

Powershell connector configuration

Connector attributes

argumentStyle

Powershell connector calls powershell command and passes mapped attributes to it. There are four different ways of passing an argument to powershell script and they all correspond with possible configuration values below.

- PARAMETERS_DASH - Executes command in this template: `COMMAND -PARAMETER_NAME VALUE`
- PARAMETERS_SLASH - Executes command in this template: `COMMAND /PARAMETER\_NAME:VALUE`
- VARIABLES - Executes command in this template: `$PARAMETER_NAME=VALUE COMMAND`
- HASH_MAP - This template exists only because powershell has problems passing multiple parameters to longer script blocks. It creates map as one input argument which will then contain all input parameters.

testScriptPath, authenticateScriptPath, searchScriptPath, deleteScriptPath, createScriptPath, updateScriptPath, syncScriptPath

These parameters contain path to scripts (on client machine) which will be executed when corresponding operation is invoked. Make sure that the script files are accessible.

useScriptCache

If true, then scripts, once read from drive, will be kept in memory for later reuse.

userName

Username used for authentication.

disableableCertificateChecks

If true, then connector will not check certificate validity.

script ResponseType

Only CSV is supported at the moment.

authenticationScheme

Specifies type of authentication to use. Possible values are:

- Basic
- NTLM
- Negotiate
- Kerberos
- CredSSP

endpointUri

Uri contains information about protocol, port and hostname. Common URIs are

- <http://HOST:5985/wsman> for http communication
- <https://HOST:5986/wsman> for https communication

domainName

Name of AD domain. It is used for authentication purposes.

multivaluedArgumentEncodingStrategy, multivalueEncodingSeparator

This is used for encoding multivalued parameters as arguments. Possible values are:

- SINGLE_VALUE_ONLY - Uses only one value
- MULTIVALUE_WITH_SEPARATOR - encodes multivalue as list of values separated by multivalueEncodingSeparator
- MULTIVALUE - encodes multivalue using Java Arrays.toString method

Setting-up WinRM

[Installation and Configuration for Windows Remote Management](#)

MS Exchange

In order to use Powershell commands such as New-Mailbox, you need to add Exchange snapIn. Example powershell create script can be seen below.

```
param([string]$__UID__ = "", [string]$__NAME__ = "",[string]$SamAccountName
= "", [string]$name = "", [string]$UserPrincipalName = "",
[string]$__PASSWORD__)
Try{
    Add-PSSnapIn -Name Microsoft.Exchange.Management.PowerShell.SnapIn
    $secure_password = ConvertTo-SecureString -String $__PASSWORD__ -
AsPlainText -Force
    New-Mailbox `
        -SamAccountName $SamAccountName `
        -Name $name `
        -UserPrincipalName $UserPrincipalName `
        -Password $secure_password `
        -Confirm:$false `
        -ErrorAction Stop | out-null -outvariable $mailbox

    Write-Host "__UID__,__NAME__"
    Write-Host "$SamAccountName,$SamAccountName"
} Catch {
    Write-Host "__ERROR__"
    Write-Host "$_.Exception.Message"
}
```

Another requirement is to use CredSSP authentication scheme.

Useful links

- [Set up CredSSP](#)
- [Guide on enabling winrm](#)
- [Troubleshooting winrm \(in czech\)](#)
- [Official guide on how to enable CredSSP](#)
- [Useful guide on winrm](#)
- [Script to configure Winrm for Ansible](#)
- [Winrm and https](#)
- [More troubleshooting](#)

From:
<https://wiki.czechidm.com/> - **IdStory Identity Manager**

Permanent link:
<https://wiki.czechidm.com/dev/documentation/systems/dev/powershell-connector>

Last update: **2019/02/26 09:19**

