ADReportingTools Help Manual v1.1.0

Table of Contents

Introduction . . . 1

ADReportingTools . . . 2

Installation . . . 3

Design Philosophy . . . 4

Module Commands . . . 5

Get-ADReportingTools . . . 5

Users . . . 5

Get-ADDepartment . . . 8

Groups . . . 9

Computers . . . 11

Reports . . . 13

Get-NTDSInfo . . . 14

Get-ADBackupStatus . . . 14

Format and Type Extensions . . . 22

ADReportingToolsOptions . . . 23

Future Work . . . 25

Magical Thinking . . . 25

Module Functions . . . 26

Get-ADBackupStatus . . . 27

Synopsis . . . 27

Syntax . . . 27

Description . . . 27

Examples . . . 27

Parameters . . . 27

Inputs . . . 28

Outputs . . . 28

Notes . . . 28

Related Links . . . 29

Get-ADBranch . . . 30

Synopsis . . . 30

Syntax . . . 30

Description . . . 30

Examples . . . 30

Parameters . . . 31

Inputs . . . 33

Outputs . . . 33

Notes . . . 33

Related Links . . . 33

Get-ADCanonicalUser . . . 34

Synopsis . . . 34

Syntax . . . 34

Description . . . 34
Examples . . . 34
Parameters . . . 35
Inputs . . . 36
Outputs . . . 37
Notes . . . 37
Related Links . . . 37
Get-ADDepartment . . . 38
Synopsis . . . 38
Syntax . . . 38
Description . . . 38
Examples . . . 38
Parameters . . . 39
Inputs . . . 40
Outputs . . . 40
Related Links . . . 40
Get-ADDomainControllerHealth . . . 41
Synopsis . . . 41
Syntax . . . 41
Description . . . 41
Examples . . . 41
Parameters . . . 42
Inputs . . . 43
Outputs . . . 43
Notes . . . 43
Related Links . . . 43
Get-ADFSMO . . . 44
Synopsis . . . 44
Syntax . . . 44
Description . . . 44
Examples . . . 44
Parameters . . . 44
Inputs . . . 45
Outputs . . . 46
Notes . . . 46
Related Links . . . 46
Get-ADGroupReport . . . 47
Synopsis . . . 47
Syntax . . . 47
Description . . . 47
Examples . . . 47
Parameters . . . 49
Inputs . . . 51

Outputs . . . 51
Notes . . . 51
Related Links . . . 51
Get-ADGroupUser . . . 52
Synopsis . . . 52
Syntax . . . 52
Description . . . 52
Examples . . . 52
Parameters . . . 53
Inputs . . . 54
Outputs . . . 54
Notes . . . 54
Related Links . . . 54
Get-ADReportingTools . . . 55
Synopsis . . . 55
Syntax . . . 55
Description . . . 55
Examples . . . 55
Parameters . . . 55
Inputs . . . 55
Outputs . . . 55
Notes . . . 55
Related Links . . . 56
Get-ADReportingToolsOptions . . . 57
Synopsis . . . 57
Syntax . . . 57
Description . . . 57
Examples . . . 57
Parameters . . . 57
Inputs . . . 57
Outputs . . . 58
Notes . . . 58
Related Links . . . 58
Get-ADSiteDetail . . . 59
Synopsis . . . 59
Syntax . . . 59
Description . . . 59
Examples . . . 59
Parameters . . . 59
Inputs . . . 60
Outputs . . . 61
Notes . . . 61
Related Links . . . 61

Get-ADSiteSummary. 62
Synopsis 62
Syntax 62
Description. 62
Examples 62
Parameters. 62
Inputs 63
Outputs. 64
Notes. 64
Related Links 64
Get-ADSummary 65
Synopsis 65
Syntax 65
Description. 65
Examples 65
Parameters. 65
Inputs 66
Outputs. 66
Notes. 66
Related Links 67
Get-ADUserAudit 68
Synopsis 68
Syntax 68
Description. 68
Examples 68
Parameters. 69
Inputs 70
Outputs. 70
Notes. 70
Related Links 71
Get-ADUserCategory 72
Synopsis 72
Syntax 72
Description. 72
Examples 72
Parameters. 74
Inputs 75
Outputs. 76
Notes. 76
Related Links 76
Get-NTDSInfo 77
Synopsis 77
Syntax 77

Description . . . 77
Examples . . . 77
Parameters . . . 77
Inputs . . . 78
Outputs . . . 78
Notes . . . 78
Related Links . . . 78
New-ADChangeReport . . . 79
Synopsis . . . 79
Syntax . . . 79
Description . . . 79
Examples . . . 79
Parameters . . . 79
Inputs . . . 82
Outputs . . . 83
Notes . . . 83
Related Links . . . 83
New-ADDomainReport . . . 84
Synopsis . . . 84
Syntax . . . 84
Description . . . 84
Examples . . . 84
Parameters . . . 84
Inputs . . . 86
Outputs . . . 87
Notes . . . 87
Related Links . . . 87
Open-ADReportingToolsHelp . . . 88
Synopsis . . . 88
Syntax . . . 88
Description . . . 88
Examples . . . 88
Parameters . . . 88
Inputs . . . 88
Outputs . . . 88
Notes . . . 88
Related Links . . . 89
Set-ADReportingToolsOptions . . . 90
Synopsis . . . 90
Syntax . . . 90
Description . . . 90
Examples . . . 90
Parameters . . . 90

Inputs . . . 91
Outputs . . . 91
Notes . . . 91
Related Links . . . 91
Show-DomainTree . . . 92
Synopsis . . . 92
Syntax . . . 92
Description . . . 92
Examples . . . 92
Parameters . . . 94
Inputs . . . 95
Outputs . . . 96
Notes . . . 96
Related Links . . . 96
Split-DistinguishedName . . . 97
Synopsis . . . 97
Syntax . . . 97
Description . . . 97
Examples . . . 97
Parameters . . . 97
Inputs . . . 98
Outputs . . . 98
Notes . . . 98
Related Links . . . 98
Changelog for ADReportingTools . . . 99
1.0.0 . . . 100
0.8.0-preview . . . 101
0.7.0-preview . . . 102
0.6.1-preview . . . 103
0.6.0-preview . . . 104
0.5.0 . . . 105
0.4.0 . . . 106
0.3.0 . . . 107
0.2.0 . . . 108
0.0.1 . . . 109

Introduction

This manual is a PDF version of several module-related reference files as well as all of the command help. The goal is to provide a single source for all module documentation. Be aware that many of the source files contain internal cross-references. Best efforts have been made to port those links to this document, but a few links may fail to open. External links should work as expected.

If you need to ask a question or report a problem, please visit the module's Github repository.

ADReportingTools

This module contains a collection of PowerShell tools that you can use to generate reports and gather information about your Active Directory domain. Many of these commands will require the ActiveDirectory module, which you can get by installing the Remote Server Administration Tools (RSAT) for Active Directory on Windows 10.

```
Get-WindowsCapability -Online -Name RSAT.Active* | Add-WindowsCapability -online
```

The assumption is that you will run these commands with administrator credentials from a Windows 10 desktop. You should not need console access to a domain controller. These commands are for working with a local Active Directory infrastructure, not anything in Azure.

Installation

This module is available in the PowerShell Gallery. Install it with `Install-Module` after you have installed the Active Directory RSAT capability.

```
Install-Module -name ADReportingTools -force
```

Once installed, you can run a command like `Get-ADReportingTools` to see list of commands. Or run `Open-ADReportingToolsHelp` to launch a PDF version of this file, as well as command documentation.

Design Philosophy

The Active Directory module from Microsoft is not especially difficult to use. It is quite easy to get information from Active Directory.

```
Get-ADuser -filter "department -eq 'sales'" -properties Title,Department
```

However, you have to be very explicit about what information you want to see. You might need to create complicated filters. You need to know the Active Directory property names. Finally, you need to format the results into something meaningful. It might be better to think of the ActiveDirectory module as a *framework*.

The ADReportingTools module is built on this framework. The goal is to create a set of commands and tools to make it very easy to get information out of Active Directory in meaningful and useful ways. Many of the functions in this module are wrappers for underlying ActiveDirectory module commands, written to be easy to use.

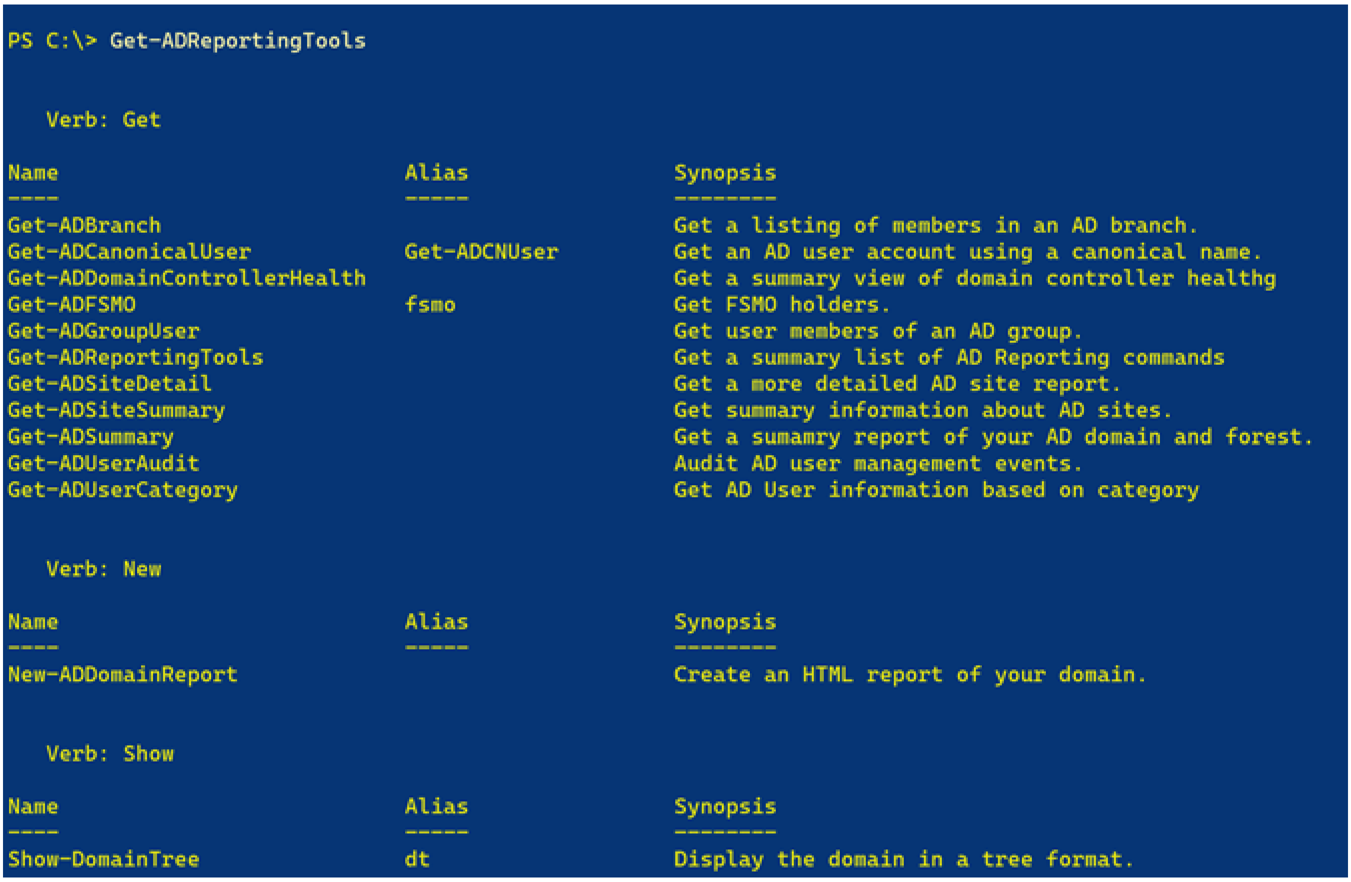
The ADReportingTools focuses primarily on working with Active Directory users, groups, and computers. The module includes commands designed to be true reporting commands. As the module name suggests, module commands are intended to ***get*** information from Active Directory. This module is not designed to manage it. There are **no** commands to set, create, or remove anything from Active Directory.

These commands have not been tested in a large domain environment, or one with cross-domain trusts and/or nested groups that cross domains. If you have used the ActiveDirectory modules in the past and had poor performance due to these types of circumstances, the modules in this command most likely won’t perform any better.

Module Commands

Get-ADReportingTools

`Get-ADReportingTools` is a meta-command. Run this command to get a formatted list of available commands in the ADReportingTools module.

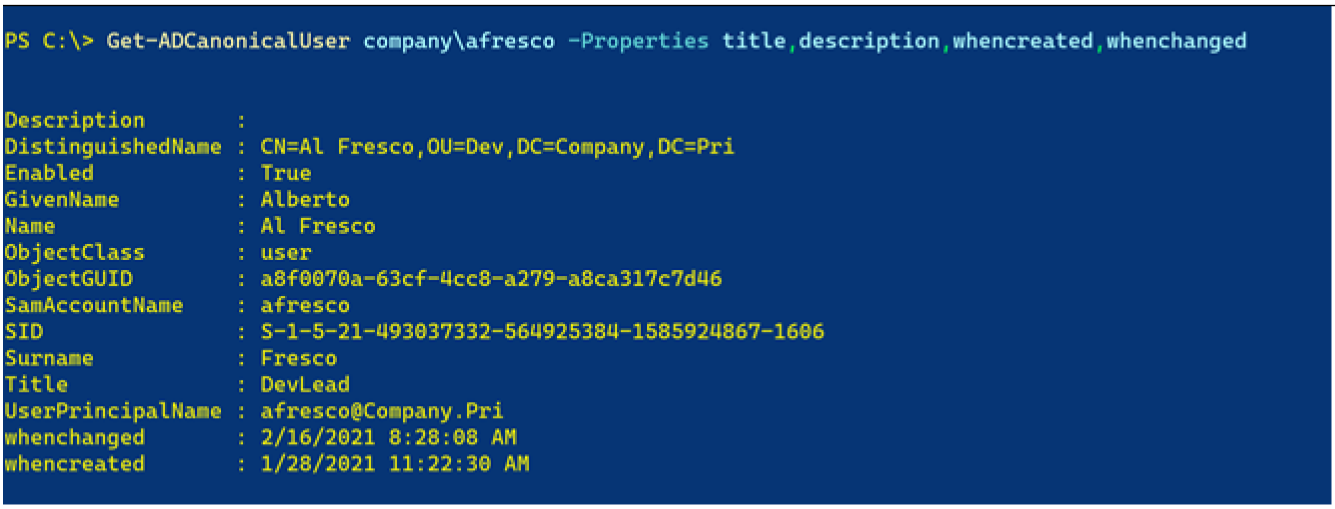


```
PS C:\> Get-ADReportingTools


   Verb: Get

Name                           Alias                Synopsis
----                           -----                --------
Get-ADBranch                                        Get a listing of members in an AD branch.
Get-ADCanonicalUser            Get-ADCNUser         Get an AD user account using a canonical name.
Get-ADDomainControllerHealth                        Get a summary view of domain controller healthg
Get-ADFSMO                     fsmo                 Get FSMO holders.
Get-ADGroupUser                                     Get user members of an AD group.
Get-ADReportingTools                                Get a summary list of AD Reporting commands
Get-ADSiteDetail                                    Get a more detailed AD site report.
Get-ADSiteSummary                                   Get summary information about AD sites.
Get-ADSummary                                       Get a sumamry report of your AD domain and forest.
Get-ADUserAudit                                     Audit AD user management events.
Get-ADUserCategory                                  Get AD User information based on category


   Verb: New

Name                           Alias                Synopsis
----                           -----                --------
New-ADDomainReport                                  Create an HTML report of your domain.


   Verb: Show

Name                           Alias                Synopsis
----                           -----                --------
Show-DomainTree                dt                   Display the domain in a tree format.
```

Users

Get-ADCanonicalUser

Often you will find user names in the form domain\username. This command makes it easier to find the Active Directory user account using this value. If the Active Directory Recycle Bin feature is enabled, you can use the `IncludeDeletedObjects` parameter to search for the user account if it can’t be found with the initial search.

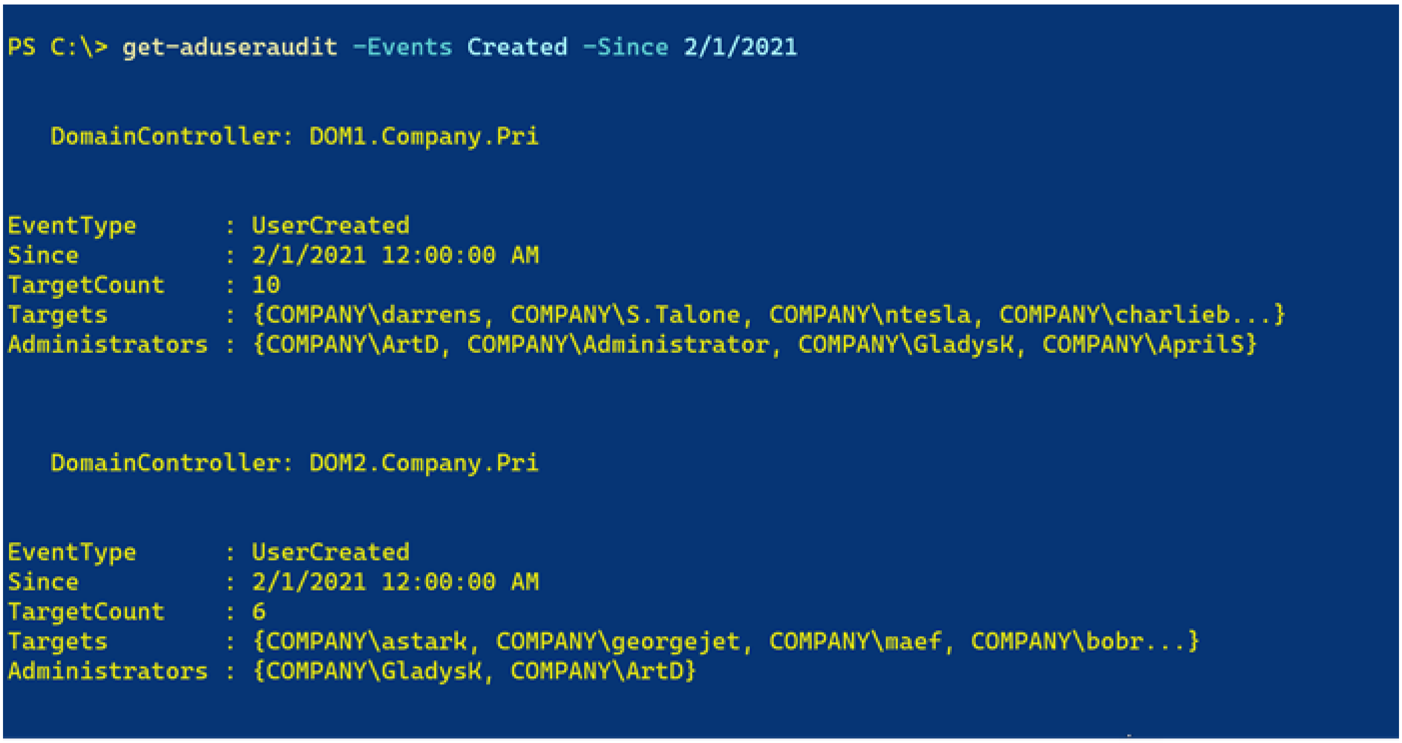


```
PS C:\> Get-ADCanonicalUser company\afresco -Properties title,description,whencreated,whenchanged


Description       :
DistinguishedName : CN=Al Fresco,OU=Dev,DC=Company,DC=Pri
Enabled           : True
GivenName         : Alberto
Name              : Al Fresco
ObjectClass       : user
ObjectGUID        : a8f0070a-63cf-4cc8-a279-a8ca317c7d46
SamAccountName    : afresco
SID               : S-1-5-21-493037332-564925384-1585924867-1606
Surname           : Fresco
Title             : DevLead
UserPrincipalName : afresco@Company.Pri
whenchanged       : 2/16/2021 8:28:08 AM
whencreated       : 1/28/2021 11:22:30 AM
```

Get-ADUserAudit

This command will search the Security event logs on your domain controllers for specific user-related events. These activities are not replicated, so you have to search each domain controller. Be aware that you may see related events for some actions. For example, if you create and enable a new user, you'll see multiple entries for the same event.

The output will show you the user accounts that match the search criteria, and the domain account that was responsible. Although, this command can't tell you which administrator is responsible for which activity. The best you can learn is that for a given time frame, these user accounts were managed. Or these administrators did something. You would need to search the event log on the domain controller for more information.

