

Crear cuentas de usuarios de forma masiva

Problemática

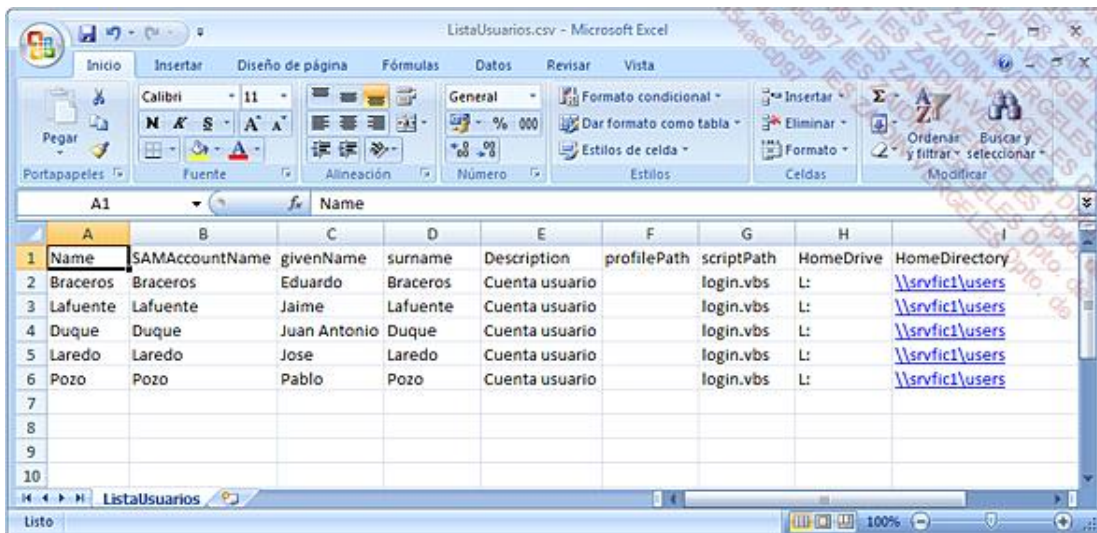
Muy pronto vendrá la vuelta a la escuela, y como cada año vamos a tener más de quinientas cuentas de usuarios por crear para los estudiantes. Pero este año no será un año como los demás, porque, iesta vez automatizaremos esta tarea tan ingrata! Aunque la elaboración del script puede ser algo larga, hay grandes probabilidades de que ahorremos tiempo respecto a una operación manual. Aunque no sea el caso, al menos nos proporcionará una cierta satisfacción personal e intelectual. Por otra parte, estaremos seguros de que todas las cuentas estarán creadas exactamente de la misma manera, lo que evitará un gran número de potenciales errores manuales. Además, podremos reutilizar este script el próximo año...

Lo ideal sería que a partir de un archivo de texto pudiésemos importar los usuarios así como todos sus parámetros asociados; y es lo que vamos a intentar hacer.

Solución

Para responder a esta problemática podríamos imaginar la creación de un archivo Excel donde cada línea contendrá la descripción de un usuario, y en particular los campos siguientes:

- Name: nombre del objeto (se trata de un nombre visible en la consola de administración del Active Directory);
- SAMAccountName: nombre de login;
- Surname: apellido del usuario;
- GivenName: nombre del usuario;
- Description: descripción de la cuenta;
- ProfilePath: ruta a especificar en caso de perfiles interactivos;
- ScriptPath: script de inicio de sesión;
- HomeDrive: letra de conexión del home directory;
- HomeDirectory: ruta de red (en formato UNC) hacia una partición del servidor.



	A	B	C	D	E	F	G	H	I
1	Name	SAMAccountName	givenName	surname	Description	profilePath	scriptPath	HomeDrive	HomeDirectory
2	Braceros	Braceros	Eduardo	Braceros	Cuenta usuario		login.vbs	L:	\\srvfict\users
3	Lafuente	Lafuente	Jaime	Lafuente	Cuenta usuario		login.vbs	L:	\\srvfict\users
4	Duque	Duque	Juan Antonio	Duque	Cuenta usuario		login.vbs	L:	\\srvfict\users
5	Laredo	Laredo	Jose	Laredo	Cuenta usuario		login.vbs	L:	\\srvfict\users
6	Pozo	Pozo	Pablo	Pozo	Cuenta usuario		login.vbs	L:	\\srvfict\users
7									
8									
9									
10									

Creación de usuarios por lotes - Archivo csv

Podremos seguidamente grabar este archivo en formato CSV (*Comma Separated Values*) para obtener un archivo de texto donde los campos estén separados por punto y coma, puesto que Excel no sabe utilizar la coma como separador. El archivo tendría un aspecto similar a este:

```
Name;SAMAccountName;givenName;surname;Description;profilePath;scriptPath;
HomeDrive;HomeDirectory
Braceros;Braceros;Eduardo;Braceros;Cuenta de usuario;;login.vbs;L;;
\\srvfic1\users
Lafuente;Lafuente;Jaime;Lafuente;Cuenta de usuario;;login.vbs;L;;
\\srvfic1\users
Duque;Duque;Juan Antonio;Duque;Cuenta de usuario;;login.vbs;L;;
\\srvfic1\users
Laredo;Laredo;Jose;Laredo;Cuenta de usuario;;login.vbs;L;;\\srvfic1\users
Pozo;Pozo;Pablo;Pozo;Cuenta de usuario;;login.vbs;L;;\\srvfic1\users
```

