Keenan's Brief Guide to the Active Directory Recycle Bin in Windows Server 2008 R2

Since Active Directory was included as part of Window Server 2000, administrators have often asked for a simple way to roll back mistakes, whether that is the incorrect deletion of the wrong user account to the accidental removal of thousands of objects by deleting an OU. Before the release of Windows Server 2008 R2 there were a number of ways using built-in or third-party methods to restore Active Directory objects, but typically they were not as quick or complete as say retrieving a deleted email or file.

Microsoft has included with their release of Windows Server 2008 R2 the facility, under the correct conditions, to enable a Recycle Bin for Active Directory and allow simple restoration of objects which have been erroneously removed. In this article we will briefly cover some of the options prior to 2008 R2 and then examine how to enable the new Recycle Bin and restore objects from it.

Pre-Windows Server 2008 R2

The 2008 R2 Recycle Bin for Active Directory is a great motivating point for upgrading your forest and domain(s) to the latest version, but this is not always a quick process in many enterprises so it is worth knowing what options are available prior to this version. Like many things it's a lot better to examine and plan for possible resolutions before a significant mistake happens that you need to deal with. Retrieving Active Directory objects typically falls into two available categories, authoritative restore from a backup or tombstone reanimation.

Authoritative Restore

The Microsoft KB article 840001(http://support.microsoft.com/kb/840001) details how to perform the restoration of a user account using a system state backup of a domain controller. Typically, you would use a global Catalog so that you can also restore all group membership information.

Tombstone Reanimation

The above article also details how to recover an account when you don't have a system state backup by using tombstone reanimation which was introduced with Windows Server 2003 – you can retrieve objects from the Deleted Objects container where they are kept after deletion until their tombstone period expires. Obviously regular system state backups of Active Directory are critical for your full disaster recovery procedures, but taking advantage of tombstone reanimation means you can get objects back quicker than having to go through the full authoritative restore process.

You could use the procedure in the article which utilises the ldp.exe tool, but there are other methods around which you may find simpler.

The drawback with tombstone reanimation is that because most of the object's attributes are removed at the time of the object's deletion, a restored object using this method requires many properties of the account, such as address fields and group membership, to be manually repopulated. Whilst this is obviously preferable to re-creating an account from scratch it does not make for a quick overall process. However, you will at least get back the objectGUID and objectSid attributes which means there would be no need to re-configure a user's workstation profile.

The original release of Windows Server 2008 introduced snapshot backups for Active Directory. You can take point-in-time snapshots of your Active Directory with the **NTDSUTIL** command line utility which utilizes Volume Shadow Copy to provide a snapshot. It is then possible to mount this snapshot using different ports on the same domain controller as the live Active Directory database and use standard tools to compare the two. This could really make the tombstone reanimation a lot simpler because after restoring the object you could view two versions of Active Directory Users and Computers side by side and view the properties of the restored object from a previous time, so making it simpler to repopulate properties. The Directory Service Comparison Tool (http://lindstrom.nullsession.com/?page_id=11) takes advantage of these snapshots and makes the repopulation process more streamlined.

For those with Microsoft Exchange messaging environments, once you have the Active Directory account back, you can use the Reconnect Mailbox feature within Exchange to tie the restored account back up with the mailbox. This is of course providing you have a similar tombstone retention period for mailboxes that you do for AD accounts.

Active Directory Recycle Bin

The key differences from previous versions of Windows Server are that by default you get all of the attributes back and the tools to use are PowerShell cmdlets, which are quickly becoming a more essential part of every Windows administrator's standard toolkit.

Firstly though the Active Directory Recycle Bin is not enabled by default and has certain domain and forest wide requirements before it can be enabled.

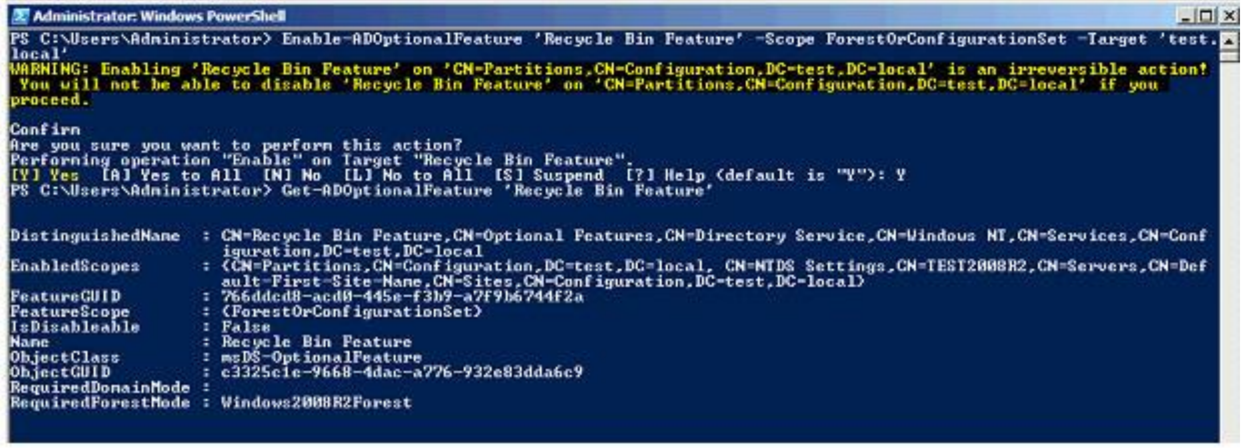
Firstly, all domain controllers within the Active Directory forest must be running Windows Server 2008 R2.

Secondly, the functional level of the Active Directory forest must be Windows Server 2008 R2.

Enabling the recycle Bin

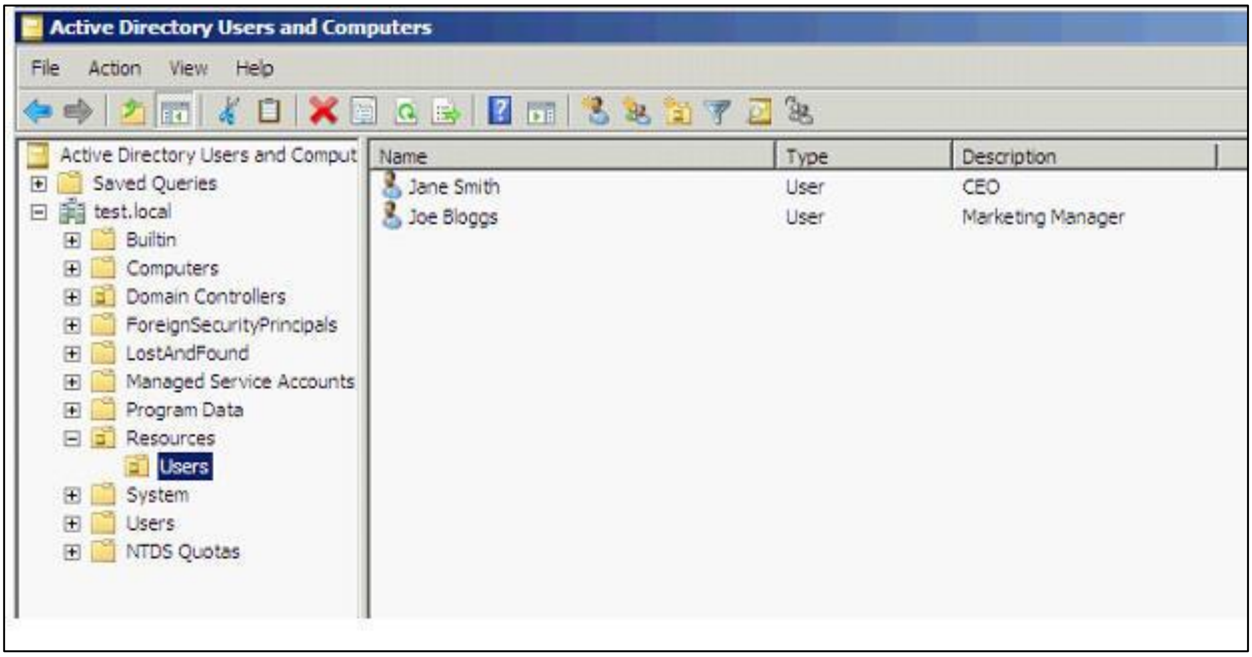
Now that our forest is at the correct functional level we can enable the Recycle Bin, to do so we use the **Enable-ADOptionalFeature** cmdlet. This must be either run on the DC with the Domain Naming Master FSMO role or directed at that server with the **–server** parameter. Again you will be prompted to confirm your command since the action is irreversible.

```
PS> Enable-ADOptionalFeature 'Recycle Bin Feature' -Scope ForestOrConfigurationSet -
target 'test.local'
```



