WAYPOINT VENTURES

IDENTITY WITH WINDOWS SERVER

20742 (55351AC)

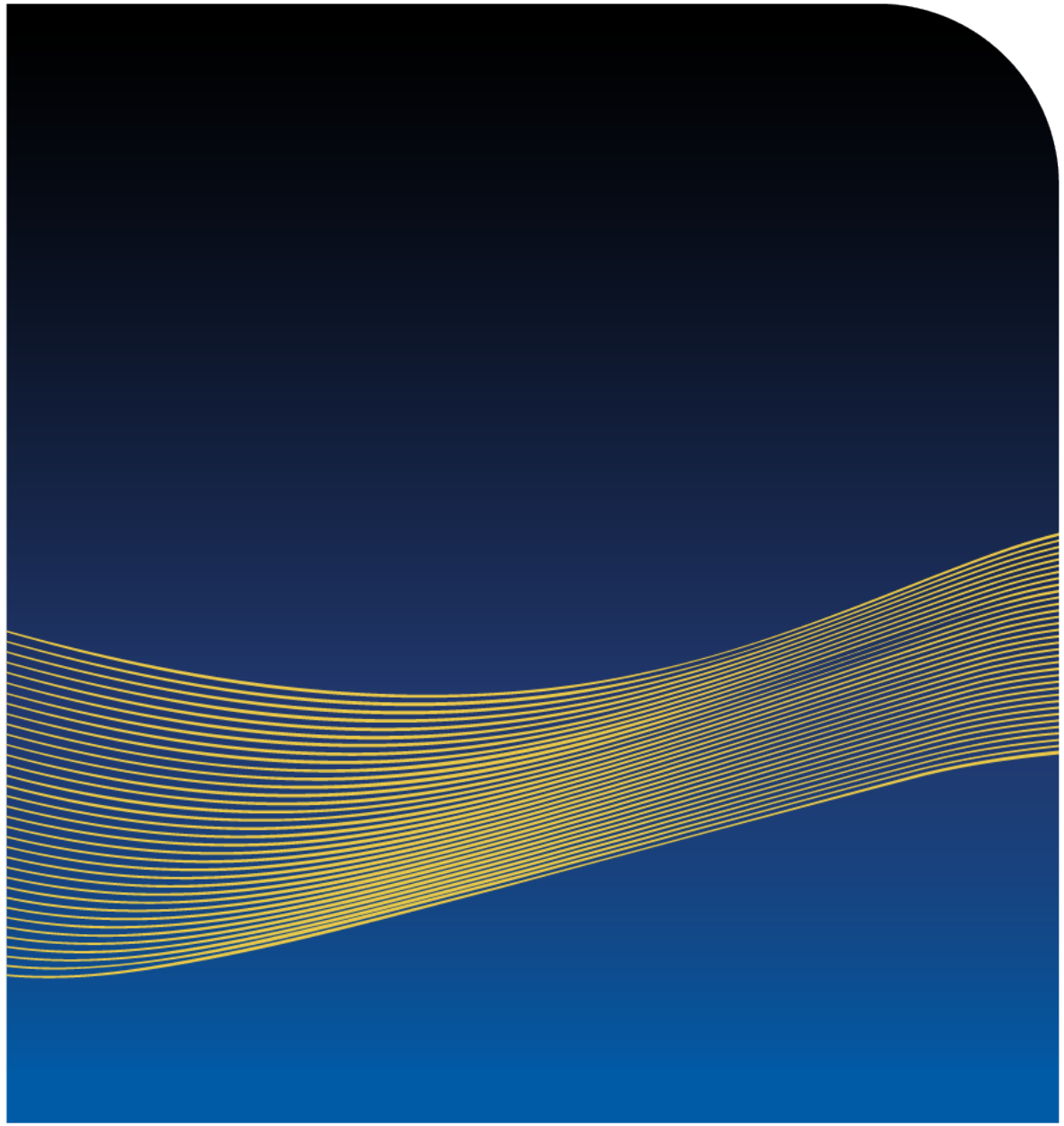
Module 2: Manage directory objects

Module overview

This module introduces managing directory objects.

The lessons in this module are:

- Lesson 1: Manage user accounts
- Lesson 2: Manage groups in AD DS
- Lesson 3: Manage computer objects in AD DS
- Lesson 4: Administer AD DS by using PowerShell
- Lesson 5: Implement and manage OUs



Lesson 1: Manage user accounts

Lesson 1 overview

This lesson introduces managing user accounts.

The topics in this lesson are:

- Create user accounts
- Demonstration: Manage user accounts
- Disable and delete user accounts
- Perform bulk operations on Active Directory objects
- Demonstration: Perform bulk operations in Active Directory Users and Computers
- User account templates
- Demonstration: Use templates to create accounts
- Manage user objects in Azure AD

Create user accounts

User accounts created in AD DS allow users to sign in to the domain from any computer in the domain (except DCs).

- User accounts can be created using:
  - Command-line tools, such as dsadd.
  - Active Directory Users and Computers.
  - Active Directory Administrative Center.
  - PowerShell.
- Many attributes can be configured for user accounts:
  - Some attributes are mandatory, such as the UPN.
  - Some attributes are optional, such as email address or phone number.

Username attributes

There are several name attributes, each with their own unique requirements.

- First Name, Middle Initial, and Last Name
- Full Name (display name)
  - Must be unique in the container.
- User Principal Name (UPN)
  - Must be unique in the forest.
- Pre-Windows 2000 logon name (SamAccountName)
  - Must be unique in the domain.

Attributes on the Account tab

In the properties of a user account, on the **Account** tab, you can set:

- Logon hours.
- Log on to.
- Account expiration date.
- Password-related:
  - User must change password at next sign in.
  - Password never expires.
  - User can’t change password.
  - Store password using reversible encryption.
  - Account is trusted for delegation.
- Smartcard is required for interactive sign in.

Additional attributes

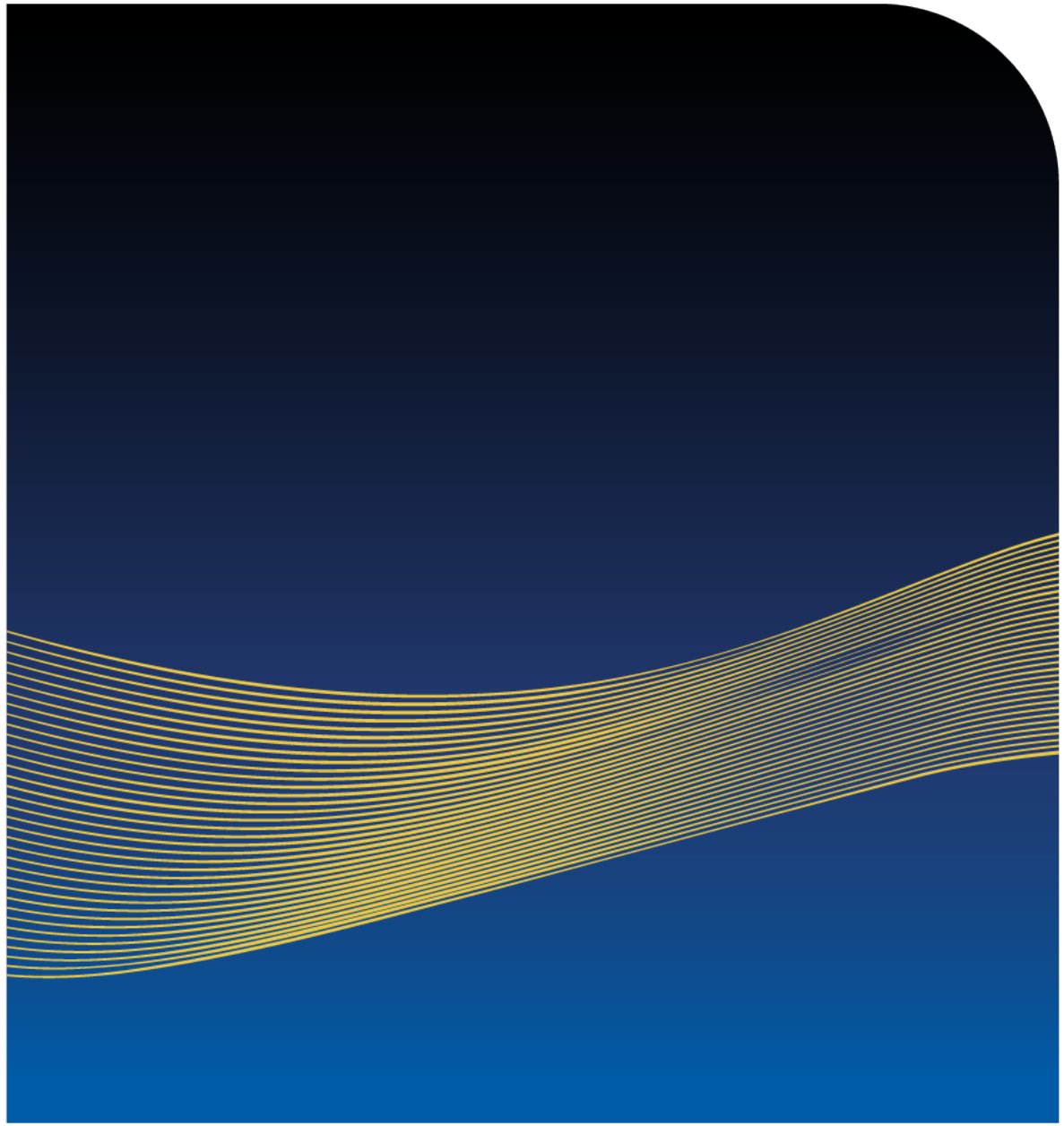
In the properties of a user account, you can also configure the following attributes:

- **Organization** tab:
  - Job title, department, manager
- **General** tab:
  - Office, email, webpage, telephone number
- **Member of** tab:
  - Groups the user belongs to
- **Profile** tab:
  - Where the user profile is stored (locally or a network server)

Other locations for user-related policies

Some settings can be configured with Group Policy or the AD Administrative Center:

- Use Group Policy to configure settings for the Kerberos authentication protocol.
- Use AD Administrative Center to configure fine-grained password policies:
  - Allows different password policies for different groups of users.



Demonstration: Manage user accounts

Manage user profiles

User profiles are a set of folders and files that store a user's files and settings:

- By default, user profiles are stored locally.
- You can configure roaming profiles for users:
  - Profiles are copied to and from a network server.
  - On the **Profiles** tab, specify the network location using UNC syntax:
    - Example: **\\server1\profiles\Adam**
  - A roaming profile becomes available to the user at any computer they sign in to.
  - Roaming profiles can generate significant amounts of network traffic.
- Microsoft recommends using Folder Redirection in Group Policy, instead of roaming profiles.

Logon scripts

Logon scripts and the **Home** folder can be specified on the **Profile** tab in the user properties:

- Logon script:
  - The name of a script that runs when the user signs in.
  - Microsoft considers this a legacy feature and recommends using Group Policy instead.
    - Group Policy allows you to configure logon, logoff, startup, and shutdown scripts.
- **Home** folder:
  - Default location for user documents.
  - Considered a legacy feature as most modern apps will ignore this setting:
    - Modern apps attempt to save documents in the user's **Documents**, **Pictures**, **Music**, or **Video** folders.

Use Group Policy to manage profiles

- Microsoft recommends using Folder Redirection in Group Policy, as an alternative to roaming profiles.
- Folder Redirection allows you to move several folders from a user's local profile to a network location, including:
  - **Documents**, **Pictures**, **Music**, **Videos**, **Desktop**, **Start Menu**, **Favorites**
- Folder Redirection is different from roaming profiles because instead of copying the profile folders back and forth, the folders are moved to the specified network location.
- When users access a redirected folder, they're transparently connected to the network location where it resides.

