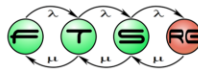


Konfigurációkezelés Windowson

Micskei Zoltán

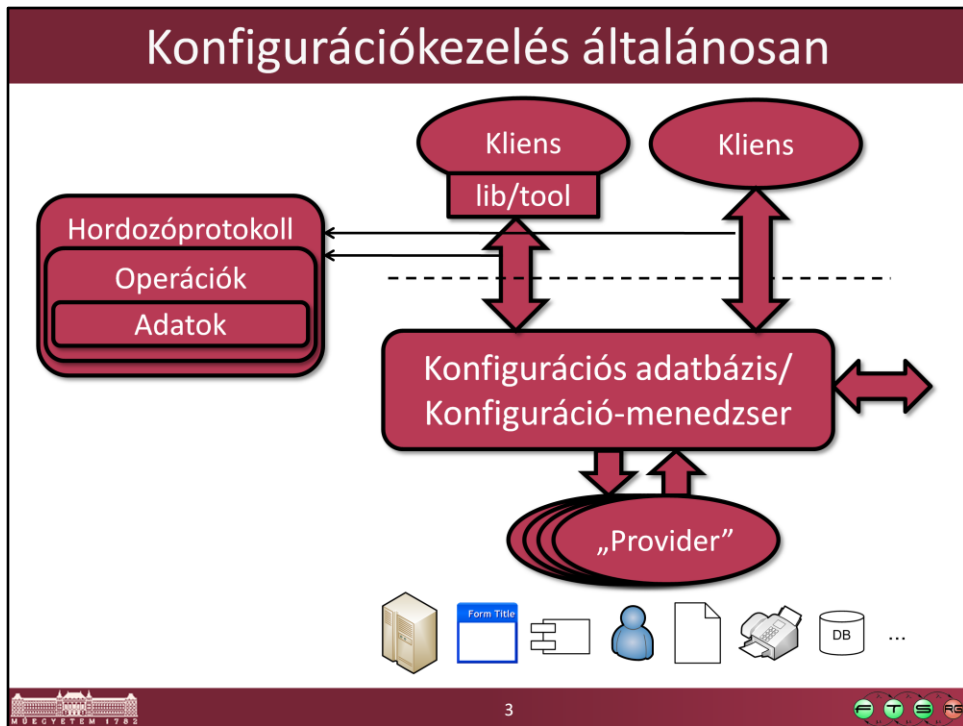
<http://mit.bme.hu/~micskeiz>



Utolsó módosítás: 2012. 03. 19.

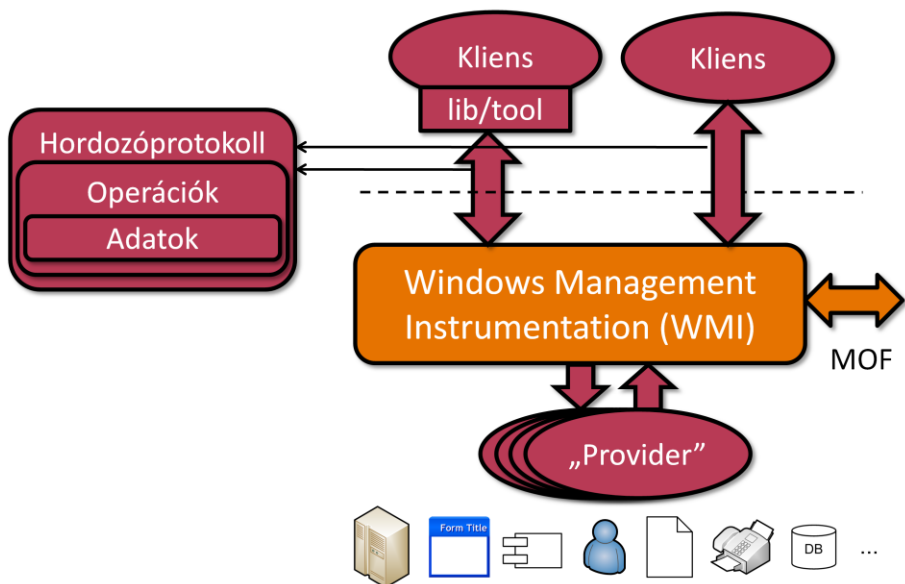
A félév eddigi anyagai

- Modellezés
- Címtárak
- Konfigurációkezelés
 - Konfigurációkezelés alapok
 - **Konfigurációkezelés Windowson**
 - Konfigurációkezelés CMDB



Ez volt az általános séma, amit előző előadáson felrajzoltunk.

Konfigurációkezelés Windowson



Windows Management Instrumentation (WMI)

