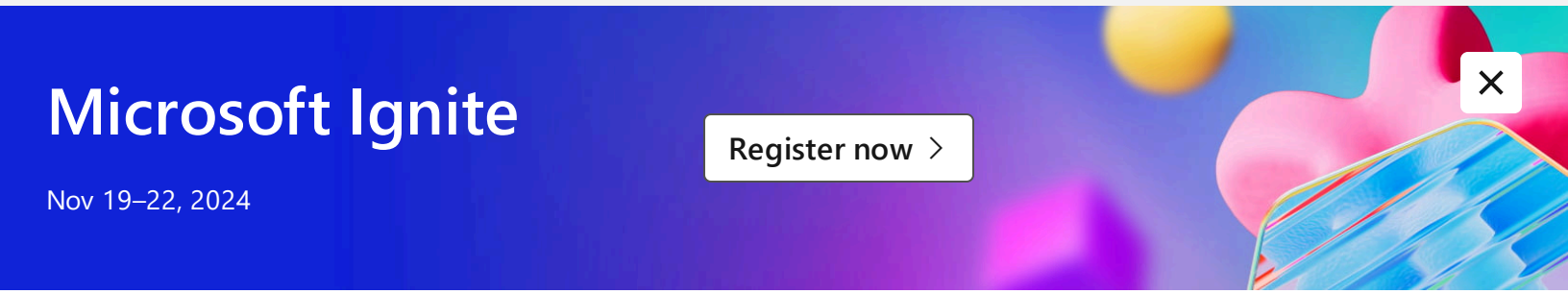

Get-Acl

Reference

Feedback

Module: Microsoft.PowerShell.Security

In this article

Gets the security descriptor for a resource, such as a file or registry key.

Syntax

```
Get-Acl
   [[-Path] <String[]>]
   [-Audit]
   [-Filter <String>]
   [-Include <String[]>]
   [-Exclude <String[]>]
   [<CommonParameters>]
```

```
Get-Acl
   -InputObject <PSObject>
   [-Audit]
   [-Filter <String>]
   [-Include <String[]>]
   [-Exclude <String[]>]
   [<CommonParameters>]
```

```
Get-Acl
   [-LiteralPath <String[]>]
   [-Audit]
   [-Filter <String>]
   [-Include <String[]>]
   [-Exclude <String[]>]
   [<CommonParameters>]
```

Description

> **This cmdlet is only available on the Windows platform.**

The `Get-Acl` cmdlet gets objects that represent the security descriptor of a file or resource. The security descriptor contains the access control lists (ACLs) of the resource. The ACL specifies the permissions that users and user groups have to access the resource.

Beginning in Windows PowerShell 3.0, you can use the **InputObject** parameter of `Get-Acl` to get the security descriptor of objects that do not have a path.

Examples

Example 1- Get an ACL for a folder

This example gets the security descriptor of the `C:\Windows` directory.

```
Get-Acl C:\Windows
```

Example 2 - Get an ACL for a folder using wildcards

This example gets the PowerShell path and SDDL for all of the `.log` files in the `C:\Windows` directory whose names begin with `s`.

```
Get-Acl C:\Windows\s*.log | Format-List -Property PSPath, Sd
```

The command uses the `Get-Acl` cmdlet to get objects representing the security descriptors of each log file. It uses a pipeline operator (`|`) to send the results to the `Format-List` cmdlet. The command uses the **Property** parameter of `Format-List` to display only the **PsPath** and **SDDL** properties of each security descriptor object.

Lists are often used in PowerShell, because long values appear truncated in tables.

The **SDDL** values are valuable to system administrators, because they are simple text strings that contain all of the information in the security descriptor. As such, they are easy to pass and store, and they can be parsed when needed.

Example 3 - Get count of Audit entries for an ACL

This example gets the security descriptors of the `.log` files in the `C:\Windows` directory whose names begin with `s`.

```
Get-Acl C:\Windows\s*.log -Audit | ForEach-Object { $_.Audit
```

It uses the **Audit** parameter to get the audit records from the SACL in the security descriptor. Then it uses the `ForEach-Object` cmdlet to count the number of audit records associated with each file. The result is a list of numbers representing the number of audit records for each log file.

