Get-ADUser: Find AD Users Using PowerShell Ultimate Deep Dive

**Get-ADUser**, Arguably one of the most used cmdlets I use on a day to day basis. I’m sure the same goes for other sysadmins around the world if they’re managing a Windows environment. Today we’re going to do a deep dive on Get-ADUser and the multiple ways to find Active Directory users using Powershell. As always, let’s touch on the requirements needed to use Get-ADUser.

Table Of Contents

* [Requirements](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#Requirements)
* [Get-ADUser Examples and Parameter Overview](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#Examples)
  + [Find AD User With Identity Parameter](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#IdentityParameter)
  + [Get AD User Using The Filter Parameter](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#FilterParameter)
    - [Filter By Property](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#FilterByProperty)
    - [Filter by Operator](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#FilterByOperator)
    - [How To Use LDAP Filters](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#FilterWithLDAP)
    - [Filter Using Ambiguous Name Resolution (ANR)](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#FilterANR)
  + [Display All Of The Properties For A Specified User](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#PropertyParameter)
  + [Query Active Directory Users By Organizational Unit](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#SearchBaseParameter)
    - [Specify The OU Depth Of A Search](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#SearchScopeParameter)
  + [Target The Domain Controller Of Your Choice](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#ServerParameter)
  + [Passing Alternate Credentials](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#CredentialParameter)
* [Get-ADUser From A Different Domain](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#QueryAlternateDomain)
* [Conclusion](https://thesysadminchannel.com/get-aduser-find-active-directory-users-using-powershell-ultimate-deep-dive/#Conclusion)

Requirements

Using the Active Directory Module has a few requirements that we’ll need to make sure are up and running in order for your queries to run successfully.

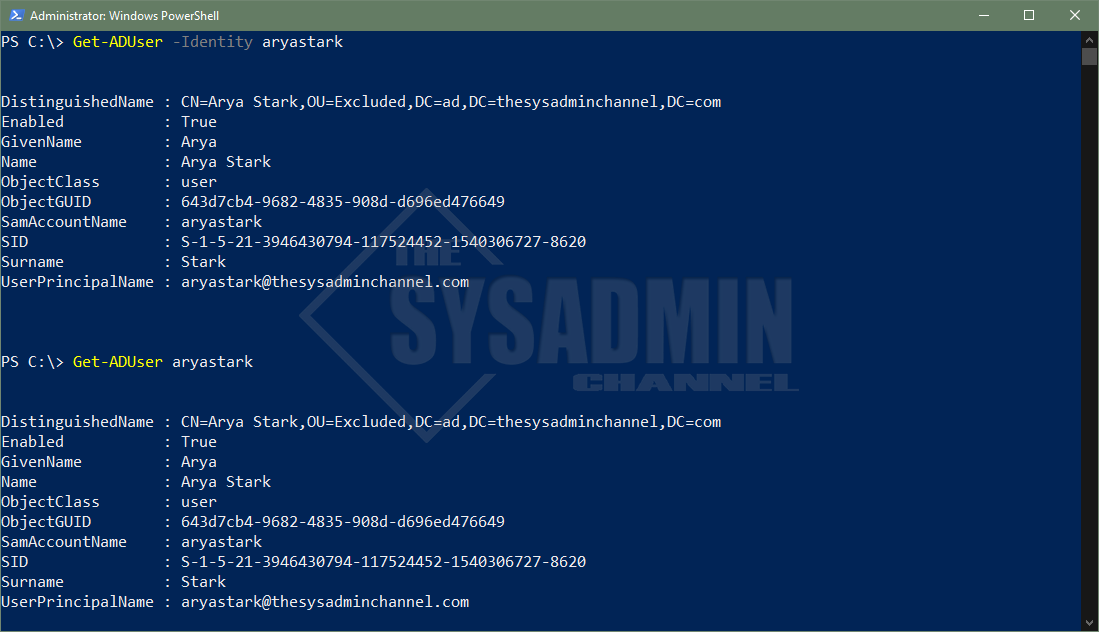
* An Active Directory Domain must be setup
* The Domain Controller you’re querying must have Active Directory Web Services Service running
* Remote Server Administration Tools (RSAT)
  + For Windows 10 1903 and later, [view setup guide](https://thesysadminchannel.com/install-rsat-for-windows-10-version-1809-1903-and-later/)
  + Active Directory Light-Weight Directory Tools Windows Feature (RSAT-AD-Tools) if running on a Windows Server

Get-ADUser Examples and Parameter Overview

In this article we’ll cover several of the parameters used in the cmdlet along with examples and screenshots so you can see exactly how to utilize these to your benefit.