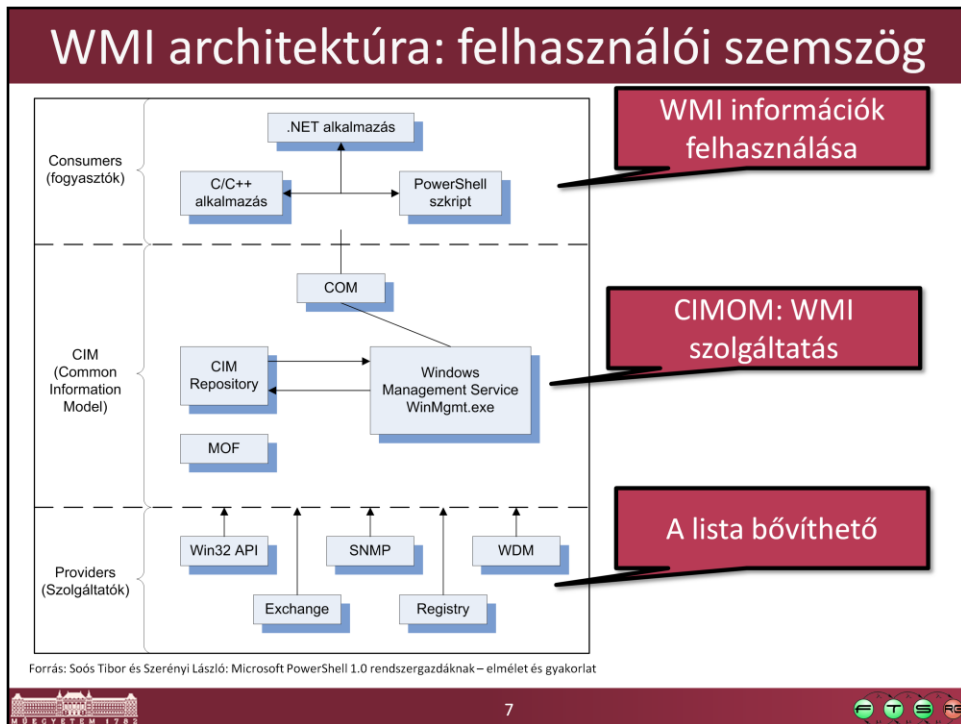


WMI felhasználói szemszögből

- CIM Object Manager Windowsra
 - „winmgmt” szolgáltatás
 - kiterjesztett sémakészlet

- A modern Windowsokon elérhető
 - Felügyeleti megoldások alapja

- Interfész: COM/DCOM
 - IWbemServices
 - lekérdezések, metódushívások, sémamódosítások, ...
 - a CIM szabvány ilyeneket nem definiál!
 - Sokféle elérés: szkriptnyelvek, .NET...



Forrás: Soós Tibor és Szerényi László: Microsoft PowerShell 1.0 rendszergazdáknak – elmélet és gyakorlat
<http://www.microsoft.com/hun/dl.aspx?id=a5b21b7c-3d64-4144-a44c-27a495dbab2c>

WMI architektúra leírása:

MSDN, WMI Architecture, [http://msdn.microsoft.com/en-us/library/aa394553\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa394553(VS.85).aspx)

Elemek:

- Managed objects: akiket a WMI kezel
- Providers: Managed objectról szolgáltat adatokat a WMI interfészének megfelelően. Egy DLL fájlból (megvalósítja az interfész műveleteit), és egy MOF fájlból áll (CIM osztályok, attribútumok és műveleteik leírása)
- WMI service: beolvassa és létrehozza a névtereket, végrehajtja a fogyasztók műveleteit
- WMI repository: statikus adatok tárolása, pl. hogy milyen osztályokat definiálnak a providerek. A legtöbb lekérdezést közvetlenül a providereknek küldi tovább
- Consumers: olyan alkalmazások, amik felhasználják a WMI-t (pl. felügyeleti szoftverek, szkriptek, stb.)

Példa: néhány WMI névtér, osztály

root\cimv2

- **Hardver**
 - Win32_BaseBoard
 - Win32_DiskDrive
- **OS**
 - Win32_PageFile
 - Win32_Registry
- **Telepített szoftver**
 - Win32_Patch
 - Win32_Product
- **Teljesítményszámlálók**
 - Win32_Perf
 - Win32_PerfRawData

directory\ldap

- ds_person
- ds_group

Microsoft\SqlServer

- SqlService
- ServerSettings

IntelNIC

- IANet_EthernetAdapter

DEMO WBEM könyvtár, MOF fájlok

- WMI részei:
 - C:\windows\system32\wbem

- Itt találhatóak:
 - *.exe
 - dll: provider-ek megvalósítása
 - *.mof



A MOF fájlokat általában át kell másolni valahova, hogy meg tudjuk nézni őket.

WMI Query Language (WQL)

- WMI: nagy mennyiségű adat
- WQL: keresés, szűrés ezekben
- SQL szerű szintaxis:
 - **SELECT** attrib **FROM** osztály **WHERE** feltétel

- Példa:

```
SELECT eventcode, message, sourcename  
FROM Win32_NTLogEvent  
WHERE type = 'error' AND logfile ='system'
```



WMI Query Language (WQL)

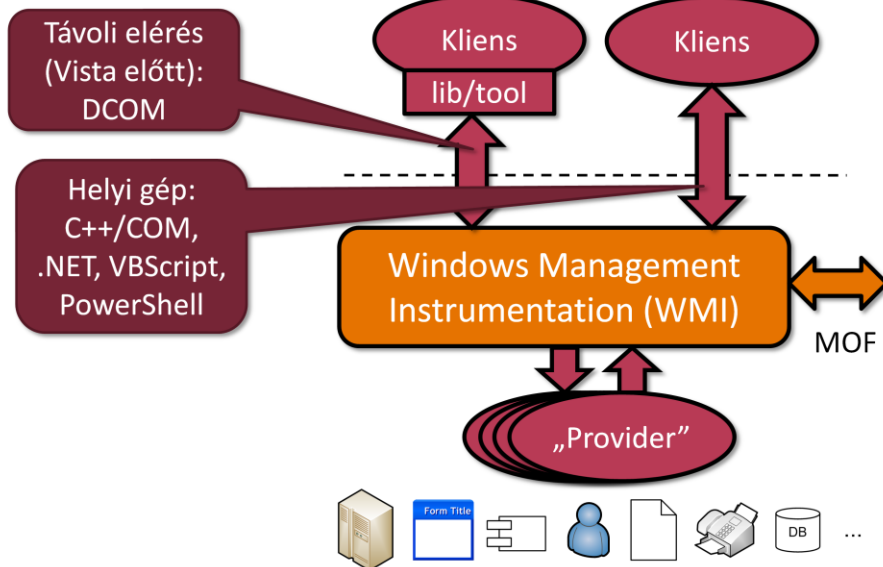
Kapcsolódó példányok lekérézése

- **ASSOCIATORS OF** kulcsszó
- Példa:
 - ASSOCIATORS OF {Win32_LogicalDisk.DeviceID="C:"}
 - Összes kapcsolódó példány bármilyen referenciából
- Csak adott kapcsolat mentén:
 - ASSOCIATORS OF {Win32_LogicalDisk.DeviceID="C:"}
WHERE AssocClass = Win32_SystemDevices
- WHERE ClassDefsOnly – csak osztályneveket



