

# El proveedor Active Directory

El proveedor es una de las múltiples formas de administrar un servicio de directorio Active Directory. Es evidente que no es nuestra forma de trabajar preferida, pero tiene mérito que exista...

El proveedor AD, y los proveedores en general, hacen que la exploración de los objetos sea similar a la exploración de una estructura de archivos y directorios. En efecto, en lugar de utilizar el conjunto de comandos dedicado, que veremos un poco más adelante, para manipular los objetos de un proveedor, tendremos que hacer uso de comandos más genéricos como los siguientes:

Get-PSProvider	Remove-PSDrive	New-Item	Rename-Item	Clear-ItemProperty
Get-PSDrive	Get-ChildItem	Remove-Item	Get-ItemProperty	Get-ACL
New-PSDrive	Get-Item	Move-Item	Set-ItemProperty	Set-ACL

El punto de entrada de un proveedor ActiveDirectory es el lector AD:. Recuerde que para listar los lectores asociados a los proveedores, es necesario utilizar el commandlet Get-PSDrive.

PS > Get-PSDrive					
Name	Used (GB)	Free (GB)	Provider	Root	CurrentLocation
----	-----	-----	-----	----	-----
A			FileSystem	A:\	
AD			ActiveDirectory	//RootDSE/	
Alias			Alias		
C	9,11	117,79	FileSystem	C:\	Users\Administrator
cert			Certificate	\	
D	2,32		FileSystem	D:\	
Env			Environment		
Function			Function		
HKCU			Registry	HKEY_CURRENT_USER	
HKLM			Registry	HKEY_LOCAL_MACHINE	
Variable			Variable		
WSMan			WSMan		

## 1. Exploración del proveedor

Ahora para empezar la exploración, el punto de partida se sitúa a nivel del lector AD:.

PS C:\Users\Administrator > cd AD :		
PS AD:\> Get-ChildItem		
Name	ObjectClass	DistinguishedName
----	-----	-----
powershell-scripting	domainDNS	DC=powershell-scripting,DC=com
Configuration	configuration	CN=Configuration,DC=powershell-script...
Schema	dMD	CN=Schema,CN=Configuration,DC=pow...
DomainDnsZones	domainDNS	DC=DomainDnsZones,DC=powershell...
ForestDnsZones	domainDNS	DC=ForestDnsZones,DC=powershell...
PS AD:\> cd '.\DC=powershell-scripting,DC=com'		

Name	ObjectClass	DistinguishedName
----	-----	-----
Builtin	builtinDomain	CN=Builtin,DC=powershell...
Computers	container	CN=Computers,DC=powershell...
Domain Controllers	organizationalUnit	OU=Domain Controllers,DC...
ForeignSecurityPrincipals	container	CN=ForeignSecurityPrincipa...
Infrastructure	infrastructureUpdate	CN=Infrastructure,DC=power...
LostAndFound	lostAndFound	CN=LostAndFound,DC=powersh...
Managed Service Accounts	container	CN=Managed Service Account...
NTDS Quotas	msDS-QuotaContainer	CN=NTDS Quotas,DC=powershe...
Program Data	container	CN=Program Data,DC=powersh...
System	container	CN=System,DC=powershell-sc...
test	organizationalUnit	OU=pruebas,DC=powershell-s...
Users	container	CN=Users,DC=powershell-scr...

La exploración se realiza como en un sistema de archivos. Observe que tenemos una «unidad organizativa» llamada «pruebas». Vamos a ver que contiene:

```
PS AD:\DC=powershell-scripting,DC=com> cd .\OU=pruebas
PS AD:\OU=pruebas,DC=powershell-scripting,DC=com> Get-ChildItem
```

Name	ObjectClass	DistinguishedName
----	-----	-----
Oscar Vázquez	user	CN=Oscar Vázquez,OU=pruebas,DC=powershell...

Esta unidad organizativa contiene un usuario llamado «Oscar Vázquez».



Para simplificarle la tarea de la navegación en este proveedor, use la tecla [Tab] (función «autocompletar») ya que es necesario indicar cada vez el DistinguishedName de los objetos a visitar; lo que puede convertirse rápidamente en algo aburrido de escribir...

## 2. Modificación de un objeto de servicio de directorio

Supongamos que dicho usuario no está en su ubicación y que queremos moverlo a la unidad organizativa Users. Al igual que para mover un archivo, vamos a utilizar el comando `Move-Item`:

```
PS > Move-Item '.\CN=Oscar Vázquez' '\CN=Users,DC=powershell-scripting,DC=com'
```

Ahora modificamos la propiedad `EmployeeID` de un objeto «user» con el comando `Set-ItemProperty`:

```
PS AD:\> Set-ItemProperty `
-path 'AD:\CN=Oscar Vázquez,CN=Users,DC=powershell-scripting,DC=com' `
-name 'EmployeeID' -value '123456'
```

Verificamos si el valor se ha modificado correctamente:

```
PS AD:\> Get-ItemProperty `
-path 'AD:\CN=Oscar Vázquez,CN=Users,DC=powershell-scripting,DC=com' `
-name 'EmployeeID'

PSPath      : ActiveDirectory:://RootDSE/CN=Oscar Vázquez,CN=Users,
              DC=powershell-scripting,DC=com
PSParentPath : ActiveDirectory:://RootDSE/CN=Users,DC=powershell-scripti...
```

```
PSChildName : CN=Oscar Vázquez
PSDrive      : AD
PSProvider   : ActiveDirectory
EmployeeID   : 123456
```

Como puede ver el resultado es un poco extenso. Si queremos restringirlo para solicitar la propiedad `EmployeeID`, podemos escribir lo siguiente:

```
PS AD:\> Get-ItemProperty `
-path 'AD:\CN=Oscar Vázquez,CN=Users,DC=powershell-scripting,DC=com' `
-name 'EmployeeID' | Select-Object EmployeeID -ExpandProperty EmployeeID

123456
```



Aquí llamaremos al parámetro **-ExpandProperty** del comando `Select-Object` para evitarnos la etapa consistente en extraer una sola propiedad. En efecto, el comando `Select-Object` recupera una única propiedad del objeto pero nos devuelve un objeto de tipo `PSObject`.

Para entenderlo mejor veamos el valor que hubiésemos obtenido si no hubiésemos especificado el parámetro **ExpandProperty**:

```
PS AD:\> Get-ItemProperty `
-path 'AD:\CN=Oscar Vázquez,CN=Users,DC=powershell-scripting,DC=com' `
-name 'EmployeeID' | Select-Object EmployeeID

EmployeeID
-----
123456
```

Para obtener únicamente el valor, necesitaremos una vez más solicitar la propiedad `EmployeeID`, como se muestra a continuación:

```
PS AD:\> (Get-ItemProperty `
-path 'AD:\CN=Oscar Vázquez,CN=Users,DC=powershell-scripting,DC=com' `
-name 'EmployeeID' | Select-Object EmployeeID).EmployeeID

123456
```

Si se procede de este modo, el uso de `Select-Object` es incluso superfluo ya que se podría simplificar la escritura de la manera siguiente:

```
PS AD:\> (Get-ItemProperty `
-path 'AD:\CN=Oscar Vázquez,CN=Users,DC=powershell-scripting,DC=com' `
-name 'EmployeeID').EmployeeID

123456
```



Esto demuestra una vez más que hay muchas maneras de escribir un script y que no tiene porqué haber una manera de hacerlo que sea la mejor. La mejor será aquella que más le guste y que le permita reutilizar los scripts meses más tarde sin necesidad de devanarse los sesos para tratar de comprender lo que había escrito.