Speaker intro

Per Schou Christensen
persch@microsoft.com

- Been at Microsoft Services for ~5 years
- Primary focus is Active Directory
- Working both proactive and reactive (parachuted into forest fires ☺)
- MCM Directory 2008

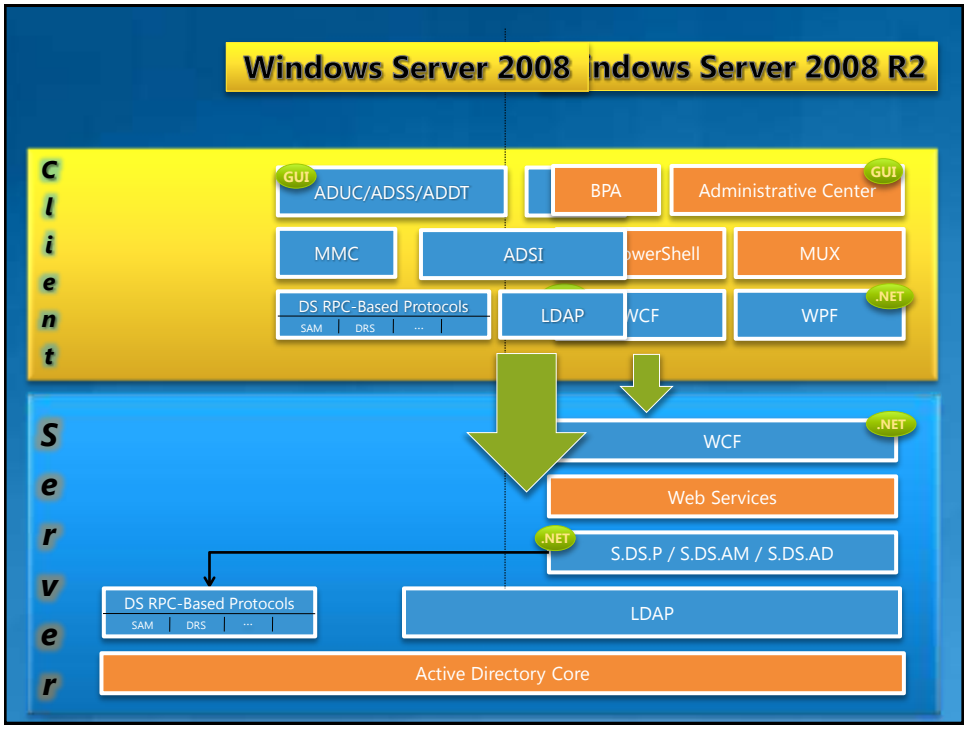
Objectives and takeaways

- **Objectives**
  - *provide an understanding of the core Active Directory feature enhancements/additions*
  - *define requirements*
  - *pique your interest & compel you to dig deeper*
  - *bootstrap that learning curve*
- **Takeaways**
  - *understand the core Active Directory features new to R2*
  - *understand the pain-points they address*
  - *limitations (a necessary evil)*

Core new features/enhancements

We'll start with architecture –

"The big-picture"



On to the features…

Offline Domain Join (ODJ)

- **What does it do?**
  - allows a client to fully achieve a domain-joined state without ever having communicated with a domain controller
- **Takeaways**
  - machines are domain-joined on initial boot without requirement for network connectivity
    - in the absence of cached credentials, *some* connectivity would be a bonus ☺
  - reduces reboots / time needed to deploy OS images
- **Requirements**
  - NO forest or domain functional level requirement
  - NO Windows 7 DCs required
  - joining machine must, however, be Windows 7 client or Windows Server 2008 R2 member

ODJ – try it yourself

1. **Get** a **new** **Windows 7** client or **Windows Server 2008 R2** machine
2. Gracefully **shut down** the **new** machine
3. Gain **writeable access** to the **new** machine's **physical** or **virtual disk**
4. On a **second** **domain-joined machine** & using **domain-join-capable credentials**, run **–**

```
djoin /provision /domain <target domain>
/machine <new machine name> /savefile <filename>

djoin /requestODJ /loadfile <filename>
/windowspath <path to new machine's %windir%>
```

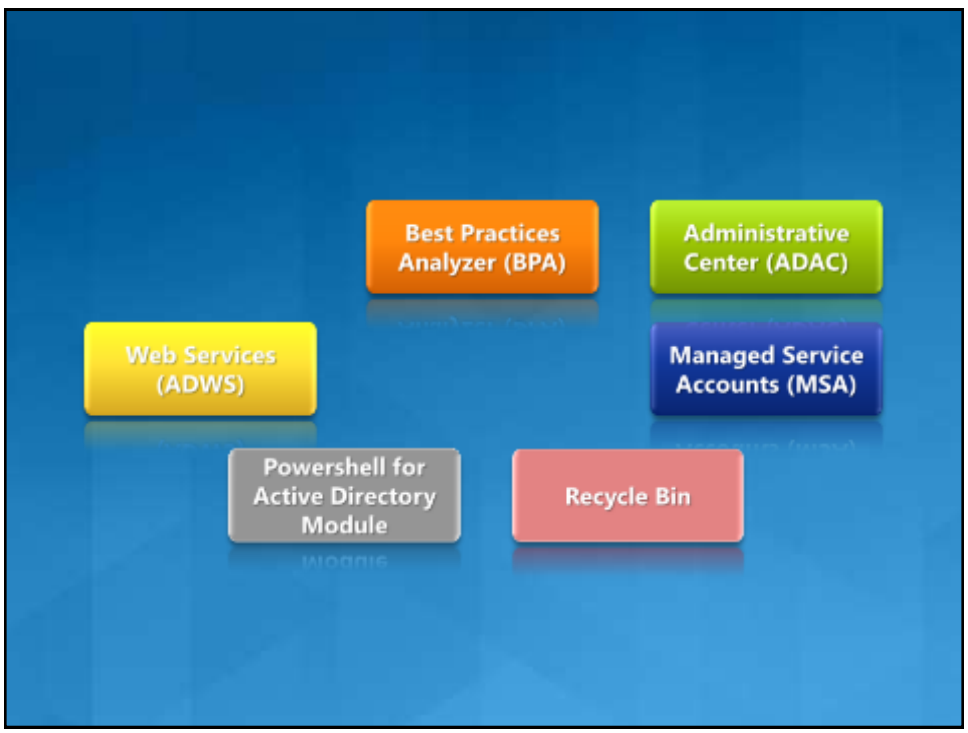
5. **Reboot** **new** machine – it's now in a **fully domain-joined** state

ODJ – how's it done?

