Group Policy Management

Table Of Contents

Groups

Add-ADGroupMember

The Add-ADGroupMember cmdlet adds one or more users, groups, service accounts, or computers as new members of an Active Directory group.

The Identity parameter specifies the Active Directory group that receives the new members. You can identify a group by its distinguished name, GUID, security identifier, or Security Account Manager (SAM) account name. You can also specify group object variable, such as $, or pass a group object through the pipeline to the Identity parameter. For example, you can use the Get-ADGroup cmdlet to get a group object and then pass the object through the pipeline to the Add-ADGroupMember cmdlet.

The Members parameter specifies the new members to add to a group. You can identify a new member by its distinguished name, GUID, security identifier, or SAM account name. You can also specify user, computer, and group object variables, such as $. If you are specifying more than one new member, use a comma-separated list. You cannot pass user, computer, or group objects through the pipeline to this cmdlet. To add user, computer, or group objects to a group by using the pipeline, use the Add-ADPrincipalGroupMembership cmdlet.

```
# Add group member
Add-ADGroupMember -Identity [Group SamAccountName] -Members [SamaccountName]
```

Get-ADGroup

The Get-ADGroup cmdlet gets a group or performs a search to retrieve multiple groups from an Active Directory.

The Identity parameter specifies the Active Directory group to get. You can identify a group by its distinguished name (DN), GUID, security identifier (SID), Security Accounts Manager (SAM) account name, or canonical name. You can also specify group object variable, such as $.

To search for and retrieve more than one group, use the Filter or LDAPFilter parameters. The Filter parameter uses the PowerShell Expression Language to write query strings for Active Directory. PowerShell Expression Language syntax provides rich type conversion support for value types received by the Filter parameter. For more information about the Filter parameter syntax, type Get-Help about_ActiveDirectory_Filter. If you have existing Lightweight Directory Access Protocol (LDAP) query strings, you can use the LDAPFilter parameter.

This cmdlet gets a default set of group object properties. To get additional properties use the Properties parameter. For more information about the how to determine the properties for group objects, see the Properties parameter description.

```
# Get a sigle group
Get-ADGroup -Identity [Group SamAccountName]

# Filter for results
Get-ADGroup -Filter 'GroupCategory -eq "Security" -and GroupScope -ne
"DomainLocal"'

# View specific properties from a group and put into a table
Get-ADGroup -Identity [Group SamAccountName] -Properties [Value],[Value] | fl
[Header Value],[Header Value]
```

New-ADGroup

The New-ADGroup cmdlet creates an Active Directory group object. Many object properties are defined by setting cmdlet parameters. Properties that cannot be set by cmdlet parameters can be set using the OtherAttributes parameter.

The Name and GroupScope parameters specify the name and scope of the group and are required to create a new group. You can define the new group as a security or distribution group by setting the GroupType parameter. The Path parameter specifies the container or organizational unit (OU) for the group.

```
New-ADGroup -Name [Group Name Value] -SamAccountName [Group Sam Account Value] -
GroupCategory [Group Category Value] -GroupScope Global -DisplayName [Display Name
Value] -Path [OU Path] -Description [Description Value]
```

Remove-ADGroup

The Remove-ADGroup cmdlet removes an Active Directory group object. You can use this cmdlet to remove security and distribution groups.

The Identity parameter specifies the Active Directory group to remove. You can identify a group by its distinguished name, GUID, security identifier, Security Account Manager (SAM) account name, or canonical name. You can also set the Identity parameter to an object variable such as $, or you can pass an object through the pipeline to the Identity parameter. For example, you can use the Get-ADGroup cmdlet to retrieve a group object and then pass the object through the pipeline to the Remove-ADGroup cmdlet.

If the ADGroup is being identified by its distinguished name, the Partition parameter is automatically determined.

```
Remove-ADGroup -Identity [Group SamAccountValue]
```

Remove-ADGroupMember

The Remove-ADGroupMember cmdlet removes one or more users, groups, service accounts, or computers from an Active Directory group.

The Identity parameter specifies the Active Directory group that contains the members to remove. You can identify a group by its distinguished name, GUID, security identifier, or Security Account Manager (SAM) account name. You can also specify a group object variable, such as $, or pass a group object through the pipeline to the Identity parameter. For example, you can use the Get-ADGroup cmdlet to retrieve a group object and then pass the object through the pipeline to the Remove-ADGroupMember cmdlet.