Folder Redirection options

- Basic redirection:
  - Redirects folders to a shared folder on a server.
  - Each user is given a folder under the specified location.
  - You can also choose to redirect all users to the same folder. Useful for teams working on shared files in a single folder.
- Advanced redirection:
  - Redirects different groups of users to different locations.

Delete user accounts

- User accounts are represented by a unique alphanumeric string known as the SID (security identifier). Deleting a user permanently deletes its SID.
- If an account is deleted by mistake, and you create another account with the same name and password, Windows will generate a new SID for that account, and treat it as a new account.
  - The new account won't have the same rights, permissions, or group membership as the account you deleted.
- It's safer to disable an account at first:
  - A disabled account retains its SID, rights, and permissions.
  - A disabled account can be easily reenabled.

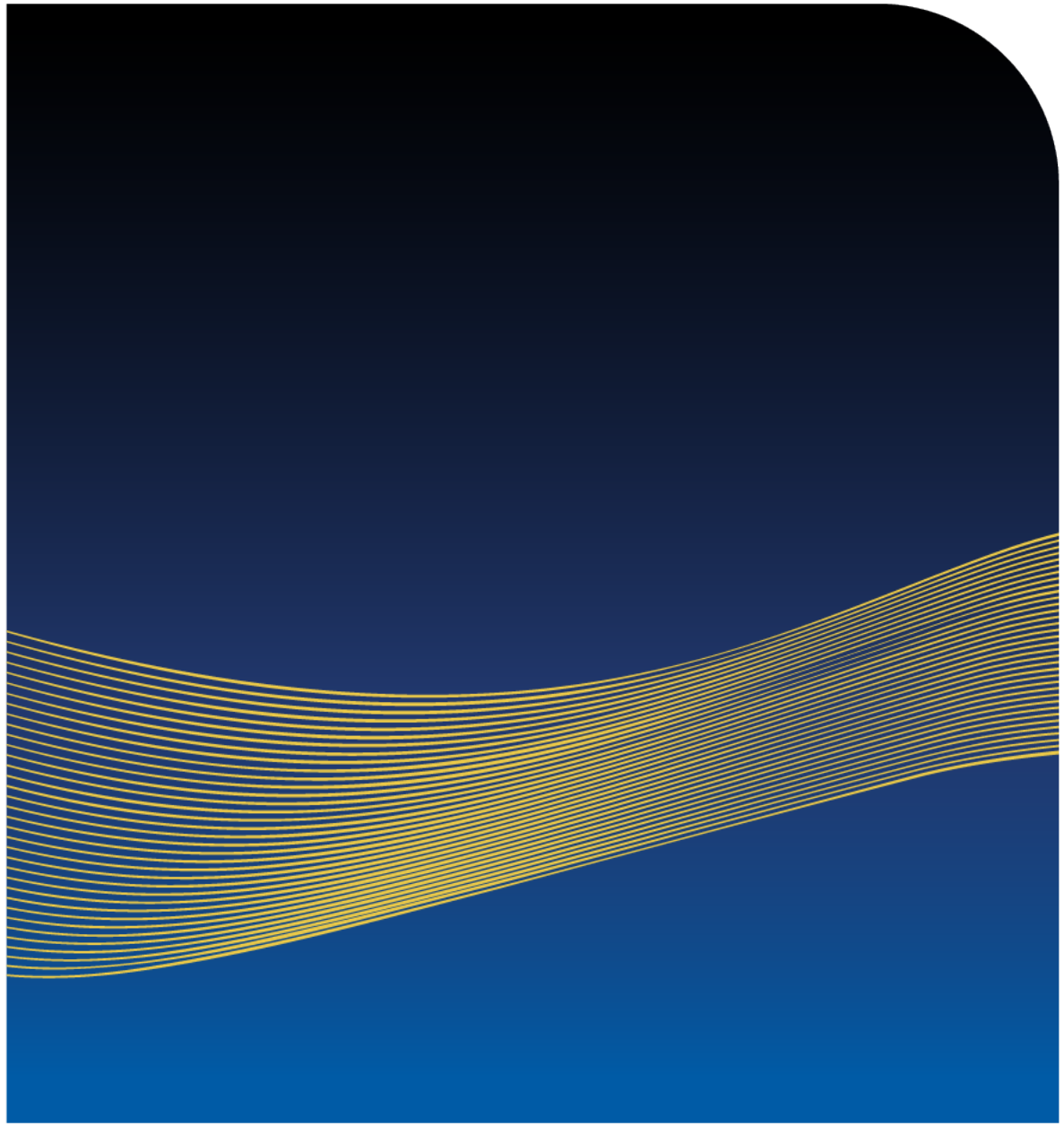
Disable user accounts

Disabling an account if it won’t be used for some time reduces the attack surface on your network:

- Malicious hackers can’t sign in using a disabled account.
- The account can be reenabled when the user returns to work.
- It is a good practice to disable accounts you intend to delete, until you are sure that the account will never be needed again.

Perform bulk operations on Active Directory objects

- If you need to perform the same operation on multiple objects in Active Directory, you can use both command-line tools and graphical tools.
- Command-line tools:
  - Legacy tools: dsadd, dsmod, dsrm, csvde, ldifde
- PowerShell cmdlets:
  - **New-ADUser**, **Set-ADUser**
- Graphical tools:
  - AD Users and Computers, AD Administrative Center
  - You can select multiple accounts and modify their common settings.



Demonstration: Perform bulk operations in AD Users and Computers

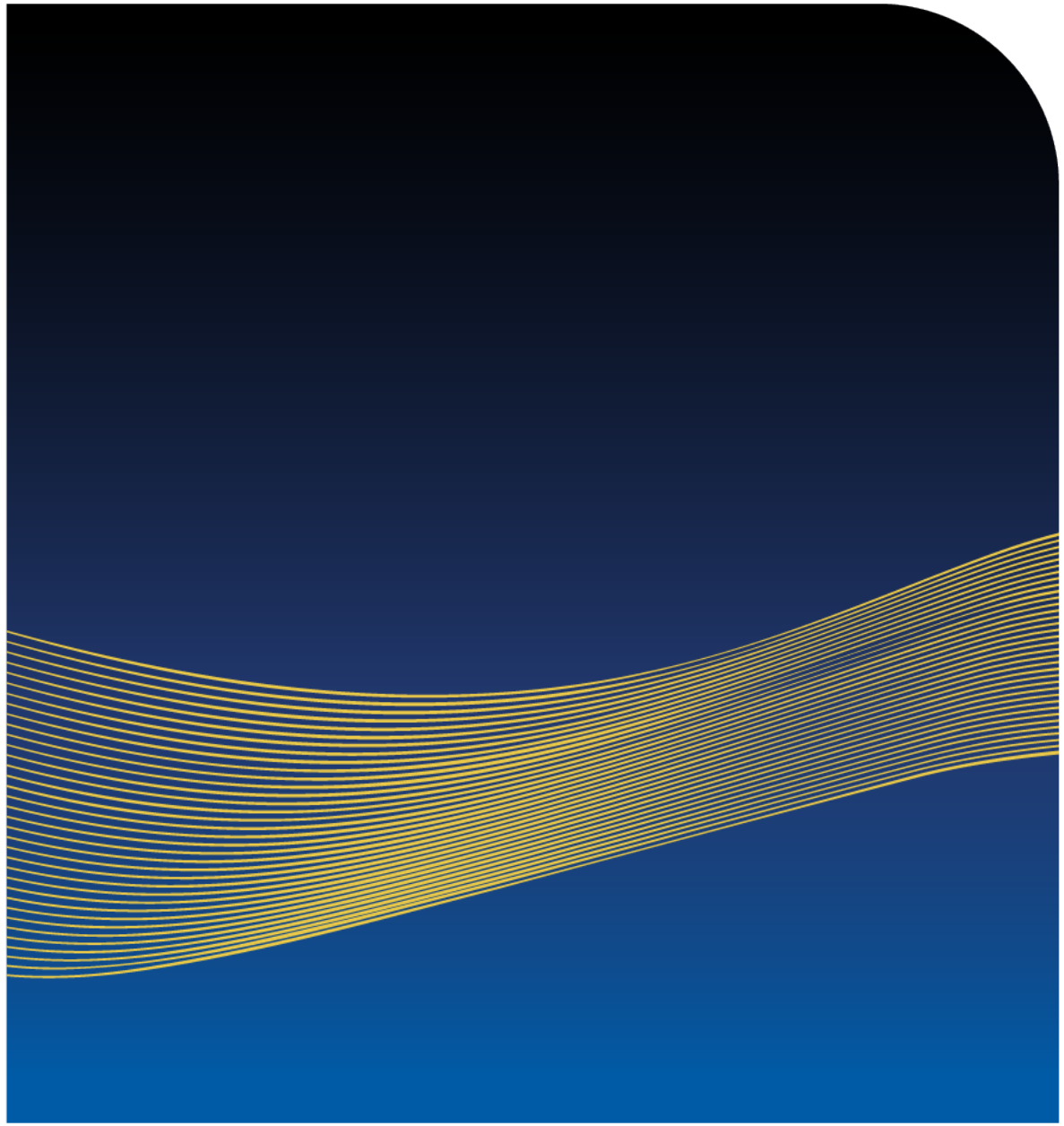
User account templates

If you need to create multiple, similar accounts, create a template account:

- Created just like any user account.
- Give it an obvious name, such as _TempWorkers.
  - The “_ “ in the name isn’t required. It causes the template to display at the top of the list of users in a container.
- Configure the template with properties that will be common to all the users you’ll be creating, such as group membership or logon hours.
- Copy the template to create new accounts.
  - Most properties are copied from the template to the newly created user.

Configure network locations in a template account

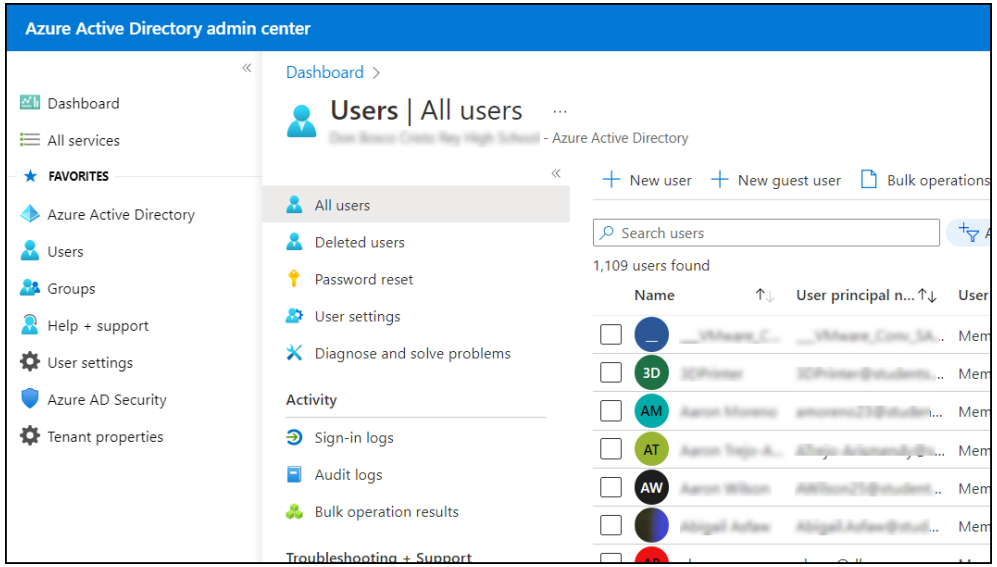
- When specifying network locations in a template account, such as the profile location, use the environment variable %username%.
  - Example: **\\server1\profiles\%username%**
- When you copy the template, Windows automatically replaces %username% with the user's sign-in name.

