WELCOME TO TECH | IMMERSION

Track: The Active Directory Recycle Bin

Presenter: Brian McCann

Global Platforms Engineer - Brian@Intel.com

Agenda

- What the AD Recycle Bin (ADRB) can do and requirements needed to use it
- Seeing deleted objects with LDP and PowerShell
- Pre R2 FFL: Reanimation with LDP
- How the ADRB works
- Enabling the ADRB
- Taking the Trash out with PowerShell

Who Cares About the AD Recycle Bin (ADRB)?

- So we've deleted a user, a couple of users, or perhaps a whole OU full of users
- We need to undelete them
- There has always been the "standard" way
  - Reboot the DC in DSRM
  - Restore the AD
  - Use NTDSUTIL to mark items as "authoritatively restored"
  - Reboot the DC in normal mode

Not Quite the Best Approach

- What else happens when you "take the DC offline"
  - Significant amount of time to reboot a DC in large organizations
  - Small shops may only have 1 DC
  - Silly apps like Exchange installed on the same Server
  - Access to backups may…where did we put that tape again?
  - Change Control
  - Static Mappings (SAMBA)

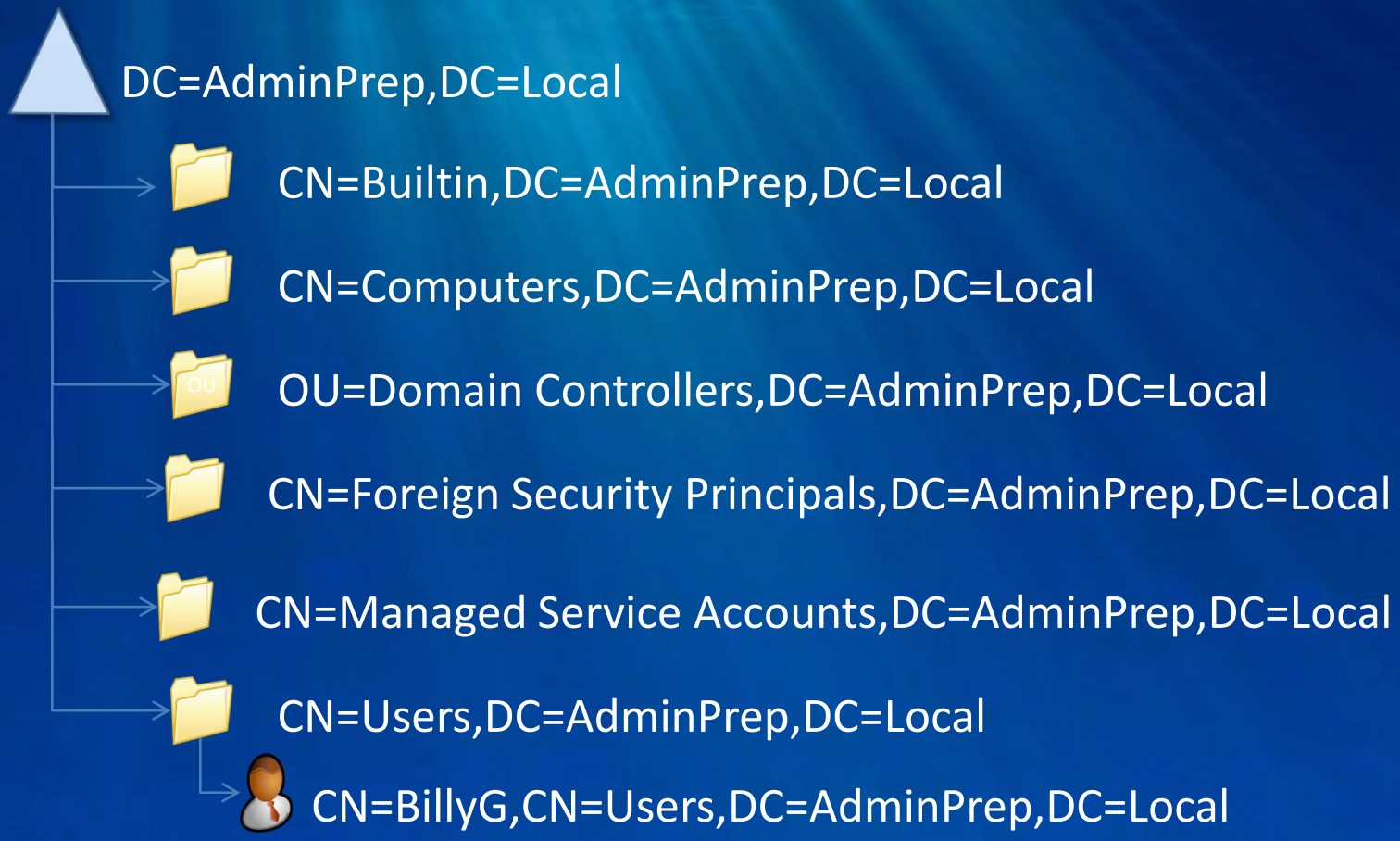
Deletion, Through the Years

- In Windows 2000, the death of an object was very nearly a final thing; undeletion was complicated, and offered no help in re-joining groups
  - 60 Day lifetime
- Things got better in 2003, with "tombstone reanimation" support, which partially undeleted accounts, but left most attributes and group memberships gone
  - 2003 SP1+ – 180 Day lifetime
- Microsoft really didn't care about AD in 2008 and thus no need features!
- With 2008 R2, you can undelete a deleted item, but requires 2008 R2 FFL

