

```
# Variables

$zipFilePath = resolve-path -path ".\anypointstudio.zip" # Resolve the relative path to absolute
$extractPath = "C:\AnypointStudio" # Path to extract the zip file
$destinationPath = "C:\Program Files\AnypointStudio" # Final installation folder
$shortcutPath = "$env:Public\Desktop\Anypoint Studio.lnk" # Desktop shortcut path

# Step 1: Ensure the zip file exists
if (-Not (Test-Path -Path $zipFilePath)) {
    Write-Host "Error: Zip file $zipFilePath does not exist." -ForegroundColor Red
    exit 1
}

# Step 2: Create the extraction folder
if (!(Test-Path -Path $extractPath)) {
    Write-Host "Creating extraction folder: $extractPath"
    New-Item -ItemType Directory -Path $extractPath -Force | Out-Null
}

Write-Host "Extracting $zipFilePath to $extractPath"
Add-Type -AssemblyName System.IO.Compression.FileSystem

# Handle long path extraction
try {
    [System.IO.Compression.ZipFile]::ExtractToDirectory($zipFilePath, $extractPath)
    Write-Host "Extraction completed successfully."
} catch {
    Write-Host "Error during extraction: $($_.Exception.Message)" -ForegroundColor Red
    exit 1
}
```

```
# Step 3: Move files from extracted folder to the final destination folder
```

```
if (!(Test-Path -Path $destinationPath)) {  
    Write-Host "Creating destination directory: $destinationPath"  
    New-Item -ItemType Directory -Path $destinationPath -Force | Out-Null  
}
```

```
Write-Host "Copying files from $extractPath to $destinationPath"
```

```
Copy-Item -Path "$extractPath\*" -Destination $destinationPath -Recurse -Force
```

```
# Step 4: Grant Full Control to All Users
```

```
Write-Host "Granting Full Control to all users on $destinationPath"
```

```
$acl = Get-Acl $destinationPath
```

```
$rule = [System.Security.AccessControl.FileSystemAccessRule]::new(  
    'Users',  
    'FullControl',  
    'ContainerInherit, ObjectInherit',  
    'None',  
    'Allow'  
)
```

```
$acl.AddAccessRule($rule)
```

```
Set-Acl -Path $destinationPath -AclObject $acl
```

```
Write-Host "Permissions successfully applied."
```

```
# Step 5: Unblock all files in the destination folder
```

```
Write-Host "Unblocking all files to prevent security warnings"
```

```
Get-ChildItem -Path $destinationPath -Recurse | ForEach-Object {
```

```
    Unblock-File -Path $_.FullName
```

```
}
```

# Step 6: Create a desktop shortcut

Write-Host "Creating desktop shortcut"

\$WScriptShell = New-Object -ComObject WScript.Shell

\$Shortcut = \$WScriptShell.CreateShortcut(\$shortcutPath)

# Updated shortcut paths

\$Shortcut.TargetPath = "C:\Program Files\AnypointStudio\AnypointStudio\AnypointStudio.exe" #  
Adjusted path

\$Shortcut.WorkingDirectory = "C:\Program Files\AnypointStudio\AnypointStudio" # Adjusted  
working directory

\$Shortcut.IconLocation = "C:\Program Files\AnypointStudio\AnypointStudio\AnypointStudio.exe" #  
Adjusted icon location

\$Shortcut.Save()

Write-Host "Anypoint Studio has been successfully installed to \$destinationPath." -  
ForegroundColor Green

# Step 7: Configure Environment Variable (Optional)

\$envPath = [System.Environment]::GetEnvironmentVariable("Path", "Machine")

\$envVarPath = "C:\Program Files\AnypointStudio\AnypointStudio" # Updated variable path

if (\$envPath -notlike "\*\$envVarPath\*") {

    Write-Host "Adding \$envVarPath to system PATH"

    [System.Environment]::SetEnvironmentVariable("Path", "\$envPath;\$envVarPath", "Machine")

} else {

    Write-Host "Path already contains \$envVarPath"

}

# Step 8: Remove extraction folder after moving the files

Write-Host "Removing extraction folder: \$extractPath"

```
Remove-Item -Path $extractPath -Recurse -Force
```

```
Write-Host "Extraction folder removed successfully."
```