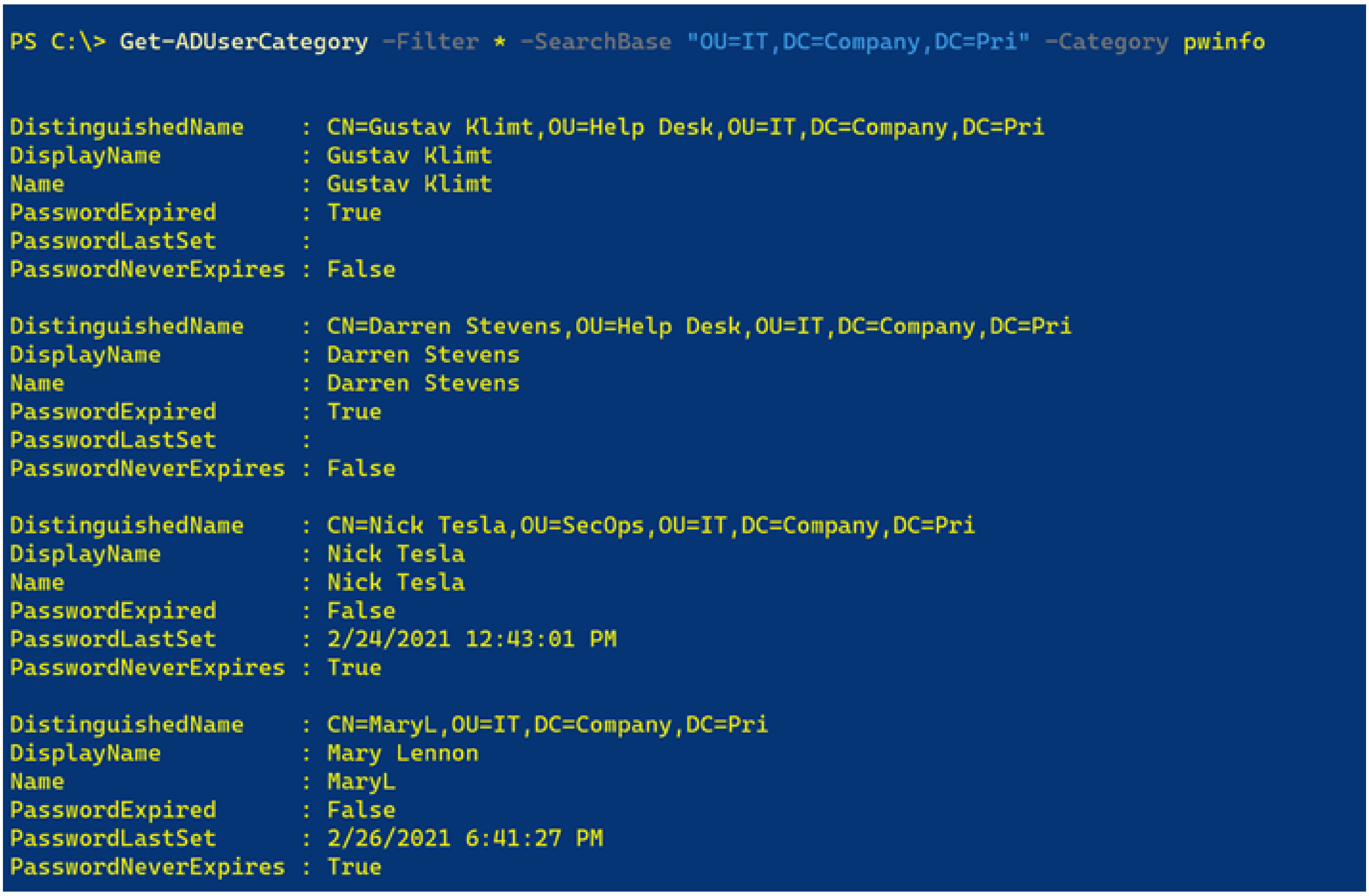


```
PS C:\> get-aduseraudit -Events Created -Since 2/1/2021


   DomainController: DOM1.Company.Pri


EventType      : UserCreated
Since          : 2/1/2021 12:00:00 AM
TargetCount    : 10
Targets        : {COMPANY\darrens, COMPANY\S.Talone, COMPANY\ntesla, COMPANY\charlieb...}
Administrators : {COMPANY\ArtD, COMPANY\Administrator, COMPANY\GladysK, COMPANY\AprilS}



   DomainController: DOM2.Company.Pri


EventType      : UserCreated
Since          : 2/1/2021 12:00:00 AM
TargetCount    : 6
Targets        : {COMPANY\astark, COMPANY\georgejet, COMPANY\maef, COMPANY\bobr...}
Administrators : {COMPANY\GladysK, COMPANY\ArtD}
```

Get-ADUserCategory

`Get-ADUserCategory` is based on the concept of getting user information from a pre-defined category. For example, you might want to get the properties DisplayName, Name, Title, Department, and Manager for a Department category. The ADReportingTools module will define a set of pre-defined categories that you can reference through `$ADUserReportingConfiguration`.

```
PS C:\> $ADUserReportingConfiguration

Name         Properties
----         ----------
Department   {DisplayName, Name, Title, Department...}
Basic        {DisplayName, Name, SamAccountname, UserPrincipalName...}
Address      {DisplayName, Name, TelephoneNumber, Office...}
Organization {DisplayName, Name, Title, Department...}
Pwinfo       {DisplayName, Name, PasswordExpired, PasswordLastSet...}
```



```
PS C:\> Get-ADUserCategory -Filter * -SearchBase "OU=IT,DC=Company,DC=Pri" -Category pwinfo


DistinguishedName    : CN=Gustav Klimt,OU=Help Desk,OU=IT,DC=Company,DC=Pri
DisplayName          : Gustav Klimt
Name                 : Gustav Klimt
PasswordExpired      : True
PasswordLastSet      :
PasswordNeverExpires : False

DistinguishedName    : CN=Darren Stevens,OU=Help Desk,OU=IT,DC=Company,DC=Pri
DisplayName          : Darren Stevens
Name                 : Darren Stevens
PasswordExpired      : True
PasswordLastSet      :
PasswordNeverExpires : False

DistinguishedName    : CN=Nick Tesla,OU=SecOps,OU=IT,DC=Company,DC=Pri
DisplayName          : Nick Tesla
Name                 : Nick Tesla
PasswordExpired      : False
PasswordLastSet      : 2/24/2021 12:43:01 PM
PasswordNeverExpires : True

DistinguishedName    : CN=MaryL,OU=IT,DC=Company,DC=Pri
DisplayName          : Mary Lennon
Name                 : MaryL
PasswordExpired      : False
PasswordLastSet      : 2/26/2021 6:41:27 PM
PasswordNeverExpires : True
```

The module ships with a JSON file that defines the categories. You can easily modify this variable to define a new category.

```
$ADUserReportingConfiguration += [pscustomobject]@{Name="Custom";Properties="DisplayName","Description"}
```

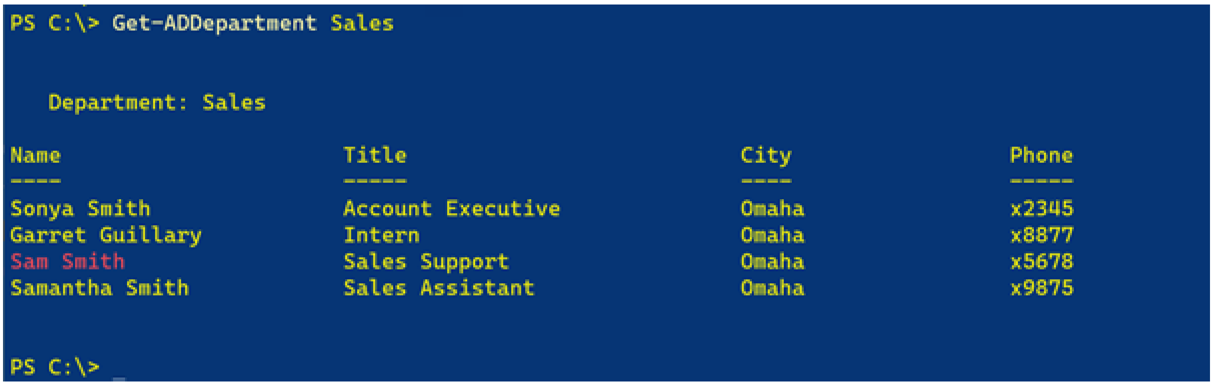
Or add a property to an existing category.

```
PS C:\> $ADUserReportingConfiguration.where({$_.name -eq 'basic'}).foreach({$_.properties+="SID"})
PS C:\> Get-ADUserCategory gladysk -Category Basic


DistinguishedName : CN=GladysK,OU=IT,DC=Company,DC=Pri
DisplayName       : Gladys Kravitz
Name              : GladysK
SamAccountname    : GladysK
UserPrincipalName : gladysk@Company.Pri
Enabled           : True
WhenCreated       : 1/25/2021 1:32:35 PM
WhenChanged       : 3/8/2021 6:52:01 PM
SID               : S-1-5-21-493037332-564925384-1585924867-1105
```

The user's distinguished name is always included in the output.

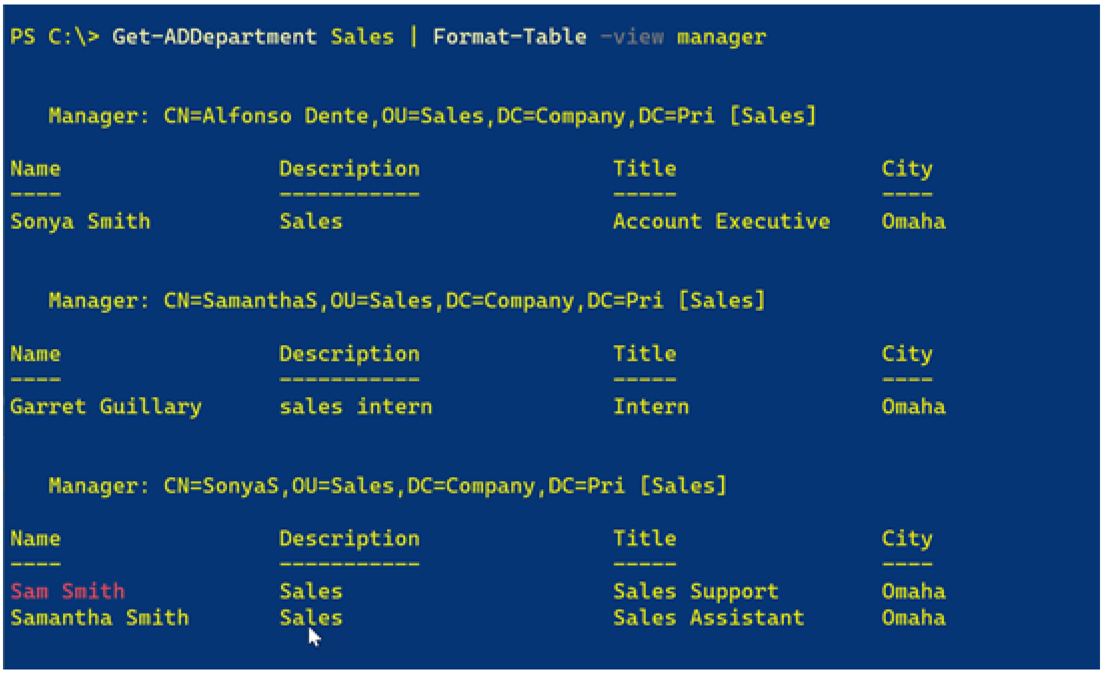
Get-ADDepartment

A related command is `Get-ADDepartment`. This command will get members of a given department. When you import the ADReportingTools module, it will define a global variable called `ADReportingHash`, which is a hashtable. The variable has a key called `Departments`. This variable is used in an argument completer for the `Department` parameter so that you can tab-complete the parameter value.



```
PS C:\> Get-ADDepartment Sales


   Department: Sales

Name                      Title                          City                 Phone
----                      -----                          ----                 -----
Sonya Smith               Account Executive              Omaha                x2345
Garret Guillary           Intern                         Omaha                x8877
Sam Smith                 Sales Support                  Omaha                x5678
Samantha Smith            Sales Assistant                Omaha                x9875


PS C:\> _
```

Disabled accounts will be displayed in red. Or you can use one of the custom views.



```
PS C:\> Get-ADDepartment Sales | Format-Table -view manager


   Manager: CN=Alfonso Dente,OU=Sales,DC=Company,DC=Pri [Sales]

Name                     Description               Title               City
----                     -----------               -----               ----
Sonya Smith              Sales                     Account Executive   Omaha


   Manager: CN=SamanthaS,OU=Sales,DC=Company,DC=Pri [Sales]

Name                     Description               Title               City
----                     -----------               -----               ----
Garret Guillary          sales intern              Intern              Omaha


   Manager: CN=SonyaS,OU=Sales,DC=Company,DC=Pri [Sales]

Name                     Description               Title               City
----                     -----------               -----               ----
Sam Smith                Sales                     Sales Support       Omaha
Samantha Smith           Sales                     Sales Assistant     Omaha
```

Split-DistinguishedName

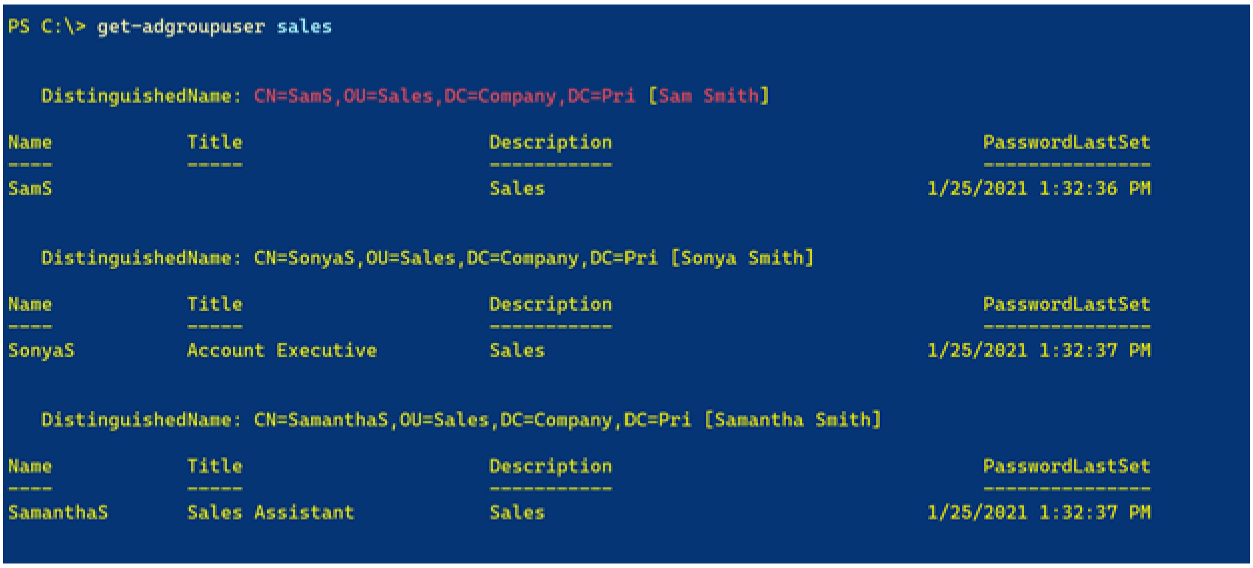
This command will take an Active Directory distinguishedname and break it down into its component elements. The command does not test or verify any of the elements. It is merely parsing a text string.

```
PS C:\> Split-DistinguishedName "CN=Foo,OU=Bar,OU=Oz,DC=Research,DC=Globomantics,DC=com"


Name      : Foo
Branch    : Bar
BranchDN  : OU=Bar,OU=Oz,DC=Research,DC=Globomantics,DC=com
Domain    : Research
DomainDN  : DC=Research,DC=Globomantics,DC=com
DomainDNS : Research.Globomantics.com
```

Groups

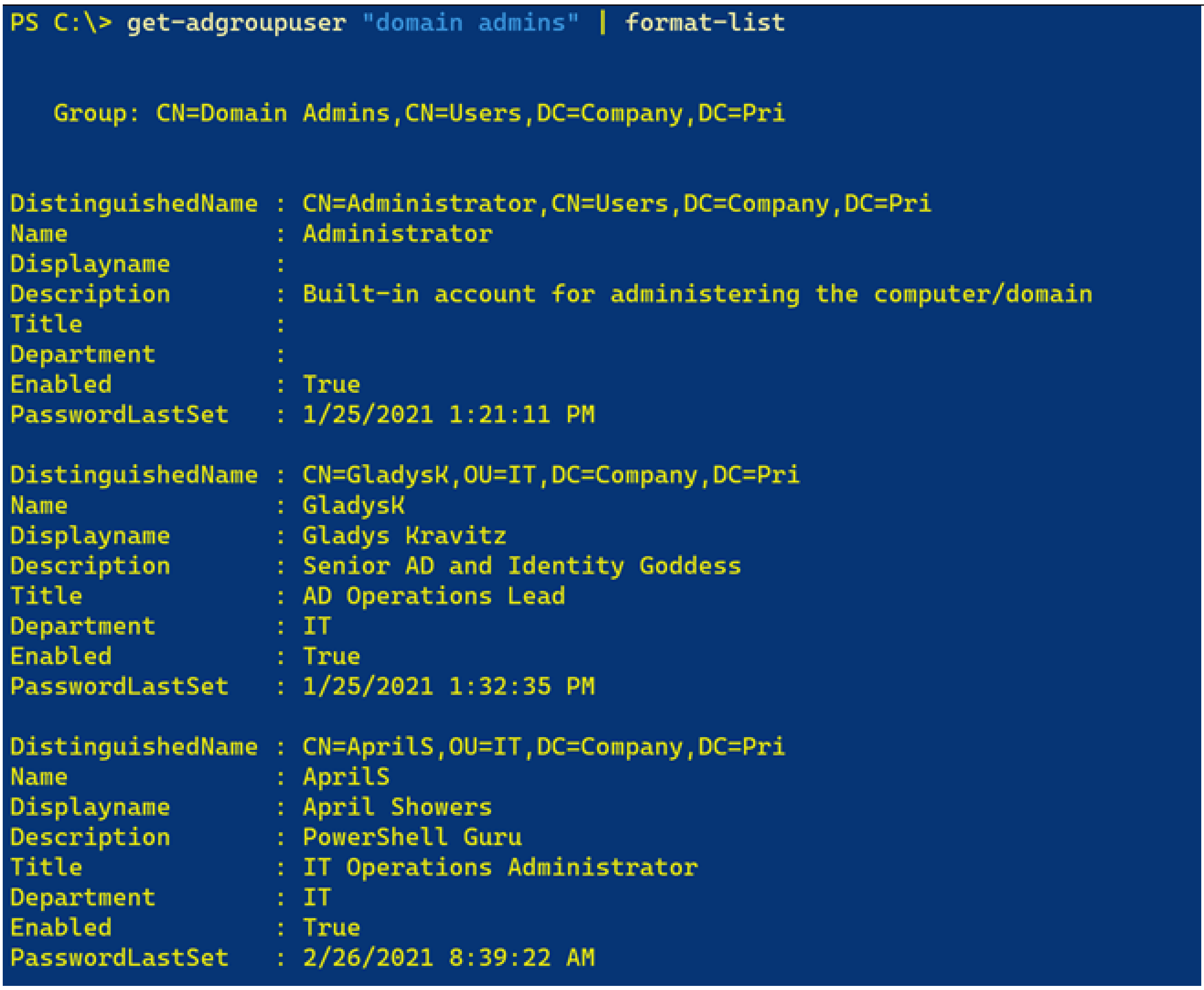
Get-ADGroupUser

The `Get-ADGroupUser` command will display all users of a given Active Directory group. The search is automatically recursive. The default output is a formatted table that will highlight disabled accounts in red. The ANSI color coding will only work in a console session.



```
PS C:\> get-adgroupuser sales

   DistinguishedName: CN=SamS,OU=Sales,DC=Company,DC=Pri [Sam Smith]

Name            Title                     Description                                 PasswordLastSet
----            -----                     -----------                                 ---------------
SamS                                      Sales                                  1/25/2021 1:32:36 PM

   DistinguishedName: CN=SonyaS,OU=Sales,DC=Company,DC=Pri [Sonya Smith]

Name            Title                     Description                                 PasswordLastSet
----            -----                     -----------                                 ---------------
SonyaS          Account Executive         Sales                                  1/25/2021 1:32:37 PM

   DistinguishedName: CN=SamanthaS,OU=Sales,DC=Company,DC=Pri [Samantha Smith]

Name            Title                     Description                                 PasswordLastSet
----            -----                     -----------                                 ---------------
SamanthaS       Sales Assistant           Sales                                  1/25/2021 1:32:37 PM
```

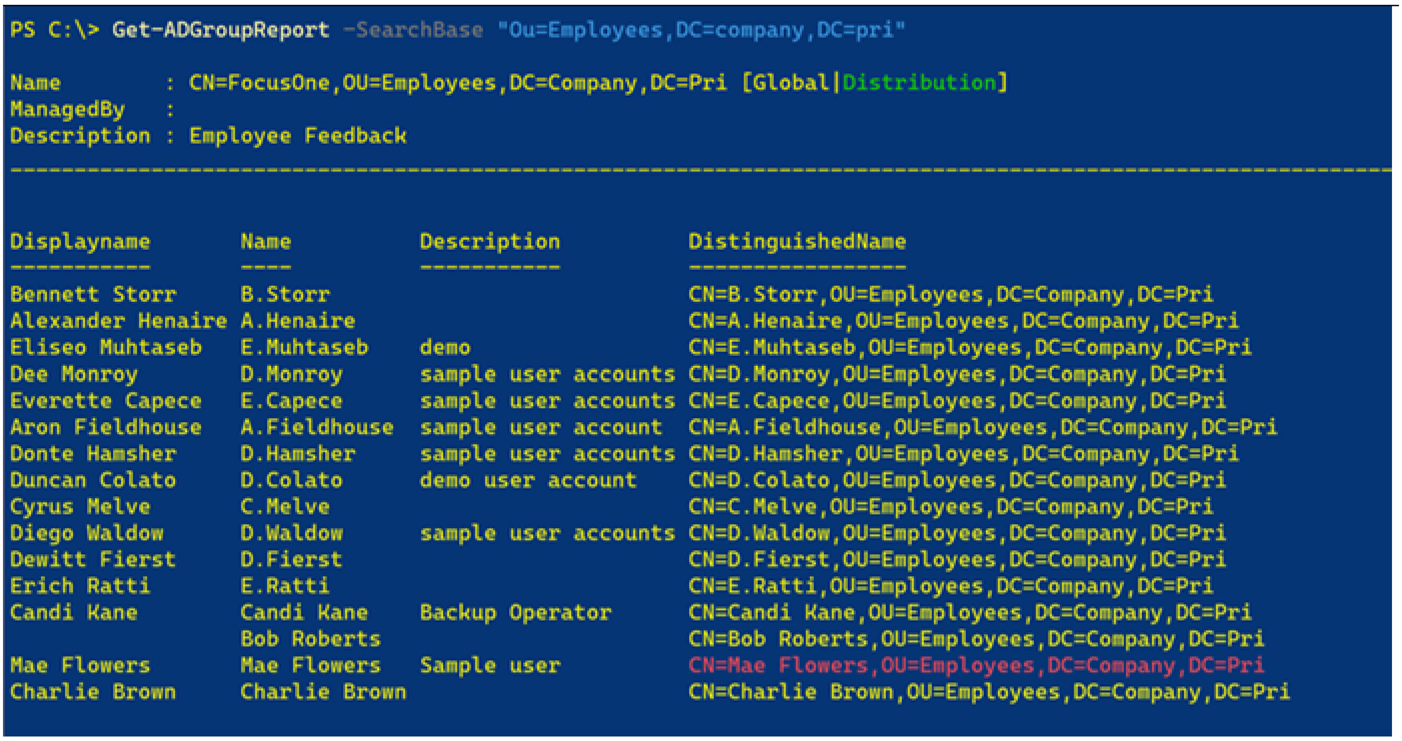
Or you can use the default list view.



```
PS C:\> get-adgroupuser "domain admins" | format-list


   Group: CN=Domain Admins,CN=Users,DC=Company,DC=Pri


DistinguishedName : CN=Administrator,CN=Users,DC=Company,DC=Pri
Name              : Administrator
Displayname       :
Description       : Built-in account for administering the computer/domain
Title             :
Department        :
Enabled           : True
PasswordLastSet   : 1/25/2021 1:21:11 PM

DistinguishedName : CN=GladysK,OU=IT,DC=Company,DC=Pri
Name              : GladysK
Displayname       : Gladys Kravitz
Description       : Senior AD and Identity Goddess
Title             : AD Operations Lead
Department        : IT
Enabled           : True
PasswordLastSet   : 1/25/2021 1:32:35 PM

DistinguishedName : CN=AprilS,OU=IT,DC=Company,DC=Pri
Name              : AprilS
Displayname       : April Showers
Description       : PowerShell Guru
Title             : IT Operations Administrator
Department        : IT
Enabled           : True
PasswordLastSet   : 2/26/2021 8:39:22 AM
```

