

Számítógépes alapismeretek

9. előadás

zoltan.illes@elte.hu

Ami eddig volt...

- Számítógépek architektúrája
- Alapvető alkotóelemek
- Hardver elemek
- Szoftver
 - Gépi kódtól az operációs rendszerig
 - Unix alapok
 - Shell script I., II, III
 - Parancsok, szűrők, sed, awk

Ami ma következik...

- Windows kiszolgáló
 - Szolgáltatások
- Műveletek automatizálása
 - Batch parancsok
 - Batch hiányosságok
- PowerShell

Windows kiszolgáló

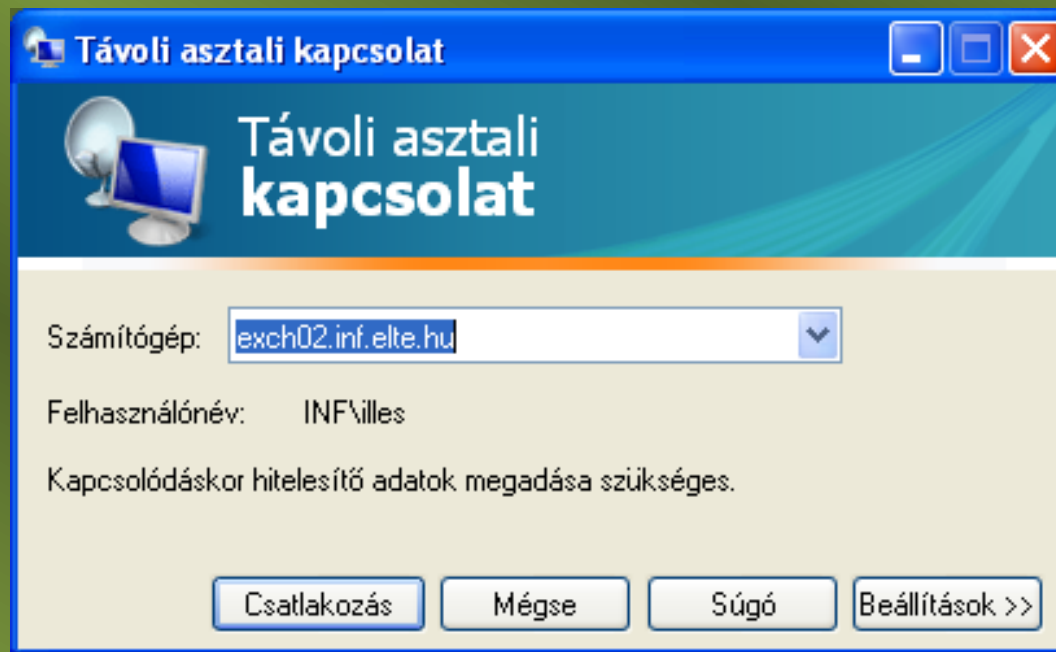
- Alap szolgáltatások
 - Fájl szerver
 - Nyomtató megosztás
 - Web szerver
 - Active Directory
 - Terminál szolgáltatás
 - Telnet, FTP
 - CA, DHCP, DNS, VPN, stb.

Windows szolgáltatások

- Kapcsolódás a kiszolgálóhoz
 - Telnet alapból nem használható
 - MS Terminál Szolgáltatás – mstsc
- Adminisztrációs lehetőségek
 - Szerver szolgáltatások installálása
 - Adminisztrációs eszközök
 - Biztonsági megfontolások