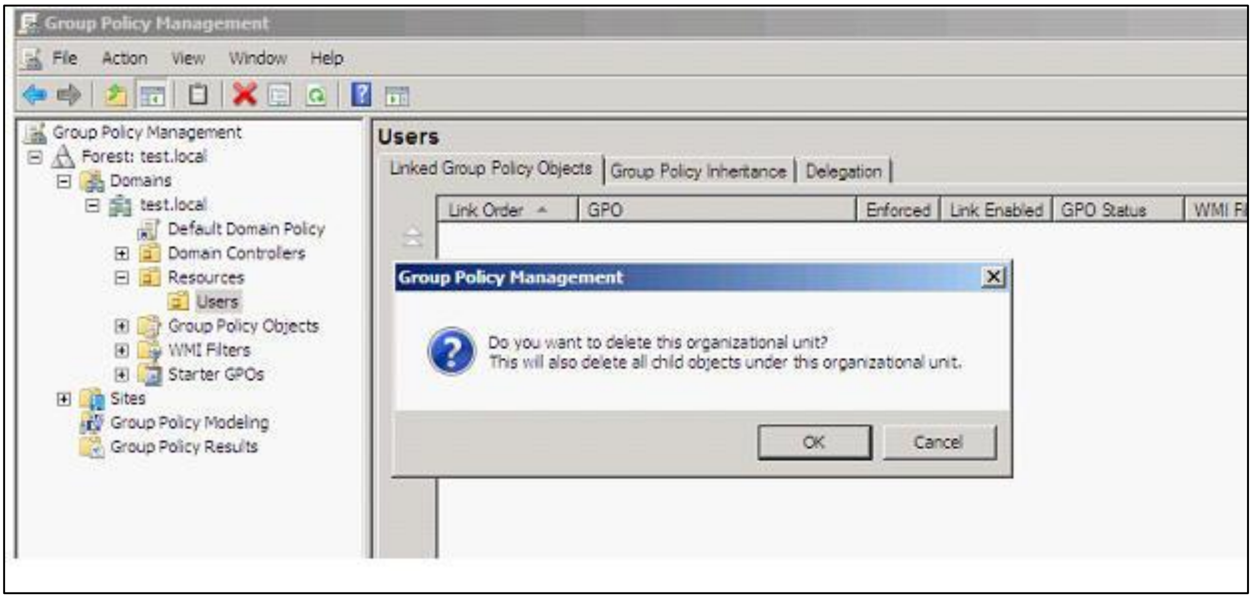
```
PS C:\Users\Administrator> Enable-ADOptionalFeature 'Recycle Bin Feature' -Scope ForestOrConfigurationSet -Target 'test.
local'
WARNING: Enabling 'Recycle Bin Feature' on 'CN=Partitions,CN=Configuration,DC=test,DC=local' is an irreversible action!
 You will not be able to disable 'Recycle Bin Feature' on 'CN=Partitions,CN=Configuration,DC=test,DC=local' if you
proceed.

Confirm
Are you sure you want to perform this action?
Performing operation "Enable" on Target "Recycle Bin Feature".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): Y
PS C:\Users\Administrator> Get-ADOptionalFeature 'Recycle Bin Feature'


DistinguishedName  : CN=Recycle Bin Feature,CN=Optional Features,CN=Directory Service,CN=Windows NT,CN=Services,CN=Conf
                     iguration,DC=test,DC=local
EnabledScopes      : {CN=Partitions,CN=Configuration,DC=test,DC=local, CN=NTDS Settings,CN=TEST2008R2,CN=Servers,CN=Def
                     ault-First-Site-Name,CN=Sites,CN=Configuration,DC=test,DC=local}
FeatureGUID        : 766ddcd8-acd0-445e-f3b9-a7f9b6744f2a
FeatureScope       : {ForestOrConfigurationSet}
IsDisableable      : False
Name               : Recycle Bin Feature
ObjectClass        : msDS-OptionalFeature
ObjectGUID         : c3325c1e-9668-4dac-a776-932e83dda6c9
RequiredDomainMode :
RequiredForestMode : Windows2008R2Forest
```

Now that we have the Recycle Bin enabled it’s time to go check out how we recover some deleted objects. In this environment we have a very simple AD structure with a couple of test accounts to illustrate the example.



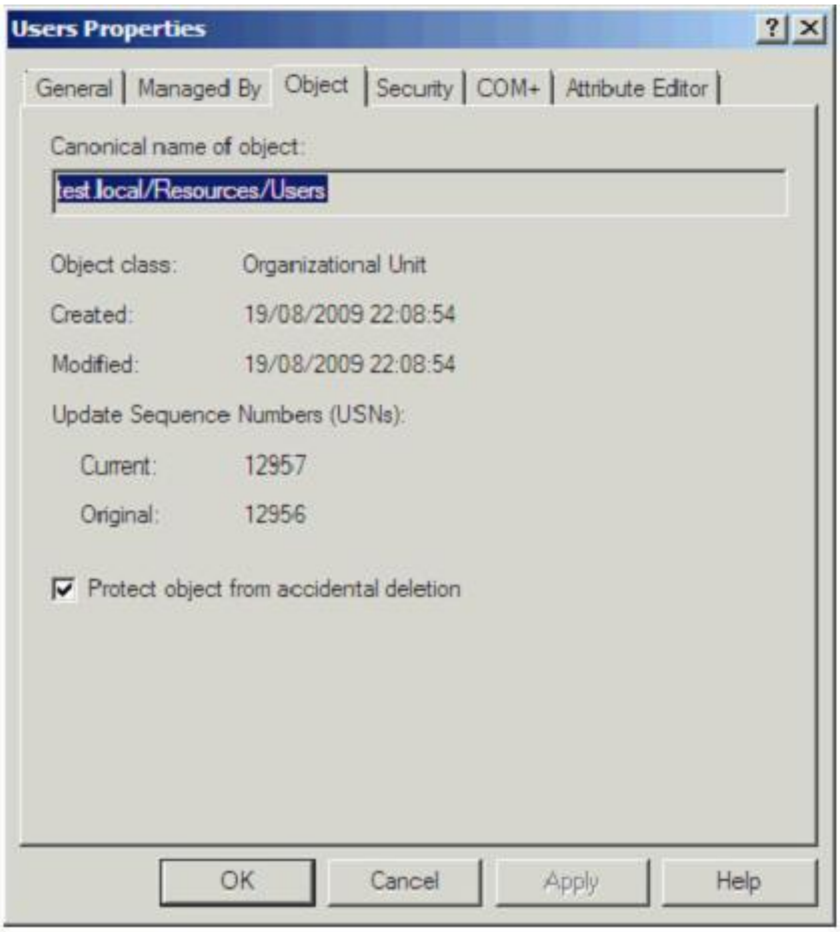
Name	Type	Description
Jane Smith	User	CEO
Joe Bloggs	User	Marketing Manager

Let’s take the situation where an administrator accidently deletes the **Users** OU. One of the most common reasons this can happen is because it is actually possible to delete OU’s from the Group Policy Management tool, not just Active Directory Users and Computers – so an administrator might think they are removing a GPO and in a bad moment delete the wrong item and remove a whole OU. The administrator is prompted for what they are about to do, but I have seen it happen more than once!



Do you want to delete this organizational unit?
This will also delete all child objects under this organizational unit.

The initial release of Windows 2008 Server actually included a new checkbox ‘Protect object from accidental deletion’. In the example of the OU below any attempt to delete the OU will be met with an **Access is denied** response and the administrator will actually have to remove the tick from that checkbox before the OU can be deleted.



