az adatokat nem, míg dimenzióelemek szerinti jogosultság kezelés esetén sem az adott elemet sem a hozzá tartozó adatot.



Actions

Az elemzések során gyakran jutunk abba a helyzetbe, hogy észreveszünk egy kiugró adatot, vagy egy tendenciát, de az okokra az adatkockából már nem kapunk a választ. Ilyenkor rendszerint elkezdünk kutakodni különböző információforrásokban, (például Word, Excel dokumentumokban, alaprendszerek adatai között) hogy megtaláljuk az okokat. Ezt a keresési folyamatot gyorsítja fel az Analysis Services azáltal, hogy a felhasználói felületről lehetővé teszi a fent említett források felkutatását. Az action-ök segítségével lehetőségünk van arra, hogy paraméterek átadásával (*kocka, dimenziók neve, egy cellát meghatározó dimenzió elemek halmaza, stb.*) meghívassunk valamilyen külső alkalmazást (*futtatható állományok, MDX lekérdezések, böngészők*). Az Action másik lehetősége, hogy ha megtaláltuk a hibát, gyorsan tudunk reagálni azok elhárítására (*megrendelések leadása, felszólítások küldése...*).

Továbbfűrés (Drillthrough)

A továbbfűrés az action-ök egy speciális fajtája. Segítségével egy adatkocka cellájának adatait tovább részletezhetjük, azaz meghatározhatjuk, hogy az adott cella értéke mely

adatok eredményeképpen állt elő. Továbbfűrés esetén az Analysis Services a forrástáblákon lefuttat egy lekérdezést, amely azokat a rekordokat adja vissza, melyekből az adatkocka adott cellájának értékét az Analysis Services kiszámította.

Ügyfél alkalmazások

A továbbfűrés, mint több más funkció felhasználói oldal kihasználása, erősen ügyfélfüggő. Ezeket ugyanis az Analysis Services mint lehetőségeket ajánlja fel, és az alkalmazásfejlesztőktől függ, kik milyen funkciót támogatnak, és azokat hogyan tálalják a felhasználók elé. A Microsoft az Analysis Services-hez nem fejlesztett önálló kliens alkalmazást, hanem az Excel 2000-et tette alkalmassá OLAP adatkockák böngészésre. Természetesen lehet komplett elemző és fejlesztőeszközöket (*Seagate Analysis, Knosys Zorro, OLAPNagiGO, Max...*) vásárolni kifejezetten MS OLAP technológiára, illetve vannak gyártók, akik saját OLAP elemző eszközüket tették alkalmasra MS OLAP adatbázisok kezelésére.

Kövári Attila, kovari@novosys.hu





A papír nélküli iroda vízióját már sokan megálmodták, mégis úgy tűnik, hogy gyakran papírgyekkel találkozunk az irodákban. Pedig az iroda, vagy nagyobb szervezetek működéséhez szükséges adatok gyakran ott vannak a számítógépek memóriájában vagy háttértárában. Nagy valószínűséggel elektronikus úton érkeztek, és hasonló módon távoznak. Verséget szenved az e-társadalom? Nos, talán nem. A Microsoft gondolt egy merészet, és megkísérelí átvágni a gordiuszi csomót. Ha az iroda-papír kapcsolat megbonthatatlan, akkor lehet, hogy az irodák számának csökkentésével lehet a problémát megoldani.

Az Internet robbanásszerűen növekedő és elburjánzó infrastruktúrája először a magánszemélyek kápráztatta el lehetőségeivel, de hamar feltűnt a különböző üzleti vállalkozások érdeklődésének horizontján is. Sokan e cégek közül már az Internet előtti időben is megértették, hogy üzleti folyamataik elektronikus útra terelésével gyorsabbá, pontosabbá, jobban felügyelhetővé válhat működésük. Az Internet-boom pedig új technológiák egész sorát hozta magával, melyekkel tovább lehet racionalizálni és optimalizálni a vállalat belső életét, partnerkapcsolatait és értékesítési csatornáit.

Ám rögső az út az e-Kánaánig! Az alapvető szoftver- és hardverinfrastruktúra kiépülése szükséges, de nem elégséges feltétele az igazi e-commerce működésének. Az elektronikus üzlet új nyelvének szerepe sokáig kiadó volt, hiszen a leggyakrabban használt protokolloknak van néhány kellemetlen hiányossága. Leveleinket SMTP protokoll segítségével küldhetjük, fogadhatjuk, ami elsődlegesen strukturálatlan adatok, csatolt állományok kezelésére lett kifejlesztve. A weboldalakat HTTP protokollon keresztül HTML oldalak formájában böngészhetjük, a HTML azonban első sorban az információk prezentációjára, egy böngészőn belüli megjelenítésére fókuszál. Egyik technológia sem arra lett szánya, hogy strukturált adatok (akár Internet fölötti) cseréjét megoldja különböző üzleti alkalmazások között.

Vegyünk egy példát! A vállalati beszerzések az Internet előtti időkből döntően papíralapon történtek. Az ügyintéző kitöltött egy rendelési űrlapot, majd normál postai úton vagy faxon továbbította a beszállítónak. Az Internetes kommunikációs csatornák lehetővé tették egyes részfeladatokban a papír nélküli eljövését, hiszen a beszerző on-line katalógusokból választhatta ki az igényelt termékeket, majd rendelését e-mail formájában küldhette el. A túldolgalon azonban a beszállítónak a kapott információkat a legtöbb esetben újra be kellett gépelnie a vállalat saját rendszerébe (vagy rosszabb esetben egy teljesen papíralapú folyamaton kellett keresztülmennie), hogy teljesíthesse üzletfele rendelését. Tehát hiába is állt rendelkezésre a megfelelő műszaki háttér mindkét oldalon és maga a megrendelés elektronikus formában, nem volt lehetőség arra, hogy egy rendelés úgy járja végig a beszállító informatikai rendszerét, hogy többszöri emberi beavatkozást ne igényelt volna.

Az XML megjelenése óriási változásokat hozott ezen a területen. Ez a szabvány kifejezetten strukturált adatok leírására lett kifejlesztve, roppant hamar nagy népszerűség-

ret te szer, és az e-commerce de facto szabványává, ABC-jévé vált. A formátum ismeretében már csupán a konkrét üzenetet meghatározó dokumentumra, az ún. sémára van szükség. A séma egy közös megállapodással, speciális funkció (pl. beszerzés, értékesítés) ellátására létrehozott, kötött elemeket és azok tulajdonságait leíró XML-dokumentum. Az előző hasonlatnál maradvá: ha az XML az e-business ABC-je, akkor a sémák látják el a szavak és mondatok szerepét. A nyelv „választékosságának” bővítésére hozta létre (széles körű támogatottsággal, pl. SAP, CommerceOne) a Microsoft a BizTalk keretrendszerét. Ez irányelvek gyűjteménye XML-ben létrehozott sémák publikálására és alkalmazására. A kezdeményezés támogatására létrehozott weboldalon (<http://www.biztalk.org>) már most rendkívül sok kész séma található, amelyek lefedik az ipar, kereskedelem, oktatás szinte összes területét.

Első pillanatban egyszerű problémának tűnik az, hogy a kommunikáló feleknek egyet kell érteniük az adatok formájában, azaz a sémában. Valójában bonyolultabb a helyzet. Egy beszállítónak számtalan különböző partnere lehet, és ezek teljesen eltérően is formázhatják rendelésüket. Ha nagyon sok partnerről (akár ezres, tízezes nagyságrendben) van szó, akkor ez rengeteg sémát is jelent. Így a sémák nyújtotta előnyöket csak akkor tudjuk kihasználni, ha egy olyan környezettel rendelkezünk, amely nemcsak üzenetek fogadására, küldésére, hanem természetesen formátumban történő transzformálására is képes, azaz integrálni tudja az ügyfelek rendszerét a kiszolgáló rendszerével.

A hatékony, papír és emberi beavatkozást nem igénylő e-businesshez szükség van tehát egy olyan rendszerre, amely automatizálni, menedzselni, integrálni képes egy vállalat belső- és partnerkapcsolatait, üzleti folyamatait, tekintet nélkül az alkalmazott platformokra és operációs rendszerekre. Ezzel a céllal alkotta meg a Microsoft a BizTalk Server 2000-t.

Üzleti megoldások BizTalk Server 2000-el

A BizTalk tehát egy nagy integrációs kihívás zászlovívőjeként indult és egyikezett magát elfogadhatn az üzleti élet résztvevőivel. Ez a közönség azonban kizárólag üzleti érveknek enged. Válasszuk most mi is ezt a nézőpontot, és vizsgáljuk meg, mit is kínál a BizTalk Server 2000.

Az üzleti életben a legfontosabb célok között mindig ott van az üzleti folyamatok gördülékenyebbé tétele és az ügyfél jobb kiszolgálása. Az üzleti folyamatok automatizálása pontosan ezeken a területeken hozza a legtöbb gyümölcsöt. Az olyan megoldások, mint a beszállító lánc, további üzleti partnerek integrálása a vállalat informatikai világába, csökkenti az emberi beavatkozás, és így a lehetséges hibák számát. Javul tehát az információk biztonsága, gördülékenyebbé, pontosabbá, kiszámíthatóbbá válik a vállalat működése, és ezáltal javul az ügyfeleknek nyújtott szolgáltatás minősége is. Üzleti partnereink bepillanthatnak a mi rendszereinkbe, és első kézből megtudhatják azt, ami őket érdekli (pl. mikor tervezünk kiszállítani az általuk megrendelt terméket, vagy hogy banki átutalásuk időben megérkezett-e). Mi is hasonlóan – első kézből – juthatunk a minket érintő információkhoz partnereink rendszereiből. Ez

a fajta információ-önkiszolgálás munkát takarít meg, gyors és pontos. Az alábbiakban pár BizTalk Server 2000 megoldást mutatunk be.



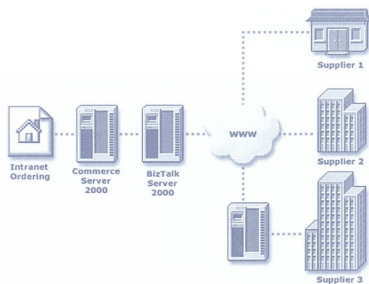
Üzleti folyamat integráció (BPI – Business Process Integration)

A végső cél az üzleti folyamatok minél magasabb szintű integrációja. Ez azt jelenti, hogy mind a beszerzési, mind az értékesítési oldal irányába meghosszabbítjuk informatikai rendszerünket. A horizont, ameddig számítógépeink – és menedzsereink – ellátnak, egyre kitolódik. A BizTalk Server 2000 segítségével olyan ERP (Enterprise Resource Planning) rendszerek köthetőek össze, mint például JD Edwards és Baan, SAP és Peoplesoft. Természetesen lehetőség van Internetalapú kereskedelmi rendszerek és egyedi fejlesztések integrálására is.

Az üzleti szemlélet után térjünk vissza a kritikus informatikai szemlélethez, és végre pillantsunk bele a rendszer működésének részleteibe. A következő paragrafusok leltárserűen ismertetik a BizTalk Server 2000 főbb jellemzőit.

Üzleti partnerek integrálása

- Üzleti partnereinknek csak a legtrikább esetben van a miénkhöz hasonló vagy azzal azonos üzleti információs rendszere. Ennek fényében nem is csoda, hogy a jelenleg kialakult üzleti kommunikáció többnyire papíralapú és manuális – ez idáig sokkal egyszerűbb volt, mint a rendszerek illesztése. Ma azonban a BizTalk Server 2000 egy még ennél is egyszerűbb módszert kínál. Lehetőség nyílik elektronikus dokumentumok küldésére/fogadására az Interneten keresztül, a beszerzési és értékesítési láncok (tradicionális üzleti folyamatokhoz történő) „hozzácsatolására”, továbbá a beszerzési, számlázási, és szállítási folyamatok jobb koordinálására.



Automatizált beszerzés

Azon cégeknek, amelyek beszerzési folyamataikat automatizálni akarják a BizTalk Server 2000 egy ún. rendeléskövető megoldást kínál, mely különösen az üzletmeneten kívül eső, „egyéb” beszerzéseknél hasznos. Ilyenek például a karbantartási, javítási szolgáltatások és az irodai eszközök beszerzése. Ezen folyamatok ma komoly erőforrásokat kötnék le annak ellenére, hogy eladható érték nem teremtenek. A megoldás vázlata az ábrán látható.

A BizTalk Server 2000 funkcionálitása

Intelligens dokumentumirányítás (Intelligent Document Routing)

A dokumentumok megfelelő kézbesítése a rendszer legalapvetőbb feladatai közé tartozik. A BizTalk Server 2000 alapesetben az üzenet fejlécéből olvassa ki a kézbesítéshez szükséges információkat. Amennyiben ez az információ hibás, vagy nem létező helyet jelöl meg, a dokumentum parkolópályára (egy speciális üzenetsorba) kerül, és kézi feldolgozásra vár.

Vannak azonban olyan esetek is, amikor egy dokumentum útját nem tudjuk teljesen előre meghatározni, és csak menet közben alakul ki. Erre is kínál megoldást a rendszer, ugyanis lehetőség van a dokumentumok tartalmától – pl. a megrendelés nettó összegétől – függően más és más helyekre eljuttatni egy dokumentumot. Ez a belső intelligencia nagyon hasznos, hiszen programozást vagy emberi beavatkozást spórolhat meg nekünk, ezzel is gyorsítva és pontosítva a rendszer munkáját.

Hibatűrő dokumentumkézbesítés (Fault-Tolerant Document Delivery)

A BizTalk Server 2000 természetesen alapvető célként tűzi ki a dokumentumok hibátlan kézbesítését, és így a rendszerek integrálásának megőrzését. Elképzelhető azonban, hogy a kommunikációs közeg nem a legbiztonságosabb, és

üzeneteink elveszhetnek, sérülhetnek, esetleg a visszaigazolás nem érkezik meg. Erre az esetre gondolva megadható a hibás kézbesítések utáni újraküldések maximális száma, melyet ha meghaladunk, még mindig van mit tenni. Erre az esetre egy másodlagos átviteli protokoll állítható be, amivel a szerver újra próbálkozik. Amennyiben ez sem jár sikerrel, a dokumentum párolópályára kerül, ahol kézi feldolgozásra vár.

