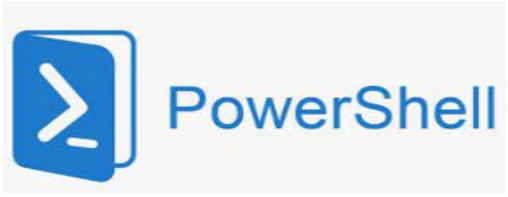

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'New-PSRoleCapabilityFile'

PS C:\Users\wahid> Get-Help New-PSRoleCapabilityFile

NAME

New-PSRoleCapabilityFile

SYNOPSIS

Creates a file that defines a set of capabilities to be exposed through a session configuration.

SYNTAX

New-PSRoleCapabilityFile [-Path] <System.String> [-AliasDefinitions <System.Collections.IDictionary[]>] [-AssembliesToLoad <System.String[]>] [-Author <System.String>] [-CompanyName <System.String>] [-Copyright <System.String>] [-Description <System.String>] [-EnvironmentVariables <System.Collections.IDictionary>] [-FormatsToProcess <System.String[]>] [-FunctionDefinitions <System.Collections.IDictionary[]>] [-Guid <System.Guid>] [-ModulesToImport <System.Object[]>] [-ScriptsToProcess <System.String[]>] [-TypesToProcess <System.String[]>] [-VariableDefinitions <System.Object>] [-VisibleAliases <System.String[]>] [-VisibleCmdlets <System.Object[]>] [-VisibleExternalCommands <System.String[]>] [-VisibleFunctions <System.Object[]>] [-VisibleProviders <System.String[]>] [<CommonParameters>]

DESCRIPTION

The `New-PSRoleCapabilityFile` cmdlet creates a file that defines a set of user capabilities that can be exposed through session configuration files. This includes determining which cmdlets, functions, and scripts are available to users. The capability file is a human-readable text file that contains a hash table of session configuration properties and values. The file has a .psrc file name extension, and can be used by more than one session configuration.

All the parameters of `New-PSRoleCapabilityFile` are optional except for the Path parameter, which specifies the path for the file. If you don't include a parameter when you run the cmdlet, the corresponding key in the session configuration file is commented-out, except where noted in the parameter description. For example, if you don't include the AssembliesToLoad parameter then that section of the session configuration file is commented out.

To use the role capability file in a session configuration, first place the file in a RoleCapabilities subfolder of a valid PowerShell module folder. Then reference the file by name in the RoleDefinitions field in a PowerShell Session Configuration (.pssc) file.

This cmdlet was introduced in Windows PowerShell 5.0.

PARAMETERS

-AliasDefinitions <System.Collections.IDictionary[]>

Adds the specified aliases to sessions that use the role capability file. Enter a hash table with the following keys:

- Name. Name of the alias. This key is required.

- Value. The command that the alias represents. This key is required.

- Description. A text string that describes the alias. This key is optional.

- Options. Alias options. This key is optional. The default value is None. The acceptable values for

this parameter are: None, ReadOnly, Constant, Private, or AllScope.

For example: `@{Name="hlp";Value="Get-Help";Description="Gets help";Options="ReadOnly"}`

-AssembliesToLoad <System.String[]>
Specifies the assemblies to load into the sessions that use the role capability file.

-Author <System.String>
Specifies the user that created the role capability file.

-CompanyName <System.String>
Identifies the company that created the role capability file. The default value is Unknown.

-Copyright <System.String>
Specifies a copyright for the role capability file. If you omit this parameter, `New-PSRoleCapabilityFile` generates a copyright statement using the value of the Author parameter.

-Description <System.String>
Specifies a description for the role capability file.

-EnvironmentVariables <System.Collections.IDictionary>

Specifies the environment variables for sessions that expose this role capability file. Enter a hash table in which the keys are the environment variable names and the values are the environment variable values.

For example: `EnvironmentVariables=@{TestShare="\\Server01\TestShare"}`

-FormatsToProcess <System.String[]>

Specifies the formatting files (`.ps1xml`) that run in sessions that use the role capability file. The value of this parameter must be a full or absolute path of the formatting files.

-FunctionDefinitions <System.Collections.IDictionary[]>

Adds the specified functions to sessions that expose the role capability. Enter a hash table with the following keys:

- Name. Name of the function. This key is required.

- ScriptBlock. Function body. Enter a script block. This key is required.

- Options. Function options. This key is optional. The default value is None. The acceptable values

for this parameter are: are None, ReadOnly, Constant, Private, or AllScope.

For example:

`@{Name="Get-PowerShellProcess";ScriptBlock={Get-Process PowerShell};Options="AllScope"}`

-Guid <System.Guid>

Specifies a unique identifier for the role capability file. If you omit this parameter, `New-PSRoleCapabilityFile` generates a GUID for the file. To create a new GUID in PowerShell, type `[guid]::NewGuid()`.