Leírás: MSDN. **ASSOCIATORS OF Statement**, <http://msdn.microsoft.com/en-us/library/aa384793%28v=vs.85%29.aspx>

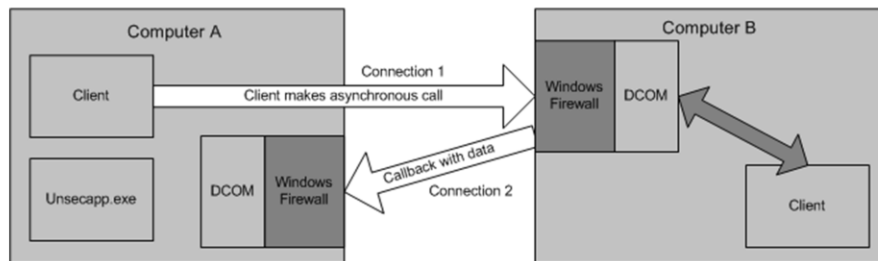
WMI elérése



WMI távoli elérés

- Megvalósítás: DCOM hívások
- Tűzfal beállításainál engedélyezni kell
 - Távoli adminisztráció (TCP 135-ös port)

netsh firewall set service RemoteAdmin enable



13



- Forrás: MSDN. Connecting Through Windows Firewall, <http://msdn.microsoft.com/en-us/library/aa389286.aspx>
- (Vista óta van már WS-Managementet használó elérés, lásd az előadás későbbi részén)
- Tűzfalszabály: a netsh Windows 8-ban már nem javasolt, ott van erre PowerShell cmdlet, például:
 - `Get-NetFirewallRule | ? {($_.displaygroup -like "*wmi*") -and ($_direction -eq "inbound")} | Set-NetFirewallRule -Enabled True`

WMI felhasználása

- Ad-hoc lekérdezések, utasítások
 - Pl.: wmic.exe
- Szkriptek készítése
 - bonyolultabb műveletek, rendszeres feladatok
 - Pl.: VBScript, PowerShell
- Felhasználás .NET-es alkalmazásban
 - System.Management névtér osztályai
- Rendszermenedzsment alkalmazáson keresztül
 - Pl.: MS System Center, HP OpenView...

WMI felhasználása – .NET

- Menyi szabad hely van a C: meghajtón?
- WQL lekérdezés végrehajtása távoli gépen

```
//query free space on drive C:  
long freeSpace;  
  
string path = String.Format(@"\\{0}\root\cimv2", machineName);  
ManagementScope scope = new ManagementScope(path);  
scope.Connect();  
  
string query = "SELECT * FROM Win32_LogicalDisk WHERE DriveType = 3 AND DeviceID = 'C:'";  
  
wqlQuery = new WqlObjectQuery(query);  
searcher = new ManagementObjectSearcher(scope, wqlQuery);  
  
ManagementObjectCollection result = searcher.Get();  
  
if (result != null && result.Count > 0)  
{  
    foreach (ManagementObject disk in result)  
    {  
        // free space in bytes  
        freeSpace = Int64.Parse(disk["FreeSpace"].ToString());  
    }  
}
```

Kapcsolódás a
ManagementScope-hoz

WQL lekérdezés

Eredményhalmaz
feldolgozása

```
//query free space on drive C:  
long freeSpace;
```

```
string path = String.Format(@"\\{0}\root\cimv2", machineName);  
ManagementScope scope = new ManagementScope(path);  
scope.Connect();
```

```
string query = "SELECT * FROM Win32_LogicalDisk WHERE DriveType = 3 AND  
DeviceID = 'C:'";
```

```
wqlQuery = new WqlObjectQuery(query);  
searcher = new ManagementObjectSearcher(scope, wqlQuery);
```

```
ManagementObjectCollection result = searcher.Get();
```

```
if (result != null && result.Count > 0)  
{  
    foreach (ManagementObject disk in result)  
    {  
        // free space in bytes  
        freeSpace = Int64.Parse(disk["FreeSpace"].ToString());  
    }  
}
```

WMI felhasználása – PowerShell

■ Get-WmiObject cmdlet segítségével

```
Windows PowerShell

PS C:\> Get-WmiObject CIM_process | Get-Member

TypeName: System.Management.ManagementObject#root\cimv2\Win32_Process

Name      MemberType Definition
-----
Handles   AliasProperty Handles = Handlecount
ProcessName AliasProperty ProcessName = Name
VM        AliasProperty VM = VirtualSize
WS        AliasProperty WS = WorkingSetSize
AttachDebugger Method      System.Management.ManagementBaseObject AttachDebugger()
GetOwner  Method      System.Management.ManagementBaseObject GetOwner()
GetOwnerSid Method      System.Management.ManagementBaseObject GetOwnerSid()
SetPriority Method      System.Management.ManagementBaseObject SetPriority(System.Int32 Priority)
Terminate Method      System.Management.ManagementBaseObject Terminate(System.UInt32 Reason)
Caption   Property     System.String Caption {get;set;}
CommandLine Property     System.String CommandLine {get;set;}
CreationClassName Property     System.String CreationClassName {get;set;}
CreationDate Property     System.String CreationDate {get;set;}
CreationClassName Property     System.String CreationClassName {get;set;}
```

■ Visszaadott objektum:

- Típus: `System.Management.ManagementObject`
- CIM-ben definiált tulajdonságok
- Metódusok



Get-WmiObject CIM_Process | Get-Member

WMI felhasználása – PowerShell

- Beépített CIM objektumok:

```
PS C:\> (Get-WmiObject -Namespace "root/cimv2" -list).count  
1077
```