The Members parameter specifies the users, computers and groups to remove from the group specified by the Identity parameter. You can identify a user, computer or group by its distinguished name, GUID, security identifier, or SAM account name. You can also specify user, computer, and group object variables, such as $. If you are specifying more than one new member, use a comma-separated list. You cannot pass user, computer, or group objects through the pipeline to this cmdlet. To remove user, computer, or group objects from a group by using the pipeline, use the Remove-ADPrincipalGroupMembership cmdlet.

```
Remove-ADGroupMember -Identity [Group SamAccountName] -Members [Member
SamAccountName]
```

Set-ADGroup

The Set-ADGroup cmdlet modifies the properties of an Active Directory group. You can modify commonly used property values by using the cmdlet parameters. Property values that are not associated with cmdlet parameters can be modified by using the Add, Replace, Clear, and Remove parameters.

The Identity parameter specifies the Active Directory group to modify. You can identify a group by its distinguished name, GUID, security identifier, or Security Account Manager (SAM) account name. You can also set the Identity parameter to an object variable such as $, or you can pass a group object through the pipeline to the Identity parameter. For example, you can use the Get-ADGroup cmdlet to get a group object and then pass the object through the pipeline to the Set-ADGroup cmdlet.

The Instance parameter provides a way to update a group object by applying the changes made to a copy of the object. When you set the Instance parameter to a copy of an Active Directory group object that has been modified, the Set-ADGroup cmdlet makes the same changes to the original group object. To get a copy of the object to modify, use the Get-ADGroup cmdlet. The Identity parameter is not allowed when you use the Instance parameter. For more information about the Instance parameter, see the Instance parameter description.

```
Set-ADGroup -Server [Server Address] -Identity [Group SamAccountName] -Description
[Description Value] -Passthru
```

Group Policy Objects

Get-GPUInheritance

The Get-GPInheritance cmdlet gets information about Group Policy inheritance for a specified domain or organizational unit (OU).

This information includes the following:

- A list of GPOs that are linked directly to the location (the GpoLinks property).
- A list of GPOs that are applied to the location when Group Policy is processed on a client (the InheritedGpoLinks property).
- Whether inheritance is blocked for the location (the GpoInheritanceBlocked property).

The InheritedGpoLinks property contains a list of the GPOs are applied to the OU or domain when Group Policy is processed on a client. The GPOs are listed according to the order of precedence with which they are applied. This list includes (in the following order):

- Inherited GPOs that are linked, enabled, and enforced at higher levels of the Group Policy hierarchy (for example, a site).
- GPOs that are linked and enabled directly at the specified location.
- If inheritance is not blocked for the specified location, inherited GPOs that are linked and enabled -- but not enforced -- at higher levels of the Group Policy hierarchy.

```
Get-GPInheritance -Target [OU Path]
```

Get-GPO

The Get-GPO cmdlet gets one Group Policy Object (GPO) or all the GPOs in a domain. You can specify a GPO by its display name or by its globally unique identifier (GUID) to get a single GPO, or you can get all the GPOs in the domain through the All parameter.

This cmdlet returns one or more objects that represent the requested GPOs. By default, properties of the requested GPOs are printed to the display; however, you can also pipe the output of the Get-GPO cmdlet to other Group Policy cmdlets.

```
Get-GPO -Name [GPO Name Value]
```

New-GPLink

The New-GPLink cmdlet links a GPO to a site, domain, or organizational unit (OU). By default, the link is enabled, which means that the settings of the GPO are applied at the level of the target Active Directory container according to the rules of inheritance and precedence when Group Policy is processed.

You can specify the GPO by either its display name or its GUID; or the GPO can be piped into the cmdlet. You specify the site, domain, or organizational unit (OU) to link to by its Lightweight Directory Access Protocol (LDAP) distinguished name. You can use other parameters to specify whether the link is enabled, whether the link is enforced, and the order in which it is applied at the site, domain, or OU.

```
# Create a GPO
New-GPLink -Name GPO_NAME -Target [OU Path]

# Create a GPO and link it
New-GPO -Name [GPO Name Value] | New-GPLink -Target [OU Path] -LinkEnabled Yes
```

New-GPO

The New-GPO cmdlet creates a GPO with a specified name. By default, the newly created GPO is not linked to a site, domain, or organizational unit (OU).

