

How to Create a New Local User with PowerShell

When you need to create a local user in Windows 10 or 11 you can use the User Accounts control panel. But we can also use PowerShell to create a new local user. This way we can easily automate creating a local account on Windows devices.

To create a local user with PowerShell you will need to have administrator access to the computer and run PowerShell as admin (elevated). Otherwise, you won't be able to create accounts.

In this article, I will explain how you can create a new localuser. At the end of the article, I have two PowerShell scripts that you can use to create a local user.

Create Local User with PowerShell

To create a new local user we are going to use the `New-LocalUser` cmdlet in PowerShell. We have the option to set a password for the account or create an account without a password.

There are also a couple of other useful parameters that we can use:

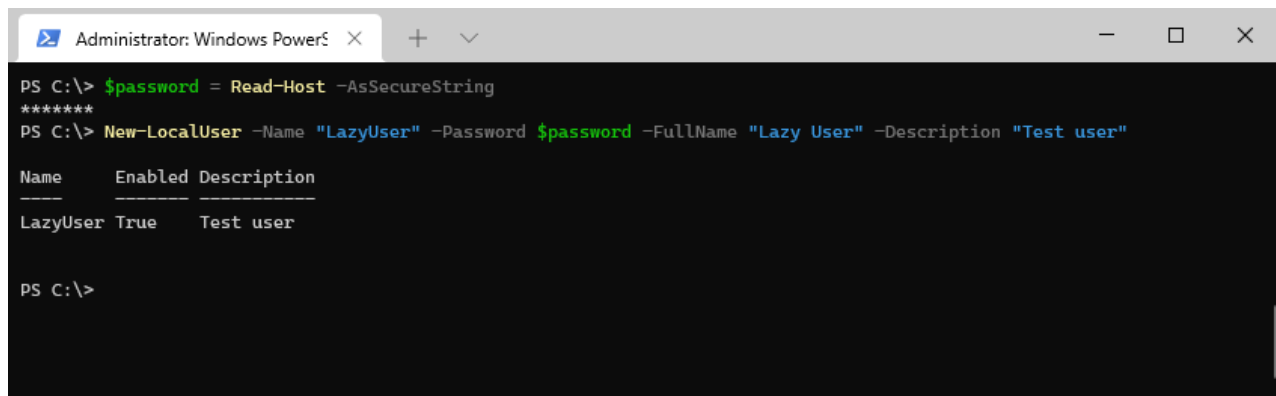
Parameter	Description
-Name	Login name of the account – max 20 characters
-Password	Password – supplied with a secure string
-Description	Description of the account
-AccountExpires	DateTime object when the account expires
-AccountNeverExpires	Account does not expire
-Disabled	Creates the account as disabled
-FullName	The display name of the account
-PasswordNeverExpires	Password does not expire
-UserMayNotChangePassword	User can't change the password

New-LocalUser cmdlet parameters

So to quickly create a local user account with PowerShell we can do the following:

```
$password = Read-Host -AsSecureString
```

New-LocalUser -Name "LazyUser" -Password \$password -FullName "Lazy User" -Description "Test user"



```
Administrator: Windows PowerShell
PS C:\> $password = Read-Host -AsSecureString
*****
PS C:\> New-LocalUser -Name "LazyUser" -Password $password -FullName "Lazy User" -Description "Test user"

Name      Enabled Description
----      -
LazyUser  True     Test user

PS C:\>
```

PowerShell New localuser

Note

PowerShell 7.3.x throws an error "New-LocalUser: Could not load type 'Microsoft.PowerShell.Telemetry.Internal.TelemetryAPI'", you can solve it by first importing the localaccounts module with: `import-module microsoft.powershell.localaccounts -UseWindowsPowerShell`

This small PowerShell script will require you to first enter the password, after which the user is created with the given password.

Providing the Password

As you can see this won't allow you to run the script autonomous, because you will need to enter a password. This is also the challenge with creating local users, most of the time you want to supply the password in a secure way.

If you run the script remotely or under your own supervision then you could write the password inside a PowerShell script and convert it to a secure string. But keep in mind, anyone who opens the script is able to read the password!

Username and Password

\$username = "LazyUser"

\$password = ConvertTo-SecureString "LazyAdminPwd123!" -AsPlainText -Force # Super strong plane text password here (yes this isn't secure at all)

Creating the user

New-LocalUser -Name "\$username" -Password \$password -FullName "\$username" -Description "Lazy Test user"

You could save this into a ps1 file and simply run it in an elevated PowerShell session.

Setting the Expired Date

By default, the new user account won't expire, but with the New-LocalUser cmdlet, we can set an expiration date for the account. For the date we will need to use a PowerShell DateTime object:

```
$date = Get-Date -Year 2022 -Month 06 -Day 10
```

```
# Creating the user
```

```
New-LocalUser -Name "$username" -Password $password -AccountExpires $date -  
FullName "$username" -Description "Lazy Test user"
```

Making user member of a group with Add-LocalGroupMember

After you have created the user you will need to make it a member of a local group. Without it, the user won't be able to log on. To make the user member of a group we are going to use the Add-LocalGroupMember cmdlet.