Canonical name of object:
test.local/Resources/Users

Object class: Organizational Unit

Created: 19/08/2009 22:08:54

Modified: 19/08/2009 22:08:54

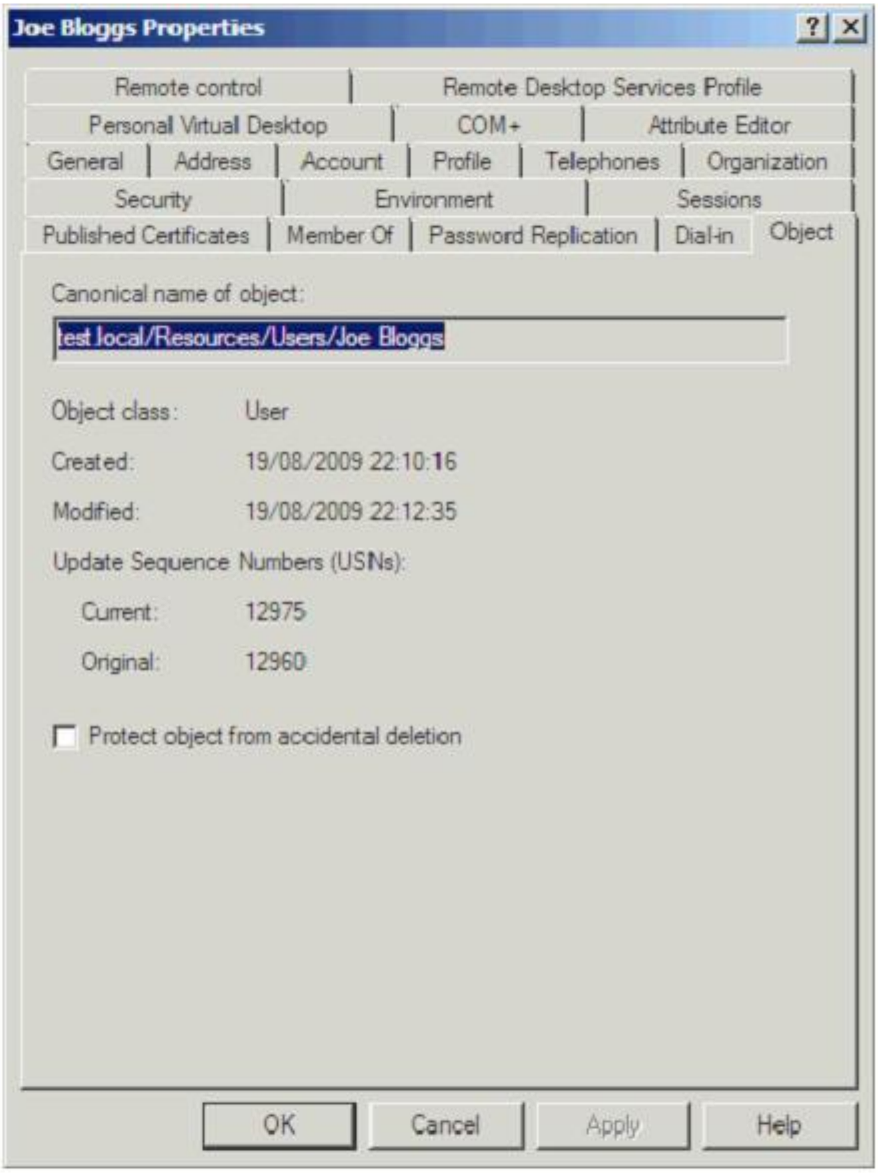
Update Sequence Numbers (USNs):

Current: 12957

Original: 12956

☑ Protect object from accidental deletion

However, what you would naturally expect to happen as a consequence of the **Protect object from accidental deletion** would be any user or computer account created in that protected OU would also be supported by the same mechanism. Unfortunately by default they are not, so as a good practise you would either need to build that into your account creation process or programmatically check and set that checkbox on all accounts in the OU on a regular basis.



Canonical name of object:

test.local/Resources/Users/Joe Bloggs

Object class: User

Created: 19/08/2009 22:10:16

Modified: 19/08/2009 22:12:35

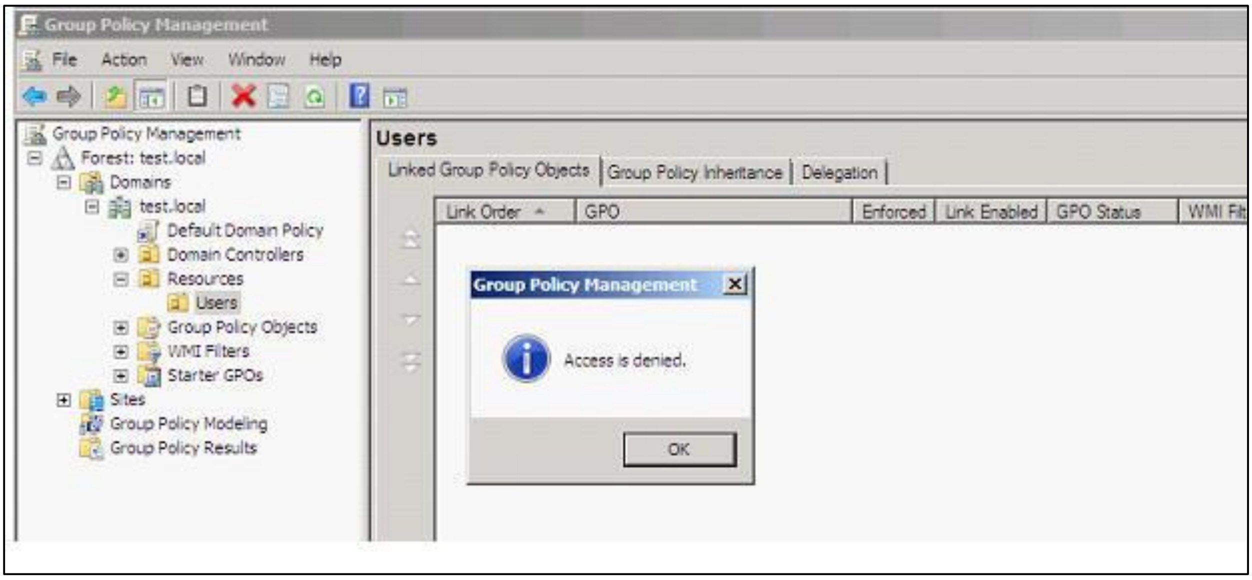
Update Sequence Numbers (USNs):

Current: 12975

Original: 12960

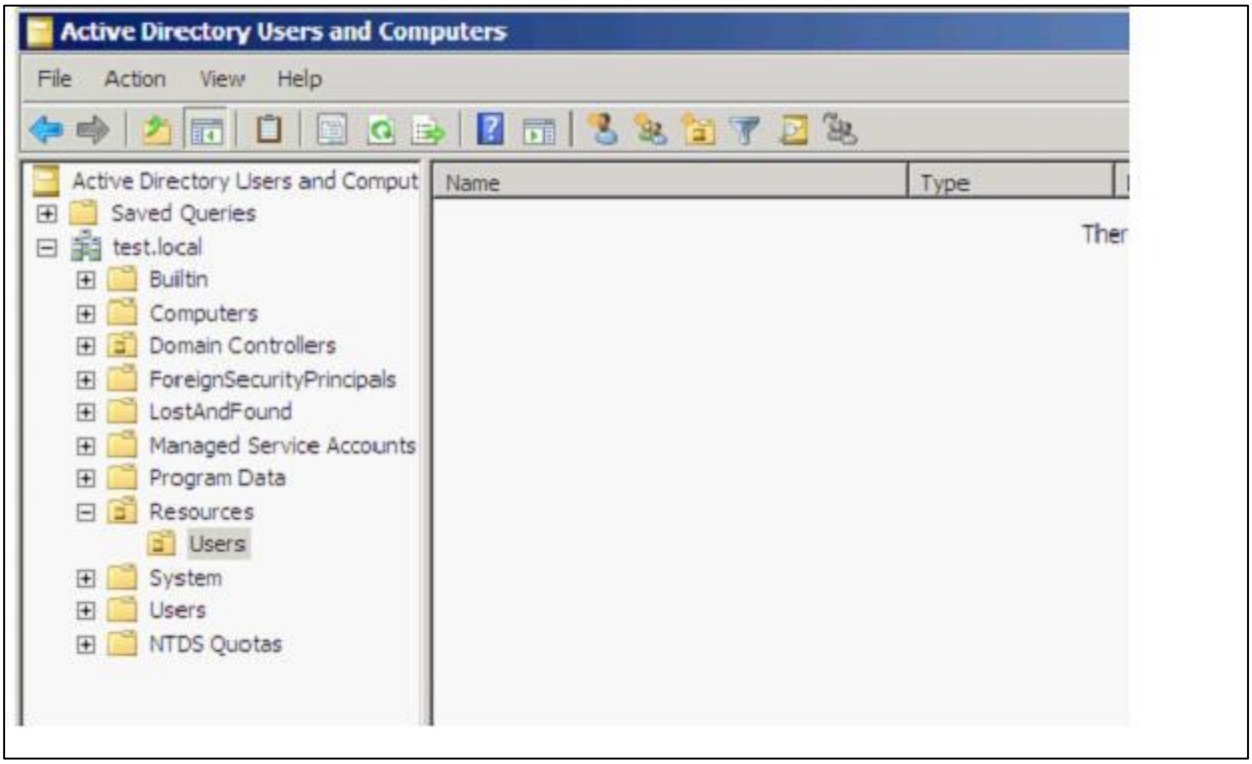
□ Protect object from accidental deletion

Consequently, in the above example if we accept the warning to delete the OU we are greeted with an **Access is denied** message since the OU has protection set.