Szinkron és aszinkron kommunikáció

A BizTalk Server 2000 üzenetkezelő struktúrája mind szinkron, mind aszinkron kommunikációkat lehetővé tesz. Szinkron kapcsolattartás esetén az üzenet elküldése után egy válaszüzenet érkezik, ami lehet egy szimpla megerősítés, vagy egy teljesen új, más típusú üzenet. Amennyiben az azonnali válasz nem fontos (pl. *hosszban futó tranzakciók esetén*), aszinkron kommunikáció is lehetséges üzenetsorok segítségével. A BizTalk Server 2000 rendkívül sok adatátviteli szabványt támogat transzportkomponenseivel, pl.: HTTP, HTTPS, SMTP, Microsoft Message Queue, DCOM, FTP. A BizTalk-ot nem használó ügyfeleknek automatizáltan küldhetünk faxot a Windows 2000 fax gateway-én keresztül.

Integráció ERP és egyedi rendszerekkel

A BizTalk Server 2000 fő feladata a rendszerintegráció, így ehhez kapcsolódó tulajdonságai talán a leglényegesebbek. Az integráció fő eszközei az ún. alkalmazásintegrációs komponensek (*Application Integration Components, AICs*). Ezek fejlesztésére a rendszer kétféle megoldást kínál. Az első komolyabb programozói munkát igényel – egy interfészt kell megvalósítani. A második a BizTalk Server 2000 által nagyban segített, úgynevezett pipeline megoldás. Itt az AIC-k kifejlesztését a rendszer hatékony grafikus szerkesztővel segíti (*lásd később*), és a programozást is elvégzi helyettünk. Ez a módszer rengeteg erőfeszítést spórolhat meg nekünk mind a kezdeti fejlesztés, mind pedig a későbbi módosítások során.

A BizTalk Server 2000 csomag tartalmaz egy már kifejlesztett AIC-t SAP R/3 rendszerek felé, így ebben az esetben még az interfész grafikus megtervezése sem szükséges. Több ERP rendszer esetében – pl.: BAAN, Great Plains, JD Edwards, Navision – a fejlesztők tervezik, hogy rendszereikbe integrációs csomagot is illesztenek. Ennek részeként AIC-k dokumentumsémák és map-ek segítik a BizTalk Server 2000 felé történő integrációt.

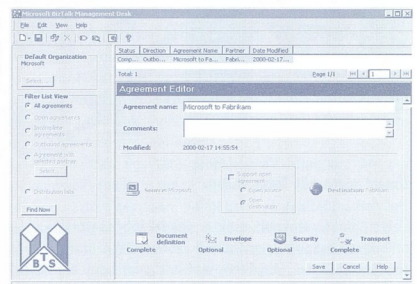
Az egyedi, örökölt rendszerekkel való integráció ugyancsak könnyebb a szakosónál. Ugyanis a BizTalk Server 2000 együttműködik a Microsoft Host Integration Server 2000 rendszerével (*a korábbi SNA Server utóda*). Így a kapcsolódás például AS/400 és mainframe rendszerekkel nagyban támogatott.

Management Desk

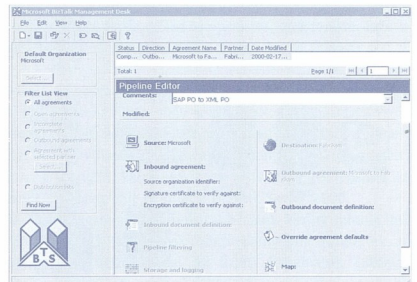
A BizTalk Management Desk az integráció egyik fontos eszköze. Segítségével lehetőségünk van a partnereinkkel között „egyezségeket” teljeskörűen adminisztrálni grafikus felületen. Ezek az „egyezségek” (*agreements*) rögzítik, hogy kívül milyen elektronikus dokumentumokat tekintünk kölcsönösen elfogadottnak és kötelező érvényűnek. Ez az

egyszerű konzolos megoldás a programozás fáradalmait spórolja meg, hiszen a kódolást a BizTalk Server 2000 végzi. A Management Desk két fő felületre az Agreement Editor és a Pipeline Editor.

Az Agreement Editor segítségével a partner és saját rendszerek közti kommunikáció eszközeként elfogadott dokumentumokat, szabványokat tarthatjuk nyilván. Természetesen lehetőség van saját-saját rendszerek közti kommunikáció nyilvántartására is. Az alábbi kép a szerkesztőfelületet mutatja:

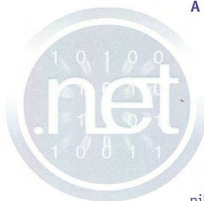


A Pipeline Editor alakítja ki egy konkrét üzleti partnerrel a fő kommunikációs csatornákat. A bejövő és kimenő dokumentumok ezen csatornákon haladnak végig, és követik a lefektetett szabványokat. A dokumentumok megfelelnek az Agreement Editor-ban rögzített formátumnak, és így a partner rendszereinek továbbíthatóak, illetve az onnan jövők rendszerünk számára értelmezhetőek. Az alábbi ábra a Pipeline Editor grafikus felületét mutatja.



BizTalk Editor

A Microsoft BizTalk Editor az a szerkesztőeszköz, amellyel magukat a dokumentumokat – a BizTalk terminológiájában specifikációkat – készíthetjük el. Ez lehet például egy számla vagy egy szállítási értesítő. Itt kell meghatározniunk a dokumentum végleges tartalmát, és azt részletesen – mező-szinten – specifikálnunk kell. Az Editor segítségével a semiből kiindulva felépíthetünk egy dokumentumot, de lehetőség van korábbi – másolt már bevált dokumentumok – átfarmálására vagy több korábbiól egy újabb kialakítására is. Ez a lehetőség sok munkát spórolhat meg a felhasználónak. A jobb átláthatóság érdekében a szerkesztőprogram az



A BizTalk filozófiája

Az e-társadalom várakozással figyelte, hogy a Microsoft hogy válaszol az üzleti világot közvetlenül érintő kihívásra, a kommunikáció rohamos fejlődésére. Mely termékek, milyen megoldások segíthetik a cégek és az üzleti világ fejlődését, melyek terjednek el – esetleg válnak szabvánnyá – és melyek bizonyulnak zsakutcáknak? Erőfeszítések fő irányát a BizTalk név jelel, ennek hátterében vetünk most egy pillantást. Saját bevallásuk szerint két fontos tényező vezette a Microsoftot a BizTalk megalkotására: Különböző alkalmazások integrációja manapság még nehéz. Akár kis, akár nagy cégekről beszélünk, a belső ERP rendszerek, raktárkezelő rendszerek vagy az értékesítés követő rendszerek integrálása komoly költségeket és bonyolult feladatot jelent. Az elektronikus kereskedelem következő hulláma magas szintű alkalmazásintegrációt igényel, mely majd túlnyúl egy szervezet határain. Ez kikerülhetetlen, mikor az üzleti partnereink az Interneten keresztül automatizálják a beszerzési folyamataikat, előrejelző rendszereket építenek ki, vagy akár újabb típusú folyamatokat honosítanak meg. A szituáció érezhetően komoly problémákat sejtet. Ha a cégek problémáival küszködnek saját rendszereik integrálásával, hogy lesznek képesek megbirkózni a feladattal egy

sokkal nagyobb szintéren, az Interneten?

A lehetőség ezen probléma megoldására egy egységes alkalmazás-közi nyelv kialakításában és elterjesztésében (*szabványosításában*) rejlik. Az XML (*Extensible Markup Language*) egy a BizTalk által is felkarolt lehetőség mely sok reménytel kecsegtet. Platformfüggetlen adatátbázolást ír le, így segítségével a különböző rendszerek hatékonyan köthetők össze. A szokásos üzleti adatok és dokumentumok – például egy értékesítési értesítő vagy egy banki átutalás – könnyen átlítható XML formátumba és ezen a nyelven beindulhat a rendszerek közti kommunikáció. Nem számít, hogy ezek a rendszerek különböző operációs rendszereken futnak, vagy hogy más programnyelvben fejlesztették őket. Ahhoz, hogy ez az XML alapú egynyelvűség valósággá váljon még sokat kell tenni. Ezt a feladatot tűzte zászlójára, a BizTalk Community. (<http://www.biztalk.org>). Feladatuknak nemcsak a konferenciákon való részvétel és a publikálást tekintik, hanem működő rendszerek felépítését, az XML terjesztését és továbbfejlesztését. A résztvevő fejlesztők folyamatosan publikálják az új sémákat a társaság Interneten is elérhető könyvtárában. Véleményük szerint csak praktikus, e-világi tapasztalatokkal lehet az egynyelvűség vízióját valósággá tenni.

Budai Zoltán, zbudai@icon.hu



TECH.NET

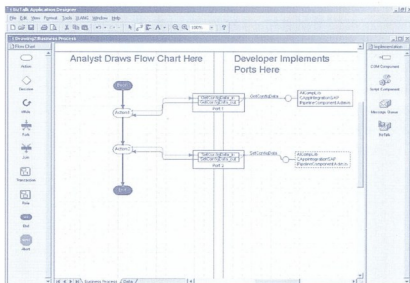
BizTalk Orchestration

Az XML segítségével pontosan és beszédesen fejezhetjük ki adatainkat és az adatcsere formai követelményeit. De... Ezek megadásával még nem mondtuk meg, hogy hogyan és miért akarunk adatokat küldeni és fogadni. Ha a kommunikációhoz illetve az integrációhoz már rendelkezünk is az összes szükséges komponenssel, még mindig kell egy olyan mechanizmus, amely meg tudja mondani a kapcsolatban álló feleknek, hogy mikor és hogyan kommunikáljanak. Napjainkban az ilyen jellegű utasítások szét vannak szórva a programok kódjába. Ez rosszul méretezhető, könnyen sebezhető megoldásokat eredményez, melyek nem képesek több szervezeti egységre (*vagy akár vállalatot*) átfogó műveletek lebonyolítására. Ennek a problémának a megoldására a Microsoft a BizTalk Orchestration-t ajánlja, ami a BizTalk Server 2000 része.

A Microsoft értelmezésében a „hangszerelés” (*orchestration*) B2B interakciók fejlesztését és végrehajtását jelenti. Az interakciók történhetnek több idősíkbán, áthidalhatnak alkalmazásokat, szervezeti egységeket, embereket a vállalatlan belülről illetve az Internet révén szerte a nagyvilágból. Ahhoz, hogy ez sikerüljön, három előfeltételt teljesülése szükséges.

A folyamat definíciójának és implementációjának különválasztása (*Separation of Definition from Implementation*)

Egy üzleti folyamat megvalósításához szükség van a folyamat modelljére, illetve a szoftverkomponensekre, amelyek megvalósítják a definíció egyes lépéseit. Napjainkban e



kettő elválaszthatatlan egymástól. Általában ugyanaz a kód valósítja meg az alkalmazásokat és komponenseket, mint amelyik elindítja, ellenőrzi magát a folyamatot. Az egymásba fonódás nem túl szerencsés megoldás, hiszen így a végtermék nehezen méretezhető, platform-, alkalmazás- és protokollfüggő. A BizTalk Orchestration segítségével vizuális környezetben alakíthatjuk ki az üzleti folyamat egyes lépéseit, majd minden egyes lépéshez egy, a funkciót megvalósító szoftverkomponenst rendelhetünk hozzá. Így például egy platformváltás csupán komponens(ek) cseréjét igényli, az üzleti folyamat egésze változatlan maradhat.

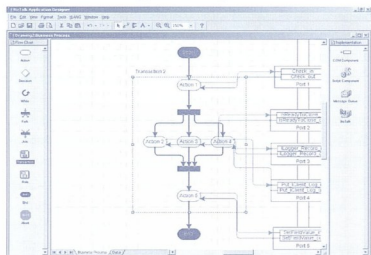
Dinamikus folyamatok (*Dynamic Processes*)

Az üzleti folyamatok általában egyáltalán nem statikusak, állandóan változnak, új résztvevőkkel működhetnek együtt. A BizTalk Orchestration-nel kezelt folyamatok dinamikusan változhatnak futási időben a különböző alkalmazásokból illetve adatforrásokból érkező információk alapján.

Egy kereskedelmi webhely esetén például tudjuk, hogy a vásárlási folyamat végén az árakat majd ki kell szállítani, de a szállító kiléte gyakorlatilag csak akkor derül ki, ha a vásárló már az összes mezőt kitöltötte a rendelési űrlapokon. A BizTalk Orchestration-nel olyan végrehajtható folyamatot alakíthatunk ki, ahol a rendelés egy absztrakt kiszállítóhoz kerül. A tényleges futási időben a kiválasztott kiszállító alkalmazása kapcsolódik a szállítási folyamathoz, és az absztrakt elem mindig ezzel lesz behelyettesítve. Így tetszőleges számú kiszállítót rendelhetünk hozzá a folyamathoz, a folyamat logikai modelljének legkisebb változtatása nélkül. A megoldás így nem csupán dinamikus, de robusztus és méretezhető is lesz.

„Bármít bármihez” integráció (*Any to Any Integration*)

A harmadik fontos feltétel szerint az üzleti folyamat összes alkotóelemének képesnek kell lennie kommunikálni vagy kapcsolódni a tényleges munkát végző és gyakran heterogén környezetben futó szoftverkomponensekhez. A BizTalk Server 2000 számos adaptert tartalmaz, melyeken keresztül elterjedt üzleti, CRM, számlázási alkalmazásokhoz tudunk kapcsolódni. A nyitott architektúra alkalmazás-, platform- és protokollfüggetlen integrációt tesz lehetővé.