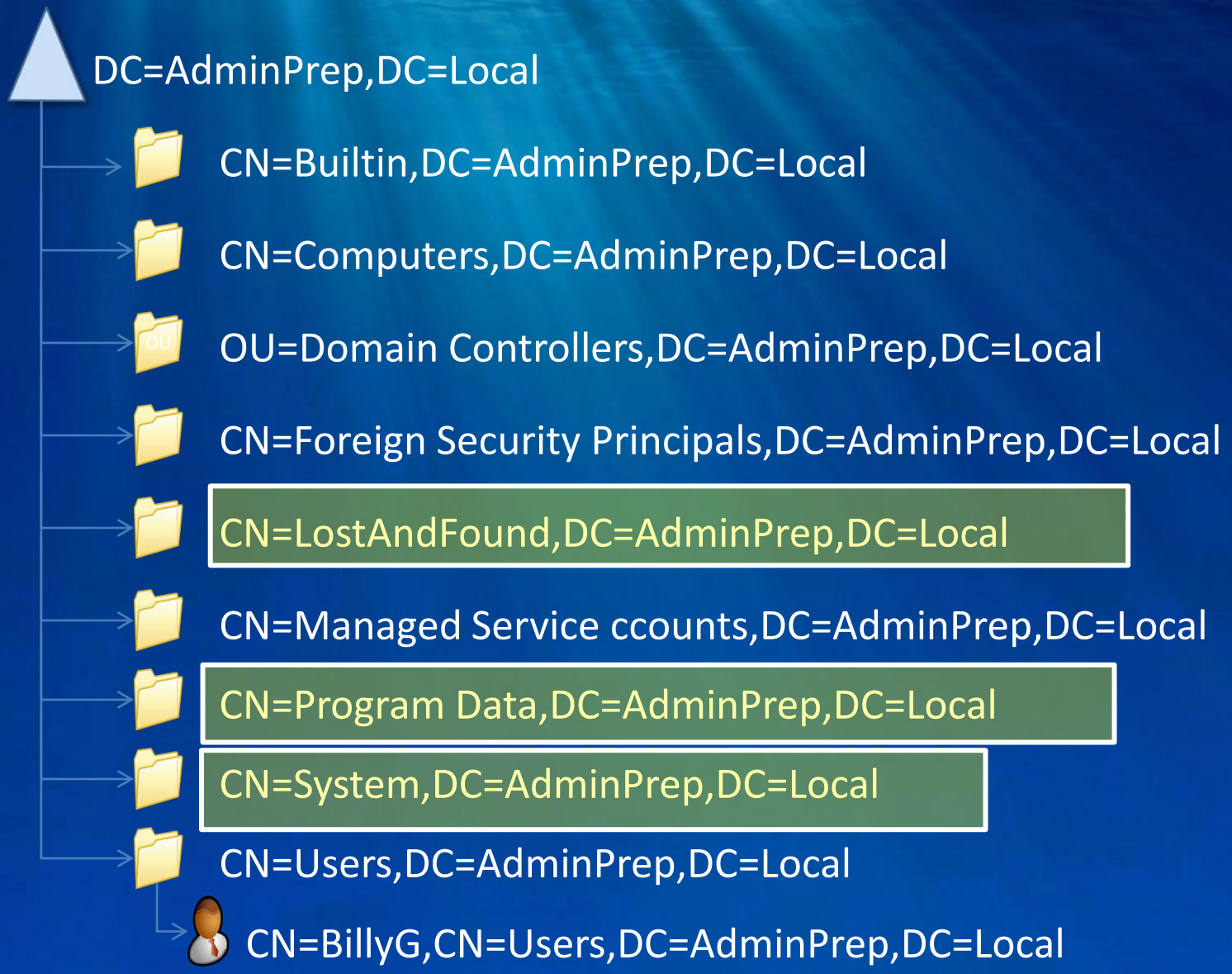
Deleted Stuff "Goes to Limbo"

- You're used to seeing some set of folders in Active Directory Users and Computers
- But you probably know that if you click View / Advanced Features, you see more
- Well, there's even *more* that you still can't see, including an important folder named "Deleted Objects"
- So let's look at what your AD contains, versus what it shows you