- Eredmények szűrése:

- WQL kifejezés átadása a -Query paraméterben
- -Filter paraméter: WQL where része csak
- vagy: PowerShell cmdletek használata

- [wmi] típus használata

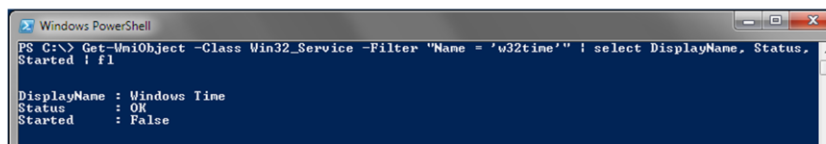
- WmiPath string alapján objektum létrehozása
- [wmi] '\\gep\root\cimv2:Win32_Processor.DeviceID="CPU0"'

WMI felhasználása – PowerShell

- Példa: keressük meg a memóriával kapcsolatos CIM objektumokat

```
PS C:\> get-wmiobject -namespace "root/cimv2" -list | where {$_.name -like "*memory*"}
CIM_MemoryCheck
Min32_PhysicalMemory
CIM_Memory
Min32_CacheMemory
Min32_MemoryArray
CIM_MemoryMappedIO
Min32_DeviceMemoryAddress
CIM_MemoryOnCard
CIM_AssociatedMemory
Min32_AssociatedProcessorMemory
Min32_MemoryDeviceLocation
CIM_MemoryCapacity
Min32_PerfRawData_NETFramework_NETCLRMemory
Min32_PerfRawData_PerFOS_Memory
CIM_PhysicalMemory
Win32_PhysicalMemoryArray
CIM_CacheMemory
Win32_SMBIOSMemory
Min32_MemoryDevice
Win32_SystemMemoryResource
Win32_MemoryDeviceArray
Win32_PhysicalMemoryLocation
CIM_AssociatedProcessorMemory
CIM_MemoryWithMedia
Win32_MemoryArrayLocation
Win32_PerfFormattedData_NETFramework_NETCLRMemory
Win32_PerfFormattedData_PerFOS_Memory
```

- Nézzük meg, hogy fut-e a W32Time szolgáltatás:



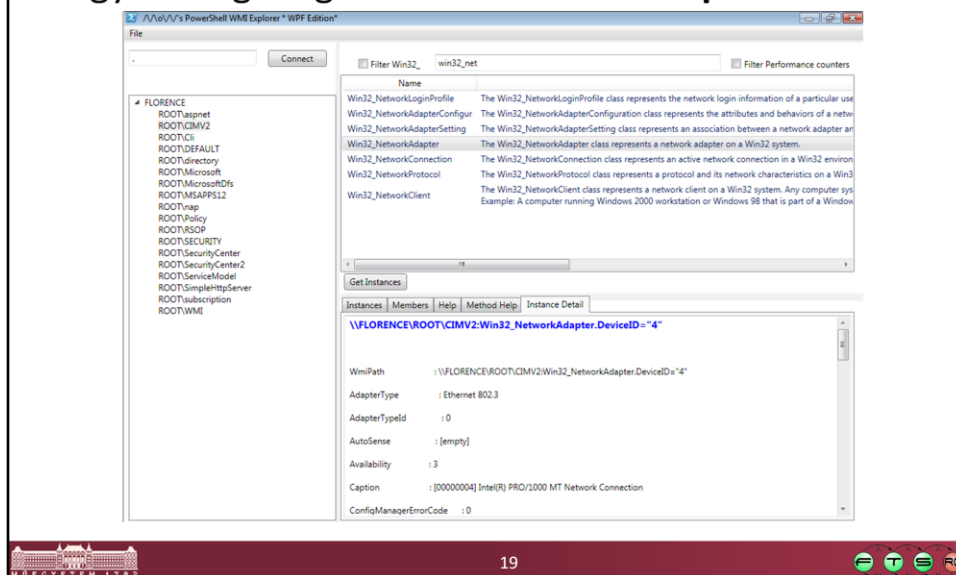
```
Windows PowerShell
PS C:\> Get-WmiObject -Class Win32_Service -Filter "Name = 'w32time'" | select DisplayName, Status, Started | fl
DisplayName : Windows Time
Status      : OK
Started     : False
```

Get-WmiObject -Namespace "root/cimv2" -list | where {\$_.name -like "*memory*"}

Get-WmiObject -Class Win32_Service -Filter "Name = 'w32time'" | select DisplayName, Status, Started | fl

WMI felhasználása – PowerShell

■ Egy kis segítség: PowerShell WMI Explorer



PowerShell WMI Explorer letöltése:

<http://thepowershellguy.com/blogs/posh/pages/powershell-wmi-explorer.aspx>

WMI felhasználása – PowerShell

- Melyik gépen nézzünk filmet este?☺

```
param (
    [string[]] $computers = $(throw "Supply machines!")
)

foreach ($computer in $computers){
    # maximális monitor felbontás
    $screenSize = (Get-WmiObject win32_DesktopMonitor
    -ComputerName $computer | Measure-Object -Maximum
    -Property screenwidth).Maximum

    # van-e divx a gépen
    $divx = (Get-WmiObject win32_codecfile
    -ComputerName $computer | where {$_.filename
    -contains "divx"} | Measure-Object).Count

    write-host $computer $screenSize $divx
}
```



WMI felhasználása – PowerShell

Windows Management Framework v3 újdonság:

- CIM-hez új cmdletek, pl.:
 - `Get-CimInstance`, `Get-CimClass`...
 - `Microsoft.Management.Infrastructure` új névtér osztályait használja!