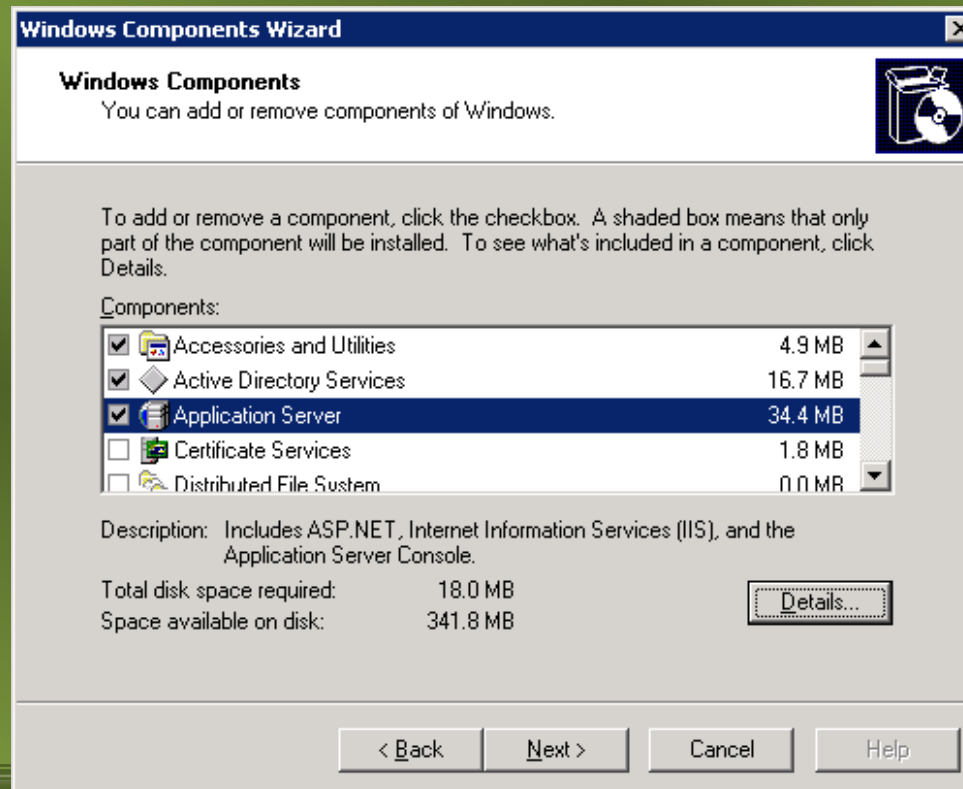
Kapcsolódás egy kiszolgálóhoz

- Terminál szolgáltatások
 - Adminisztrációs céllal alap
 - Start-Futtatás-mstsc



Adminisztrációs lehetőségek

- Programok hozzáadása-eltávolítása
 - Start- Eszköztár – Programok hozzáadása- Windows összetevők



Adminisztrációs eszközök

Administrative Tools

File Edit View Favorites Tools Help

Back Search Folders

Address Administrative Tools Go

Name	Size	Type	Date Modified	Attributes
Certification Authority	2 KB	Shortcut	2007. 01. 17. 13:59	A
Cluster Administrator	2 KB	Shortcut	2007. 04. 28. 13:50	A
Component Services	2 KB	Shortcut	2007. 01. 17. 13:56	A
Computer Management	2 KB	Shortcut	2007. 01. 21. 17:43	A
Configure Your Server Wizard	2 KB	Shortcut	2007. 01. 19. 21:59	A
Data Sources (ODBC)	2 KB	Shortcut	2007. 01. 21. 23:35	A
Distributed File System	2 KB	Shortcut	2008. 11. 01. 19:10	A
Event Viewer	2 KB	Shortcut	2007. 01. 22. 8:38	A
Internet Information Services...	2 KB	Shortcut	2007. 01. 18. 12:17	A
Licensing	2 KB	Shortcut	2007. 01. 17. 14:00	A
Local Security Policy	2 KB	Shortcut	2007. 01. 21. 17:43	A
Manage Your Server	2 KB	Shortcut	2007. 01. 21. 23:36	A
Microsoft .NET Framework 1....	2 KB	Shortcut	2007. 01. 19. 23:02	A
Microsoft .NET Framework 1....	2 KB	Shortcut	2007. 01. 21. 17:41	A
Microsoft .NET Framework 2....	2 KB	Shortcut	2007. 06. 24. 20:22	A
Microsoft Identity Manageme...	2 KB	Shortcut	2007. 01. 19. 21:25	A
Network Load Balancing Mana...	2 KB	Shortcut	2007. 01. 17. 13:55	A
Performance	2 KB	Shortcut	2007. 01. 18. 9:29	A
Remote Desktops	2 KB	Shortcut	2007. 01. 17. 13:56	A
Routing and Remote Access	2 KB	Shortcut	2008. 02. 27. 23:22	A
Services	2 KB	Shortcut	2007. 01. 19. 23:02	A
Terminal Server Licensing	2 KB	Shortcut	2007. 01. 17. 13:57	A
Terminal Services Configuration	2 KB	Shortcut	2007. 01. 17. 13:56	A
Terminal Services Manager	2 KB	Shortcut	2007. 01. 21. 17:42	A
Web Interface for Remote Ad...	1 KB	Shortcut	2007. 04. 28. 13:51	A

Biztonsági megfontolások

- Szerveren nem „internetezünk”
- Szerveren lehetőleg csak szolgáltatásokat menedzselünk!
- Biztonsági frissítések automatikus, vagy gyakori kézi installálása
- Tűzfal használat
- Vírusirtó ??