GetADGroupReport

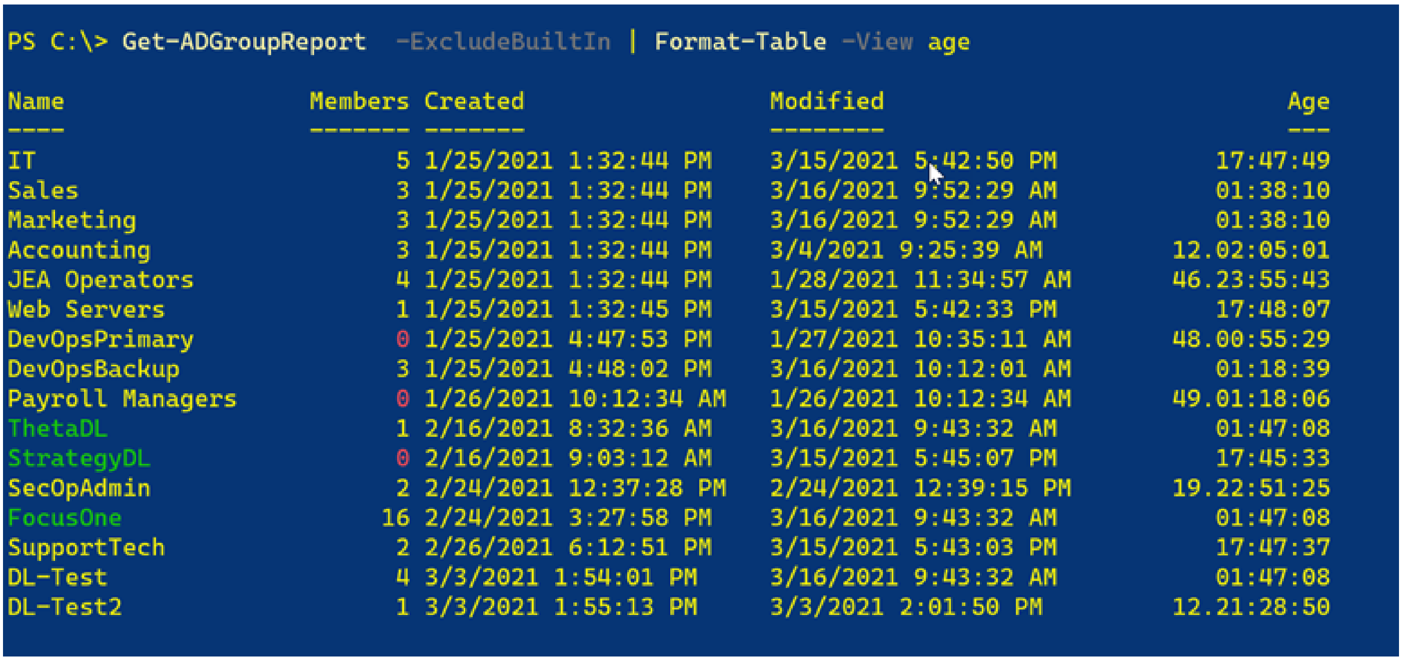
`Get-ADGroupReport` will create a custom report for a group showing members. `Get-ADGroupUser` is intended to display group membership details `Get-ADGroupReport` focuses on the group, although members are also displayed. Members are always gathered recursively. You can filter for specific types of groups. You can also opt to exclude groups under CN=Users and CN=BuiltIn. The groups "Domain Users", "Domain Computers", and "Domain Guests" are always excluded from this command.



```
PS C:\> Get-ADGroupReport -SearchBase "Ou=Employees,DC=company,DC=pri"

Name        : CN=FocusOne,OU=Employees,DC=Company,DC=Pri [Global|Distribution]
ManagedBy   :
Description : Employee Feedback
-----------------------------------------------------------------------------------------------------------


Displayname       Name          Description          DistinguishedName
-----------       ----          -----------          -----------------
Bennett Storr     B.Storr                            CN=B.Storr,OU=Employees,DC=Company,DC=Pri
Alexander Henaire A.Henaire                          CN=A.Henaire,OU=Employees,DC=Company,DC=Pri
Eliseo Muhtaseb   E.Muhtaseb    demo                 CN=E.Muhtaseb,OU=Employees,DC=Company,DC=Pri
Dee Monroy        D.Monroy      sample user accounts CN=D.Monroy,OU=Employees,DC=Company,DC=Pri
Everette Capece   E.Capece      sample user accounts CN=E.Capece,OU=Employees,DC=Company,DC=Pri
Aron Fieldhouse   A.Fieldhouse  sample user account  CN=A.Fieldhouse,OU=Employees,DC=Company,DC=Pri
Donte Hamsher     D.Hamsher     sample user accounts CN=D.Hamsher,OU=Employees,DC=Company,DC=Pri
Duncan Colato     D.Colato      demo user account    CN=D.Colato,OU=Employees,DC=Company,DC=Pri
Cyrus Melve       C.Melve                            CN=C.Melve,OU=Employees,DC=Company,DC=Pri
Diego Waldow      D.Waldow      sample user accounts CN=D.Waldow,OU=Employees,DC=Company,DC=Pri
Dewitt Fierst     D.Fierst                           CN=D.Fierst,OU=Employees,DC=Company,DC=Pri
Erich Ratti       E.Ratti                            CN=E.Ratti,OU=Employees,DC=Company,DC=Pri
Candi Kane        Candi Kane    Backup Operator      CN=Candi Kane,OU=Employees,DC=Company,DC=Pri
                  Bob Roberts                        CN=Bob Roberts,OU=Employees,DC=Company,DC=Pri
Mae Flowers       Mae Flowers   Sample user          CN=Mae Flowers,OU=Employees,DC=Company,DC=Pri
Charlie Brown     Charlie Brown                      CN=Charlie Brown,OU=Employees,DC=Company,DC=Pri
```

If your PowerShell hosts supports it, ANSI color schemes will be used to highlight things such as Distribution groups and disabled user accounts.

You can also use a custom table view.



```
PS C:\> Get-ADGroupReport  -ExcludeBuiltIn | Format-Table -View age

Name                    Members Created                Modified                      Age
----                    ------- -------                --------                      ---
IT                            5 1/25/2021 1:32:44 PM   3/15/2021 5:42:50 PM     17:47:49
Sales                         3 1/25/2021 1:32:44 PM   3/16/2021 9:52:29 AM     01:38:10
Marketing                     3 1/25/2021 1:32:44 PM   3/16/2021 9:52:29 AM     01:38:10
Accounting                    3 1/25/2021 1:32:44 PM   3/4/2021 9:25:39 AM   12.02:05:01
JEA Operators                 4 1/25/2021 1:32:44 PM   1/28/2021 11:34:57 AM 46.23:55:43
Web Servers                   1 1/25/2021 1:32:45 PM   3/15/2021 5:42:33 PM     17:48:07
DevOpsPrimary                 0 1/25/2021 4:47:53 PM   1/27/2021 10:35:11 AM 48.00:55:29
DevOpsBackup                  3 1/25/2021 4:48:02 PM   3/16/2021 10:12:01 AM    01:18:39
Payroll Managers              0 1/26/2021 10:12:34 AM  1/26/2021 10:12:34 AM 49.01:18:06
ThetaDL                       1 2/16/2021 8:32:36 AM   3/16/2021 9:43:32 AM     01:47:08
StrategyDL                    0 2/16/2021 9:03:12 AM   3/15/2021 5:45:07 PM     17:45:33
SecOpAdmin                    2 2/24/2021 12:37:28 PM  2/24/2021 12:39:15 PM 19.22:51:25
FocusOne                     16 2/24/2021 3:27:58 PM   3/16/2021 9:43:32 AM     01:47:08
SupportTech                   2 2/26/2021 6:12:51 PM   3/15/2021 5:43:03 PM     17:47:37
DL-Test                       4 3/3/2021 1:54:01 PM    3/16/2021 9:43:32 AM     01:47:08
DL-Test2                      1 3/3/2021 1:55:13 PM    3/3/2021 2:01:50 PM   12.21:28:50
```

Distribution groups will be shown in green and member counts of 0 in red. The Age reflects how long since the group has been modified.

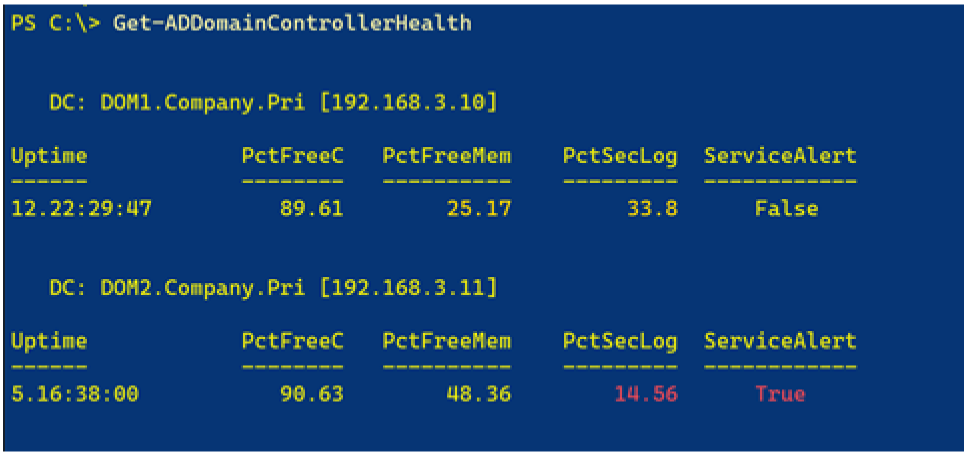
Computers

Get-ADDomainControllerHealth

`Get-ADDomainControllerHealth` is intended to give you a quick summary of the overall health of your Active Directory domain controllers. The concept of "health" is based on the following:

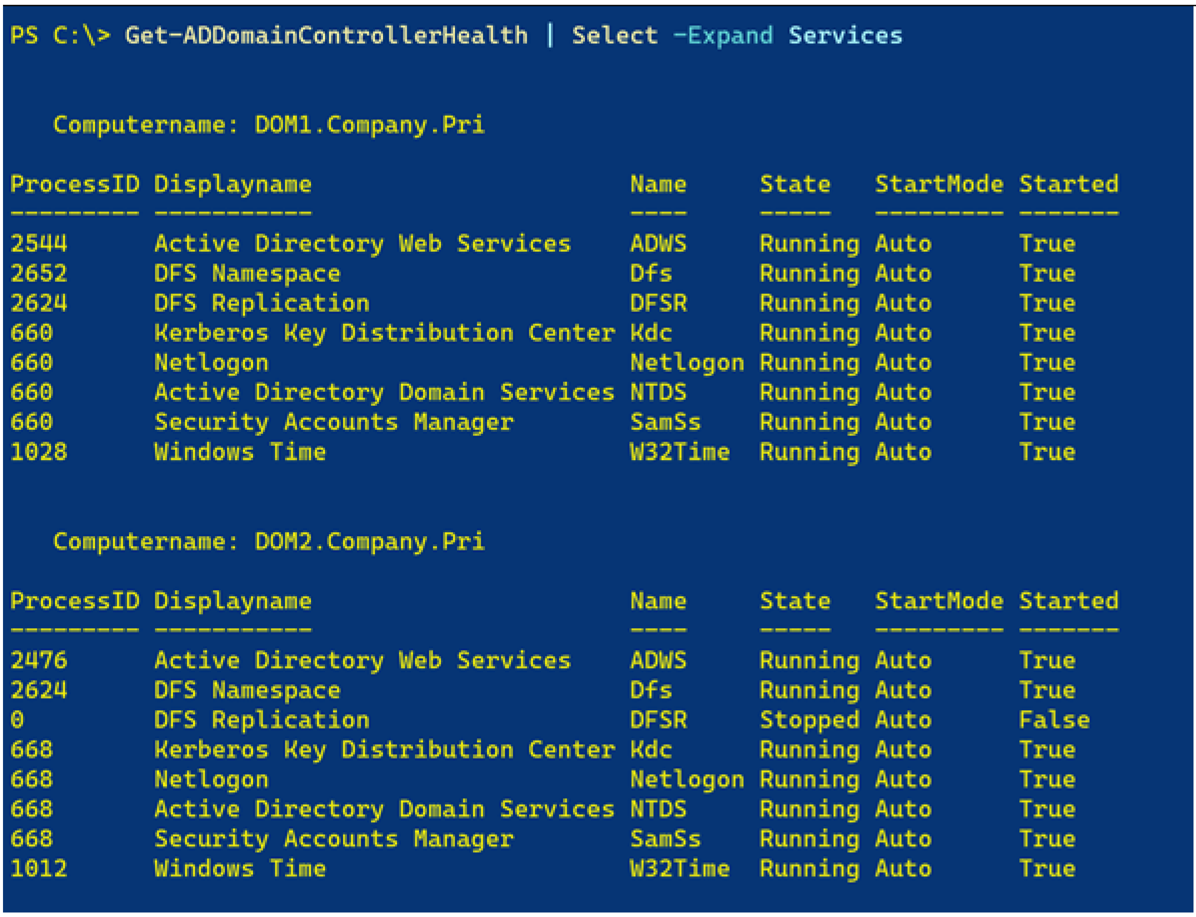
- How much free space remains on drive C:\?
- How much free physical memory?
- What percentage of the Security event log is in use?
- Are any critical services not running? The services checked are ntds,kdc,adws,dfs,dfsr,netlogon,samss, and w32time. Not every organization runs DNS and/or DHCP on their domain controllers, so those services have been omitted.

Output will be color-coded using ANSI escape sequences, if the PowerShell session supports it.



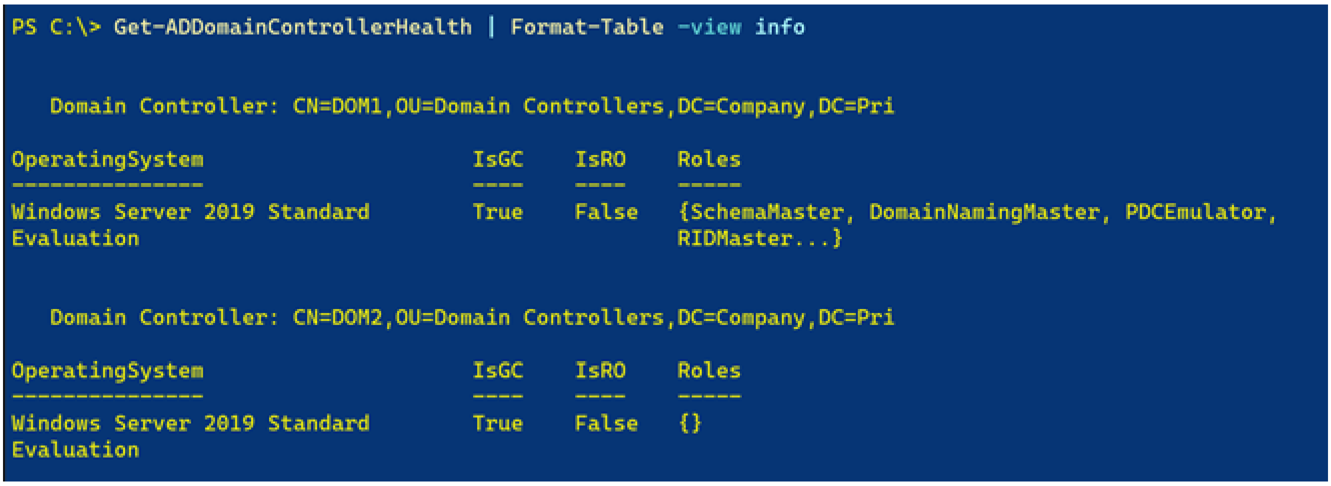
```
PS C:\> Get-ADDomainControllerHealth


   DC: DOM1.Company.Pri [192.168.3.10]

Uptime            PctFreeC   PctFreeMem    PctSecLog  ServiceAlert
------            --------   ----------    ---------  ------------
12.22:29:47          89.61        25.17         33.8         False


   DC: DOM2.Company.Pri [192.168.3.11]

Uptime            PctFreeC   PctFreeMem    PctSecLog  ServiceAlert
------            --------   ----------    ---------  ------------
5.16:38:00           90.63        48.36        14.56        True
```

The domain controller services are a nested object, but if you expand them, they have a defined and formatted view.



```
PS C:\> Get-ADDomainControllerHealth | Select -Expand Services


   Computername: DOM1.Company.Pri

ProcessID Displayname                       Name     State   StartMode Started
--------- -----------                       ----     -----   --------- -------
2544      Active Directory Web Services     ADWS     Running Auto      True
2652      DFS Namespace                     Dfs      Running Auto      True
2624      DFS Replication                   DFSR     Running Auto      True
660       Kerberos Key Distribution Center  Kdc      Running Auto      True
660       Netlogon                          Netlogon Running Auto      True
660       Active Directory Domain Services  NTDS     Running Auto      True
660       Security Accounts Manager         SamSs    Running Auto      True
1028      Windows Time                      W32Time  Running Auto      True


   Computername: DOM2.Company.Pri

ProcessID Displayname                       Name     State   StartMode Started
--------- -----------                       ----     -----   --------- -------
2476      Active Directory Web Services     ADWS     Running Auto      True
2624      DFS Namespace                     Dfs      Running Auto      True
0         DFS Replication                   DFSR     Stopped Auto      False
668       Kerberos Key Distribution Center  Kdc      Running Auto      True
668       Netlogon                          Netlogon Running Auto      True
668       Active Directory Domain Services  NTDS     Running Auto      True
668       Security Accounts Manager         SamSs    Running Auto      True
1012      Windows Time                      W32Time  Running Auto      True
```

You can use additional custom views to format the results.



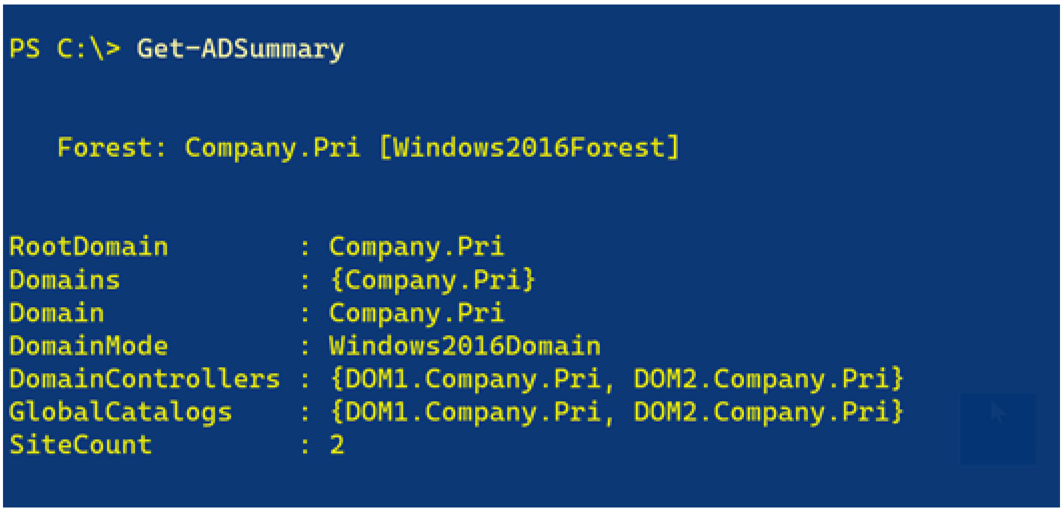
```
PS C:\> Get-ADDomainControllerHealth | Format-Table -view info


   Domain Controller: CN=DOM1,OU=Domain Controllers,DC=Company,DC=Pri

OperatingSystem                    IsGC    IsRO    Roles
---------------                    ----    ----    -----
Windows Server 2019 Standard       True    False   {SchemaMaster, DomainNamingMaster, PDCEmulator,
Evaluation                                         RIDMaster...}


   Domain Controller: CN=DOM2,OU=Domain Controllers,DC=Company,DC=Pri

OperatingSystem                    IsGC    IsRO    Roles
---------------                    ----    ----    -----
Windows Server 2019 Standard       True    False   {}
Evaluation
```

Reports

The primary goal for this module is reporting. The intention is to provide easy-to-use commands that will provide at least a snapshot view of information you might want to know.

