

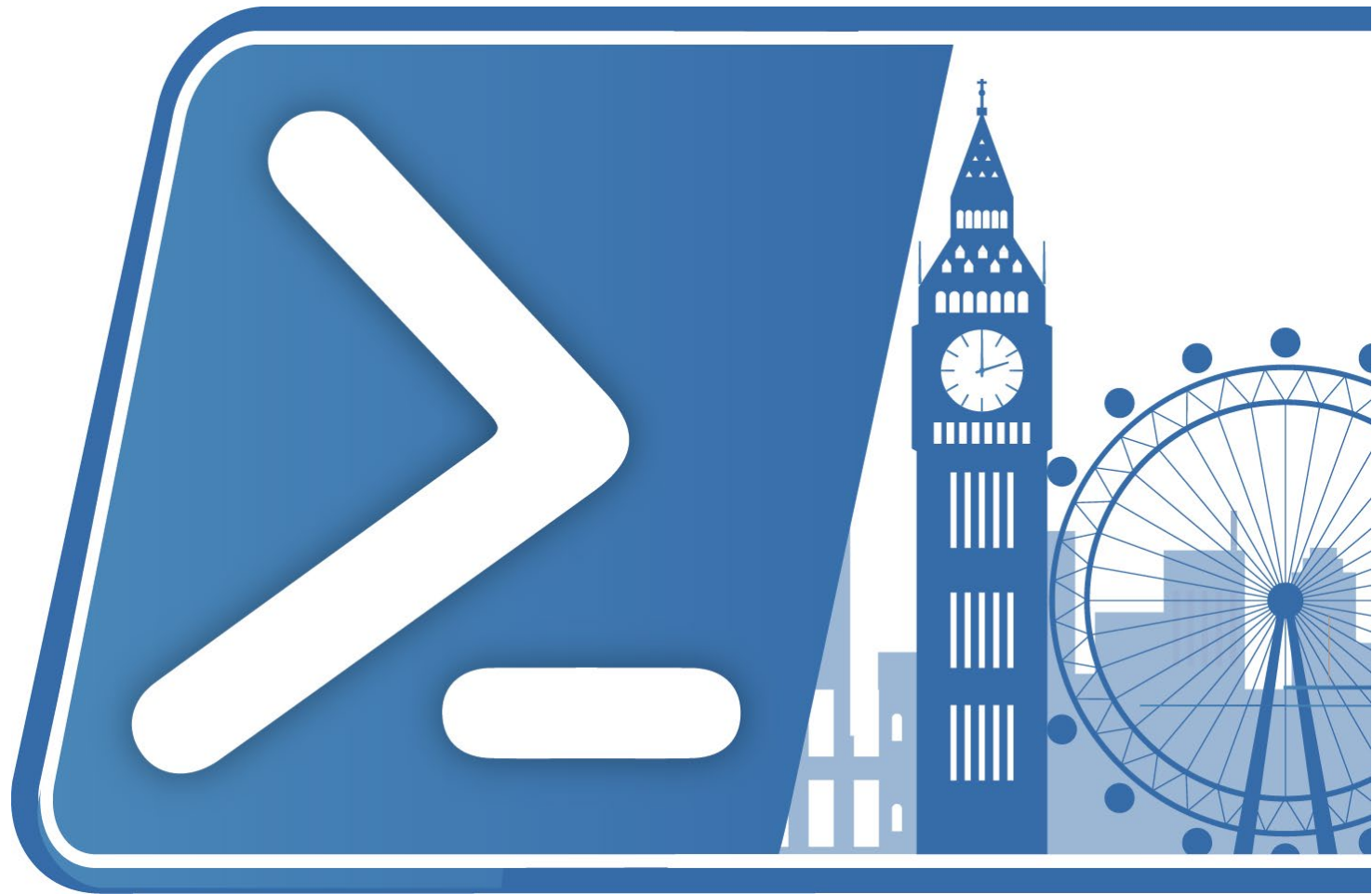
# London PowerShell Meetup

## PowerShell Tips & Tricks

by Daniel Krebs

4 December 2018

@lonpsug



# File or path-friendly date/time

B

```
PS C:\> Get-Date -Format FileDate
```

```
20190110
```

```
PS C:\> Get-Date -Format FileDateUniversal
```

```
20190110Z
```

```
PS C:\> Get-Date -Format FileDateTime
```

```
20190110T1535107108
```

```
PS C:\> Get-Date -Format FileDateTimeUniversal
```

```
20190110T1535220023Z
```

