Services Day - 07 November 2017

Deep Dive into Azure AD Connect

Anca Robu
PFE
About

Anca Robu
Premier Field Engineer - AD

www.linkedin.com/in/anca-robu-94054117/
arobu@microsoft.com
Agenda

Intro
Understanding the architecture
sourceAnchor Dilemma
Accounts and permissions
Upgrade Azure AD Connect
Scheduler
Intro
Azure AD Connect

- Synchronization services
  - Core component
- ADFS “Easy” Mode
  - Configure an on-premises AD FS farm.
- Health (Monitoring)
  - Agent Connect Health
- Configure Hybrid Env.
  - Password Sync / Passthrough Auth
  - Device Writeback SCP
  - ...
Understanding the architecture
The sync engine

• Creates an integrated view of objects that are stored in multiple connected data sources ...

• This view is determined by the identity information retrieved from connected data sources and a set of rules that determine how to process this information.

• Wizard defines the scope of objects, select the attributes to synchronize (attribute inclusion list) ensuring the required attributes are present.
Sync engine consists of two namespaces

- **connector space** is a distinct staging area that contains representations of the designated objects from a connected data source: stages incoming/outgoing changes, track changes in the data source.
- **metaverse** is a storage area that contains the aggregated identity information from multiple connected data sources, providing a single global, integrated view of all combined objects.
Connector space: Staging Obj. and Placeholders

- **Anchor attribute** uniquely identifies objects in the connected data source (e.g., objectGUID for AD).
- Objects with new identity information are flagged as *pending import* or *export*.
- New Objects are created as **import** or **export Objects**. Export objects become import objects when sync engine receives them in the next import flow.
- **Placeholders** represents a component of an object's hierarchical name that has not been imported into sync engine (OUs, Manager Attributes, ...)

![Diagram](image)
Metaverse Objects

- Sync engine creates metaverse objects by using the information in import objects. Relationship is **MV Obj One-to-Many CS Obj**
- **Extensible MV schema** with a predefined set of object types and associated attributes.
- CS staging object linked to a MV object is called a **joined object** (connector). A staging object that is not linked to a metaverse object is called a **disjoined object** (disconnector). MV objects that do not have a link to any CS object are deleted.
Identity management process

- **Import** process first tries to locate a representation of the object in the CS matching the anchor attribute or distinguished name. Staging objects with updated data are marked as **pending import** (Add, Update, Delete)
- **Inbound synchronization** will project/provision new objects in the MV, Join existing MV objects to a Staging Obj or update attribute values (attribute flow)
- **Outbound synchronization** updates export objects on MV object change:
  - Rename a joined object.
  - Create joined objects, where a metaverse object is linked to a newly created export object
  - Disjoin links between a metaverse object and staging objects, creating a disjoined object.
Identity management process

- During the export process, sync engine examines all export objects that are flagged as pending export in the connector space, and then sends updates to the connected data source.
- Sync engine uses the import process to confirm attribute values that have been exported to the connected data source. A comparison between the imported and exported information enables sync engine to determine whether the export was successful or if it needs to be repeated.
Azure AD Connect – The Rules

• **Scope**
  
  • Determines the rules that are in scope and should be included in the processing: EQUAL, CONTAINS, ISNULL, ISBITSET, ISMEMBEROF ...

• **Join**
  
  • Finding the relationship between the object in the source and an object in the target. For example an object in the CS linking to one in the MV (Inbound Rule).
  • Search for an object already in the metaverse to link to.
Azure AD Connect – The Rules

• Join
  • The join rules are only evaluated once. When a CS object and a MV object are joined, they remain joined as long as the scope of the Synchronization Rule is still satisfied.
  • Only one Synchronization Rule with join rules defined must be in scope: precedence is not managed for Join Rules.
  • The groups in join rules are processed from top to bottom.
  • If there’s not exactly one match, the Link Type is used: only if this option is set to Provision, then a new object in the target is created.
Azure AD Connect – The Rules

• Transformations
  • Define the Attribute Flow as:
    • Direct flows an attribute value as-is with no additional transformations.
    • Constant sets the specified value.
    • Expression uses the declarative provisioning expression language to express how the transformation should be.
  • Precedence is set for each rule: lowest number wins. See default ruleset example (enabled Accounts have higher precedence).
  • Multi-valued attributes from several different Connectors can be merged instead of updated by precedence.
Azure AD Connect – OOB Rules

• **User/contact out-of-box rules:**
  • Must have a sourceAnchor; after creation in AAD this cannot be changed.
  • UserAccountControl must be populated (default in ADDS).

• **Exclusions:**
  • `IsPresent([isCriticalSystemObject])`
  • `IsPresent([sAMAccountName]) = False`
  • `Left([sAMAccountName], 4) = "AAD_", Left([sAMAccountName], 5) = "MSOL_"`
  • Not synchronize objects and Exchange objects that would not work in Exchange Online.
Azure AD Connect – The Rules

The default AD Connect ruleset is complex and carefully built: modifying the ruleset can quickly become a dangerous and daunting task!

The out-of-box sync rules have a thumbprint. If you make a change to these rules, the thumbprint is no longer matching. You might have problems in the future when you try to apply a new release of Azure AD Connect. Only make changes the way it is described in the article below.

Consider the supportability of the customizations, follow the recommendations on how to change the default configuration

The Rules Demo!
sourceAnchor (dilemma)
Definition:

- An attribute immutable during the lifetime of an object. It uniquely identifies an object as being the same object on-premises and in Azure AD. The attribute is also called immutableId.

What to choose?