Access is denied.

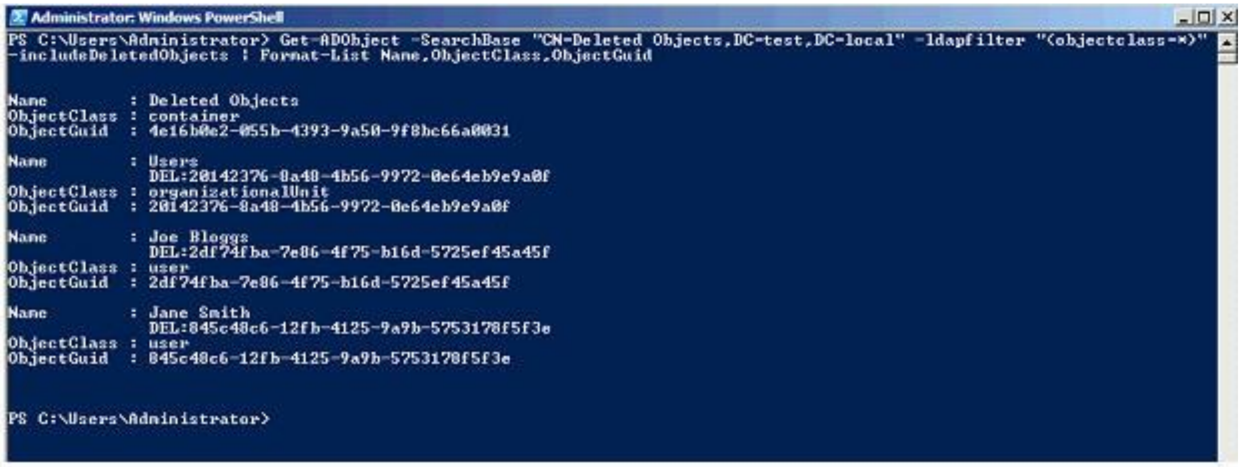
So we were saved from deleting the OU, but all of the unprotected child objects were deleted.



(For the purposes of this guide I will now remove the **Users** OU by first clearing the checkbox for protecting the object from accidental deletion.)

We can browse the current contents of the Active Directory Recycle Bin using the **Get-ADObject** cmdlet, directing it at the **Deleted Objects** container and using the **–includeDeletedObjects** parameter.

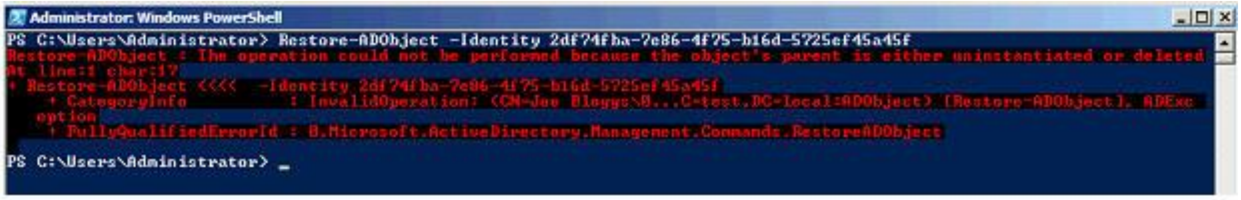
```
PS> Get-ADObject –SearchBase “CN=Deleted Objects,DC=test,DC=local” –ldapFilter
“(objectClass=*)” -includeDeletedObjects | Format-List Name,ObjectClass,ObjectGuid
```



```
PS C:\Users\Administrator> Get-ADObject -SearchBase "CN=Deleted Objects,DC=test,DC=local" -ldapfilter "(objectclass=*)"
-includeDeletedObjects | Format-List Name,ObjectClass,ObjectGuid


Name        : Deleted Objects
ObjectClass : container
ObjectGuid  : 4e16b0e2-055b-4393-9a50-9f8bc66a0031

Name        : Users
              DEL:20142376-8a48-4b56-9972-0e64eb9e9a0f
ObjectClass : organizationalUnit
ObjectGuid  : 20142376-8a48-4b56-9972-0e64eb9e9a0f

Name        : Joe Bloggs
              DEL:2df74fba-7e86-4f75-b16d-5725ef45a45f
ObjectClass : user
ObjectGuid  : 2df74fba-7e86-4f75-b16d-5725ef45a45f

Name        : Jane Smith
              DEL:845c48c6-12fb-4125-9a9b-5753178f5f3e
ObjectClass : user
ObjectGuid  : 845c48c6-12fb-4125-9a9b-5753178f5f3e


PS C:\Users\Administrator>
```

We can see from the resultant output that we have both the **Users** OU in there and the two user accounts. So let's try restoring one of the user accounts back, to do so we need the **Restore-ADObject** cmdlet and supply the ObjectGuid property of the user account.

```
PS> Restore-ADObject –identity 2df74fba-7e86-4f75-b16d-5725ef45a45f
```



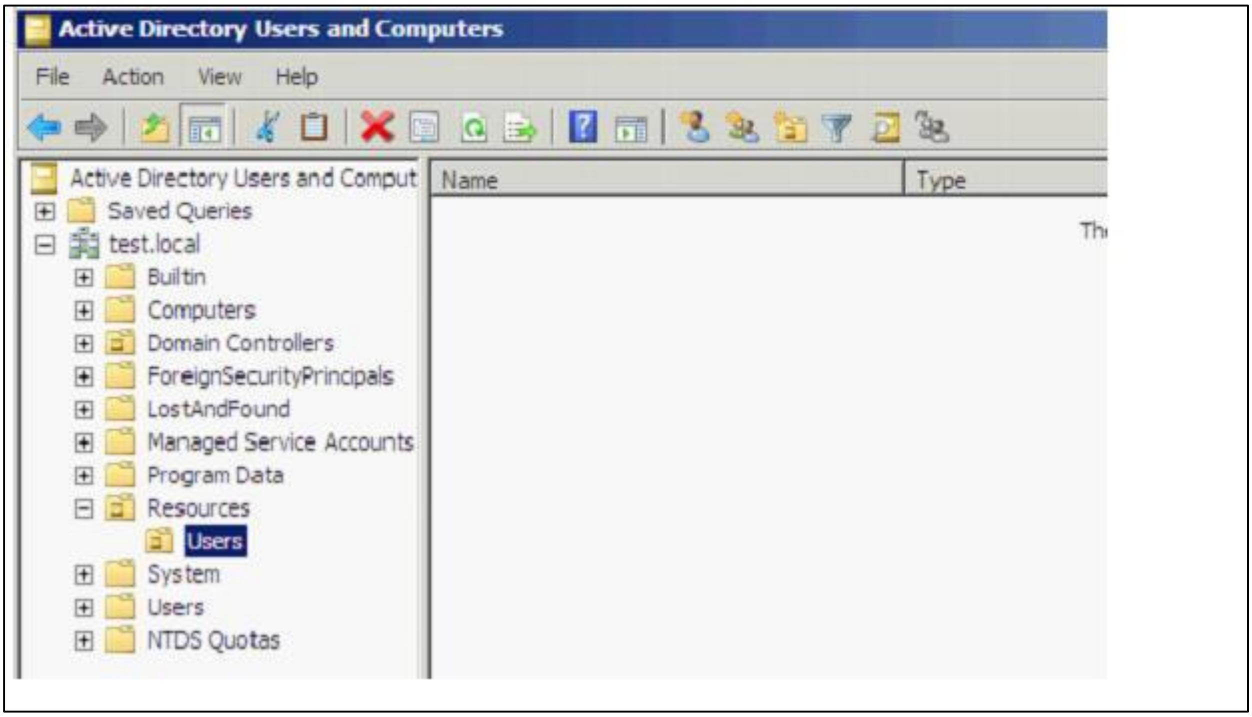
```
PS C:\Users\Administrator> Restore-ADObject -Identity 2df74fba-7e86-4f75-b16d-5725ef45a45f
Restore-ADObject : The operation could not be performed because the object's parent is either uninstantiated or deleted
At line:1 char:17
+ Restore-ADObject <<<<  -Identity 2df74fba-7e86-4f75-b16d-5725ef45a45f
    + CategoryInfo          : InvalidOperation: (CN=Joe Bloggs\0...C=test,DC=local:ADObject) [Restore-ADObject], ADExc
   eption
    + FullyQualifiedErrorId : 0,Microsoft.ActiveDirectory.Management.Commands.RestoreADObject

PS C:\Users\Administrator> _
```