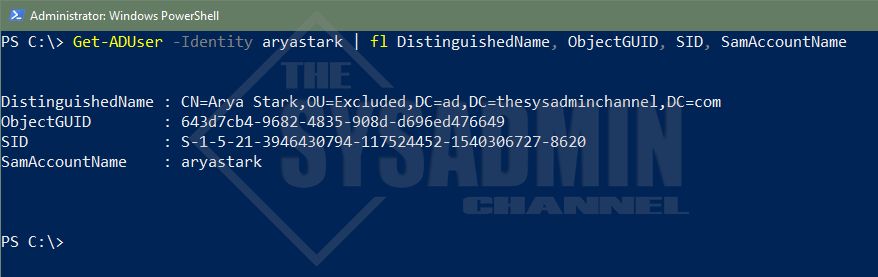
Find ADUser With Identity Parameter

Get-ADUser using the -Identity Parameter is typically the most commonly used parameter when people want to query a specific user. This is because the -Identity parameter is positioned as the first parameter so it can be omitted when running the actual query.

Example: Get-ADUser -Identity aryastark will produce the exact same results as Get-ADUser aryastark  
[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Identity-Parameter.png)

There are 4 attributes that are allowed when using Identity parameter. Let’s list them here along with an example of what it typically looks like.

* Distinguished Name
  + CN=Arya Stark,OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com
* ObjectGuid
  + 643d7cb4-9682-4835-908d-d696ed476649
* Security Identifier (SID)
  + S-1-5-21-3946430794-117524452-1540306727-8620
* sAMAccountName (username)
  + aryastark

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-AD-User-Powershell-Identity-Parameter.png)

*Example of the 4 attributes that are accepted*

Get-ADUser Using The Filter Parameter

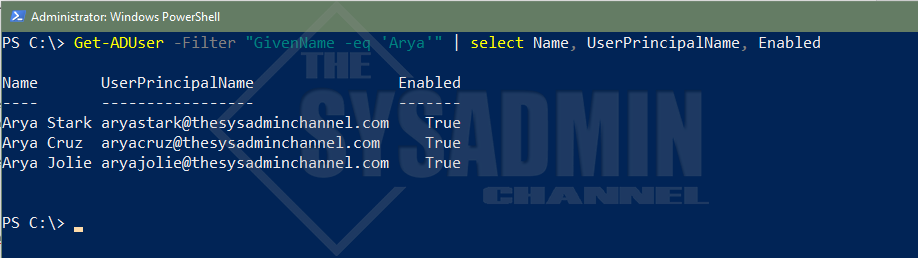
The -Filter parameter in the Get-ADUser cmdlet is definitely also another fan favorite. The phrase “Filter Left, Format right” definitely applies here in getting the data you need in a reduced amount of time. This is one of those fundamental Powershell concepts that everyone should learn early on.

**Pro-Top**: Replace Where-Object with Filter. Anytime a filter parameter is available use that instead of Where-Object for faster results.

Get AD User Properties

Let’s take a look at get ad user properties in action. Say we wanted to get everyone with the GivenName (firstname) of ‘Arya’ – What exactly would that query look like?