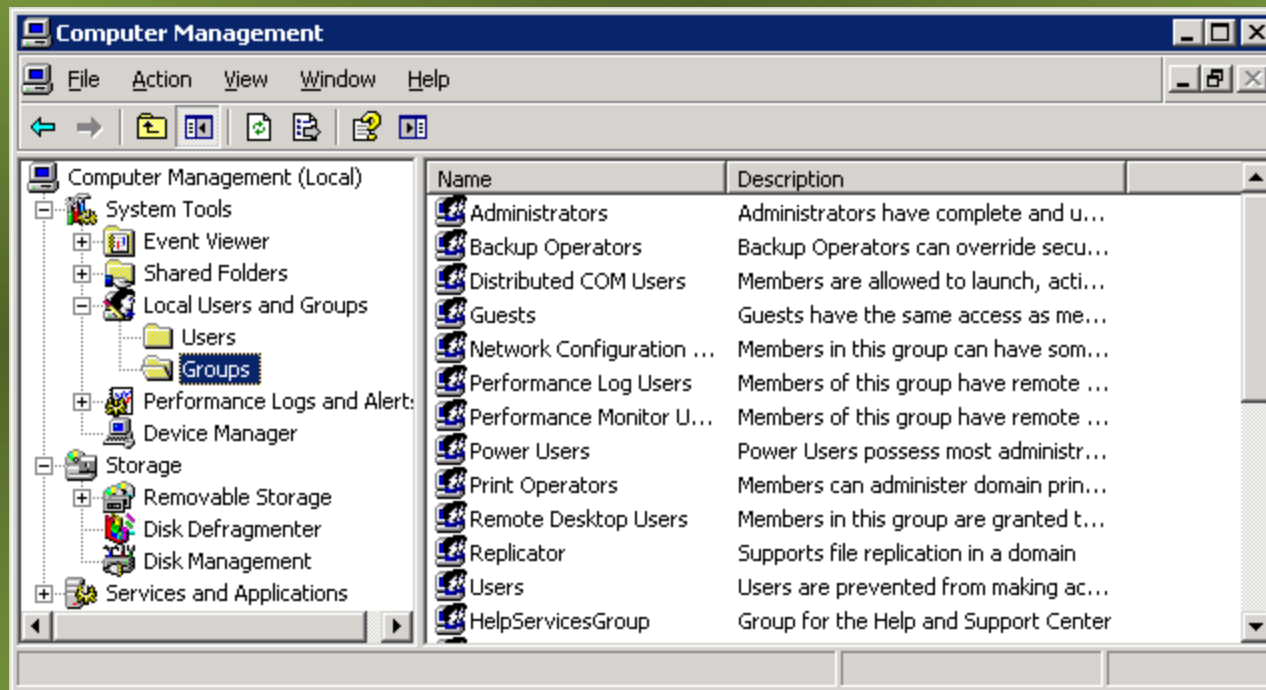
MS Windows Defender

The screenshot shows the Windows Defender application window. The title bar reads "Windows Defender". The navigation bar includes "Home", "Scan", "History", "Tools", and a help icon. The main content area features a "Check for new definitions" section with a yellow background, explaining that definitions allow detection of harmful software. A "Check for Updates Now" button and a link to "View troubleshooting tips on the web" are present. Below this is a "Status" section with a table of scan details.

Last scan:	Today at 2:09. (Quick scan).
Scan schedule:	Daily around 2:00.
Real-time protection:	On
Definition version:	1.45.731.0 created on 2008. 10. 16. at 10:16.

