Get-ADUser

Module: addsadministration

Gets one or more Active Directory users.

In this article

PowerShell Copy

```
Get-ADUser
   [-AuthType <ADAuthType>]
   [-Credential <PSCredential>]
   -Filter <String>
   [-Properties <String[]>]
   [-ResultPageSize <Int32>]
   [-ResultSetSize <Int32>]
   [-SearchBase <String>]
   [-SearchScope <ADSearchScope>]
   [-Server <String>]
   [<CommonParameters>]
```

PowerShell Copy

```
Get-ADUser
   [-AuthType <ADAuthType>]
   [-Credential <PSCredential>]
   [-Identity] <ADUser>
   [-Partition <String>]
   [-Properties <String[]>]
   [-Server <String>]
   [<CommonParameters>]
```

PowerShell Copy

```
Get-ADUser
   [-AuthType <ADAuthType>]
   [-Credential <PSCredential>]
   -LDAPFilter <String>
   [-Properties <String[]>]
   [-ResultPageSize <Int32>]
   [-ResultSetSize <Int32>]
   [-SearchBase <String>]
   [-SearchScope <ADSearchScope>]
   [-Server <String>]
   [<CommonParameters>]
```

Description

The **Get-ADUser** cmdlet gets a specified user object or performs a search to get multiple user objects.

The *Identity* parameter specifies the Active Directory user to get. You can identify a user by its distinguished name (DN), GUID, security identifier (SID), Security Account Manager (SAM) account name or name. You can also set the parameter to a user object variable, such as `$<localUserObject>` or pass a user object through the pipeline to the *Identity* parameter.

To search for and retrieve more than one user, 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 retrieves a default set of user object properties. To retrieve additional properties use the **Properties** parameter. For more information about the how to determine the properties for user objects, see the **Properties** parameter description.

Examples

Example 1: Get all of the users in a container

PowerShell Copy

```
PS C:\> Get-ADUser -Filter * -SearchBase "OU=Finance,OU=UserAccounts,DC=FABRIKAM,DC=COM"
```

This command gets all users in the container OU=Finance,OU=UserAccounts,DC=FABRIKAM,DC=COM.

Example 2: Get a filtered list of users

PowerShell Copy

```
PS C:\> Get-ADUser -Filter 'Name -like "*SvcAccount"' | Format-Table Name,SamAccountName -A
Name             SamAccountName
----             --------------
SQL01 SvcAccount SQL01
SQL02 SvcAccount SQL02
IIS01 SvcAccount IIS01
```

This command gets all users that have a name that ends with SvcAccount.

Example 3: Get all of the properties for a specified user

PowerShell Copy

```
PS C:\> Get-ADUser -Identity ChewDavid -Properties *
Surname           : David
Name              : Chew David
UserPrincipalName :
GivenName         : David
Enabled           : False
SamAccountName    : ChewDavid
ObjectClass       : user
SID               : S-1-5-21-2889043008-4136710315-2444824263-3544
ObjectGUID        : e1418d64-096c-4cb0-b903-ebb66562d99d
DistinguishedName : CN=Chew
David,OU=NorthAmerica,OU=Sales,OU=UserAccounts,DC=FABRIKAM,DC=COM
```

This command gets all of the properties of the user with the SAM account name ChewDavid.

Example 4: Get a specified user

PowerShell Copy

```
PS C:\> Get-ADUser -Filter {Name -eq "ChewDavid"} -SearchBase "DC=AppNC" -Properties "mail"
-Server lds.Fabrikam.com:50000
```

This command gets the user with name ChewDavid in the Active Directory Lightweight Directory Services (AD LDS) instance.

Required Parameters

`-Filter`

Specifies a query string that retrieves Active Directory objects. This string uses the PowerShell Expression Language syntax. The PowerShell Expression Language syntax provides rich type-conversion support for value types received by the *Filter* parameter. The syntax uses an in-order representation, which means that the operator is placed between the operand and the value. For more information about the *Filter* parameter, type `Get-Help about_ActiveDirectory_Filter`.

Syntax:

The following syntax uses Backus-Naur form to show how to use the PowerShell Expression Language for this parameter.

<filter> ::= "{" <FilterComponentList> "}"

<FilterComponentList> ::= <FilterComponent> | <FilterComponent> <JoinOperator> <FilterComponent> | <NotOperator> <FilterComponent>

<FilterComponent> ::= <attr> <FilterOperator> <value> | "(" <FilterComponent> ")"

<FilterOperator> ::= "-eq" | "-le" | "-ge" | "-ne" | "-lt" | "-gt"| "-approx" | "-bor" | "-band" | "-recursivematch" | "-like" | "-notlike"

<JoinOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not"

<attr> ::= <PropertyName> | <LDAPDisplayName of the attribute>

<value>::= <compare this value with an <attr> by using the specified <FilterOperator>>

For a list of supported types for <value>, type `Get-Help about_ActiveDirectory_ObjectModel`.