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | #Get All Active Directory users that have a first name of Arya  Get-ADUser -Filter "GivenName -eq 'Arya'" | select Name, UserPrincipalName, Enabled    Name       UserPrincipalName                Enabled  ----       -----------------                -------  Arya Stark aryastark@thesysadminchannel.com    True  Arya Cruz  aryacruz@thesysadminchannel.com     True  Arya Jolie aryajolie@thesysadminchannel.com    True |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Filter-Parameter-GivenName.png)

*Select object was added to condense output*

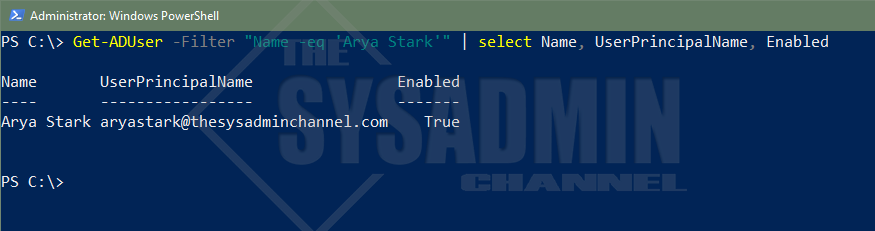
You can find other filterable attributes by choosing any one of the attributes when running -Properties \*. Commonly used filters are *UserPrincipalName*, *Surname*, *Mail* and even *Name* or *DisplayName*.

Filter With Operators

Regarding operators, there are several choices such as equal, like, less than and even greater than that’s convenient for us to use.

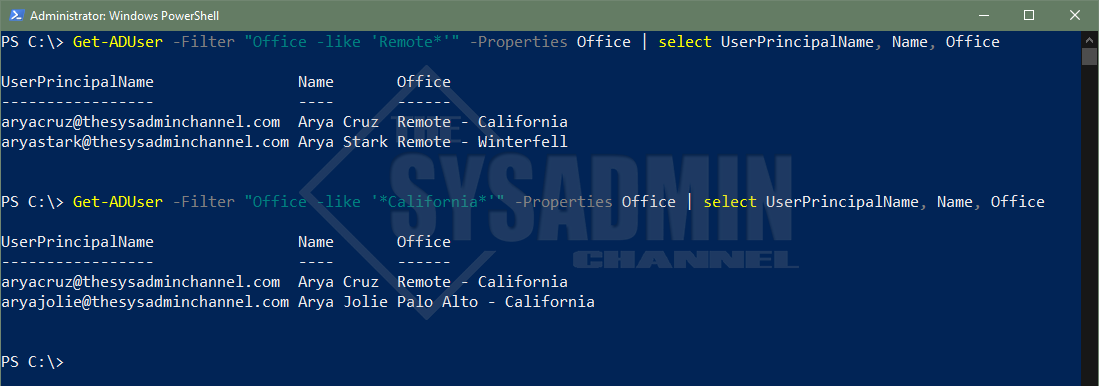
When using the -eq operator, the filter has to match the property exactly so make sure you specify the text exactly as it’s shown in AD. As noted in the above example, we searched for all users with the first name ‘Arya.’ Say we wanted to only filter for the Name ‘Arya Stark’.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | #Get the AD user whos name is Arya Stark  Get-ADUser -Filter "Name -eq 'Arya Stark'" | select Name, UserPrincipalName, Enabled      Name       UserPrincipalName                Enabled  ----       -----------------                -------  Arya Stark aryastark@thesysadminchannel.com    True |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Filter-Name-eq-Arya.png)

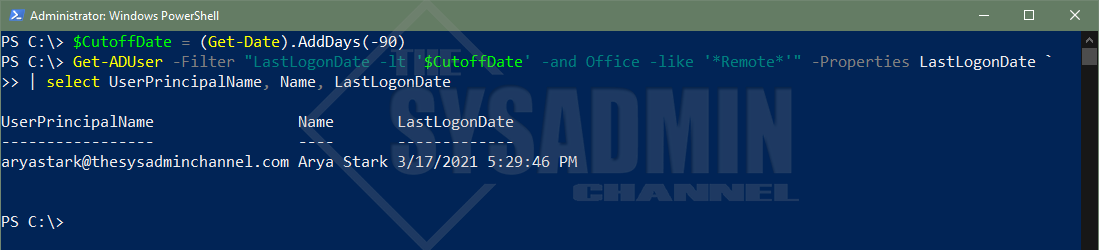
Let’s now dive into the -like operator and how to specifically use it for filters. A great example I’ve used in the past is to see who are all the people that have the word Remote in their AD Office Attribute.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | #Get all users who are remote  Get-ADUser -Filter "Office -like 'Remote\*'" -Properties Office | select UserPrincipalName, Name, Office    UserPrincipalName                Name       Office  -----------------                ----       ------  aryacruz@thesysadminchannel.com  Arya Cruz  Remote - California  aryastark@thesysadminchannel.com Arya Stark Remote - Winterfell      #Get all users who are in California  Get-ADUser -Filter "Office -like '\*California\*'" -Properties Office | select UserPrincipalName, Name, Office    UserPrincipalName                Name       Office  -----------------                ----       ------  aryacruz@thesysadminchannel.com  Arya Cruz  Remote - California  aryajolie@thesysadminchannel.com Arya Jolie Palo Alto - California |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Filter-Office-like-Operator.png)