Get-ADSummary

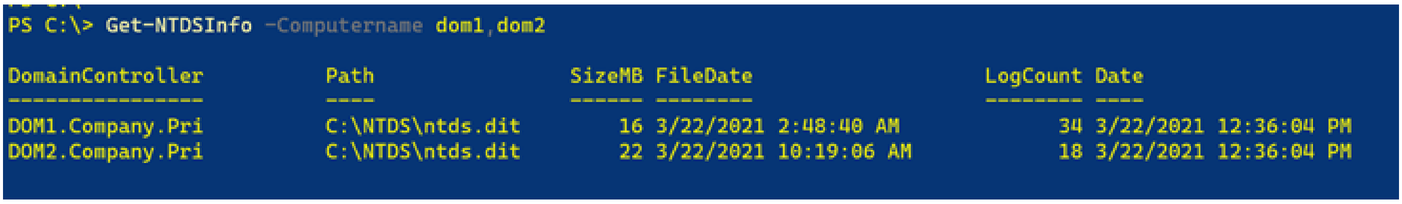
This simple command will give you a snapshot-sized summary of your Active Directory domain and forest.



```
PS C:\> Get-ADSummary


   Forest: Company.Pri [Windows2016Forest]


RootDomain        : Company.Pri
Domains           : {Company.Pri}
Domain            : Company.Pri
DomainMode        : Windows2016Domain
DomainControllers : {DOM1.Company.Pri, DOM2.Company.Pri}
GlobalCatalogs    : {DOM1.Company.Pri, DOM2.Company.Pri}
SiteCount         : 2
```

Get-NTDSInfo

`Get-NTDSInfo` will query a domain controller using PowerShell remoting to get information about the NTDS.dit and related files. You might use this to track the size of the file or to check on backups. A high log count might indicate a backup is needed.

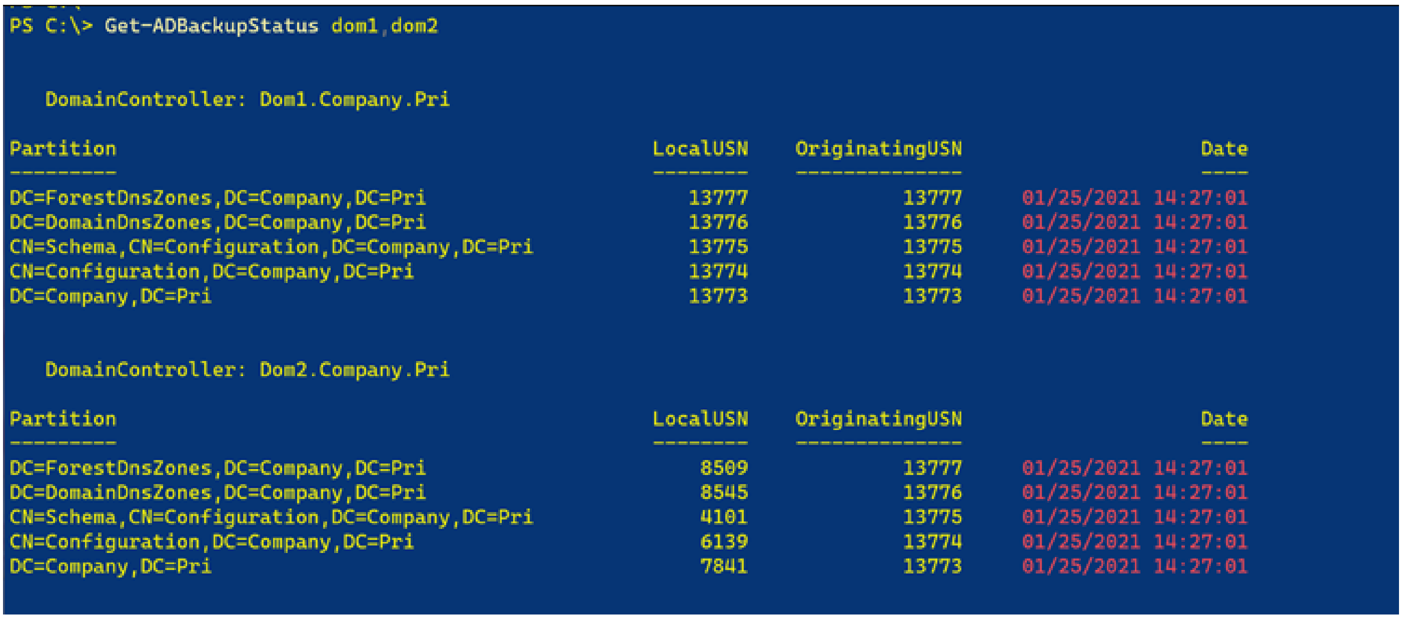


```
PS C:\> Get-NTDSInfo -Computername dom1,dom2

DomainController     Path             SizeMB FileDate               LogCount Date
----------------     ----             ------ --------               -------- ----
DOM1.Company.Pri     C:\NTDS\ntds.dit     16 3/22/2021 2:48:40 AM         34 3/22/2021 12:36:04 PM
DOM2.Company.Pri     C:\NTDS\ntds.dit     22 3/22/2021 10:19:06 AM        18 3/22/2021 12:36:04 PM
```

Get-ADBackupStatus

There aren't any explicit PowerShell commands to tell if Active Directory has been backed up. One indirect approach is to use the command-line tool `repadmin.exe`. This command has a `/showbackup` parameter which will indicate when the different Active Directory partitions have been backed up. This command is a PowerShell wrapper for `repadmin.exe` that runs on the specified domain controller in a PowerShell remoting session.

If running in a console host, the date value may be shown in red, if the date is beyond the backup limit of 3 days.



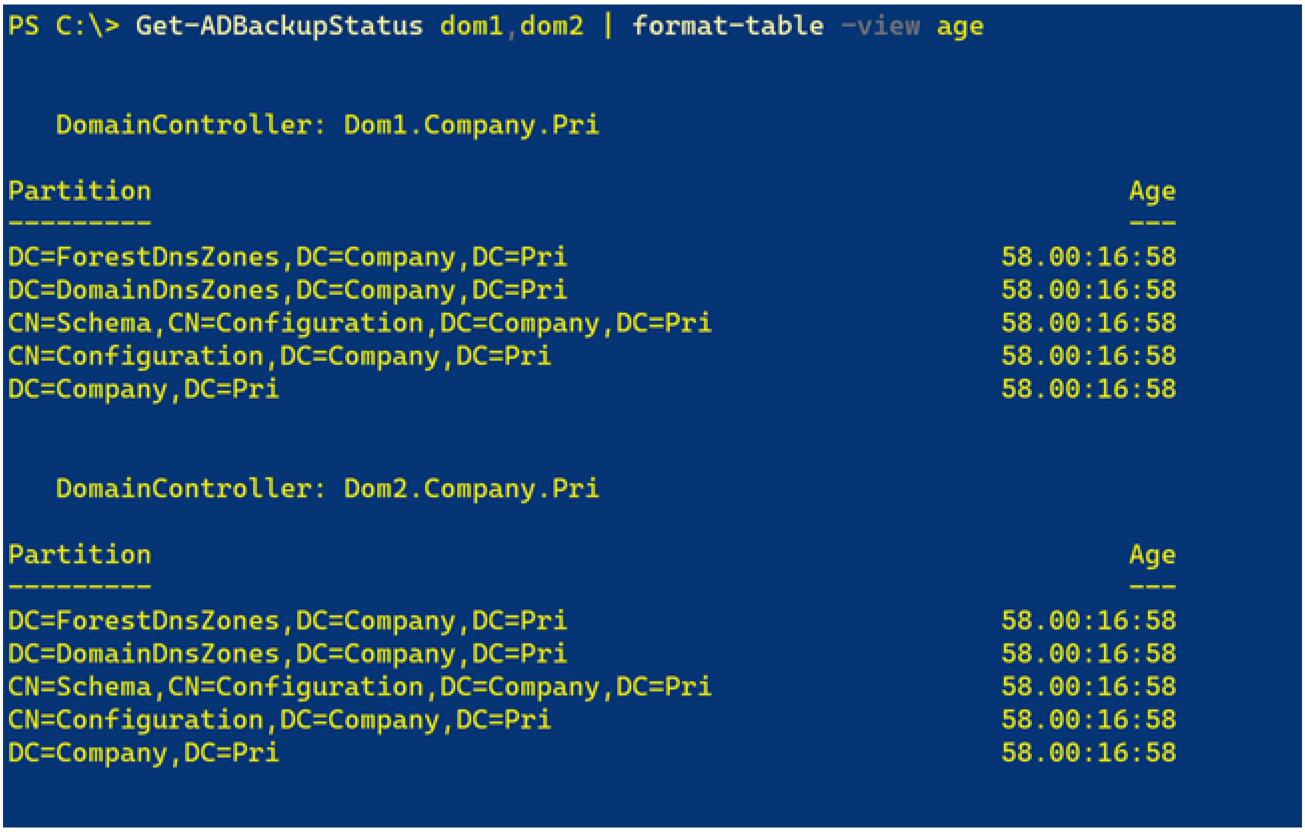
```
PS C:\> Get-ADBackupStatus dom1,dom2


   DomainController: Dom1.Company.Pri

Partition                                              LocalUSN    OriginatingUSN                 Date
---------                                              --------    --------------                 ----
DC=ForestDnsZones,DC=Company,DC=Pri                       13777             13777      01/25/2021 14:27:01
DC=DomainDnsZones,DC=Company,DC=Pri                       13776             13776      01/25/2021 14:27:01
CN=Schema,CN=Configuration,DC=Company,DC=Pri              13775             13775      01/25/2021 14:27:01
CN=Configuration,DC=Company,DC=Pri                        13774             13774      01/25/2021 14:27:01
DC=Company,DC=Pri                                         13773             13773      01/25/2021 14:27:01


   DomainController: Dom2.Company.Pri

Partition                                              LocalUSN    OriginatingUSN                 Date
---------                                              --------    --------------                 ----
DC=ForestDnsZones,DC=Company,DC=Pri                        8509             13777      01/25/2021 14:27:01
DC=DomainDnsZones,DC=Company,DC=Pri                        8545             13776      01/25/2021 14:27:01
CN=Schema,CN=Configuration,DC=Company,DC=Pri               4101             13775      01/25/2021 14:27:01
CN=Configuration,DC=Company,DC=Pri                         6139             13774      01/25/2021 14:27:01
DC=Company,DC=Pri                                          7841             13773      01/25/2021 14:27:01
```

The date limit is a user-customizable value in `$ADReportingHash`.

```
$ADReportinghash.BackupLimit = 5
```

If you want a limit like this all the time, in your PowerShell profile script, import the module and add this line.

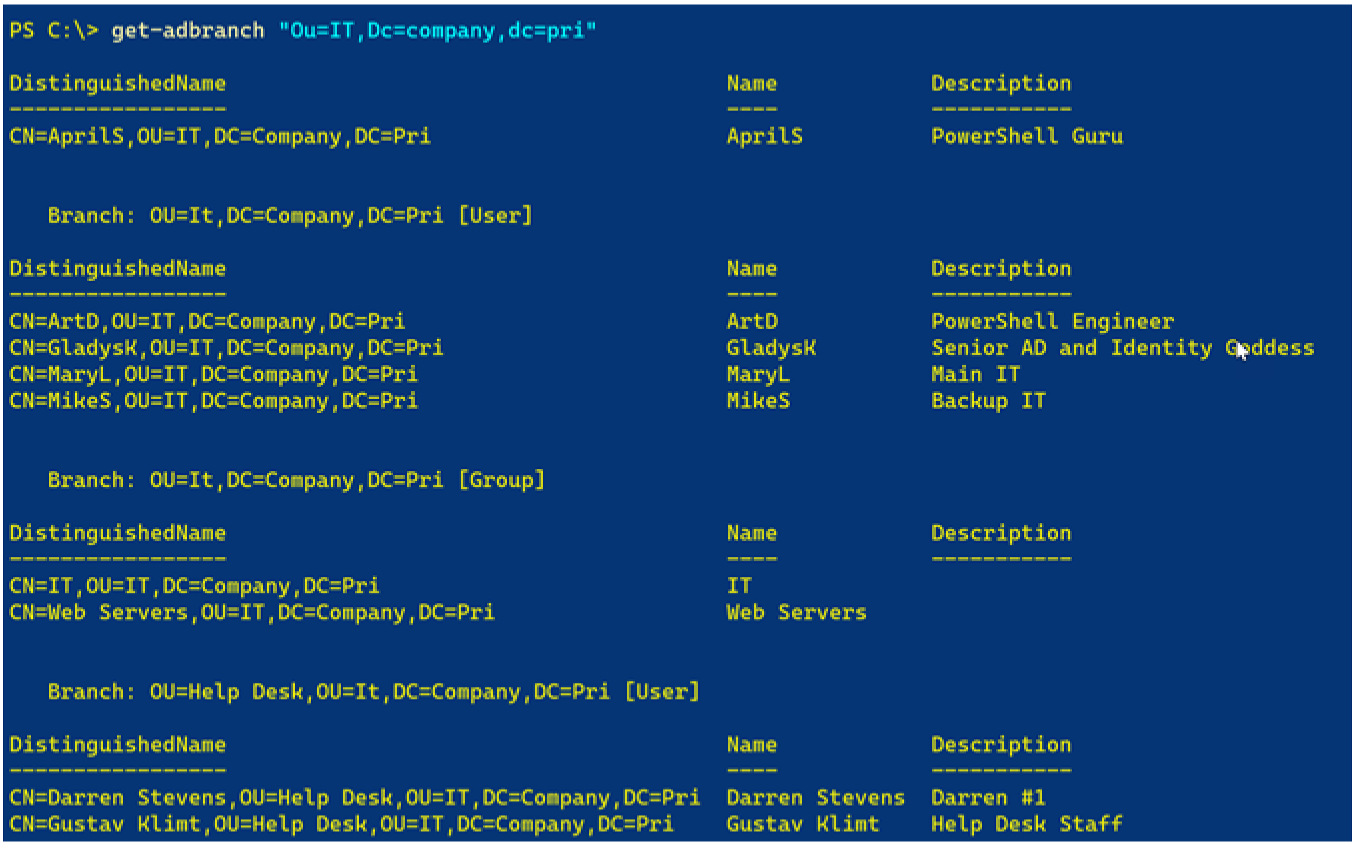
The command output also has a second formatted view.



```
PS C:\> Get-ADBackupStatus dom1,dom2 | format-table -view age


   DomainController: Dom1.Company.Pri

Partition                                              Age
---------                                              ---
DC=ForestDnsZones,DC=Company,DC=Pri            58.00:16:58
DC=DomainDnsZones,DC=Company,DC=Pri            58.00:16:58
CN=Schema,CN=Configuration,DC=Company,DC=Pri   58.00:16:58
CN=Configuration,DC=Company,DC=Pri             58.00:16:58
DC=Company,DC=Pri                              58.00:16:58


   DomainController: Dom2.Company.Pri

Partition                                              Age
---------                                              ---
DC=ForestDnsZones,DC=Company,DC=Pri            58.00:16:58
DC=DomainDnsZones,DC=Company,DC=Pri            58.00:16:58
CN=Schema,CN=Configuration,DC=Company,DC=Pri   58.00:16:58
CN=Configuration,DC=Company,DC=Pri             58.00:16:58
DC=Company,DC=Pri                              58.00:16:58
```

Get-ADBranch

`Get-ADBranch` will get all users, groups, and computers from a given Active Directory organizational unit or container and display a hierarchical report. The search is recursive from the starting search base. The output is grouped by organizational unit or container. Within each level, Active Directory objects are grouped by type, e.g. User.

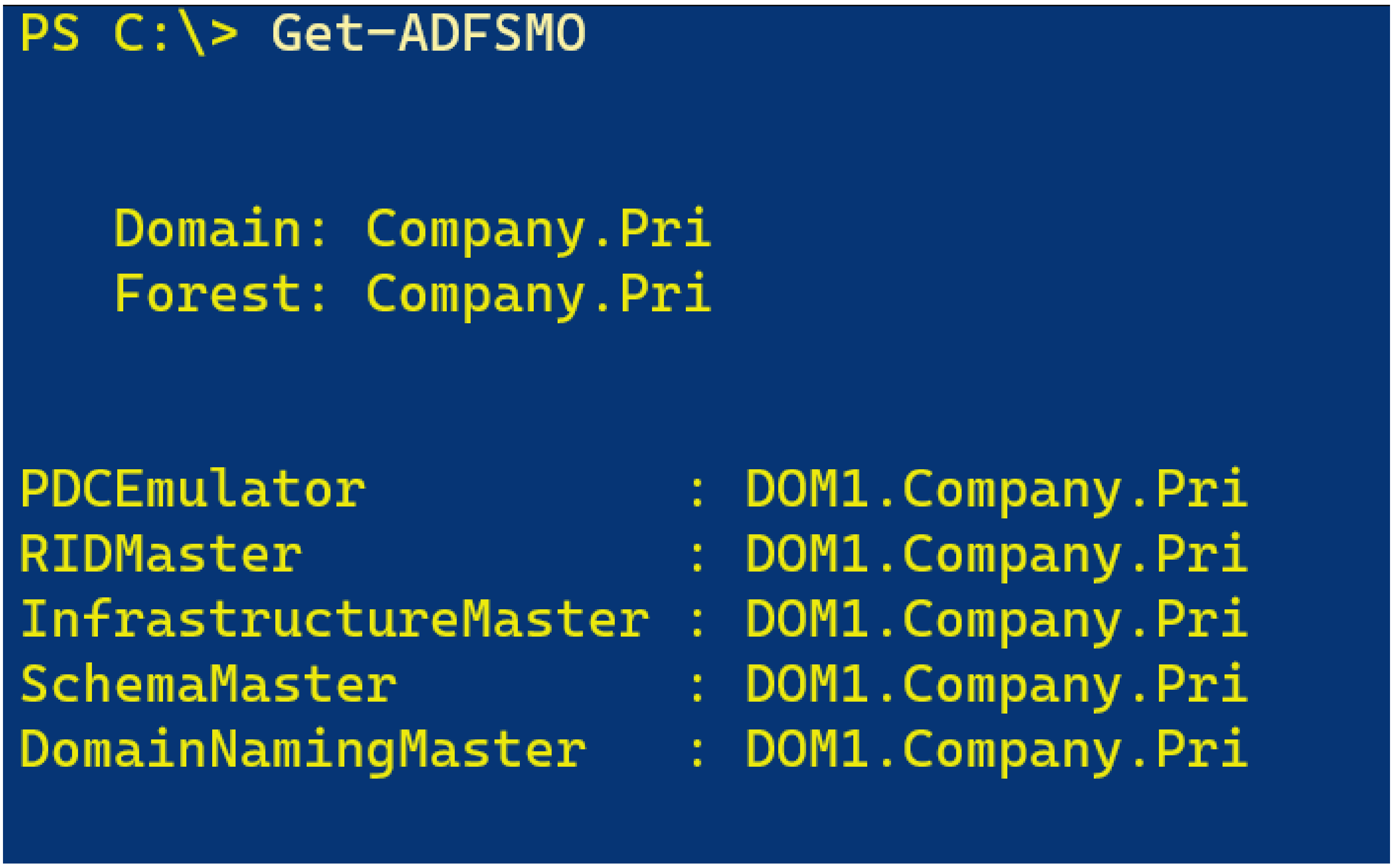


```
PS C:\> get-adbranch "Ou=IT,Dc=company,dc=pri"

DistinguishedName                                        Name           Description
-----------------                                        ----           -----------
CN=AprilS,OU=IT,DC=Company,DC=Pri                        AprilS         PowerShell Guru


   Branch: OU=It,DC=Company,DC=Pri [User]

DistinguishedName                                        Name           Description
-----------------                                        ----           -----------
CN=ArtD,OU=IT,DC=Company,DC=Pri                          ArtD           PowerShell Engineer
CN=GladysK,OU=IT,DC=Company,DC=Pri                       GladysK        Senior AD and Identity Goddess
CN=MaryL,OU=IT,DC=Company,DC=Pri                         MaryL          Main IT
CN=MikeS,OU=IT,DC=Company,DC=Pri                         MikeS          Backup IT


   Branch: OU=It,DC=Company,DC=Pri [Group]

DistinguishedName                                        Name           Description
-----------------                                        ----           -----------
CN=IT,OU=IT,DC=Company,DC=Pri                            IT
CN=Web Servers,OU=IT,DC=Company,DC=Pri                   Web Servers


   Branch: OU=Help Desk,OU=It,DC=Company,DC=Pri [User]

DistinguishedName                                        Name           Description
-----------------                                        ----           -----------
CN=Darren Stevens,OU=Help Desk,OU=IT,DC=Company,DC=Pri   Darren Stevens Darren #1
CN=Gustav Klimt,OU=Help Desk,OU=IT,DC=Company,DC=Pri     Gustav Klimt   Help Desk Staff
```

There is a formatting bug that prevents the first item from being properly grouped.