- **DJOIN.exe** captures the secrets generated during domain-join (typically exchanged over-the-wire) and stuffs them in a "***blob***"
  - the ***"blob"*** contains
    - the joining machine's
      - *name , password*
    - the target domain's
      - *name, GUID, SID*
    - the target forest's
      - *name*
    - the helper DC's
      - *name, address, attributes, DC's site*

ODJ – specifics on the *"blob"*

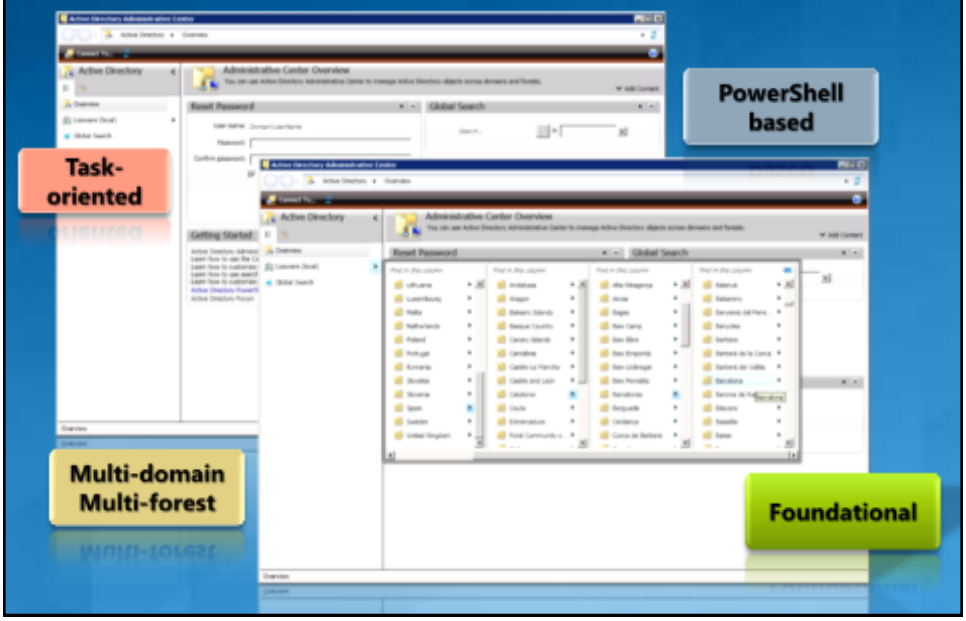
- ONLY one ***"blob"*** per joined-machine
  - CANNOT be re-used
- ***"Blobs"*** are NOT encrypted *(base64 encoded)*
  - so treat them as securely as you would a plain-text password
- No lifetime associated with the ***"blob"***



Administrative Center (ADAC)

- **What is it?**
  - a new domain-administration and navigation tool designed for multi-domain, multi-forest Active Directory environments
- **What is it not?**
  - it is NOT a replacement for the existing tools
- **Takeaways**
  - built on PowerShell and the Active Directory cmdlets
    - literally → ADAC generates & executes the required syntax behind-the-scenes
    - better consistency between command-line and GUI
  - built on the new MUX platform *(Management User Experience)*
    - the foundation for future graphical management tools
  - comfortably supports larger datasets
    - loads directory-data asynchronously *(unlike ADUC / no waiting)*

Administrative Center (ADAC)



ADAC – cool things you can do

- **Manage multiple navigation-nodes**
  - navigation-nodes source from multiple domains or forests
  - seamlessly transition from one node to another
    - multiple nodes able to use a single set of credentials
  - search across multiple nodes
- **Convert UI queries to LDAP filters**
- **Inline & on-the-fly filtering while navigating**
  - NB: filtered client-side
- **Saved views**
  - customized views (column layout, etc.) maintained for each navigation node

ADAC – current limitations

- **PowerShell syntax not exposed**
- **Domain administration only**
  - no topology management
- **No drag & drop**
- **No inline-rename**

demo

Active Directory Administrative Center (ADAC)

Best Practices Analyzer (BPA)

Web Services (ADWS)

Managed Service Accounts (MSA)

Powershell for Active Directory Module

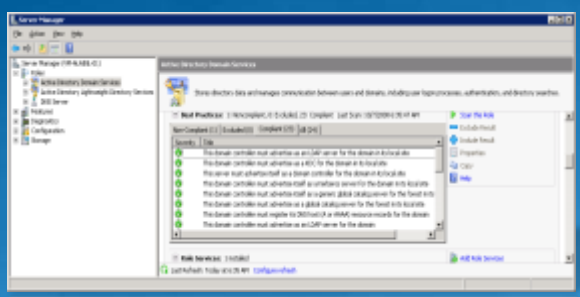
Recycle Bin

Best Practice Analyzer (BPA)

- **What does it do?**
  - analyzes configuration and identifies items that don't conform to established best-practices
  - provides directly-actionable guidance *(ONLY)*
    - does not take action/modify configuration itself
- **Takeaways**
  - scans initiated through *Server Manager* or *PowerShell* cmdets
    - scans initiated remotely using both *Server Manager* and/or PS-remoting
  - scans are user-initiated *(can be scheduled though)*
  - NOT a replacement for monitoring solutions
  - quarterly BPA-scenario updates released post-R2 RTM
    - shipped via Windows update
  - ANYONE can provide BPA-scenario feedback and/or additions at
    - http://connect.microsoft.com/ADBPA
    - additional scenarios MUST be actionable to qualify though