With regard to auditing, I’ve always found [filtering accounts by LastLogonDate](https://thesysadminchannel.com/get-last-logon-date-for-all-users-in-your-domain/) has always been extremely helpful. For an in-depth write-up check out the link above. Otherwise, let’s go over a quick example to get the gist of what’s happening. We’ll also couple it with the -and operator to string multiple queries together and narrow down your filter.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | #Get Remote Users who have not logged in, in over 90 days  $CutoffDate = (Get-Date).AddDays(-90)  Get-ADUser -Filter "LastLogonDate -lt '$CutoffDate' -and Office -like '\*Remote\*'" -Properties LastLogonDate `   | select UserPrincipalName, Name, LastLogonDate    UserPrincipalName                Name       LastLogonDate  -----------------                ----       -------------  aryastark@thesysadminchannel.com Arya Stark 3/17/2021 5:29:46 PM |

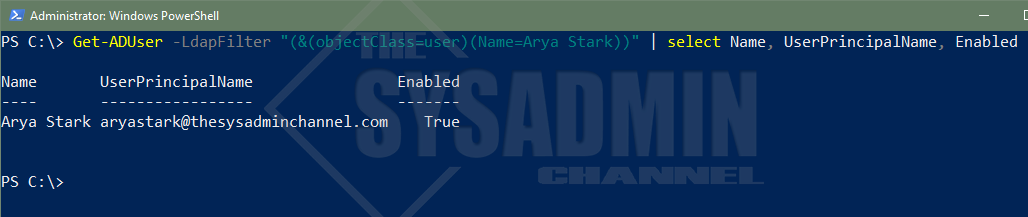
[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Filter-LastLogonDate.png)

How To Use LDAP Filters

To be perfectly honest, I can probably count the number of times on one hand that I’ve used an LDAP filter. The methods mentioned above have been ingrained into my brain since that’s how I learned. The reason being is that the syntax is a bit more complex and the standard operators like -and/-or don’t really come into play here.

If you’re great with VBScript then it might be up your alley. In any event, here we go.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | #Get AD user using an LDAP filter query  Get-ADUser -LdapFilter "(&(objectClass=user)(Name=Arya Stark))" | select Name, UserPrincipalName, Enabled    Name       UserPrincipalName                Enabled  ----       -----------------                -------  Arya Stark aryastark@thesysadminchannel.com    True |

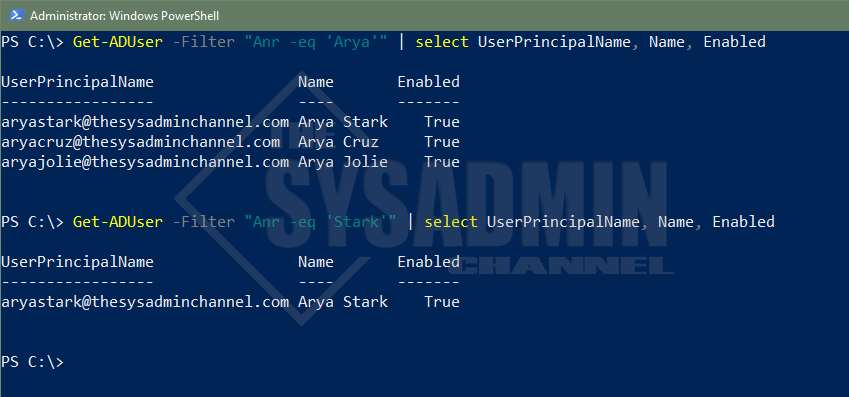
[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-LDAPFilter.png)

Filter Using Ambiguous Name Resolution (ANR)

Ambiguous Name Resolution, aka ANR, allows multiple objects to be resolved on a single query. Think of it like a built-in -like operator that queries against GivenName, Surname, DisplayName, SamAccountName, physicalDeliveryOfficeName and even the Exchange MailNickName without any added effort.