A BizTalk Orchestration elemei

A BizTalk Orchestration működését az ún. Engine felügyeli. Futtatja a folyamatokat, felügyeletet, hibakeresést tesz lehetővé. Az összes tranzakció aktuális állapota egy adatbázisban tárolódik XML formátumban, így szabványos eszközökkel felderíthető, hogy mely folyamatok várnakoznak, hol vannak a szűk keresztmetszetek. A széleskörű integrációt az előbb említett adapterek biztosítják. Az üzleti folyamatok kialakítását, komponensek hozzárendelését az egyes lépésekhez az Application Designer segíti, aminek van néhány olyan tulajdonsága, amely mindenképpen megér még néhány szót.

Az üzleti folyamatok egyes lépései gyakran függetlenek egymástól. (Például, ha egy rendelés tételei más és más szállítótól érkeznek, akkor ezeket párhuzamosan is elküldhetjük az érintett partnerekhez.) A párhuzamosság kérdéséről elválaszthatatlan a szinkronizáció problémája. (Amíg az összes rendelést hibátlanul el nem küldtük, addig nem küldhetünk rendelésmegerősítést.) A programozási nyelvek és fejlesztőeszközök általában mostohán kezelik ezeket a problémákat. Az Application Designer segítségével viszont grafikus eszközökkel alakíthatunk ki párhuzamos folyamatokat, melyek kezelését és szinkronizálását az Engine automatikusan elvégzi.

A BizTalk Orchestration hosszan futó, akár napokat vagy heteket átívelő tranzakciók kezelésére is lehetőséget ad. Ezek a tranzakciók számos résztvevőt és platformot foglalhatnak magukba egyetlen tranzakációs kontextusban. Az Application Designer ezt is nagyban segíti, hiszen csak a folyamat tranzakció alá eső részeit kell néhány kattintással egy téglaalapba foglalni, hogy a lépések atomi egységet alkossanak. Ha a tranzakció futása közben hiba lép fel, akkor egy szintén általunk definiált folyamattal vizsgálgrgethető a változások.

Dániel Kolos, kdaniel@icon.hu



Mi van a BizTalk Server 2000 csomagban?

☞ Grafikai fejlesztőeszközök (*Graphical Development Tools*)

Két eszköz – a BizTalk Editor és BizTalk Mapper – segíti az XML sémák és üzleti dokumentumok fejlesztését és fordítását.

☞ Nagymegbízhatóságú dokumentumközvetítő (*Reliable Document Interchange Engine*)

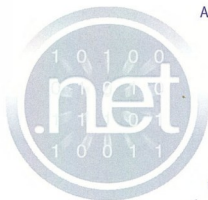
Lehetővé teszi a tartalom alapján történő útválasztást, és többfajta dokumentumtípust támogat: XML, EDI (*EDIFACT* és *X12*), továbbá egyedí és egyszerű szöveges.

☞ Magasszintű biztonsági eszközök

Nyilvános kulcsú titkosítás, digitális aláírás, titkosítás, stb.

☞ Alkalmazásadapterek (*Application Adapters*)

A háttérrendszerekkel való integráció megvalósítására



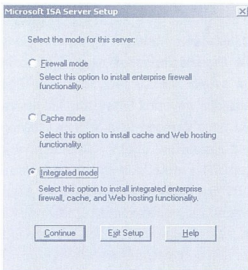
Az Internet használatának folyamatos és dinamikus növekedésével együtt kerülnek előtérbe a biztonság és teljesítmény kérdéseinek kihívásai. A vállalatoknak nemcsak az Internetről a belső hálózatba áramló, valamint a kifelé irányuló adatfolyamot kell ellenőrizni, felügyelni, hanem figyelembe kell venni a belső és a kapcsolódási infrastruktúra erőforrásait, melyek jelentős mértékben befolyásolják a sávszélességet, kliensek válaszidejével szemben támasztott követelményeket.

Az Internet robbanásszerű növekedése váltotta ki az igényt egy nagy teljesítményű, könnyen adminisztrálható internet átjáró után, ami biztonságos kapcsolaton keresztül növeli a hálózat teljesítményét.

A Microsoft Internet Security and Acceleration Server 2000 kielégíti ezeket a követelményeket, magas fokú biztonságot és erőteljes hálózatgyorsítást magaslátva. Skálázható nagyvállalati tűzfal és web-gyorsító (*cache*) szerver, a legrögzőbb integrációt nyújtva a Microsoft Windows 2000 operációs rendszer házirenden alapuló biztonsági, gyorsító és Internet felügyeleti szolgáltatásaival, és nem utolsósorban az Active Directory-val.

Hálózati biztonság és gyorsító szolgáltatások egységes felügyelete

Az ISA Server ISA Server biztonságos, gyors és felügyelhető Internet elérést biztosít. A nagyvállalati tűzfal és a nagyteljesítményű web-cache funkcionalitást kombinálva az ISA Server átlagos felügyeleti infrastruktúrájával csökkenti a hálózat bonyolultságát és költségeit. Akár egy integrált rendszer alternatíváját, akár egy elkülönített tűzfal és gyorsítótár (*cache*) alternatíváját választva a vállalat az általa kívánt erőteljes teljesítményhez és flexibilitáshoz jut.



Üzem módok

Rugalmas felügyeleti eszközök biztosítják a többszintű felhasználó házirend, alkalmazás, ütemezés és tartalom felügyeletét. Az ISA Server szorosan integrálódik a Windows 2000 biztonsági rendszerével, címterával (*Active Directory*), virtuális magánhálózati (*VPN*) rendszerével és a Quality of Service (*QoS*) által megvalósított sávszélesség-felügyeletével, leegyszerűsítve a menedzselési feladatokat.

Biztonság

Az ISA Server skálázható, többretegű nagyvállalati tűzfalat tartalmaz, mely a dinamikus csomagszűrés, a belső IP címek átfordítását végző NAT (*Network Address Translation*), az intelligens alkalmazás szűrők és a beépített betörés észlelés szolgáltatásait nyújtja.

Az ISA Server megoldást nyújt:

- a hálózat védelmére illetéktelen hozzáférés ellen,
- a web szerverek védelmére külső támadások ellen,
- a ki- és bejövő hálózati forgalom szűrésére,
- a gyanús tevékenységen alapuló riasztásra.

Gyorsítótár

Az ISA Server nagy teljesítményű gyorsítótár funkciója a Web elérés sebességét növeli takarékoskodva a sávszélességgel. A Windows 2000 Advanced Server előnyeit kihasználva (*Load Balancing*) jól skálázható, dinamikus terhelésselosztást biztosít. A gyulakon keresett Weboldalakat aktívan tárolja megtanulva az egyes hozzáférési modelleket.

Az ISA szerver gyorsítótár képessége biztosítja:

- a gyakran használt oldalak tárolását,
- a hálózati forgalom csökkentését sávszélesség me takarítással,
- a felhasználók számára gyors Web elérést a túlterhelt Internet helyett az objektumok lokális biztosításával,
- a világ számára Weboldalak terjesztését nagy hatásokkal és költséghatékonyan.

Bővíthető és testreszabható

Az ISA Server széleskörű szoftverfejlesztő készletet tartalmaz - SDK (*Software Development Kit*) - és egyéb API-kat az adminisztrációhoz, az alkalmazásszűrőkhöz, a webszűrőkhöz, és a gyorsítótár felügyelethez, tág felület biztosítva a bővítéshez és teste szabáshoz. Az ISA Server kiterjeszti a vállalatok biztonságát és gyorsítótár megoldásait:

- Bő választékot nyújt külső gyártók termékeivel (*pl.: vírus irtás, oldal szűrés...*) való bővítéshez.
- Testre szabott modulok fejlesztésével vállalati biztonsági házirend és különböző jelentések kialakításával.

Nagyvállalati biztonság (Enterprise Security)

A hálózatok és felhasználók Internetre csatlakoztatása a biztonság és termelékenység kérdéseit vetik fel. Az ISA Server szolgáltatásai a vállalati Web hozzáférések ellenőrzésében és a monitorozás használatában segítenek. Az ISA szerver védelmet nyújt az illetéktelen behatolás ellen, vizsgálja a hálózati forgalmat, és támadás észlelésekor riasztja a rendszergazdát.

Többretegű tűzfal

A hálózatok rendkívül változatos módon támadhatók. Az ISA Server maxi-málisan védi a hálózat biztonságát csomag (*packet*)-, port (*circuit*)- és alkalmazásszintű (*application*) szűrési szolgáltatásaival, csökkentve az illetéktelen hozzáférés rizikóját.

Intelligens alkalmazásszűrők

Az alkalmazásszűrők felismerik a hálózaton keresztül-



folyó adat tartalmát és érvényesítik a beállított házi rendet. Ellenőrzik az alkalmazás specifikus forgalmát – mint pl. elektronikus levél vagy videokonferencia – adattípus ismerő szűrőkkel növelve a biztonságot. Az ISA Server kihasználja az Active Directory házi renden alapuló felületesi funkcióit.

☞ Dinamikus IP szűrés

A „csak a szükséges port-ok engedélyezése” és „a port nyitottsága csak az aktív folyamatok idejére” elvén alapuló korlátozások csökkentik a külső támadások rizikóját.

Gyorsítótár szolgáltatás (Web caching)

A vállalatok megjelenése az Interneten hatalmas gazdasági és piaci előnyökkel bír számukra, de csak akkor, ha weboldaluk tartalma gyorsan és költséghatékonyan érhető el. Az ISA Server webgyorsítótár szolgáltatása csökkenti a teljesítményből fakadó szűk keresztmetszetet és felszabadítja a hálózati sávszélesség erőforrásait lokális tartalomszolgáltatásával.

☞ Nagy teljesítményű gyorsítótár szolgáltatás (Web cache)

Az ISA Server extra gyors RAM kezelésével és hatékony lemezkezelésével gyorsítja a webelérést takarékoskodva a hálózati sávszélességgel.

☞ Skálázhatóság

Ha egy proxy szerver nem skálázható, a tárolandó tartalom növekedésével a szolgáltatás teljesítménye negatívan hathat vissza, azaz bedugulhat az Internet átjáró. Az ISA Server fejlesztése a skálázhatóságot szem előtt tartva történt, hatékony bővíthetőséget és az ún. Cache Array Routing Protocol (CARP) implementálásával dinamikus teljesítményelosztást nyújtva.

☞ Aktív gyorsítótár használat (Active caching)

Az ISA szerver vizsgálja az egyes oldalak elérésének módját és gyakoriságát, eközben kiválasztja és alkalmazza a legjobb algoritmust a tartalom tárolására. Proaktív tárolási eljárásaival és komplett weboldaluk ütemezett letöltésével az ISA Server mindig a legfrissebb tartalmat szolgáltatja a felhasználók számára.

Rugalmas felületes

Az ISA Server hatékony keretrendszerrel biztosítja a házi renden alapuló felületeseknek és leegyszerűsíti az internetezés felületesi feladatait.

☞ Windows 2000 integráció

Az ISA Server integrálja a Windows 2000 operációs rendszer több alapszolgáltatását, ezzel erőteljes és konzisztens felületet nyújtva a felhasználók, konfigurációk és szabályok adminisztrációjához. Az alapszolgáltatások része a felhasználói azonosítás, a felületesi eszközök, a sávszélesség felületes (QoS), a NAT és a virtuális magánhálózatok (VPN), melyek a Windows 2000 technológiáján alapulnak. Az Active Directory szolgáltatásait használva az ISA Server egyszerűsíti a felületesi feladatokat.

☞ Házi rend szabályozás.

A biztonságot fokozó házi rendek az ISA Serverrel könnyebben létrehozhatók és végrehajthatók. A hálózat- és rendszer-felületesi eszközök adminisztrálhatják a csoportok és felhasználók jogosultságait, az alkalmazásokat, a tartalom típusát és az ütemezést.

☞ Felületesi eszközök gazdag választéka

A biztonságot és hálózat-felületes fontos szempontja a hálózat megbízható és egyszerű monitorozása. Az ISA szerver részletes naplózása és testre szabható riasztási funkciója szolgáltat információt a rendszergazda számára a hálózat állapotáról, esetleges biztonsági réseiről. Grafikus jelentések, és távoli felületes segíti a hálózati szakemberek munkáját a szerver teljesítményének optimális hangolásában.

Skálázható biztonság

A hálózati biztonság szintje és házi rendje minden vállalatnál más és más. A sávszélesség elosztása, különböző oldalak elérése és az engedélyezett protokollok beállítása eltérő megközelítést igényel. Az ISA Server széleskörű szoftverfejlesztő készletet és külső gyártók megoldásainak integrálhatóságát nyújtja a testreszabhatóság megvalósítása érdekében.

☞ Széleskörű fejlesztőkészlet.

Az ISA szerver olyan széleskörű fejlesztőkészletet tartalmaz (Software Development Kit), mely teljes API dokumentációjával és példakod végigvezetett szűrők és adminisztrációs modulok megvalósításával támogatja a vállalatok biztonsági koncepcióit. A fejlesztőkészlet (SDK) letölthető a Microsoft Web oldaláról.

☞ Külső gyártók megoldásainak nagy választéka.

Külső gyártók egyre bővülő termékpalettája tágítja ki az ISA szerver funkcionalitását, melyek olyan lehetőségeket kínálnak, mint a vírus ellenőrzés, a tartalomszűrés, a valós idejű monitorozás, a Weboldal tiltás, a jelentés-készítés és felületesi eszközök.

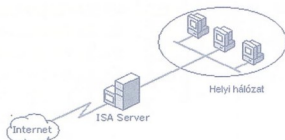
☞ Hatékony felületes.

Csökkentsük le a felületesi feladatokat azok automatizálásával! Az ISA Server támogatja a szkriptelést, mely a felületesi funkciók és szabályok automatizálásának hatékony eszköze.