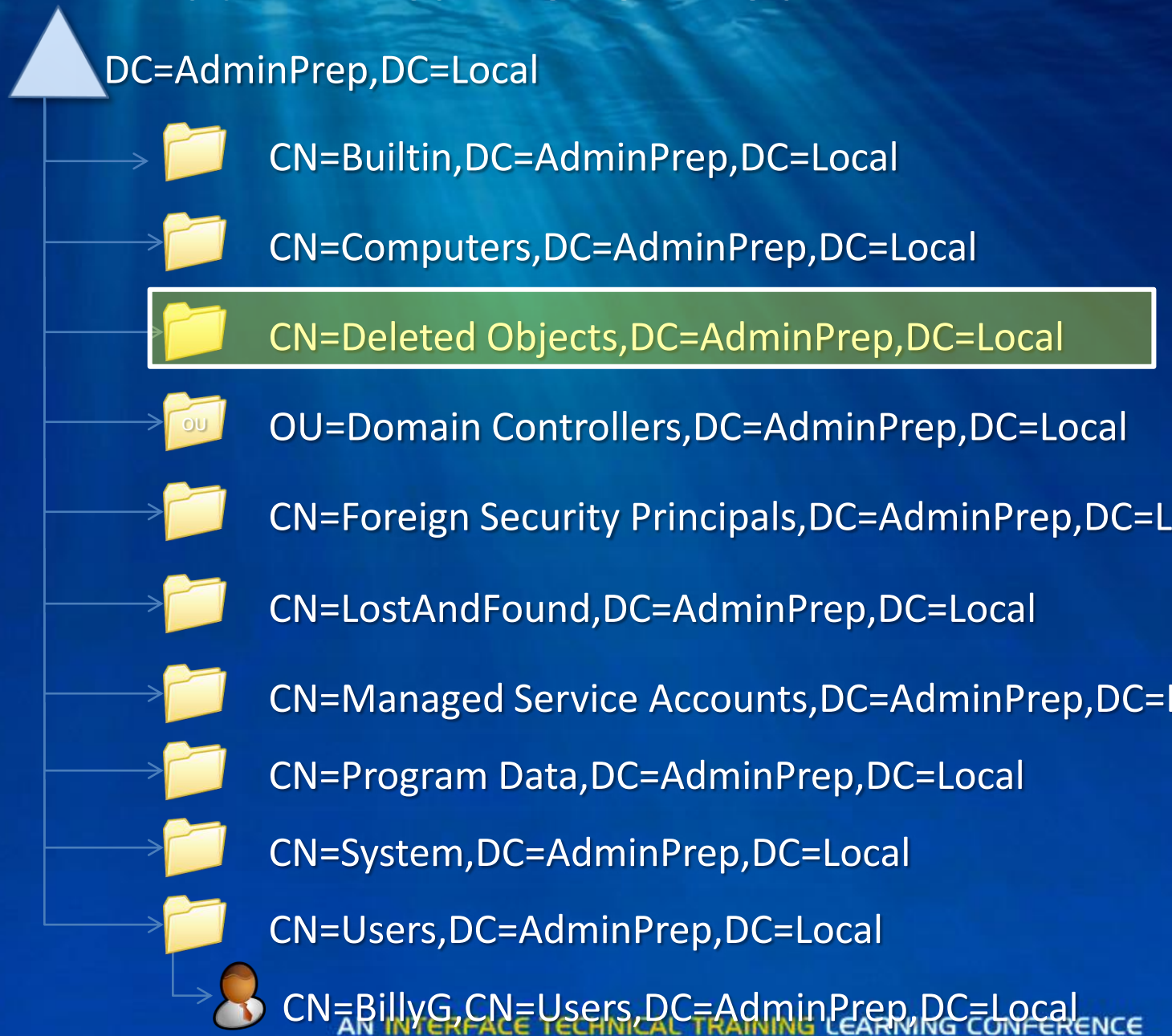
What ADUC Shows You



What ADUC with Advanced Features Shows You



What LDP “can” Show You



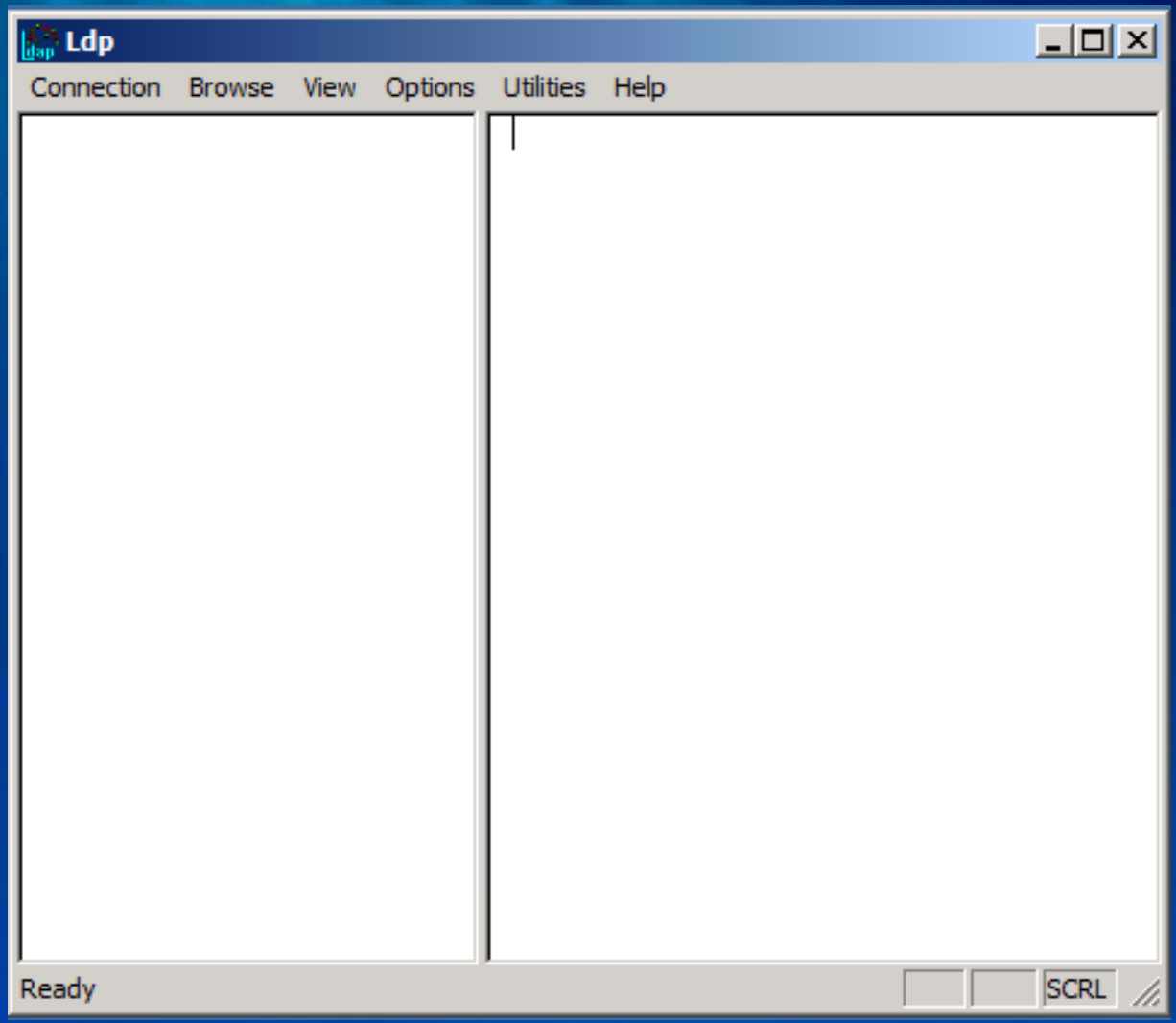
When We Delete Objects, AD…

- Sets the attribute called isDeleted to True
- Removes attributes (as directed by the schema and yes, that could be changed); keeps objectClass, objectGUID, objectSID, sAMAccountName (and others) -- but almost everything else (names, attribs) is gone
- Changes distinguished name (DN) from something like cn=BillyG,cn=users,dc=AdminPrep,DC=Local to a longer "mangled" name containing the objectGUID (example coming)
- Moves AD object in a container called "Deleted Objects"
- Calls the object a "tombstone”

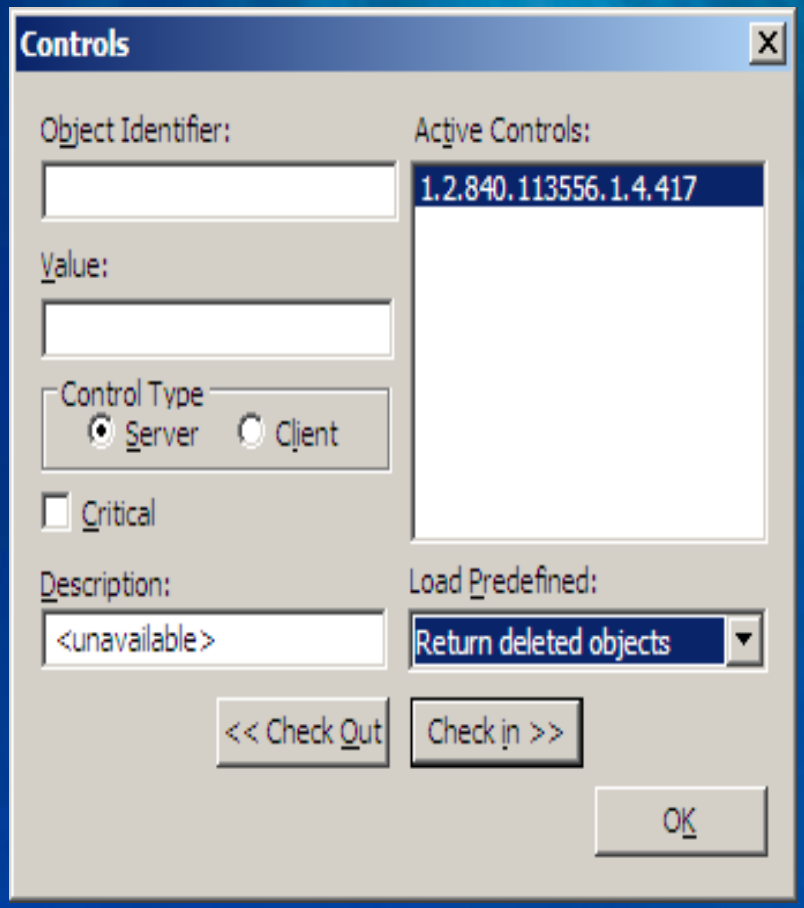
Seeing Your AD's Deleted Objects

- Several tools:
  - ldp.exe (which is in Support Tools for 2003 R2 and earlier, and in-the-box for Server 2008 and 2008 R2)
  - AD PowerShell cmdlets
  - Sysinternals' adrestore.exe