It failed to restore, but PowerShell tells us that it failed because the object's parent no longer exists either, i.e. we need to first restore the **Users** OU. (Note: an alternative would be to use the -targetpath parameter and re-direct the restore to a different OU)

To restore the **Users** OU we can use the same cmdlet (**Restore-ADObject**) as to restore users, just supply the ObjectGuid of the OU.

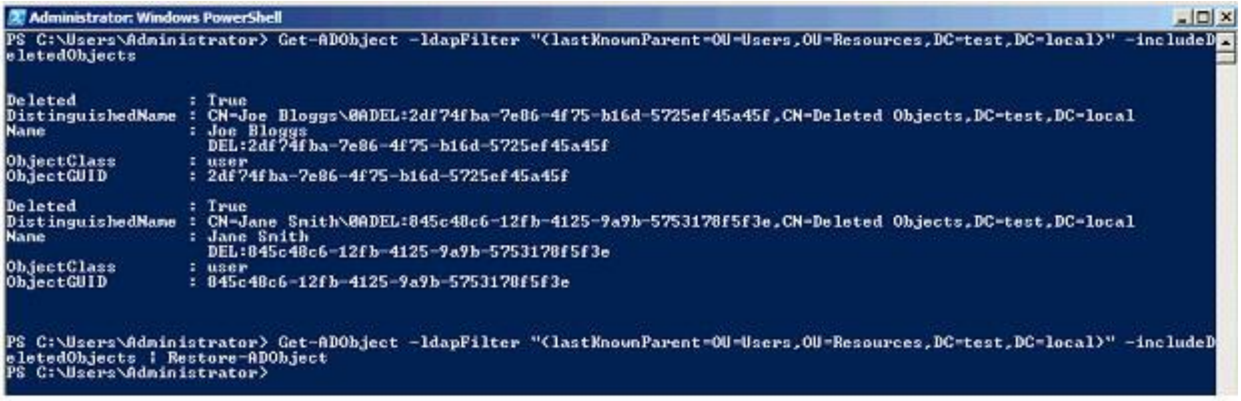
```
PS> Restore-ADObject –identity 20142376-8a48-4b56-9972-0e64eb9e9a0f
```

The **Users** OU returns.



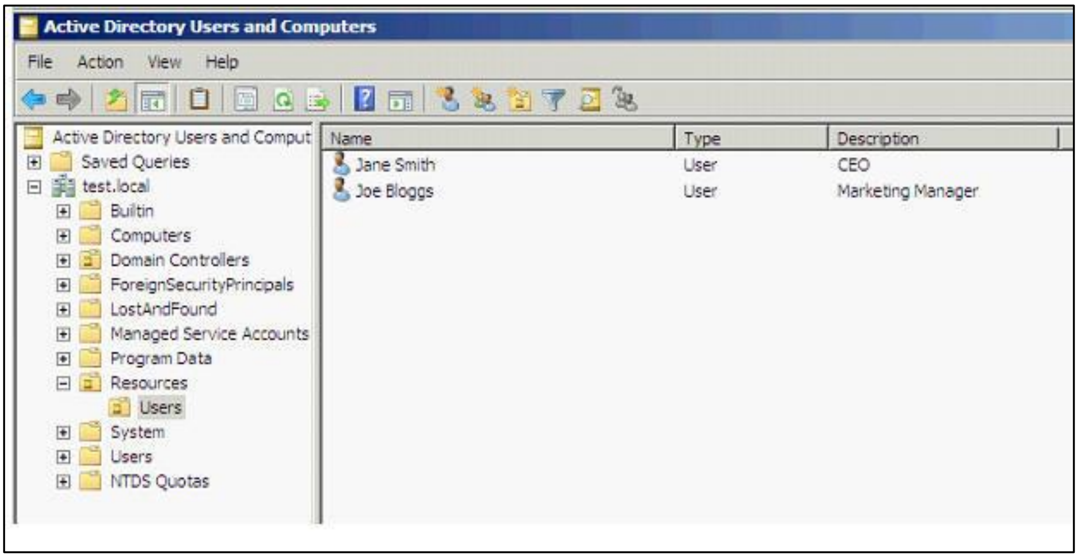
Now we just need to get those user accounts back. Rather than have to type out the ObjectGuid for each account we wish to restore we can instead create a search which will match all of the accounts we wish to restore and then use the PowerShell pipeline to send those results to the **Restore-ADObject** cmdlet.

```
PS> Get-ADObject -ldapFilter
"(lastKnownParent=OU=Users,OU=Resources,DC=test,DC=local)" -includeDeletedObjects |
Restore-ADObject
```



```
PS C:\Users\Administrator> Get-ADObject -ldapFilter "(lastKnownParent=OU=Users,OU=Resources,DC=test,DC=local)" -includeD
eletedObjects


Deleted           : True
DistinguishedName : CN=Joe Bloggs\0ADEL:2df74fba-7e86-4f75-b16d-5725ef45a45f,CN=Deleted Objects,DC=test,DC=local
Name              : Joe Bloggs
                    DEL:2df74fba-7e86-4f75-b16d-5725ef45a45f
ObjectClass       : user
ObjectGUID        : 2df74fba-7e86-4f75-b16d-5725ef45a45f

Deleted           : True
DistinguishedName : CN=Jane Smith\0ADEL:845c48c6-12fb-4125-9a9b-5753178f5f3e,CN=Deleted Objects,DC=test,DC=local
Name              : Jane Smith
                    DEL:845c48c6-12fb-4125-9a9b-5753178f5f3e
ObjectClass       : user
ObjectGUID        : 845c48c6-12fb-4125-9a9b-5753178f5f3e



PS C:\Users\Administrator> Get-ADObject -ldapFilter "(lastKnownParent=OU=Users,OU=Resources,DC=test,DC=local)" -includeD
eletedObjects | Restore-ADObject
PS C:\Users\Administrator>
```

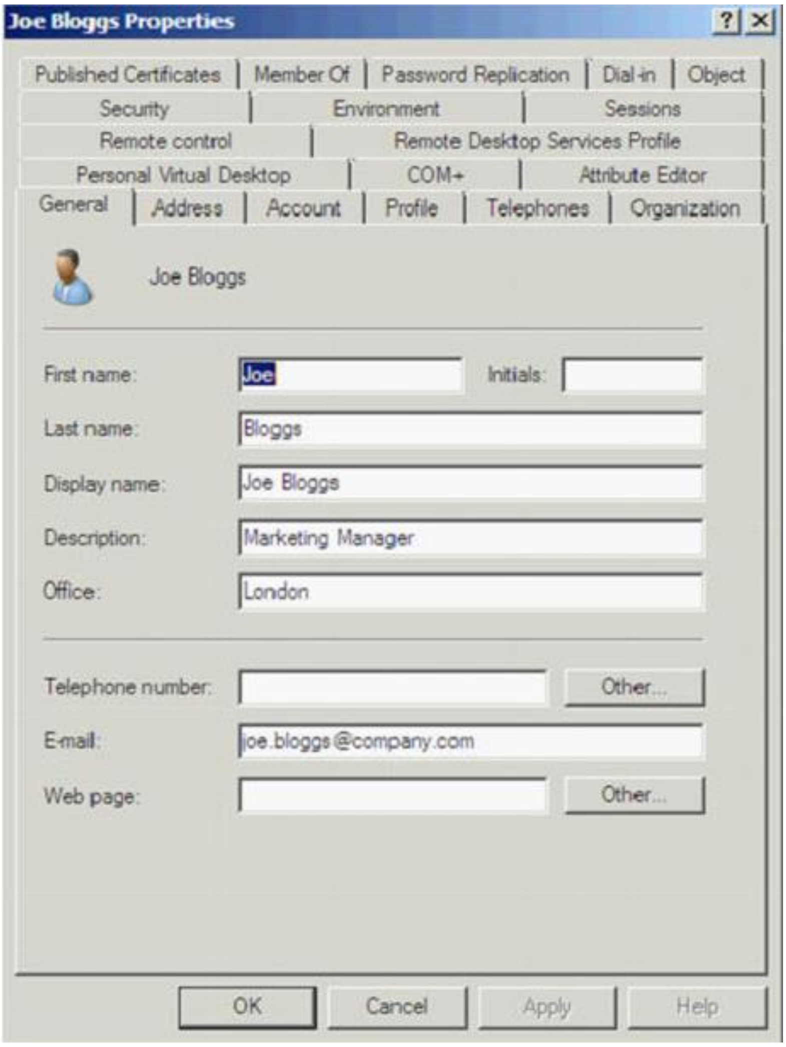
The user accounts are back in the **Users** OU.