-ModulesToImport <System.Object[]>

Specifies the modules that are automatically imported into sessions that use the role capability file. By default, all the commands in listed modules are visible. When used with VisibleCmdlets or VisibleFunctions , the commands visible from the specified modules can be restricted.

Each module used in the value of this parameter can be represented by a string or by a hash table. A module string consists only of the name of the module. A module hash table can include ModuleName , ModuleVersion , and GUID keys. Only the ModuleName key is required.

For example, the following value consists of a string and a hash table. Any combination of strings and hash tables, in any order, is valid.

`"TroubleshootingPack", @{ModuleName="PSDiagnostics"; ModuleVersion="1.0.0.0";GUID="c61d6278-02a3-4618-ae37-a524d40a7f44"}`

-Path <System.String>

Specifies the path and filename of the role capability file. The file must have a `.psrc` filename extension.

-ScriptsToProcess <System.String[]>

Specifies scripts to add to sessions that use the role capability file. Enter the path and file names of the scripts. The value of this parameter must be a full or absolute path of the script file names.

-TypesToProcess <System.String[]>

Specifies type files (`.ps1xml`) to add to sessions that use the role capability file. Enter the type filenames. The value of this parameter must be a full or absolute path of the type filenames.

-VariableDefinitions <System.Object>

Specifies variables to add to sessions that use the role capability file.
Enter a hash table with the following keys:

- Name. Name of the variable. This key is required.

- Value. Variable value. This key is required.

For example: `@{Name="WarningPreference";Value="SilentlyContinue"}`

-VisibleAliases <System.String[]>
Limits the aliases in the session to those aliases specified in the value of this parameter, plus any aliases that you define in the AliasDefinition parameter. Wildcard characters are supported. By default, all aliases that are defined by the PowerShell engine and all aliases that modules export are visible in the session.

For example, to limit the available aliases to gm and gcm use this syntax: `VisibleAliases="gcm", "gp"`

When any Visible parameter is included in the role capability file, PowerShell removes the `Import-Module` cmdlet and its `ipmo` alias from the session.

-VisibleCmdlets <System.Object[]>
Limits the cmdlets in the session to those specified in the value of this parameter. Wildcard characters and Module Qualified Names are supported.

By default, all cmdlets that the modules in the session export are visible in the session. Use the SessionType and ModulesToImport parameters to determine which modules and snap-ins are imported into the session. If no modules in ModulesToImport expose the cmdlet, `New-PSRoleCapabilityFile`

tries to load the appropriate module.

When any Visible parameter is included in the session configuration file, PowerShell removes the `Import-Module` cmdlet and its `ipmo` alias from the session.

-VisibleExternalCommands <System.String[]>

Limits the external binaries, scripts and commands that can be executed in the session to those specified in the value of this parameter.

By default, no external commands are visible in this session.

When any Visible parameter is included in the session configuration file, PowerShell removes the `Import-Module` cmdlet and its `ipmo` alias from the session.

-VisibleFunctions <System.Object[]>

Limits the functions in the session to those specified in the value of this parameter, plus any functions that you define in the FunctionDefinitions parameter. Wildcard characters are supported.

By default, all functions exported by modules in the session are visible in that session. Use the SessionType and ModulesToImport parameters to determine which modules are imported into the session.

When any Visible parameter is included in the session configuration file, PowerShell removes the `Import-Module` cmdlet and its `ipmo` alias from the session.

-VisibleProviders <System.String[]>

Limits the PowerShell providers in the session to those specified in the value of this parameter. Wildcard characters are supported.

By default, all providers exported by a module in the session are visible in the session. Use the SessionType and ModulesToImport parameters to determine which modules are imported into the session.

When any Visible parameter is included in the session configuration file, PowerShell removes the `Import-Module` cmdlet and its `ipmo` alias from the session.

<CommonParameters>
This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

-------- Example 1: Create a blank role capability file --------

```
New-PSRoleCapabilityFile -Path ".\ExampleFile.psrc"
```

Example 2: Create a role capability file allowing users to restart services and any VDI computer

```
$roleParameters = @{
    Path = ".\Maintenance.psrc"
    Author = "User01"
    CompanyName = "Fabrikam Corporation"
    Description = "This role enables users to restart any service and restart any VDI computer."
    ModulesToImport = "Microsoft.PowerShell.Core"
    VisibleCmdlets = "Restart-Service", @{
                Name = "Restart-Computer"
                Parameters = @{ Name = "ComputerName"; ValidatePattern = "VDI\d+" }
```

```
    }
}
New-PSRoleCapabilityFile @roleParameters
```

REMARKS

To see the examples, type: "get-help New-PSRoleCapabilityFile -examples".

For more information, type: "get-help New-PSRoleCapabilityFile -detailed".

For technical information, type: "get-help New-PSRoleCapabilityFile -full".

For online help, type: "get-help New-PSRoleCapabilityFile -online"