ANR is especially useful in larger organizations where people share a similar display name. It just helps to truncate multiple -and/-or queries into a single function to ease your searches. Let’s cover an example of using ambiguous name resolution in an actual filter (using Arya Stark as our example).

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | #Get all users who have Arya in their name  Get-ADUser -Filter "Anr -eq 'Arya'" | select UserPrincipalName, Name, Enabled    UserPrincipalName                Name       Enabled  -----------------                ----       -------  aryastark@thesysadminchannel.com Arya Stark    True  aryacruz@thesysadminchannel.com  Arya Cruz     True  aryajolie@thesysadminchannel.com Arya Jolie    True      #Get all users who have Stark in their name  Get-ADUser -Filter "Anr -eq 'Stark'" | select UserPrincipalName, Name, Enabled    UserPrincipalName                Name       Enabled  -----------------                ----       -------  aryastark@thesysadminchannel.com Arya Stark    True |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Filter-ANR.png)

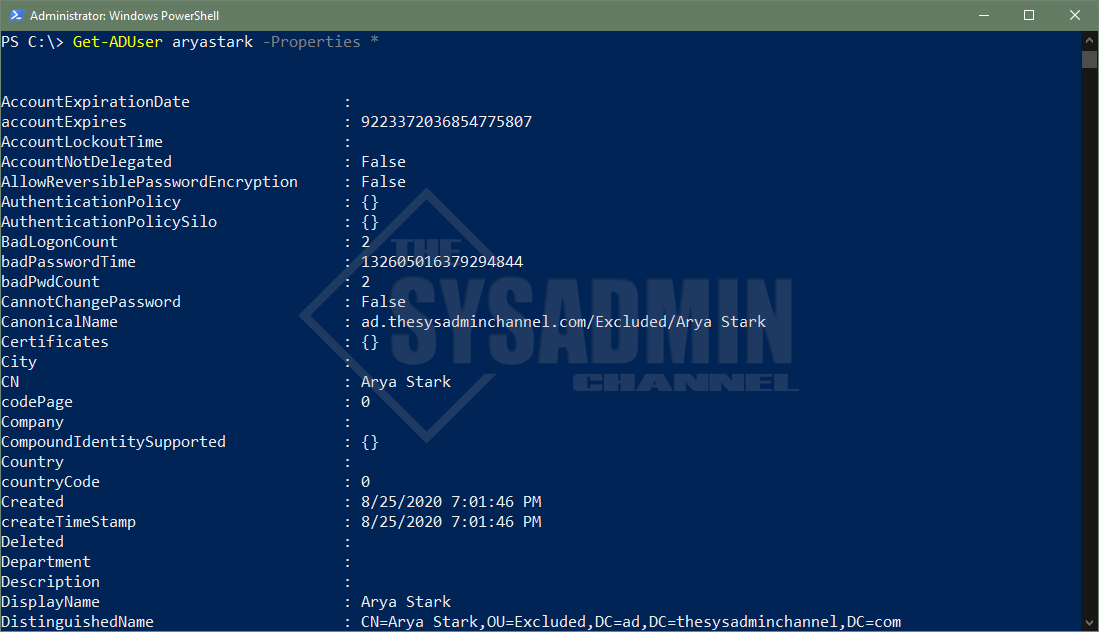
*Notice we didn’t need to specify GivenName, Surname or even use the -Like Operator.*

Display All Of The Properties For A Specified User

All Active Directory users have the same core attributes populated but they’re not displayed by default. If you notice in the examples above, I had to specify -Property in order for Powershell to know to check those AD properties. If you omit the property parameter, the filter won’t find it even though the attribute is there on the user’s account.

