

File System Security PowerShell Module 3.2.3

Source are available on CodePlex

<https://ntfssecurity.codeplex.com/>

Download

<https://gallery.technet.microsoft.com/scriptcenter/1abd77a5-9c0b-4a2b-acef-90dbb2b84e85/file/48905/1/NTFSSecurity%201.3.zip>

Introduction

Managing permissions with PowerShell is only a bit easier than in VBS or the command line as there are no cmdlets for most day-to-day tasks like getting a permission report or adding permission to an item. PowerShell only offers Get-Acl and Set-Acl but everything in between getting and setting the ACL is missing. This module closes the gap.

Documentation

For documentation please refer to:

[NTFSSecurity Tutorial 1 – Getting, adding and removing permissions](#)

[NTFSSecurity Tutorial 2 – Managing NTFS Inheritance and Using Privileges](#)

*Comments, feature requests and bug reports are very welcome:
raandree@live.com*

Installation

Just create the folder «NTFSSecurity» in one of the standard module folders and copy the files attached in there. The standard module folders are in the environment variable

`%PSModulePath%,` for example
`C:\Users\Documents\WindowsPowerShell\Modules.`
For example, all the files in the zip file have to be in
«C:\Users\raandree\Documents\WindowsPowerShell\Modules\NTFSSecurity». If you did this then the module should be listed in
«Get-Module -ListAvailable» and can be imported using «Import-Module NTFSSecurity».

Description

The module provides 10 cmdlets to manage permissions on the file system, like adding and removing ACEs, setting the inheritance, getting the current permissions or even get the effective permissions for a certain user.

The available cmdlets are listed below with a short description. More information can be retrieved in the PowerShell using Get-Help.

The name / SID translation is done by the Security2 class:
Security2 1.2.zip

All cmdlets have at least one parameter that supports the pipeline. They all can work with pipeline input coming from Get-ChildItem but some do more with what comes from the pipeline. For example you can remove permission by piping what Get-Ace returns to Remove-Ace:

```
[crayon-6a9d5756d9a82853310827/]
```

The pipeline support can also be used to backup and restore permissions of one or many items:

```
[crayon-6a9d5756d9a8a593900334/]
```

All cmdlets can handle SIDs and also SamAccountNames. The output contains always both unless a SID is not resolvable.

The types.ps1xml file is extending the common objects with some useful information and the format.ps1xml file formats all the output in almost the same way like the Get-ChildItem output.

By implementing the [Process Privilege

<https://processprivileges.codeplex.com/>] project the cmdlets can activate the required privileges for setting the ownership for example.

Add-Ace

Adds a specific ace to the current object. This can be done in just one line:

```
[crayon-6a9d5756d9a8d960890176/]
```

Get-Ace

Gives you a list of all permissions . normally you are interested not in the inherited permissions so the switch ExcludeInherited can be useful

```
[crayon-6a9d5756d9a8f514420029/]
```

Filtering works with Where-Object

```
[crayon-6a9d5756d9a91808210667/]
```

Get-OrphanedAce

Lists all permissions that can no longer be resolved. This normally happens if the account is no longer available so the permissions show up as a SID and not as an account name.

To remove all non-resolvable or orphaned permissions you can use the following line. But be very careful with that as maybe the account is not resolvable due to a network problem.

```
[crayon-6a9d5756d9a94133942568/]
```

Remove-Ace

Removes the permission for a certain account. As the pipeline is supported it takes also

```
[crayon-6a9d5756d9a96177327199/]
```

Get-EffectivePermissions

Shows the permissions an account actually has on a file or folder. If no parameter is specified it shows the effective permissions for the current user. However you can supply a user by using the SID or account name

```
[crayon-6a9d5756d9a98670633012/]
```

Get-Inheritance

Shows if inheritance is blocked

Enable-Inheritance

It can be a problem if certain files or folders on a volume have inheritance disabled. Making sure that inheritance is enabled can be done using this cmdlets:

```
[crayon-6a9d5756d9a9b521853770/]
```

Disable-Inheritance

See Enable-Inheritance

Get-Owner

Shows the owner of a file or folder

```
[crayon-6a9d5756d9a9d967753624/]
```

Set-Owner

Sets the owner to a specific account like:

```
[crayon-6a9d5756d9a9f077066641/]
```

More information

<https://gallery.technet.microsoft.com/scriptcenter/1abd77a5-9c0b-4a2b-acef-90dbb2b84e85>