The Add-LocalGroupMember only requires the group name and the member that you want to add:

```
Add-LocalGroupMember -Group Users -Member LazyUser
```

The cmdlet doesn't give any output on success, only an error when the group name or member isn't found.

You can also add multiple users to a local group with PowerShell. Simply comma separate the members in the cmdlet:

```
Add-LocalGroupMember -Group Users -Member "LazyUser", "LazyUser2"
```

Complete Script for new localuser in PowerShell

I have created two scripts that will help you with creating a local user account with PowerShell. In both scripts, I have added the option to write a log file. This log file is stored on a network share, allowing you to easily check if the creation is successful on the computer.

The first script has a password set in the script, so you can simply run the script on a computer. Keep in mind that you will need to have administrator access to create a local user account!

```
<#
```

```
.SYNOPSIS
```

```
Create local admin acc
```

```
.DESCRIPTION
```

```
Creates a local administrator account on the computer. Requires RunAs permissions to  
run
```

```
.OUTPUTS
```

```
none
```

```
.NOTES
```

Version: 1.0

Author: R. Mens - LazyAdmin.nl

Creation Date: 25 march 2022

Purpose/Change: Initial script development

#>

Configuration

\$username = "adminTest" # Administrator is built-in name

\$password = ConvertTo-SecureString "LazyAdminPwd123!" -AsPlainText -Force # Super strong plane text password here (yes this isn't secure at all)

\$logFile = "\\server\folder\log.txt"

Function Write-Log {

param(

[Parameter(Mandatory = \$true)][string] \$message,

[Parameter(Mandatory = \$false)]

[ValidateSet("INFO","WARN","ERROR")]

[string] \$level = "INFO"

)

Create timestamp

\$timestamp = (Get-Date).ToString("yyyy/MM/dd HH:mm:ss")

Append content to log file

Add-Content -Path \$logFile -Value "\$timestamp [\$level] - \$message"

}

Function Create-LocalAdmin {

process {

try {

New-LocalUser "\$username" -Password \$password -FullName "\$username" -Description "local admin" -ErrorAction stop

Write-Log -message "\$username local user crated"

Add new user to administrator group

Add-LocalGroupMember -Group "Administrators" -Member "\$username" -ErrorAction stop

Write-Log -message "\$username added to the local administrator group"

}catch{

Write-log -message "Creating local account failed" -level "ERROR"

}

}

}

Write-Log -message "#####"

Write-Log -message "\$env:COMPUTERNAME - Create local admin account"

Create-LocalAdmin

Write-Log -message "#####"

The script will make the user member of the Administrators group in this case. You can of course change this to any other group. Make sure that you set the username, password, and logfile path in this first part of the script.

You can also download the complete script [here from my Github repository](#).

Local User account script

The second script creates a local user account that is a member of the user's groups. The difference with the first script is that this script will ask for the password.

```
<#
.SYNOPSIS
Create local user acc
.DESCRPTION
Creates a local user account on de computer. Requires RunAs permissions to run
.OUTPUTS
none
.NOTES
Version: 1.0
Author: R. Mens - LazyAdmin.nl
Creation Date: 25 march 2022
Purpose/Change: Initial script development
#>
# Configuration
$username = "LazyTestUser" # UserName
$fullName = "Lazy Test User" # Full name
$logFile = "\\server\folder\log.txt"
Function Write-Log {
param(
[Parameter(Mandatory = $true)][string] $message,
[Parameter(Mandatory = $false)]
[ValidateSet("INFO","WARN","ERROR")]
[string] $level = "INFO"
)
# Create timestamp
$timestamp = (Get-Date).ToString("yyyy/MM/dd HH:mm:ss")
# Append content to log file
Add-Content -Path $logFile -Value "$timestamp [$level] - $message"
}
Function Create-LocalUser {
process {
try {
New-LocalUser "$username" -Password $password -FullName "$fullname" -Description
"local user" -ErrorAction stop
Write-Log -message "$username local user created"
# Add new user to administrator group
Add-LocalGroupMember -Group "Users" -Member "$username" -ErrorAction stop
Write-Log -message "$username added to the local users group"
```

```
}catch{
Write-log -message "Creating local account failed" -level "ERROR"
}
}
}
# Enter the password
Write-Host "Enter the password for the local user account" -ForegroundColor Cyan
$password = Read-Host -AsSecureString
Write-Log -message "#####"
Write-Log -message "$env:COMPUTERNAME - Create local user account"
Create-LocalUser
Write-Log -message "#####"
Again, you can download the complete script here from my Github repository.
```

Wrapping Up

The New-LocalUser should also be capable of creating a local account that is connected to a Microsoft account. But the username is still limited to 20 characters and doesn't accept the @ symbol. So for now we are limited to local accounts only.

I hope this article helped you with creating a local user account with PowerShell. If you have any questions, just drop a comment below.

Did you **Liked** this **Article**?

Get the latest articles like this **in your mailbox**
or share this article

I hate spam to, so you can unsubscribe at any time.