source_compress_CTR-X.ps1

```
#region script global variables

#get the list of source directories in the original folder
$rootFolder = "<pathToRootFolder\>"
$destinFolder = "<pathToDestinFolder\>"
$archives = "<pathToArchivesFolder\>"
$tempVariable = $rootFolder

#endregion

#region collect information about .in files

#decide how long back to go
$timespan = new-timespan -Seconds 60

      #create a temporary folder using today's date
      $tempFolderRoot = "$rootFolder"
      $tempFolderDestin = "$destinFolder"
      $tempArchive = "$archives"
      $date = Get-Date
      $date = $date.ToString("yyyy-MM-dd-HH-mm-ss")
      $dateLog = Get-Date
      $dateLog = $dateLog.ToString("yyyy-MM-dd")
      $dateHeader = Get-Date
      $dateHeader = $dateHeader.ToString("yyyy/MM/dd-HH:mm")
      $tempFinalFolder = "$archives$dateLog\"

      New-Item -ItemType directory -Path $tempFinalFolder
      Write-Output
"########################################################################################
################################`r`n                                       Created on
$dateHeader
`r`n######################################################################################
###################################" | Add-Content "$archives\$dateLog.log"
      Write-Output "$date - Temporary workpath $tempFinalFolder created!`r`n" | Add-
Content "$archives\$dateLog.log"

      #lists files created more than 60 seconds ago
      $inFiles = Get-ChildItem "$rootFolder\*.in" | where {((Get-Date) -
$_.LastWriteTime) -gt ($timespan)}

      #counts the number of file created more than 60 seconds ago
      $numFiles = ($inFiles.count)

      Write-Output "$date - $numFiles files to be processed now!" | Add-Content
"$archives\$dateLog.log"
      Write-Verbose "$numFiles files to be processed now!" -verbose

#endregion

#region temporary workpath for zipping

      #move files to temporary folder before zipping.
      foreach ($in in $inFiles)
      {
            Copy-Item "$in" -destination $tempFinalFolder -Force
            Write-Output "$date - File $in moved to temporary workpath $tempFinalFolder"
| Add-Content "$archives\$dateLog.log"
```

```
      }
      Start-Sleep -Seconds 1

      #Creates zip files to each source directory withing .txt files
      $CompressionToUse = [System.IO.Compression.CompressionLevel]::Optimal
      $IncludeBaseFolder = $false
      $zipTo = "{0}<CTR-X>_{1}.zip" -f $tempFolderRoot,$date

      #add the files in the temporary location to a zip file
      [Reflection.Assembly]::LoadWithPartialName( "System.IO.Compression.FileSystem" )
      [System.IO.Compression.ZipFile]::CreateFromDirectory($tempFinalFolder, $zipTo,
$CompressionToUse, $IncludeBaseFolder)
      Write-Output "`r`n$date - Zip file $zipTo created!" | Add-Content
"$archives\$dateLog.log"

      #list zip files to move to destination folder
      $sourceZipFiles = Get-ChildItem "$rootFolder\*.zip"

      foreach ($zip in $sourceZipFiles)
      {
        #move zip files to destination folder
        Move-Item $zip -destination $tempFolderDestin
        Write-Output "$date - Zip file $zip moved to $tempFolderDestin`r`n" | Add-Content
"$archives\$dateLog.log"
      }

#endregion

#region cleaning up files already zipped

    #remove files already sent to zip package
      foreach($in in $inFiles)
      {
        Move-Item "$in" -destination $archives -Force
        Write-Verbose "$in" -verbose
        Write-Output "$date - Collected file $in moved to $archives" | Add-Content
"$archives\$dateLog.log"
      }

      #remove temporary folder on each source dir
      Remove-Item $tempFinalFolder -Recurse
      Write-Output "`r`n$date - Temporary workpath $tempFinalFolder deleted!" | Add-
Content "$archives\$dateLog.log"

#endregion

#region check new files for next run

    #list new .in files to be collected on next run
      $inFilesNewCount = Get-ChildItem "$rootFolder\*.in"
      $numFilesNewCount = ($inFilesNewCount.count)
      Write-Output "`r`n$date - $numFilesNewCount new files found to be collected on
$tempFolderRoot`r`n" | Add-Content "$archives\$dateLog.log"
      Write-Verbose "`r`n$numFilesNewCount new files to be processed on $tempFolderRoot"
-verbose

      Start-Sleep -Seconds 1

#endregion
```