ISA tűzfal

Az ISA Server könnyű, biztonságos és költséghatékony utat kínál a vállalati asztali gépek Internet eléréséhez. Az ISA Server az Internet és a belső hálózat között elhelyezkedve továbbítja megbízható úton a kliens gépek kéréseit.

Az ISA szerver biztonságos átjáró a helyi hálózat és az Internet között. Az ISA Server tűzfal transzparens a hálózati kommunikációban résztvevők számára. Az Internet felhasználónak nem kell tudnia a tűzfal létezéséről addig, amíg meg nem próbál az ISA Server által védett szolgáltatáshoz hozzáférni.



ISA tűzfal

A tűzfal támogatja az internetes protokollok minden fajtáját, többek között a HTTP-t (*Hypertext Transfer Protocol*), az FTP-t (*File Transfer Protocol*), a Real Audio/Video-t, az IRC-t (*Internet Relay Chat*), a H.323-at, a Windows Media-t, valamint a levelező és hírolvasó protokollokat.

Port és alkalmazás szintű védelem

Az ISA Server a kliens gépek és az Internet között elhelyezkedve port és alkalmazási szinten védi a belső hálózatot. Port védelmi szinten engedélyezhető a csomagszűrés, mellyel lényegében megszüntethető minden forgalom a helyi hálózat és az Internet között. Következésképpen olyan statikus csomagszűrők állíthatók be, melyek blokkolják a nemkívánatos csomagok bejutását a lokális hálózatra. Alkalmazás szinten a tűzfal védi a hálózatot, miközben engedélyezhető olyan Internet szolgáltatások, mint a Telnet, RealAudio, IRC és a Windows Media.

Csomagszűrés (*packet filtering*)

Az ISA szerverben olyan IP csomagszűrők alkalmazhatók, melyekben definiálható a szolgáltatás típusa, a port szám és a forrás vagy cél számítógép neve. Az IP csomagszűrő olyan statikus filter, amely az adatsomag bizonyos portokon történő áthaladását engedélyezi vagy tiltja. Az engedélyező szűrők átértesítik a csomagot, de csak a definiált portokon. A tiltó szűrők mindig blokkolják egy csomag átjutását az ISA Serveren.

Azok a csomagok, melyekre nincs definiálva se blokkoló, se engedélyező szűrő, az ISA Server alkalmazás szintű szolgáltatásaihoz továbbíthatók. Ezen a szinten állítható be olyan házirend, melyben szabályok összességével definiálható, hogy milyen típusú kommunikáció engedélyezett. Az ISA Server házirendje mondja meg azt, hogy melyik típusú kommunikáció érheti el a Web Proxy vagy a tűzfal szolgáltatást. A portok csak a tényleges kommunikáció idejéig vannak nyitva, a kapcsolat befejezése után rögtön lezáródnak.

Dinamikus csomagszűrés

Az ISA Server támogatja a kimenő és bejövő csomagszűrés is. Dinamikusan ismeri fel, hogy mely csomagok férhetnek hozzá a helyi hálózat port és alkalmazás szintű szolgáltatásaihoz. Olyan házirendek konfigurálhatók be, melyek automatikusan nyitnak meg egy portot, ha az a csomag számára engedélyezett, és azt a kommunikáció befejeztével rögtön le is zárják. Ez a megközelítés csökkenti a védtelen portok számát mindkét irányban és magas szintű, problémamentes biztonságot nyújt a hálózat számára.

Közzététel funkciók

Az ISA Server a helyi hálózat biztonságának veszélyeztetése nélkül teszi lehetővé belső szolgáltatások közzétételét. Olyan szabályok definiálhatók, melyek az Internet felőli kéréseknek megfelelően továbbítják a kérést az ISA Server mögötti Web vagy egyéb szerver-szolgáltatás felé. Az ISA Server olyan hidat épít a szolgáltatás vagy alkalmazás és a külső hálózati csatlólója közé, mellyel a szolgáltatás figyelheti a külső porton érkező kéréseket, azaz meghatározott külső portját "rábízza" egy belső szerver szolgáltatására.

Szerver szolgáltatások közzététele

Vegyünk például egy Exchange szervert a helyi hálózatban, amely leveleket küld az Internet felé. Az SMTP levél-továbbítás szükséges feltétele a 25-ös port be- és kimenő kommunikációjának engedélyezése, ezért az Exchange szerver SMTP szolgáltatása az ISA Server külső portjához van kötve. Ezzel a technikával az Exchange figyelheti a bejövő kérélmeket. E szolgáltatás üzemeltetéséhez olyan szerver közzétételi szabályok szükségesek, amelyek specifikálják azokat a belső szervereket, amelyek számára a web publikálás vagy szolgáltatás közzététel engedélyezett. Az ISA Server figyeli a kéréseket és továbbítja a megfelelő kiszolgálóhoz.

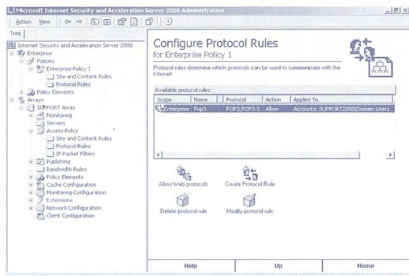
Web közzététel (*Web publishing*)

A szerver szolgáltatások másik példája a Web közzététel. A Web kiszolgáló az ISA Server mögé helyezhető a Web közzétételi szabály definiálásával, amely engedélyezi a Web szerver szolgáltatásait az Internet felé. A Web szerver felé irányuló kéréseket az ISA Server megszakítja az Internet kliensei felé azt közpetet nyújtja, mintha a Web szolgáltatást maga futtatná. Az ISA Server a kliensek kéréseit a saját gyorstárból teljesíti, azokat csak a kívánt oldalak hiányában továbbítja a Web szerver felé.

Amíg a Web szerver a helyi hálózat biztonságát élvezzi, az ISA Server gyorstároló módszerével tehermentesíti az Internet felé irányuló kapcsolatot.

Web gyorstár

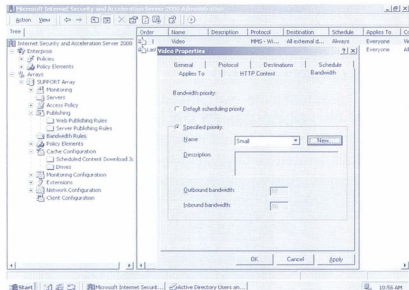
A legtöbb webböngésző támogatja a lokális gyorstárolást, amivel a weboldalak az Internet helyett a helyi tárolóból hívhatók elő. Az ISA Server ugyanezt a koncepciót követve a gyakran lekért objektumok központosított tárolását nyújtja, ezzel növelve a kliensek böngészőinek teljesítményét, csökkentve a válaszidőt és takarékoskodva az Internet szolgáltató felé irányuló gerincvonal sávszélességével. A lemez gyorstárból szolgáltatott objektumok nagyságrenddel kevesebb feldolgozási időt igényelnek az Internetről kért objektumokkal szemben.



IP szűrők

Sávszélesség házirend

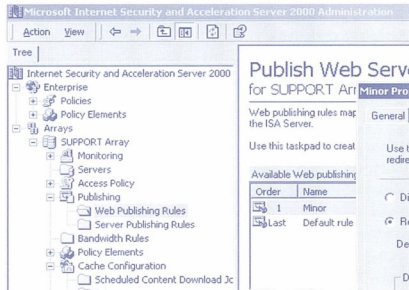
A sávszélesség szabályaival konfigurálható, mekkora sávszélesség biztosítandó az egyes Internet felé irányuló kéréseknek.



Sávszélesség házirend

Közzététel házirend

A szerver közzétételi szabályok szűrnék minden ki- és bejövő kérést. A kérésnek megfelelően építenek hidat a szolgáltatást nyújtó szerver és az Internet kliense közé.



Közzététel házirend

Vállalati (enterprise) és tömb (array) házirend

Az előbb felsorolt szabályok kombinációjával tömb szintű házirend definiálható, amely csak a tömböt alkotó ISA Serverekre érvényes. A vállalati házirend - mely oldal és tartalom, valamint protokoll szabályokból áll, bármely tömbre alkalmazható, kibővíve a tömbök saját házirendjét.

Szétosztott gyorsítók készítése

Az ISA Serverek hatékony tulajdonsága a gyorsítók skálázható felépítése, hierarchikus gyorsítók kialakítása. A kialakítás alapja a Cache Array Routing Protocol, mely segítségével láncba vagy hierarchiába fűzött gyorsítók alakíthatók ki.

A gyorsítók tömbök a skálázhatóságot adják, ekkor több gyorsítót egy tömbben egyesítve a terheléselosztást tudjuk elérni.

A láncba (hierarchiába) szervezett gyorsítók pedig a megfelelő információkat a láncban végighaladva keresik, az alsóbb szinten meglévő információkérés nem továbbítódik a felső szintre. Az SSL kapcsolatos a láncba kötött gyorsítók esetén is működnek.

Természetesen a két megoldás kombinálható egymással, ahol szükség van rá, tömböket készíthetünk a hierarchián belül.

Varázslók

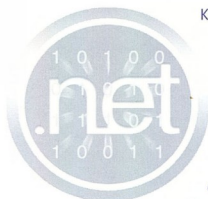
Az ISA Server könnyebb felügyelete érdekében tartalmaz néhány varázslót, hogyan azt a Windowsban már megszoktuk. Az alábbi kép az új protokoll varázslót mutatja be.



Új protokoll szabály varázsló

Windows 2000 integráció

A Windows 2000 és az Active Directory felügyeleti funkcióit az ISA Serverbe is beépítették. Az integráció segítségével a felhasználó Active Directoryba történő bejelentkezésekor az ISA Serverben beállított házirendeknek megfelelő jogosultságokat is megkapja. Az Active Directoryban lehet a külső és a belső Web elérés szabályokat megadni, a szolgáltatások engedélyezésével, vagy tiltásával. A Performance Monitor segítségével az ISA Server teljesítménye távolról is felügyelhető, az Event Log az MMC és a távoli felügyelet (termindül szolgáltatások) egyaránt működnek.



Kezdjük az elején, mire jó a HIS 2000?

A Microsoft Host Integration Server 2000 a Microsoft Windows operációs rendszerek és a nem Windows alapú vállalati rendszerek közötti kapcsolatok hatékony kialakítását segíti. A legtöbb szervezetben az úgynevezett „nagygépes” platformokon kritikus alkalmazások futnak, ezért rendkívül fontos, hogy ezen alkalmazások és adatok is integráltnak illeszkedjenek a vállalati környezetbe. Megbízható, biztonságos és kényelmes legyen ügyfelőladról is ezek elérhetősége. Legyen jó menedzselhető és teremtdíjjon meg a más rendszerekkel való integrálhatóság is.

A Microsoft Host Integration Server 2000 alkalmazásával hatékony kapcsolat teremthető az IBM mainframe, AS/400 és UNIX rendszerek irányába anélkül, hogy akár kliens, akár szerveroldalon a használni kívánt alkalmazások kódját módosítani kellene.

Segítségével mind a hostok mind pedig a munkaállomások oldalán jelentős erőforrások szabadíthatók fel, például a terminálkapcsolatokat, az alkalmazások forgalmát, a print jobok kezelésének terhé is átveheti. Ez azt eredményezi, hogy a kliensek nincsenek szükségű pécádál a nyomtatáshoz szükséges alkalmazások futtatására, de host oldalán is jelentősen egyszerűsödik az adminisztráció. A kliensek és a HIS 2000, illetve az egyes HIS 2000-ek között tömörített, kódolt adatátvitelt alakítható ki, amely jelentős teljesítményjavulást eredményez és csökkenti a hálózat terheltségét.

A Host Integration Server 2000 megszünteti a fejlesztők számára a különböző platformok által szabott korlátokat és megnyitja az utat ezzel az olyan elosztott és méretezhető megoldások előtt, amelyek kiterjeszthetők a meglévő folyamatokat és adatelérést anélkül, hogy a meglévő host kódot újra kellene írni vagy esetlegesen teljesen áttervezni. Ez a tulajdonság lehetővé teszi, hogy gyorsan készüljenek olyan új üzleti alkalmazások, amelyek Windows DNA 2000 technológiát használnak, de emellett nem vész el a korábbi saját fejlesztésekbe fektetett energia és költség sem. A Host Integration Server 2000 a fejlesztők számára három szinten kínál nagyfokú integrációs lehetőséget. Ez általánosan a következőket jelenti:

Hálózati szint

Sok szervezet számára a nagygépes környezetben futó alkalmazások nélkülözhetetlenek és üzleti szempontból rendkívül kritikusak. A Host Integration szerver egység, menedzselhető host elérést biztosít mind a kliens/szerver, mind a webalapú hálózatok számára. A szerver integrálja az Active Directory és a hostok által nyújtott biztonsági szolgáltatásokat.

Adatszint

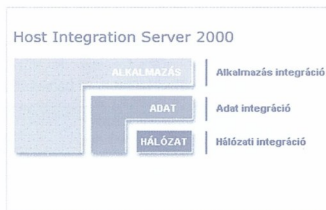
Minden olyan szervezet, amelynek az IT infrastruktúrája nem egyetlen rendszeren alapul, illetve használ valamilyen szinten szétosztott adat- vagy információbázist, találkozott már a problémával, hogy milyen módon teremtesse meg a különböző rendszerekben elhelyezkedő adatok konzisztenciáját és szinkronizációját.

A HIS 2000 a mainframe-ken, AS/400, UNIX Windows NT vagy Windows 2000 Server-eken elhelyezkedő relációs DB2 adatbázisban vagy a filerrendszerben elhelyezkedő adatok teljes, megbízható és biztonságos elérését biztosítja akár objektumorientált módszerekkel is.