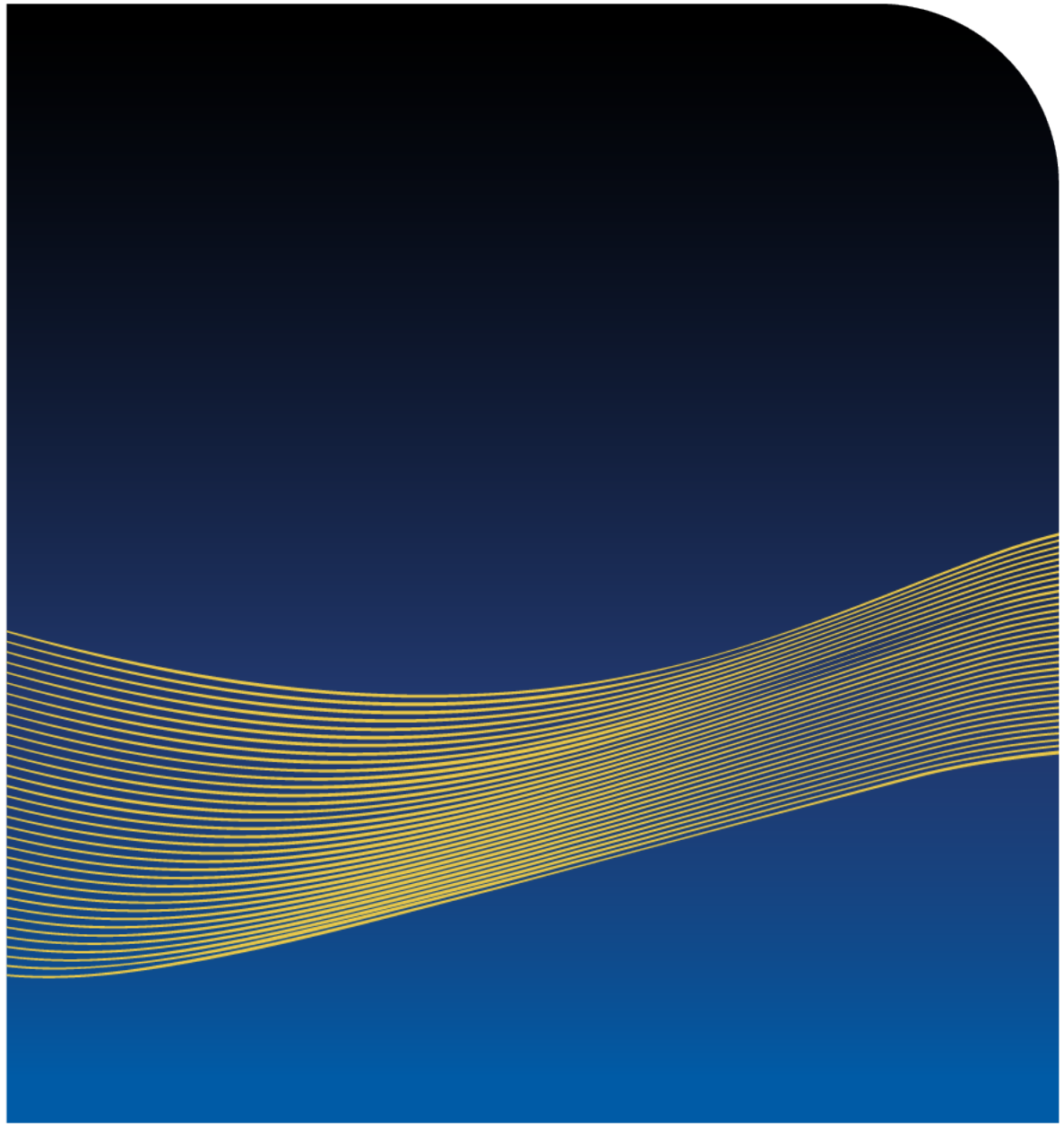


Demonstration: Use templates to create accounts

Manage user objects in Azure AD

- Azure AD is an implementation of Active Directory in the cloud:
  - Organizations must subscribe to have access to Azure AD.
- You can use Azure AD, or use it in conjunction with on-premises Active Directory:
  - When you have both, it is called a hybrid implementation.
  - You can install the AD connector on premises to sync data between Azure AD and on-premises Active Directory.





Lesson 2: Manage groups in AD DS

Lesson 2 overview

This lesson introduces managing user accounts.

The topics in this lesson are:

- Security and distribution groups
- Group scopes
- Implement group management
- Delegate management of groups in Active Directory
- Restricted groups
- Default groups
- Special identities
- Demonstration: Manage groups in Windows Server
- Manage groups in Azure AD

Security and distribution groups

- You can create security and distribution groups.
- Security groups:
  - Have a SID.
  - Can be assigned rights and permissions.
  - Can be email-enabled.
- Distribution groups:
  - Do not have a SID.
  - Can’t be assigned rights and permissions.
  - Are intended for email distribution.
- Converting group type:
  - If you convert a security group to distribution, it loses its SID.

Group scopes

The scope of a group indicates where it can be used to assign rights and permissions.

Group scope	Can be assigned rights and permissions…
Local	To resources on the local computer only
Domain Local	Resources in its own domain only
Global	To any resource in the forest
Universal	To any resource in the forest

Group membership rule

There are rules for group membership.

Group scope	Can contain members from...
Local	Any domain in the forest and local user accounts
Domain Local	Any domain in the forest
Global	Its own domain only
Universal	Any domain in the forest

Group nesting

There are rules about which types of groups can be members of which types of groups.

Group	Can contain....
Local	Domain local, global, or universal groups from any domain in the forest
Domain Local	Other domain local groups, global, or universal groups from any domain in the forest
Global	Global groups from its own domain
Universal	Global and universal groups from any domain in the forest

Group conversion

Some groups can be converted.

Group	Can be converted into...
Local	No other groups
Domain Local	Universal groups, but not if they contain other domain local groups
Global	Universal groups
Universal	Global and universal groups from any domain in the forest

Implement group management: IGDLA

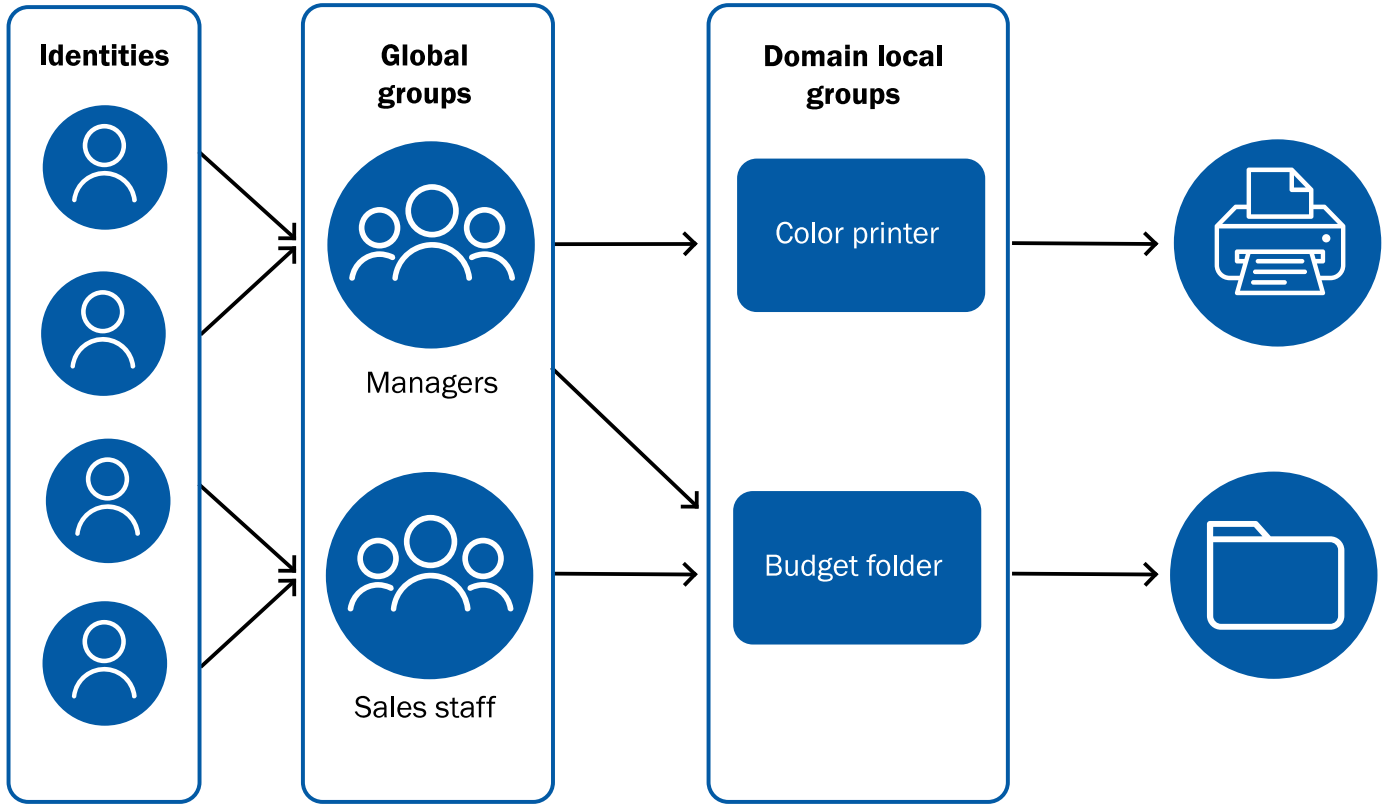
- Microsoft recommends that **Identities** (users and computers) should be added to **Global** groups.
- **Global** group should be added to **Domain Local** groups.
- **Domain Local** groups should be given **access** to resources.

Global groups should reflect the type of users that are members, such as managers, sales staff, HR (Human Resources).

Domain Local groups provide access to resources and are often named after the resource, such as ColorPrinter or BudgetFolder.

Managing access becomes a simple matter of adding the appropriate **Global** group to the appropriate **Domain Local** group.

The IGDLA strategy



Implement group management: IGUDLA

Universal groups impose a cost on the network. Any time you change a universal group, it triggers replication to all Global Catalog servers in the forest.

- Consider using universal groups if your users are in multiple domains, *and* you need to provide access to resources in multiple domains.
- **Universal** groups are useful for bringing together **Global** groups from multiple domains into one **Universal** group.
- The **Universal** group is added to **Domain Local** groups in multiple domains.
- The **Domain Local** groups provide **access** to resources in multiple domains.

Delegate management of groups in Active Directory

- Delegation is normally performed by setting permissions in the properties of an object.
  - Delegation is covered in detail in Lesson 5 on OUs.
- With groups, however, there is a simpler method:
  - On the **Managed by** tab, add the user who is to manage the group.
    This user will be able to modify the membership of the group.

Restricted groups

- Restricted groups is a setting in Group Policy. You can:
  - Specify the membership of a group.
- When you apply the **Members of this group** setting, the membership list is updated:
  - Users specified in the policy are added to the group.
  - Users not specified in the policy are deleted from the group's membership list.
- If you modify the **This group is a member of** setting as follows:
  - The group is added to the specified group.