- Active Directory Users and Comput
  - Saved Queries
  - test.local
    - Builtin
    - Computers
    - Domain Controllers
    - ForeignSecurityPrincipals
    - LostAndFound
    - Managed Service Accounts
    - Program Data
    - Resources
      - Users
    - System
    - Users
    - NTDS Quotas

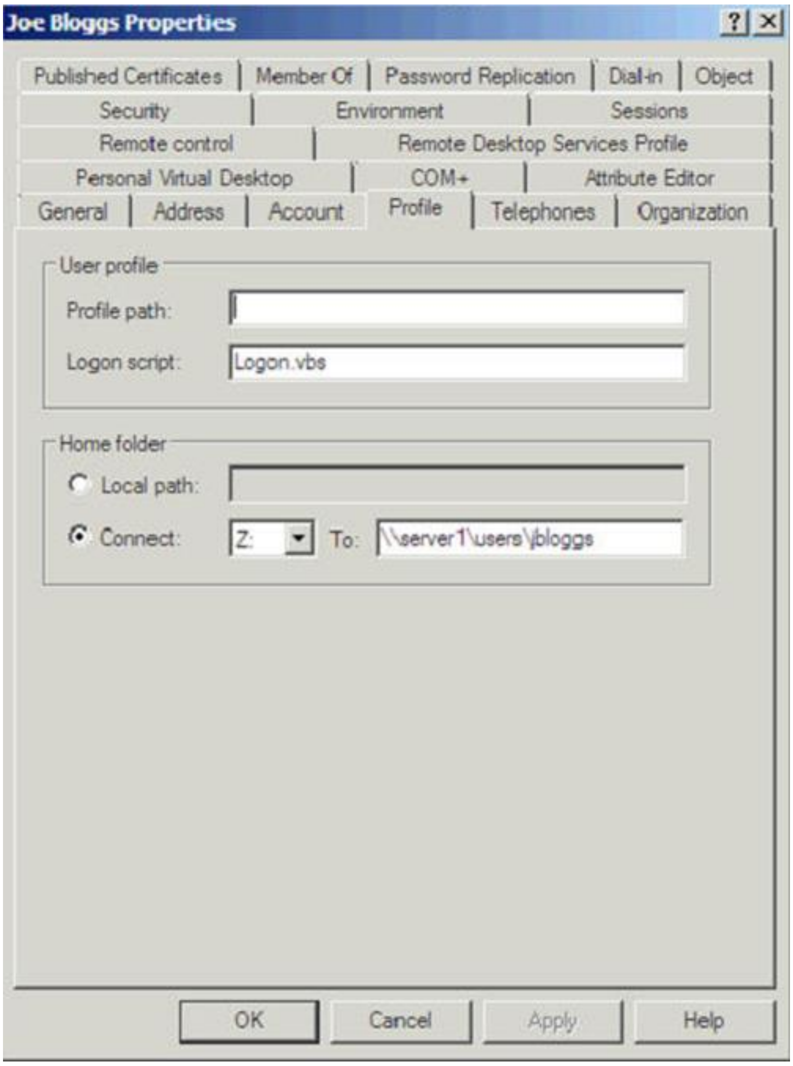
Name	Type	Description
Jane Smith	User	CEO
Joe Bloggs	User	Marketing Manager

If we check the properties of the account we can confirm that different from tombstone re-animation we get all of the properties back.



Joe Bloggs

First name: Joe
Initials:
Last name: Bloggs
Display name: Joe Bloggs
Description: Marketing Manager
Office: London
Telephone number:
E-mail: joe.bloggs@company.com
Web page:



User profile

Profile path:

Logon script: Logon.vbs

Home folder

○ Local path:

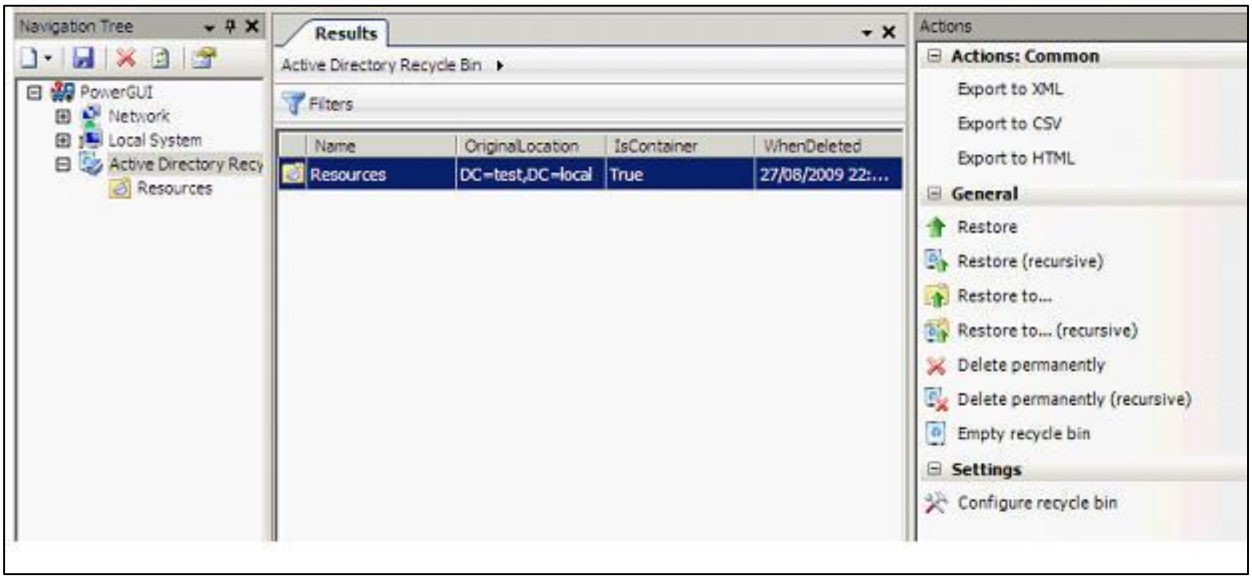
● Connect: Z: To: \\server1\users\jbloggs

Active Directory Recycle Bin PowerPack for PowerGUI

Although the Recycle Bin is a great new feature within Windows Server 2008 R2 Microsoft is already getting feedback that there is no GUI for managing it. Whilst a lot of administrators are comfortable with PowerShell, some may still prefer to use a GUI based management tool for these tasks.

Fortunately a great tool to plug this gap has already been provided by the community; PowerShell MVP Kirk Munro has created the Active Directory Recycle Bin PowerPack for PowerGUI (http://www.powergui.org/entry.jspa?categoryID=21&externalID=2461). This free tool has bundled up scripts using the previously demonstrated Active Directory PowerShell cmdlets and provides a graphical front end for administration.