BPA – initiating a scan

- **…from *Server Manager***



- **…from *PowerShell***

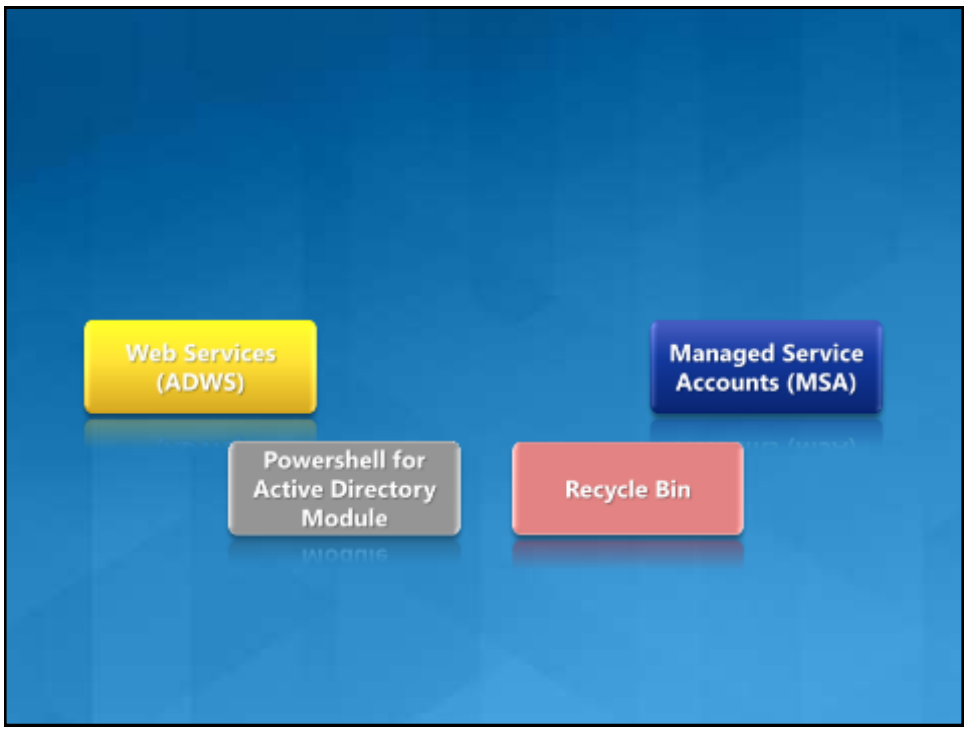
```
Import-Module BestPractices
Invoke-BPAmodel Microsoft\Windows\DirectoryServices
Get-BPAresult Microsoft\Windows\DirectoryServices
```

BPA – what do we look at?

- **DNS**
  - registration & discovery of A/AAAA records
- **Disaster Recovery**
  - multiple DCs per domain
  - backup lifetime
- **Replication**
  - at least one GC per site
  - KCC enabled
  - Virtual Machine-aware
    - provides link to best practices whitepaper
- **Topology**
  - FSMO-role assignment and availability of role holder
- **Lingering Objects**
  - Strict Replication Consistency
- **Time service**
  - PDC time source
  - Max[POS|NEG]PhaseCorrection limits
    - reduces potential risk of forest-wide time skews/slews

BPA – current limitations

- **Limited to local-scan only**
  - doesn't perceive the directory holistically; BPA sees DCs, not domains
    - some exceptions, e.g. – understands *Site-Link* options, TSL, etc.
- **Not yet user-extensible**
  - can't add scenarios
  - can't change tolerances/thresholds/values
- **Requires Windows Server 2008 R2 DC**
  - cannot run against non R2 DCs
    - ADMG (the OOB release of ADWS for downlevel DCs) is NOT sufficient
- **Scope of what we report on is not configurable**
  - reports on DC's view of whole forest incl. all sites and all domains



PowerShell Module

- **What is it?**
  - the *PowerShell for Active Directory Module* is a comprehensive suite of Active-Directory-specific cmdlets and a PowerShell provider
  - provides administration, configuration and querying capabilities
- **Takeaways**
  - brings the power and flexibility of PowerShell to Active Directory
  - the foundation and future of Active Directory administration
  - the emerging de-facto standard for automation and management
- **Requirements**
  - Windows 7 or Windows Server 2008 R2
  - PowerShell V2.0
  - ADWS (or ADMG) on suitable DC(s)
    - cmdlets/provider don't speak LDAP

PowerShell – getting started

- **PowerShell basics**
  - help is built-in and consistent
  - tab-completion eases discoverability
    - also supports argument tab-completion
  - well-known, legacy commands supported through aliasing
  - includes support for traditional commands/legacy binaries
    - e.g. *dir, cls, cd, md , ping.exe, fsutil.exe*
  - built-on and exposes the .NET framework
- **PowerShell cmdlets**
  - commands formatted as verb-noun pairs
    - e.g. ***get-ADuser***
  - an action (verb) is taken on an object (noun)
  - easily composed (pipelined) to solve complex end-to-end management/automation problems

PowerShell – getting started

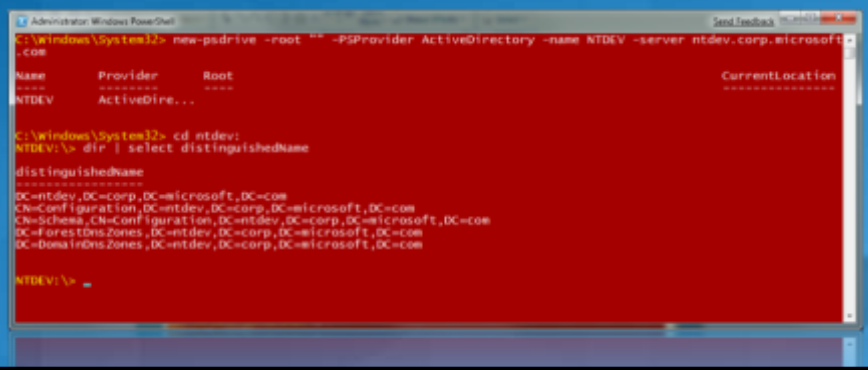
- **Installed with**
  - *Server Manager* / Windows Server 2008 R2's DCpromo
  - the Remote Server Admin Tools for Windows 7 client *(RSAT)*
- **Module extend's PowerShell's native capabilities**