You can use this cmdlet to create a GPO that is based on a starter GPO by specifying the GUID or the display name of the Starter GPO, or by piping a StarterGpo object into the cmdlet.

The cmdlet returns a GPO object, which represents the created GPO that you can pipe to other Group Policy cmdlets.

```
New-GPO -Name [GPO Name Value]
```

Remove-GPLink

The Remove-GPLink cmdlet removes the link between a Group Policy Object (GPO) and a specified site, domain, or OU. This cmdlet does not delete the actual GPO or any other links between the specified GPO and other sites, domains, or OUs.

```
Remove-GPLink -Name [GPO Name Value] -Target [OU Path]
```

Remove-GPO

The Remove-GPO cmdlet removes the Group Policy Object (GPO) container and data from the directory service and the system volume folder (SysVol).

```
Remove-GPO -Name [GPO Name Value]
```

Set-GPInheritance

The Set-GPInheritance cmdlet blocks or unblocks inheritance for a specified domain or organizational unit (OU).

GPOs are applied according to the Group Policy hierarchy in the following order: local GPO, GPOs linked to the site, GPOs linked to the domain, GPOs linked to OUs. By default, an Active Directory container inherits settings from GPOs that are applied at the next higher level in the hierarchy. Blocking inheritance prevents the settings in GPOs that are linked to higher-level sites, domains, or organizational units from being automatically inherited by the specified domain or OU, unless the link for a GPO is enforced.

You use the Target parameter to specify the Lightweight Directory Access Protocol (LDAP) distinguished name of the domain or OU, and use the IsBlocked parameter to specify whether to block or unblock inheritance.

```
# Block/Unblock inheritance in a domain
Set-GPInheritance -Target [OU Path] -IsBlocked [Bool]
```

Set-GPLink

The Set-GPLink cmdlet sets the properties of a Group Policy Object (GPO) link.

You can set the following properties:

- Enabled. If the GPO link is enabled, the settings of the GPO are applied when Group Policy is processed for the site, domain or OU.
- Enforced. If the GPO link is enforced, it cannot be blocked at a lower-level (in the Group Policy processing hierarchy) container.

- Order. The order specifies the precedence that the settings of the GPO take over conflicting settings in other GPOs that are linked, and enabled, to the same site, domain, or OU.

```
# Enable a link between a GPO and OU
Set-GPLink -Name [GPO Link Name Value] -Target [OU Path] -LinkEnabled Yes
```

Group Policies

Policy Inheritance Order

1. Local Policy - Local machine. Rare policies are implemented here. Often disabled completely.
2. Site-Level Policy - Physical site. Change based on the computer's physical location.
3. Domain-Level Policy - Applies to every machine or user in the domain.
4. OU-Level Policy - Applies to a specific OU. The most common. Nested OUs always have priority.

Each new level overrides the previous, unless blocked or filtered, etc.

View Local Group Policy

```
gpedit.msc
```

Utils

gpresult

Displays the Resultant Set of Policy (RSoP) information for a remote user and computer. To use RSoP reporting for remotely targeted computers through the firewall, you must have firewall rules that enable inbound network traffic on the ports.

```
gpresult /r
```

Flag	Desc
`/r`	Short for Resultant Set of Policy (RSOP). Shows the actual results.
`/scope [user | computer]`	Shows only the user or computer security policies.
`/x`	Export results in XML format.
`/h`	Export results in HTML format.
`/s`	Specifies a remote computer.
`/u`	Username for remote computer.

Flag	Desc
`/p`	Password for remote computer.

gpupdate

Updates Group Policy settings.

```
gpupdate /force
```

Asynchronous Mode is the default mode for security updates. This means the update does not have to have fully finished before the user logs in and begins work.

Synchronous Mode will not show the desktop until all of the updates are applied.

Flag	Desc
`/force`	Forces the re-access and re-application of ALL security settings.
`/sync`	Switches the next foreground update to synchronous mode.
`/boot`	Restart after applying. Needed for software install/updates and logon scripts.
`/logoff`	Log out after update. Kind of useless because a normal logout triggers an update anyway.
`/wait <seconds>`	Wait while the gpupdate finishes, then return cursor to normal use.
`/target:[user | computer]`	Will re-access and re-apply only the user or computer pooicy updates. Cuts down on processing time.