Alkalmazásszint

Mivel a legtöbb cég komoly befektetéseket eszközölt a hostokhoz futó üzleti alkalmazások kifejlesztésére, megvásárlására és teszte szabására, ezért fontos szempont, hogy az üzleti logika módosítása nélkül tudjuk ezen rendszereket integrálni, akár egy webfelületen megjelenő front-ennel. A különböző kapcsolatok kialakításán túl nem utolsó szempont az sem, hogy ehhez lehetőleg ne keljen ismerni és módosítani a meglévő alkalmazások kódját.

A COM Transaction Integrator segítségével olyan elosztott alkalmazások készíthetők, amelyek a Microsoft Transaction Services és IBM CISC és IMS rendszereket képesek integráltnak kezelni. Így a bonyolult host kódot nem kell áttérni, hanem a webfejlesztők használhatják a COMTI szolgáltatásait a CISC és IMS elfedésére és COM objektumként használhatók szétosztott webkörnyezetben.



Háromszintű integrációs lehetőséget nyújt az új Host Integration Server 2000

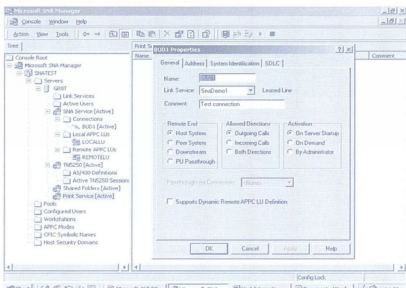
Mint látható ez a termék a korábbi években sikeres SNA szerver utóda. Ez azonban rögtön felvet két nyilvánvaló kérdést. Az egyik természetesen az, hogy mi a helyzet a régi SNA Server-rel, a másik pedig hogy miben más ez, miben jelent ez a termék újat? Az első kérdésre a választ nem nehéz kitalálni, vagyis a Host Integration Server 2000 az SNA Server teljes funkcionalitását öröklí, de emellett számos területen növelték használhatóságát, megbízhatóságát, illetve jónéhány új funkcióval bővült. Néhány gondolatban strukturáltan megpróbáltuk összefoglalni, hogy melyek ezek.

Windows 2000 integráció.

A Microsoft Host Integration Server 2000 intelligensen használja az Active Directory által nyújtott alapszolgáltatásokat. Ez azt is jelenti többek között, hogy a kliensek az Active Directory segítségével találják meg a szervereket a hagyományos subdomain struktúrán túl. A felhasználóknak a részletes protokollbeállítások helyett elegendő csak az organization unit nevét konfigurálni. Természetesen a kiszolgáló erőforrások elérhetők több OU-n keresztül is.

MMC alapú többgépes adminisztráció

A Microsoft Management Console alapú adminisztráció segítségével az egész szervezetben belüli konzisztens módon lehet adminisztrálni a HIS 2000, illeszkedve a többi Windows 2000 alapú komponenshez. Vannak olyan komponensek, amelyek központi adminisztrációt tesznek lehetővé, mások gépszinten konfigurálhatók, így egy konzolablakban természetesen több szervert is adminisztrálhatunk.



MMC konzolon keresztül adminisztrálható a HIS 2000

MWI alapú telepítés

A Microsoft Windows Installer alkalmazásával az üzemeltetési és fenntartási költségek jelentős csökkentése érhető el, mivel a termékek telepítésére és konfigurálására hatékony módot kínál. Így a HIS 2000 esetében is használhatók a Group Policy Editor vagy a System Policy Editor kínálta lehetőségek, illetve az esetleges hibás telepítések automatikusan korrigálhatók, amely a rendszer rendelkezésre állása és az állásidők csökkentése szempontjából fontos.

- ☞ Megszakadt telepítésnél visszagargetés (rollback)
- ☞ A megsérült telepítések rugalmas helyreállítása
- ☞ Lehetőséget biztosít az adminisztrátornak, hogy meghatározza a MWI viselkedését a Group Policy Editor használatával a Microsoft Windows 2000 rendszereken, vagy a System Policy Editor segítségével a Windows 95, Windows 98 és Windows NT rendszereken.

SNA Server Windows 2000 környezetben

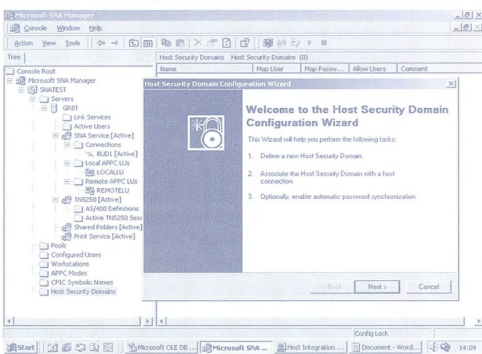
Az SNA Server 4.0 változata a service pack 3 után természetesen támogatja a Windows 2000 operációs rendszert, de sajnos ebben az esetben kihasználatlanok maradtak a Windows 2000 által biztosított alapszolgáltatások és az azokkal való szoros integrációból adódó előnyök.

Host Integration Server Windows NT 4.0 alatt?

A Host Integration Server 2000 teljes mértékben támogatott Windows NT 4.0-án. A Windows 2000 specifikus szolgáltatások nem lesznek elérhetőek. Ilyen Windows 2000 szolgáltatás például a COMTI által használt COM+ Object Pooling és az Active Directory kliens-szerver felhasználása.

Hálózatbiztonsági rendszer változásai

A Host Integration Server 2000 a RAFC, AFC/2 és Top Secret alkalmazásával egyirányú jelszószinkronizációt biztosít a mainframe számítógépek felé. A Host Integration Server 2000-nél nincs szükség harmadik fél által gyártott termékekre a Windows NT vagy Windows 2000 operációs rendszerekről RAFC-re, AFC/2-re és Top Secretre történő jelszószinkronizálás kezdeményezéséhez. A Host Integration Server 2000 a host PEM (Password expiration management) – komponensét használja a hoston történő változtatások elvégzéséhez. Ez a komponens része az host APPC/CICS-ének. Ez az új tulajdonság lehetővé teszi a Windows 2000 számára, Active Directory-val, vagy anélkül, hogy központi biztonsági adatbázisként működjön. A mainframe gépekkel való kétirányú jelszó szinkronizáláshoz továbbra is szükséges a kiegészítő Proginet vagy NEON Systems termékek használata.



A legtöbb komponens wizardok segítségével konfigurálható

MSDE-alapú host biztonság

A password szinkronizáláshoz és automatikus (Single Sign-On) bejelentkezéshez a felhasználóazonosító és a jelszó tárolására az SNA Server egyszerű fájlalapú adatbázist használ. A Host Integration Server 2000 ezt a fájl adatbázist egy MS Data Engine-en alapuló adatbázissal cseréli fel. Mivel az MSDE a Microsoft SQL Server technológiára épül, így ez a megoldás megbízhatóbbá, jobban támogathatóvá és skálázhatóbbá válik.

SNA gateway változások

Az SNA Server 4.0-val járó 3270 kliens desktopként egyetlen eseményt tesz lehetővé. A Host Integration Server 2000 több eseményre biztosít lehetőséget. Egy végfelhasználó így akár 16 eseményhez juthat a munkaszálton futó 3270 kliensből. Elméletben a felhasználó a helyi PC erőforrások számával egyező számú eseményt futtathatna, de 16 a tesztelt és megerősített szám.

Host Printing skálázhatóság

A Host Print Server funkció 1024 helyett 4096 nyomtatási esemény támogatására lett méretezve. Ez valószínűleg a legterheltebb host print környezet igényeit is meghaladja.

LU6.2 2PC terheléelosztás, hotbackup

A HIS 2000 alkalmazásával lehetővé válik, hogy egyes APPC alkalmazások esetén terheléelosztást és hotbackupot biztosítsunk több Host Integration Server 2000 kiszolgálón keresztül.

VPN támogatás

A kliensek távolról biztonságosan, akár az Interneten keresztül létesített „alagút” segítségével is elérhetik a host alkalmazásokat.

Adminisztráció

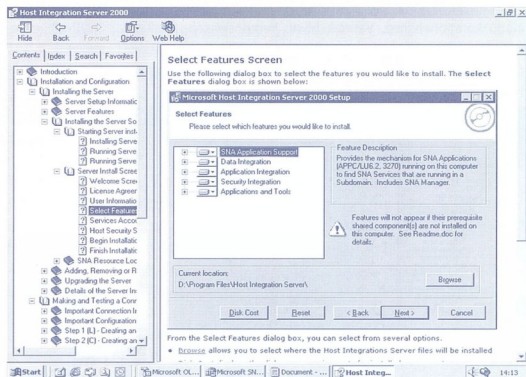
WMI alapú menedzsment és konfiguráció

A Host Integration Server 2000 az SNA adminisztrátorok számára további menedzsment lehetőségeket biztosít a Host Integration környezeti SNA gateway funkcióinak ellenőrzésére és szabályozására. A Windows Management Interface nézet a Windows platformot alkalmazva teszi vállalaton belül a szoftvert, a hardvert, a hálózatot, a felhasználót és a policy monitorozására, ellenőrzésére és elemzésére. A WMI/COM használatával egy automatizáló interfész biztosítja a felhasználó számára, hogy saját adminisztratív eszközeit létrehozza, melyeket monitorozáshoz, konfigurálásához és minden SNA adminisztratív objektumon végzett tevékenységhez használhat. A WMI a fejlesztőknek infrastruktúrát biztosít a közvetlenül erre az infrastruktúrára épülő menedzsmentalkalmazások létrehozására.

Online dokumentáció

A Host Integration Server 2000 telepítője és dokumentációja a funkcionális szempontokra összpontosítva teljes áttevzéses esett át. A termék minden egyes funkcionális területéhez (COMTI, replikáció stb) készült egy online könyv, így a vásárlók kiválaszthatják a számukra megfelelő MS Management konzolt. A termék telepítője is ezek köré a funkcionális területek köré szerveződik, így egyszerűbb csak a felhasználó igényeinek megfelelő komponenseket telepíteni. Az online könyv kiegészítői a multimédiás hibaelhárítók és további hibaelhárítási témák. A Host Integration Server 2000 mindegyik online könyv nyomtatható változatát is tartalmazza (Adobe Acrobat). A teljes dokumentumot megtekinthetjük a Start menűn keresztül elérhető HIServer.chm segítségével. A környezetérzékeny sűgő kapcsolódik a tartalomjegyzékhez, így a áttekinthetjük a vonatkozó témaköröket és bönegszhetünk bennük. A Host Integration Server 2000 az online könyvekben tárolt konfigurációs adatok bejegyzéséhez online munkalapokat is biztosít, valamint egy link-térképet, melynek segítségével az adminisztrátorok,

fejlesztők és adatbázis adminisztrátorok gyorsan megtalálják a feladatuk elvégzéséhez szükséges specifikus információkat.



A dokumentáció képekkel és mozgóképekkel segíti a felhasználókat

Adatelérés

A DB2 elérés kiemelkedő tulajdonságai.

A Host Integration Server 2000-ben mind a Microsoft OLE DB szolgáltató, mind az ODBC DB2 meghajtója újra lett tervezve, hogy megnövelt teljesítményt és skálázhatóságot biztosítson. Ez lehetővé teszi a fejlesztők számára, hogy dinamikus webalkalmazásokat készítsenek, amelyek valós időben érnek el nagyon fontos DB2 adatokat. A WMI felül a Windows teljesítményfigyelőjében (Performance Monitor), új számlálók segítségével az adminisztrátorok könnyen tervezhetnek és végezhetnek előzetes tesztek.

Adatintegráció

AS/400 adatsorok

Az adatsor (data queue) egy AS/400 rendszerobjektum, amely több program vagy feladat közötti belső kommunikációhoz használható. Az adatsorok segítségével lehetőség van arra, hogy több program közös üzeneteket küldjön vagy fogadjon egy központi helyet használva anélkül, hogy az üzenet adatait fizikai adatbázisba kellene írni. Általában amikor egy adatrekord kiolvasásra kerül, a rekord törlődik a sorból. Az adatsorok használatának előnye az adatbizs-fájlok használatával szemben az, hogy az adatsorok sokkal kevesebb input/output (I/O) műveletet igényelnek, ezáltal növelik az egész rendszer teljesítményét. A Host Integration Server 2000 segítségével a 32 bites Windows alapú alkalmazások adatsorokat tudnak elérni egy COM komponens használatával. A vezérlőkomponens az AS/400 kódjai szerint megtervezett, így gyakorlatilag a legtöbb opcióit lehet vele elérni.

nem állnak fenn. Kiegészítésképpen, a Host Integration Server 2000-hez egy natív 3270-es emulátorral történő kapcsolódáskor a következő kiegészítő funkciók érhetőek el a felhasználók részére:

- ☛ WindowsNT tartományi hitelesítés. A felhasználók nem csatlakozhatnak a host-hoz, amíg először nem történt meg a biztonságos azonosításuk egy WindowsNT domain által.
- ☛ A Logical Units (LUs) felhasználókhöz, felhasználói csoportokhoz történő jogosultságkiosztás. Az adminisztrátorok további biztonsági beállításokat eszközölhetnek, a Gateway szintjén, azzal, hogy bizonyos felhasználók/csoportok csak bizonyos LU-kat, LU pool-okat érhetnek el.
- ☛ Az LU-k „fix desktop”-hoz történő hozzárendelése. Az adminisztrátoroknak további biztonsági beállítási lehetőségeik vannak, ugyanis az egyes LU-khoz csak meghatározott munkaállomásokról lehet hozzáférni (*IP cím, vagy gépnév alapján*).
- ☛ Minden, a host-tal kapcsolatos adatforgalom titkosítása. Egy opcionális titkosítási tulajdonság révén az adminisztrátorok „kierőszakolhatják”, hogy az HIS 2000 és a desktop közötti adatforgalom titkosítva történjen.
- ☛ Automatikus bejelentkezés a hostalkalmazásba a Host Integration Server 2000 Single Sign-On tulajdonságának segítségével. Ha a 3270-es emuláció a jelszó-szinkronizálási tulajdonsággal van kombinálva (*ami természetesen a TN3270-nel is működik*), az adminisztrátorok beállíthatják a 3270-es felhasználók automatikus alkalmazásba történő bejelentkezését az adott alkalmazásra.
- ☛ Automatikus terhelés-elosztás.
- ☛ Automatikus „Hot Failover” a szerverrel lévő host kapcsolat hibája esetén.
- ☛ Képesség az IPX, NetBEUI, Banyan VINES alapú kliensek támogatására.