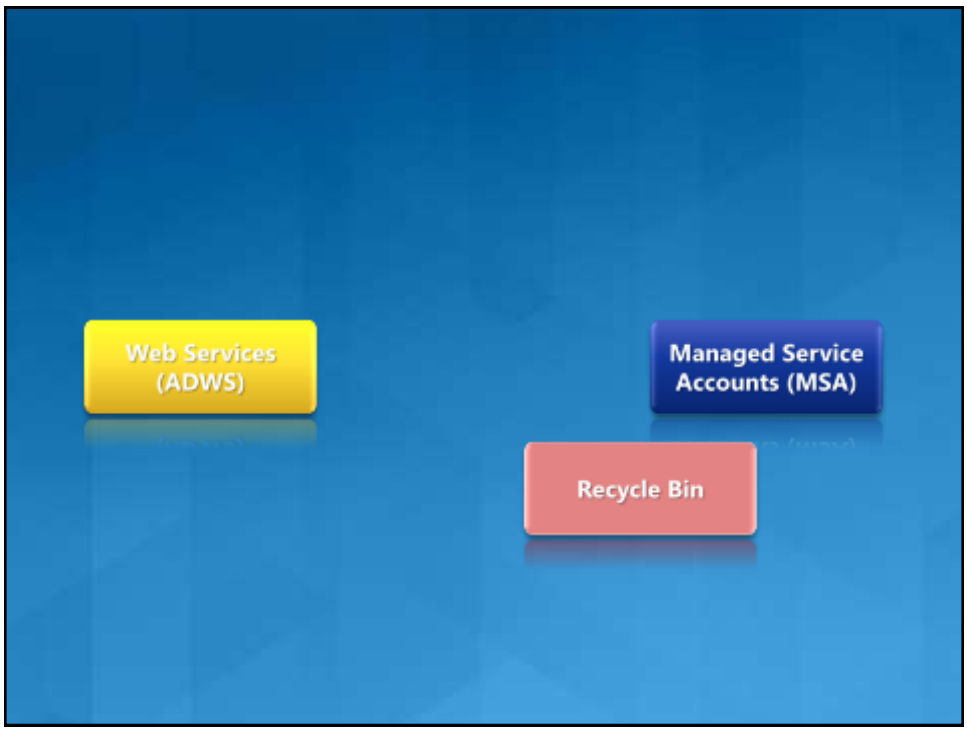
```
PS C:\> import-module ActiveDirectory
PS C:\> Get-Command -module ActiveDirectory
```

- **~90 AD-specific cmdlets for both DS & LDS**
  - entirely consistent syntax and output model
  - passes well-known, property-rich data between cmdlets
  - consistent management of Windows Server roles 'out-of-the-box'

PowerShell Providers

- **What are *PowerShell Providers?***
  - they permit the use of common commands across discrete services that possess compatible notions of hierarchy and data
- **What the heck does that mean?**
  - perform operations in the filesystem, the registry, the certificate store, Active Directory etc. using IDENTICAL syntax by *CD'ing* into them



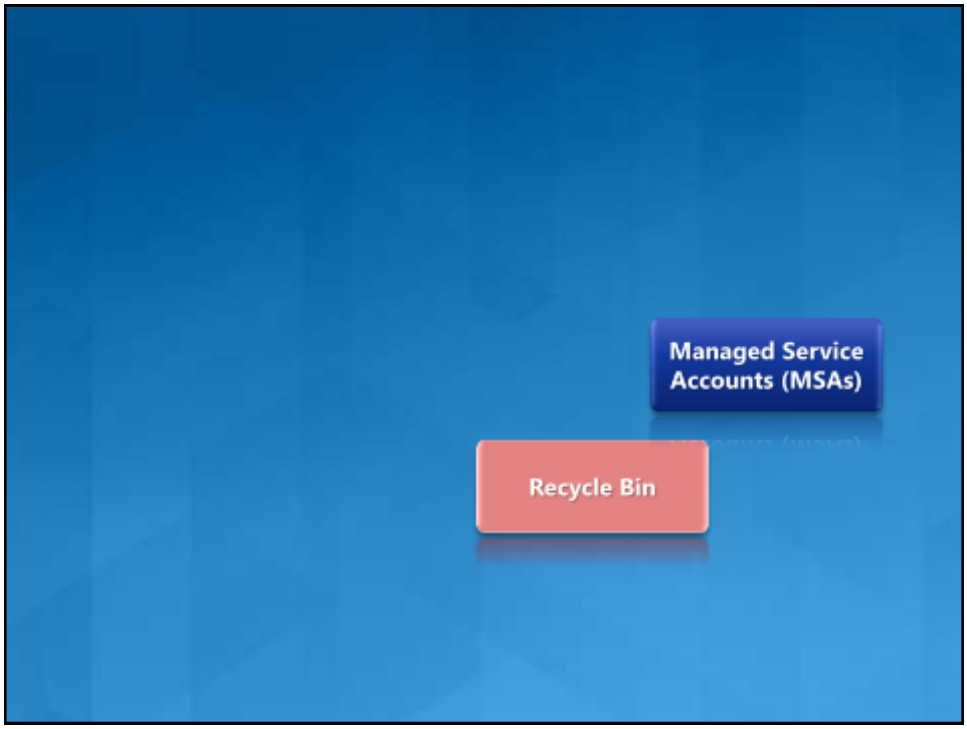


Web Services (ADWS)

- **What is it?**
  - *Web Services* implementation listening on TCP/9389
    - supports traditional *Domain Services* (DS) & *Lightweight Directory Services (LDS)*
  - built on WS* and WCF protocols
    - WS-enum, WS-transfer, IMDA
  - paving the way for a new developer experience
- **Takeaways**
  - supplements LDAP and RPC for remote administration
  - not intended for developer consumption in this release
  - discovery achieved via DC-locator LDAP ping *(scale concerns?)*
  - does NOT require IIS