Solución PowerShell v1

Ahora necesitaremos pasar cada valor de cada línea a un script que se ocupe de todo. Para importar un archivo CSV, disponemos del commandlet `import-csv`, pero el problema es que sólo soporta la coma como separador de campos. Por lo que tendremos que realizar una pequeña transformación en nuestro archivo CSV. Ésta se hará fácilmente con la siguiente instrucción:

```
PS > (Get-Content listaUsuarios.csv) -replace ';' , ',' >
Lista-SepComa.csv
```

Obtenemos el siguiente resultado:

```
Name,SAMAccountName,givenName,surname,Description,profilePath,scriptPath,
HomeDrive,HomeDirectory
Braceros,Braceros,Eduardo,Braceros,Cuenta usuario,,login.vbs,L:,
\\srvfic1\users
Lafuente,Lafuente,Jaime,Lafuente,Cuenta usuario,,login.vbs,L:,
\\srvfic1\users
Duque,Duque,Juan Antonio,Duque,Cuenta usuario,,login.vbs,L:,\\srvfic1\users
Laredo,Laredo,Jose,Laredo,Cuenta usuario,,login.vbs,L:,\\srvfic1\users
Pozo,Pozo,Pablo,Pozo,Cuenta usuario,,login.vbs,L:,\\srvfic1\users
```

Únicamente nos falta hacer la importación del archivo, por ejemplo con el script siguiente:

```
# Create-User.ps1
param (
    [string]$archivo=$(throw 'Debe informar un archivo donde el separador
sea la coma!'),
    [string]$OU='CN=Users')

# Recuperación del dominio actual en formato @ps-scripting.com
$dominioSMTP = "@$env:USERDNSDOMAIN"

# Recuperación del dominio actual en formato @ps-scripting.com
$dominioLDAP=$env:USERDNSDOMAIN

# Transformación de la cadena DNS en cadena de conexión LDAP
# midominio.ps-scripting.com -> midominio,DC=ps-scripting,DC=com
# donde ps-scripting.com -> ps-scripting,DC=com
$dominioLDAP = $dominioLDAP.replace('.', ',DC=')
```

```

# añade DC= al inicio de cadena
# ps-scripting,DC=com -> DC=ps-scripting,DC=com
$dominioLDAP="DC=$dominioLDAP"

$objOU=[ADSI]"LDAP://$OU,$dominioLDAP"

$users = import-csv $archivo
foreach ($user in $users)
{
    # Nombre del objeto Active Directory
    $objUser=$objOU.Create('user', "CN=$( $user.Name)")

    # Definición de SAMAccountName
    $objUser.put('SAMAccountName', $user.SamAccountName)

    # Definición del USPN para el inicio de sesión segura
    # Ejemplo: Ponce@ps-scripting.com
    $objUser.put('userprincipalName', $user.SamAccountName + $dominioSMTP)

    # Nombre
    $objUser.put('GivenName', $user.givenName)

    # Apellido
    $objUser.put('sn', $user.surname)

    # Descripción
    $objUser.put('description', $user.description)

    # Ruta del perfil Windows
    if ($user.profilePath -ne '')
    {
        $objUser.put('profilePath', $user.profilePath)
    }

    # Logon Script
    $objUser.put('scriptPath', $user.scriptPath)

    # Home Drive
    $objUser.put('homeDrive', $user.homeDrive)

    # Home Path
    $objUser.put('homeDirectory', $user.homeDirectory + '\'+
$user.SamAccountName)

    # Creación del home directory
    new-item -path $($user.homeDirectory + '\'+ $user.SamAccountName)
-type directory

    $objUser.SetInfo()
    Write-Host "Usuario $( $user.SamAccountName) creado."
}

```

Utilización:

```
./Create-User.ps1 -archivo Lista-SepComa.csv -OU 'OU=Usuarios'
```

Si no especificamos el parámetro `-ou`, el script creará, por defecto, los usuarios en el contenedor «Users».

Aunque este script pueda mejorarse considerablemente, le ofrece un punto de partida interesante para iniciar la creación masiva de usuarios.

Solución PowerShell v2/Módulo Active Directory Server 2008 R2

Con PowerShell v2 unido a los commandlets introducidos por el módulo Active Directory, verá con qué facilidad responderemos a esta problemática en comparación a la solución anterior...

El hecho de que Excel no sepa guardar archivos CSV con un separador distinto al punto y coma ya no nos molesta porque el commandlet `Import-CSV` está dotado del parámetro `Delimiter` que permite especificar un delimitador distinto a la coma.

Además, con el módulo Active Directory disponemos del comando `New-ADUser`. A éste, cuando le pasamos como entrada de la tubería un objeto con las propiedades correctas creará directamente el usuario. Esto recortará considerablemente nuestro script. Veámoslos:

```
PS > (Import-Csv .\ListaUsuarios.csv -Delimiter ';') | New-ADUser
```

Si se desea crear los usuarios en una unidad organizativa concreta, bastará con detallar el parámetro `Path`, como se muestra en el ejemplo siguiente:

```
PS > (Import-Csv .\ListaUsuarios.csv -Delimiter ';') |  
New-ADUser -Path 'OU=Finanzas,DC=Powershell-scripting,DC=com'
```