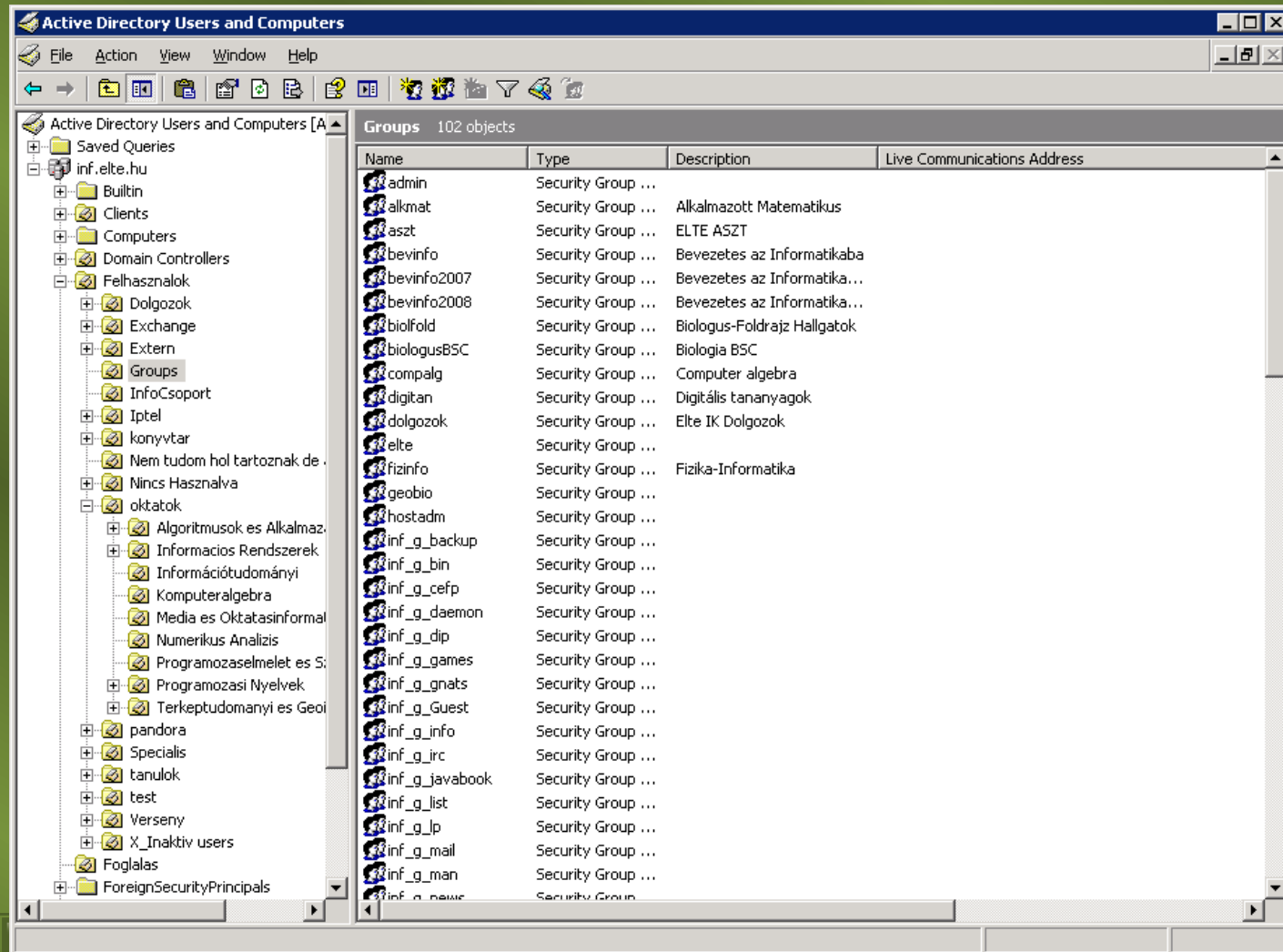
Felhasználó adminisztráció I.

- Helyi gépen: Computer Management



Felhasználó adminisztráció II.

- Tartomány – Domain – Active Directory



Mire való a script ?

- Adminisztráció megkönnyítésére
 - Shell script
 - UNIX, Linux operációs rendszer alatt
- Mi van MS Windows alatt?
 - Batch program (alapok)
 - Windows Scripting Host (VB Script alapú)
 - PowerShell

Batch alapok

- Szöveges parancsok
- Fájlnev kiterjesztés: .bat
- Megjegyzés: rem
- Fő feladata: Parancsok összegyűjtése és indítása egy parancs a batch parancs segítségével.
- echo utasítás
- call masik.bat

Batch változók, paraméterek

- Kis-nagybetű azonos!
- Változó definiálás: set a=5
- Összes változó kiírása: set
- Változó értéke: %név%, Pl: %a%
- path parancs, %path% változó
- prompt utasítás
- %1, %9 a batch paramétere
- %0 a batch program neve
- shift parancs, balra tolja a paramétereket

Batch vezérlési szerkezetek

- Címke deefiniálás: `:cimke1`
- Ugrás egy címkére: `goto cimke1`
- Elágazás: `if [not] feltétel utasítás`
 - `if errorlevel 5 goto ot`
 - Igaz, ha az előző parancs visszatérési értéke nagyobb vagy egyenlő mint 5!!
 - `if %a%=="5" goto 5`
 - `if exist fájlnev goto hat`
- Ciklus: `for %%változó in (lista) do (utasítás)`

Batch ciklus példa

- Hasonlít a shell script for ciklusához
- XP-ben /f kapcsoló

```
rem for ciklus példa
rem ne írja ki a parancsokat
@echo off
rem %%név formát kell használni!!!
for %%i in (alma korte) do (
    echo %%i
)
rem egy file sorainak első szavait vegyük
for /f %%j in (alma.txt) do (
    echo %%j
)
```

