

Robocopy with SMTP Status

Robocopy has been around forever, and if you've ever had to copy a bunch of files on Windows you've probably used it. It's tough, reliable, and way more powerful than just dragging and dropping in Explorer. But one thing it doesn't do out of the box is let you know when it's finished — which is annoying if you're running long jobs or setting it up as part of a backup. That's where this little PowerShell script comes in. It not only runs Robocopy for you, but also shoots off an email when it's done, letting you know whether everything worked or if something went wrong.

The script starts by defining the basics: where your files are coming from, where they're going, and where the log file should be saved. Then it sets up the email details — things like the SMTP server, port, username, password, and the from/to addresses for the notification emails. Once that's ready, it kicks off Robocopy with a bunch of useful switches. `/E` makes sure all subfolders (even empty ones) get copied, `/COPYALL` preserves everything about the files including permissions and timestamps, `/R:0` and `/W:0` mean it won't keep retrying endlessly if something fails, and `/LOG+` appends the output into your log file so you've always got a record.

When Robocopy finishes, the script grabs the exit code. Robocopy uses special numbers to tell you what happened — zero means nothing needed copying, one or two usually mean files were copied or extra files were found, and those are all considered “success” cases. Anything above three usually signals an error. With that in mind, the script checks the code and, if it's three or below, it sends a “success” email with the details and a link to the log file. If it's higher than that, it sends a “failure” email instead. Either way, you get a notification in your inbox telling you how the job went.

The best part is that you can drop this script into Task Scheduler and let it run whenever you want — overnight backups, weekly sync jobs, whatever works for your setup. Instead of manually checking logs or sitting around waiting for Robocopy to finish, you'll just get an email saying “all good” or “something broke.” It's a simple trick, but it makes Robocopy way more practical for real-world use, especially if you're looking after servers or backups and don't want to babysit them.

```
$Source = "C:\Source"
$Dest   = "D:\Dest"
$Log    = "C:\robocopy.log"

$SmtpServer = "SmtpServer"
$SmtpPort   = 465
$Username   = "Username"
$Password   = "Password"
$From       = "robocopy@example.com"
$To        = "alerts@example.com"
```

```
Write-Host "Starting Robocopy from $Source to $Dest ..."
& robocopy $Source $Dest /E /COPYALL /R:0 /W:0 /ETA /V /LOG+:$Log
$ExitCode = $LASTEXITCODE

$SecurePassword = ConvertTo-SecureString $Password -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PSCredential($Username, $SecurePassword)

if ($ExitCode -le 3) {
    Send-MailMessage -From $From -To $To -Subject "Robocopy Completed" -Body "Robocopy
finished successfully. Exit code: $ExitCode. See log at $Log" -SmtpServer $SmtpServer -Port
$SmtpPort -UseSsl -Credential $Cred
    Write-Host " Robocopy completed successfully. Email sent."
} else {
    Send-MailMessage -From $From -To $To -Subject "Robocopy Failed" -Body "Robocopy failed.
Exit code: $ExitCode. See log at $Log" -SmtpServer $SmtpServer -Port $SmtpPort -UseSsl -
Credential $Cred
    Write-Host " Robocopy failed. Email sent."
}
```

Revision #3

Created 26 August 2025 11:29:45 by Luke Carter

Updated 26 August 2025 13:03:43 by Luke Carter