- objectGUID is unique *but* unfortunately does change when a user is migrated between AD forests: when this happens there’s no easy way to relink the migrated user to his cloud self (your mileage may vary)
- employeeID can be a good choice, should be unique and immutable.
- msDS-ConsistencyGuid adds a bit of rules complexity but is the recommended choice, especially in multi-forest environments.
- sourceAnchor should *not* be case-sensitive and should avoid characters that may vary by case.
Azure AD Connect – sourceAnchor

- Azure AD Connect (1.1.524.0+) now facilitates the use of msDS-ConsistencyGuid as sourceAnchor:
  - Use *msDS-ConsistencyGuid* as the *sourceAnchor attribute* for User objects. ObjectGUID is used for other object types.
  - Rule will try to Join using *msDS-ConsistencyGuid* or *ObjectGUID*.
  - If *msDS-ConsistencyGuid* attribute isn't populated, Azure AD Connect writes its objectGUID value back to the *msDS-ConsistencyGuid* attribute in on-premises Active Directory.
Azure AD Connect – sourceAnchor

• Nice to Know
  • Service account must be granted write permission to the msDS-ConsistencyGuid attribute in on-premises Active Directory.
  • Azure AD Connect (1.1.524.0 and after) stores information in your Azure AD tenant about the sourceAnchor attribute used during installation.
  • Wizard checks the state of the msDS-ConsistencyGuid attribute in your on-premises Active Directory: if attribute is configured on one or more objects falls back to using objectGUID as the sourceAnchor attribute.
  • Azure AD Connect (1.1.552.0 and after) supports switching from ObjectGuid to ConsistencyGuid as the Source Anchor attribute.
  • Azure AD Connect automatically updates the claim rules to use the same AD attribute as sourceAnchor
msDS-ConsistencyGuid
Accounts and Permissions
Azure AD Connect – Accounts

Created/Required Service Accounts

- Active Directory account
- AAD Connect Sync Service Account
  - Virtual Service Account (VSA)
  - Group Managed Service Account (gMSA)
  - Local / domain account
- Azure AD Service account

With build from 2017 March or earlier, do not reset the password on the AAD Connect Sync Service Account since Windows destroys the encryption keys for security reasons.
Azure AD Connect – Accounts

Express settings installation <> required permissions

- Administrator of the local server
  - Creates the local account that is used as the sync engine service account.

- Enterprise Admin credentials
  - Creates an account in Active Directory and grants permissions to it.

- Global administrator role in Azure AD
  - Creation of the Azure AD account that is used for on-going sync operations in Azure AD.
  - Enabling sync in the Azure AD directory.
Azure AD Connect – Accounts

Custom settings installation <> Required permissions

- Administrator of the local server
- Creates the local account that is used as the sync engine service account.
- AD or local user account credentials
- If this account is specified, it is used as the service account for the sync service.
- On-premises ADDS creds for each forest that is connected to AAD
- Domain Administrator for Installation and configuration of the AD FS server role.
- Local admin on the Web application proxy servers
- Domain account that is a local administrator of the AD FS server(s) for the proxy trust credentials
- AD user account credentials for AD FS Service Account
Upgrade Azure AD Connect / Staging Mode
Azure AD Connect – Upgrade Paths

• Upgrade from DirSync or Azure AD Sync
  • In-place migration of all supported custom configurations
  • Side by Side for > 50K objects
  • Will not migrate unsupported configurations (such as removed attribute flows)

• Upgrade Azure AD Connect
  • Automatic Upgrade
  • In-Place Upgrade
  • Swing Migration
Azure AD Connect – Staging Mode

- Use Staging Mode in scenarios as:
  - High availability.
  - Test and deploy new configuration changes.
  - Introduce a new server and decommission the old server.
  - Server is active for import and synchronization, but does not run any exports.
- Disaster Recovery recommendations
  - Rebuild when needed.
  - Have a spare standby server, known as staging mode.
  - Use virtual machines.
- Optional: SQL High Availability
  - Support for SQL AOA from version 1.1.524.0.

You must enable SQL AOA before installing Azure AD Connect!
Scheduler
Azure AD Connect – Scheduler

- Responsible for two tasks:
  - **Synchronization Cycle**
    - Import, sync and export processes
  - **Maintenance Tasks**
    - Renew keys and certificates for Password Reset and DRS.
    - Purges old entries in the operations logs.

- **Configuration**
  - View by using `Get-ADSyncScheduler` in PowerShell
  - Modify:
    - `Set-ADSyncScheduler`
    - `CustomizedSyncCycleInterval`
    - `NextSyncCyclePolicyType`
    - `PurgeRunHistoryInterval`
    - `SyncCycleEnabled`
    - `MaintenanceEnabled`
Azure AD Connect – Scheduler

- Disable the scheduler:
  - `Set-ADSyncScheduler -SyncCycleEnabled $false`

- Start the scheduler
  - Delta or full (initial) Sync Cycle:
    - `Start-ADSyncSyncCycle -PolicyType Delta`
    - `Start-ADSyncSyncCycle -PolicyType Initial`

- Stop the scheduler
  - `Stop-ADSyncSyncCycle`
Azure AD Connect – Scheduler

- Custom scheduler:
  - `Invoke-ADSyncRunProfile -ConnectorName "name of connector" -RunProfileName "name of profile"`

Recommended Order:
1. (Full/Delta) Import from on-premises directories, such as Active Directory
2. (Full/Delta) Import from Azure AD
3. (Full/Delta) Synchronization from on-premises directories, such as Active Directory
4. (Full/Delta) Synchronization from Azure AD
5. Export to Azure AD
6. Export to on-premises directories, such as Active Directory

- Get the status: `Get-ADSyncConnectorRunStatus`