A good thing is this allows a wildcard (\*) so you can see what’s available. I would also recommend to explicitly specify your properties when querying many users so you’re not putting to much stress on the remote Domain Controller.

|  |  |
| --- | --- |
| 1  2 | #Get all properties for a user.  Get-ADUser aryastark -Properties \* |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Property-All.png)

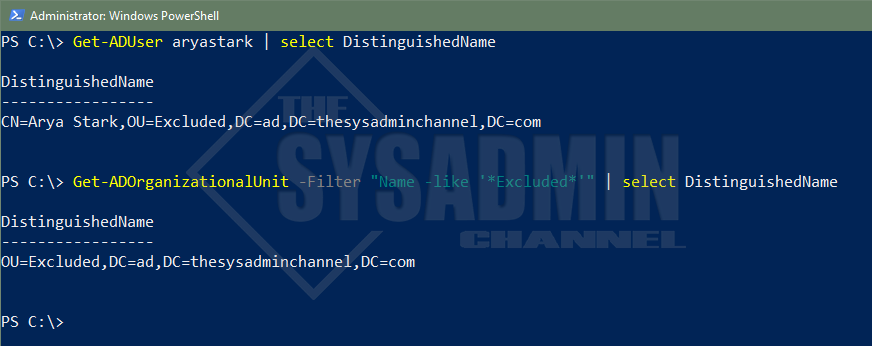
Query Active Directory Users By Organizational Unit

The ability to query users by an Organizational Unit is an excellent method to ensure you’re getting the most out of your Active Directory OU structure. A great, real world example for this would be if you have your AD Org units structured by regional location and you’re looking to get all users in that location.

SearchBase uses the DistinguishedName as the parameter input. You can grab the DN by one of 2 ways.

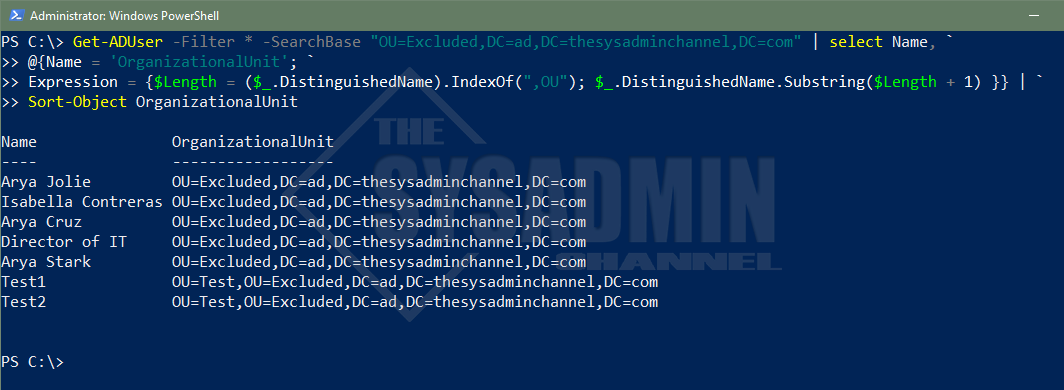
* Query a user in that OU and select the DN property. Extract OU DN from there
* Use Get-ADOrganizationalUnit and filter by name

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | #Query a user in the OU and select the DN property to get the OU syntax.  Get-ADUser aryastark    DistinguishedName  -----------------  CN=Arya Stark,OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com      #Use Get-ADOrganizationalUnit and filter by name  Get-ADOrganizationalUnit -Filter "Name -like '\*Excluded\*'" | select DistinguishedName    DistinguishedName  -----------------  OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-SearchBase.png)

Now that we have the Organizational Unit’s DistinguishedName, we can use that as the input parameter. This coupled with the -Filter parameter will help narrow your search by Org Unit.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | #Get All users under the Excluded OU.  Use a custom label to show Organizational Unit  Get-ADUser -Filter \* -SearchBase "OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com" | select Name, `  @{Name = 'OrganizationalUnit'; `  Expression = {$Length = ($\_.DistinguishedName).IndexOf(",OU"); $\_.DistinguishedName.Substring($Length + 1) }} | `  Sort-Object OrganizationalUnit    Name               OrganizationalUnit  ----               ------------------  Arya Jolie         OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Isabella Contreras OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Arya Cruz          OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Director of IT     OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Arya Stark         OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Test1              OU=Test,OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Test2              OU=Test,OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Filter-All-SearchBase-OU.png)