Using LDP


Ldp
Connection
Browse
View
Options
Utilities
Help
Ready
SCRL

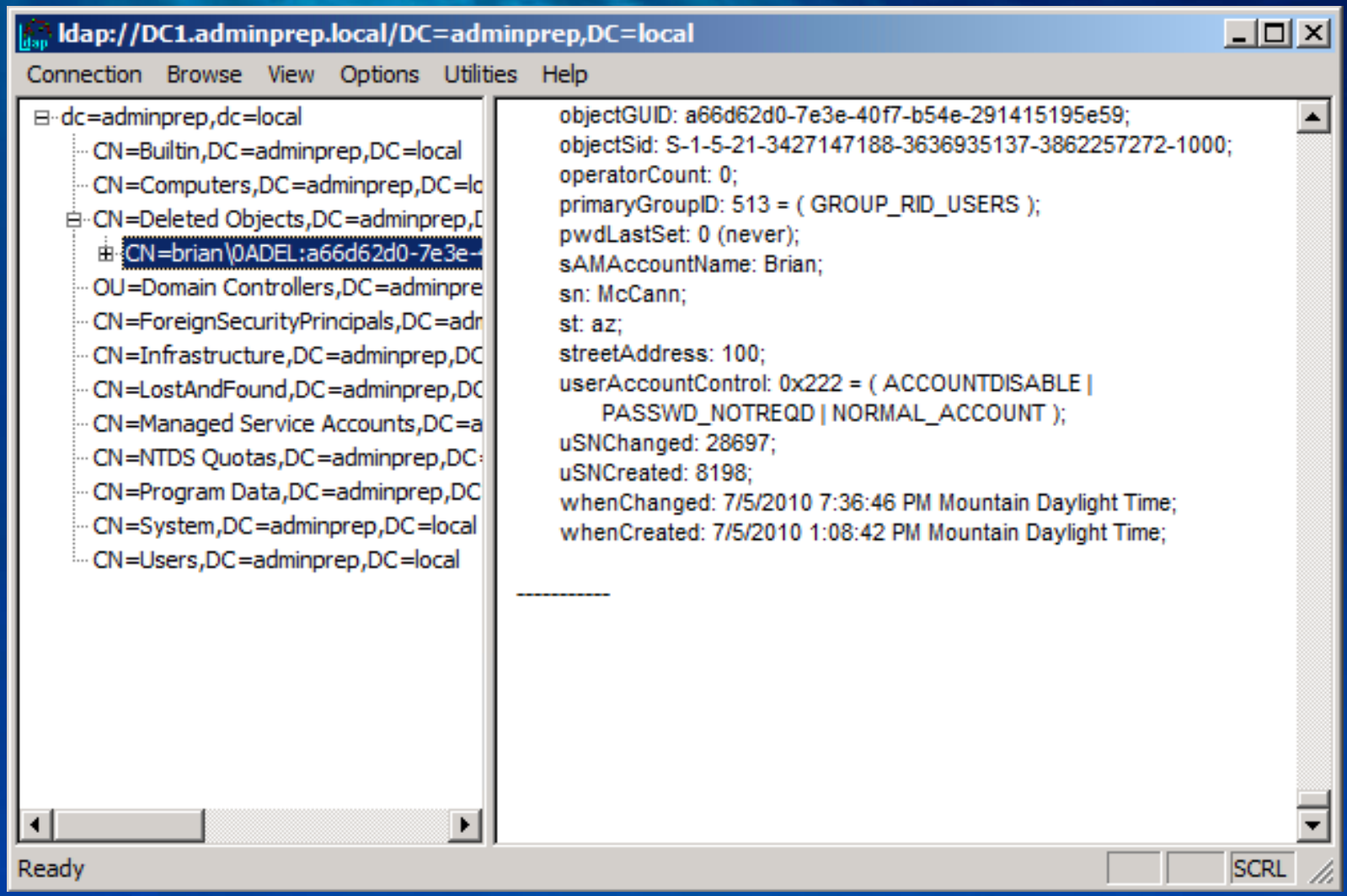
Removing the Veil



By default LDP hides deleted objects. You can enable it by clicking the drop-down labeled "Load Predefined" and choose "Return deleted objects," as you see in the lower right-hand part of the dialog at left. Then click "OK" to return to LDP.

Just be sure that the "Active Controls" field contains 1.2.840.113556.1.4.417.

Deletion, Up Close



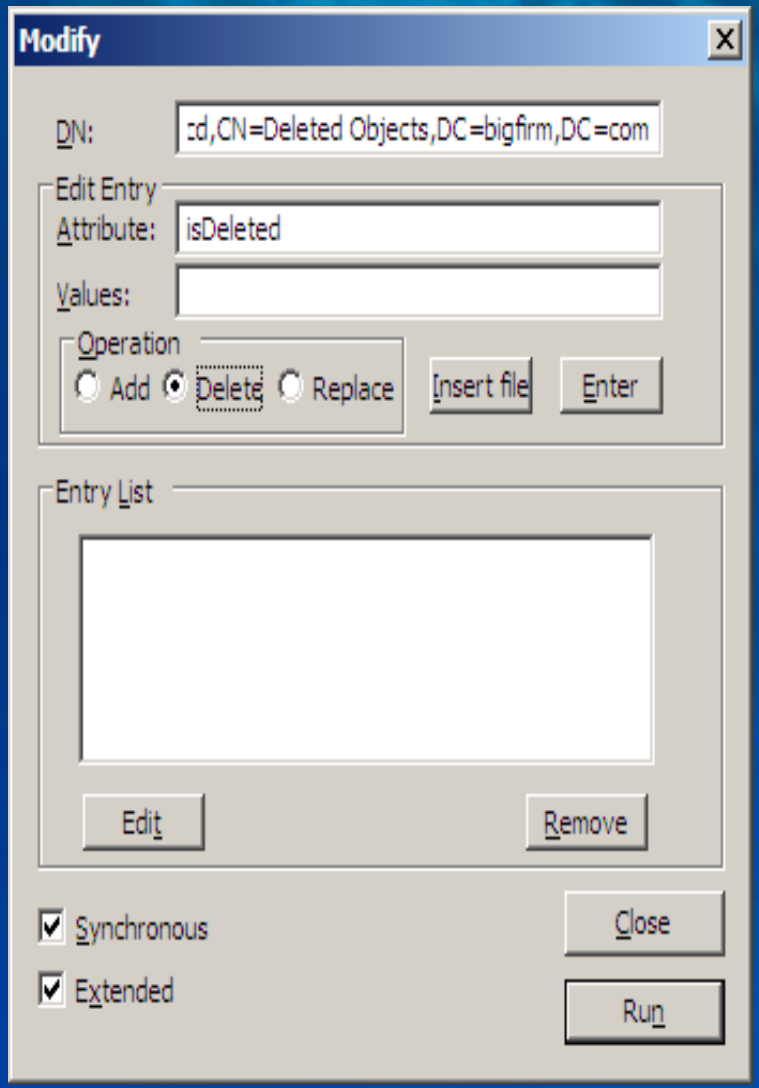
Now, suppose someone wants to delete BillyG…

Let's say that BillyG has an objectGUID value of 6e2971d91 (and yes, that GUID is way too small, but it's just an example)

After deletion…

- DC=AdminPrep,DC=Local
  - CN=Builtin,DC=AdminPrep,DC=Local
  - CN=Computers,DC=AdminPrep,DC=Local
  - CN=Deleted Objects,DC=AdminPrep,DC=Local
    - CN=BillyG\0ADEL:6e2971d91,CN=Deleted Objects,DC=AdminPrep,DC=Local
  - OU=Domain Controllers,DC=AdminPrep,DC=Local
  - CN=Foreign Security Principals,DC=AdminPrep,DC=Local
  - CN=LostAndFound,DC=AdminPrep,DC=Local
  - CN=Managed Service Accounts,DC=AdminPrep,DC=Local
  - CN=Program Data,DC=AdminPrep,DC=Local
  - CN=System,DC=AdminPrep,DC=Local
  - CN=Users,DC=AdminPrep,DC=Local

