

Get-ADUser

By Archiveddocs

Archived: 2026-04-06 01:51:38 UTC

Gets one or more Active Directory users.

```
Get-ADUser -Filter <string> [-ResultPageSize <int>] [-ResultSetSize <System.Nullable[System.Int32]>] [-SearchBase
```

- *Filter*
- *ResultPageSize*
- *ResultSetSize*
- *SearchBase*
- *SearchScope*
- *SearchScope*
- *AuthType*
- *Credential*
- *Partition*
- *Properties*
- *Server*

```
Get-ADUser [-Identity] [-AuthType { | }] [-Credential ] [-Partition ] [-Properties <string[]>] [-Server ] []
```

- *Identity*
- *AuthType*
- *Credential*
- *Partition*
- *Properties*
- *Server*

```
Get-ADUser -LDAPFilter [-ResultPageSize ] [-ResultSetSize <System.Nullable[System.Int32]>] [-SearchBase ] [-SearchScope { | | }] [-AuthType { | }] [-Credential ] [-Partition ] [-Properties <string[]>] [-
```

Server] []

- *LDAPFilter*
- *ResultPageSize*
- *ResultSetSize*
- *SearchBase*
- *SearchScope*
- *AuthType*
- *Credential*
- *Partition*
- *Properties*
- *Server*

The Get-ADUser cmdlet gets a user object or performs a search to retrieve 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 Accounts 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, see [about_ActiveDirectory_Filter](#). If you have existing 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.

Specifies the authentication method to use. Possible values for this parameter include:

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.

The following example shows how to set this parameter to Basic.

`-AuthType Basic`

The following lists the acceptable values for this parameter:

- Negotiate**
- Basic**

Default Value: *Microsoft.ActiveDirectory.Management.AuthType.Negotiate*

Data Type: *ADAuthType*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>false</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>

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. The following example shows how to create credentials.
`$AdminCredentials = Get-Credential "Domain01\User01"`

The following shows how to set the Credential parameter to these credentials.

`-Credential $AdminCredentials`

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

Default Value: ******

Data Type: *PSCredential*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>false</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>

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, see [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>, see [about_ActiveDirectory_ObjectModel](#).

Examples:

The following examples show how to use this syntax with Active Directory cmdlets.

To get all objects of the type specified by the cmdlet, use the asterisk wildcard:

All user objects:

```
Get-ADUser -Filter *
```

-or-

All computer objects:

```
Get-ADComputer -Filter *
```

To get all user objects that have an e-mail message attribute, use one of the following commands:

```
Get-ADUser -Filter {EmailAddress -like "*"}
```

```
Get-ADUser -Filter {mail -like "*"}
```

-or-

```
Get-ADObject -Filter {(mail -like "*") -and (ObjectClass -eq "user")}
```

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

To get all users objects that have surname of Smith and that have an e-mail attribute, use one of the following commands:

```
Get-ADUser -filter {(EmailAddress -like "*") -and (Surname -eq "smith")}
```

-or-

Get-ADUser -filter {(mail -eq "*") -and (sn -eq "Smith")}

To get all user objects who have not logged on since January 1, 2007, use the following commands:

\$logonDate = New-Object System.DateTime(2007, 1, 1)

Get-ADUser -filter { lastLogon -le \$logonDate }

To get all groups that have a group category of Security and a group scope of Global, use one of the following commands:

Get-ADGroup -filter {GroupCategory -eq "Security" -and GroupScope -eq "Global"}

-or-

Get-ADGroup -filter {GroupType -band 0x80000000}

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

Default Value: **

Data Type: *string*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>

Name	Value	PSMAML Attribute
Variable Length?	<i>false</i>	<i>variableLength</i>

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.

Distinguished Name

Example: CN=SaraDavis,CN=Europe,CN=Users,DC=corp,DC=contoso,DC=com

GUID (objectGUID)

Example: 599c3d2e-f72d-4d20-8a88-030d99495f20

Security Identifier (objectSid)

Example: S-1-5-21-3165297888-301567370-576410423-1103

SAM account name (sAMAccountName)

Example: saradavis

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.

This example shows how to set the parameter to a distinguished name.

`-Identity "CN=SaraDavis,CN=Europe,CN=Users,DC=corp,DC=contoso,DC=com"`

This example shows how to set this parameter to a user object instance named "userInstance".

`-Identity $userInstance`

Default Value: **

Data Type: *ADUser*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>

Name	Value	PSMAML Attribute
Accept Pipeline Input?	<i>true (ByValue)</i>	<i>pipelineInput</i>
Position?	<i>1</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>

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 and the about_ActiveDirectory_Filter.

The following example shows how to set this parameter to search for all objects in the organizational unit specified by the SearchBase parameter with a name beginning with "sara".

```
-LDAPFilter "(name=sara*)" -SearchScope Subtree -SearchBase "DC=NA,DC=fabrikam,DC=com"
```

Default Value: **

Data Type: *string*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>

Name	Value	PSMAML Attribute
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>

Specifies the distinguished name of an Active Directory partition. The distinguished name must be one of the naming contexts on the current directory server. The cmdlet searches this partition to find the object defined by the Identity parameter.

The following two examples show how to specify a value for this parameter.

-Partition "CN=Configuration,DC=EUROPE,DC=TEST,DC=CONTOSO,DC=COM"

-Partition "CN=Schema,CN=Configuration,DC=EUROPE,DC=TEST,DC=CONTOSO,DC=COM"

In many cases, a default value will be used for the Partition parameter if no value is specified. The rules for determining the default value are given below. Note that rules listed first are evaluated first and once a default value can be determined, no further rules will be evaluated.

In AD DS environments, a default value for Partition will be set in the following cases: - If the Identity parameter is set to a distinguished name, the default value of Partition is automatically generated from this distinguished name.