Windows 8 újdonság:

- CIM-t becsomagoló cmdletek
 - Pl.: `Get-DnsClient`, `Enable-NetAdapter`
 - Összesen 664 darab
 - ~Az összes régi parancssori eszköz lecserélése



21



- PowerShell v3 új CIM cmdletek:
 - `Get-Command -Module CimCmdlets`
- `CimCmdlets` modul cmdletjei alpból WS-Managementet használnak, és nem DCOM-ot (ez a `New-CimSession` segítségével szabályozható)

WMI felhasználása – PowerShell

~~ipconfig.exe~~

~~netsh.exe~~

~~diskpart.exe~~

~~dnscmd.exe~~

~~wmic.exe~~

~~...~~

Get-NetIPAddress

Get-NetFirewallRule

Get-Disk

Add-DnsServerPrimaryZone

Get-CimInstance

...

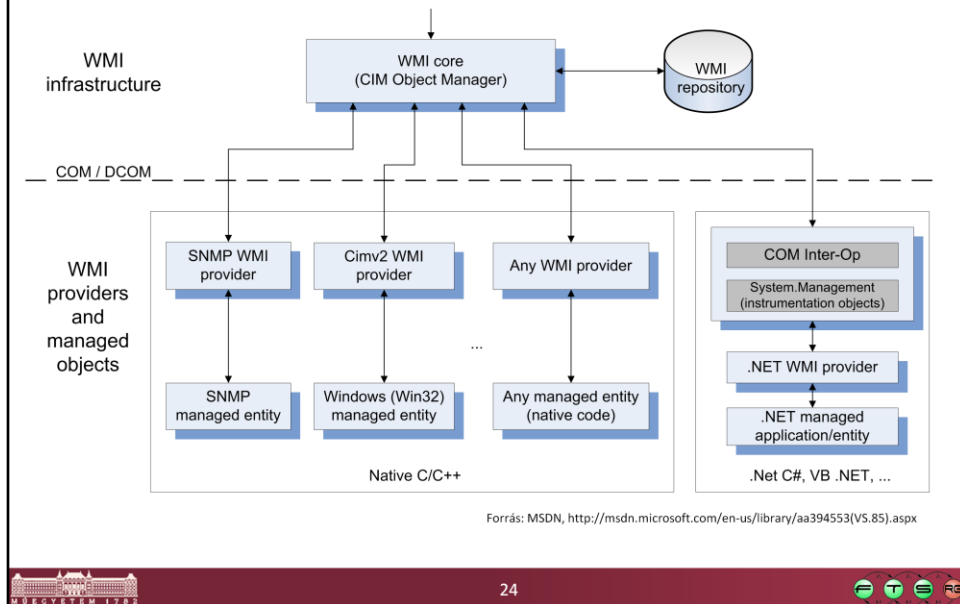


- A legtöbb régi parancssori eszközt felváltja valamelyik PowerShell modul és a benne lévő cmdletek a következő Windowsokban.
- Ezeknek az eredménye szűrhető, kereshető, távoli hozzáférés egyszerűen, szabványosan megy.

WMI fejlesztői szemszögből

- Provider feladatai
 - Metódushívás hatására „történjen is valami”
 - Osztályok példányai „a valóságot tükrözzék”
 - ...
- Megvalósító komponensek: WMI provider-ek
 - IWbemServices-t implementáló COM kiszolgálók
 - Meglehetősen bonyolult implementáció
 - Alternatíva: .NET System.Management névtér

WMI Providerek



Forrás: MSDN. WMI Architecture, [http://msdn.microsoft.com/en-us/library/aa394553\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa394553(VS.85).aspx)

WMI Provider készítése .NET-ben

- **Adatok megadása: attribútumokkal**
 - WmiConfiguration: névtér megadása
 - ManagementEntity: osztály dekorálása
 - ManagementProbe: WMI-ből olvasható tulajdonság
 - ManagementTask: WMI-ből elérhető metódus
- **Példányosítás:**
 - **Singleton**: csak egy példánya van
 - **Multi-instance**: ManagementKey adja meg, hogy melyik példány kell nekünk
- **Futtatási mód:**
 - **Coupled**: WMI szolgáltatás folyamatán belül
 - **Decoupled**: külön folyamatban, az alkalmazással együtt
- Lásd később a „Felügyeletre tervezés” segédletben



MSDN. WMI Provider Extensions, <http://msdn.microsoft.com/en-us/library/bb404670.aspx>

Egyszerű WMI Provider C#-ban

```
// provider will be in the root/MortgageCalc namespace.
[assembly: WmiConfiguration("root/MortgageCalc", HostingModel = ManagementHostingModel.Decoupled)]

// This is the installer class that installs an instrumented assembly.
[System.ComponentModel.RunInstaller(true)]
public class TheInstaller : DefaultManagementInstaller
{
}

namespace DecoupledWMIProvider
{
    // Use the ManagementEntity attribute on the class to specify that this is a provider
    [ManagementEntity(Name = "MortgageCalc")]
    [ManagementQualifier("Description", Value = "Allows you to read and set configuration settings.")]
    public class MortgageCalcWMIProvider
    {
        // Specify that a property is a read/write property in the provider.
        [ManagementConfiguration]
        public double highestLoanAmount
        {
            get
            {
                return this.highestLoanAmount;
            }
        }
    }
}
```