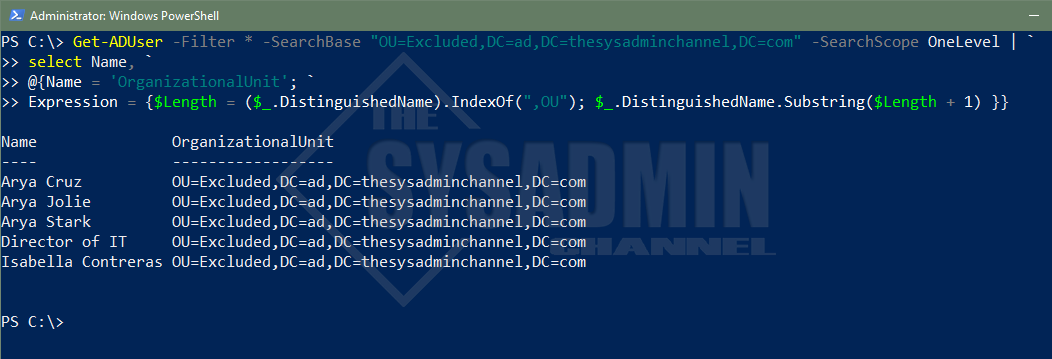
*Wildcards are also allowed to use with Filter to search for All*

Specify The OU Depth Of A Search

Building off of the SearchBase parameter from above, you might have noticed that the search was recursive. Meaning that it drilled down to all Sub OU’s without having the need to specify them. The question however, is what if we don’t want to drill down. What if we only want *that* explicit OU?

This is where the SearchScope parameter comes into play. Using the same query above, let’s exclude the two test accounts in the Test OU.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | #Get Users in the Excluded OU and Exclude the Test OU Users  Get-ADUser -Filter \* -SearchBase "OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com" -SearchScope OneLevel | `  select Name, `  @{Name = 'OrganizationalUnit'; `  Expression = {$Length = ($\_.DistinguishedName).IndexOf(",OU"); $\_.DistinguishedName.Substring($Length + 1) }}    Name               OrganizationalUnit  ----               ------------------  Arya Cruz          OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Arya Jolie         OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Arya Stark         OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Director of IT     OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com  Isabella Contreras OU=Excluded,DC=ad,DC=thesysadminchannel,DC=com |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Filter-All-SearchBase-SearchScope-OU.png)

*When SearchScope is omitted, it will default to Subtree*

Target The Domain Controller Of Your Choice

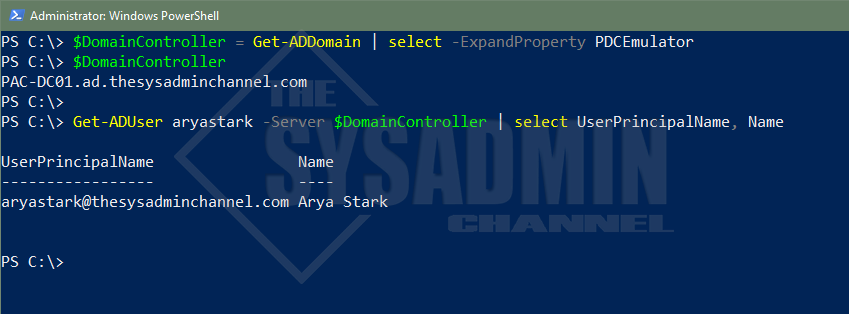
Anytime you make an Active Directory query, you’ll most likely always default to a Domain Controller in your site. This is defined by Active Directory Sites and Services and an easy way to check what Domain Controller you’re currently authenticating against is to use $env:LogonServer.

This is great and all, but what if you wanted to query a Domain Controller in another site, perhaps one across the globe? You would use the -Server parameter to do this. Specifically for me, I always like to use the Primary Domain Controller, PDC Emulator, as this is the heart of all replication changes. If you specify this Domain Controller specifically, you can avoid waiting for replication and can move on with your script without adding sleep commands.

You can use Powershell to query, or [transfer FSMO Roles to a different Domain Controller](https://thesysadminchannel.com/how-to-transfer-fsmo-roles-in-server-2019-using-powershell/).