Get-ADFSMO

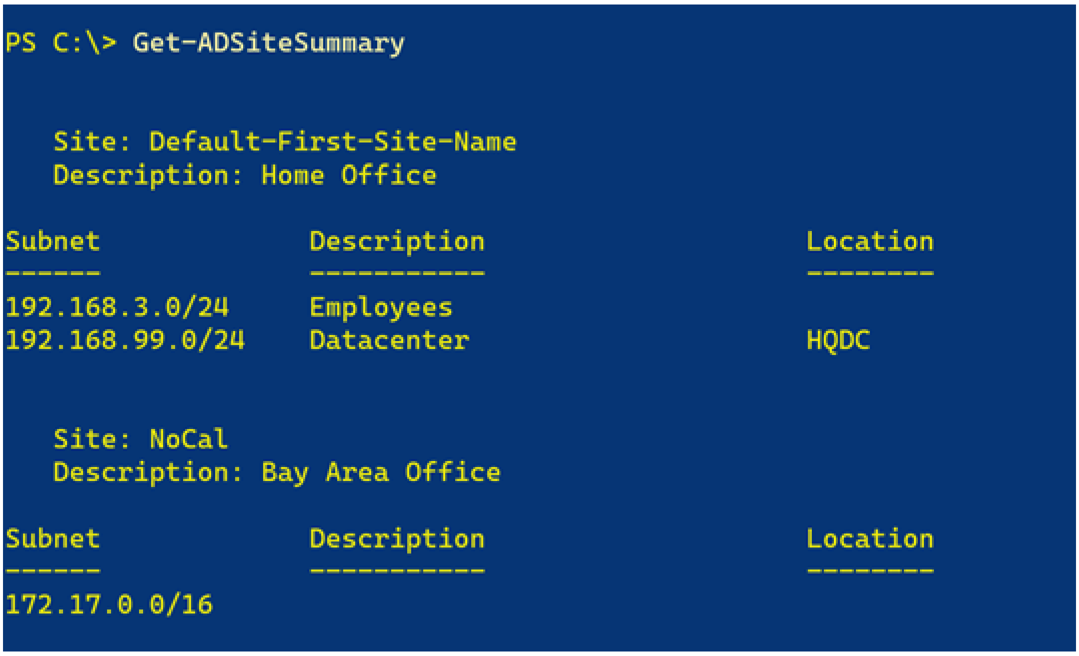
`Get-ADFSMO` will display all FSMO role holders for the forest and domain at a glance.



```
PS C:\> Get-ADFSMO


   Domain: Company.Pri
   Forest: Company.Pri


PDCEmulator          : DOM1.Company.Pri
RIDMaster            : DOM1.Company.Pri
InfrastructureMaster : DOM1.Company.Pri
SchemaMaster         : DOM1.Company.Pri
DomainNamingMaster   : DOM1.Company.Pri
```

Get-ADSiteSummary

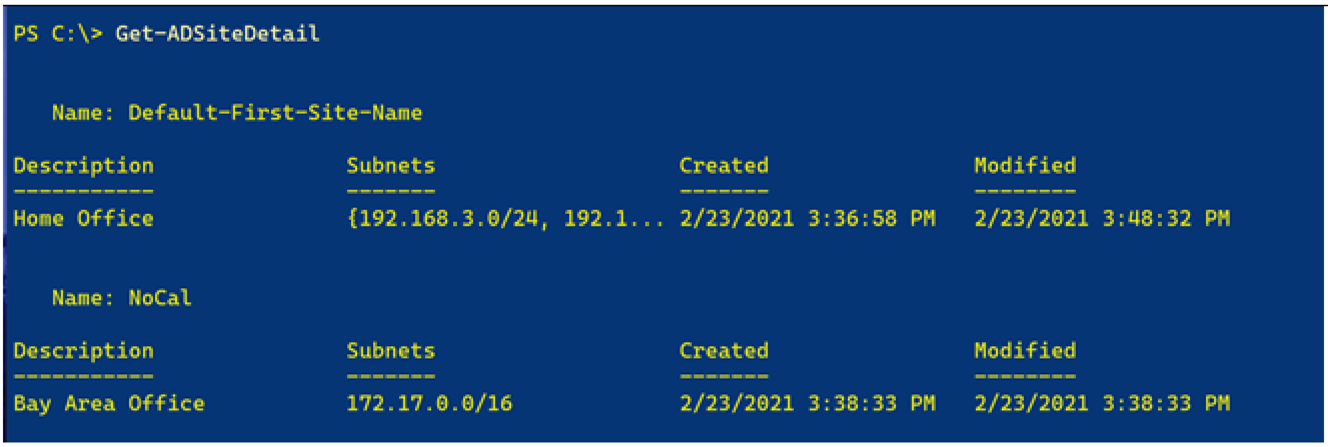
`Get-ADSiteSummary` presents a quick view of your sites and subnets.



```
PS C:\> Get-ADSiteSummary


   Site: Default-First-Site-Name
   Description: Home Office

Subnet             Description                    Location
------             -----------                    --------
192.168.3.0/24     Employees
192.168.99.0/24    Datacenter                     HQDC


   Site: NoCal
   Description: Bay Area Office

Subnet             Description                    Location
------             -----------                    --------
172.17.0.0/16
```

Get-ADSiteDetail

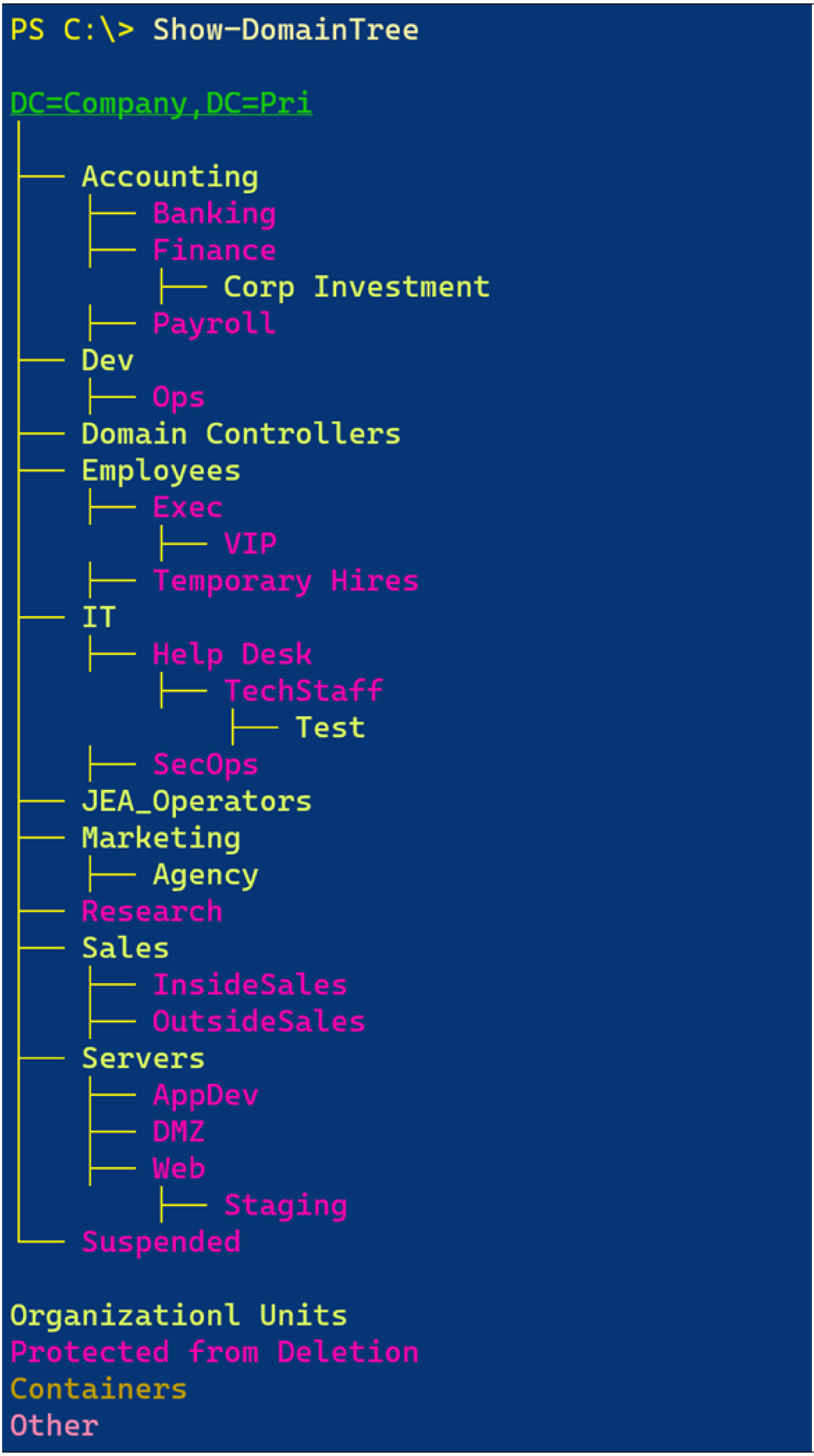
`Get-ADSiteDetail` will present a summary report of your Active Directory sites with a bit more detail. This command will show the site description, associated subnets, and when the site object was created and last modified. Information is displayed in a formatted table.



```
PS C:\> Get-ADSiteDetail


   Name: Default-First-Site-Name

Description               Subnets                   Created                Modified
-----------               -------                   -------                --------
Home Office               {192.168.3.0/24, 192.1... 2/23/2021 3:36:58 PM   2/23/2021 3:48:32 PM


   Name: NoCal

Description               Subnets                   Created                Modified
-----------               -------                   -------                --------
Bay Area Office           172.17.0.0/16             2/23/2021 3:38:33 PM   2/23/2021 3:38:33 PM
```

Show-DomainTree

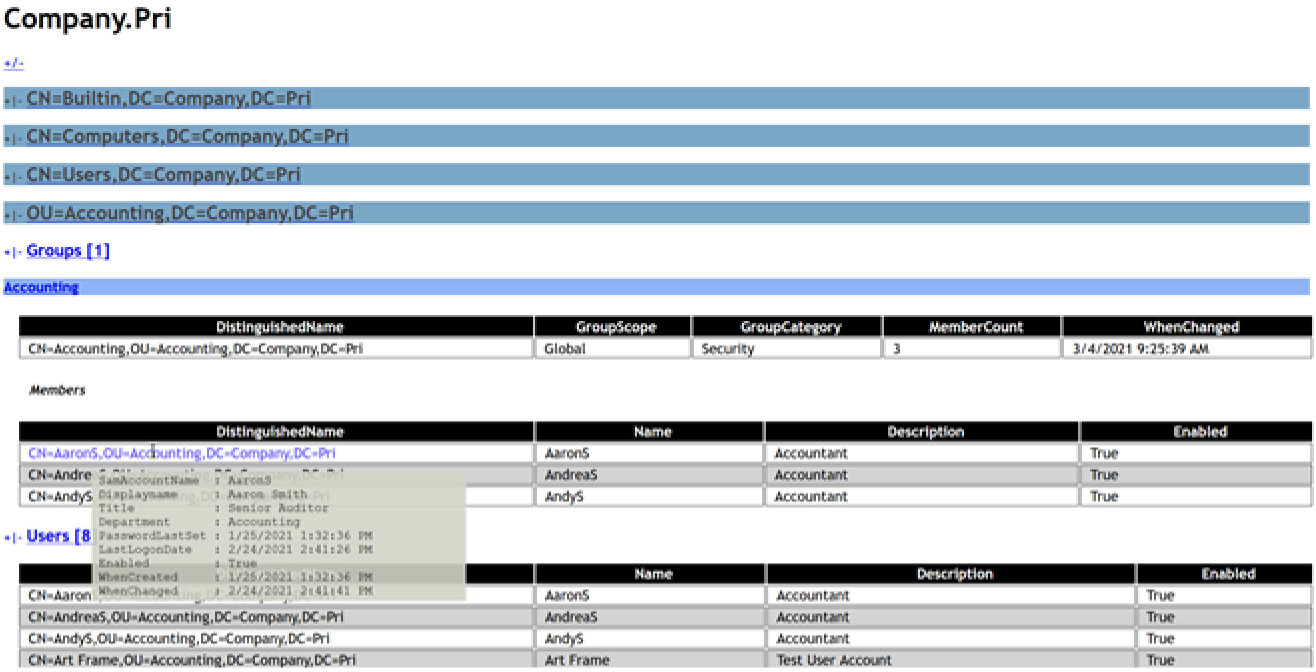
Show-DomainTree will display your domain in a tree view at the console. By default, the function will use color-coded ANSI formatting, assuming your PowerShell console supports it. The default display uses the organizational unit names. Although, you can use the distinguishedname of each branch. If you use -Containers, containers like Users will be included.



```
PS C:\> Show-DomainTree

DC=Company,DC=Pri
│
├── Accounting
│   ├── Banking
│   ├── Finance
│       ├── Corp Investment
│   ├── Payroll
├── Dev
│   ├── Ops
├── Domain Controllers
├── Employees
│   ├── Exec
│       ├── VIP
│   ├── Temporary Hires
├── IT
│   ├── Help Desk
│       ├── TechStaff
│           ├── Test
│   ├── SecOps
├── JEA_Operators
├── Marketing
│   ├── Agency
├── Research
├── Sales
│   ├── InsideSales
│   ├── OutsideSales
├── Servers
│   ├── AppDev
│   ├── DMZ
│   ├── Web
│       ├── Staging
└── Suspended

Organizationl Units
Protected from Deletion
Containers
Other
```

New-ADDomainReport

`New-ADDomainReport` will create an HTML report of your domain. The report layout is by container and organizational unit. Underneath each branch will be a table display of users, computers, and groups. Beneath each group will be table of recursive group members. You should get detail about users and computers if you hover the mouse over the distinguished name. The report includes javascript to enable collapsible regions.



Company.Pri

+/-

+|- CN=Builtin,DC=Company,DC=Pri

+|- CN=Computers,DC=Company,DC=Pri

+|- CN=Users,DC=Company,DC=Pri

+|- OU=Accounting,DC=Company,DC=Pri

+|- Groups [1]

Accounting

DistinguishedName	GroupScope	GroupCategory	MemberCount	WhenChanged
CN=Accounting,OU=Accounting,DC=Company,DC=Pri	Global	Security	3	3/4/2021 9:25:39 AM

Members

DistinguishedName	Name	Description	Enabled
CN=AaronS,OU=Accounting,DC=Company,DC=Pri	AaronS	Accountant	True
CN=Andre	AndreaS	Accountant	True
CN=AndyS	AndyS	Accountant	True

```
SamAccountName    : AaronS
Displayname       : Aaron Smith
Title             : Senior Auditor
Department        : Accounting
PasswordLastSet   : 1/25/2021 1:32:36 PM
LastLogonDate     : 2/24/2021 2:41:26 PM
Enabled           : True
WhenCreated       : 1/25/2021 1:32:36 PM
WhenChanged       : 2/24/2021 2:41:41 PM
```

+|- Users [8

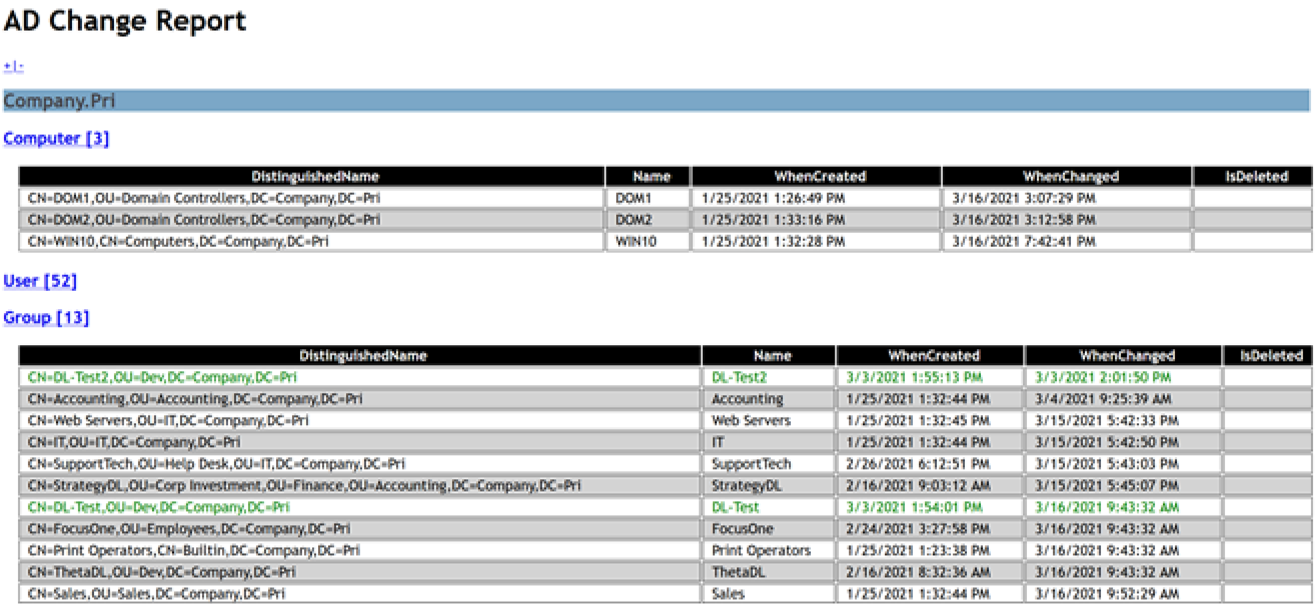
	Name	Description	Enabled
CN=Aaron	AaronS	Accountant	True
CN=AndreaS,OU=Accounting,DC=Company,DC=Pri	AndreaS	Accountant	True
CN=AndyS,OU=Accounting,DC=Company,DC=Pri	AndyS	Accountant	True
CN=Art Frame,OU=Accounting,DC=Company,DC=Pri	Art Frame	Test User Account	True

The ADReportingTools module includes a CSS file, which will be used by default. But you can specify an alternate CSS file. If you want to make the file portable, you can opt to embed the CSS into the HTML file. You can only embed from a file, not a URL reference.

The module's CSS file can be found in the reports folder. You can view a complete sample report here.

New-ADChangeReport

`New-ADChangeReport` will create an HTML report showing changes to Active Directory users, computers, and groups since a given date and time. The command uses `Get-ADObject` to query the `WhenChanged` property. The objects are organized by class and/or container and written to an HTML file. The command uses a CSS file from the ADReportingTools module, although you can specify your own. To make the HTML file portable, you can opt to embed the CSS content from a file source.



AD Change Report

+|-

Company.Pri

Computer [3]

DistinguishedName	Name	WhenCreated	WhenChanged	IsDeleted
CN=DOM1,OU=Domain Controllers,DC=Company,DC=Pri	DOM1	1/25/2021 1:26:49 PM	3/16/2021 3:07:29 PM	
CN=DOM2,OU=Domain Controllers,DC=Company,DC=Pri	DOM2	1/25/2021 1:33:16 PM	3/16/2021 3:12:58 PM	
CN=WIN10,CN=Computers,DC=Company,DC=Pri	WIN10	1/25/2021 1:32:28 PM	3/16/2021 7:42:41 PM	

User [52]

Group [13]

DistinguishedName	Name	WhenCreated	WhenChanged	IsDeleted
CN=DL-Test2,OU=Dev,DC=Company,DC=Pri	DL-Test2	3/3/2021 1:55:13 PM	3/3/2021 2:01:50 PM	
CN=Accounting,OU=Accounting,DC=Company,DC=Pri	Accounting	1/25/2021 1:32:44 PM	3/4/2021 9:25:39 AM	
CN=Web Servers,OU=IT,DC=Company,DC=Pri	Web Servers	1/25/2021 1:32:45 PM	3/15/2021 5:42:33 PM	
CN=IT,OU=IT,DC=Company,DC=Pri	IT	1/25/2021 1:32:44 PM	3/15/2021 5:42:50 PM	
CN=SupportTech,OU=Help Desk,OU=IT,DC=Company,DC=Pri	SupportTech	2/26/2021 6:12:51 PM	3/15/2021 5:43:03 PM	
CN=StrategyDL,OU=Corp Investment,OU=Finance,OU=Accounting,DC=Company,DC=Pri	StrategyDL	2/16/2021 9:03:12 AM	3/15/2021 5:45:07 PM	
CN=DL-Test,OU=Dev,DC=Company,DC=Pri	DL-Test	3/3/2021 1:54:01 PM	3/16/2021 9:43:32 AM	
CN=FocusOne,OU=Employees,DC=Company,DC=Pri	FocusOne	2/24/2021 3:27:58 PM	3/16/2021 9:43:32 AM	
CN=Print Operators,CN=Builtin,DC=Company,DC=Pri	Print Operators	1/25/2021 1:23:38 PM	3/16/2021 9:43:32 AM	
CN=ThetaDL,OU=Dev,DC=Company,DC=Pri	ThetaDL	2/16/2021 8:32:36 AM	3/16/2021 9:43:32 AM	
CN=Sales,OU=Sales,DC=Company,DC=Pri	Sales	1/25/2021 1:32:44 PM	3/16/2021 9:52:29 AM	

You can view the default CSS file here. A complete sample report can be found here.

Format and Type Extensions

The module includes format and type extensions to simplify using the commands in the Active Directory module. The extensions are automatically imported into your PowerShell session when you import the ADReportingTools module.

Currently, only AD User objects have been extended.