ADWS – things to know

- **Requirements**
  - Windows Server 2008 R2 Domain Controller or LDS instance
  - Windows Server 2003 & 2008 Domain Controller's supported
    - via OOB release named Active Directory Management Gateway (ADMG)
  - Must run locally on DC or LDS instance
  - Distribute to enough DCs/instances to
    - locally represent every NC managed through the PowerShell for Active Directory Module or ADAC



Managed Service Accounts (MSA)

- **What are they?**
  - a new class of security principal
  - used exclusively by *Services*
  - replacement for the legacy notion of *Service Accounts*
  - provides automatic password management
- **Takeaways**
  - eliminates management burden
  - enhances security
  - strictly one MSA per Service per Server
    - i.e. MSAs CANNOT be shared across multiple machines
  - usable ONLY on Windows 7 or Windows Server 2008 R2

MSA – password details

- **MSA passwords**
  - machine generated
    - using *CryptGenRandom*
  - uses maximum available entropy
    - 240 bytes in length
  - cycled according to NETLOGON *MaximumPasswordAge* policy
  - can be reset by

```
PS C:\> reset-ADServiceAccountPassword <MSA>
PS C:\> nltest /sc_change_pwd:<SAMAcctName>
```

- **MSA passwords are NOT**
  - affected by Domain password policy
  - affected by fine-grain password policies

MSA – how to(s)

1. **To create a *Managed Service Account***

```
PS C:\> New-ADServiceAccount -Name {MSA name} -Path {directory path}
```

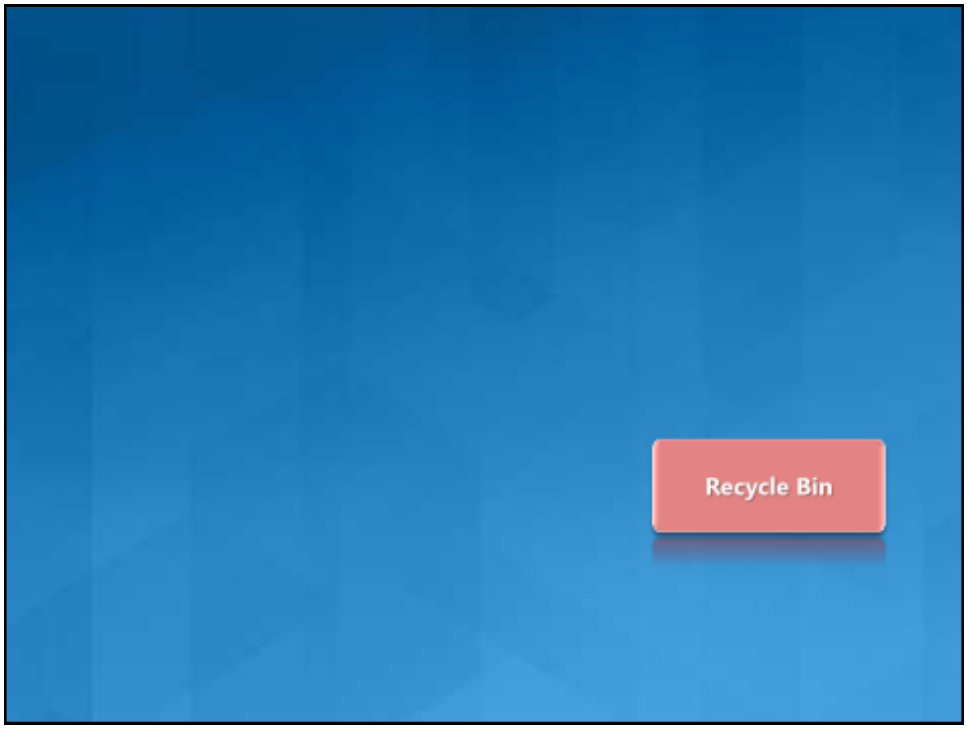
2. **To associate an MSA with a server**

```
Add-ADServiceAccount -Identity {FQDN} -ServiceAccount {MSA}
```

3. **To install the MSA on the local server**

```
Install-ADServiceAccount -Identity {MSA}
```

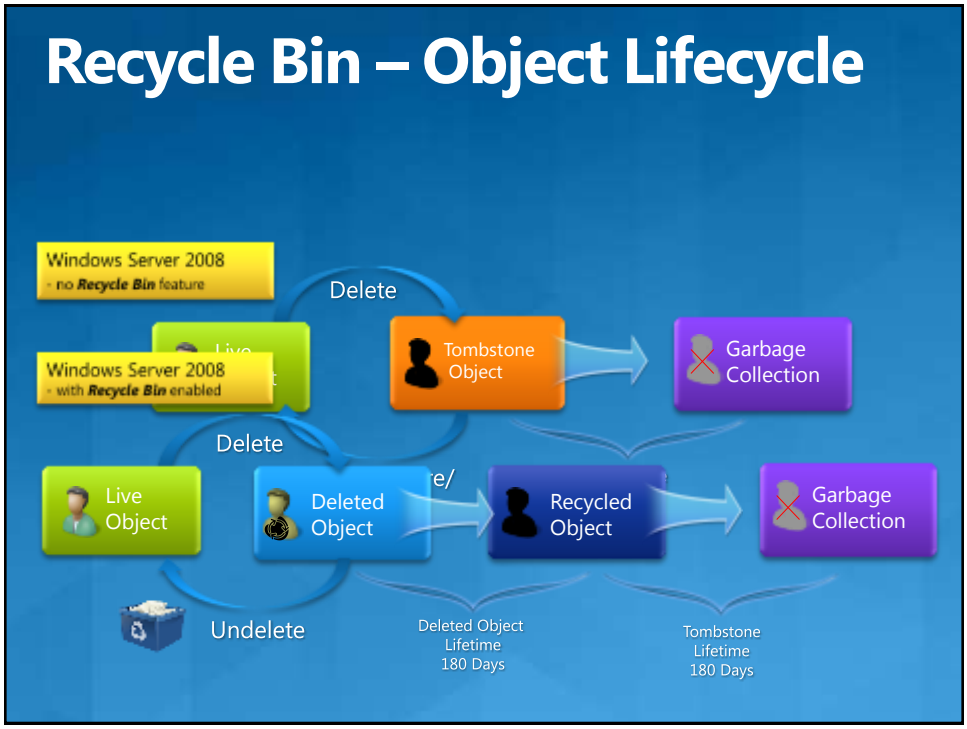
4. **NOTE:**
   - DON'T forget to configure the service to use the MSA



