

# Move- ADDirectoryServerOperation MasterRole

<https://learn.microsoft.com/en-us/powershell/module/activedirectory/move-addirectoryserveroperationmasterrole?view=windowsserver2022-ps>

## Description

The **Move-ADDirectoryServerOperationMasterRole** cmdlet moves one or more operation master roles to a directory server. You can move operation master roles to a directory server in a different domain if the credentials are the same in both domains.

The *Identity* parameter specifies the directory server that receives the roles. You can specify a directory server object by one of the following values:

- Name of the server object (name)
- The distinguished name of the NTDS Settings object
- The distinguished name of the server object that represents the directory server
- GUID (objectGUID) of server object under the configuration partition
- GUID (objectGUID) of NTDS settings object under the configuration partition

For Active Directory Lightweight Directory Services (AD LDS) instances the syntax for the server object name is `<computer-name>${instance-name}`. The following is an example of this syntax:

```
asia-w7-vm4$instance1
```

When you type this value in Windows PowerShell, you must use the backtick ( ` ) as an escape character for the dollar sign ( \$ ). Therefore, for this example, you would type the following:

```
asia-w7-vm4`$instance1
```

You can also set the parameter to a directory server object variable, such as

```
`${localDirectoryServerObject}
```

The **Move-ADDirectoryServerOperationMasterRole** cmdlet provides two options for moving operation master roles:

**Role transfer**, which involves transferring roles to be moved by running the cmdlet using the *Identity* parameter to specify the current role holder and the *OperationMasterRole* parameter to specify the roles for transfer. This is the recommended option.

Operation roles include PDCEmulator, RIDMaster, InfrastructureMaster, SchemaMaster, or DomainNamingMaster. To specify more than one role, use a comma-separated list.

**Role seizure**, which involves seizing roles you previously attempted to transfer by running the cmdlet a second time using the same parameters as the transfer operation, and adding the *Force* parameter. The *Force* parameter must be used as a switch to indicate that seizure, instead of transfer, of operation master roles is being performed. This operation still attempts graceful transfer first, then seizes if transfer is not possible.

Unlike using Ntdsutil.exe to move operation master roles, the **Move-ADDirectoryServerOperationMasterRole** cmdlet can be remotely executed from any domain joined computer where the Active Directory module for Windows PowerShell administration module is installed and available for use. This can make the process of moving roles simpler and easier to centrally administer as each of the two command operations required can be run remotely and do not have to be locally executed at each of the corresponding role holders involved in the movement of the roles, for instance, role transfer only allowed at the old role holder, role seizure only allowed at the new role holder.

## Examples

### Example 1: Move a PDC emulator to a domain controller

PowerShell 

```
PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity "USER01-DC1" -OperationMasterRole PDCEmulator
```

This command moves the primary domain controller (PDC) Emulator role to the domain controller USER01-DC1.

## Example 2: Move the PDC emulator and schema master roles to a domain controller

PowerShell 

```
PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity "USER02-DC2" -OperationMasterRole PDCEmulator,SchemaMaster
```

This command moves the PDC Emulator and schema master roles to the domain controller USER02-DC2.

## Example 3: Move the schema master FSMO owner to the AD LDS instance on a server

PowerShell 

```
PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity User03-DC`$instance1 -OperationMasterRole schemaMaster -Server User03-DC:50000
```

This command moves the schema master flexible single master operations (FSMO) owner to the AD LDS instance instance1 on the server User03-DC.

## Example 4: Seize specific roles for a specified user

PowerShell 

```
PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity USER04-DC1 -OperationMasterRole RIDMaster,InfrastructureMaster,DomainNamingMaster -Force
```

This command seizes the roles RID master, infrastructure master, and domain naming master.

# Example 5: Transfer roles to a specific domain controller

PowerShell [Copy](#)

```
PS C:\> $Server = Get-ADDomainController -Identity "TK5-CORP-DC-10.fabrikam.com"
PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity $Server -OperationMasterRole
SchemaMaster,DomainNamingMaster,PDCEmulator,RIDMaster,InfrastructureMaster
```

This command transfers the FSMO role to the specified domain controller. When using the fully qualified domain name (FQDN) to identify the domain controller, the **Get-ADDomainController** cmdlet must be used first as a preliminary step. There is a known issue where the **Move-ADDirectoryServerOperationMasterRole** cmdlet fails when an FQDN is specified directly as the value of the *Identity* parameter.

## Parameters

### -AuthType

Specifies the authentication method to use. The acceptable values for this parameter are:

- Negotiate or 0
- Basic or 1

The default authentication method is Negotiate.

A Secure Sockets Layer (SSL) connection is required for the Basic authentication method.