Default groups

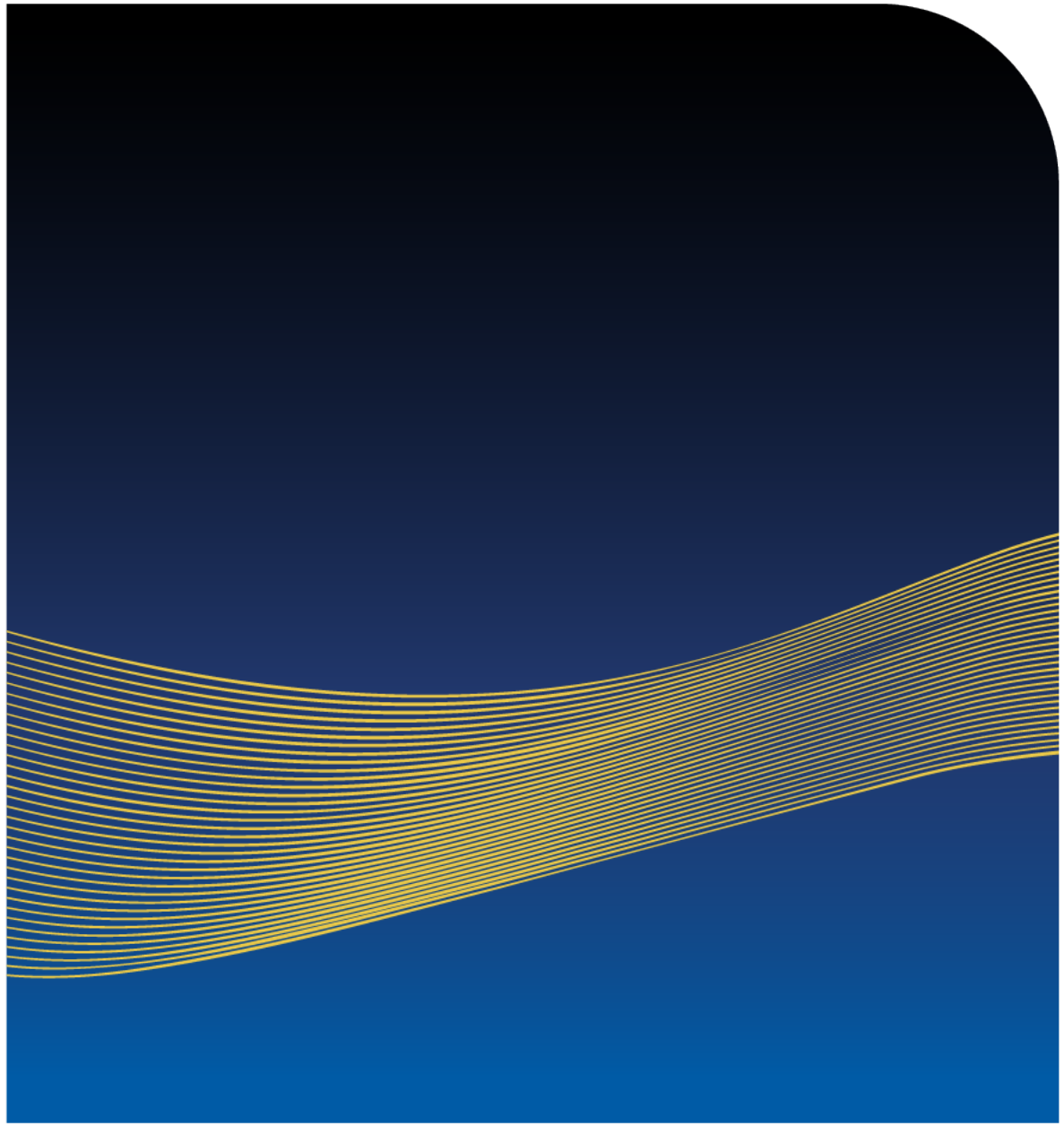
- Default *builtin local* groups are created automatically and given preassigned rights and permissions.
  - Examples include **Administrators**, **Backup Operators**, **Account Operators**
- In Active Directory, these groups display in the Builtin container:
  - The scope of many of these groups is *all domain controllers*.
- Default *global* groups are created in the Users container.
  - Examples include **Domain Admins** and **Domain Users**
- In keeping with the IGDLA strategy, the builtin local groups are given permissions and rights and the builtin global groups are added as members.
  - Example: **users -> Domain Admins > Administrators**

Protected groups

- Many builtin groups are defined as protected groups.
  - Protected groups inherit permissions from the group, rather than the OU.
- Protected groups include:
  - Administrators
  - Account Operators
  - Backup Operators
  - Cert Publishers
  - Domain Admins & Enterprise Admins
  - Print Operators
  - Server Operators

Special identities

- Special identities are groups that are created automatically, but their membership can’t be assigned.
  - Users are added to these groups dynamically, based on activity.
- For example:
  - A user signing in locally is added to the **Interactive** group.
  - The same user connecting across the network is added to **Network**.
- Special identities include:
  - Anonymous logon
  - Authenticated Users
  - Everyone
  - Creator Owner

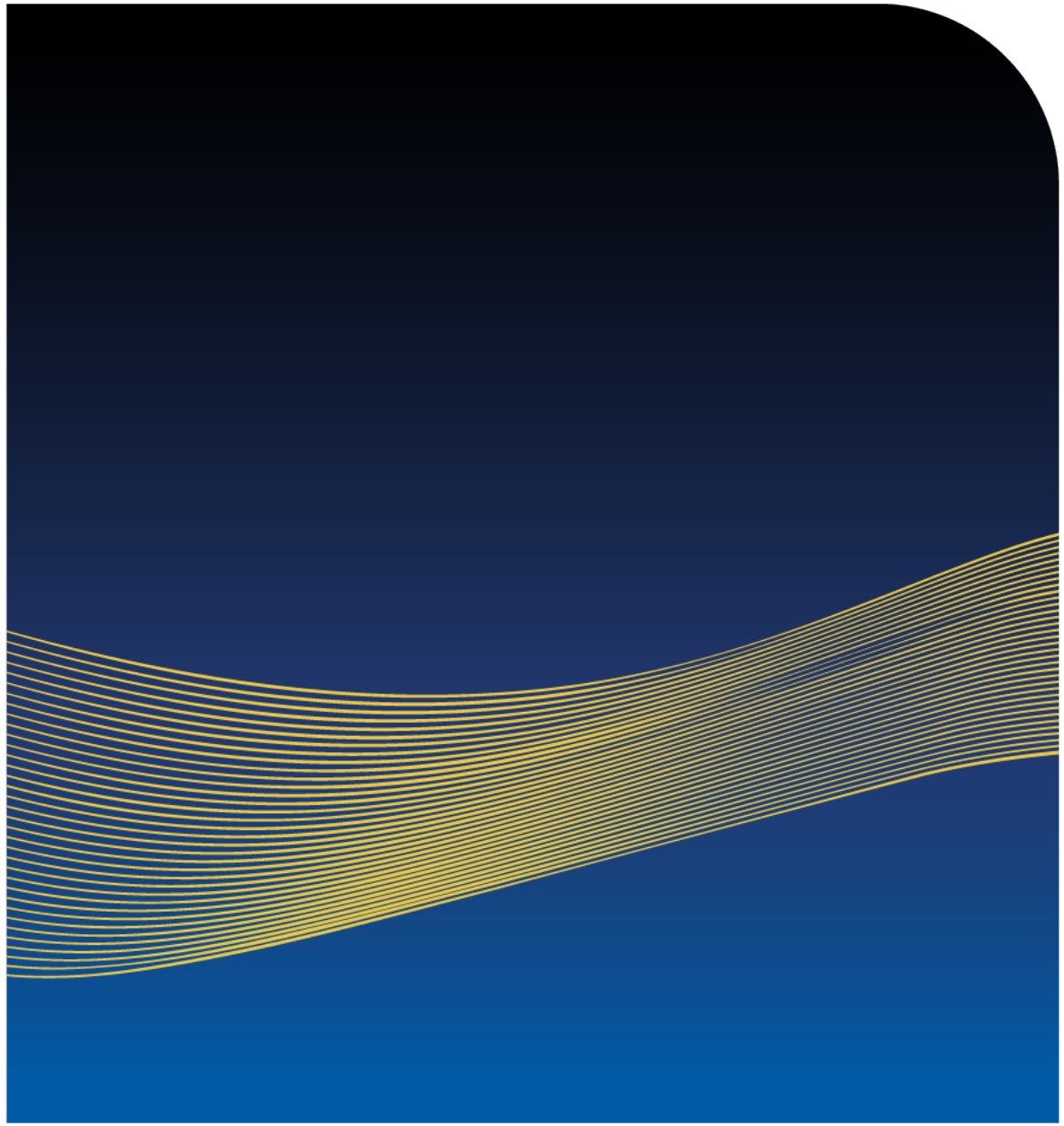


Demonstration: Manage groups in Windows Server

Manage groups in Azure AD

- Managed through the online Azure portal or the Microsoft 365 portal.
- The Azure portal allows you to add members to the group by:
  - Explicitly assigning members.
  - Dynamically adding members.
- To dynamically add members, a rule builder is used to create a rule.
  - Rules allow you to define a query with:
    - A property, such as **department**.
    - An operator, such as **equals**.
    - A value, such as **sales**.

```
(user.department -eq 'sales')
```



Lesson 3: Manage computer objects in AD DS

Lesson 3 overview

This lesson introduces managing computer accounts.

The topics in this lesson are:

- The default Computers container
- Create an OU structure for managing computer objects
- Control who can create computer objects
- Join a computer to a domain
- Computer accounts and secure channels
- Offline domain joins

The default Computers container

- Computers is a builtin container.
- It is not an organizational unit (OU).
  - Can’t have group policies linked to it.
  - Can delegate control over it.
  - LDAP abbreviation of CN is used, instead of OU.
- LDAP distinguished name:
  - CN=computers,DC=Contoso,DC=com
- Set an OU as the default location for computers:
  - Redircmp OU=contosocomputers,DC=Contoso,DC=Com

Create an OU structure for managing computers

Depending on your OU management strategy, you can:

- Create OUs to represent locations or departments.
- Create OUs to represent different types of computers: desktops, laptops, tablets, or servers.
- Or, use a combination of these:
  - Top-level OUs for departments or locations.
  - Child OUs to represent different types of computers.

Control who can create computer objects

- Joining a computer to a domain creates a computer account or attaches to a pre-staged computer account.
- The ability to join computers to the domain resets on:
  - The user right **Add workstations to domain**, as configured in the **Default Domain Controllers Policy**.
    - By default, it allows any authenticated user to add up to 10 computers to the domain.
  - The delegated **permissions** to create computer objects.
    - An administrator with this permission can add an unlimited numbers of computers.

Join a computer to a domain

When you join a computer to the domain, it creates the computer account in Active Directory. Alternatively, you can pre-stage the computer account and then join to the domain.

- In **Advanced System properties**, on the **Computer Name** tab, select the **Change** button:
  - Enter the name of the domain.
  - Provide credentials when challenged.
  - Reboot.
- Remove a computer from the domain:
  - Use **Advanced System properties**, select **Workgroup**, and enter a name.
  - Provide credentials when challenged.
  - Reboot.

Hybrid join

- If you have on-premises Active Directory and you're signed up for Azure AD, you can join the computer to both. This is called a *hybrid join*.
- Requires the AD connector to be installed.
  - The AD connector syncs objects between on-premises Active Directory and Azure AD.

1. Join the computer to the on-premises domain.
   - Wait for sync to complete.
2. Verify the join to Azure AD.
   a. At the computer, enter **dsregcmd /status** at an administrator command prompt.
   b. In the Azure portal, verify the computer is listed with a **Join type** of **Hybrid Azure AD joined**.

Secure channel failures

- Computers sign in to the domain by providing their secret password:
  - The secret password is negotiated when the computer first joins the domain and is changed every 30 days.
  - A secure, encrypted connection is created between the computer and the DC, if the sign in is successful.
- Secure channels can fail when:
  - You restore a computer from a backup.
  - You reinstall the operating system.
  - The computer account has been deleted.
- You'll get error messages that indicate that the trust with the domain has failed.