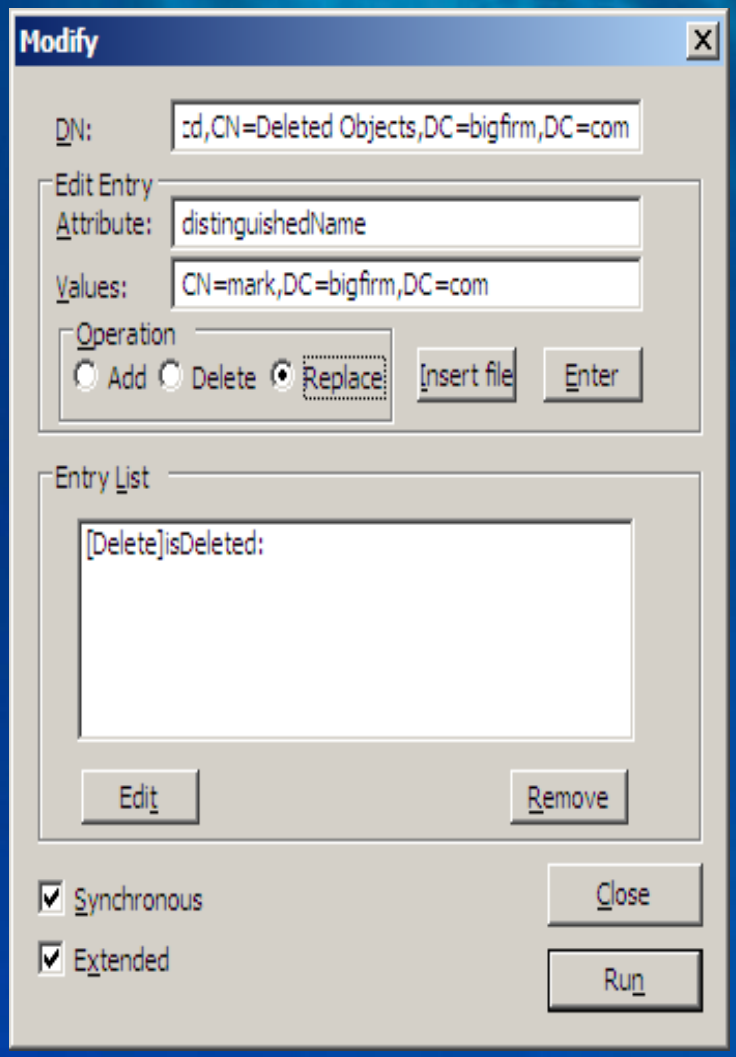
Reanimating with LDP (1)



In the Modify dialog box, create the "delete isDeleted" command by

- type "isDeleted" in the "Attribute: field inside the "Edit Entry" group
- Click the "Delete" radio button in the "Operation" group
- Click Enter to queue it
- Check the "Extended" check box so that LDP knows to use the "let me see deleted stuff" control

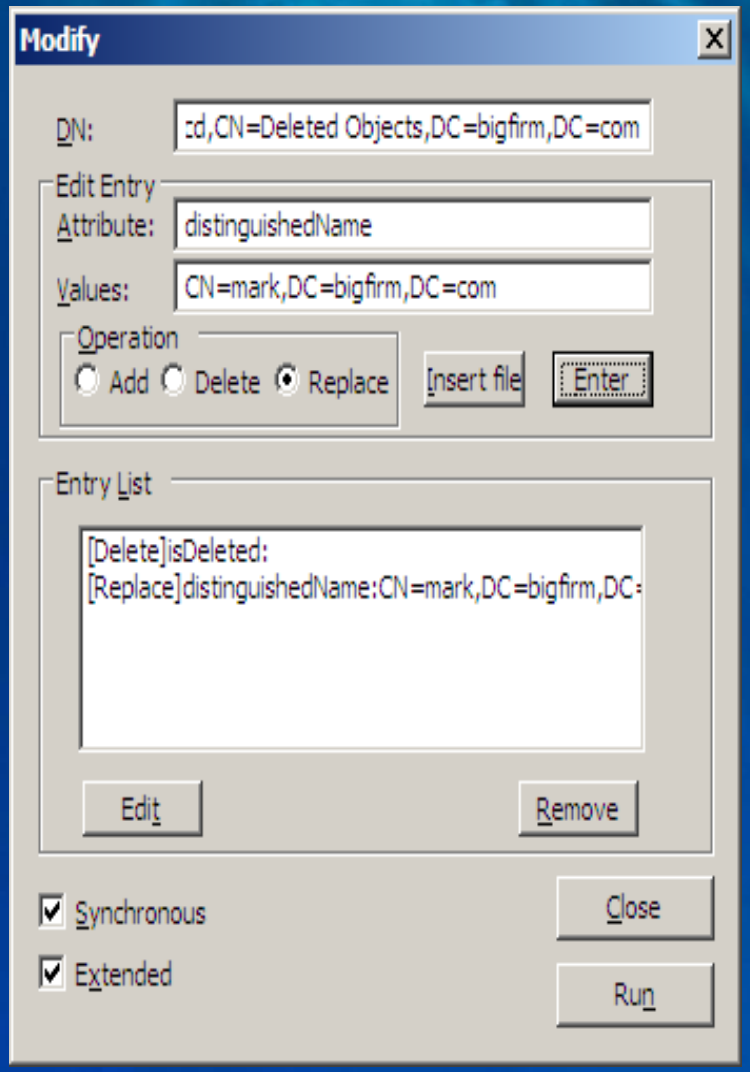
Reanimating with LDP (2)



Now, the first command's in the queue; time for the second.

- In "Edit Entry," change "Attribute:" to "distinguishedName“
- Enter a new DN in "values:"
- In "Operation," click "Replace" as we're not wiping out the DN, we're replacing it
- Then click Enter to get it queued in the "Entry List" field