Teljesítmény

Kétfázisú commit műveletek

A Host Integration Server 2000-ben a Microsoft OLE DB provider for DB2 támogatja az elosztott munkaegységeket, a két-fázisú commit protokollon keresztül, a DB2 for OS/390-hez vagy DB2 for OS/400-hoz való csatlakozáskor, az SNA LU6.2 hálózaton. Ehhez a tulajdonsághoz szükséges az SNA LU6.2 Resync Service, valamint a Microsoft Distributed Transaction Coordinator szolgáltatások megléte. A TCP/IP-n keresztül DB2-be történő kétfázisú commit támogatása csak a Host Integration Server következő verzióiban kerülnek implementálásra. A kétfázisú commit-hoz nem szükséges az MTS.

Az ODBC Driver for DB2-n és az OLE DB Provider for DB2 változásai:

- ☛ Az ODBC Driver for DB2 alapjaitól fogva megújult, teljesen újraírták.
- ☛ Lényegesen gyorsabb adatbázis séma elérésénél, valamint a nagyméretű adatok mozgatójánál.
- ☛ Az adatkonverzió szintén továbbfejlesztésre került.

Megnövelt session szám

A HIS 2000 által kiszolgált maximális host session szám 30000-re emelkedett, amely egy HIS szerveret tekintve elég imponáns mennyiség.

AS/400 és a HIS 2000 kapcsolata

Jelenleg akár ha magyarországi példákat nézünk, akkor is sok ügyfél alkalmazza AS/400 környezetben a Microsoft SNA Servert hatékonyságnövelés és terhelésselosztás céljából. Ezek a lehetőségek a továbbiakban a Host Integration Server számára is piacot jelentenek – az alkalmazásintegrációs lehetőségeken túl – még abban az esetben is, ha a AS/400 környezetben is TCP/IP alapúvá válik a kommunikáció, hiszen a kliensek hatékonyan menedzselhetők, a kommunikáció tömörített és akár titkosított.

Erfőforrás igény:

- ☛ Pentium 90 processzor Windows NT 4.0 SP 6a operációs rendszer vagy Pentium133 processzor Windows 2000 SP1 operációs rendszer
 - ☛ 32 MB memória Windows NT 4.0 esetében, illetve legalább 128 MB Windows 2000 esetében
 - ☛ 120-150 Megabyte szabad lemezkapacitás
- A Microsoft a Host Integration Server 2000 fejlesztésével a kliensek és a különböző Hostok közötti kapcsolatok menedzselésétől elmozdult az alkalmazásintegrációs szolgáltatások felé a nagygyépes környezet esetében is. A változatoknak, az alkalmazásoknak, a webfejlesztőknek a piac rezdüléseire gyorsan kell reagálniuk és ez megfelelő eszközök és technológia nélkül rendkívül költséges. A gyors fejlesztés és az alkalmazások közötti együttműködés nélkülözhetetlen. Hostoldali üzleti alkalmazásokba sokat investáló cég számára költséghatékony és technológiailag is előremutató megoldás a HIS 2000 alkalmazás integrációs lehetőségeinek kihasználása.

Lóki Róbert, robert.loki@synergon.hu



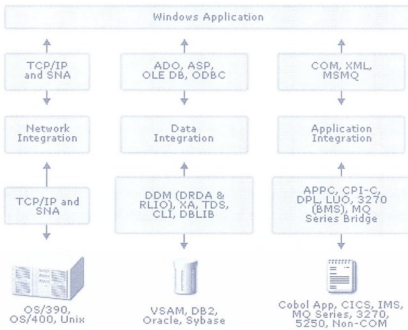
A Microsoft Host Integration Server 2000 új lehetőségei

A Microsoft Host Integration Server (HIS) 2000 az SNA Server utódja, emiatt a korábbi SNA Server teljes funkcionalitását nyújtja, kiegészítve azokat számos új lehetőséggel. Használatával egy vállalat meglévő mainframe (pl. IBM MVS) és/vagy AS/400 rendszerei és Windows alapú hálózata között lehet kapcsolatot teremteni. Ily módon lehetővé válik a mainframe-ek adatainak elérése egyrészt a vállalati intranetről, vagy adatok egyszerű módon történő publikálása az Internetre.

Az SNA Server 4.0-hoz képest egyik legjelentősebb különbség, hogy a HIS 2000-et Windows 2000-re optimalizálták. Ez nemcsak azt jelenti, hogy a legjobb teljesítmény csak Windows 2000 alatt hozható ki belőle, hanem, hogy vannak olyan funkciók, melyek Windows 2000 specifikusak, azokat NT-n nem lehet kihasználni. Ilyen funkciók a COM+ Object Pooling használata COMTI-n (COM Transaction Integrator) keresztül, vagy az Active Directory kliens/szerver módú használata.

A Host Integration Server szolgáltatásai

A HIS szolgáltatások felhasználási jellegüktől függően három kategóriába vannak sorolva:

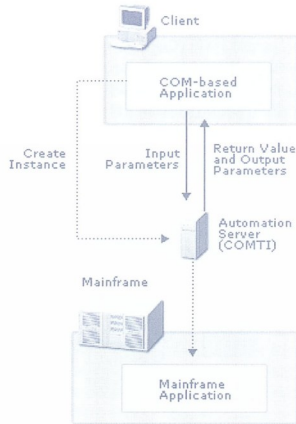


A HIS szerver működése három fő területre csoportosul

- ☞ **Alkalmazásintegrációs szolgáltatások:** a különböző rendszerek közötti alkalmazásszintű kapcsolat megoldása ezen keresztül történik.
- ☞ **Adatbázisintegrációs szolgáltatások:** az HIS által kiszolgált rendszerek adatbázisaihoz biztosít többszintű elérési lehetőséget.
- ☞ **Hálózatintegrációs szolgáltatások:** különböző hálózatok között teremti meg a kapcsolatot – a mainframe vagy AS/400 hálózat felé mintegy gateway-ként funkcionál a windows gépek felé, így azoknak nem szükséges a háttérrendszerhez való kapcsolódáshoz külön szoftvert használniuk. A hálózati integrációt biztosító egyes komponensekre csak SNA protokoll használata esetén van szükség – így, ha a HIS és a mainframe/AS400 közötti kommunikáció TCP/IP alapú, akkor a telepített szerver kevesebb helyet foglal.

Alkalmazásintegráció – a COMTI újdonságai
A COMTI segítségével az IBM Customer Information Control System (CICS) és Information Management System (IMS) elérése van lehetőség. A COMTI Automation Server komponensek regisztrálhatóak bármely Windows 9x, Windows NT, vagy Windows 2000 rendszeren, illetve elvben bármely DCOM-ot ismerő operációs rendszerben.

Az egyik komoly újdonság a COMTI (COM Transaction Integrator) integrációja COM+ komponenseken vagy MTS csomagokon keresztül. A COMTI nem COM komponens, hanem egy type library. Egy COM+ alkalmazásban történő felhasználásával az alkalmazás számára biztosítja a hostrendszer elérését: létrejön egy COM objektum (= COMTI Automation server). Ezen keresztül történne a különböző architektúrájú gépek közötti konverzió, és vezérli a host TP-keket (Transaction Program) a kommunikációt. Mindez a COMTI run-time environment segítségével történik, amit a Windows automatikusan elindít, amikor egy COMTI komponens is használó COM+ alkalmazás elindul. Tulajdonképpen ez felelős a COMTI működéséért: fogadja a COMTI komponensekhez érkező hívásokat, elvégzi a paraméterek az adott host felé szükséges konverzióit és egyéb formátummódosításokat, végül elvégzi és fogadja a host TP-k hívását, illetve a hívás eredményét visszaadja a hívó félnek feldolgozásra. A COMTI által biztosított felületen a többi alkalmazás így egyszerűen képes a hosttal kommunikálni.



COMTI alkalmazása host és Windows alkalmazás közti kapcsolat megteremtésére

A COMTI gyakorlatilag minden egyes metódushoz egy TP-t rendel a hoston. Amikor egy alkalmazás meghívja a COMTI komponens egy metódusát, akkor a HIS szerver a hoston elindítja a metódushoz rendelt TP-t (azaz *egyed-egy TP-hez akár több metódus is tartozhat*).

A hatékony hibakezelés érdekében a COMTI definiál egyfajta metadatum mechanizmust, melynek használatával – ha fel lett rál készítve – a hoston futó program hiba esetén egy exception blockon keresztül hibaizenetet tud küldeni a HIS felé. Az így küldött hibaizenetet a HIS-en egy exception előállítására alkalmas, amit a kliensalkalmazásnak rendezsen le kell tudnia kezelni.

A COMTI támogatja a COM+ Object Pooling mechanizmusát: ekkor az objektumok egy cache-ben tárolódnak a kliensalkalmazások hívásai között, ezzel is csökkentve az objektumok létrehozására és eltávolítására fordított erőforrásigényt.

További újdonság a COMTI-val kapcsolatban, hogy az IBM Open Transaction Management Architecture (OTMA) szervert alkalmazásával a hoston a COMTI-val való TCP alapú kommunikáció megvalósításához nem kell újrafordítani az IMS TP-eket, mint ahogyan azt korábban kellett.

A hosttal való kapcsolat továbbra is request-reponse alapú, a „párbeszéd” jellegű kommunikációs modelleket a COMTI egyelőre nem támogatja.

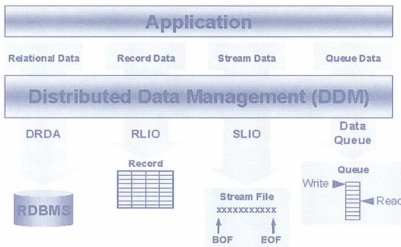
Egy COMTI komponens létrehozása a COMTI Component Builder segítségével történik, míg a COMTI adminisztráció a COMTI Manager segítségével végezhető el.

Adatintegráció

Az adatelérési módokra többféle lehetőséget biztosít a HI szerver:

- ☞ Relációs adatbázis elérés
- ☞ Fájlok rekordszintű elérése
- ☞ AS/400 data queue-k használata

Ezen felül létezik még fájltranszfer lehetőség. A különböző kódkonverziók a host CCSID (Coded Character Set Identifier) és az alapértelmezett ANSI kódlap alapján történnek. Mindegyik adatelérési mód az IBM Distributed Data Management Architecture (DDM) alapján működik, ami pl. az AS/400-as rendszereken, de más IBM rendszereken is általánosan használt.

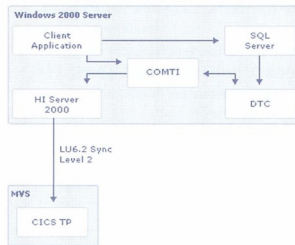


A HIS adatelérési lehetőségei

Az AS/400 és VSAM (Virtual Storage Access Method) OLE DB Provider segítségével lehetőség van az AS/400 vagy más rendszerek fájljainak rekord szinten történő elérésére. Ha figyelembe vesszük, hogy az OS/400-ban operációs rendszer szinten egy DB2 relációs adatbáziskezelő rendszer működik, és a fájlok kezelése az AS/400-on ilyen módon történik, akkor ebben nincsen semmi meglepő. Gyakori igény, hogy

egy (pl. webes) alkalmazás egy hoston levő fájlból olvasson, vagy akár írjon is oda. Az OLE DB Provider használatával ezt egyszerűen lehet megvalósítani, APPC programozás nélkül.

A DB2 adatbázisokkal történő jobb integráció érdekében a DB2 OLE DB Provider és ODBC driverek újra lettek írva. Az új driverek a korábbiaknál hatékonyabban támogatják a DB2 adatbázisok elérését, és a COMTI segítségével támogatják a two-phase commit (2PC) protokollt. Ennek segítségével a COMTI-t alkalmazó COM+ alkalmazások számára lehetőség van pl. a host DB2 és egy MS SQL szerver együttes használatával, MS DTC segítségével tranzakció-alapú működés megvalósítására, tehát az egy tranzakción belüli műveletek egy lépésben való jóváhagyására (commit) vagy visszautasítására (rollback). Ez egy több adatbázisból dolgozó alkalmazás számára igen fontos és hatékony lehetőség az adatok konzisztenciájának fenntartására. Az új driverek használatával ez a feladat jóval egyszerűbben oldható meg, mint korábban.



2PC művelet DTC és COMTI segítségével

Aki dolgozott már AS/400-on, az tudja, hogy milyen jelentősége van az ún. data queue-k használatának. A data queue-k egyfajta kommunikációs lehetőséget biztosítanak az egyes programok között (IPC, Inter Process Communication) fizikai fájl használatával nélkül. Ezek használata ilyen környezetben teljesen alapvető, éppolyan fontos, mint pl. UN*X rendszereken a half-duplex pipe-ok használata. A Host Integration Server egy COM automation control használatával lehetőséget biztosít az ilyen data queue-k elérésére, ezzel jelentősen megkönnyítve a rendszerek közötti adatcserét.