Example 4 - Get an ACL for a registry key

This example uses the `Get-Acl` cmdlet to get the security descriptor of the Control subkey (`HKLM:\SYSTEM\CurrentControlSet\Control`) of the registry.

```
Get-Acl -Path HKLM:\System\CurrentControlSet\Control | Forma
```

The **Path** parameter specifies the Control subkey. The pipeline operator (`|`) passes the security descriptor that `Get-Acl` gets to the `Format-List` command, which formats the properties of the security descriptor as a list so that they are easy to read.

Example 5 - Get an ACL using **InputObject**

This example uses the **InputObject** parameter of `Get-Acl` to get the security descriptor of a storage subsystem object.

```
Get-Acl -InputObject (Get-StorageSubSystem -Name S087)
```

Parameters

-Audit

Gets the audit data for the security descriptor from the system access control list (SACL).

Expand table

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

-Exclude

Omits the specified items. The value of this parameter qualifies the **Path** parameter. Enter a path element or pattern, such as `*.txt`. Wildcards are permitted.

Expand table

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

-Filter

Specifies a filter in the provider's format or language. The value of this parameter qualifies the **Path** parameter. The syntax of the filter, including the use of wildcards, depends on the provider. Filters are more efficient than other parameters, because the provider applies them when getting the objects, rather than having PowerShell filter the objects after they are retrieved.

Expand table

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

-Include

Gets only the specified items. The value of this parameter qualifies the **Path** parameter. Enter a path element or pattern, such as `*.txt`. Wildcards are permitted.

Expand table

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

-InputObject

Gets the security descriptor for the specified object. Enter a variable that contains the object or a command that gets the object.

You cannot pipe an object, other than a path, to `Get-Acl`. Instead, use the **InputObject** parameter explicitly in the command.

This parameter is introduced in Windows PowerShell 3.0.

Expand table

Type:	PSObject
Position:	Named
Default value:	None
Required:	True
Accept pipeline input:	False

Accept wildcard characters:	False

-LiteralPath

Specifies the path to a resource. Unlike **Path**, the value of the **LiteralPath** parameter is used exactly as it is typed. No characters are interpreted as wildcards. If the path includes escape characters, enclose it in single quotation marks. Single quotation marks tell PowerShell not to interpret any characters as escape sequences.

This parameter is introduced in Windows PowerShell 3.0.

Expand table

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

-Path

Specifies the path to a resource. `Get-Acl` gets the security descriptor of the resource indicated by the path. Wildcards are permitted. If you omit the **Path** parameter, `Get-Acl` gets the security descriptor of the current directory.

The parameter name ("Path") is optional.

Expand table

Type:	String[]
Position:	1
Default value:	None
Required:	False
Accept pipeline input:	True
Accept wildcard characters:	True

Inputs

String

You can pipe a string that contains a path to this cmdlet.

Outputs

FileSecurity

DirectorySecurity

RegistrySecurity

This cmdlet returns an object that represents the ACLs that it gets. The object type depends upon the ACL type.

Notes

This cmdlet is only available on Windows platforms.

By default, `Get-Acl` displays the PowerShell path to the resource (`<provider>::<resource-path>`), the owner of the resource, and "Access", a list (array) of the access control entries in the discretionary access control list (DACL) for the resource. The DACL list is controlled by the resource owner.

When you format the result as a list, (`Get-Acl | Format-List`), in addition to the path, owner, and access list, PowerShell displays the following properties and property values:

- **Group**: The security group of the owner.
- **Audit**: A list (array) of entries in the system access control list (SACL). The SACL specifies the types of access attempts for which Windows generates audit records.
- **Sddl**: The security descriptor of the resource displayed in a single text string in Security Descriptor Definition Language format. PowerShell uses the **GetSddlForm** method of security descriptors to get this data.

Because `Get-Acl` is supported by the file system and registry providers, you can use `Get-Acl` to view the ACL of file system objects, such as files and directories, and registry objects, such as registry keys and entries.

Related Links

- Set-Acl

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see our contributor guide.

PowerShell feedback

PowerShell is an open source project. Select a link to provide feedback:

Open a documentation issue

Provide product feedback