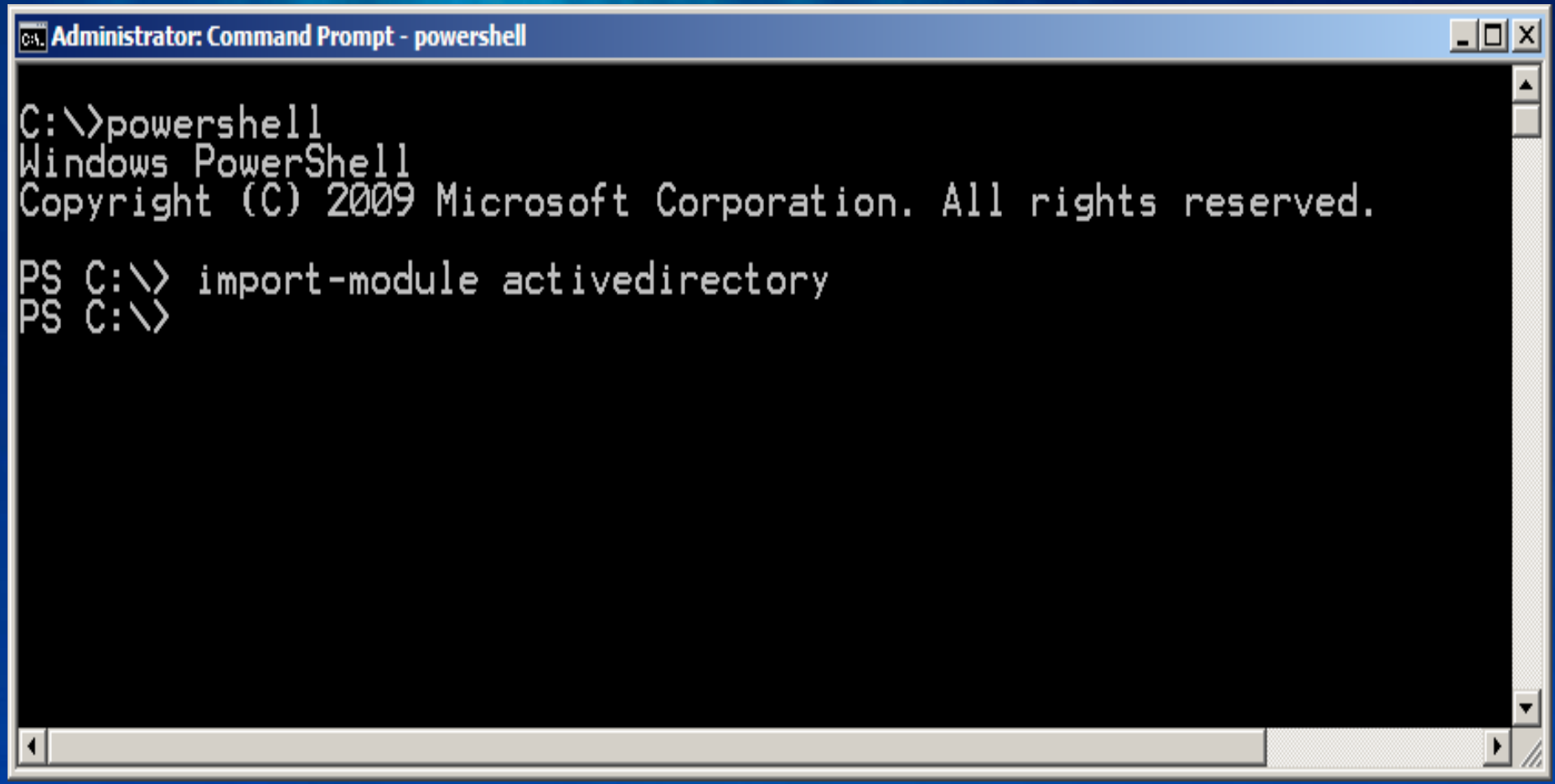
Reanimating with LDP (3)



With both commands queued in "Entry List," double-check that you remembered to check "Extended" and then click Run…

… and your account's returned! (but disabled)

Let's see this in PowerShell



Seeing Deleted Objects in PowerShell

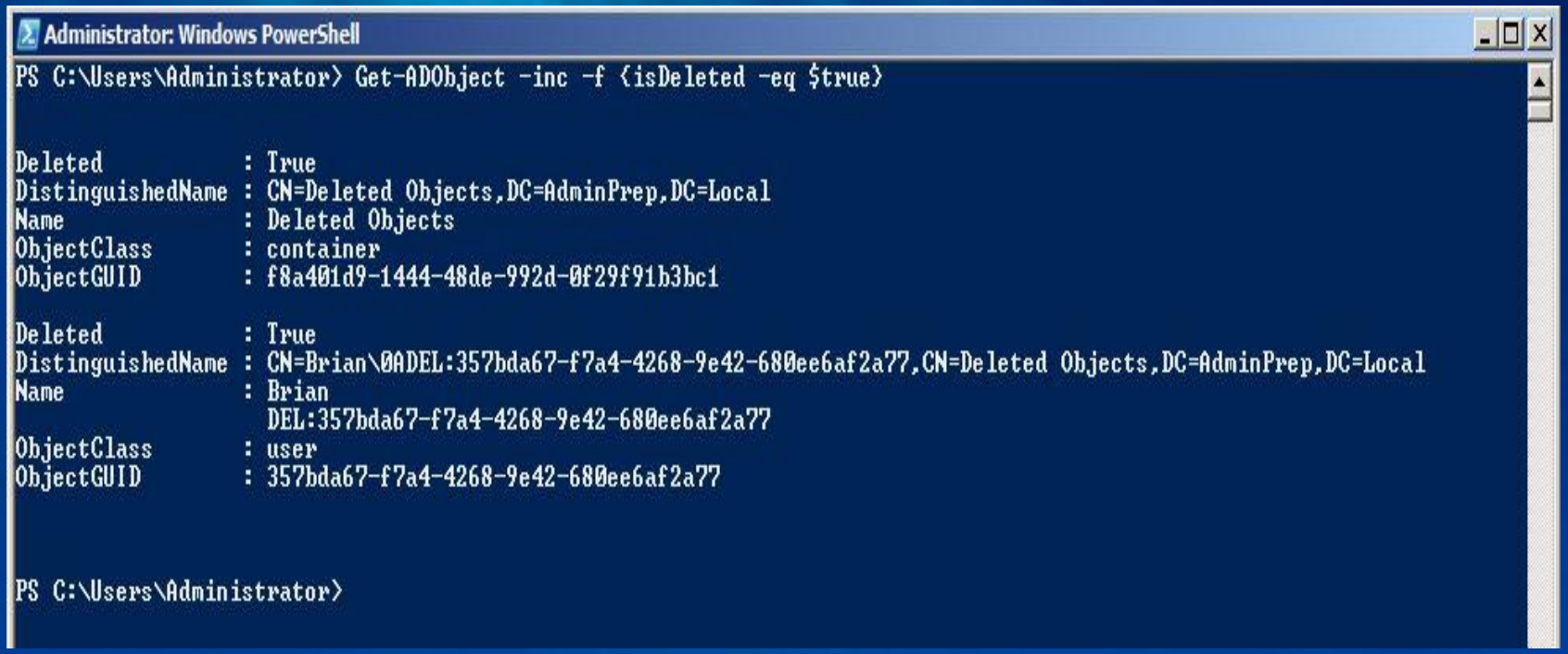
- The basic PowerShell command to see deleted stuff looks like
  - get-adobject –filter * -includedeletedobjects
- And you can shorten it to
  - get-adobject –f * -inc
- But that will show you every item in the whole AD, deleted or not; this
- shows just the deleted stuff:
  - get-adobject -inc -filter {isDeleted –eq $true}

(You probably would not want to see all of the deleted objects in a real domain)

Seeing Deleted Objects in PowerShell

- Another way to see just the deletes:
  - get-adobject -inc –f * -searchbase "cn=Deleted Objects, dc=AdminPrep,DC=Local"
- Or use just the –filter command and match the samaccountname (which is, recall, one of the few things not wiped out by the deletion):
  - get-adobject –f {samaccountname –eq "BillyG"} –inc
- Yet another:
  - get-adobject -inc -f {name -like "*DEL:*"}

get-adobject Example



And Once Tombstoned…

- AD doesn't physically delete the tombstone immediately; in fact, BillyG's tombstone stays around for six months
- That's because AD can't safely delete BillyG's record until every DC knows that BillyG's gone – that is, until every DC contains a tombstone for BillyG
- Reason: once DC1 gets a tombstone for BillyG, it knows that BillyG is no longer around, and blocks various conditions which might cause BillyG to re-appear because DC6 (which *doesn't* know that BillyG's gone) tries to send out BillyG-relevant updates to DC1