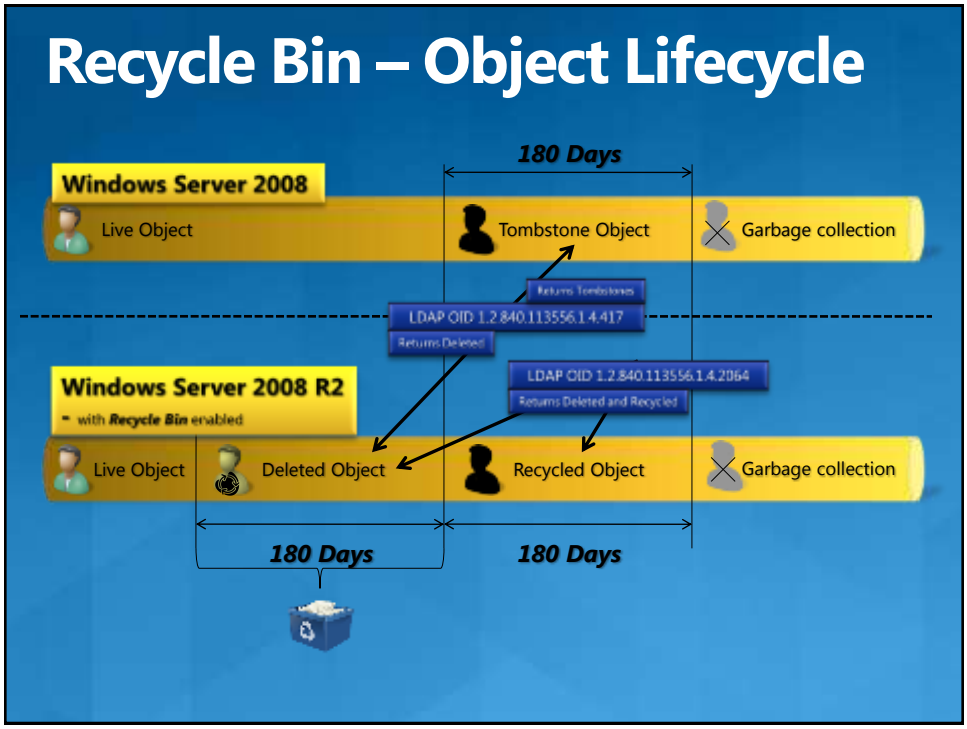
Recycle Bin

- **What is it?**
  - allows recovery of any deleted Active Directory object in its complete & original condition
  - primary enabler – no longer strips linked-attributes at deletion-time
    - adds additional column to link-table (*link_deActiveTime*)
    - traditional attribute value-deletion delayed
- **Takeaways**
  - able to fully recover deleted-objects without authoritative restore
  - tombstones are a thing-of-the-past
- **Requirements**
  - requires forest-functional level 4 *(WIN2008R2)*
    - due to changes to phantom cleanup process
  - enable the feature

Recycle Bin – Object Lifecycle



Recycle Bin – Object Lifecycle



Recycle Bin – things to know

- **Impact on the DIT**
  - the first Windows Server 2008 R2 DC generates churn, why?

```
foreach (object in set(deletedObjects_in_writableNCs)) {
    add(object, "isRecycled", "TRUE");  // Replicated operation
)
```

  - DIT-size increases between 5 & 10% / ongoing increase usage-dependent
- **Feature NOT enabled by functional level alone**
  - our first (and currently only) *optional feature* *(rootDSE mod.)*
  - optional features need to be switched on / bound to schema FSMO

```
Enable-ADOptionalFeature  'Recycle Bin Feature' -Scope ForestOrConfigurationSet
                          -Target {target DC or LDS-instance DN}
```

- **Resulting behavioral changes**
  - once object *isRecycled,* traditional tombstone renanimation blocked
    - NTDSUTIL option will permit legacy tombstone reanimation through auth. restore

Recycle Bin – worth a mention

- **Where's the graphical interface?** ☹
- **Changes in notion of TSL *(Deleted Object Lifetime)***
  - DOL = TSL = 180 days (by default)
  - both can be modified independently *(cn=Directory Services,cn=Windows NT…)*
    - *msDS-deletedObjectLifetime*
    - *tombstoneLifetime*
- **Affects on backup strategy**
  - backups remain valid for the lesser of DOL, TSL
- **Demand-deletion *(double-delete)***
  - delete the object from the Deleted Objects container

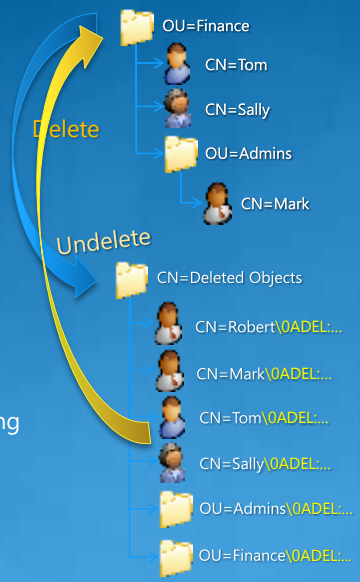
```
Get-ADObject -Filter {<suitable filter>} -IncludeDeletedObjects | Remove-ADObject
```

- **Restore an object**

```
Get-ADObject -Filter {displayName -eq "Mary"} -IncludeDeletedObjects | Restore-ADObject
```