Hosting
modell

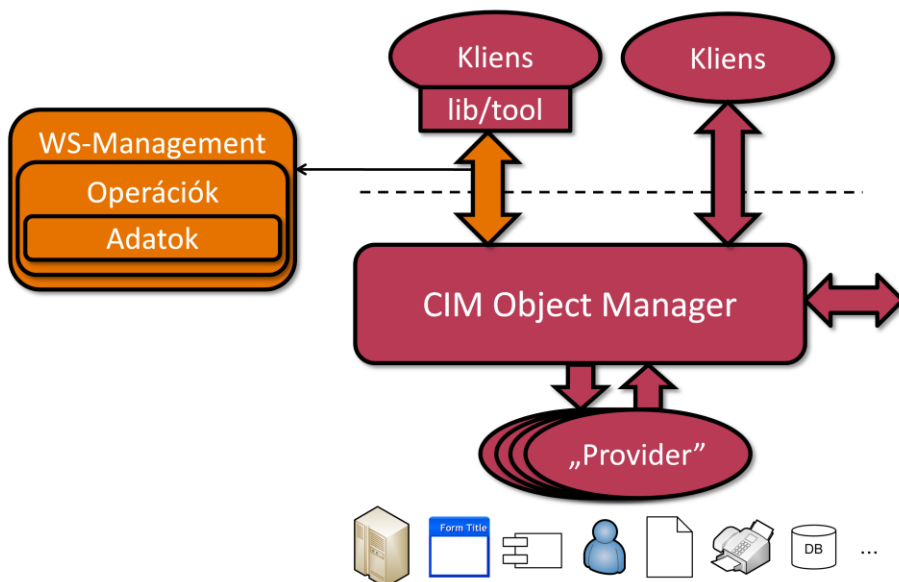
Provider
neve

WMI-ből
lekérdezhető
tulajdonság

Web Services for Management (WS-Management)

Kibocsátó:	Distributed Management Task Force (DMTF)
Megalkotók:	AMD, Dell, Intel, Microsoft, Sun...
Verzió:	első draft (2005. 09.), aktuális: 1.1.0 (2010. 03. 03.)
Cél:	Webszolgáltatás alapú protokoll rendszerek menedzseléséhez

Konfigurációkezelés általánosan



WS-Management

- WBEM egyik protokollja
- Web Service alapú
 - sok egyéb WS-* protokollra épül
- CIMOM elérése, lekérdezése, menedzselése
 - De használható nem csak CIMOM-hoz!

WS-Management példa üzenet

The diagram shows an XML message structure with callouts explaining its components:

- Melyik gép** (Which machine) points to the `<wsa:To> Address </wsa:To>` field.
- Mit akarunk** (What do we want) points to the `<wsa:Action> Action URI </wsa:Action>` field.
- Gép melyik erőforrásával** (Which machine with which resource) points to the `<wsman:ResourceURI>resURI</wsman:ResourceURI>` field.
- Ha az erőforrás egy csoport (pl. CPU-k), akkor a csoport melyik tagja** (If the resource is a group (e.g., CPUs), which member of the group) points to the `<wsman:Selector Name="selector-name"> selector-value </wsman:Selector>` block.

```
<s:Header>
  <wsa:To> Address </wsa:To>
  <wsa:Action> Action URI </wsa:Action>
  <wsman:ResourceURI>resURI</wsman:ResourceURI>
  <wsman:SelectorSet>
    <wsman:Selector Name="selector-name">
      selector-value
    </wsman:Selector>
  </wsman:SelectorSet>
  ...
</s:Header>
```

```
<s:Header>
  <wsa:To> Address </wsa:To>
  <wsa:Action> Action URI </wsa:Action>
  <wsman:ResourceURI>resURI</wsman:ResourceURI>
  <wsman:SelectorSet>
    <wsman:Selector Name="Selector-name">
      Selector-value
    </wsman:Selector>
  </wsman:SelectorSet>
  ...
</s:Header>
```

WS-Management: műveletek

- **DISCOVER**
 - erőforrások felderítése
- **GET, PUT, CREATE, DELETE**
 - erőforrások kezelése
- **ENUMERATE**
 - gyűjtemények elemeinek felsorolása
- **SUBSCRIBE**
 - eseményekre feliratkozás
- **EXECUTE**
 - metódusok meghívása



Implementációk

- Windows Remote Management (WinRM)
 - WMI lekérdezés és távoli parancssor
 - CLI, scripting API

- Openwsman; F/OSS
 - Kiszolgáló: CIM-XML-t „fordít” WS-Managementre
 - Adapter tervezési minta; SBLIM SFCC, mint kliens
 - CLI, mod_wsman, Ruby kötések

- wiseman; F/OSS
 - Java alapú kliens és kiszolgáló megvalósítás támogatása

Ki használja?

- Windows Server 2008 R2
 - Server Manager távoli gépre
 - PowerShell 2.0 Remoting
- VMware ESX és ESXi
 - openwsman daemon futtatása
- Microsoft Operations Manager
 - Cross Platform: Linux gépek monitorozása
- HP System Insights Manager
- ...

DEMO ESXi lekérdezése WinRM-ből

- Kliens: Windows 7 (WinRM)
- Kiszolgáló: VMware ESXi (openwsman)

```
Get-WSManInstance cimv2/CIM_Processor  
-Enumerate  
-ComputerName aragorn.mit.bme.hu  
-UseSSL -Port 443 -ApplicationName wsman  
-Authentication Basic -Credential root
```

- [VMware API leírás](#)



35



- VMware CIM API leírás: <http://www.vmware.com/support/developer/cim-sdk/>
- Ugyanez a lekérdezés a régebbi winrm.cmd eszközzel:

```
winrm enum cimv2/CIM_Processor -  
r:https://aragorn.mit.bme.hu/wsman -skipCAcheck -  
skipCNcheck -auth:basic -username:root -password:password
```

Windows Remote Management (WinRM)

Windows Remote Management (WinRM)

- Új távoli menedzsment szolgáltatás a Windowsban
- WS-Management implementáció
- Elérhetőség:
 - Windows 7, Server 2008 R2: gyárilag (2.0)
 - Vista, Server 2008: gyárilag (1.0), telepíthető (2.0)
 - XP, Windows Server 2003 R2: külön telepíthető (2.0 is)
- WMI beágyazható WinRM-be, de több annál



37



A WinRM 3.0 CTP verziója 2011 végén jelent meg, a végleges verzió a Windows 8 végleges változatokor érkezik majd valószínűleg.