Eventually, AD Deletes Tombstones

- In the perfect world, AD would physically delete BillyG's tombstone as soon as every DC knows that every other DC has a BillyG tombstone
- But in a practical sense, that's not easy to do, as not every DC is running and connected to other DCs at every moment
- So Microsoft's compromise was to cause AD to delete a tombstone after it has existed for some fixed period of time
- That was 60 days on 2000 and 2003 RTM-based ADs, 180 days thereafter

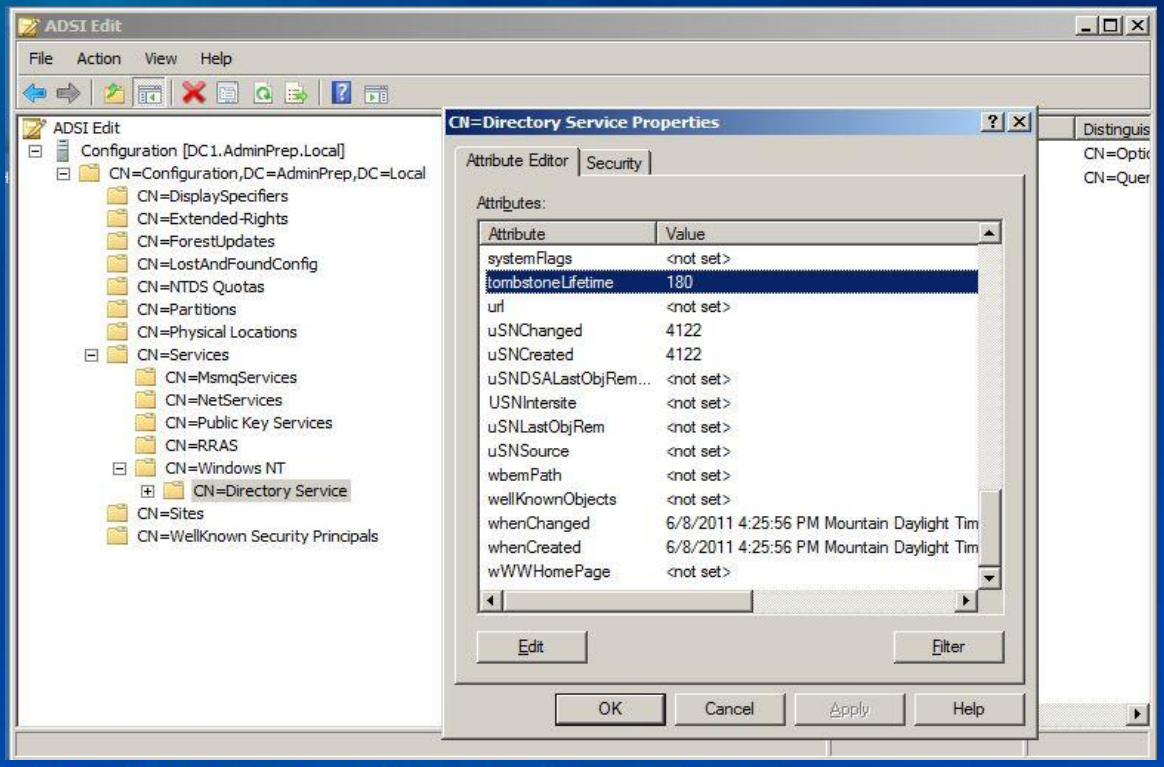
Seeing Your Tombstone Period

- From a PowerShell prompt, type

(get-adobject "cn=Directory Service,cn=Windows NT,cn=Services,cn=Configuration,dc=AdminPrep,DC=Local" -properties "tombstonelifetime").tombstonelifetime

- Value returned is (surprisingly) in days

- ADSIEDIT →



The Final Delete: Garbage Collection

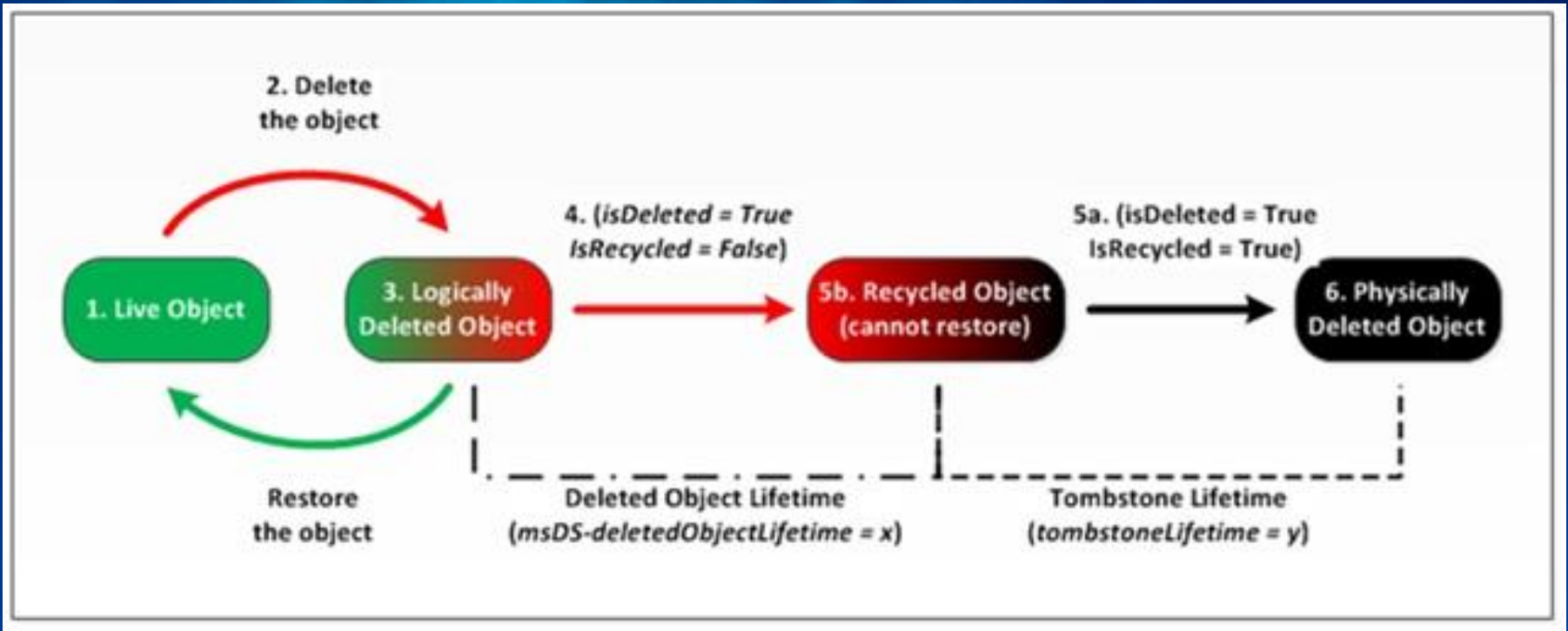
- Once a given DC notices that its local copy of the AD database contains one or more tombstones that are expired, then it's safe to physically delete them
- AD checks for and deletes expired tombstones twice a day during its "garbage collection" period
- So be careful when you reboot your DCs, as you don't want them doing garbage collection first thing in the morning while everyone's trying to log on!