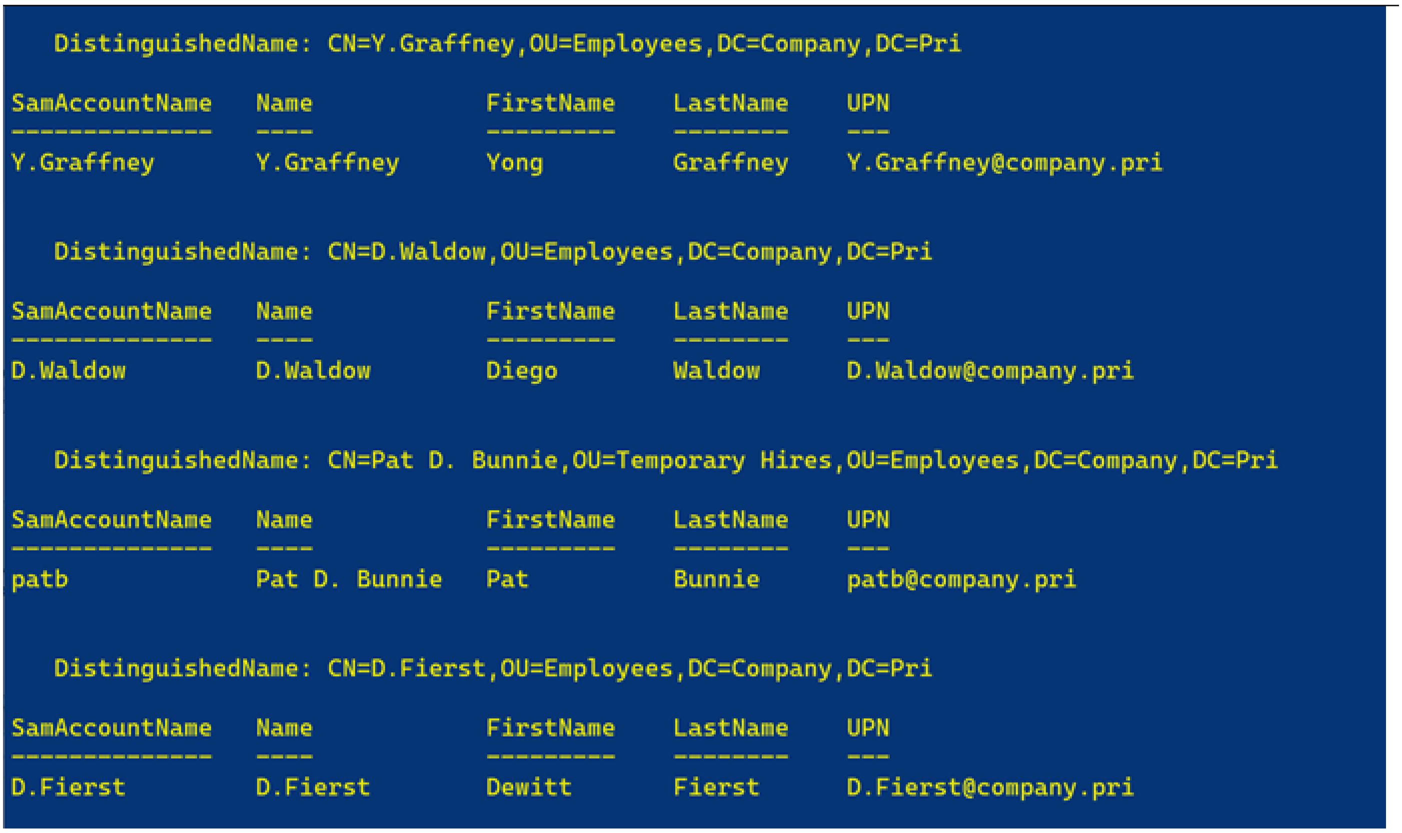
Name	Type	Value
LastName	AliasProperty	Surname
DN	AliasProperty	DistinguishedName
FirstName	AliasProperty	GivenName
UPN	AliasProperty	UserPrincipalName

These extensions have been grouped as a property set called *Names*.

```
PS C:\>Get-ADUser artd | Select-Object Names

DN             : CN=ArtD,OU=IT,DC=Company,DC=Pri
Name           : ArtD
FirstName      : Art
LastName       : Deco
SamAccountName : ArtD
UPN            : artd@company.com
```

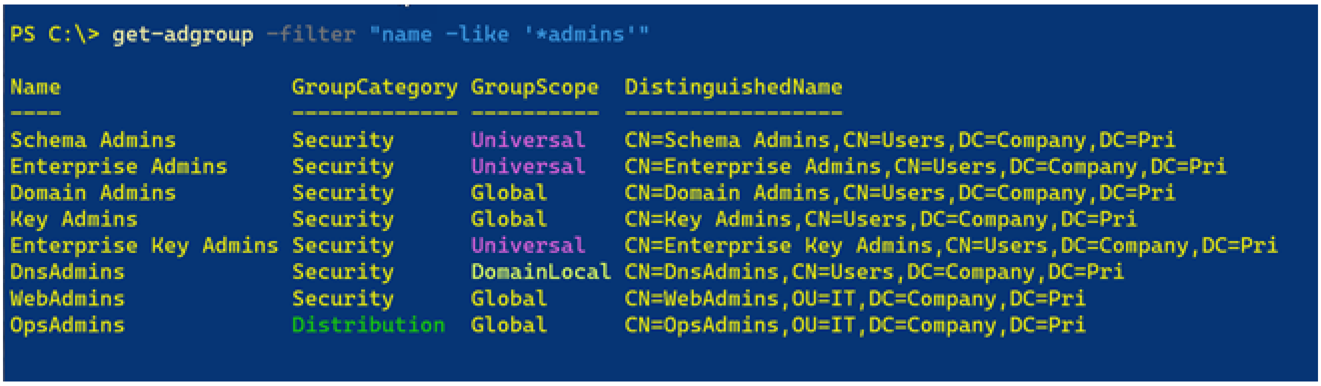
Or use a defined view for Active Directory user objects.

```
Get-ADUser -SearchBase "ou=employees,dc=company,dc=pri" -filter * |
Format-Table -view names
```



```
   DistinguishedName: CN=Y.Graffney,OU=Employees,DC=Company,DC=Pri

SamAccountName    Name            FirstName    LastName    UPN
--------------    ----            ---------    --------    ---
Y.Graffney        Y.Graffney      Yong         Graffney    Y.Graffney@company.pri


   DistinguishedName: CN=D.Waldow,OU=Employees,DC=Company,DC=Pri

SamAccountName    Name            FirstName    LastName    UPN
--------------    ----            ---------    --------    ---
D.Waldow          D.Waldow        Diego        Waldow      D.Waldow@company.pri


   DistinguishedName: CN=Pat D. Bunnie,OU=Temporary Hires,OU=Employees,DC=Company,DC=Pri

SamAccountName    Name            FirstName    LastName    UPN
--------------    ----            ---------    --------    ---
patb              Pat D. Bunnie   Pat          Bunnie      patb@company.pri


   DistinguishedName: CN=D.Fierst,OU=Employees,DC=Company,DC=Pri

SamAccountName    Name            FirstName    LastName    UPN
--------------    ----            ---------    --------    ---
D.Fierst          D.Fierst        Dewitt       Fierst      D.Fierst@company.pri
```

The module adds a default table view for AD group objects.

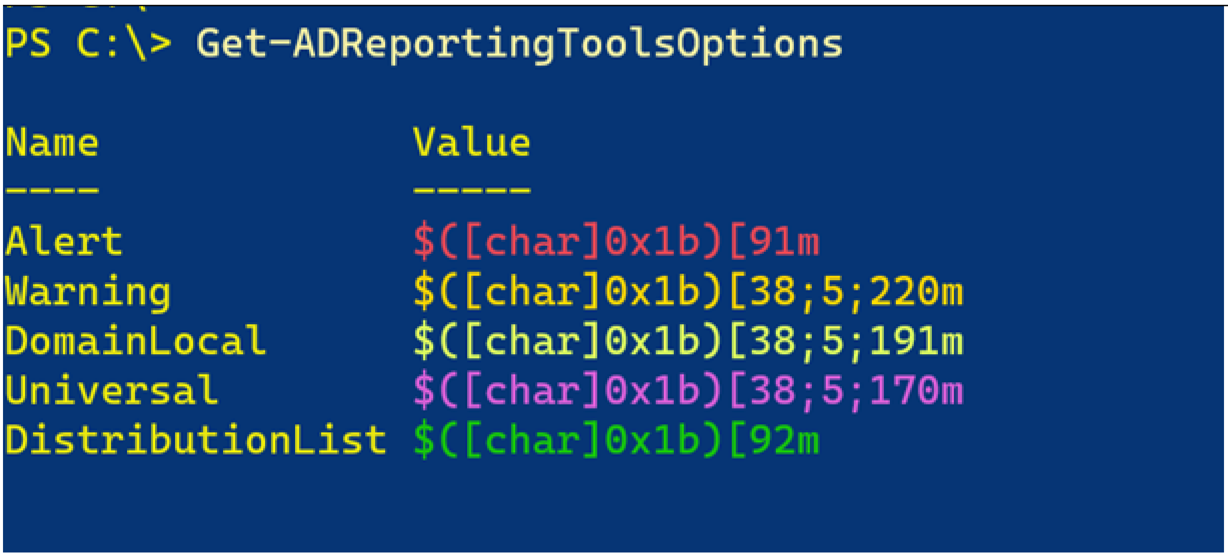


```
PS C:\> get-adgroup -filter "name -like '*admins'"

Name                  GroupCategory GroupScope  DistinguishedName
----                  ------------- ----------  -----------------
Schema Admins         Security      Universal   CN=Schema Admins,CN=Users,DC=Company,DC=Pri
Enterprise Admins     Security      Universal   CN=Enterprise Admins,CN=Users,DC=Company,DC=Pri
Domain Admins         Security      Global      CN=Domain Admins,CN=Users,DC=Company,DC=Pri
Key Admins            Security      Global      CN=Key Admins,CN=Users,DC=Company,DC=Pri
Enterprise Key Admins Security      Universal   CN=Enterprise Key Admins,CN=Users,DC=Company,DC=Pri
DnsAdmins             Security      DomainLocal CN=DnsAdmins,CN=Users,DC=Company,DC=Pri
WebAdmins             Security      Global      CN=WebAdmins,OU=IT,DC=Company,DC=Pri
OpsAdmins             Distribution  Global      CN=OpsAdmins,OU=IT,DC=Company,DC=Pri
```

If your PowerShell console supports it, Distribution, Universal, and DomainLocal groups will be highlighted in color.

ADReportingToolsOptions

The ANSI sequences used in the format files are user-configurable. Values are stored in an exported variable called `ADReportingToolsOptions`, although you shouldn't try to access the variable directly. Use `Get-ADReportingToolsOptions` to see the current values.



```
PS C:\> Get-ADReportingToolsOptions

Name             Value
----             -----
Alert            $([char]0x1b)[91m
Warning          $([char]0x1b)[38;5;220m
DomainLocal      $([char]0x1b)[38;5;191m
Universal        $([char]0x1b)[38;5;170m
DistributionList $([char]0x1b)[92m
```

The module uses the `[char]0x1b` escape sequence because it works in both Windows PowerShell and PowerShell 7.x.

If you prefer to customize the sequence, use `Set-ADReportingToolsOptions`.

```
Set-ADReportingToolsOptions DistributionList -ANSI "$([char]0x1b)[38;5;50m"
```

This change is only for the duration of your PowerShell session. Add the command to a PowerShell profile script to make it more permanent.

Future Work

These are items I'm considering adding to the module:

- Get-ADPasswordPending (look at Get-ADUserResultantPasswordPolicy)
- Store `ADReportingToolsOptions` in a JSON file.

Magical Thinking

These are items that I'm dreaming about:

- a toolset to build HTML reports on the fly
- a WPF based OU browser or a simplified version of ADUC

I welcome suggestions, feedback, and comments in the module repository's Discussion section.

last updated *2021-03-29 15:29:03Z*

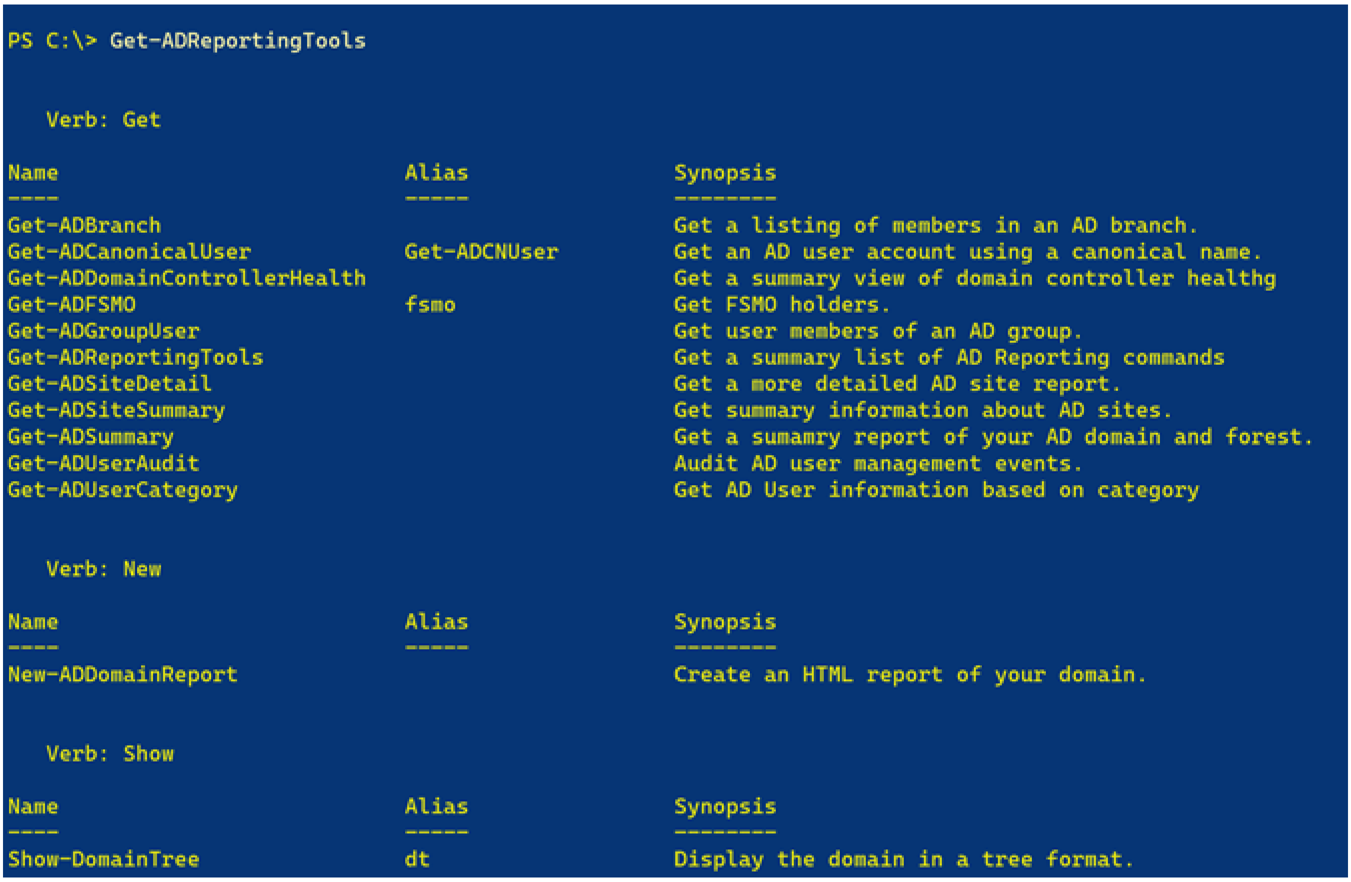
Module Functions

This section contains the same help content you would get from a PowerShell prompt using `Get-Help`. Note that most code examples have been formatted to fit the 80 character page width and sometimes with artificial formatting. Don't assume you can run examples *exactly* as they are shown. Some of the help examples might also use special or custom characters that might not render properly in the PDF.

If you can't remember what commands are in this module, you can always ask PowerShell.

```
Get-Command -module ADReportingTools
```

Or use the `Get-PSScriptTools` command.



```
PS C:\> Get-ADReportingTools


   Verb: Get

Name                           Alias                Synopsis
----                           -----                --------
Get-ADBranch                                        Get a listing of members in an AD branch.
Get-ADCanonicalUser            Get-ADCNUser         Get an AD user account using a canonical name.
Get-ADDomainControllerHealth                        Get a summary view of domain controller healthg
Get-ADFSMO                     fsmo                 Get FSMO holders.
Get-ADGroupUser                                     Get user members of an AD group.
Get-ADReportingTools                                Get a summary list of AD Reporting commands
Get-ADSiteDetail                                    Get a more detailed AD site report.
Get-ADSiteSummary                                   Get summary information about AD sites.
Get-ADSummary                                       Get a sumamry report of your AD domain and forest.
Get-ADUserAudit                                     Audit AD user management events.
Get-ADUserCategory                                  Get AD User information based on category


   Verb: New

Name                           Alias                Synopsis
----                           -----                --------
New-ADDomainReport                                  Create an HTML report of your domain.


   Verb: Show

Name                           Alias                Synopsis
----                           -----                --------
Show-DomainTree                dt                   Display the domain in a tree format.
```

Get-ADBackupStatus

Synopsis

Get an Active Directory backup status

Syntax

```
Get-ADBackupStatus [-DomainController] <String[]> [-Credential <PSCredential>] [<CommonParameters>]
```

Description

There aren’t any explicit PowerShell commands to tell if Active Directory has been backed up. One indirect approach is to use the command-line tool repadmin.exe. This command has a /showbackup parameter which will indicate when the different Active Directory partitions have been backed up. This command is a PowerShell wrapper for repadmin.exe that runs on the specified domain controller in a PowerShell remoting session.

If running in a console host, the date value may be shown in red, if the date is beyond the backup limit of 3 days. This is a user-customizable value in $ADReportingHash.

$ADReportinghash.BackupLimit = 5

If you want a limit like this all the time, in your PowerShell profile script import the module and add this line.

Examples

Example 1

```
PS C:\> Get-ADBackupStatus dom1

   DomainController: Dom1.Company.Pri

Partition                              LocalUSN OriginUSN                Date
---------                              -------- ---------                ----
DC=ForestDnsZones,DC=Company,DC=Pri       13777     13777   01/25/2021 14:27:01
DC=DomainDnsZones,DC=Company,DC=Pri       13776     13776   01/25/2021 14:27:01
CN=Schema,CN=Configuration,DC=Comp....    13775     13775   01/25/2021 14:27:01
CN=Configuration,DC=Company,DC=Pri        13774     13774   01/25/2021 14:27:01
DC=Company,DC=Pri                         13773     13773   01/25/2021 14:27:01
```

Any date that is beyond the number of days that is beyond $ADReportingHash.BackupLimit, will be displaySed in red, if running in a console host.

Parameters

-Credential

Specify an alternate credential

```
Type: PSCredential
Parameter Sets: (All)
Aliases:

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

-DomainController

Specify the name of a domain controller

```
Type: String[]
Parameter Sets: (All)
Aliases:

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

System.Object

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-NTDSInfo

repadmin.exe

Get-ADBranch

Synopsis

Get a listing of members in an AD branch.

Syntax

```
Get-ADBranch [-SearchBase] <String> [-ObjectClass <String[]>] [-IncludeDeletedObjects] [-ExcludeContainers] [-Server <String>]
[-Credential <PSCredential>] [<CommonParameters>]
```

Description

This command will get all users, groups, and computers from a given Active Directory organizational unit or container and display a hierarchical report. The search is recursive from the starting search base.

Examples

Example 1

```
PS C:\> Get-ADBranch "OU=IT,DC=company,DC=pri"

DistinguishedName                            Name           Description
-----------------                            ----           -----------
CN=AprilS,OU=IT,DC=Company,DC=Pri            AprilS         PowerShell Guru


   Branch: OU=It,DC=Company,DC=Pri [User]

DistinguishedName                            Name           Description
-----------------                            ----           -----------
CN=ArtD,OU=IT,DC=Company,DC=Pri              ArtD           PowerShell Engineer
CN=GladysK,OU=IT,DC=Company,DC=Pri           GladysK        Senior AD and Ide...
CN=MaryL,OU=IT,DC=Company,DC=Pri             MaryL          Main IT
CN=MikeS,OU=IT,DC=Company,DC=Pri             MikeS          Backup IT


   Branch: OU=It,DC=Company,DC=Pri [Group]

DistinguishedName                            Name           Description
-----------------                            ----           -----------
CN=IT,OU=IT,DC=Company,DC=Pri                IT
CN=Web Servers,OU=IT,DC=Company,DC=Pri       Web Servers
...
```

Get members of the IT organizational unit. There is a formatting bug where the first item isn’t properly grouped.

Example 2

```
PS C:\> Get-ADBranch "Ou=accounting,Dc=company,dc=pri" -objectclass group

DistinguishedName                           Name            Description
-----------------                           ----            -----------
CN=Accounting,OU=Accounting,              Accounting        Company Accounting DC=Company,DC=Pri


   Branch: OU=Corp Investment,OU=Finance,OU=Accounting,DC=Company,DC=Pri [Group]

DistinguishedName                           Name            Description
-----------------                           ----            -----------
CN=StrategyDL,OU=Corp                       StrategyDL      Strategic plann... Investment,OU=Finance,OU=Accounting,
DC=Company,DC=Pri


   Branch: OU=Payroll,OU=Accounting,DC=Company,DC=Pri [Group]

DistinguishedName                           Name            Description
-----------------                           ----            -----------
CN=Payroll Managers,OU=Payroll,             Payroll Managers
OU=Accounting,DC=Company,DC=Pri
```

Get only groups in the Accounting OU tree.