PowerShell

- A Microsoft új generációs script nyelve
 - Batch, VBS, WSH utód
- Ingyenes, jelenleg utólag kell installálni XP, Vista alá, a WS 2008 része
- Letölthető:
<http://www.microsoft.com/windowsserver2003/technologies/management/powershell/default.msp>
- Jelenlegi verzió: 1.0

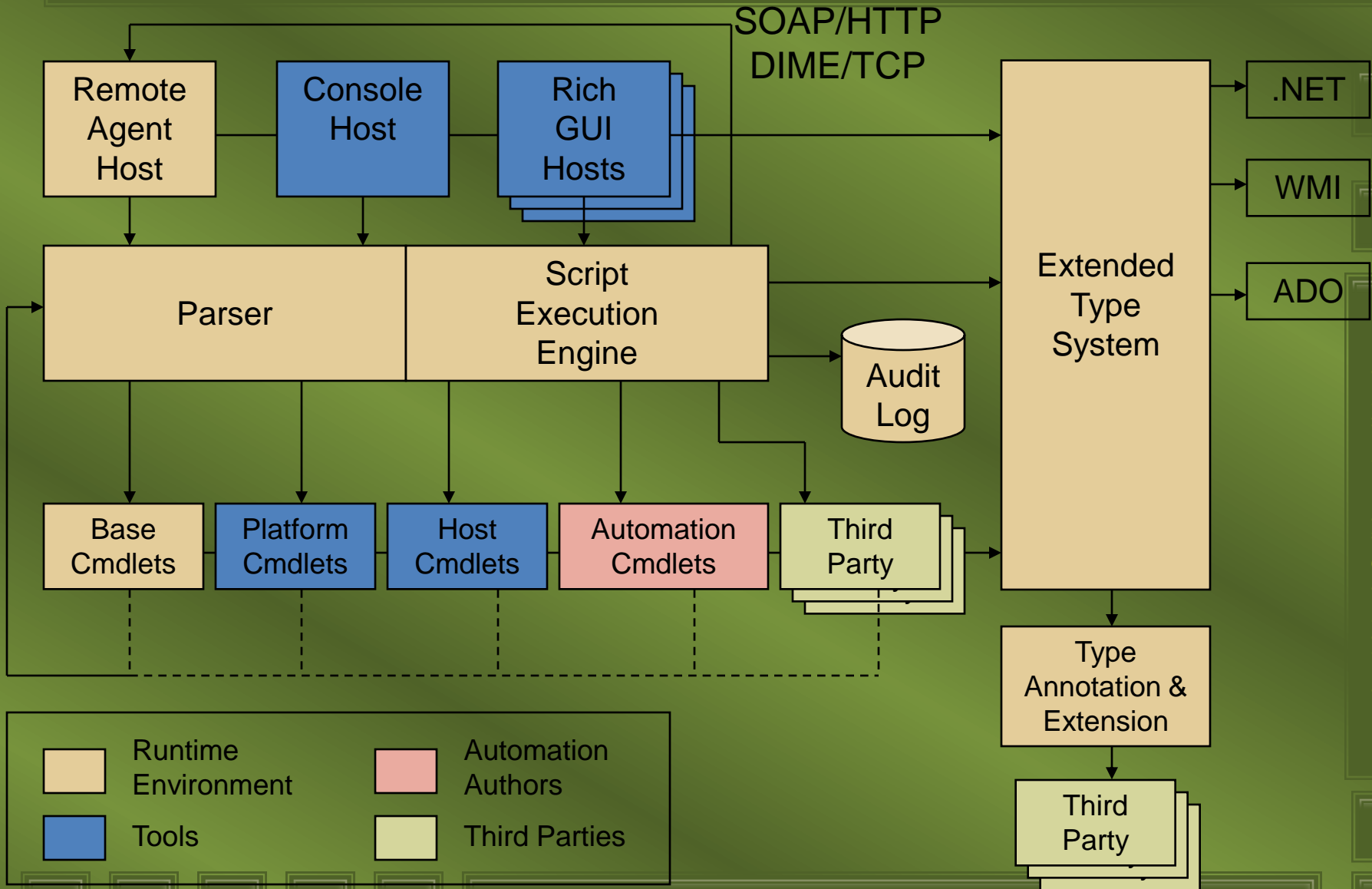
Powershell Install

- .NET Framework 2.0 szükséges
- Start- Programok- Windows Powershell 1.0 programcsoport
- Dokumentációs anyagok (ajánlott)
 - PowerShell 1.0 Documentation Pack
- A parancs egy cmd (command.com) –hoz hasonló karakteres ablakként jelenik meg!