Reset the secure channel

- Reset the computer account in **AD Users and Computers**, and then rejoin the computer to the domain.
- Command-line tools:

  Dsmod computer "*computerDN*" -reset

  netdom reset <*computername*> /domain <*DomainName*> /UserO *UserName* /PasswordO {*Password | **}

  nltest /server:<*servername*> /sc_reset:<*domain\domaincontroller*>

- PowerShell:

  Test-ComputerSecureChannel -Repair

Offline domain joins

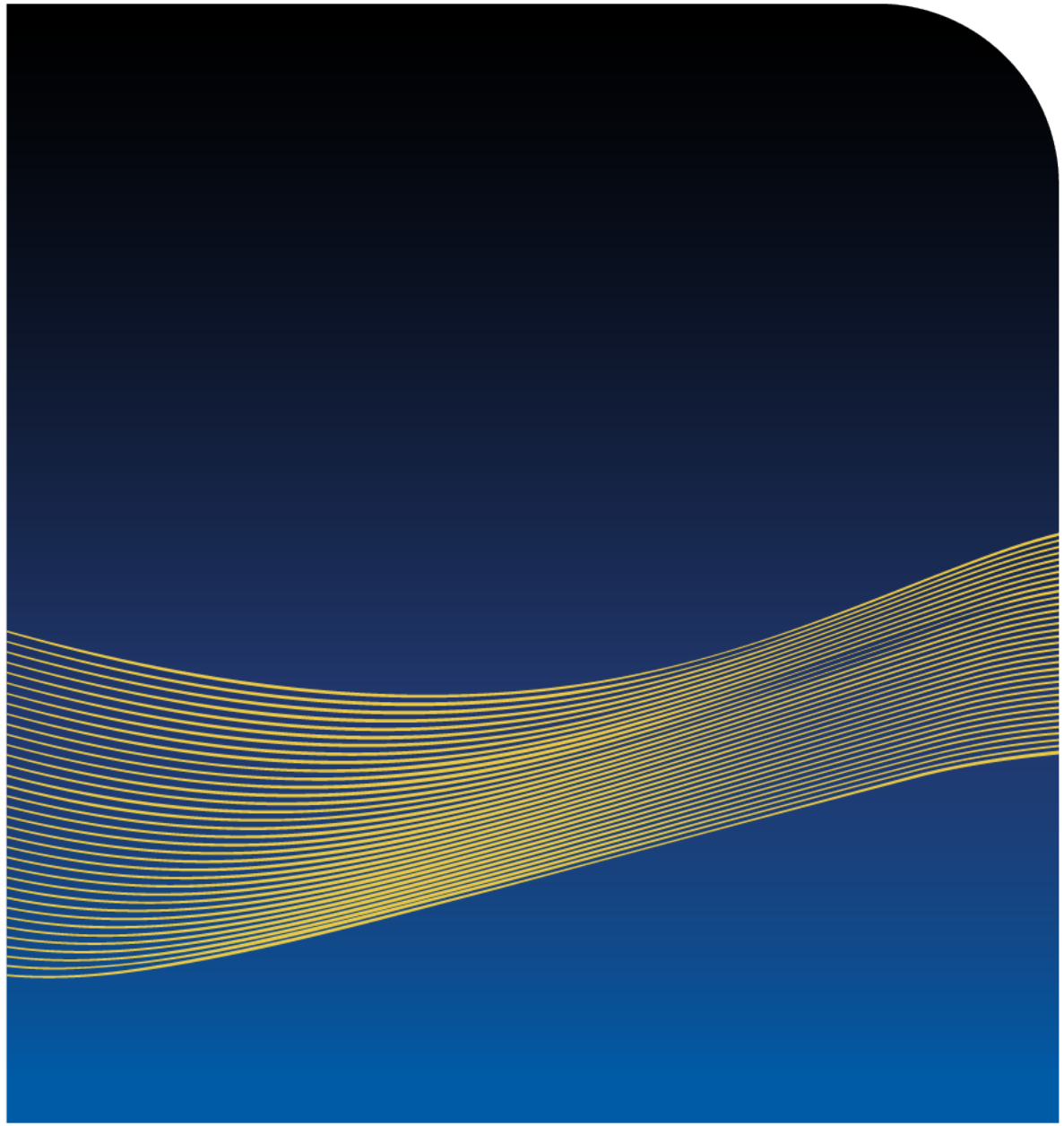
You can join computers to the domain, even when there's no network connectivity to the DCs:

- At a domain-joined computer, create a file with this command:

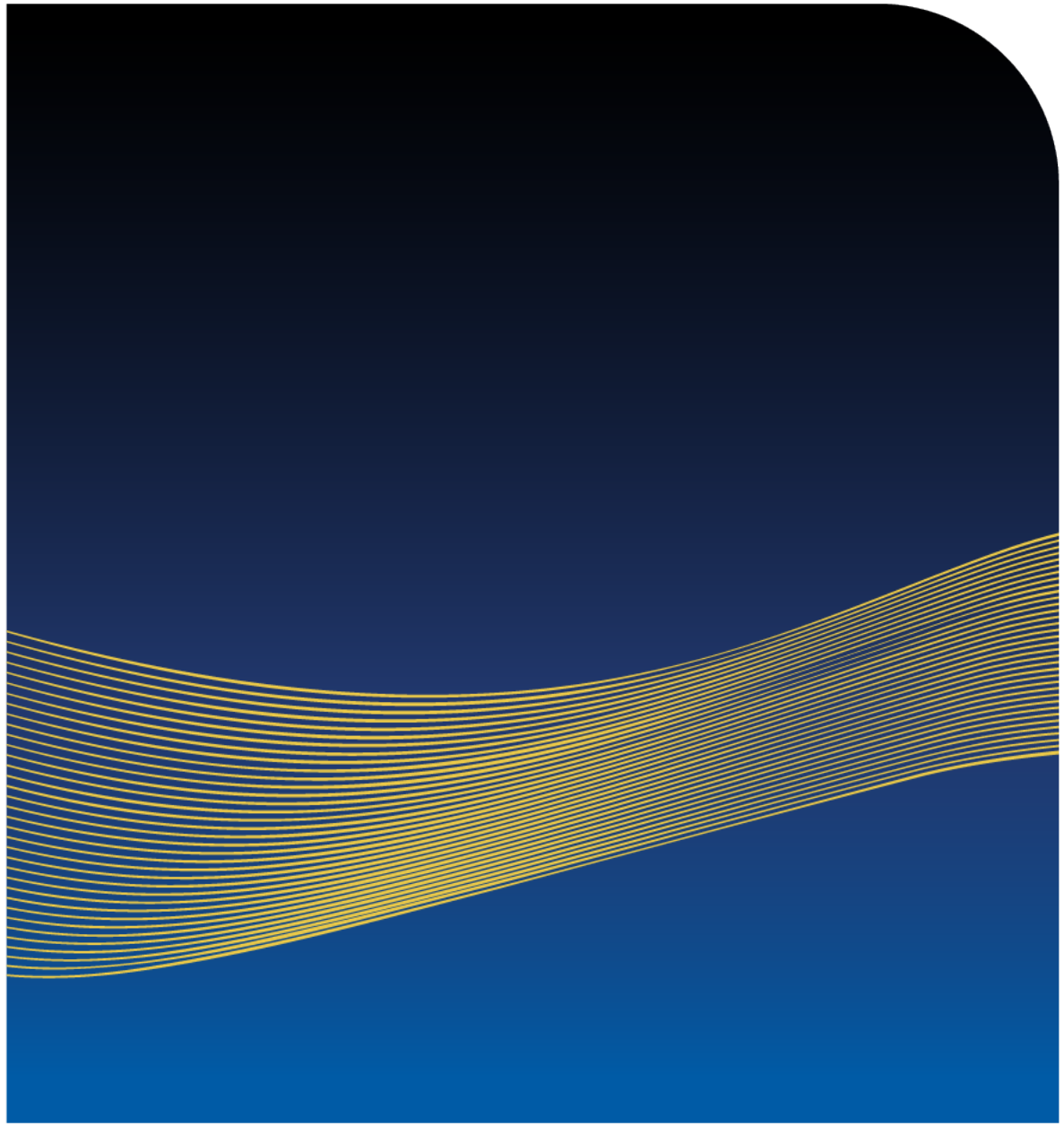
  djoin.exe /Provision /Domain <*DomainName*> /Machine <*MachineName*> /SaveFile <*filepath*>

- Copy the file to the computer that will be offline-joined, and then run this command:

  djoin.exe /requestODJ /LoadFile <*filepath*> /WindowsPath <*location of the Windows folder*> /localOS



Lab 2: Manage AD DS objects



Lesson 4: Administer AD DS by using PowerShell

Lesson 4 overview

This lesson introduces using PowerShell to manage objects in AD DS.

The topics in this lesson are:

- Use Windows PowerShell to manage user accounts
- Use PowerShell for bulk operations
- Demonstration: Use graphical tools to perform bulk operations
- Query objects with PowerShell
- Use text files for bulk operations
- Demonstration: Perform bulk operations with Windows PowerShell

Use Windows PowerShell to manage user accounts

Cmdlets:

- New-ADUser
- Set-ADUser
- Remove-ADUser
- Set-ADAccountPassword
- Set-ADAccountExpiration
- Unlock-ADAccount
- Enable-ADAccount
- Disable-ADAccount

Parameters for **New-ADUser**:

- AccountExpirationDate
- AccountPassword
- ChangePasswordAtLogon
- Department
- HomeDirectory
- HomeDrive
- GivenName
- Surname
- Path

PowerShell cmdlets to manage groups

Cmdlets:

- New-ADGroup
- Set-ADGroup
- Get-ADGroup
- Remove-ADGroup
- Add-ADGroupMember
- Get-ADGroupMember
- Remove-ADGroupMember
- Add-ADPrincipalGroupMembership
- Get-ADPrincipalGroupMembership
- Remove-ADPrincipalGroupMembership

Parameters for New-ADGroup

Parameters:

- Name
- GroupScope
- DisplayName
- GroupCategory
- ManagedBy
- Path
- SamAccountName

PowerShell cmdlets to manage computer accounts

Cmdlets:

- New-ADComputer
- Set-ADComputer
- Get-ADComputer
- Remove-ADComputer
- Test-ComputerSecureChannel
- Reset-ComputerMachinePassword

Parameters for **New-ADComputer**:

- Name
- Path
- Enabled

PowerShell cmdlets to manage OUs

Cmdlets:

- NewADOrganizationalUnit
- Set-ADOrganizationalUnit
- Get-ADOrganizationalUnit
- Remove-ADOrganizationalUnit

Parameters for **New-ADOrganizationalUnit**:

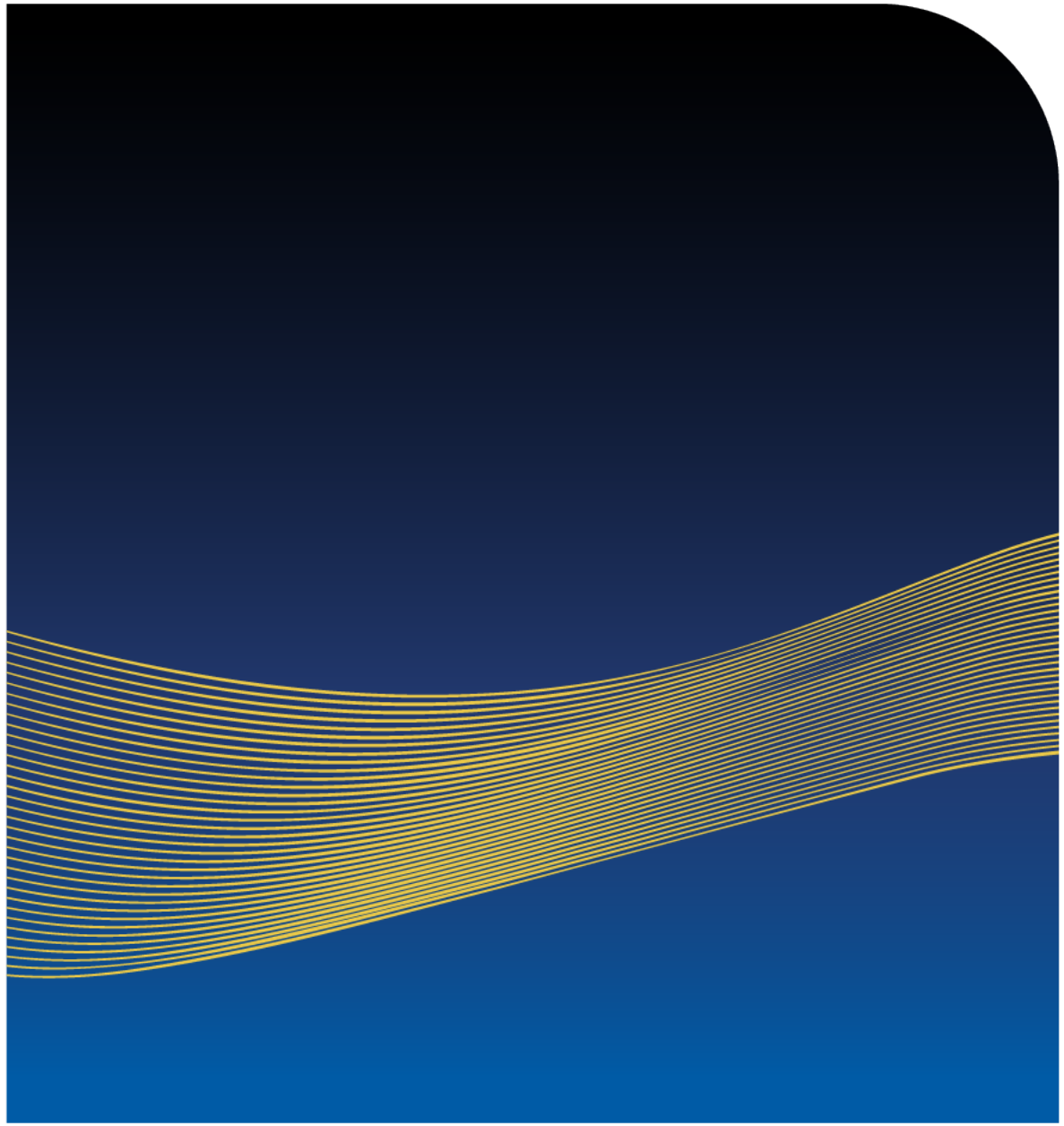
- Name
- Path
- ProtectedFromAccidentalDeletion