AD Recycle Bin Requirements and Setup

How R2's AD Recycle Bin Works

- First, enable the ADRB feature
- Then, delete an AD object and it enters the "deleted state"
- You now have 180 days (by default) to un-delete it, much as we did with reanimation
- Then it enters "recycled state," which is much like the old tombstone phase, but that cannot be brought back to life, even with reanimation, and it's 180 days by default
- After that, it's scavenged and actually wiped from the AD database during garbage collection
- You can change either of the "180 day" periods

Visual ADRB


2. Delete
the object
4. (isDeleted = True
IsRecycled = False)
5a. (isDeleted = True
IsRecycled = True)
1. Live Object
3. Logically
Deleted Object
5b. Recycled Object
(cannot restore)
6. Physically
Deleted Object
Restore
the object
Deleted Object Lifetime
(msDS-deletedObjectLifetime = x)
Tombstone Lifetime
(tombstoneLifetime = y)

AD Recycle Bin Requirements

- 2008 R2 Forest Functional Level (not just DFL)
- 2008 FFL's not enough, though -- you've got to enable the feature, and once you do, **you can only undelete things deleted after you've enabled the feature**
- PowerShell is the only way to turn this feature on!!! Too bad it is ugly ☹

Enabling AD Recycle Bin

The command looks like this

Enable-ADOptionalFeature -Identity "CN=Recycle Bin Feature,CN=Optional Features,CN=Directory Service, CN=Windows NT,CN=Services, CN=Configuration, DC=AdminPrep,DC=Local" -Scope ForestorConfigurationSet -Target "AdminPrep.local"

Recycling AD Objects with PoSH

- The new PowerShell cmdlet for this is "restore-adobject"
- If you know the object's current distinguished name or its objectGUID, you can just plug that right in, as in:
- restore-adobject dbc3a389-2ce8-4ae7-a377-fde26203efcb, or
- restore-adobject "CN=BillyG\0ADEL:9b16ae67-6a84-4687-ba6c-eddeb69e9dcd,CN=Deleted Objects,DC=AdminPrep,DC=Local"
- Wait, don't run away, there's a better way!

Using restore-adobject

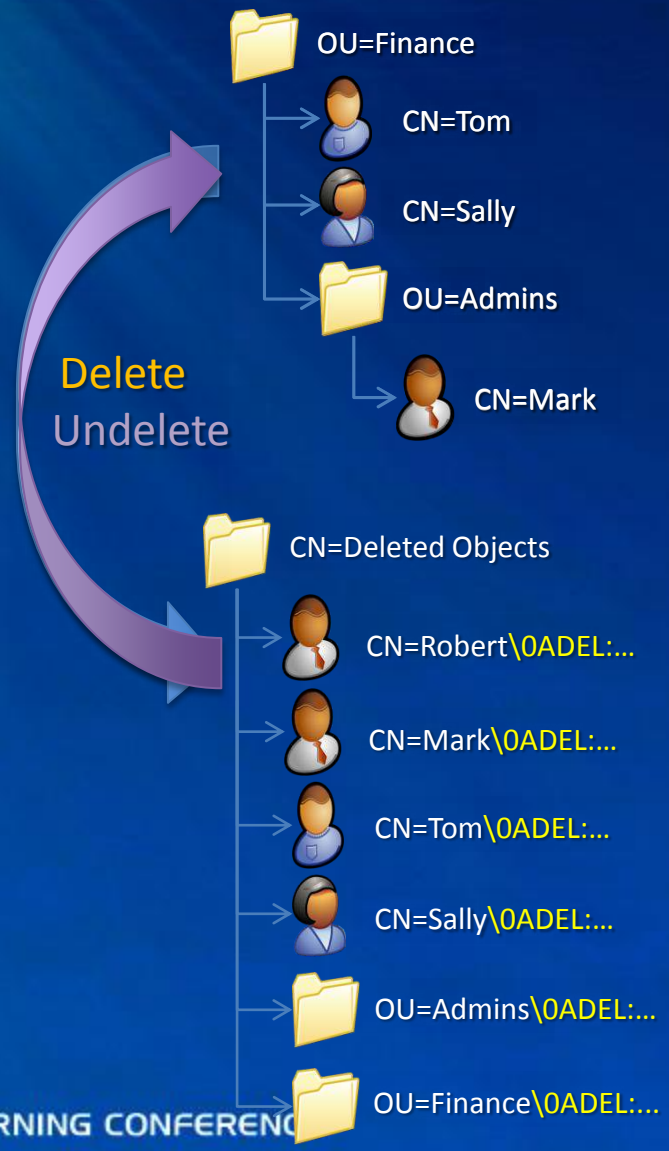
- Best bet is to use the get-adobject command with the –inc option and a filter of some kind, then pipe that into restore-adobject, like
- get-adobject –f {samaccountname –eq "BillyG"} –inc | restore-adobject
- To use wild cards in get-adobject, replace "-eq"with "-like" as in this:
- get-adobject –inc –f {samaccountname –like “bri*"} | restore-adobject

Container "Gotcha"

- Suppose you have deleted an OU inside an OU inside an OU, with a user Jane in it
- You try to undelete Jane, but she lived in an OU that's still deleted… what happens?
- restore-adobject fails
- Workaround: use –newname or –target to give her a place to go

Recovering Multiple Objects

- Deleted Objects container
  - A flat list of all objects in the Deleted state
  - DN is mangled, attributes preserved, lastKnownParent
- Restore objects to live parent
  - Deleted objects must be restored to a live parent
  - Perform restore in top-down order
  - lastKnownParent and lastKnownRDN properties useful in rebuilding hierarchy
  - RDN over 128 chars truncated



Permanent Object Deletion

- Recall that "tombstoned" objects (i.e., those more than 180 days since deletion) cannot be recycled
- This lets us add a new capability – immediate permanent object deletion
- Delete, then delete it again from the Deleted Objects container
- Get-ADObject –f {<*whatever*>} –IncludeDeletedObjects | Remove-ADObject
- Of course, it's not truly irrevocable and permanent; if you have a system state backup, then the original object undelete methods will work fine

Best Practices

- Keep taking real System State backups
- Lowering deleted object lifetime not recommended
- Practice and train
- Turn on DS change auditing for deletions
- Use PowerShell, not LDP.exe

Thank you.

http://blogs.msmvps.com/ad

Backup

Recycle Bin Considerations

- WS08R2 Forest Functional Level → Enable Recycle Bin Feature
- Impact on backup strategy (backup shelf life may change)
  - Backups, IFM Seeds and Packaged Domain Controllers remain valid for the lesser of DeletedObjectLifetime or TombstoneLifetime
- Impact on the database size
  - WS08 R2 DIT size is 10-15% more than WS08 DIT size
  - Subsequent growth depends on size and frequency of object deletions. 15% growth in size of a deleted user observed in the MS production forest.
- No GUI – Management only through PowerShell
- Tombstones can not be auth restored
- Purging deleted objects
  - Delete the object from the Deleted Objects container

```
Get-ADObject -Filter {} -IncludeDeletedObjects | Remove-ADObject
```