WinRM vs. WMI

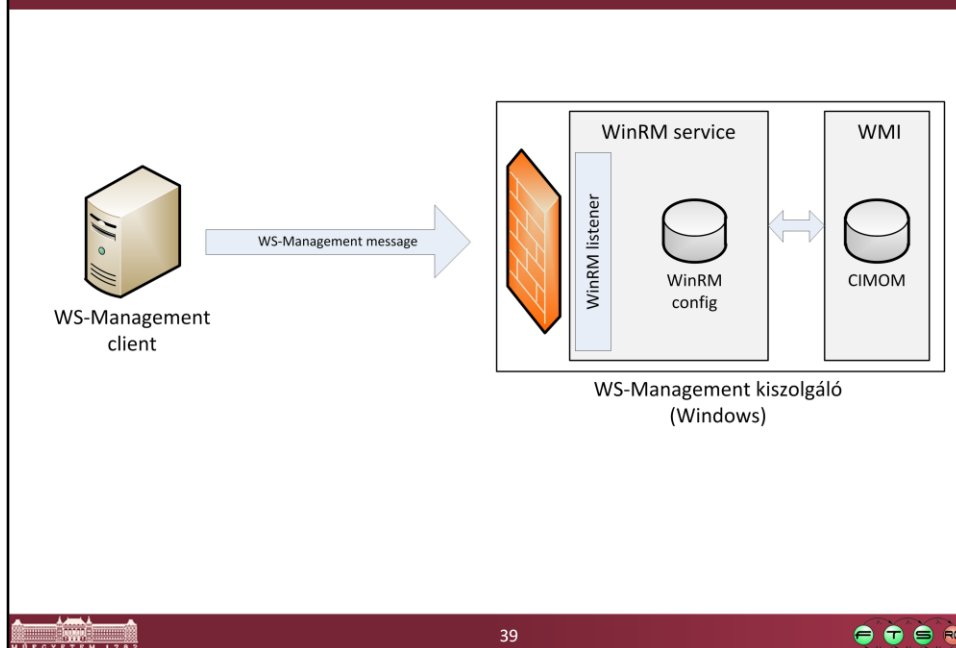
- WMI:
 - DCOM-ot használ a távoli elérésre
 - Nem szabványos, nem tűzfalbarát

- WinRM:
 - WS-Management
 - DMTF szabvány, 5985/5986 porton keresztül



A WinRM 1.0-ás verziója még a sima HTTP és HTTPS portokat (80 és 443) használta.

WinRM architektúra



Egy gépen több listener is lehet, pl. az SSL-t használóhoz külön egy másik.

WinRM inicializálása

- WinRM részei:
 - WinRM szolgáltatás
 - WinRM listener: WS-Management üzenetek fogadására szolgáló végpont definíciója
- Inicializálás (kiszolgáló oldalán!):

```
PS C:\Windows\system32> Set-WSManQuickConfig

WinRM Quick Configuration
Running the Set-WSManQuickConfig command has significant security implications, as it enables
remote management through the WinRM service on this computer.
This command:
1. Checks whether the WinRM service is running. If the WinRM service is not running, the service
is started.
2. Sets the WinRM service startup type to automatic.
3. Creates a listener to accept requests on any IP address. By default, the transport is HTTP.
4. Enables a firewall exception for WS-Management traffic.
5. Enables Kerberos and Negotiate service authentication.
Do you want to enable remote management through the WinRM service on this computer?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"):
```



Példa WMI lekérdezés WinRM-en keresztül

- Művelet: GET
- Erőforrás URI: WMI objektum
- Távoli gép, hitelesítési információ megadása

```
PS C:\> Get-WSManInstance -ResourceURI http://schemas.microsoft.com/wbem/urman/1/wmi/root/cimv2/Win32_LogicalDisk?DeviceID=C: -ComputerName 10.90.1.10 -Credential irfdemo\administrator

xsi          : http://www.w3.org/2001/XMLSchema-instance
p            : http://schemas.microsoft.com/wbem/urman/1/wmi/root/cimv2/Win32_LogicalDisk
cin         : http://schemas.dmtf.org/wbem/wscim/1/common
type        : p:Win32_LogicalDisk_Type
lang        : en-US
Access      : 0
Availability : Availability
BlockSize   : BlockSize
Caption     : C:
Compressed  : false
ConfigManagerErrorCode : ConfigManagerErrorCode
ConfigManagerUserConfiguration : ConfigManagerUserConfiguration
CreationClassName : Win32_LogicalDisk
Description  : Local Fixed Disk
DeviceID    : C:
DriveType   : 3
ErrorCleared : ErrorCleared
ErrorDescription : ErrorDescription
ErrorMethodology : ErrorMethodology
FileSystem   : NTFS
FreeSpace    : 32250716160
```



Get-WSManInstance -ResourceURI wmicimv2/Win32_LogicalDisk?DeviceID=C: -
ComputerName 10.90.1.10 -Credential irfdemo\administrator

További műveletek

- Enumerate:
 - Több objektum lekérése
 - -Enumerate wmicimv2/Win32_LogicalDisk

- Invoke:
 - Beavatkozás
 - Invoke-WSManAction -Action StopService
-ResourceURI wmicimv2/win32_service
-SelectorSet @{name="W32Time"}

WMI objektum elérése

- Resource URI forma:
 - prefix/osztály?tulajdonság=érték

- URI WMI prefix:
 - `http://schemas.microsoft.com/wbem/wsman/1/wmi/<WmiNamespace>`

- URI példa:
 - `http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Service?Name=Winmgmt`



Figyelem: a WSMAN ResourceURI más, mint a CIM object path-ban szereplő URL!