[Expand table](#)

Type:	ADAuthType
Accepted values:	Negotiate, Basic
Position:	Named
Default value:	None
Required:	False

Accept pipeline input:	False
Accept wildcard characters:	False

## -Confirm

Prompts you for confirmation before running the cmdlet.

Expand table

Type:	SwitchParameter
Aliases:	cf
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

## -Credential

Specifies the user account credentials to use to perform this task. The default credentials are the credentials of the currently logged on user unless the cmdlet is run from an Active Directory module for Windows PowerShell provider drive. If the cmdlet is run from such a provider drive, the account associated with the drive is the default.

To specify this parameter, you can type a user name, such as User1 or Domain01\User01 or you can specify a **PSCredential** object. If you specify a user name for this parameter, the cmdlet prompts for a password.

You can also create a **PSCredential** object by using a script or by using the **Get-Credential** cmdlet. You can then set the *Credential* parameter to the **PSCredential** object.

If the acting credentials do not have directory-level permission to perform the task, Active Directory module for Windows PowerShell returns a terminating error.

## Expand table

Type:	PSCredential
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

## -Force

Indicates that the cmdlet is used for seize operations on domain controllers with the flexible single master operations (FSMO) role.

## Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

## -Identity

Specifies an Active Directory server object by providing one of the following values. The identifier in parentheses is the Lightweight Directory Access Protocol (LDAP) display name for the attribute.

- Name of the server object (name)

For Active Directory Lightweight Directory Services (AD LDS) instances the syntax is of a name is `<computer-name>${<instance-name>}`.

Note: When you type this value in Windows PowerShell, you must use the backtick ( ` ) as an escape character for the dollar sign ( \$ ). For instance, *asia-w7-vm4`\$instance1*.

For other Active Directory instances, use the value of the name property.

- The distinguished name of the NTDS Settings object
- The distinguished name of the server object that represents the directory server
- GUID (objectGUID) of server object under the configuration partition
- GUID (objectGUID) of NTDS settings object under the configuration partition

The cmdlet searches the default naming context or partition to find the object. If two or more objects are found, the cmdlet returns a non-terminating error.

This parameter can also get this object through the pipeline or you can set this parameter to an object instance.

Expand table

Type:	ADDirectoryServer
Position:	0
Default value:	None
Required:	True
Accept pipeline input:	True
Accept wildcard characters:	False

## -OperationMasterRole

Specifies one or more operation master roles to move to the specified directory server in Active Directory Domain Services. The acceptable values for this parameter are:

- PDCEmulator or 0
- RIDMaster or 1
- InfrastructureMaster or 2
- SchemaMaster or 3
- DomainNamingMaster or 4

To specify multiple operation master roles, use a comma-separated list.

#### Expand table

Type:	ADOperationMasterRole[]
Accepted values:	PDCEmulator, RIDMaster, InfrastructureMaster, SchemaMaster, DomainNamingMaster
Position:	1
Default value:	None
Required:	True
Accept pipeline input:	False
Accept wildcard characters:	False

## -PassThru

Returns an object representing the item with which you are working. By default, this cmdlet does not generate any output.

#### Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

## -Server

Specifies the Active Directory Domain Services instance to connect to, by providing one of the following values for a corresponding domain name or directory server. The service may be any of

the following: Active Directory Lightweight Domain Services, Active Directory Domain Services or Active Directory snapshot instance.

Specify the Active Directory Domain Services instance in one of the following ways:

Domain name values:

- Fully qualified domain name
- NetBIOS name

Directory server values:

- Fully qualified directory server name
- NetBIOS name
- Fully qualified directory server name and port

The default value for this parameter is determined by one of the following methods in the order that they are listed:

- By using the *Server* value from objects passed through the pipeline
- By using the server information associated with the Active Directory Domain Services Windows PowerShell provider drive, when the cmdlet runs in that drive
- By using the domain of the computer running Windows PowerShell

Expand table

Type:	String
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

## -WhatIf

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Expand table

Type:	SwitchParameter
Aliases:	wi
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

# Inputs

## **[ADDirectoryServer](#)**

A directory server object is received by the *Identity* parameter.

# Outputs

## **None or Microsoft.ActiveDirectory.Management.ADDirectoryServer**

Returns the modified directory server object when the *PassThru* parameter is specified. By default, this cmdlet does not generate any output.

# Notes

- This cmdlet does not work with an Active Directory snapshot.
- This cmdlet does not work with a read-only domain controller.

# Related Links

- [Move-ADDirectoryServer](#)

- [AD DS Administration Cmdlets in Windows PowerShell](#)
- 

Revision #1

Created 5 January 2024 05:29:33 by ColtM

Updated 7 August 2024 23:24:38 by ColtM