Mire jó a PowerShell?

- Mint a shell script!
- Elsősorban menedzsment célra
 - Hivatalos MS tanfolyam: Course 6434A
 - Automating Windows Server® 2008 Administration with Windows PowerShell
 - Windows Szerver 2008-hoz nem kell letölteni, az operációs rendszer része, de utólag kell hozzáadni!

PowerShell Architecture



PS parancsok formája

- Két nagy család:
 - Get-parancs
 - Set-parancs
- PS parancs : commandlets- cmdlets
 - Több mint 130 beépített parancs.
 - „Hagyományos parancsok” (echo) is használhatók!
- A Tab billentyű kiegészíti a parancsot
- Megjegyzés: #
- Kis-nagybetűre nem érzékeny

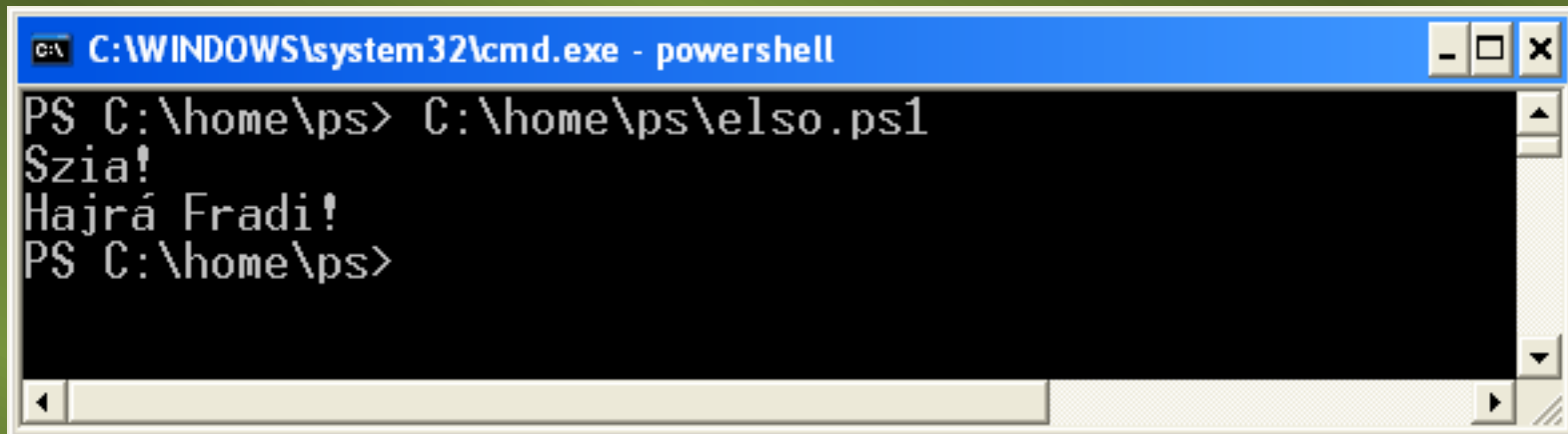
Első PS script- futtatási jog beállítása

- Kiterjesztés: .ps1
 - Jegyzetömb-höz van rendelve
- Get-ExecutionPolicy
 - Restricted, nem engedi futtatni a scripteket, ez az alapértelmezés
- Set-ExecutionPolicy Remotesigned vagy unrestricted
 - Remotesigned: Internetről letöltött állományok esetén csak akkor futtatja, ha megbízható partner írta alá.

Első PS script

- Parancs futtatás, teljes útvonal beírásával

```
# Megjegyzés  
echo Szia!  
#  
Write-Host "Hajrá Fradi!"
```



The screenshot shows a Windows PowerShell terminal window titled "C:\WINDOWS\system32\cmd.exe - powershell". The prompt is "PS C:\home\ps>". The user enters the command "C:\home\ps\első.ps1". The output is "Szia!" followed by "Hajrá Fradi!". The prompt returns to "PS C:\home\ps>".

```
C:\WINDOWS\system32\cmd.exe - powershell  
PS C:\home\ps> C:\home\ps\első.ps1  
Szia!  
Hajrá Fradi!  
PS C:\home\ps>
```

PowerShell script futtatása

- 1. Set-ExecutionPolicy
- 2. Teljes útvonal megadása a parancsnak
 - Ez gyakran: `.\parancsnév` formában
 - Ritkábban: `c:\home\alma.ps1` alakban
- 3. Ha helyköz van egy könyvtárban, akkor az `&` jelet írjuk a parancs elé, és `„` között legyen a parancs.
 - `&”c:\alma fa\jonatán.ps1”`
- 4. CMD-ből: `powershell –noexit c:\elso.ps1`

