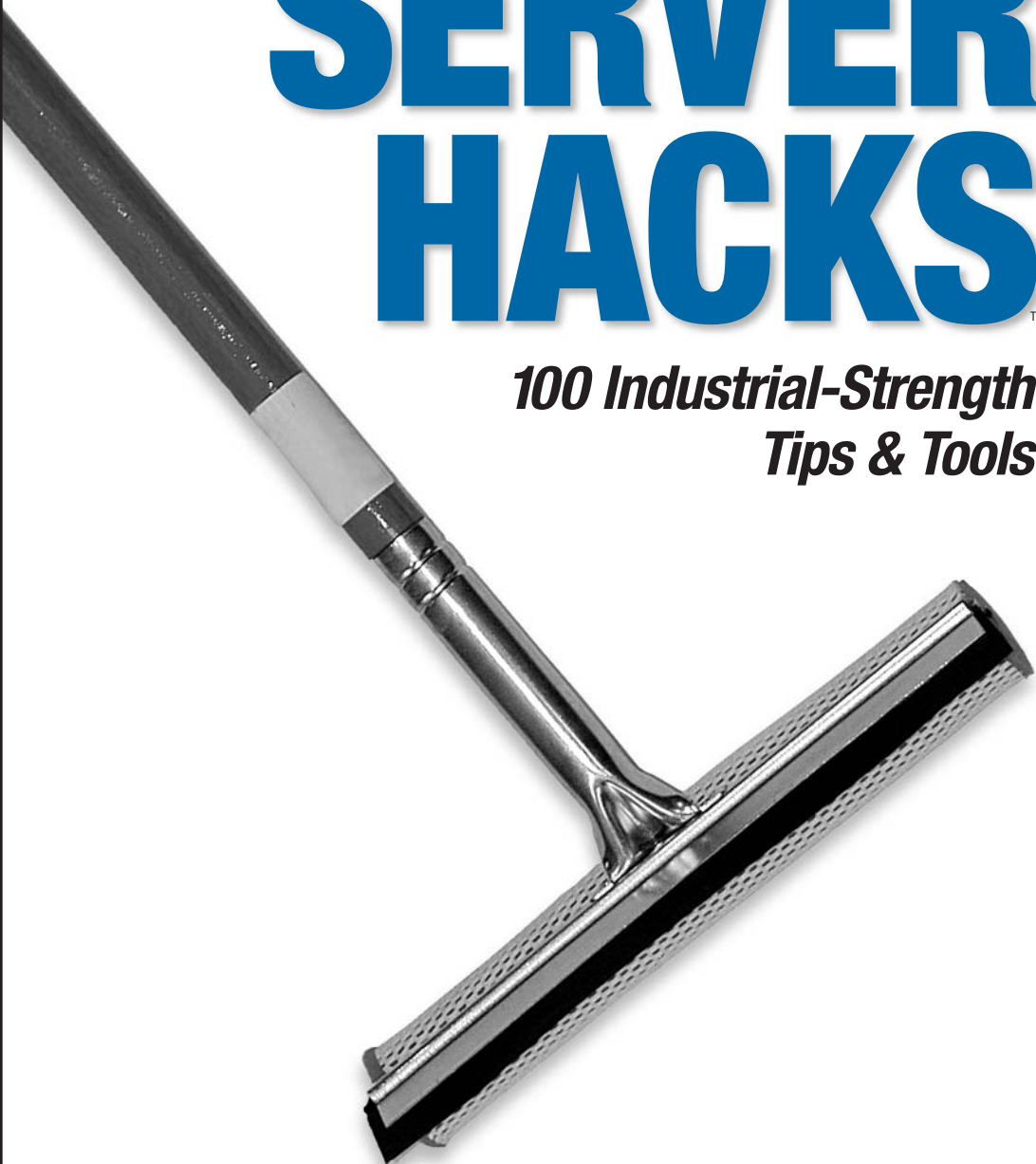


WINDOWS SERVER HACKS™

*100 Industrial-Strength
Tips & Tools*

O'REILLY®

Mitch Tulloch



**HACK**
#18

Automate Creation of OU Structure

Here's a snappy method for creating a standard hierarchy of organizational units (OUs) for a domain.

If you manage deployment of Active Directory in a medium-sized or large organization, you probably are spending a significant amount of time trying to maintain consistency in the Active Directory hierarchy. Even within a single domain, it typically makes sense to keep your organizational units (OUs) structured according to some agreed-upon rules. Regardless of whether your top-tier OU design is based on functional, business, geographic, or some other criteria, you will likely benefit from keeping the lower tiers arranged in the same fashion. This way, for example, you can formulate standard operating procedures that will apply across the entire organization. You can also attempt to automate some of the common administrative tasks, such as user, group, or computer account creation; script delegations and permission assignments; and group policy object management on the OU level.

One of the ways to make sure that the structure will remain consistent throughout Active Directory deployment is to script the OU-creation process. The script in this hack creates a sample OU hierarchy. The assumption is that the top-level OUs are created manually, while the lower layers are always the same. The structure follows Microsoft best practices and includes two second-tier OUs: Accounts and Resources. The Accounts OU is further divided into Users, ServiceAccounts, Groups, and Admins. Resources consists of Workstations and Servers. It is fairly easy to extend this structure (for example, you could create separate OUs for different server types, such as File, Print, or TerminalServices, beneath the Servers OU). The script performs some error checking to verify that the respective organizational units haven't been created yet.