Use PowerShell for bulk operations

- PowerShell can create and modify objects by importing data from a text file or CSV file.
  - Many databases and spreadsheets can export data as a CSV file.
- For example, a list of new users could be stored in a spreadsheet and exported to a CSV file.
  - PowerShell cmdlets or scripts can be used to import the CSV file to create the accounts.
- PowerShell cmdlets and scripts are powerful, and you need to be cautious,
  - Test out scripts thoroughly in a lab environment or on a small set of objects.

Using graphical tools for bulk operations

In **AD Users and Computers** or **AD Administrative Center**, you can select multiple objects and modify their properties, delete them, or move them to another OU.



Demonstration: Use graphical tools to perform bulk operations

Query objects with PowerShell

The various **Get-AD*** cmdlets support some common parameters:

- SearchBase
- SearchScope
- ResultSetSize
- Properties

Use the Filter parameter to create a query

Operators used with the filter parameter

Operator	Description
-eq	Equal to
-ne	Not equal to
-lt	Less Than
-le	Less than or equal to
-gt	Greater than
-ge	Greater than or equal to
-like	Uses wildcards and pattern matches

Parameters for Search-ADAccount

Parameters:

- AccountDisabled
- AuthType
- AccountExpired
- AccountExpiring
- AccountInactive
- LockedOut
- PasswordExpired
- PasswordExpiring
- PasswordNeverExpires
- ComputersOnly
- UsersOnly
- TimeSpan
- DateTime
- SearchBase
- SearchScope

Find and modify objects with pipelining

- The pipeline symbol **|** allows you to feed the output of one cmdlet as the input of another cmdlet.
- In this example, **Get-ADUser** with the **filter** parameter is used to find users with the department attribute set to **Sales.** The list is piped into **Set-ADUser**, which modifies the **department** attribute to **Marketing.**

```
Get-ADUser –filter {department –eq “Sales”} | Set-Aduser –department
“Marketing”
```

Use text files for bulk operations

Data can be listed in a text file and **Get-Content** can be used to get that list and pipe it into a cmdlet, such as **Set-ADUser**.

- This example retrieves a list of users from a file called **mylist.txt** and pipes that into **Set-ADUser** to set the department attribute to **Marketing**:

```
Get-Content C:\mylist.txt | Set-ADuser -Department “Marketing”
```

Use comma-separated value (CSV) files (1 of 2)

CSV files can be exported from spreadsheets and databases.

- In this example, each row represents user objects, and the columns represent the attributes of each object.
- The first row contains the names of the attributes in each column.

SamAccountName	Department	Mail
MGomez	Sales	MGomez@contoso.com
HJarvis	Sales	HJarvis@contoso.com
QWatson	HR	QWatson@contoso.com

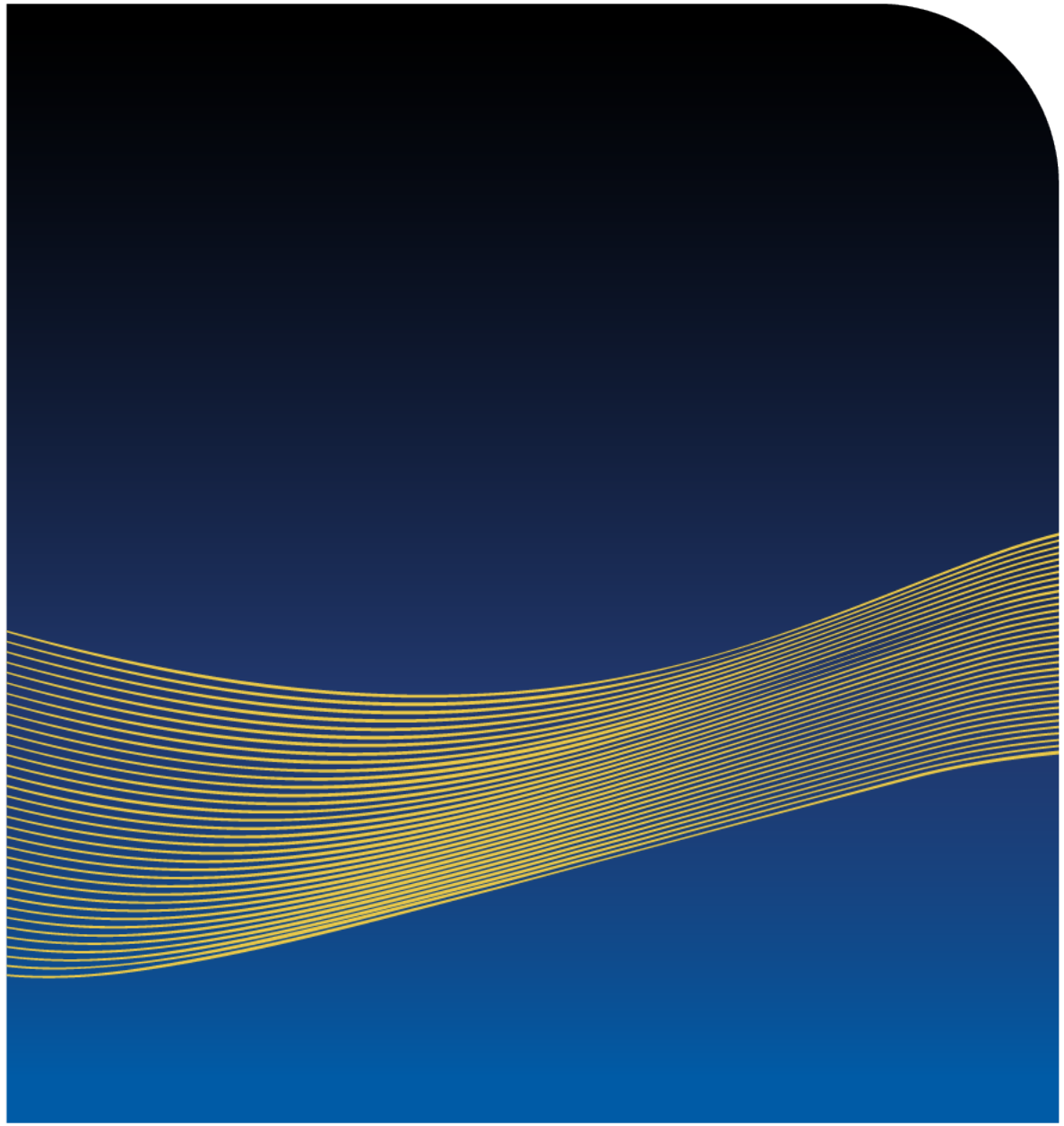
Use comma-separated value (CSV) files (2 of 2)

You can also create CSV files with a simple text editor:

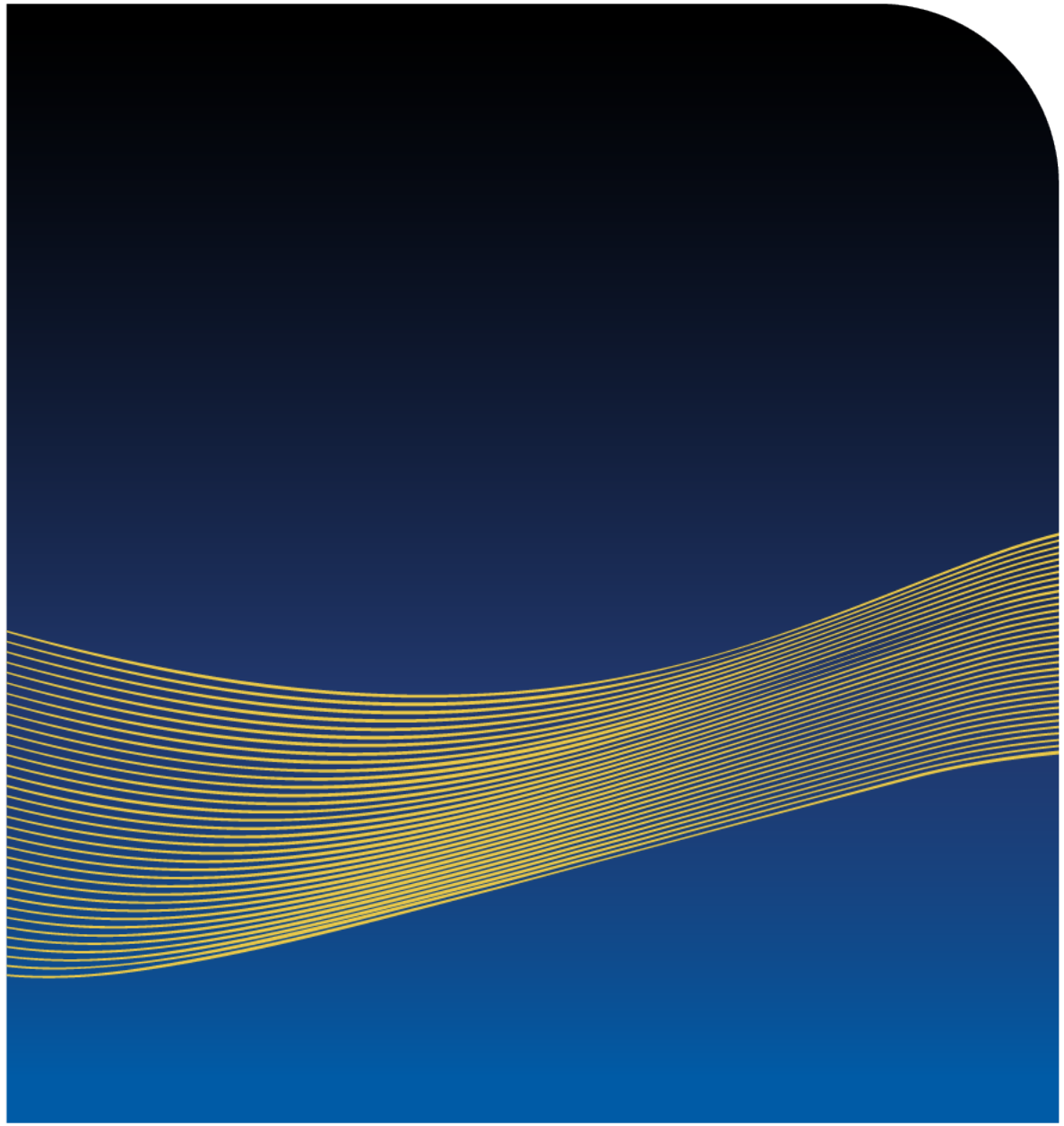
SamAccountName,Department,Mail
MGomez,Sales,MGomez@contoso.com
HJarvis,Sales,HJarvis@contoso.com
QWatson,HR,QWatson@contoso.com

- The first row contains the attribute names, separated by commas.
- **Import-Csv** can be used to import the contents of the CSV file, and processed in a **Foreach** loop:

```
$users=import-csv -LiteralPath "C:\users\public\widget.csv"
foreach ($user in $users)
{
Write-Host $user.samaccountname "works in" $user.department
}
```



Demonstration: Perform bulk operations with Windows PowerShell



Lesson 5: Implement and manage OUs

Lesson 5 overview

This lesson introduces the creation and management of OUs.