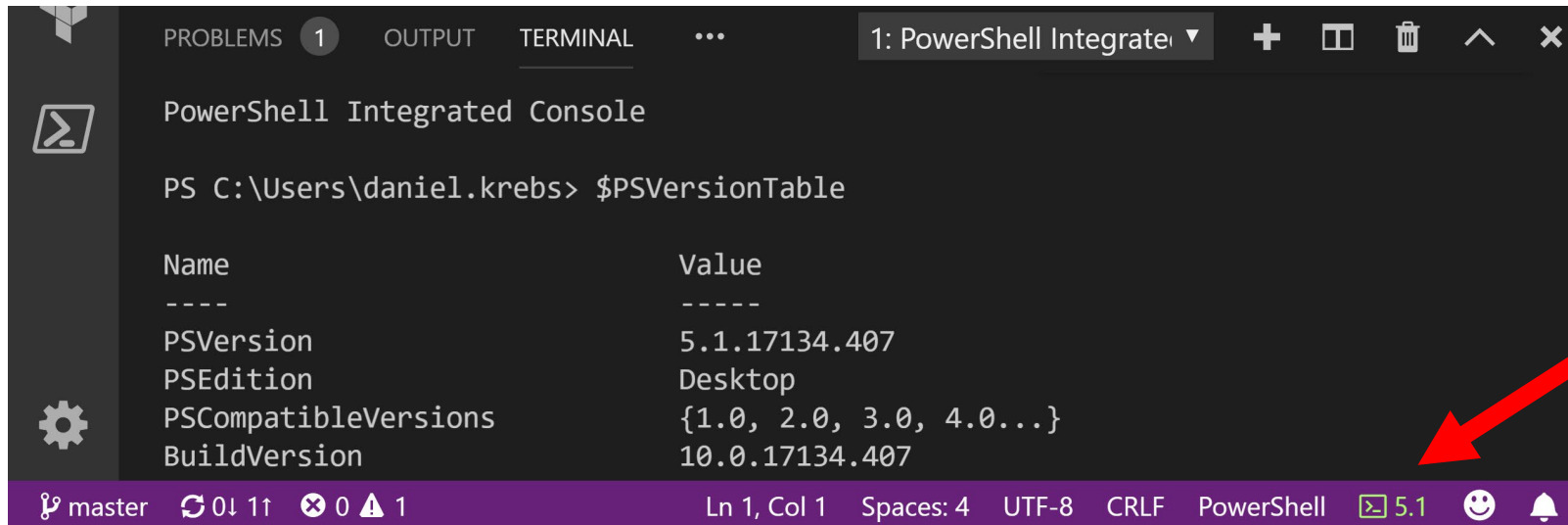
```
PS C:\> Get-Date -Format 'yyyy-MM-dd hh-mm'
```

```
2019-01-10 03-36
```