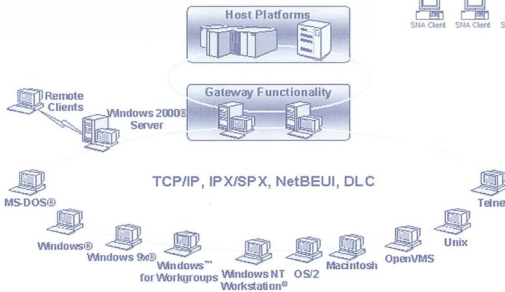
További fontos beállítási lehetőség a DB2 OLE DB Provider és ODBC driver-ek esetében a Connection Pooling lehetősége. Ennek segítségével és megfelelő beállításával a megadott timeout-on belül az egyszer már létrehozott kapcsolat a DB2 adatbázis felé nem zárul le. Ha egy újabb kapcsolatot kezdeményez az alkalmazás a host DB2 felé, akkor azt nem kell újra felépíteni, hanem a fenntartott kapcsolatot átadva az alkalmazásnak jelentős teljesítménynövekedést lehet elérni a kiszolgálásban. A létrehozott kapcsolat mindaddig nyitva marad, amíg a beállított timeout-on belül valamelyik alkalmazás használni akarja. Ez igen jól használható funkció pl. egy webalkalmazás esetén, amikor pl. egy ISAPI modul számára az adatokat a DB2 adatbázisból kell lekérdezni.



Hálózati integráció

A hálózati integrációt megvalósító szolgáltatások feladata, hogy HIS kliensei és az SNA hálózat közötti kapcsolat létrejöhessen. Ekkor a HIS mintegy hálózati gateway funkcionál a pl. TCP/IP alapú hálózatban levő rendszerek és az SNA hálózat között.

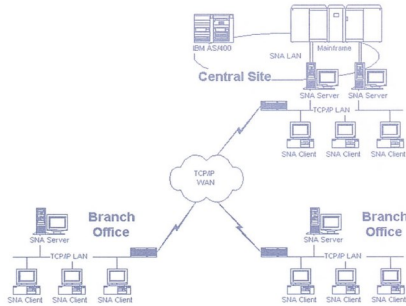
A hálózati szolgáltatások ezen a szinten jórészt az SNA Server 4.0 funkcióira épülnek, azok némi javításával. Ilyen javítás például, hogy a HIS mintegy hálózati gateway használata esetén a 3270 típusú terminálemulációk használatakor a kliensen a korábbi egyhez képest immár akár 16 session is indítható (sőt, elvileg akárennyi).



A HI szerver mint SNA gateway hálózatban

Az SNA gateway mint szerver szolgáltatás három komponens köré épül:

- **SNASERVER:** a protokollkonverziót vezérlő rész, SNA PU 2 (hierarchikus) vagy 2.1 LEN (peer-to-peer) node emulációt biztosít
 - **SNABASE:** ennek feladata az SNA hálózatban jelen levő szerverek, LU-k, stb. folyamatos nyilvántartása
 - **SNADMOD:** egy RPC-t használó kommunikációs modul, melyen keresztül a szerver a kliensekkel kommunikál
- A Host Integration Server alkalmazásával különböző telepítési modellek alapján alakítható ki a hálózat:
- **központosított (centralized) modell,** amikor a HI szerver fizikailag ugyanott van, mint a host rendszer, és az összes kliens a központi HI szerverrel van kapcsolatban
 - **leágaztatott (branch) modell,** ekkor az egyes telephelyeken vannak telepítve a HI szerverek, a kliensek lokálisan ehhez kapcsolódnak, és a hálózaton keresztül csak az egyes HI szerverek kommunikálnak a központi host-al
 - **elosztott (distributed) modell,** ami az előző kettő összerakva: mind a központban, mind a telephelyeken működnek a HI szerverek, és a telephelyekről az ottani szerverek a központi HI szerveren keresztül kapcsolódnak a host-hoz.



Az elosztott telepítési modell

A kliensek képesek bármely, a hálózaton az általuk ismert SNA subdomainben (HI szerverek és/vagy SNA Szerverek logikai összetartozása egy csoportban) aktívan működő Host Integration Server szolgáltatásait igénybe venni. Ez lehetőséget ad a HI szervereken load balancing alkalmazására, valamint hibátörést is biztosít: ha az egyik szerver üzemén kívül van, a kliens bármely más szerveren keresztül kapcsolódhat a host rendszerhez.

Megjegyzendő, hogy ez azért nem jelent igazi elosztást: a kliens az ismert szerverek közül véletlenszerűen választ, és nem a terhelés alapján dől el, hogy a kapcsolat melyik szerver használatával jöjjön létre.

Újdonság, hogy a HI szerver a hoston a Password Expiration Management (PEM) komponenst használja, ami az APPC/CICS része. Így egy Windows 2000 szerver az Active Directory használatával vagy anélkül is képes központi biztonsági adatbázisként működni. Megjegyzendő, hogy a mainframe hosttal történő kétirányú jelszó-szinkronizációhoz kiegészítő szoftverekre van szükség, de ezeket be lehet szerezni.

A jelszavak tárolásához, karbantartásához a HI szerver a Microsoft Data Engine-t használja. Windows 2000 telepítés esetén a Host Integration Server használja az Active Directory-t, így a kliensek a szervereket azon keresztül is képesek elérni, akárcsak egyes, a Host Integration Server által nyújtott erőforrásokat (pl. LU-k). Az egyes HIS-ek az Active Directory segítségével is regisztrálhatják magukat egymásnak, de még a klienseknek sincs szükségük direkt kapcsolatra a HIS ahhoz, hogy egy általa regisztrált erőforrást az Active Directory-ban megtaláljanak (persze annak használatához már kell a HIS, ha a hostot el akarja érni az alkalmazás).

A jelszavak tárolásához, karbantartásához a HI szerver a Microsoft Data Engine-t használja. Windows 2000 telepítés esetén a Host Integration Server használja az Active Directory-t, így a kliensek a szervereket azon keresztül is képesek elérni, akárcsak egyes, a Host Integration Server által nyújtott erőforrásokat (pl. LU-k). Az egyes HIS-ek az Active Directory segítségével is regisztrálhatják magukat egymásnak, de még a klienseknek sincs szükségük direkt kapcsolatra a HIS ahhoz, hogy egy általa regisztrált erőforrást az Active Directory-ban megtaláljanak (persze annak használatához már kell a HIS, ha a hostot el akarja érni az alkalmazás).

Kónya Boldizsár, konya.boldizsár@hms.hu





A Commerce Server 2000 a Microsoft legújabb e-commerce rendszere, mely történetileg tekintve a Site Server család utódja, valójában jóval több annál: új megoldásokat kínál az új problémákra. A Commerce Server 2000-et a jövő e-commerce webszerverének szánták és az általa felkínált megoldások szorosan kapcsolódnak a Windows 2000 platform új generációs termékeihez.

Jelen cikkben először áttekintést nyújtunk a termék fontosabb tulajdonságairól, felhasználási területeiről majd egy egyszerű kereskedelmi site funkcionalitásán végighaladva, gyakorlati példával igyekszünk bemutatni az egyes funkciók felhasználási lehetőségét.

Tulajdonságok

A termék számos olyan új tulajdonsággal és szolgáltatással rendelkezik, amely eddig nem volt megtalálható egyetlen Microsoft termékben sem, és számos olyanval, amely elődjének, a Site Server Commerce Edition-nek része volt. Utóbbi funkciók megvalósítási módjai azonban újragondolásra kerültek, ezáltal korszerűbb megoldások születtek.

A termék az alábbi funkciócsoportokat valósítja meg:

- ☞ **Profile System.** Az e-commerce folyamat összes szereplője, a honlap regisztrált és nem regisztrált felhasználói, a partnercégek vagy ezek csoportjai, vagyis az összes üzleti entitás rendelkezhet testre szabott profillal, mely felhasználható egyedi oldalak megjelenítésére.
- ☞ **Product Catalog System.** A rendszer használatával kialakíthatjuk webes áruházunk termékinventárát katalógusokba szervezve és a termékekhez könnyen rendelkezünk árral. Átjárás biztosít külső források felé, lehetőség nyílik a katalógusok más rendszerekből történő feltöltésére.
- ☞ **Targeting System.** Segítségével egyszerűen fogalmazhatunk meg szabályokat web- és komponensalapú alkalmazások számára, melyek a tartalom megjelenítését és menedzselését végzik. Segítségével felhasználókra vagy cégekre vonatkoztatott egyedi termékkatalógusok és árak kialakítása válik lehetővé. Akciókat hozhatunk létre és a célcsoportok számára direkt email küldést valósíthatunk meg.
- ☞ **Business Processing Pipelines System.** Az üzleti folyamatokat szabályzó és vezérlő rendszer segítségével a rendelés egyes stádiumait könnyen kapcsolhatjuk össze. Szabályokat fogalmazhatunk meg egy folyamat lépéseit illetően, vagy például figyelemmel kísérhetjük egy adott rendelés állapotát.
- ☞ **Business Analytics System.** Naprakész információkat nyerhetünk a honlap látogatottságáról, az eladási statisztikákról, vásárlói szokásokról. Grafikonok segítségével könnyen átláthatjuk a rendelkezésre álló statisztikai adatokat. Ezen túlmenően lehetőség van trendek figyelésére, előrejelzések készítésére az analízis adatok alapján.
- ☞ **Administration.** A termék minden szempontot kielégítő adminisztrációs felületekkel rendelkezik, mind a rendszert fejlesztő és üzemeltető számítástechnikai

szakemberek, mind az üzleti folyamatokban jártas menedzserek számára. Az adott területért felelős munkatárs egyszerűen tudja szabályozni a hatáskörébe tartozó folyamatokat és munkáját nagymértékben megkönnyítő eszközök állnak rendelkezésére.

Nézzük meg részletesebben a fentebb felsorolt témaköröket.

Profile System

Egy webes áruház esetén nemcsak anonim látogatókról vagy regisztrált vásárlókról van szó, hanem partnercégekről, kis- és nagykereskedőkről és ezek csoportjairól is nyilván kell tartanunk adatokat. Ezeknek az adatlapoknak a szerkezete más és más lehet, ahogy az alábbi példák is szemléltetik:

Regisztrált felhasználó adatlapja	
Vezetéknév:	Kovács
Keresztnév:	Huba
Életkor:	25
Cím:	Érd, 1299 Ács u. 19/B
Email cím:	khuba@mail.hu
Hobbi:	Nyelvtanulás, utazgatás
Vizsonteladói adatlapja	
Cégnév:	Gigastore Ltd.
Cím:	Mohács, 1526 Vész u. 13.
Telefon:	64-555-8695
Forgalom:	152154 Ft

Az adatlapokon szereplő mezők tetszőlegesen bővíthetők és illeszthetők meglévő forrásokból. Ha például partnereink adatainak egy része már rendelkezésre áll egy adatbázisban, akkor ezt direkt módon hozzá tudjuk csatolni a webes áruház számára kialakított céges adatlapjához. Nem kell tehát átalakítanunk, migrálnunk a meglévő adatokat ahhoz, hogy felhasználhassuk.

Az egyes profilok esetén a Commerce Server kétfajta adatlapot tud kezelni. Az egyik a felhasználó vagy partner által közvetlenül kitöltött (*explicit profiling*) adatok összessége, amelyet például egy vásárlás esetén ki kell töltenie magáról. A másik a közvetett adatokon alapul (*implicit profiling*), mint például az anonim látogatók által leggyakrabban látogatott oldal, legkeresettebb termék vagy a legsikeresebb nyári akció.

A Commerce Server 2000 három nagy területre osztja a profiling rendszert:

- ☞ **Felhasználók adatai.** Jól skálázható felhasználói adatbázis, mely a pár száz regisztrált látogatótól akár a milliós nagyságrendig terjedhet.
- ☞ **Csoportok adatai.** Csoportba szervezett felhasználók, akik különböző lehetőségekkel és tulajdonságokkal rendelkeznek.
- ☞ **Profilmenedzsment.** Az egyes profilok struktúrájának felépítése és meghatározása.

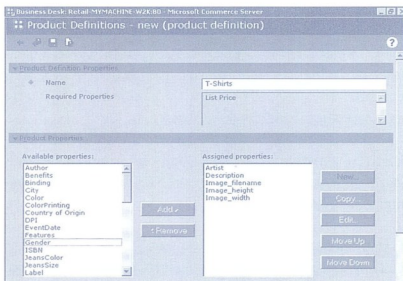
Mindhárom területen grafikus felület áll a rendelkezésünkre ahhoz, hogy egyszerűen és könnyen el tudjuk végezni a szükséges lépéseket.



Fizikailag a fent említett adatok tárolására a Commerce Server 2000 két lehetőséget ajánl fel. Egyrészt Microsoft SQL 7.0 vagy 2000 adatbázisban tárolhatjuk a profilokat, másrészt a Windows 2000 platform Active Directory technológiáját használjuk fel így biztosítva a nagy teljesítményt és skálázhatóságot.

Product Catalog System

Egy webes áruházban vásárolható termékek megfelelő csoportosítása, katalógusok kialakítása és a hozzájuk tartozó árak meghatározása elsődleges feladat. A Commerce Server 2000 segítségével a termékkatalógusok összeállítása jelentősen leegyszerűsödik. Az egyes termékekre vonatkozóan szükséges adatokat vehetünk fel.



Termékkatalógus összeállítás

Ehhez nincs szükség adatbázisvezetésre, tárolt eljárások frázására, mindez helyettünk elvégzi a rendszer.

Az árak vagy termékek feltöltéséhez és folyamatos frítéséhez külső forrásokhoz is csatlakozhatunk. A rendszer kapcsolat teremthet a Microsoft BizTalk Serverrel, melynek segítségével XML alapú katalógusok importálhatók vagy exportálhatók.

Targeting System

Ha feltöltöttük a webáruházat termékkel és vannak felhasználók, ajánlhatunk nekik testre szabottan termékkatalógust érdeklődési területük, vagy látogatási szokásaik alapján. Függetlenül attól, hogy a felhasználókat milyen hierarchiába, csoportokba rendeztük, külön listákat állíthatunk össze, például a tíz legnagyobb összegben, vagy legtöbbször vásárló felhasználót kereshetjük ki. Ezen listák számára aztán direkt leveleket küldhetünk.