- If running cmdlets from an Active Directory provider drive, the default value of Partition is automatically generated from the current path in the drive.

- If none of the previous cases apply, the default value of Partition will be set to the default partition or naming context of the target domain.

In AD LDS environments, a default value for Partition will be set in the following cases:

- If the Identity parameter is set to a distinguished name, the default value of Partition is automatically generated from this distinguished name.

- If running cmdlets from an Active Directory provider drive, the default value of Partition is automatically generated from the current path in the drive.

- If the target AD LDS instance has a default naming context, the default value of Partition will be set to the

default naming context. To specify a default naming context for an AD LDS environment, set the msDS-defaultNamingContext property of the Active Directory directory service agent (DSA) object (nTDSDSA) for the AD LDS instance.

- If none of the previous cases apply, the Partition parameter will not take any default value.

Default Value: **

Data Type: *string*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>false</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>

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. The following examples show how to retrieve properties for a group where the Administrator's group is used as the sample group object.

Get-ADGroup -Identity Administrators | Get-Member

To retrieve and display the list of all the properties for an ADGroup object, use the following command:

Get-ADGroup -Identity Administrators -Properties * | Get-Member

The following examples show how to use the Properties parameter to retrieve individual properties as well as the default, extended or complete set of properties.

To retrieve the extended properties "OfficePhone" and "Organization" and the default properties of an ADUser object named "SaraDavis", use the following command:

GetADUser -Identity SaraDavis -Properties OfficePhone,Organization

To retrieve the properties with LDAP display names of "otherTelephone" and "otherMobile", in addition to the default properties for the same user, use the following command:

GetADUser -Identity SaraDavis -Properties otherTelephone, otherMobile | Get-Member

Default Value: **

Data Type: *string[]*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>false</i>	<i>required</i>
Variable Length?	<i>true</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>true</i>	<i>variableLength</i>

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

The default is 256 objects per page.

The following example shows how to set this parameter.

`-ResultPageSize 500`

Default Value: 256

Data Type: *int*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>false</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>

Name	Value	PSMAML Attribute
Variable Length?	<i>false</i>	<i>variableLength</i>

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.

The following example shows how to set this parameter so that you receive all of the returned objects.

-ResultSetSize \$null

Default Value: **

Data Type: *System.Nullable[System.Int32]*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>false</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>

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.

The following example shows how to set this parameter to search under an OU.

```
-SearchBase "ou=mfg,dc=noam,dc=corp,dc=contoso,dc=com"
```

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.

The following example shows how to set this parameter to an empty string. -SearchBase ""

Default Value: **

Data Type: *string*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>false</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>

Specifies the scope of an Active Directory search. Possible 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.

The following example shows how to set this parameter to a subtree search.

`-SearchScope Subtree`

The following lists the acceptable values for this parameter:

- Base**
- OneLevel**
- Subtree**

Default Value: *Subtree*

Data Type: *ADSearchScope*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>false</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>

Name	Value	PSMAML Attribute
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>

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

Examples: corp.contoso.com

NetBIOS name

Example: CORP

Directory server values:

Fully qualified directory server name

Example: corp-DC12.corp.contoso.com

NetBIOS name

Example: corp-DC12

Fully qualified directory server name and port

Example: corp-DC12.corp.contoso.com:3268

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.

The following example shows how to specify a full qualified domain name as the parameter value.

-Server "corp.contoso.com"

Default Value: **

Data Type: *string*

Attributes

Name	Value	PSMAML Attribute
Required?	<i>false</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>
Accept wildcard characters?	<i>false</i>	<i>globbing</i>
Accept Pipeline Input?	<i>false</i>	<i>pipelineInput</i>
Position?	<i>named</i>	<i>position</i>

Value Attributes

Name	Value	PSMAML Attribute
Required?	<i>true</i>	<i>required</i>
Variable Length?	<i>false</i>	<i>variableLength</i>

A user object is received by the Identity parameter.

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

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

Command Prompt: C:\PS>

```
Get-ADUser -Filter * -SearchBase "OU=Finance,OU=UserAccounts,DC=FABRIKAM,DC=COM"
```

Get all users under the container 'OU=Finance,OU=UserAccounts,DC=FABRIKAM,DC=COM'.

Command Prompt: C:\PS>

```
Get-ADUser -Filter 'Name -like "*SvcAccount"' | FT Name,SamAccountName -A
```

Name SamAccountName

```
----  
SQL01 SvcAccount SQL01  
SQL02 SvcAccount SQL02  
IIS01 SvcAccount IIS01
```

Description

```
-----
```

Get all users that have a name that ends with 'SvcAccount'.

Command Prompt: C:\PS>

```
Get-ADUser GlenJohn -Properties *
```

```
Surname : John  
Name : Glen John  
UserPrincipalName :  
GivenName : Glen  
Enabled : False  
SamAccountName : GlenJohn  
ObjectClass : user  
SID : S-1-5-21-2889043008-4136710315-2444824263-3544
```

ObjectGUID : e1418d64-096c-4cb0-b903-ebb66562d99d

DistinguishedName : CN=Glen

John,OU=NorthAmerica,OU=Sales,OU=UserAccounts,DC=FABRIKAM,DC=COM

Description

Get all properties of the user with samAccountName 'GlenJohn'.

Command Prompt: *C:\PS>*

```
Get-ADUser -Filter {Name -eq "GlenJohn"} -SearchBase "DC=AppNC" -Properties mail -Server lds.Fabrikam.com:5000
```

Get the user with name 'GlenJohn' on the AD LDS instance.

[New-ADUser](#)

[Set-ADUser](#)

[Remove-ADUser](#)

[Online version:](#)

Source: <https://technet.microsoft.com/library/ee617241.aspx>