PowerShell parancssor

- Ha konstanst írunk, azt az értelmező próbálja egy típushoz illeszteni.
- Ha mást nem mondunk, az alap művelet:

write-host

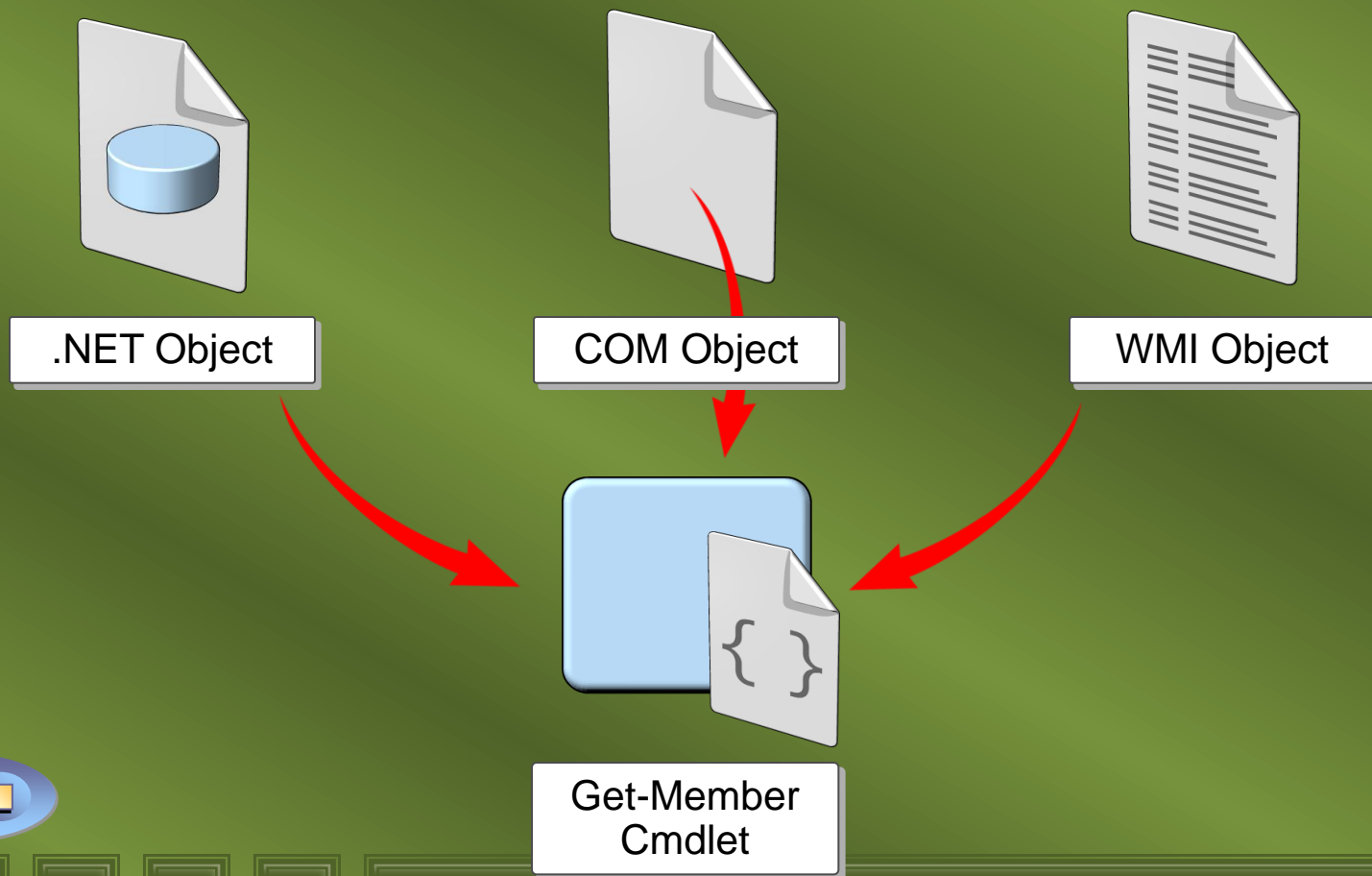
```
PS C:\Users\illes> fradi
The term 'fradi' is not recognized as a cmd in. At line:1
char:5 + fradi <<<<
PS C:\Users\illes> "fradi"
fradi
PS C:\Users\illes> 5
5
PS C:\Users\illes> 'F'
F
PS C:\Users\illes> F
The term 'F' is not recognized as a cmdlet, At line:1char:1
+ F <<<<
```

