Get-ADComputer

Reference

Feedback

Module: ActiveDirectory

In this article

Gets one or more Active Directory computers.

Syntax

PowerShell Copy

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

PowerShell Copy

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

PowerShell Copy

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

Description

The `Get-ADComputer` cmdlet gets a computer or performs a search to retrieve multiple computers.

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

To search for and retrieve more than one computer, 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 computer object properties. To retrieve additional properties use the **Properties** parameter. For more information about the how to determine the properties for computer objects, see the **Properties** parameter description.

Examples

Example 1: Get specific computer that shows all properties

PowerShell Copy

```
Get-ADComputer -Identity "User01-SRV1" -Properties *

AccountExpirationDate              :
accountExpires                     : 9223372036854775807
AccountLockoutTime                 :
AccountNotDelegated                : False
AllowReversiblePasswordEncryption  : False
BadLogonCount                      :
CannotChangePassword               : False
CanonicalName                      : fabrikam.com/Computers/User01-srv1
Certificates                       : {}
CN                                 : User01-srv1
codePage                           : 0
countryCode                        : 0
Created                            : 3/16/2009 4:15:00 PM
createTimeStamp                    : 3/16/2009 4:15:00 PM
Deleted                            :
Description                        : DisplayName                          :
DistinguishedName                  : CN= User01-srv1,CN=Computers,DC=fabrikam,DC
DNSHostName                        : User01-srv1
DoesNotRequirePreAuth              : False
dSCorePropagationData              : {3/16/2009 4:21:51 PM, 12/31/1600 4:00:01 F
Enabled                            : True
HomedirRequired                    : False
HomePage                           :
instanceType                       : 0
IPv4Address                        :
IPv6Address                        :
isCriticalSystemObject             : False
isDeleted                          :
LastBadPasswordAttempt             :
LastKnownParent                    :
LastLogonDate                      :
localPolicyFlags                   : 0
Location                           : NA/HQ/Building A
LockedOut                          : False
ManagedBy                          : CN=SQL Administrator 01,OU=UserAccounts,OU=
MemberOf                           : {}
```

```
MNSLogonAccount                     : False
Modified                            : 3/16/2009 4:23:01 PM
modifyTimeStamp                     : 3/16/2009 4:23:01 PM
msDS-User-Account-Control-Computed  : 0
Name                                : User01-srv1
nTSecurityDescriptor                : System.DirectoryServices.ActiveDirectorySec
ObjectCategory                      : CN=Computer,CN=Schema,CN=Configuration,DC=f
ObjectClass                         : computer
ObjectGUID                          : 828306a3-8ccd-410e-9537-e6616662c0b0
objectSid                           : S-1-5-21-41432690-3719764436-1984117282-113
OperatingSystem                     :
OperatingSystemHotfix               :
OperatingSystemServicePack          :
OperatingSystemVersion              :
PasswordExpired                     : False
PasswordLastSet                     :
PasswordNeverExpires                : False
PasswordNotRequired                 : False
PrimaryGroup                        : CN=Domain Computers,CN=Users,DC=fabrikam,DC
primaryGroupID                      : 515
ProtectedFromAccidentalDeletion     : False
pwdLastSet                          : 0
SamAccountName                      : User01-srv1$
sAMAccountType                      : 805306369
sDRightsEffective                   : 0
ServiceAccount                      : {}
servicePrincipalName                : {MSOLAPSVC.3/User01-SRV1.fabrikam.com:analy
ServicePrincipalNames               : {MSOLAPSVC.3/User01-SRV1.fabrikam.com:analy
SID                                 : S-1-5-21-41432690-3719764436-1984117282-113
SIDHistory                          : {}
TrustedForDelegation                : False
TrustedToAuthForDelegation          : False
UseDESKeyOnly                       : False
userAccountControl                  : 4096
userCertificate                     : {}
UserPrincipalName                   :
uSNChanged                          : 36024
uSNCreated                          : 35966
whenChanged                         : 3/16/2009 4:23:01 PM
whenCreated                         : 3/16/2009 4:15:00 PM
```

This command gets a specific computer showing all the properties.

Example 2: Get all computers with a name starting with a particular string

PowerShell Copy

```
Get-ADComputer -Filter 'Name -like "User01*"' -Properties IPv4Address |
    Format-Table Name, DNSHostName, IPv4Address -AutoSize

name        dnshostname             ipv4address
----        -----------             -----------
User01-SRV1 User01-SRV1.User01.com 10.194.99.181
User01-SRV2 User01-SRV2.User01.com 10.194.100.3
```

This command gets all the computers with a name starting with a particular string and shows the name, DNS hostname, and IPv4 address.