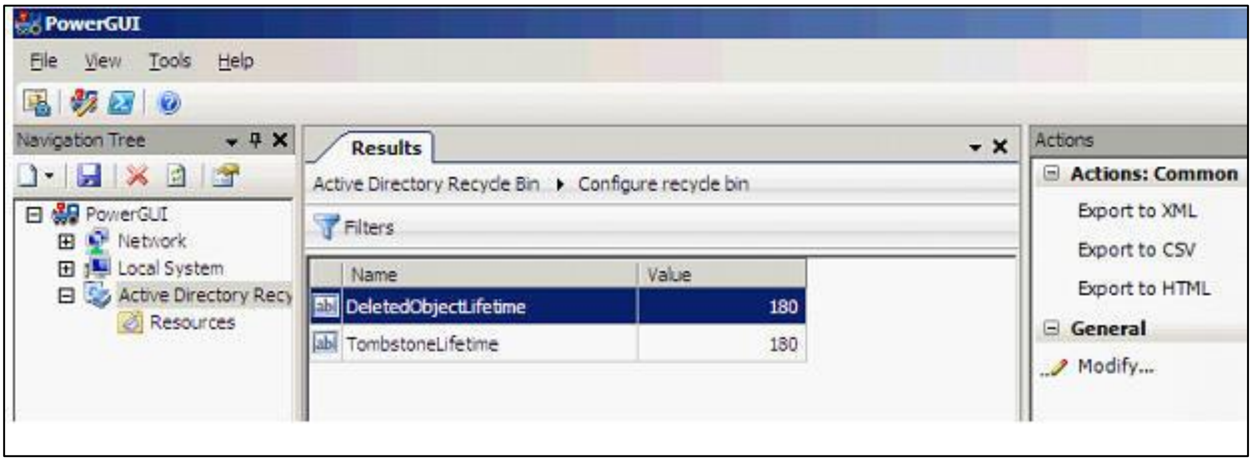
Simply download the PowerGUI tool plus the Active Directory Recycle Bin PowerPack and import it into PowerGUI. Open up the PowerPack and you will have a graphical view of the current contents of the Recycle Bin with the ability to drill down through Organisational Units. Options for restoring single items or recursively are provided in the Actions column as well as alternate restoration paths and emptying items from the Recycle Bin.



Active Directory Recycle Bin

Name	OriginalLocation	IsContainer	WhenDeleted
Resources	DC=test,DC=local	True	27/08/2009 22:...

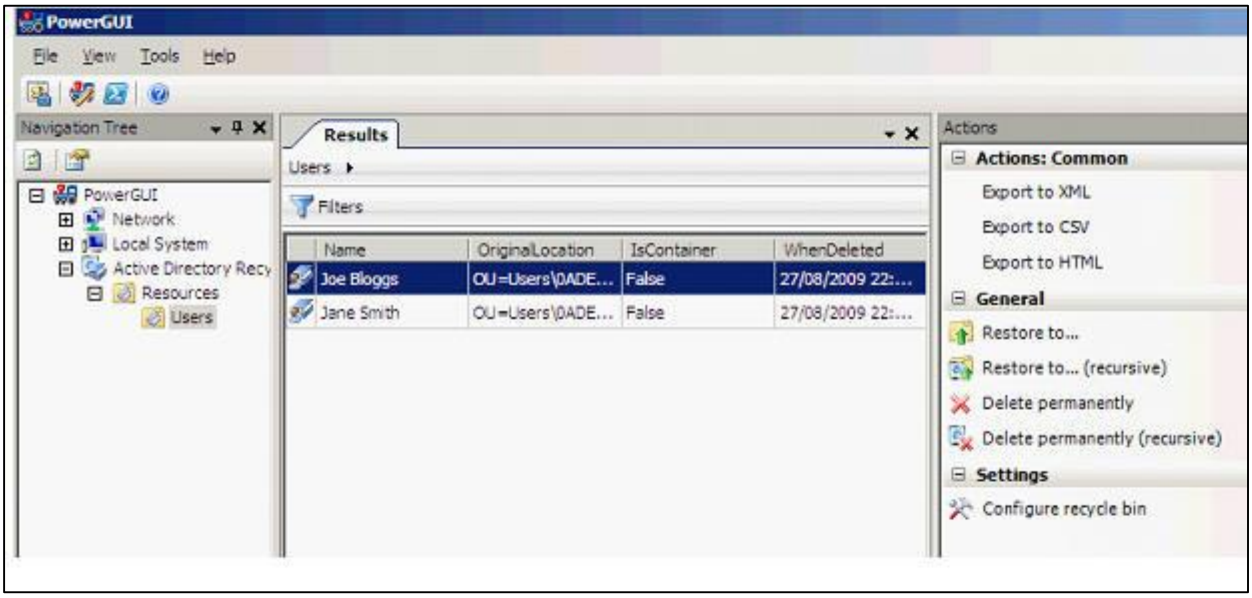
It is also possible to use the **Configure recycle bin** action to set the values for **DeletedObjectLifetime**, the amount of days objects reside in the Recycle Bin, and **TombstoneLifetime**, the amount of days objects can be restored using Tombstone Reanimation after they have left the Recycle Bin. In Windows Server 2008 R2 both of these values default to 180 days, in some earlier versions of Windows Server this value was 60 days and if you upgrade those domain controllers it will remain the same so you may wish to change the values – you can use the **Modify** action to do this.



Active Directory Recycle Bin ▸ Configure recycle bin

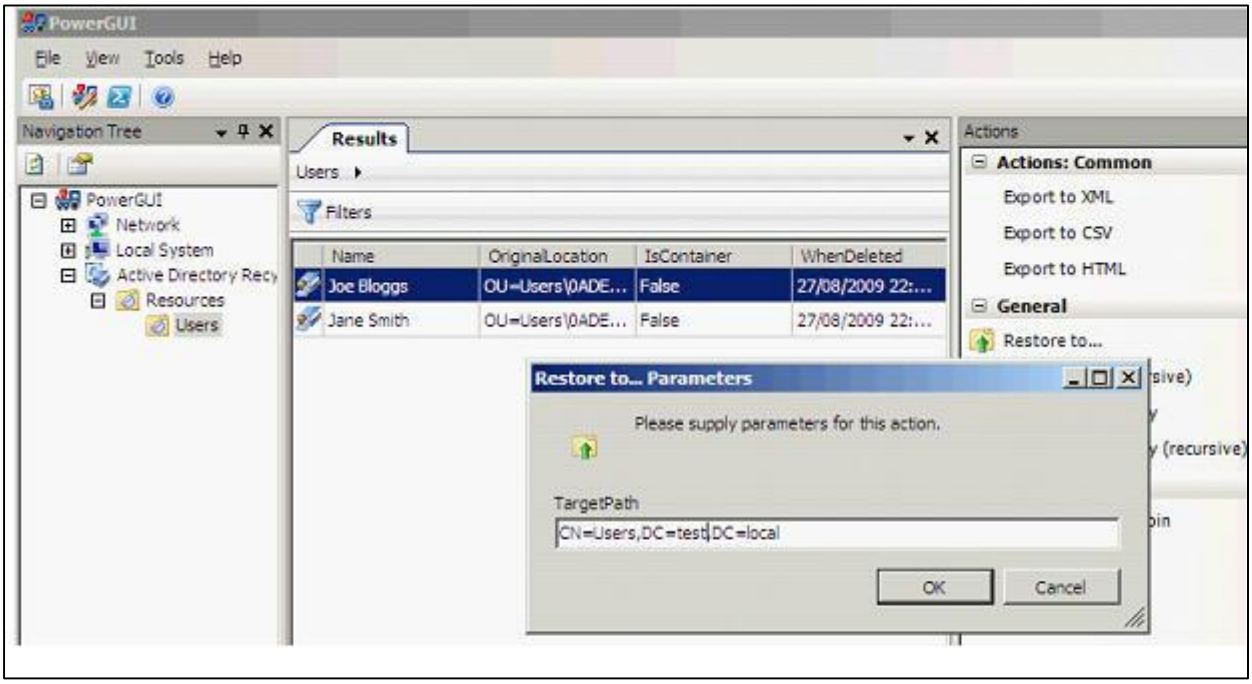
Name	Value
DeletedObjectLifetime	180
TombstoneLifetime	180

For this example I have deleted from Active Directory the **Resources** and **Users** containers and the two user accounts which you can see nicely in the below screenshot using PowerGUI.



Name	OriginalLocation	IsContainer	WhenDeleted
Joe Bloggs	OU=Users\0ADE...	False	27/08/2009 22:...
Jane Smith	OU=Users\0ADE...	False	27/08/2009 22:...

This time we will restore the account **Joe Bloggs**, but to an alternative location using the **Restore to....** Action. (Remember: this is done in PowerShell using the **–targetpath** parameter of the **Restore-ADObject cmdlet**) Simply input the path to the Organisational Unit you wish to restore the object to. In this example we use the default **Users** container as the target location.