Parameters

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

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

-IncludeDeletedObjects

Show deleted objects. This parameter has no effect unless you are searching from the domain root.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

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

-SearchBase

Enter the distinguished name of the top-level container or organizational unit.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

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

-ExcludeContainers

Exclude containers like USERS. This will only have no effect unless your search base is the domain root.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

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

-ObjectClass

Only show objects of the matching classes. Valid choices are user, group, and computer.

```
Type: String[]
Parameter Sets: (All)
Aliases:

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADBranchMember

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Show-Domain

Get-ADCanonicalUser

Synopsis

Get an AD user account using a canonical name.

Syntax

```
Get-ADCanonicalUser [-Name] <String> [-Properties <String[]>] [-IncludeDeletedObjects] [-Server <String>] [-Credential <PSCredential>] [<
CommonParameters>]
```

Description

Often you will find user names in the form domain\username. This command makes it easier to find the Active Directory user account using this value. If you have enabled the Active Directory Recycle Bin feature, you can use the IncludeDeletedObjects parameter to search for the user account if it can’t be found with the initial search.

There is an assumption that you will know the domain controller responsible for the given domain component. Or that all accounts are in your current user domain.

Examples

Example 1

```
PS C:\> Get-ADCanonicalUser company\gladysk -Properties title,description,department


Department        : IT
Description       : Senior AD and Identity Goddess
DistinguishedName : CN=GladysK,OU=IT,DC=Company,DC=Pri
Enabled           : True
GivenName         : Gladys
Name              : GladysK
ObjectClass       : user
ObjectGUID        : 445c8817-3c53-4861-9221-407b5af8bdc6
SamAccountName    : GladysK
SID               : S-1-5-21-493037332-564925384-1585924867-1105
Surname           : Kravitz
Title             : AD Operations Lead
UserPrincipalName : gladysk@Company.Pri
```

Get the Active Directory user account for Company\Gladysk and some select properties.

Example 2

```
PS C:\> $a = Get-ADUserAudit -Since "2/1/2021" -Events Disabled
PS C:\> $a.targets | Get-Unique | Get-ADCanonicalUser |
Select-Object DistinguishedName

DistinguishedName
-----------------
CN=MaryL,OU=IT,DC=Company,DC=Pri
CN=E.Ratti,OU=Employees,DC=Company,DC=Pri
CN=Roy Biv,OU=Accounting,DC=Company,DC=Pri
CN=D.Monroy,OU=Employees,DC=Company,DC=Pri
CN=MaryL,OU=IT,DC=Company,DC=Pri
CN=S.Montbriand,OU=Employees,DC=Company,DC=Pri
CN=R.Freil,OU=Employees,DC=Company,DC=Pri
CN=N.Wobser,OU=Employees,DC=Company,DC=Pri
CN=Y.Graffney,OU=Employees,DC=Company,DC=Pri
CN=D.Waldow,OU=Employees,DC=Company,DC=Pri
```

The first command is using the Get-ADUserAudit command to find all user accounts disabled since February 1. The resulting targets in the canonical name format. These values are piped to Get-ADCanonicalUser to retrieve the corresponding distinguished name values.

Parameters

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

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

-IncludeDeletedObjects

Search deleted objects if the user account can’t be found.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

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

-Name

Enter the username in the form domain\username.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Properties

Enter one or more user properties or * to select everything.

```
Type: String[]
Parameter Sets: (All)
Aliases:

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

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

System.String

Outputs

Microsoft.ActiveDirectory.Management.ADUser

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADUser

Get-ADObject

Get-ADDepartment

Synopsis

Get members of a department from Active Directory.

Syntax

```
Get-ADDepartment [-Department] <String[]> [-Server <String>] [-Credential <PSCredential>] [<CommonParameters>]
```

Description

Use this command to retrieve user account information from Active Directory for members of a specific department. You can specify multiple departments. User information is displayed in a grouped table by default.

When you import the ADReportingTools module, it will define a global variable called ADReportingHash, which is a hashtable. The variable has a key called Departments. This variable is used in an argument completer for the -Department parameter. This allows you to tab-complete the parameter value. If you add a department after loading the module, you will need to update the variable. You can manually add a department:

$ADReportingHash.Departments+='Bottle Washing'

Or reload the module:

Import-Module ADReportingTools -force

Examples

Example 1

```
PS C:\> Get-ADDepartment -Department sales -Server dom1 -Credential company\artd

   Department: Sales

Name                      Title                         City                Phone
----                      -----                         ----                -----
Sonya Smith               Account Executive             Omaha               x2345
Garret Guillary           Intern                        Omaha               x8877
Sam Smith                 Sales Support                 Omaha               x5678
Samantha Smith            Sales Assistant               Omaha               x9875
```

Get all members of the Sales department. This example queries a specific domain controller and uses alternate credentials. If your PowerShell session supports it, disabled accounts will be displayed in red.

Example 2

```
PS C:\> Get-ADDepartment Sales | Format-Table -view manager


   Manager: CN=Alfonso Dente,OU=Sales,DC=Company,DC=Pri [Sales]

Name                 Description               Title                City
----                 -----------               -----                ----
Sonya Smith          Sales                     Account Executive    Omaha


   Manager: CN=SamanthaS,OU=Sales,DC=Company,DC=Pri [Sales]

Name                 Description               Title                City
----                 -----------               -----                ----
Garret Guillary      sales intern              Intern               Omaha


   Manager: CN=SonyaS,OU=Sales,DC=Company,DC=Pri [Sales]

Name                 Description               Title                City
----                 -----------               -----                ----
Sam Smith            Sales                     Sales Support        Omaha
Samantha Smith       Sales                     Sales Assistant      Omaha
```

The command has a corresponding formatting file with a custom view.

Parameters

-Credential

Specify alternate credentials for authentication.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: runas

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

-Department

Specify one or more department names.

```
Type: String[]
Parameter Sets: (All)
Aliases:

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

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: DC

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADDeptMember

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADUserCategory

Get-ADUser

Get-ADDomainControllerHealth

Synopsis

Get a summary view of domain controller health.

Syntax

```
Get-ADDomainControllerHealth [[-Server] <String>] [[-Credential] <PSCredential>] [<CommonParameters>]
```

Description

This command is intended to give you a quick summary of the overall health of your Active Directory domain controllers. The concept of "health" is based on the following:

- How much free space remains on drive C:?
- How much free physical memory?
- What percentage of the Security event log is in use?
- Are any critical services not running?

The services checked are ntds,kdc,adws,dfs,dfsr,netlogon,samss, and w32time. Not every organization runs DNS and/or DHCP on their domain controllers so those services have been omitted.

Output will be color-coded using ANSI escape sequences.

Examples

Example 1

```
PS C:\> Get-ADDomainControllerHealth


   DC: DOM1.Company.Pri [192.168.3.10]

Uptime            PctFreeC   PctFreeMem    PctSecLog  ServiceAlert
------            --------   ----------    ---------  ------------
12.22:29:47          89.61        25.17         33.8      False


   DC: DOM2.Company.Pri [192.168.3.11]

Uptime            PctFreeC   PctFreeMem    PctSecLog  ServiceAlert
------            --------   ----------    ---------  ------------
5.16:38:00           90.63        48.36        14.56      True
```

Get a health snapshot of your domain controllers. A ServiceAlert of True means that one of the defined critical services is not running.

Output might be color-coded. A ServiceAlert value of True will be displayed in Red. Free space on C and percent free physical memory will be shown in red if the value is 10% or less. A percent free less than 30$ will be displayed in an orange/yellow color. The percent Security log usage threshholds are 15% and 50%.

Example 2

```
PS C:\> Get-ADDomainControllerHealth | Format-Table -view info


   Domain Controller: CN=DOM1,OU=Domain Controllers,DC=Company,DC=Pri

OperatingSystem                    IsGC    IsRO    Roles
---------------                    ----    ----    -----
Windows Server 2019 Standard       True    False   {SchemaMaster,DomainNam...


   Domain Controller: CN=DOM2,OU=Domain Controllers,DC=Company,DC=Pri

OperatingSystem                    IsGC    IsRO    Roles
---------------                    ----    ----    -----
Windows Server 2019 Standard       True    False   {}
```

Get domain controller health using a custom table view.

Example 3

```
PS C:\> Get-ADDomainControllerHealth | Select-Object -Expand Services


   Computername: DOM1.Company.Pri

ProcessID Displayname                       Name     State   StartMode Started
--------- -----------                       ----     -----   --------- -------
2544      Active Directory Web Services     ADWS     Running Auto      True
2652      DFS Namespace                     Dfs      Running Auto      True
2624      DFS Replication                   DFSR     Running Auto      True
660       Kerberos Key Distribution Center Kdc      Running Auto      True
660       Netlogon                          Netlogon Running Auto      True
660       Active Directory Domain Services NTDS     Running Auto      True
660       Security Accounts Manager         SamSs    Running Auto      True
1028      Windows Time                      W32Time  Running Auto      True
...
```

View the service status for each domain controller.

Parameters

-Credential

Specify an alternate credential. This will be used to query the domain and all domain controllers.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Server

Specify a domain controller to query for a list of domain controllers.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADDomainControllerHealth

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADDomainController

Get-ADFSMO

Synopsis

Get FSMO holders.

Syntax

```
Get-ADFSMO [[-Identity] <String>] [-Server <String>] [-Credential <PSCredential>] [<CommonParameters>]
```

Description

This command will display all FSMO role holders for the forest and domain at a glance.

Examples

Example 1

```
PS C:\> PS C:\> Get-ADFSMO


   Domain: Company.Pri
   Forest: Company.Pri


PDCEmulator          : DOM1.Company.Pri
RIDMaster            : DOM1.Company.Pri
InfrastructureMaster : DOM1.Company.Pri
SchemaMaster         : DOM1.Company.Pri
DomainNamingMaster   : DOM1.Company.Pri
```

Get the FSMO holders for the current domain and forest.

Parameters

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

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

-Identity

Specify the domain name. The default is the user domain.

```
Type: String
Parameter Sets: (All)
Aliases: name

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADFSMORole

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADSummary

Get-ADDomain

Get-ADForest

Get-ADGroupReport

Synopsis

Create a custom group report

Syntax

```
Get-ADGroupReport [[-Name] <String>] [-SearchBase <String>][-Category <String>]
[-Scope <String>] [-ExcludeBuiltIn] [-Server <String>] [-Credential <PSCredential>] [<CommonParameters>]
```

Description

Get-ADGroupReport will create a custom report for a group showing members. Get-ADGroupUser is intended to display group membership details Get-ADGroupReport focuses on the group, although members are also displayed. Members are always gathered recursively. You can filter for specific types of groups. You can also opt to exclude groups under CN=Users and CN=BuiltIn. The groups "Domain Users", "Domain Computers", and "Domain Guests" are always excluded from this command.

If your PowerShell hosts supports it, ANSI color schemes will be used to highlight things such as Distribution groups and disabled user accounts.

Examples

Example 1

```
PS C:\> Get-ADGroupReport sales

Name        : CN=Sales,OU=Sales,DC=Company,DC=Pri [Global|Security]
ManagedBy   : CN=SamanthaS,OU=Sales,DC=Company,DC=Pri
Description : Sales Force Resources
────────────────────────────────────────────────────────────────────────────────


Displayname    Name      Description DistinguishedName
-----------    ----      ----------- -----------------
Sam Smith      SamS      Sales       CN=SamS,OU=Sales,DC=Company,DC=Pri
Sonya Smith    SonyaS    Sales       CN=SonyaS,OU=Sales,DC=Company,DC=Pri
Samantha Smith SamanthaS Sales       CN=SamanthaS,OU=Sales,DC=Company,DC=Pri
```

If your PowerShell host supports it, Disabled user accounts will display the distinguished name in red.

Example 2

```
PS C:\> Get-ADGroupReport  -ExcludeBuiltIn | Format-Table -View age

Name                 Members Created                 Modified                            Age
----                 ------- -------                 --------                            ---
IT                         5 1/25/2021 1:32:44 PM    3/15/2021 5:42:50 PM            17:04:02
Sales                      3 1/25/2021 1:32:44 PM    3/16/2021 9:52:29 AM            00:54:23
Marketing                  3 1/25/2021 1:32:44 PM    3/16/2021 9:52:29 AM            00:54:24
Accounting                 3 1/25/2021 1:32:44 PM    3/4/2021 9:25:39 AM          12.01:21:14
JEA Operators              4 1/25/2021 1:32:44 PM    1/28/2021 11:34:57 AM        46.23:11:56
Web Servers                1 1/25/2021 1:32:45 PM    3/15/2021 5:42:33 PM            17:04:20
DevOpsPrimary              0 1/25/2021 4:47:53 PM    1/27/2021 10:35:11 AM        48.00:11:42
DevOpsBackup               3 1/25/2021 4:48:02 PM    3/16/2021 10:12:01 AM           00:34:52
Payroll Managers           0 1/26/2021 10:12:34 AM   1/26/2021 10:12:34 AM        49.00:34:19
ThetaDL                    1 2/16/2021 8:32:36 AM    3/16/2021 9:43:32 AM            01:03:21
StrategyDL                 0 2/16/2021 9:03:12 AM    3/15/2021 5:45:07 PM            17:01:46
SecOpAdmin                 2 2/24/2021 12:37:28 PM   2/24/2021 12:39:15 PM        19.22:07:38
FocusOne                  16 2/24/2021 3:27:58 PM    3/16/2021 9:43:32 AM            01:03:21
SupportTech                2 2/26/2021 6:12:51 PM    3/15/2021 5:43:03 PM            17:03:51
DL-Test                    4 3/3/2021 1:54:01 PM     3/16/2021 9:43:32 AM            01:03:22
DL-Test2                   1 3/3/2021 1:55:13 PM     3/3/2021 2:01:50 PM          12.20:45:04
```

If your console supports it, Distribution Lists will be displayed in green, and a member count of 0 will be displayed in red.

Example 3

```
PS C:\> Get-ADGroupReport -ExcludeBuiltIn | Format-Table -view summary


   DistinguishedName: CN=IT,OU=IT,DC=Company,DC=Pri

Name                              Members Category        Scope       Branch
----                              ------- --------        -----       ------
IT                                      5 Security        Global      OU=IT,DC=Company,DC=Pri


   DistinguishedName: CN=Sales,OU=Sales,DC=Company,DC=Pri

Name                              Members Category        Scope       Branch
----                              ------- --------        -----       ------
Sales                                   3 Security        Global      OU=Sales,DC=Company,DC=Pri


   DistinguishedName: CN=Marketing,OU=Marketing,DC=Company,DC=Pri

Name                              Members Category        Scope       Branch
----                              ------- --------        -----       ------
Marketing                               3 Security        Global      OU=Marketing,DC=Company,DC=Pri
...`
```

Get groups and format with a custom view. If your console session supports it, some of the output will be color-coded with ANSI sequences.

Parameters

-Category

Filter on the group category

```
Type: String
Parameter Sets: (All)
Aliases:
Accepted values: All, Distribution, Security

Required: False
Position: Named
Default value: All
Accept pipeline input: False
Accept wildcard characters: False
```

-Credential

Specify an alternate credential. This will be used to query the domain and all domain controllers.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

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

-ExcludeBuiltIn

Exclude BuiltIn and Users. Domain Users, Domain Guests, and Domain Computers are always excluded regardless of this parameter.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

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

-Name

Enter an AD Group name. Wildcards are allowed.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-Scope

Filter on group scope

```
Type: String
Parameter Sets: (All)
Aliases:
Accepted values: Any, DomainLocal, Global, Universal

Required: False
Position: Named
Default value: Any
Accept pipeline input: False
Accept wildcard characters: False
```

-SearchBase

Enter the distinguished name of the top-level container or organizational unit.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-Server

Specify a domain controller to query for a list of domain controllers.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADGroupReport

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADGroupUser

Get-ADGroup

Get-ADGroupMember

Get-ADGroupUser

Synopsis

Get user members of an AD group.

Syntax

```
Get-ADGroupUser [-Name] <String> [-Server <String>] [-Credential <PSCredential>] [<CommonParameters>]
```

Description

This command will display all users of a given Active Directory group. The search is automatically recursive. The default output is a formatted table that will highlight disabled accounts in red.

Examples

Example 1

```
PS C:\> Get-ADGroupUser sales


   DistinguishedName: CN=SamS,OU=Sales,DC=Company,DC=Pri [Sam Smith]

Name            Title              Description                  PasswordLastSet
----            -----             -----------                    --------------
SamS                               Sales Staff             1/25/2021 1:32:36 PM


   DistinguishedName: CN=SonyaS,OU=Sales,DC=Company,DC=Pri [Sonya Smith]

Name            Title              Description                  PasswordLastSet
----            -----              -----------                   --------------
SonyaS          Account Executive  Sales                   1/25/2021 1:32:37 PM


   DistinguishedName: CN=SamanthaS,OU=Sales,DC=Company,DC=Pri [Samantha Smith]

Name            Title              Description                  PasswordLastSet
----            -----              -----------                   --------------
SamanthaS       Sales Assistant    Sales Staff             1/25/2021 1:32:37 PM
```

Disabled accounts will have their distinguished name displayed in red.

Example 2

```
PS C:\> Get-ADGroupUser sales | format-list


   Group: CN=Sales,OU=Sales,DC=Company,DC=Pri


DistinguishedName : CN=SamS,OU=Sales,DC=Company,DC=Pri
Name              : SamS
Displayname       : Sam Smith
Description       : Sales Staff
Title             :
Department        : Sales
Enabled           : False
PasswordLastSet   : 3/4/2021 4:03:23 PM

DistinguishedName : CN=SonyaS,OU=Sales,DC=Company,DC=Pri
Name              : SonyaS
Displayname       : Sonya Smith
Description       : Sales
Title             : Account Executive
Department        : Sales
Enabled           : True
PasswordLastSet   : 1/25/2021 1:32:37 PM
...
```

Using the defined list view.

Parameters

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

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

-Name

Enter the name of an Active Directory group.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

System.String

Outputs

ADGroupUser

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADGroupReport

Get-ADGroupMember

Get-ADReportingTools

Synopsis

Get a summary list of AD Reporting commands

Syntax

```
Get-ADReportingTools [<CommonParameters>]
```

Description

This command will present a summary of commands in the ADReportingTools module grouped by verb. The default output will show the command name, any defined aliases, and the help synopsis.

Examples

Example 1

```
PS C:\> Get-ADReportingTools
```

Parameters

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADReportingTool

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Open-ADReportingToolsHelp

Get-Module

Get-Command

Get-ADReportingToolsOptions

Synopsis

Get ADReportingTools color options

Syntax

```
Get-ADReportingToolsOptions [<CommonParameters>]
```

Description

Many of the commands in the ADReportingTools module have custom format files that utilize ANSI escape sequences to highlight key elements. The module defaults are stored in a variable called ADReportingToolsOptions. Use this command to view the current settings. If you access the variable directly, you won’t see the actual ANSI settings, and you might have to reset your console by typing "$([char]0x1b)[0m".

The ANSI sequences use the [char]0x1b escape character because it works in both Windows PowerShell and PowerShell 7.

Examples

Example 1

```
PS C:\> Get-ADReportingToolsOptions

Name             Value
----             -----
Alert            $([char]0x1b)[91m
Warning          $([char]0x1b)[38;5;220m
DistributionList $([char]0x1b)[92m
```

The actual values will be color-coded with the ANSI sequence.

Parameters

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADReportingToolsOption

Notes

An easy way to see ANSI samples is to install the PSScriptTools module from the PowerShell Gallery and use the Show-ANSISequence command.

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Set-ADReportingToolsOIptions

Get-ADSiteDetail

Synopsis

Get a more detailed AD site report.

Syntax

```
Get-ADSiteDetail [-Name <String>] [[-Server] <String>] [[-Credential] <PSCredential>] [<CommonParameters>]
```

Description

This command will present a summary report of your Active Directory sites showing a description, associated subnets, and when the site object was created and last modified.

Examples

Example 1

```
PS C:\> Get-ADSiteDetail


   Name: Default-First-Site-Name

Description        Subnets                 Created                Modified
-----------        -------                 -------                --------
Home Office        {192.168.3.0/24, 19... 2/23/2021 3:36:58 PM   2/23/2021...


   Name: NoCal

Description        Subnets                 Created                Modified
-----------        -------                 -------                --------
Bay Area Office    172.17.0.0/16           2/23/2021 3:38:33 PM   2/23/2021...
```

Parameters

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Name

Specify the name of an Active Directory site. The default is all sites.

```
Type: String
Parameter Sets: (All)
Aliases:

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADSiteDetail

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADSiteSummary

Get-ADReplicationSite

Get-ADSiteSummary

Synopsis

Get summary information about AD sites.

Syntax

```
Get-ADSiteSummary [-Name <String>] [[-Server] <String>] [[-Credential] <PSCredential>] [<CommonParameters>]
```

Description

This command will display a summary report of each Active Directory site.

Examples

Example 1

```
PS C:\> Get-ADSiteSummary


   Site: Default-First-Site-Name
   Description: Home Office

Subnet             Description                    Location
------             -----------                    --------
192.168.3.0/24     Employees
192.168.99.0/24    Datacenter                     HQDC


   Site: NoCal
   Description: Bay Area Office

Subnet             Description                    Location
------             -----------                    --------
172.17.0.0/16
```

Parameters

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Name

Specify the name of an Active Directory site. The default is all sites.

```
Type: String
Parameter Sets: (All)
Aliases:

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADSiteSummary

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADSiteDetail

Get-ADReplicationSite

Get-ADSummary

Synopsis

Get a summary report of your AD domain and forest.

Syntax

```
Get-ADSummary [[-Identity] <String>] [-Server <String>] [-Credential <PSCredential>] [<CommonParameters>]
```

Description

This simple command will give you a snapshot-sized summary of your Active Directory domain and forest.