Recycle Bin – recovering objects

- ***Deleted Objects* container**
  - flat list of objects in deleted state
  - RDN mangled *(<RDN>+DEL:+CHAR(0A))*
  - linked/non-linked attributes preserved
    - *lastKnownParent* and *lastKnownRDN* populated
- **Restore objects to live parent**
  - deleted objects MUST be restored to a live parent
    - perform restore top-down
  - *lastKnownXX* properties useful in rebuilding hierarchy



... more details on Recycle Bin *in the appendix*

demo

Recycle Bin

DSRM Password Synchronization

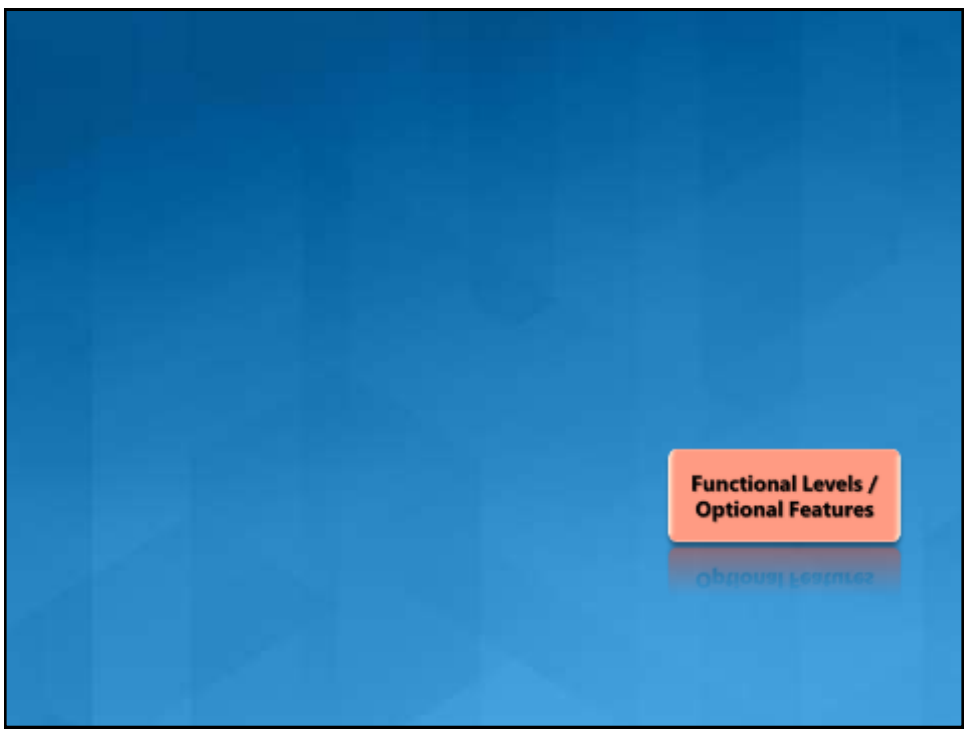
... two more things worth a mention

Functional Levels / Optional Features

DSRM Password Synchronization

- **What is it?**
  - the ability to synchronize a Domain Controller's DSRM password to an account of your choosing in the DC's domain
- **Restrictions**
  - available to Windows Server 2008 (QFE) and R2 only
  - it's a ONE-TIME synchronization
    - copies the password as it is at the time the command is executed
- **How do you do it?**
  - exposed in NTDSUTIL

```
ntdsutil "Set DSRM Password" "Sync from domain account <suitable user> " q q
```



Functional Levels / Optional Features

- **Windows Server 2008 R2 introduces new domain/forest functional level *(4)***
  - functional level increases carry *NO* features with them, except –
    - preventing the introduction of legacy DCs
    - allowing applicable features to be enabled
  - customers more comfortable raising functional level
    - *NO* unforeseen side effects
  - features can be enabled one at a time
    - future features may be capable of being disabled *(none today)*
    - optional features governed by new CAR *(aka: extended right)*
  - functional level can, therefore, be ROLLED BACK
    - assuming no features are enabled that block it
    - *Recycle Bin (once enabled)* CANNOT be disabled
    - no graphical interface to roll back functional level
      - edit *msDS-Behavior-Version* on *domain head* or *Partitions* container

Features & Functional Levels

Minimum requirements

With this minimum requirement…	… you get these features
One or more Windows 7 clients or Windows Server 2008 R2 member servers	**Offline Domain Join** **Managed Service Accounts**
+ *one or more Web Services (ADMG included) instances*	**Active Directory Administrative Center** **PowerShell for Active Directory Module**
+ *one or more Windows Server 2008 R2 Domain Controllers*	**Best Practices Analyzer** **DSRM Password Sync.** - **(QFE also available for Windows Server 2008)**
+ *Windows Server 2008 R2 Domain Functional Level*	**Authentication Mechanism Assurance** **** enhanced MSA-SPN management***
+ *Windows Server 2008 R2 Forest Functional Level*	**Recycle Bin**

Microsoft®

Your potential. Our passion.™

Appendix

Deleting an Object

Object deletion in WS08:	Object deletion in WS08 R2:
isDeleted=TRUE	isDeleted=TRUE; **isRecycled=NULL**
lastKnownParent set	lastKnownParent, **ms-DS-lastKnownRDN** set
Moved to the DeletedObjects container	Moved to the DeletedObjects container
DN is mangled •rDN beyond 128 char would be truncated •Hierarchy is effectively flattened	DN is mangled •rDN beyond 128 char would be truncated •Hierarchy is effectively flattened
All but a few non-linked attributes (e.g. GUID, SID, sidHistory, etc.) are preserved	**All non-linked attributes are preserved**
All linked attributes (e.g. member/memberOf) are stripped away	**All linked attributes are preserved**
Only visible with ShowDeletedObjects LDAP control	Only visible with ShowDeletedObjects LDAP control
Purged after tombstoneLifetime expires	**Purged after deletedObjectLifetime expires if that value is set, else after tombstoneLifetime expires**