Example 3: Gets all computers that have changed their password in specific time frame

PowerShell Copy

```
$Date = [DateTime]::Today.AddDays(-90)
Get-ADComputer -Filter 'PasswordLastSet -ge $Date' -Properties PasswordLastSet |
    Format-Table Name, PasswordLastSet

Name                                                  PasswordLastSet
----                                                  ---------------
```

```
USER01-SRV4                                                  3/12/2009 6:40:37 PM
USER01-SRV5                                                  3/12/2009 7:05:45 PM
```

This command gets all the computers that have changed their password in the last 90 days.

Example 4: Get computer accounts in a specific location using an LDAPFilter

PowerShell Copy

```
Get-ADComputer -LDAPFilter "(name=*laptop*)" -SearchBase "CN=Computers,DC= User0

name
----
pattiful-laptop
davidche-laptop
```

This command gets the computer accounts in the location `CN=Computers,DC=User01,DC=com` that are listed as laptops by using an **LDAPFilter**.

Example 5: Get all computer accounts using a filter

PowerShell Copy

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

This command gets all computer accounts.

Example 6: Get all computers with a name starting with Computer01 or Computer02

PowerShell Copy

```
Get-ADComputer -Filter 'Name -like "Computer01*" -or Name -like "Computer02*"' -
    Format-Table Name, DNSHostName, IPv4Address -AutoSize

name        dnshostname            ipv4address
----        -----------            -----------
Computer01-SRV1 Computer01-SRV1.Computer01.com 10.194.99.181
Computer02-SRV2 Computer02-SRV2.Computer02.com 10.194.100.3
```

Example 7: Get all computers with a name starting with a string AND password last set before 30 days

PowerShell Copy

```
$Date = [DateTime]::Today.AddDays(-30)
Get-ADComputer -Filter 'Name -like "Computer01*" -and PasswordLastSet -ge $Date'
    Format-Table Name, DNSHostName, IPv4Address -AutoSize

name        dnshostname            ipv4address
----        -----------            -----------
Computer01-SRV1 Computer01-SRV1.Computer01.com 10.194.99.181
```

This command shows the name, DNS hostname, and IPv4 address.

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

-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, the cmdlet returns a terminating error.

Expand table

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

-Filter

Specifies a query string that retrieves Active Directory objects. This string uses the Windows PowerShell Expression Language syntax. The Windows 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 Windows 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**
>
> Wildcards other than `*`, such as `?`, are not supported by the **Filter** syntax.

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

Expand table

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

-Identity

Specifies an Active Directory computer 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 Security Accounts Manager account name (`sAMAccountName`)

The cmdlet searches the default naming context or partition to find the object. If the identifier given is a distinguished name, the partition to search is computed from that distinguished name. 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 a computer object instance.

Expand table

Type:	ADComputer
Position:	0
Default value:	None
Required:	True
Accept pipeline input:	True
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.

Expand table

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

-Partition

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.

In many cases, a default value is 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 are evaluated.

In Active Directory Domain Services environments, a default value for **Partition** is 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** is set to the default partition or naming context of the target domain.

In Active Directory Lightweight Directory Services (AD LDS) environments, a default value for **Partition** is 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** is 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.

Expand table

Type:	String
Position:	Named
Default value:	None
Required:	False
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.

Expand table

Type:	String[]
Aliases:	Property
Position:	Named
Default value:	None
Required:	False
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.

Expand table

Type:	Int32
Position:	Named
Default value:	None
Required:	False

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.

Expand table

Type:	Int32
Position:	Named
Default value:	None
Required:	False
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 Active Directory Domain Services 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 AD LDS instance if one has been specified by setting the **msDS-defaultNamingContext** property of the Active Directory directory service agent 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 global catalog port, all partitions are searched. If the value of the **SearchBase** parameter is set to an empty string and you are not connected to a global catalog port, an error is thrown.

Expand table

Type:	String
Position:	Named
Default value:	None
Required:	False
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.

Expand table

Type:	ADSearchScope
Accepted values:	Base, OneLevel, Subtree
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

Inputs

None or Microsoft.ActiveDirectory.Management.ADComputer

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

Outputs

ADComputer

Returns one or more computer objects.

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

To view the properties for an **ADComputer** object, see the following examples. To run these examples, replace `<computer>` with a computer identifier such as the SAM account name of your local computer.

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

```
Get-ADComputer <computer> | Get-Member
```

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

```
Get-ADComputer <computer> -Properties ALL | Get-Member
```

Notes

- This cmdlet doesn't work with AD LDS with its default schema. By default the AD LDS schema doesn't have a computer class, but if the schema is extended to include it, this cmdlet will work with LDS.

Related Links

- Add-ADComputerServiceAccount
- Get-ADComputerServiceAccount
- New-ADComputer
- Remove-ADComputer
- Remove-ADComputerServiceAccount
- Set-ADComputer
- AD DS Administration Cmdlets in Windows PowerShell

Feedback

Was this page helpful? Yes No

Provide product feedback