# VSCode – Switch PowerShell versions

B

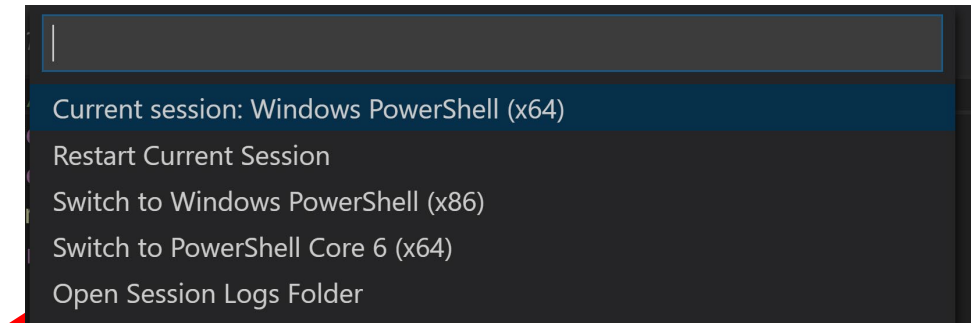


PowerShell Integrated Console

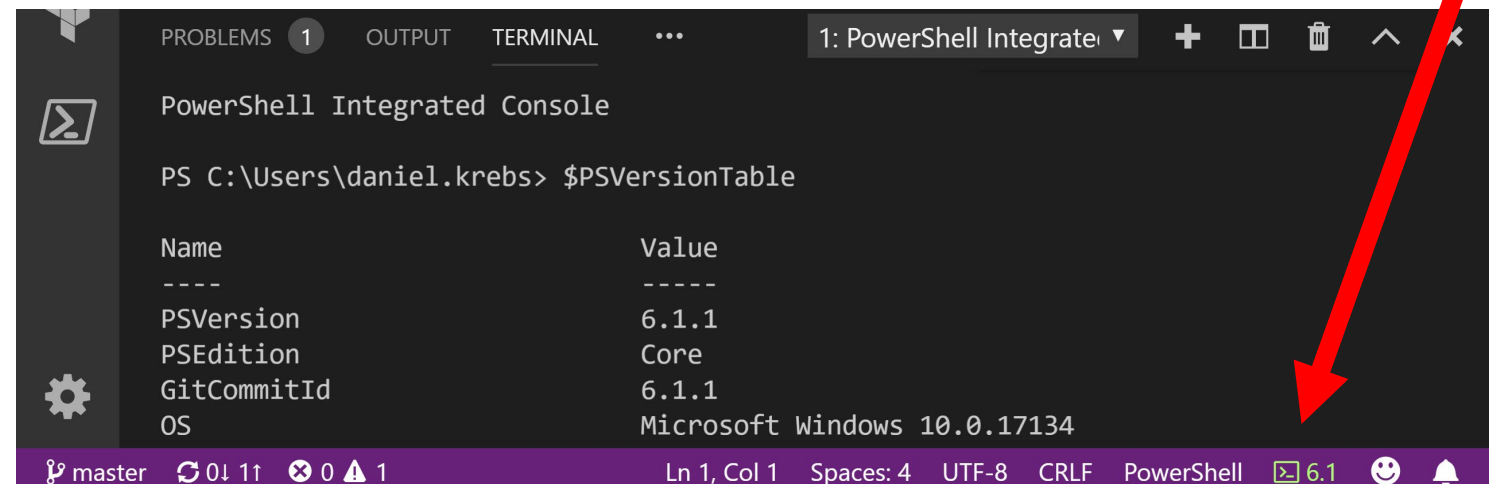
```
PS C:\Users\daniel.krebs> $PSVersionTable
```

Name	Value
PSVersion	5.1.17134.407
PSEdition	Desktop
PSCompatibleVersions	{1.0, 2.0, 3.0, 4.0...}
BuildVersion	10.0.17134.407

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PowerShell 5.1



Show PowerShell Session Menu



PowerShell Integrated Console

```
PS C:\Users\daniel.krebs> $PSVersionTable
```

Name	Value
PSVersion	6.1.1
PSEdition	Core
GitCommitId	6.1.1
OS	Microsoft Windows 10.0.17134

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PowerShell 6.1



# Install/upgrade PowerShell 6.x

B

## Chocolatey - Software Management Automation

<https://chocolatey.org/install>

To install powershell-core (Install), run the following command from the command line or from PowerShell:

```
C:\> choco install powershell-core
```

To upgrade powershell-core (Install), run the following command from the command line or from PowerShell:

```
C:\> choco upgrade powershell-core
```



# Write-Information

A

```
PS C:\> $InformationPreference  
SilentlyContinue  
PS C:\> Write-Information "Executing task YXZ ..."  
PS C:\> $InformationPreference = 'Continue'  
PS C:\> Write-Information "Executing task YXZ ..."  
Executing task YXZ ...  
PS C:\>  
PS C:\> .\FakeDemo.ps1 -Task XYZ -InformationAction Continue
```



# PS v6: Test-Connection & Progress Bar

A

```
PS C:\Users\daniel.krebs> Test-Connection -TargetName 8.8.8.8  
Pinging 8.8.8.8 [8.8.8.8] with 32 bytes of data:
```

```
    Pinging 8.8.8.8 [8.8.8.8] with 32 bytes of data:  
        Reply from 8.8.8.8: bytes=32 time=1ms TTL=122
```

```
Reply from 8.8.8.8: bytes=32 time=1ms TTL=122
```

```
PS C:\Users\daniel.krebs> $ProgressPreference
```

```
Continue
```

```
PS C:\Users\daniel.krebs> $ProgressPreference = 'SilentlyContinue'
```

```
PS C:\Users\daniel.krebs> Test-Connection -TargetName 8.8.8.8
```

```
Pinging 8.8.8.8 [8.8.8.8] with 32 bytes of data:
```

```
Reply from 8.8.8.8: bytes=32 time=1ms TTL=122
```

```
Reply from 8.8.8.8: bytes=32 time=1ms TTL=122
```



## Session Manager

AWS Systems Manager > Session Manager > Start a session

### Start a session

Select the instance that you would like to start a session on

#### Target instances



< 1 >

	Instance name	Instance ID	Agent version	Instance state	Availability zone	Platform
<input type="radio"/>	Linux2	i-01a2ad6ad86e38712	2.3.274.0	✔ running	eu-west-2a	Amazon Linux
<input type="radio"/>	Win2019-Core	i-05327dc53608a3adb	2.3.274.0	✔ running	eu-west-2a	Microsoft Windows Server 2019 Datacenter
<input type="radio"/>	Win2016-Core	i-07be8b2d1b44eeb0d	2.3.274.0	✔ running	eu-west-2a	Microsoft Windows Server 2016 Datacenter