Let’s walk through an example for how to use the server parameter to specify the PDC emulator dynamically.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | #Get PDC Emulator dynamically and save it to a variable for later use  $DomainController = Get-ADDomain | select -ExpandProperty PDCEmulator  $DomainController  PAC-DC01.ad.thesysadminchannel.com    Get-ADUser aryastark -Server $DomainController | select UserPrincipalName, Name    UserPrincipalName                Name  -----------------                ----  aryastark@thesysadminchannel.com Arya Stark |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Server-Parameter.png)

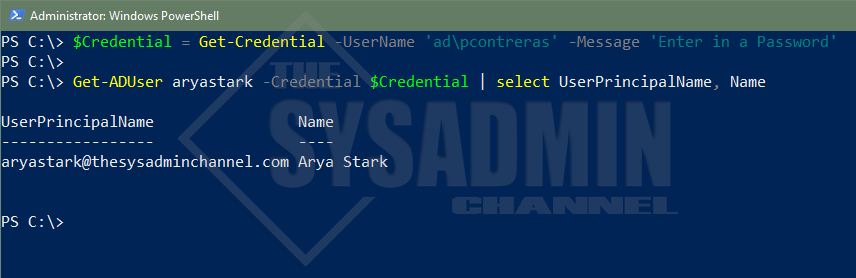
*Using the Server parameter can bypass replication times and it recommended for automation.*

Passing Alternate Credentials for Get-ADUser

Being able to pass a different set of credentials would come in handy for use cases like automation or other use cases like users in a different domain. Since Active Directory grants read-only access to all users by default, there really isn’t a need to pass in alternate credentials if you’re querying something in the same domain. It should be able to do it with no problem.

When this comes in handy is if you need to make changes to AD Objects and you need to use different credentials. To make this happen you’ll use the -Credential parameter and use Get-Credential to securely set the username and password. Since we’re so keen on examples, let’s test it.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | #Save user credentials into a variable using Get-Credential  $Credential = Get-Credential -UserName 'ad\pcontreras' -Message 'Enter in a Password'  PS C:\>  Get-ADUser aryastark -Credential $Credential | select UserPrincipalName, Name    UserPrincipalName                Name  -----------------                ----  aryastark@thesysadminchannel.com Arya Stark |

[](https://thesysadminchannel.com/wp-content/uploads/2021/10/Get-ADUser-Credential-Parameter.png)

*In the sprit of this article, we’ll pass on credentials for Get-ADUser*

Get-ADUser From A Different Domain

If you happen to have multiple Domains in your forest and you’re too lazy to Remote Desktop into a Domain Controller on that domain to run the query (guilty of it myself from time to time), it’s absolutely helpful to be able to run your query from a single machine. You can do this by combining two of the parameters above. Those parameters being -Credential as well as -Server.

I don’t have any other domains in my forest so I won’t be able to provide a working screenshot. However, one thing to keep in mind is that you’ll need to provide the Fully Qualified Domain Name (FQDN) for the remote DC. Overall, the basic syntax should look like this:

|  |  |
| --- | --- |
| 1  2  3  4 | #Save user credentials into a variable using Get-Credential  $Credential = Get-Credential -UserName 'otherdomain\myaccount' -Message 'Enter in a Password'    Get-ADUser myaccount -Server DC01.otherdomain.thesysadminchannel.com -Credential $Credential |

Conclusion

Hopefully this deep dive on how to use Powershell Get AD User has been incredible helpful for you. I’m also hoping you learned a thing or two that you can implement in your environment. As I mentioned, Get-ADUser is probably one of the most fundamental cmdlets that anyone administrator should have in their arsenal of tools.

It can be useful, especially when providing reports on the current state of your environment. If you liked this article, feel free to browse our other [Active Directory](https://thesysadminchannel.com/active-directory/) as well as our own personal [Powershell gallery](https://thesysadminchannel.com/powershell/" \t "_blank) full of useful scripts. Finally, if you’re interested in video content, check out our [Youtube Channel](https://www.youtube.com/c/theSysadminChannel" \t "_blank) for sysadmin videos