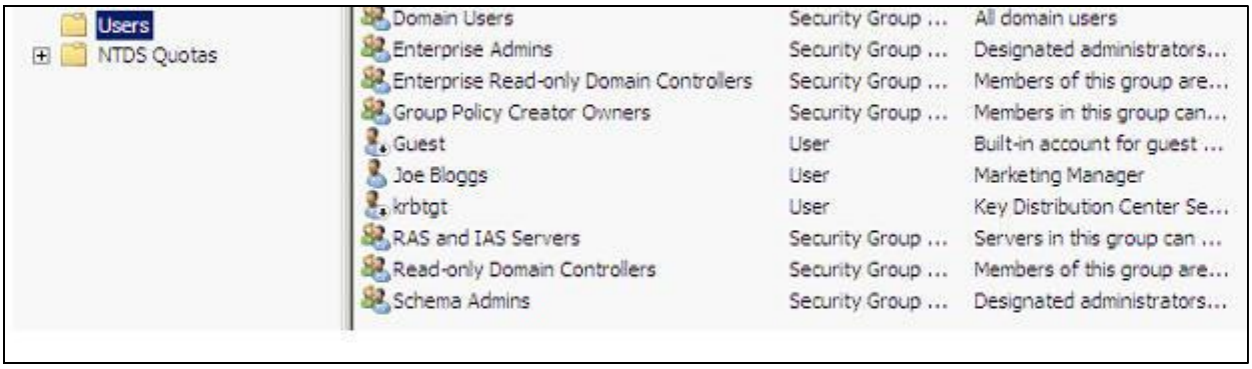
Name	OriginalLocation	IsContainer	WhenDeleted
Joe Bloggs	OU=Users\0ADE...	False	27/08/2009 22:...
Jane Smith	OU=Users\0ADE...	False	27/08/2009 22:...

Restore to... Parameters

Please supply parameters for this action.

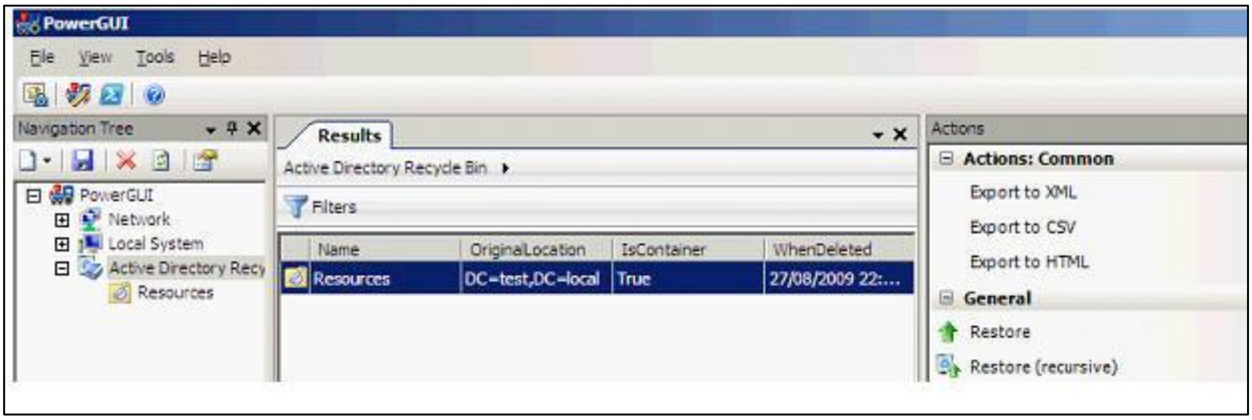
TargetPath: CN=Users,DC=test,DC=local

The user has been restored to the alternate location; this is particularly useful if we did not wish to bring back the entire OU(s) as we did previously.



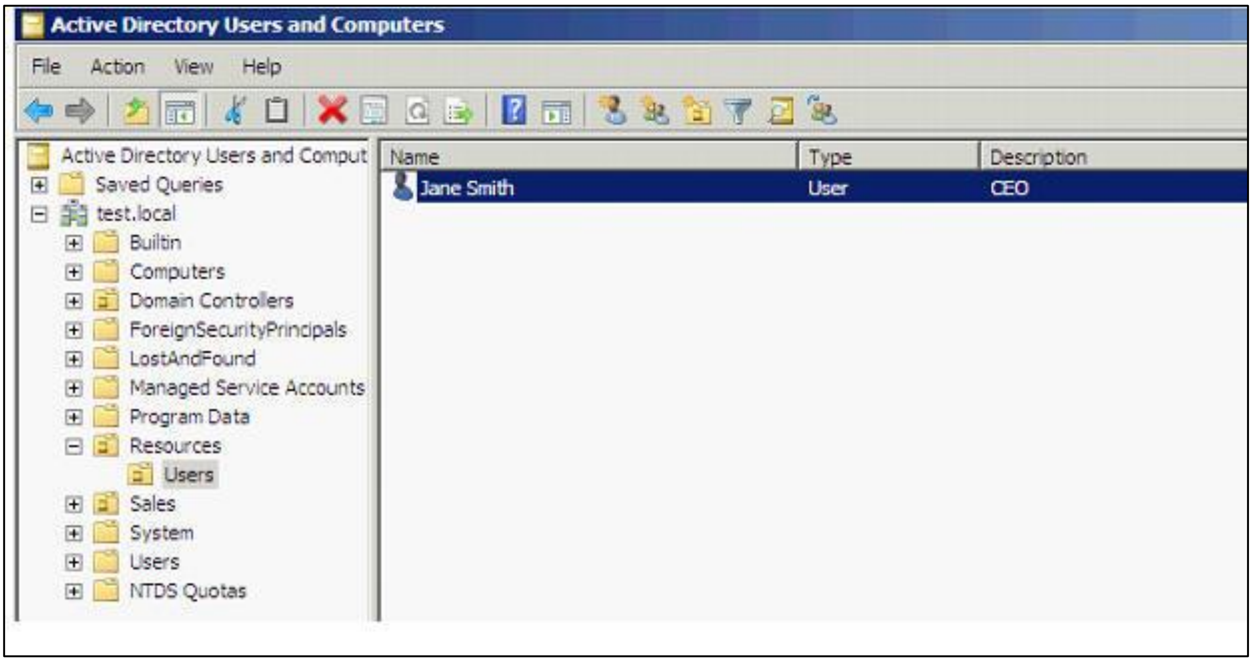
Domain Users	Security Group ...	All domain users
Enterprise Admins	Security Group ...	Designated administrators...
Enterprise Read-only Domain Controllers	Security Group ...	Members of this group are...
Group Policy Creator Owners	Security Group ...	Members in this group can...
Guest	User	Built-in account for guest ...
Joe Bloggs	User	Marketing Manager
krbtgt	User	Key Distribution Center Se...
RAS and IAS Servers	Security Group ...	Servers in this group can ...
Read-only Domain Controllers	Security Group ...	Members of this group are...
Schema Admins	Security Group ...	Designated administrators...

If however, you do wish to bring back the contents of an entire OU and everything below it there is an action, **Restore (recursive)**.



Name	OriginalLocation	IsContainer	WhenDeleted
Resources	DC=test,DC=local	True	27/08/2009 22:...

Using the **Restore (recursive)** action in this scenario brings back both the **Resources** and **Users** OU's as well as the single account remaining in it, **Jane Smith**.



Name	Type	Description
Jane Smith	User	CEO

Hopefully in a future release of Windows Server this functionality will be provided out of the box, the most natural home would be a viewable container within Active Directory Users and Computers, until then the Recycle Bin PowerPack for PowerGUI will prove very useful.

Summary

One of the most requested features for a long time with Active Directory has been a Recycle Bin. Microsoft has finally delivered this with the release of Windows Server 2008 R2. It may not be a feature that enterprises get to use for a little while given the system requirements of all 2008 R2 Domain Controllers and your Active Directory Forest at 2008 R2 functional level, but it could be one of those compelling reasons that enables you to pursue an upgrade.

Administration is via the new Active Directory PowerShell cmdlets which Microsoft is using to provide a consistent command line interface across all of their products. Although currently there is no native GUI for these administration tasks, the Active Directory Recycle Bin PowerPack for PowerGUI enables administrators to leverage the underlying PowerShell functionality and provide a graphical interface for carrying out these tasks.