A PowerShell objektumorientált

- `get-date` – eredmény: 2009. november x...
- Pipeline: `get-date|get-member`
 - A `get-date` objektum a `get-member` bemenetére kerül majd az objektum mezőit kapjuk.
 - `-inputobject` paraméter sok cmdlet-nél él
 - „fradi” | `get-member`
- `get-date|get-member –membertype method`
 - `get-date` metódusait kapjuk meg, hasonlóan `property-t`, tulajdonságokat is lekérhetünk.
- `(get-date).month, day, ...ticks`

.NET, COM, WMI Objektumok

- PowerShell a .NET-et natív módon használja. COM, WMI objektumokat is közvetlenül elér.



Alap PowerShell parancsok

- `alias` , kiírja a definiált rövidítéseket
 - Unix-hoz hasonló parancsok
- `gcm` – Get-Command, kiírja parancsokat
- `echo` – Write-Host, képernyőre írás
- `Get-Help` – rövid leírás parancsokról
 - `Get-Help -full Write-Host`
 - `set-alias gh get-help`
- `dir, ls` – Get-ChildItem, könyvtár tartalom
- `ps` – Get-Process, futó processzek kiírása
- `Sleep` – Start-Sleep, várakozás

PowerShell parancsok, paraméterek

- PowerShell parancs felépítés: Ige-főnév
 - PL: Get-Command
- Paraméterek megadása jellemzően: -név érték
 - Érték lehet: szám, szöveg, dátum
 - Pl: Get-Command –Verb write
- History – F7 előző parancsok
 - felfelnyíl, előző parancs
- Profile:Dokumentumok\WindowsPowerShell könyvtárban: Microsoft.PowerShell_profile.ps1

PowerShell változók

- \$név=érték, kötelező a \$ jel a definiáláskor is
 - Pl: \$f=„fradi”; echo \$f
 - egy sorba több parancs írható, ; az elválasztó
- Támogatott típusok:

Adattípus	Értelmezése	Példa
[int]	Egész szám	-273, -1, 0, 10, 42
[byte]	8-bit, bájt	0, 1, ..., 254, 255
[boolean]	Logikai	\$false, \$true
[char]	Karakter	a, b, c, 1, 2, 3, !, #
[string]	Szöveg	“hello, world”
[datetime]	Idő	April 1, 2008

PowerShell változók használata

- Ha nem jelölünk semmit, az értelmező eldönti a típusát.
 - `$d=6.2e-4; echo $d # 0,00062`, valós lesz
- Magunk is megadhatjuk (típuskényszerítés):
 - `[int] $d=6.2e-4; echo $d # 0`, `$d` egész lesz
 - `$s= [string] 65; echo $s # 65` szöveggként
 - `$s1=[string] [char] 65; echo $s1 # A`
 - `$i=[int] "65"; echo $i # 65` szöveggként

PowerShell változók definiálása parancs segítségével

- `Set-Variable -Name alma -value „jonatán” -option constant`
 - Konstans definiálás
 - Egy leírás adható a `-description` paraméterrel
 - `Get-Variable alma`
- `Clear-Variable alma # alma létezik, csak tartalma nincs.`
- `Remove-Variable alma # alma nem létezik`

Aritmetikai műveletek PowerShell-ben

- `+, -, *, /, %` (maradék)- alapműveletek
 - Nem kell külön parancsot, mint pl. az `expr`!
 - `$a= 32*3; echo $a # 96`
 - `$a=„alma”; $f=„fa”; $c=$a + $f; echo $c #almafa`
 - `$a= „125” + „2”; echo $a # 1252!`
 - `$a= 12 + „4”; echo $a # 16`
 - automatikusan konvertálja a „4”-et
- Értékadások: `=, +=, -=, *=, /=, %=`
- Post növelés, csökkenés: `$a++, $b--`

Még több művelet

- A PowerShell mögött a .NET Framework áll.
 - Az összes típus, double, decimal stb. elérhető
 - Nem csak alaptípusok
 - Példa: `[System.IO.DirectoryInfo]$home=Get-Item D:\home`
- Teljes Math osztály is rendelkezésre áll
 - `[math]::pi`
 - `[math]::sin(2)`, Stb.
- Konverzió: `[system.convert]::toint32(„32”)`
 - Stb. ,Net Framework könyvtár használat

Köszönöm a figyelmet!

zoltan.illes@elte.hu

Illés Zoltán ELTE IK