The Code

The following VBScript is a Windows script (*.wsf) file, a text document that contains Extensible Markup Language (XML) code. Using a text editor such as Notepad (with Word Wrap turned off) type the following code and save it as *CreateOU.wsf*:

```
<?xml version="1.0"?>
<job id="CreateOUs">
<script language="VBscript">
<![CDATA[

'*****
'*** The script creates OU structure underneath top level OU
'*** Second level: Accounts and Resources
```

```

'*** Third level:
'*** Accounts children OUs - Users, ServiceAccounts, Groups, Admins
'*** Resources children OUs - Workstations, Servers
'***
'*** To execute, run cscript.exe //nologo CreateOUs.wsf OUName
'*** where OUName is the name of the top level OU
Option Explicit

Dim strOU1 'the first level OU
Dim strOU2 'the second level OU
Dim strOU3 'the third level OU
Dim arrOUTier2 'array of the second level OUs
Dim arrOUTier3a 'first array of the third level OUs
Dim arrOUTier3b 'second array of the third level OUs

Dim strDomainDN 'name of the domain
Dim strADsPath 'ADsPath of the first level OU
Dim strADsSubPath 'ADsPath of the second level OU
Dim adsRootDSE 'adsRootDSE object
Dim adsContainer, adsSubContainer, adsOU
'variables representing AD container objects

'*****
'*** Connect to the current domain

Set adsRootDSE = GetObject("LDAP://rootDSE")
strDomainDN = adsRootDSE.Get("defaultNamingContext")

'*****
'*** Connect to the top level OU

strOU1 = WScript.Arguments(0)
strADsPath = "LDAP://OU=" & strOU1 & "," & strDomainDN
Set adsContainer = GetObject(strADsPath)

On Error Resume Next

arrOUTier2 = Array("Accounts", "Resources")
arrOUTier3a = Array("Users", "ServiceAccounts", "Groups", "Admins")
arrOUTier3b = Array("Workstations", "Servers")

'*****
'*** Populate the OU structure

For Each strOU2 in arrOUTier2

Set adsOU = adsContainer.Create("OrganizationalUnit", "OU=" & strOU2)
adsOU.SetInfo
If ErrCheck(Err, strOU2) <> 2 Then

strADsSubPath = "LDAP://OU=" & strOU2 & ",OU=" & strOU1 & "," & strDomainDN
Set adsSubContainer = GetObject(strADsSubPath)

Select Case strOU2
Case "Accounts"

```

```
For Each strOU3 in arrOUTier3a
Set adsOU = adsSubContainer.Create("OrganizationalUnit", "OU=" & strOU3)
adsOU.SetInfo
Call ErrCheck(Err, strOU3)
Next
Case "Resources"
For Each strOU3 in arrOUTier3b
Set adsOU = adsSubContainer.Create("OrganizationalUnit", "OU=" & strOU3)
adsOU.SetInfo
Call ErrCheck(Err, strOU3)
Next
End Select

End If

Next

On Error GoTo 0

Set adsOU = Nothing
Set adsContainer = Nothing

'*****
'*** Error checking function

Function ErrCheck(objErr, strObj)

If objErr.Number <> 0 Then
'if the object already exists
If objErr.Number = &H80071392 Then
WScript.Echo "The OU " & strObj & " already exists"
ErrCheck = 1
Else
WScript.Echo "Unexpected error " & objErr.Description
ErrCheck = 2
End If

Else

ErrCheck = 0

End If

objErr.Clear

End Function

]]>
</script>
</job>
```

Running the Hack

To execute the script, open a command prompt, change to the directory in which *CreateOUs.wsf* resides, and type `cscript.exe //nologo CreateOUs.wsf "OUName"`, where *OUName* is the name of the top-level OU. If *OUName* does not already exist, you'll get an error. To illustrate how this script works, I first created an OU named Boston in the *mtit.com* domain and then ran `cscript.exe //nologo CreateOUs.wsf "Boston"` from the command line. Figure 2-4 shows the result in Active Directory Users and Computers.

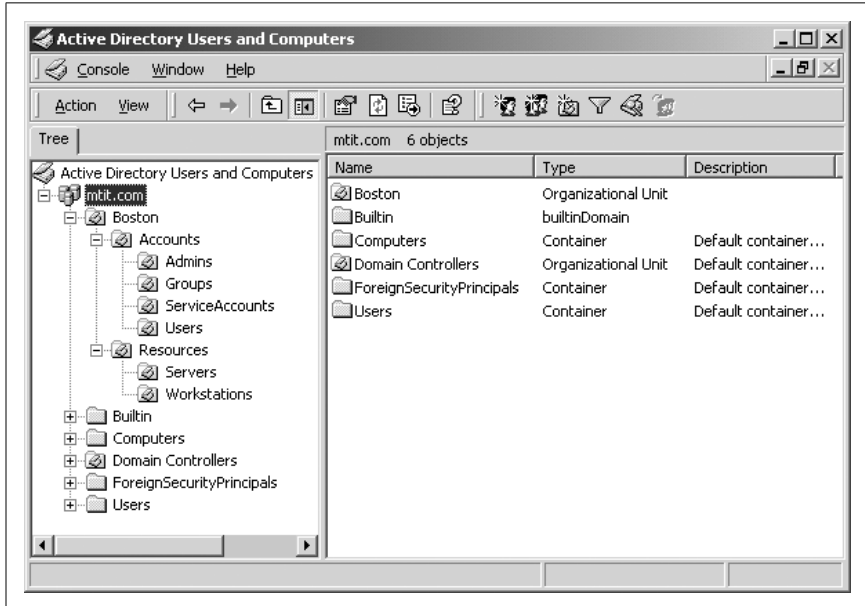


Figure 2-4. OU hierarchy for Boston

—Marcin Policht