Examples

Example 1

```
PS C:\> Get-ADSummary


   Forest: Company.Pri [Windows2016Forest]


RootDomain        : Company.Pri
Domains           : {Company.Pri}
Domain            : Company.Pri
DomainMode        : Windows2016Domain
DomainControllers : {DOM1.Company.Pri, DOM2.Company.Pri}
GlobalCatalogs    : {DOM1.Company.Pri, DOM2.Company.Pri}
SiteCount         : 2
```

Parameters

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

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

-Identity

Specify the domain name. The default is the user domain.

```
Type: String
Parameter Sets: (All)
Aliases: name

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

ADSummary

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADFSMO

Get-ADDomain

Get-ADForest

Get-ADUserAudit

Synopsis

Audit AD user management events.

Syntax

```
Get-ADUserAudit [[-DomainController] <String[]>] [-Since <DateTime>]
[-Events <String[]>] [-Credential <PSCredential>] [<CommonParameters>]
```

Description

This command will search the Security event logs on your domain controllers for specific user-related events. These activities are not replicated, so you have to search each domain controller. Be aware that you may see related events for some actions. For example, if you create and enable a new user, you'll see multiple entries for the same event.

The output will show you the user accounts that match the search criteria, and the domain account that was responsible. Although, this command can't tell you which administrator is responsible for which activity. The best you can learn is that for a given time frame, these user accounts were managed. Or these administrators did something. You would need to search the event log on the domain controller for more information.

You may need to enable logging and/or increase the size of the Security event log.

Examples

Example 1

```
PS C:\> get-aduseraudit -Events Created -Since 2/1/2021


   DomainController: DOM1.Company.Pri


EventType      : UserCreated
Since          : 2/1/2021 12:00:00 AM
TargetCount    : 10
Targets        : {COMPANY\darrens, COMPANY\S.Talone, COMPANY\ntesla, COMPANY...}
Administrators : {COMPANY\ArtD, COMPANY\Administrator, COMPANY\GladysK, COMP...}



   DomainController: DOM2.Company.Pri


EventType      : UserCreated
Since          : 2/1/2021 12:00:00 AM
TargetCount    : 6
Targets        : {COMPANY\astark, COMPANY\georgejet, COMPANY\maef, COMPANY\bo..}
Administrators : {COMPANY\GladysK, COMPANY\ArtD}
```

Find all user accounts created since February 1, 2021.

Parameters

-Credential

Specify an alternate credential

```
Type: PSCredential
Parameter Sets: (All)
Aliases:

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

-DomainController

Specify one or more domain controllers to query. The default is all domain controllers in the user domain.

```
Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Events

Select one or more user account events

```
Type: String[]
Parameter Sets: (All)
Aliases:
Accepted values: Created, Deleted, Enabled, Disabled, Changed

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

-Since

Find all matching user management events since what date and time?

```
Type: DateTime
Parameter Sets: (All)
Aliases:

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

System.Object

Notes

An earlier version of this command was first published at: http://bit.ly/ADUserAudit

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-WinEvent

Get-ADUserCategory

Synopsis

Get AD User information based on category

Syntax

filter (Default)

```
Get-ADUserCategory [[-Filter] <String>] [-SearchBase <String>] -Category <String> [-Server <String>] [-Credential <PSCredential>] [<CommonParameters>]
```

id

```
Get-ADUserCategory [-Identity] <String> -Category <String> [-Server <String>] [-Credential <PSCredential>] [<CommonParameters>]
```

Description

Get-ADUserCategory is based on the concept of getting user information from a pre-defined category. For example, you might want to get the properties DisplayName, Name, Title, Department, and Manager for a Department category. The ADReportingTools module will define a set of pre-defined categories that you can reference through $ADUserReportingConfiguration.

These are the current defaults.

Department DisplayName,Name,Title,Department,Manager Basic DisplayName,Name,SamAccountname,UserPrincipalName,Enabled,WhenCreated,WhenChanged Address DisplayName,Name,TelephoneNumber,Office,StreetAddress,POBox,City,State,PostalCode Organization DisplayName,Name,Title,Department,Manager,Company,Office Pwinfo DisplayName,Name,PasswordExpired,PasswordLastSet,PasswordNeverExpires

The user’s distinguishedname will always be included.

You don’t have to remember what property names to include or reference.

Examples

Example 1

```
PS C:\> Get-ADUserCategory artd -Category basic


DistinguishedName : CN=ArtD,OU=IT,DC=Company,DC=Pri
DisplayName       : Art Deco
Name              : ArtD
SamAccountname    : ArtD
UserPrincipalName : artd@company.com
Enabled           : True
WhenCreated       : 1/25/2021 1:32:35 PM
WhenChanged       : 3/11/2021 6:32:58 PM
```

Example 2

```
PS C:\> Get-ADUserCategory -filter "department -eq 'sales'" -Category Department


DistinguishedName : CN=SamS,OU=Sales,DC=Company,DC=Pri
DisplayName       : Sam Smith
Name              : SamS
Title             :
Department        : Sales
Manager           : CN=SonyaS,OU=Sales,DC=Company,DC=Pri

DistinguishedName : CN=SonyaS,OU=Sales,DC=Company,DC=Pri
DisplayName       : Sonya Smith
Name              : SonyaS
Title             : Account Executive
Department        : Sales
Manager           :

DistinguishedName : CN=SamanthaS,OU=Sales,DC=Company,DC=Pri
DisplayName       : Samantha Smith
Name              : SamanthaS
Title             : Sales Assistant
Department        : Sales
Manager           : CN=SonyaS,OU=Sales,DC=Company,DC=Pri
```

Example 3

```
PS C:\> $ADUserReportingConfiguration += [pscustomobject]@{Name="Custom";Properties="DisplayName","Description"}
PS C:\> Get-ADUserCategory -filter "givenname -like 'a*'" -Category custom

DistinguishedName                              DisplayName       Description
-----------------                              -----------       -----------
CN=AaronS,OU=Accounting,DC=Company,DC=Pri      Aaron Smith       Accountant
CN=Al Fresco,OU=Dev,DC=Company,DC=Pri          Al Fresco
CN=A.Henaire,OU=Employees,DC=Company,DC=Pri    Alexander Henaire
CN=Alfonso Dente,OU=Sales,DC=Company,DC=Pri    Alfonso Dente
CN=AndreaS,OU=Accounting,DC=Company,DC=Pri     Andrea Smith      Accountant
CN=AndyS,OU=Accounting,DC=Company,DC=Pri       Andy Smith        Accountant
CN=Anthony Stark,OU=Research,DC=Company,DC=Pri Tony Stark
CN=AprilS,OU=IT,DC=Company,DC=Pri              April Showers     PowerShell Guru
CN=A.Fieldhouse,OU=Employees,DC=Company,DC=Pri Aron Fieldhouse   sample user ...
CN=ArtD,OU=IT,DC=Company,DC=Pri                Art Deco          PowerShell E...
CN=Art Frame,OU=Accounting,DC=Company,DC=Pri   Art Frame         Test User
```

The first command is adding a new category. The second command uses the category.

Parameters

-Category

Select a defined category.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-Credential

Specify an alternate credential. This will be used to query the domain and all domain controllers.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

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

-Filter

Specify an AD filter like "department -eq 'sales'". The default is all Enabled user accounts.

```
Type: String
Parameter Sets: filter
Aliases:

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Identity

Enter an AD user identity

```
Type: String
Parameter Sets: id
Aliases:

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

-SearchBase

Enter the distinguished name of the top-level container or organizational unit.

```
Type: String
Parameter Sets: filter
Aliases:

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

-Server

Specify a domain controller to query for a list of domain controllers.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

System.Object

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADUser

Get-ADDepartmentMember

Get-NTDSInfo

Synopsis

Get information about the NTDS.dit and related files.

Syntax

```
Get-NTDSInfo [-Computername] <String[]> [-Credential <PSCredential>] [<CommonParameters>]
```

Description

Get-NTDSInfo will query a domain controller using PowerShell remoting to get information about the NTDS.dit and related files. You might use this to track the size of the file or to check on backups. A high log count might indicate a backup is needed.

Examples

Example 1

```
PS C:\> Get-NTDSInfo -computername dom1 | format-list

DomainController : DOM1.Company.Pri
Path             : C:\NTDS\ntds.dit
Size             : 16777216
FileDate         : 3/26/2021 1:13:26 PM
LogCount         : 34
Date             : 3/26/2021 4:15:00 PM
```

The default display is a table. The LogCount is the number of temp edb files in the NTDS folder. The FileDate is the timestamp of ntds.dit, and the Date property reflects when you ran the command.

Parameters

-Computername

Specify a domain controller name.

```
Type: String[]
Parameter Sets: (All)
Aliases: name

Required: True
Position: 0
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases:

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

System.String[]

Outputs

NTDSInfo

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADBackupStatus

New-ADChangeReport

Synopsis

Create an HTML change report.

Syntax

```
New-ADChangeReport [[-Since] <DateTime>] [-ReportTitle <String>]
[-Logo <String>] [--CSSUri <String>] [-EmbedCSS] [-ByContainer]
[-Path <String>] [-Server <String>] [-Credential <PSCredential>]
[-AuthType <String>] [<CommonParameters>]
```

Description

New-ADChangeReport will create an HTML report showing changes to Active Directory users, computers, and groups since a given date and time. The command uses Get-ADObject to query the WhenChanged property. The objects are organized by class and/or container and written to an HTML file. The command uses a CSS file from the ADReportingTools module, although you can specify your own. To make the HTML file portable, you can opt to embed the CSS content from a file source.

Examples

Example 1

```
PS C:\> New-ADChangeReport -Since "3/1/2021" -Path C:\work\March-2021-Change.html -ReportTitle "March AD Change Report" -EmbedCSS
```

This example will create a report called March-2021-Change.html with Active Directory changes since March 1, 2021l. The HTML report will use the default CSS file from the ADReportingTools module and embed it into the file.

Parameters

-AuthType

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.
```

```
Type: String
Parameter Sets: (All)
Aliases:
Accepted values: Negotiate, Basic

Required: False
Position: Named
Default value: Negotiate
Accept pipeline input: False
Accept wildcard characters: False
```

-ByContainer

Add a second grouping based on the object’s container or OU.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

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

--CSSUri

Specify the path to the CSS file. If you don’t specify one, the default module file will be used.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: changereport.css
Accept pipeline input: False
Accept wildcard characters: False
```

-Credential

Specify an alternate credential for authentication.

```
Type: PSCredential
Parameter Sets: (All)
Aliases:

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

-EmbedCSS

Embed the CSS file into the HTML document head. You can only embed from a file, not a URL.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

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

-Logo

Specify the path to an image file to use as a logo in the report.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-Path

Specify the path for the output file.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-ReportTitle

What is the report title?

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: "Active Directory Change Report"
Accept pipeline input: False
Accept wildcard characters: False
```

-Server

Specifies the Active Directory Domain Services domain controller to query. The default is your Logon server.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-Since

Enter a last modified datetime for AD objects. The default is the last 4 hours.

```
Type: DateTime
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

System.IO.FileInfo

Notes

An earlier version of this command was first described at https://jdhitsolutions.com/blog/powershell/8087/an-active-directory-change-report-from-powershell/

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADObject

New-ADDomainReport

Synopsis

Create an HTML report of your domain.

Syntax

```
New-ADDomainReport [[-Name] <String>] -FilePath <String>
[-ReportTitle <String>] [-CSSUri <String>] [-EmbedCSS] [-Server <String>] [-Credential <PSCredential>] [<CommonParameters>]
```

Description

This command will create an HTML report of your domain. The report layout is by container and organizational unit. Underneath each branch will be a table display of users, computers, and groups. Beneath each group will be a table of recursive group members. You should get detail about users and computers if you hover the mouse over the distinguished name.

The ADReportingTools module includes a CSS file which will be used by default. But you can specify an alternate CSS file. If you want to make the file portable, you can opt to embed the CSS into the HTML file. You can only embed from a file, not a URL reference.

Examples

Example 1

```
PS C:\> New-ADDomainReport -filepath c:\work\company.html -embedcss
```

Create the HTML report and embed the default CSS file.

Parameters

-CSSUri

Specify the path to the CSS file. If you don’t specify one, the default module file will be used. The default file is in the Reports folder of this module.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

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

-EmbedCSS

Embed the CSS file into the HTML document head. You can only embed from a file, not a URL.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

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

-FilePath

Specify the output HTML file.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-Name

Specify the domain name. The default is the user domain.

```
Type: String
Parameter Sets: (All)
Aliases: domain

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ReportTitle

Enter the name of the report to be displayed in the web browser.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: Domain Report
Accept pipeline input: False
Accept wildcard characters: False
```

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

System.IO.File

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Show-DomainTree

Open-ADReportingToolsHelp

Synopsis

Open a PDF help file.

Syntax

```
Open-ADReportingToolsHelp [<CommonParameters>]
```

Description

Open-ADReportingToolsHelp will launch a PDF file with all module documentation for the ADReportingTools module. The command should launch the file with whatever application is associated with the .PDF extension.

Examples

Example 1

```
PS C:\> Open=ADReportingToolsHelp
```

Launch the help PDF file.

Parameters

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

None

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADReportingTools

Set-ADReportingToolsOptions

Synopsis

Change an ADReportingToolsOptions setting.

Syntax

```
Set-ADReportingToolsOptions [-Name] <String> -ANSI <String> [<CommonParameters>]
```

Description

Many of the commands in the ADReportingTools module have custom format files that utilize ANSI escape sequences to highlight key elements. The module defaults are stored in a variable called ADReportingToolsOptions. Use this command to modify a current setting.

Examples

Example 1

```
PS C:\> Set-ADReportingToolsOptions DistributionList -ANSI "$([char]0x1b)[36m"
```

This will change the color value for DistributionList entries. The change is not persistent unless you put it in a PowerShell profile script.

Parameters

-ANSI

Specify the opening ANSI sequence. The module uses the [char]0x1b escape sequence because it works in both Windows PowerShell and PowerShell 7.x.

```
Type: String
Parameter Sets: (All)
Aliases:

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

-Name

Specify an option.

```
Type: String
Parameter Sets: (All)
Aliases:
Accepted values: DistributionList, Alert, Warning

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

None

Notes

An easy way to see ANSI samples is to install the PSScriptTools module from the PowerShell Gallery and use the Show-ANSISequence command.

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-ADReportingToolsOptions

Show-DomainTree

Synopsis

Display the domain in a tree format.

Syntax

```
Show-DomainTree [[-Name] <String>] [-UseDN] [-Server <String>]
[-Credential <PSCredential>] [-Containers] [<CommonParameters>]
```

Description

This command will display your domain in a tree view at the console. By default, Show-DomainTree will use color-coded ANSI formatting. The default display uses the organizational unit names. Although, you can use the distinguishedname of each branch. If you use -Containers, containers like Users will be included.

Examples

Example 1

```
PS C:\> Show-DomainTree

DC=Company,DC=Pri
│
├── Accounting
│   ├── Banking
│   ├── Finance
│       ├── Corp Investment
│   ├── Payroll
├── Dev
│   ├── Ops
├── Domain Controllers
├── Employees
│   ├── Exec
│       ├── VIP
│   ├── Temporary Hires
├── IT
│   ├── Help Desk
│       ├── TechStaff
│           ├── Test
│   ├── SecOps
├── JEA_Operators
├── Marketing
│   ├── Agency
├── Research
├── Sales
│   ├── InsideSales
│   ├── OutsideSales
├── Servers
│   ├── AppDev
│   ├── DMZ
│   ├── Web
│       ├── Staging
└── Suspended
```

Output will color-coded using ANSI escape sequences.

Example 2

```
PS C:\> PS C:\> Show-DomainTree -usedn

DC=Company,DC=Pri
│
├── OU=Accounting,DC=Company,DC=Pri
│   ├── OU=Banking,OU=Accounting,DC=Company,DC=Pri
│   ├── OU=Finance,OU=Accounting,DC=Company,DC=Pri
│       ├── OU=Corp Investment,OU=Finance,OU=Accounting,DC=Company,DC=Pri
│   ├── OU=Payroll,OU=Accounting,DC=Company,DC=Pri
├── OU=Dev,DC=Company,DC=Pri
│   ├── OU=Ops,OU=Dev,DC=Company,DC=Pri
├── OU=Domain Controllers,DC=Company,DC=Pri
├── OU=Employees,DC=Company,DC=Pri
│   ├── OU=Exec,OU=Employees,DC=Company,DC=Pri
│       ├── OU=VIP,OU=Exec,OU=Employees,DC=Company,DC=Pri
│   ├── OU=Temporary Hires,OU=Employees,DC=Company,DC=Pri
├── OU=IT,DC=Company,DC=Pri
│   ├── OU=Help Desk,OU=IT,DC=Company,DC=Pri
│       ├── OU=TechStaff,OU=Help Desk,OU=IT,DC=Company,DC=Pri
│           ├── OU=Test,OU=TechStaff,OU=Help Desk,OU=IT,DC=Company,DC=Pri
│   ├── OU=SecOps,OU=IT,DC=Company,DC=Pri
...
```

Display the domain tree using distinguishednames.

Parameters

-Containers

Include containers and non-OU elements. Items with a GUID in the name will be omitted.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cn

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

-Credential

Specify an alternate credential.

```
Type: PSCredential
Parameter Sets: (All)
Aliases: RunAs

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

-Name

Specify the domain name. The default is the user domain.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Server

Specify a domain controller to query.

```
Type: String
Parameter Sets: (All)
Aliases: dc, domaincontroller

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

-UseDN

Display the domain tree using distinguished names.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: dn

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

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

String

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

New-ADDomainReport

Split-DistinguishedName

Synopsis

Split a distinguished name into its components.

Syntax

```
Split-DistinguishedName [-DistinguishedName] <String> [<CommonParameters>]
```

Description

Split-DistinguishedName will take a disdinguishedname and break it down to its component elements. The command does not verify the name or any of its elements.

Examples

Example 1

```
PS C:\>Get-ADGroup supporttech | Split-Distinguishedname


Name       : SupportTech
Branch     : Help Desk
BranchDN   : OU=Help Desk,OU=IT,DC=Company,DC=Pri
Domain     : Company
DomainDN   : DC=Company,DC=Pri
DomainDNS  : Company.Pri
```

Example 2

```
PS C:\> Split-DistinguishedName "CN=Foo,OU=Bar,OU=Oz,DC=Research,DC=Globomantics,DC=com"


Name       : Foo
Branch     : Bar
BranchDN   : OU=Bar,OU=Oz,DC=Research,DC=Globomantics,DC=com
Domain     : Research
DomainDN   : DC=Research,DC=Globomantics,DC=com
DomainDNS  : Research.Globomantics.com
```

Parameters

-DistinguishedName

Enter an Active Directory DistinguishedName.

```
Type: String
Parameter Sets: (All)
Aliases: dn

Required: True
Position: 0
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

System.String

Outputs

ADDistinguishedNameInfo

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Changelog for ADReportingTools

1.0.0

- First stable release.
- Updated `README.md`.
- Added command `Get-ADDepartment` and format file `addepartmentmember.format.ps1xml`.
- Exporting a global variable called `$ADReportingHash` which is used as an argument completer for `Get-ADDepartment`.
- Moved ANSI colors from `Show-DomainTree` to `$ADReportingToolsOptions`. (Issue #17)
- Added class coloring to ADBranch output.
- Modified ADBranch output to show disabled user accounts in red.
- Added command `Get-ADComputerReport` and format file `adcomputerreport.format.ps1xml`.
- Modified `adgroupreport.format.ps1xml` to add member count to the default output. (Issue #21)
- Added a view called `summary` to `adgroupreport.format.ps1xml`.
- Added command `Get-NTDSInfo` and format file `adntds.format.ps1xml`. (Discussion #18)
- Modified `Get-ADSummary` to better display PSBoundParameters with Verbose output in the PowerShell ISE.
- Updated format files to ensure ANSI formatting only happens in a Console host.
- Added command `Get-ADBackupStatus` and format file `adbackupstatus.format.ps1xml`.
- Help updates.

0.8.0-preview

- Updated `README.md`.
- Added `New-ADChangeReport`. (Issue #15)
- Added sample CSS file `changereport.css`.
- Added sample HTML report `samplechange.html`.
- Added private function `_convertObjects`.
- Added variable `ADReportingToolsOptions` and functions `Get-ADReportingToolsOptions` and `Set-ADReportingToolsOptions`. (Issue #16)
- Modified format files to use values from `$ADReportingToolsOptions`.
- Added `Universal` to $ADReportingToolsOptions` to highlight Universal groups.
- Added `DomainLocal` to $ADReportingToolsOptions` to highlight DomainLocal groups.

0.7.0-preview

- Fixed typo in `adbranchmember.format.ps1xml`.
- Added custom table view called `group` for ADGroup objects in `adgroup.format.ps1xml`.
- Added command `Split-DistinguisedName`.
- Added command `Get-ADGroupReport` with a custom format file `adgroupreport.format.ps1ml` and a custom type file `adgroupreport.types.ps1xml`. (Issue #3)
- Removed `About_ADReportingTools`.
- Updated `README.md`.

0.6.1-preview

- Fixed pre-release tag in the module.

0.6.0-preview

- Added online help links.
- Help updates.
- Modified `Get-ADSiteSummary` and `Get-ADSiteDetail` to allow getting site by name. (Issue #14)
- Modified `Get-ADBranch` to allow filter of users, groups, or computers, and to exclude containers. (Issue #13
- Published pre-release module to the PowerShell Gallery.
- Added category `Basic` to `adusers-categories.json` with properties `DisplayName`,`Name` ,`SamAccountname`,`UserPrincipalName`,`Enabled`,`WhenCreated`, and `WhenChanged`.
- Updated `README.md`.

0.5.0

- Minor file organization.
- Added type file `aduser.types.ps1xml`.
- Added custom view called `names` defined in `formats\aduser.format.ps1xml`.
- Added function `Get-ADUserCategory`.
- Help updates.

0.4.0

- Moved `_formatDN` to `private.ps1`.
- Updated module manifest with private data.
- Added command help. (Issue #1)
- Modified `New-ADDomainReport` to fix bug converting file path. (Issue #4)
- Added `Get-ADReportingTools` command. (Issue #5)
- Modified default view for `Get-ADFSMO` to be a list. (Issue #6)
- Added a view for `ADDomainControllerService` type to `addchealth.format.ps1xml`. (Issue #7)
- Added alias `fsmo` for `Get-ADFSMO`. (Issue #8)
- Modified `Get-ADDomainControllerHealth` to include computer name in the `Services` property. (Issue #9)
- Modified `Get-ADGroupUser` to fix a bug that was not getting the user’s Title. (Issue #10)
- Added `Get-ADCanonicalUser` with an alias of `Get-ADCNUser`. (Issue #11)

0.3.0

- Updated `Get-ADGroupUser` to get member detail depending on the class.
- Modified `Get-ADBranch` to include an `Enabled` property.
- Added private helper functions `_inserttoggle` and `_getpopData`.
- Added `New-ADDomainReport`.

0.2.0

- Added a default List view to `adgroupuser.format.ps1xml`.dir
- Added format file `adsummary.format.ps1xml`.
- Added `Get-ADDCHealth` and format file `addchealth.format.ps1xml`.

0.0.1

- Initial files