



Windows PowerShell Get-Help on Cmdlet 'New-LocalUser'

PS:\>Get-HELP New-LocalUser -Full

NAME

New-LocalUser

SYNOPSIS

Creates a local user account.

SYNTAX

```
New-LocalUser [-Name] <System.String> [-AccountExpires <System.DateTime>] [-AccountNeverExpires] [-Description  
<System.String>] [-Disabled] [-FullName  
<System.String>] [-NoPassword] [-UserMayNotChangePassword] [-Confirm] [-WhatIf] [<CommonParameters>]
```

```
New-LocalUser [-Name] <System.String> [-AccountExpires <System.DateTime>] [-AccountNeverExpires] [-Description  
<System.String>] [-Disabled] [-FullName  
<System.String>] [-Password <System.Security.SecureString>] [-PasswordNeverExpires] [-UserMayNotChangePassword]  
[-Confirm] [-WhatIf] [<CommonParameters>]
```

DESCRIPTION

The `New-LocalUser` cmdlet creates a local user account. This cmdlet creates a local user account.

> [!NOTE] > The Microsoft.PowerShell.LocalAccounts module isn't available in 32-bit PowerShell on a 64-bit > system.

PARAMETERS

-AccountExpires <System.DateTime>

Specifies when the user account expires. You can use the `Get-Date` cmdlet to get a DateTime object. If you don't specify this parameter, the account doesn't expire.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-AccountNeverExpires <System.Management.Automation.SwitchParameter>

Indicates that the account doesn't expire.

Required?	false
Position?	named
Default value	False
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-Description <System.String>

Specifies a comment for the user account. The maximum length is 48 characters.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

Accept wildcard characters? false

-NoPassword <System.Management.Automation.SwitchParameter>

Indicates that the user account doesn't have a password.

Required? true

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Password <System.Security.SecureString>

Specifies a password for the user account. You can use ``Read-Host -AsSecureString``, ``Get-Credential``, or ``ConvertTo-SecureString`` to create a `SecureString` object for the password.

If you omit the `Password` and `NoPassword` parameters, ``New-LocalUser`` prompts you for the new user's password.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PasswordNeverExpires <System.Management.Automation.SwitchParameter>

Indicates whether the new user's password expires.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-UserMayNotChangePassword <System.Management.Automation.SwitchParameter>

Indicates that the user can't change the password on the user account.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet isn't run.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

<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>).

INPUTS

System.String

You can pipe a string to this cmdlet.

System.DateTime

You can pipe a DateTime object to this cmdlet.

System.Boolean

You can pipe a boolean value to this cmdlet.

System.Security.SecureString

You can pipe a secure string to this cmdlet.

OUTPUTS

System.Management.Automation.SecurityAccountsManager.LocalUser

This cmdlet returns a LocalUser object representing the created user account.

NOTES

Windows PowerShell includes the following aliases for `New-LocalUser`:

- `nlu`

A user name can't be identical to any other user name or group name on the computer. A user name can't consist only of periods `.` or spaces. A user name can

contain up to 20 uppercase characters or lowercase characters. A user name can't contain the following characters:

`" , / , ` , [,] , : , ; , | , = , , , + , * , ? , < , > , @`

A password can contain up to 127 characters.

The PrincipalSource property is a property on LocalUser , LocalGroup , and LocalPrincipal objects that describes the source of the object. The possible sources are as follows:

- `Local`
- `Active Directory`
- `AzureAD`
- `MicrosoftAccount`

> [!NOTE] > PrincipalSource is supported only by Windows 10, Windows Server 2016, and later versions of > the Windows operating system. For earlier versions, the property is blank.

----- Example 1: Create a user account -----

```
New-LocalUser -Name 'User02' -Description 'Description of this account.' -NoPassword
```

Name	Enabled	Description
------	---------	-------------

----	-----	-----
------	-------	-------

User02	True	Description of this account.
--------	------	------------------------------

This command creates a local user account and doesn't specify the AccountExpires or Password parameters. The account doesn't expire or have a password.

----- Example 2: Create a user account that has a password -----

```
$Password = Read-Host -AsSecureString
```

```
$params = @{
```

```

Name      = 'User03'

Password  = $Password

FullName  = 'Third User'

Description = 'Description of this account.'
}

New-LocalUser @params

```

```

Name  Enabled Description
----  -
User03 True    Description of this account.

```

The first command uses the ``Read-Host`` cmdlet to prompts you for a password. The command stores the password as a secure string in the ``$Password`` variable.

The second command creates a local user account and sets the new account's password to the secure string stored in ``$Password``. The command specifies a user name, full name, and description for the user account.

RELATED LINKS

Online

Version:

[https://learn.microsoft.com/powershell/module/microsoft.powershell.localaccounts/new-localuser?view=powershell-5.1&WT.](https://learn.microsoft.com/powershell/module/microsoft.powershell.localaccounts/new-localuser?view=powershell-5.1&WT.mc_id=ps-gethelp)

[mc_id=ps-gethelp](https://learn.microsoft.com/powershell/module/microsoft.powershell.localaccounts/new-localuser?view=powershell-5.1&WT.mc_id=ps-gethelp)

Disable-LocalUser

Enable-LocalUser

Get-LocalUser

Remove-LocalUser

Rename-LocalUser

Set-LocalUser