Alias-ok használata

```
winrm help alias
```

```
wmi          =  
  http://schemas.microsoft.com/wbem/wsman/1/wmi  
wmicimv2    =  
  http://schemas.microsoft.com/wbem/wsman/1/wmi/  
  root/cimv2  
  
cimv2       =  
  http://schemas.dmtf.org/wbem/wscim/1/cim-  
  schema/2  
  
wsman       =  
  http://schemas.microsoft.com/wbem/wsman/1
```



Figyeljünk arra, hogy más URL prefixet kell használni ha WMI specifikus osztályt, és más, ha DMTF szabványos osztályt akarunk lekérdezni (különböző platformok közötti lekérdezéshez).

WinRM felhasználása

- WinRM.cmd parancssori eszköz
- PowerShell Remoting
- WinRM API (WSMan Automation)
 - pl.: VBScript, C#
- XML üzenet legyártása egyéb módon

WinRM példa üzenet

■ Win32_LogicalDisk WMI objektum lekérdezése

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:w="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
  <s:Header>
    <a:To>http://RemoteComputer:80/wsman</a:To>
    <w:ResourceURI s:mustUnderstand="true">
      http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/win32_logicaldisk
    </w:ResourceURI>
    <a:ReplyTo>
    <a:Address s:mustUnderstand="true">
      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </a:Address>
    <a:ReplyTo>
    <a:Action s:mustUnderstand="true">
      http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </a:Action>
    <w:MaxEnvelopeSize s:mustUnderstand="true">153600</w:MaxEnvelopeSize>
    <a:MessageID>uuid:4ED2993C-4339-4E99-81FC-C2FD3812781A</a:MessageID>
    <w:Locale xml:lang="en-US" s:mustUnderstand="false"/>
    <w:SelectorSet>
    <w:Selector Name="DeviceId">c</w:Selector>
    </w:SelectorSet>
    <w:OperationTimeout>PT60.000S</w:OperationTimeout>
  </s:Header>
  <s:Body/>
</s:Envelope>
```



Biztonság

- Titkosítás:
 - Nincs / HTTPS csatorna / XML üzenet titkosítása
- Hitelesítés:
 - Basic / Negotiate / Kerberos ...
- Távoli fél azonosítása
 - Tanúsítvány / tartományi környezetben Kerberos
- Trusted hosts

- Leírások:
 - `Get-Help about_Remote_Troubleshooting`

WinRM használata PowerShellből

- Beépített cmdletek a winrm.cmd feladataira
 - Lásd: `Get-Help about_wsman`
 - `Set-WSManQuickConfig`, `Test-WSMan`, `Get-WSManInstance`, `cd wsman`:

- PowerShell Remoting
 - parancsok/szkriptek végrehajtása távoli gépen
 - WinRM-et használ alul
 - Interaktív / batch mód (akár több gépen párhuzamosan)

DEMO PowerShell remoting

- Követelmény:
 - PowerShell 2.0, WinRM 2.0
 - Lásd: `Get-Help about_remoting_requirements`
- Remoting engedélyezése: `Enable-PSRemoting`
- Egy parancs távoli végrehajtása:
`Invoke-Command -ComputerName nev -ScriptBlock {...}`
- Távoli munkamenet:
 - `Enter-PSSession gepnev`
- Lásd: `Get-Help about_remoting`

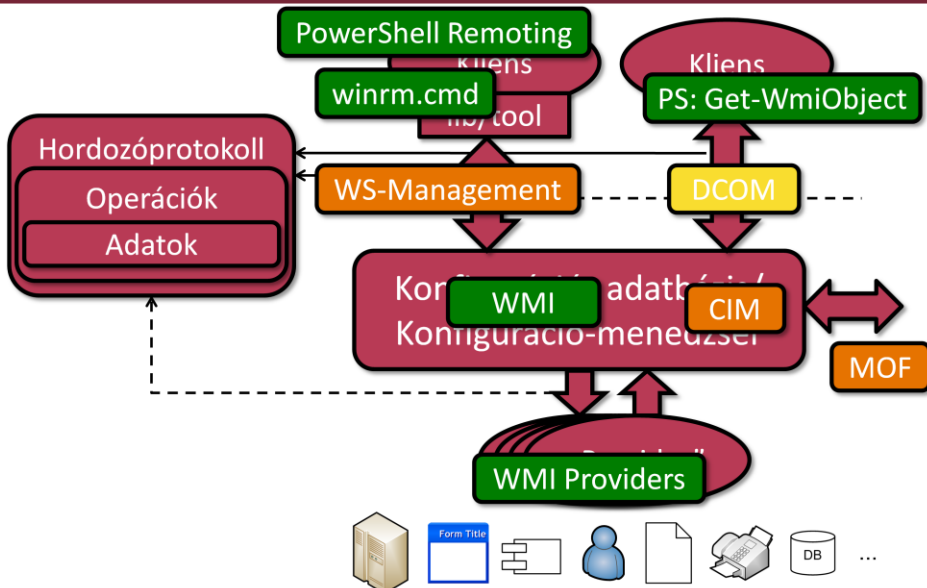


49



WMF v2 letöltése: Windows Management Framework (Windows PowerShell 2.0, WinRM 2.0, and BITS 4.0), <http://support.microsoft.com/kb/968929>

Összefoglalás



Összefoglalás

- Konfigurációkezelés Windowson:
 - CIM, WMI
- Saját alkalmazásunk kiegészítése:
 - WMI provider készítése
- Távoli lekérdezés, beavatkozás:
 - WinRM (alacsony szinten)
 - PowerShell 2.0 Remoting

További információ

- WMI
 - [MSDN: WMI](#)
 - [WMI Blog](#)
- WinRM
 - [MSDN: Windows Remote Management](#)
- Openwsman
 - <http://www.openwsman.org/>



- MSDN. Windows Management Instrumentation, [http://msdn.microsoft.com/en-us/library/aa394582\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa394582(VS.85).aspx)
- MSDN Blogs. Windows Management Infrastructure Blog, <http://blogs.msdn.com/b/wmi/>
- MSDN. Windows Remote Management, [http://msdn.microsoft.com/en-us/library/aa384426\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa384426(VS.85).aspx)