Restoring an Object

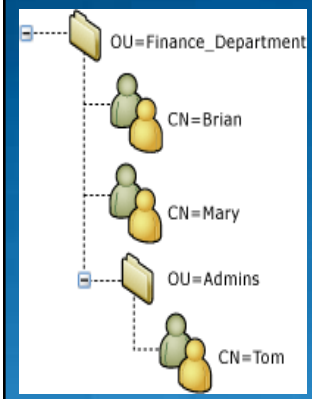
Object restoration in WS08:	Object restoration in WS08 R2:
Delete isDeleted attribute	Delete isDeleted attribute
Change DN based on lastKnownParent and mangled DN	Change DN based on lastKnownParent and **ms-DS-lastKnownRDN**
Only some non-linked attributes (e.g. GUID, SID, sidHistory, etc.) are restored → import old values from snapshots	**All non-linked attributes are restored**
None of the linked attributes (e.g. member/memberOf) are restored → regenerate links using LDIFs from auth restore	**All linked attributes, even cross-domain links, are restored**
Tool: ldp.exe	Tool: ldp.exe, **Active Directory PowerShell**

Turning on Recycle Bin - ADPsh

- #Raise forest functional level
- Set-ADForestMode –Identity contoso.com -ForestMode Windows2008R2Forest

- #Turn on Recycle Bin
- Enable-ADOptionalFeature –Identity 'CN=Recycle Bin Feature,CN=Optional Features,CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration, DC=contoso,DC=com' –Scope Forest –Target 'contoso.com'

Restoring Object(s) - ADPsh



- #Restore a single object
- Get-ADObject -Filter {displayName -eq "Mary"} -IncludeDeletedObjects | Restore-ADObject

- #Restore a tree
- Get-ADObject -ldapFilter:"(msDS-LastKnownRDN=Finance_Department)" –IncludeDeletedObjects | Restore-ADObject

- Get-ADObject -SearchBase "CN=Deleted Objects,DC=contoso,DC=com" -Filter {lastKnownParent -eq "OU=Finance_Department,DC=contoso,DC=com"} -IncludeDeletedObjects | Restore-ADObject

- Get-ADObject -SearchBase "CN=Deleted Objects,DC=contoso,DC=com" -Filter {lastKnownParent -eq "OU=Admins,OU=Finance_Department,DC=contoso,DC=com"} -IncludeDeletedObjects | Restore-ADObject

Restore AD Subtree script

- Sample demo script is published at Technet
- http://technet.microsoft.com/en-us/library/dd379504(WS.10).aspx

How an Object is Deleted

- In WS08:
  - Object is first locally deleted
    - Non-linked attributes that are not preserved are cleared (in the data table)
    - All linked attributes are removed (from the link table)
  - Object deletion is replicated
    - isDeleted=TRUE is the replicated event
    - Same operations as above are repeated on each notified DC
    - Cross-domain DCs are notified as follows:
      - Infrastructure Master checks Global Catalog server for referenced objects with mangled DN (indicating object deletion)
      - If so, creates an InfrastructureUpdateObject to trigger deletions of object on various DCs

How an Object is Deleted

- In WS08 R2:
  - Object is first locally deleted
    - All non-linked attributes are preserved (in the data table)
    - All linked attributes (in the link table) are marked as *deactivated*
  - Object deletion is replicated
    - isDeleted=TRUE is the replicated event
    - Same operations as above are repeated on each notified DC
    - Cross-domain DCs are notified as follows:
      - *Every DC* checks Global Catalog server for referenced objects with mangled DN (indicating object deletion)
      - If so, deletes object locally

Deleted Object Lifetime

- **DeletedObjectLifetime**
  - **Is the period during which a deleted object can be restored, fully without loss of attributes**
  - Not set by default
    - If DOL = null, a deleted object stays deleted for tombstoneLifetime (fallback)
- Recycled Object Lifetime
  - tombstoneLifetime is the actual attribute (default = 180 days)
  - User should not worry about this as it is simply an artifact of replication
  - Too short → lingering objects; too long → database bloat

Setting object lifetimes - ADPsh

- #Change deletedObjectLifetime
- Set-ADObject -Identity "CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=mydomain,DC=com" –Partition "CN=Configuration,DC=mydomain,DC=com" –Replace:@{"msDS-DeletedObjectLifetime" = 60}

- #Change tombstoneLifetime
- Set-ADObject -Identity "CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=mydomain,DC=com" –Partition "CN=Configuration,DC=mydomain,DC=com" –Replace:@{"tombstoneLifetime" = 365}

Authoritative Restore Notes

- **Backup shelf life**
  - To avoid clashing of an undeletion and recycling
  - Applies to install-from-media as well
- Recycling wins over deletion/undeletion
  - Even auth restore cannot restore recycled objects
- Use new option in NTDSUtil to restore a tree

Authentication Mechanism Assurance (AMA)

- **What is it?**
  - allows Administrators to map certificate issuance policy to a security-group
    - if specified OID is present during cert-driven/SmartCard authentication, the associated SID is added to the token
  - permits applications to control access to resources according to authentication type/strength
  - control access to resources based on claims
    - use of smartcard for logon
    - specific OID present in cert.
- **Takeaways**
  - built on information obtained during cert.-based authentication
  - additional credential attributes added to Kerberos tickets and consumed by claims aware applications as authorization data

AMA – things to know

- **Requires Windows Server 2008 R2 domain functional level**
- **Requires Kerberos** *(NTLM not supported)*
- **Kerberos passes OIDs to SAM → SAM determines mapping between OIDs and security-groups (1:1 OID to group mapping)**
  - group to which the issuance policy is mapped MUST –
    - contain NO members
      - membership additions attempted after-the-fact are blocked
    - be a *Universal Security*-group *(not a Global/Domain-Local Distribution-group)*
  - if requirements are met → group-SID injected into the PAC
- **Tools to configure AMA comprises LDP, etc. or two scripts –**
  - `set-IssuancePolicyToGroupLink.ps1`
  - `get-IssuancePolicy.ps1`
- **Scripts are NOT included with Windows Server**
  - step-by-step guide and scripts can be downloaded from – http://technet.microsoft.com/en-us/library/dd378897(WS.10).aspx