The topics in this lesson are:

- Plan OUs
- OU planning strategies
- Delegate administrative control
- Create OUs
- Manage permissions in Active Directory
- Demonstration: Delegate administrative permissions on an OU

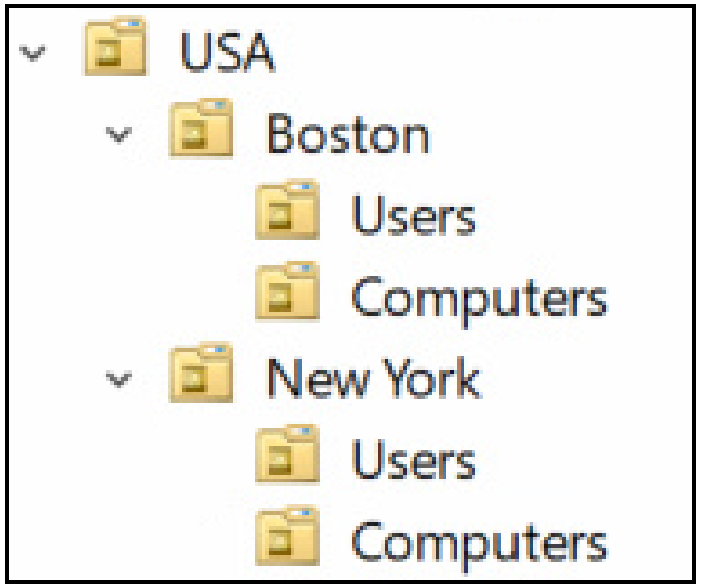
Plan OUs

OUs can have permissions set on them, and they can have group policies linked to them.

- Setting permissions on an OU allows you to delegate administration.
- Linking group policies to OUs allows you to configure thousands of settings for users or computers.
- Your OU design should be based on the following administrative requirements:
  - If you need to delegate control over objects, put them in the same OU and set permissions on that OU.
  - If you need to apply distinct policies to User or Group objects, put them in the same OU.

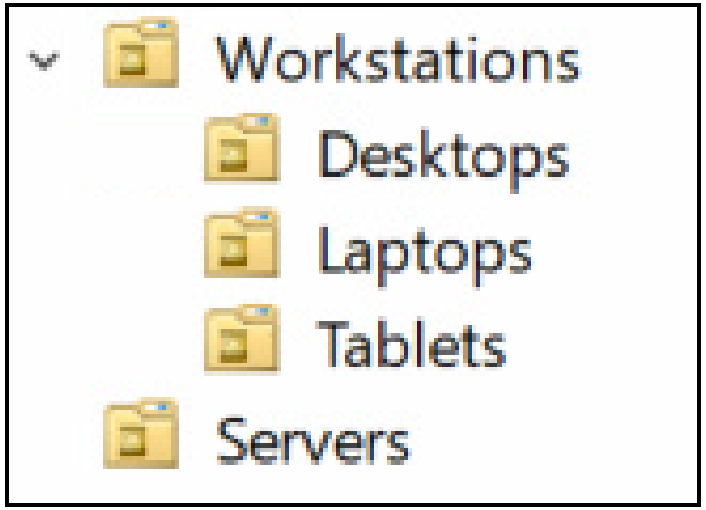
OU planning strategies: location-based

If your organization spans multiple locations, you can create top-level OUs to represent locations and place objects that belong in a location in the appropriate OU.



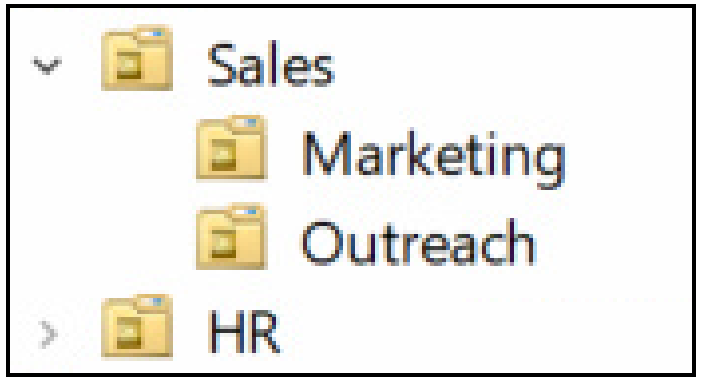
OU planning strategies: resource-based

- In simpler networks without multiple locations, you can create OUs based on resources.
- Create OUs based on the type of object that will be stored there.



OU planning strategies: organization-based

- Mirrors the organizational structure of the company.
  - Allows for delegation of administration and application of distinct Group Policies for objects in a department.
- Location agnostic.
  - Useful if users and their laptops frequently move from one location to another.

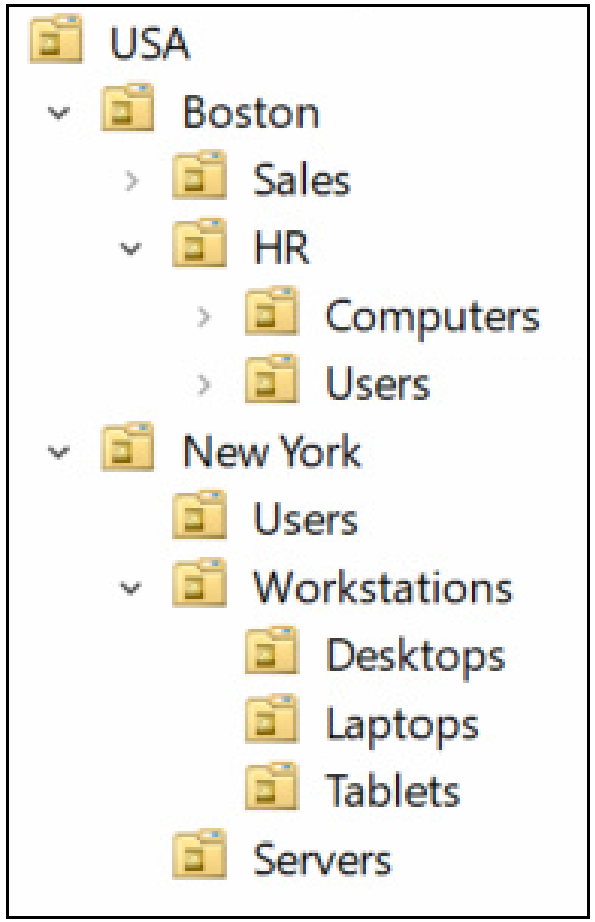


OU planning strategies: multi-tenancy based

- For organizations that provide cloud services to clients, referred to as tenants.
- Create top-level OUs to represent the different tenants.
  - Create lower-level OUs, depending on the administrative needs for tenants.
  - Each tenant could have a different OU structure, as appropriate.
- Allows you to delegate administration at the tenant level and to have distinct Group Policies for each tenant.

OU planning strategies: hybrid strategy

- In reality, most organizations use a combination of these strategies.
- For example:
  - Top-level OUs represent locations.
  - Second-level OUs represent departments in a location.
  - Third-level OUs represent types of objects.



Delegate administrative control

- Delegate administration by giving administrators permissions over an OU or an entire OU subtree.
  - Be default, child OUs inherit permissions from their parent OU.
  - Permissions inheritance can be blocked if needed.
- More granular delegation might only give an administrator permissions for certain object types, such as computers only. The administrator would have no permissions for other object types, such as users in the OU.
- Group policies are also inherited from parent containers.
  - Group policy inheritance can also be blocked at particular OUs.

Create OUs

To create and manage OUs, You can use:

- Graphical tools, like **AD Users and Computers.**
- Powershell cmdlets, such as **New-ADOrganizationalUnit.**
- Legacy command-line tools, such as **dsadd** OU.

Prevent accidental deletion

- Deleting an OU deletes the objects it contains.
  - There could be thousands of objects in an OU.
- In Windows Server 2008 R2, Microsoft introduced the **Prevent Accident Deletion** attribute:
  - It's enabled by default on new OUs.
  - It can also be enabled for objects, such as users or computers.

  Manage the attribute in the properties of the OU on the **Object** tab.

Manage permissions in Active Directory

- Typically, permissions are set on OUs. You can:
  - Give full control permission to all objects in the OU and child OUs.
  - Give limited permissions, such as read only for an OU.
  - Give permissions on an object class, such as users only or computers only, within the OU.
- Permissions are set on the **Security** tab in the properties of the OU.
- Advanced permissions can be applied to:
  - **This object only** (the OU itself, but not its contents).
  - **This object and all descendent objects**.
  - Specific types of objects, such as **descendent computer objects**.

Block inheritance

- If you don’t want the permissions set on an OU to be inherited by child OUs or certain objects, you can block inheritance.
  - This is useful for creating exceptions in the hierarchy.
- Manage inheritance in the **Advanced security** settings in the **properties** of an OU or object.

Security descriptors

Objects (and containers) have an associated security descriptor. It contains the following information:

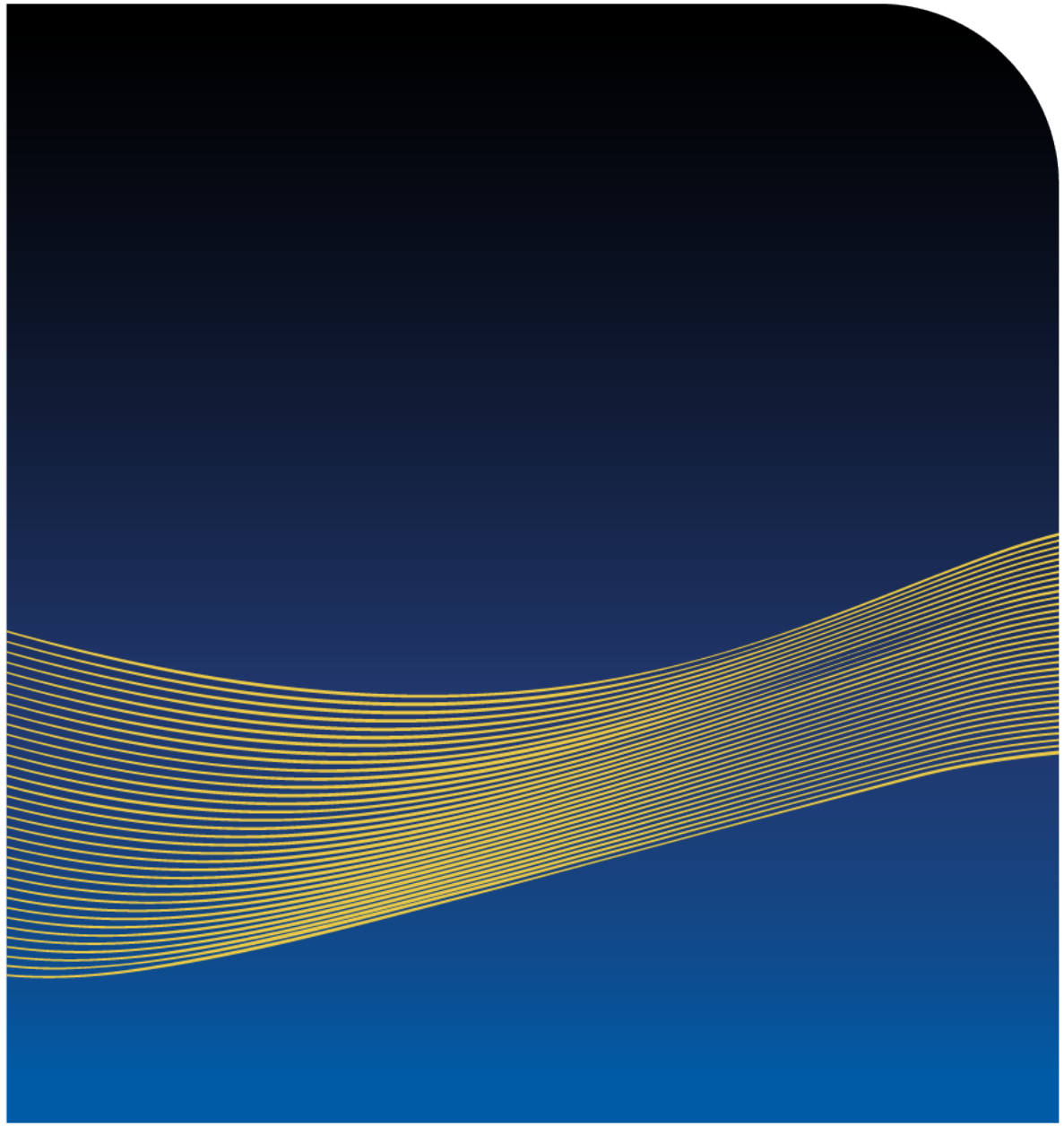
- The owner of the object.
- The SACL, which describes what type of auditing is set up, if any.
- The DACL, which describes the permissions set for the object.
- The DACL contains ACEs. Each ACE lists a security principal (such as a user, group, or computer) and the permissions given to that security principal.
- A flag indicating whether permissions are inherited from the parent container.

Delegation of Control Wizard

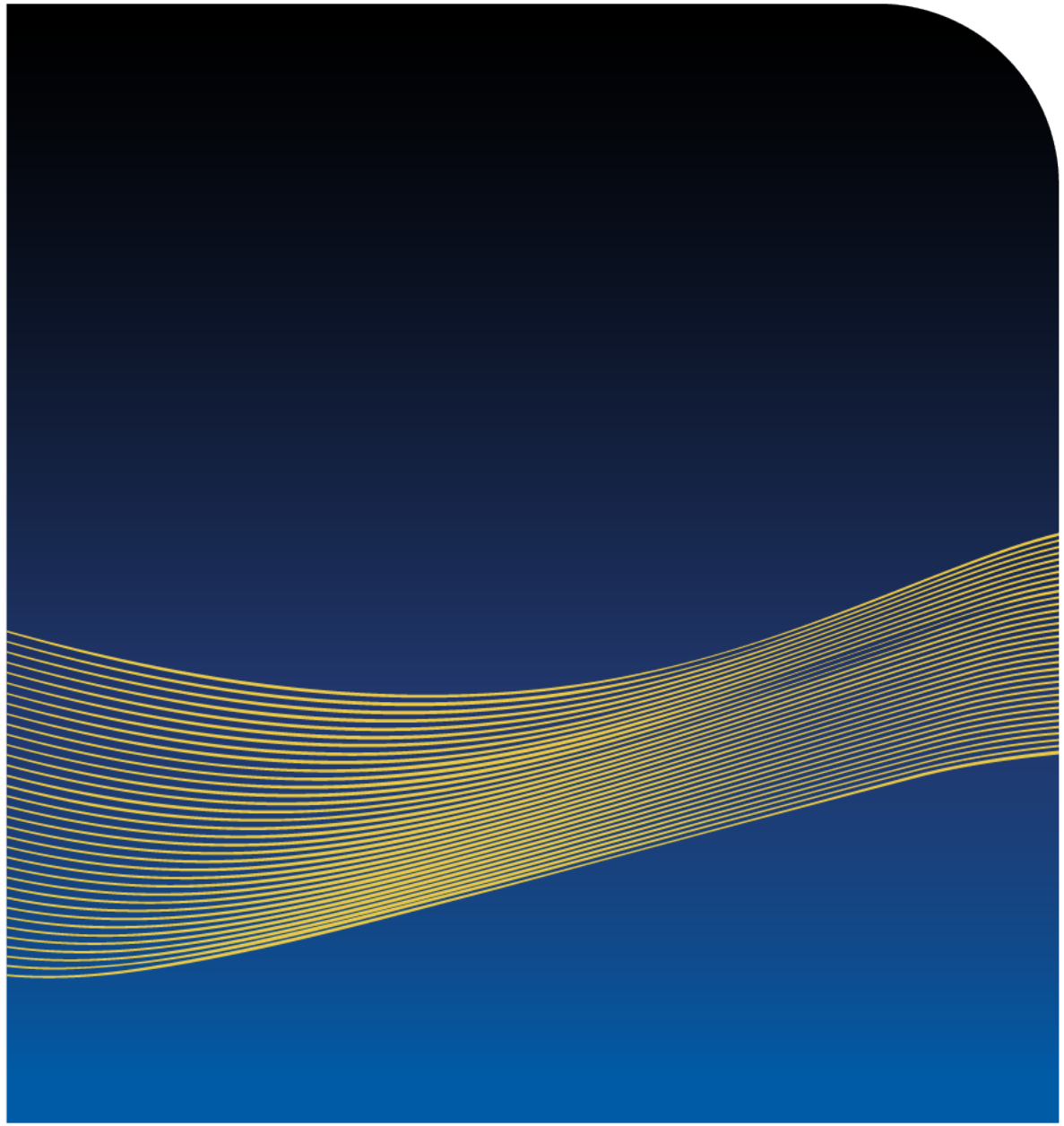
- Permissions in Active Directory are very flexible and granular.
  - The granularity means setting permissions can get complex.
- Microsoft provides the **Delegation of Control wizard** to simplify permissions assignment.
  - You can run the wizard for OUs, some default containers, domains, and sites.

Move objects in Active Directory

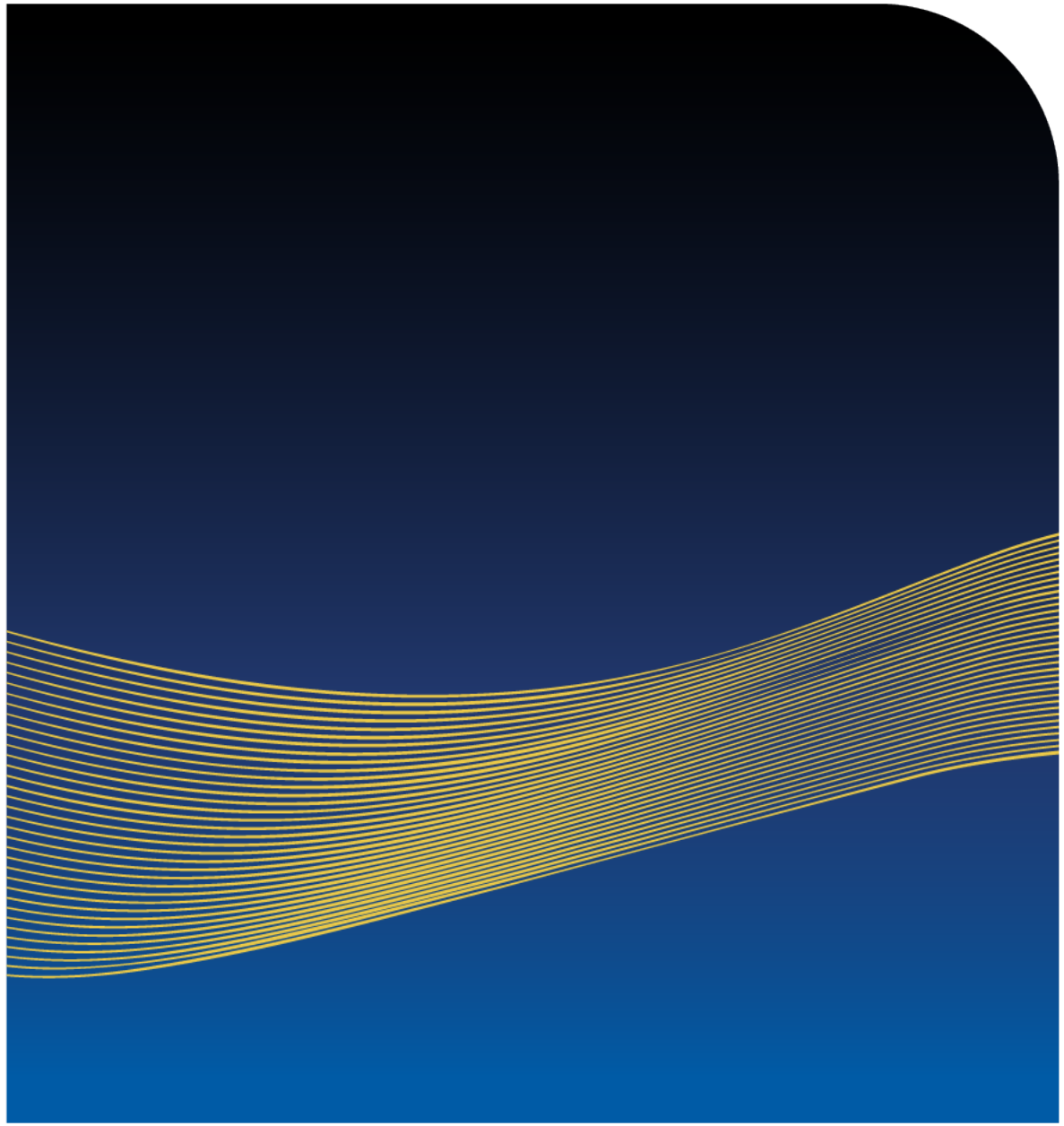
- Moving an object potentially changes what permissions are set on it and what group policies apply to it.
- Moving an object requires you to have:
  - **Delete** permissions for the source OU.
  - **Write** permissions for the naming attributes of the object.
  - **Create** permissions for the destination OU.
- In addition, the **Prevent Accidental Deletion** attribute must ***not*** be set on the object you want to move.



Demonstration: Delegate administrative permissions on an OU



Lab 3: Administer AD



Knowledge check

Knowledge check (1 of 3)

1. What is the unique requirement for a user's UPN (User Principal Name)?
2. Name two command-line methods of creating new user accounts.
3. If a user account is in DomainA, can it be added as a member to a **Global** group in DomainB?
4. If a user is made a manager of a group (in the **properties** of the group, on the **Managed By** tab), what does that allow the user to do?
5. When you join a computer to a domain, a computer object is created in Active Directory. What is the default location for new computer object? Can you change this default location?

Knowledge check (2 of 3)

1. What's the uniqueness requirement for a user's UPN (User Principal Name)?
   - It must be unique in the forest.
2. Name two command-line methods of creating new user accounts.
   - **DSAdd User** and the PowerShell cmdlet **New-ADUser**.
3. If a user account is in DomainA, can it be added as a member to a **Global** group in DomainB?
   - No. **Global** groups can only contain accounts from their own domain only.

Knowledge check (3 of 3)

4. If a user is made a manager of a group (in the properties of the group, on the **Managed By** tab), what does that allow the user to do?
    - The user appointed as a manager can modify the membership of the group.
5. When you join a computer to a domain, a computer object is created in Active Directory. What's the default location for new computer object? Can you change this default location?
    - The default location for new computer accounts is the **Computers** container. The default container can be changed by the **redircmp** command.

Thank you