Business Processing Pipelines System

Az üzleti folyamatok szabályainak leírása és állapotainak nyomon követésére a rendszer egyedülálló lehetőségeket biztosít. A pipeline technológia segítségével definiálhatjuk és összekapcsolhatjuk a rendelés egyes állapotait. Meghatározhatjuk hogy ezekben az állapotokban milyen szabályokkal, milyen feltételek teljesülése esetén kerülhet a rendelés. Az állapotok leírása egy állapoton belül több mint 80 elemet használhatunk fel, melyek segítségével tetszőlegesen

bonyolult folyamatokat is definiálhatunk. Természetesen mindehhez grafikus felület áll rendelkezésünkre.

A Commerce Server 2000 rendelkezik előre összeállított folyamatokkal, mind az egyes termék vásárlásával, rendelés feladásával és teljesítésükre, mind direkt levelek küldésével vagy hirdetések összeállításával kapcsolatban.

Business Analytics System

Egy működő e-commerce rendszer rengeteg adattal és információval szolgál, melyek elemzése és felhasználása az üzleti döntésekben nagy jelentőségű. A Commerce Server 2000 nemcsak a webáruházat látogatók vásárlási szokásainak vizsgálatában, statisztikai adatok minőség megjelölésében van segítségünk, hanem a trendek figyélésében, előrejelzések készítésében.

Az analízisnél a Commerce Server 2000 felhasználja a Microsoft SQL 2000 Server adattanszformációs (DTS) és OLAP szolgáltatásait, így lehetőség van üzleti adattárházak kialakítására is. A DTS segítségével lehetőség van:

- ☞ összegyűjteni a webkiszolgáló által készített napló-állományokat
- ☞ IP cím alapján és név szerint azonosítani a webáruház klienseit
- ☞ előfeldolgozott és statisztikai adatokat tárolni Microsoft SQL Serveren
- ☞ a kisebb tárgyú érdeklődésben törölni az előző analízis feleslegessé vált adatait

Az üzleti adatelemzés az OLAP Server segítségével történhet, amely előkészíti az adatokat az üzleti elemzés számára. Az információt sokdimenziós adatkockákban tárolja, ezért gyorsabban tud olyan összetett riportokat készíteni, amelyeknek az összeállítása hagyományos SQL esetén lassú lenne. Például egyszerűen lekérdezhettük azoknak a vásárlóknak az életkorát, akik idén először látogattak meg az áruházat, és 5.000 forint felett vásároltak. Az adatkockákban az információk két nagy csoportra oszthatóak:

- ☞ **Mértékegység.** Annak a paraméternek a neve, amiről információt szeretnénk megtudni, például méret, terméknév, szín, ár.
- ☞ **Mérvszám.** A statisztikai adatok alapján meghatározott érték.

E két adat relációja adja azokat az összefüggéseket, amelyek kifejezik a vásárlók szokásait.

Az elkészült riportok két nagy csoportba oszthatóak. Egyrészt készíthetünk *dinamikus riportot*, amely minden egyes alkalommal az épp aktuális adatokat dolgozza fel és a legfrissebb eredményeket adja. Másrészt használhatunk *statisztikus riportot*, melyek egy adott napszakban, például hajnalban generálódtak. Ezek alapján állíthatunk össze levelezési listákat.

A riportok formátuma sem szigorúan kötött. Választhatunk HTML oldal, Word dokumentum vagy Excel táblázat között, aszerint, hogy milyen felhasználási területre szánjuk az elkészített analízist.

A riportok megjelenítésére grafikus felület is rendelkezésre áll,

erre a célra használhatóak például az Office webkomponensei. Természetesen a riportok készítésében is segítségünkre van a Commerce Server 2000. Előre összeállított riportok tucatjai felhasználásával az üzleti döntések előkészítéséhez szükséges anyagokat jeleníthetünk meg: lemérhetjük egy reklámkampány sikerességét, kiválaszthatjuk a legjobban, vagy épp a legkevésbé keresett termékek listáját, feltérképezhetjük, milyen vásárlóréteget kell ösztönözni kedvezményekkel a további vásárlásokra.

Mint már a profiloknál említettük a Commerce Server 2000 rendelkezik úgynevezett *implicit profiling* lehetőséggel. Ezt összekapcsolhatjuk az analízissel: például ha a felhasználó nem töltötte ki az életkor mezőt, az analízis a többi vásárló adatai és az illető személy vásárlási szokásai alapján megjósolhatja az adott személy korát.

A rendszer lehetőséget ad *szegmensmodell használatára*, vagyis az egymáshoz közelítő tulajdonságokkal rendelkező vásárlókat szegmensbe tudjuk sorolni.

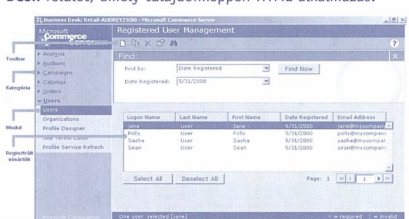
Egy másik érdekes felhasználási lehetősége az Commerce Server 2000 Business Analytics rendszerének az ún. *cross-sell statisztika*. Egy vásárlás során a felhasználónak ötleteket adhatunk, hogy a hozzá hasonló vásárlók még mit szoktak a kosarukba helyezni, ezáltal ösztönözve őt további vásárlásokra.

Adminisztráció

Mint láthattuk, a felhasználó számára egyszerűen és könnyen használható webes áruház a háttérben igen összetett felépítésű. Másrészt egy áruház üzemeltetése és fejlesztése más és más ismereteket igényel. Ez hasonlít egy valódi áruházhoz: az eladónak, raktárosnak és üzemeltetőnek más ismeretekkel kell rendelkeznie a különböző feladatok ellátására. A Commerce Server 2000 három szerepre és felelősségre osztja a rendszer szereplőit:

- ✎ **Honlapfejlesztő:** Ő építi fel és fejlesztja a webes áruházat, migrálja a meglévő adatokat az adatbázisába, új modulokat illeszt a rendszerbe
- ✎ **Rendszergazda:** Az operációs rendszer működéséért felelős, az áruház rendszerkezeli folyamatait felügyeli
- ✎ **Menedzser:** A fejlesztés irányait határozza meg, termék-katalógusokat állít össze, reklámkampányokat indít, karbantartja a beszállítók adatait, analízálja a vásárlói szokásokat és a kampányok hatását

Ez utóbbi szerepök rendelkezésére áll az ún. **Business Desk** felület, amely tulajdonképpen HTML alkalmazás.



Business Desk modul segítségével minden üzleti beállítás könnyen elérhető

A Business Desk tiszta, jól kezelhető felületet ad az összes szükséges terület irányítására:

- ✎ **Analysis.** A már említett analízismodul, minden igényt kielégítő lehetőségekkel.
- ✎ **Campaigns.** reklámkampányok szervezése, indítása és leállítása, célcsoportok meghatározása, direkt email küldés.
- ✎ **Catalogs.** Teljes termékaletta adminisztrációja, külső árlisták, katalógusok importja.
- ✎ **Orders.** Rendelések listája, keresés, állapotok megtekintése.
- ✎ **Users.** Regisztrált vásárlók és partnerek adatainak menedzselése, profilok beállításai.

Az egyes kategóriák további modulokra osztottak. Ezeken belüli használatot sűgő és toolbar is segíti.

A Commerce Server 2000 alkotói természetesen a fejlesztőkről és a rendszeradminisztrátorokról sem feledkeztek meg. Számukra a már megszokott Management Console-ba illeszthető adminisztrációs modulok alakítottak ki, mely minden részletre kiterjedően biztosítja a rendszer felügyeletét és a szükséges beállítások kezelését.

Az elkészített webáruházat egyedülálló módon telepítő-készlettel alakíthatjuk a *Site Packager* segítségével, így más kiszolgálókra is könnyen feltelepíthetjük a bevált e-commerce megoldást.

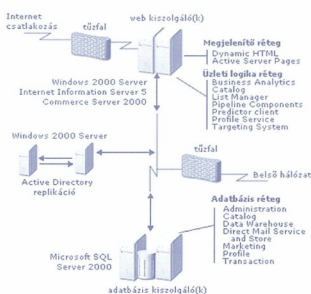
E-commerce a gyakorlatban

Most, hogy átrágtuk magukat a funkciók és leírások tömegén, nézzünk egy kézzelfogható példát arra, hogyan építhetünk fel webáruházat a Commerce Server 2000 segítségével.

Tételezzük fel, hogy megfelelő borkultúrával és üzleti érzékkel rendelkezve, magyar borokat árusító webáruházat szeretnénk létrehozni. Milyen lépések szükségesek ehhez?

Infrastruktúra

Ne felejtünk el, hogy először a rendszer infrastruktúráját kell meg terveznünk. Tételezzük fel, hogy fizikailag saját cégünknel kívánjuk elhelyezni a webáruház kiszolgáló számítógépeit. Mivel internetre csatlakozunk, ezért a rendszert két tűzfal közötti ún. demilitarizált zónában helyezük el. A webkiszolgálót, az adatbázis kiszolgálót, a regisztrált vásárlókat és beszállítókat külön szerveren helyezük el.



Infrastruktúra vázlat



A rendszer alapjait Windows 2000 kiszolgálókra helyezzük. Mint látható, alkalmazászinten háromrétegű architektúrát valósítottunk meg, igaz, fizikailag a megjelenítés és az üzleti logika szint egy kiszolgálón található. Ez a megvalósítás inkább a kis és közepes áruházak esetén alkalmazható. Nagy forgalmú e-commerce megoldások esetén általában az adatbázisréteg egyes elemeit külön kiszolgálókra helyezik el, ezáltal növelve a teljesítményt.

Profilok

Mivel jó minőséget akarunk biztosítani, ezért közvetlenül a pincészetektől vásárlunk. Először elkészítjük a beszállítókat adatait, azokat az adatokat, amiket rögzíteni szeretnénk róluk, majd feltöltjük az adatbázist:

Beszállító	
Cégnév	Fine Wine
Város	Eger
Irányítószám	9898
Utca	Kossuth utca 34.
Telefon	64-555-8695
Borvidék	egri
Forgalom	

Természetesen az adatbázisban borvidékek szerint is csoportosítjuk a pincészeteket, és az úrlapon hagyunk helyet a forgalomra vonatkozó adatoknak is, amelyeket később töltünk fel. Ezután kialakítjuk a vásárlók profilját, de ennek feltöltése majd a látogatók dolga lesz, illetve a statisztikai adatokat majd a vásárlás alapján töltjük fel:

Vásárló Vezetékvé	
Keresztnév	
Város	
Irányítószám	
Utca	
Telefon	
Vásárlások száma	
Vásárolt összeg	

Termékkatalógus

Kialakítjuk egy adott bor paramétereit, amely alapján egyrészt azonosítható, másrészt a vásárlók számára szolgál információval:

Termék	
Címke	Tekenőhati Cabernet Savignon
Típus	száraz
Kategória	vörösbor
Pincészet	Fine Wine
Úrtartalom	0,75 liter
Ár	5500 Ft

Az így kapott termékkatalógust pincészet, évjárat vagy borvidék szerint is rendezni lehet. Ezt azonban nem kell nekünk elvégezni, a rendszer megteszi helyettünk.

A termékkatalógus alapján a rendszer létrehoz egy XML dokumentumot, melyet BizTalk Server-hez használhatunk föl. Ez tartalmazza ugyanis az ügyfeleinkkel, beszállítóinkkal kapcsolatos üzleti kommunikáció módját, vagyis azt, hogy milyen úrlap kitöltésével rendelünk meg tőlük terméket, és nekik egy szállítás során milyen úrlapot kell kitölteniük. Ezáltal mindkét oldalon automatizálható a folyamat, hiszen egy szállítmány beérkezésekor a Commerce Server 2000 képes feldolgozni az XML dokumentumot és frissíteni a termékkatalógust, vagy éppen az árlistát. Az BizTalk Servert ismerők tudják, hogy ez a technológia jóval több az említett példánál. A jövő minden e-commerce megoldásának alapelemét fogja képezni.

Pipeline

Minden vásárlónak van egy „virtuális” kosara, amibe bele rakja a termékeket. A vásárlás végén pedig megrendeli a kosár tartalmát. Ez így elmondva elég egyszerűnek tűnik, de ne felejtjük el, hogy a folyamat közben rengeteg ellenőrzési pontot kell beiktatnunk. Például csak az a vásárló rendelhet, aki regisztráltatta magát és csak olyan bort vehet, ami még van raktáron, stb. Ezeknek a folyamatoknak a szabályozására a Bussiness Processing Pipelines rendszer használjuk fel, pontosan meghatározva az egyes folyamatok lépéseit.

Analízis és testreszabott áruház

Áruházunk szép forgalmat bonyolít le, de a kereslet fokozásának érdekében szeretnénk elemezni a vásárlói szokásokat és szeretnénk, ha a látogatók teste szabott weboldalakat látnának viszont. Lássunk pár ötletet:

Először is megnézzük mely pincészetek mely borai fogynak a legkevésbé. Ezeket összehasonlítjuk az azonos kategóriájú borok fogyási statisztikáival, így megállapítjuk, hogy a minőséggel, vagy az árral lehet gond:

A honlapon legkevésbé vásárlók számára akciós csomagokat állítunk össze, amelyről direkt emailt kapnak.

Igényesebb vásárlóink részére különleges borokról külön információs oldalt állítunk össze. Ha az általuk legtöbbet vásárolt kategóriában új termék kerül a kínálatba, erről külön értesítjük őket.

Bizonyos összeg feletti vásárlás esetén törzsvásárlói kört alakítunk ki, melyek számára engedményt biztosítunk minden bor árából.

Azoknak a vásárlóknak, akik a vásárlás napján ünneplik névnapjukat, vagy születésnapjukat, külön ajándékkal kedveskedünk.

Ezen ötletek megvalósítására könnyen kezelhető és hatékony felületet biztosít a Commerce Server 2000 számunkra.

Borbély András, aborbely@protomix.hu



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

2000.10.