Note: PowerShell wildcards other than , *such as ?, are not supported by the *Filter* syntax.

Note: To query using LDAP query strings, use the *LDAPFilter* parameter.

Type:
String

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

-Identity

Specifies an Active Directory user object by providing one of the following property values. The identifier in parentheses is the LDAP display name for the attribute. The acceptable values for this parameter are:

- A distinguished name
- A GUID (objectGUID)
- A security identifier (objectSid)
- A SAM account name (sAMAccountName)

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.

Type:
ADUser

Position:
0

Default value:
None

Accept pipeline input:
True (ByValue)

Accept wildcard characters:
False

-LDAPFilter

Specifies an LDAP query string that is used to filter Active Directory objects. You can use this parameter to run your existing LDAP queries. The *Filter* parameter syntax supports the same functionality as the LDAP syntax. For more information, see the *Filter* parameter description or type `Get-Help about_ActiveDirectory_Filter`.

Type:
String

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

Optional 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.

Type:
ADAuthType

Accepted values:
Negotiate, Basic

Position:
Named

Default value:
None

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 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 PowerShell returns a terminating error.

Type:
PSCredential

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

-Partition

The default authentication method is Negotiate.

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

Type:
String

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

-Properties

Specifies the properties of the output object to retrieve from the server. Use this parameter to retrieve properties that are not included in the default set.

Specify properties for this parameter as a comma-separated list of names. To display all of the attributes that are set on the object, specify * (asterisk).

To specify an individual extended property, use the name of the property. For properties that are not default or extended properties, you must specify the LDAP display name of the attribute.

To retrieve properties and display them for an object, you can use the Get-* cmdlet associated with the object and pass the output to the **Get-Member** cmdlet.

Type:
String[]

Aliases:
Property

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

-ResultPageSize

Specifies the number of objects to include in one page for an Active Directory Domain Services query.

The default is 256 objects per page.

Type:
Int32

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

-ResultSetSize

Specifies the maximum number of objects to return for an Active Directory Domain Services query. If you want to receive all of the objects, set this parameter to $Null (null value). You can use Ctrl+C to stop the query and return of objects.

The default is $Null.

Type:
Int32

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

-SearchBase

Specifies an Active Directory path to search under.

When you run a cmdlet from an Active Directory provider drive, the default value of this parameter is the current path of the drive.

When you run a cmdlet outside of an Active Directory provider drive against an AD DS target, the default value of this parameter is the default naming context of the target domain.

When you run a cmdlet outside of an Active Directory provider drive against an AD LDS target, the default value is the default naming context of the target LDS instance if one has been specified by setting the **msDS-defaultNamingContext** property of the Active Directory directory service agent (DSA) object (**nTDSDSA**) for the AD LDS instance. If no default naming context has been specified for the target AD LDS instance, then this parameter has no default value.

When the value of the *SearchBase* parameter is set to an empty string and you are connected to a GC port, all partitions will be searched. If the value of the *SearchBase* parameter is set to an empty string and you are not connected to a GC port, an error will be thrown.

Type:
String

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

`-SearchScope`

Specifies the scope of an Active Directory search. The acceptable values for this parameter are:

- Base or 0
- OneLevel or 1
- Subtree or 2

A Base query searches only the current path or object. A OneLevel query searches the immediate children of that path or object. A Subtree query searches the current path or object and all children of that path or object.

Type:
ADSearchScope

Accepted values:
Base, OneLevel, Subtree

Position:
Named

Default value:
None

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.

Domain name values:

- Fully qualified domain name (FQDN)
- NetBIOS name

Directory server values:

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

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

- By using *Server* value from objects passed through the pipeline.
- By using the server information associated with the Active Directory PowerShell provider drive, when running under that drive.
- By using the domain of the computer running PowerShell.

Type:
String

Position:

Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

Inputs

None or Microsoft.ActiveDirectory.Management.ADUser

A user object is received by the *Identity* parameter.

Outputs

Microsoft.ActiveDirectory.Management.ADUser

Returns one or more user objects.

This cmdlet returns a default set of **ADUser** property values. To retrieve additional **ADUser** properties, use the *Properties* parameter.

To get a list of the default set of properties of an **ADUser** object, use the following command:

`Get-ADUser` <user> `| Get-Member`

To get a list of the most commonly used properties of an ADUser object, use the following command:

`Get-ADUser` <user> `-Properties Extended | Get-Member`

To get a list of all the properties of an **ADUser** object, use the following command:

`Get-ADUser` <user> `-Properties * | Get-Member`

Notes

- This cmdlet does not work with an Active Directory snapshot.

Related Links

- New-ADUser
- Remove-ADUser
- Set-ADUser