Cancel

Start session

# AWS Systems Manager

B

## Session Manager – Managing Windows server

Session ID: root-0df81802e54af32ed Instance ID: i-07be8b2d1b44eeb0d Terminate

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> $PSVersionTable

Name                           Value
----                           -
PSVersion                      5.1.14393.2636
PSEdition                     Desktop
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
BuildVersion                   10.0.14393.2636
CLRVersion                     4.0.30319.42000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1

PS C:\Windows\system32> Get-Module -ListAvailable

Directory: C:\Program Files\WindowsPowerShell\Modules

ModuleType Version      Name                                ExportedCommands
-----
Script      1.0.1        Microsoft.PowerShell.Operation.V... {Get-OperationValidation, Invoke-OperationValidation}
Binary      1.0.0.1      PackageManagement                  {Find-Package, Get-Package, Get-PackageProvider, Get-PackageSource...}
Script      3.4.0        Pester                             {Describe, Context, It, Should...}
Script      1.0.0.1      PowerShellGet                      {Install-Module, Find-Module, Save-Module, Update-Module...}
```





# AWS Systems Manager

B

## Session Manager – Managing Linux server

Session ID: root-0eef0aeb0bc2aa89d

Instance ID: i-01a2ad6ad86e38712

Terminate

```
sh-4.2$ pwsh
PowerShell 6.1.1
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/pscore6-docs
Type 'help' to get help.

Name                           Value
----                           -
PSVersion                      6.1.1
PSEdition                      Core
GitCommitId                   6.1.1
OS                             Linux 4.14.88-88.76.amzn2.x86_64 #1 SMP Mon Jan 7 18:43:26 UTC 2019
Platform                      Unix
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
WSManStackVersion              3.0
```

```
PS /usr/bin> Get-Module -ListAvailable
```

```
Directory: /opt/microsoft/powershell/6/Modules
```

ModuleType	Version	Name	PSEdition	ExportedCommands
-----	-----	----	-----	-----



## Session Manager – Preferences

AWS Systems Manager > Session Manager

Sessions

Session history

Preferences

### Preferences

Session preferences let you specify a location to store log output for all sessions in your account. You can also enable server-side encryption using an AWS Key Management Service (KMS) key for a specified stream to ensure your session records are transferred securely.

Write session output to an Amazon S3 bucket

☐ S3 bucket

Send session output to CloudWatch Logs

☐ Cloudwatch logs

Cancel

Save

# PowerShell Call Operator & with Arguments

A

```
$command = '/usr/bin/dynamo-backup-to-s3'
$arguments = @(
    '--bucket', $BucketName
    '--backup-path', ("{0}/{1}" -f $BucketKeyPrefix,
        (Get-Date -Format FileDateTimeUniversal))
    '--included-tables', "$($tables -join ', ')"
    '--base64-encode-binary'
    '--stop-on-failure'
    '--aws-region', $TableRegion
)
Write-Information "Executing command: ${command} ${arguments}"
& $command $arguments
Write-Information "- Exit code: ${LASTEXITCODE}"
```



# PowerShell Call Operator & with Arguments

A

```
$command = '/usr/bin/dynamo-backup-to-s3'  
$arguments = @(
```

```
Executing command: /usr/bin/dynamo-backup-to-s3 --bucket powershell-demo --backup-path  
dynamodb/20190110T1103281896Z --included-tables Movies --base64-encode-binary --stop-  
on-failure --aws-region eu-west-1
```

```
Starting to copy table Movies  
Done copying table Moves. Took 0.00 minutes  
Finished backing up DynamoDB
```

```
- Exit code: 0
```

```
Write-Information "Executing command: ${command} ${arguments}"  
& $command $arguments  
Write-Information "- Exit code: ${LASTEXITCODE}"
```



# Compare file hash of downloads

B

## Download Apache OpenOffice

(Hosted by SourceForge.net - A trusted website)

Select your favorite operating system, language and version:

Windows (EXE) ▼

English [British] ▼

4.1.6 ▼

Download full installation

Download language pack

**Release:** Milestone AOO416m1 | Build ID 9790 | SVN r1844436 | Released 2018-11-18 | [Release Notes](#)

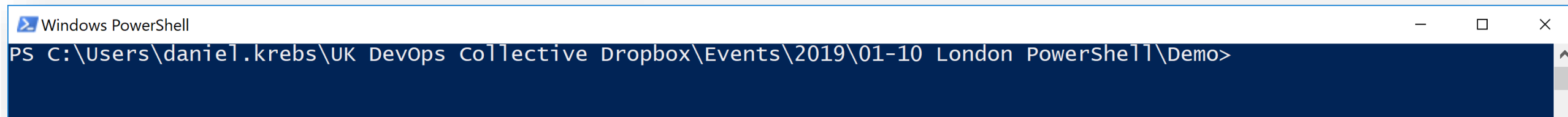
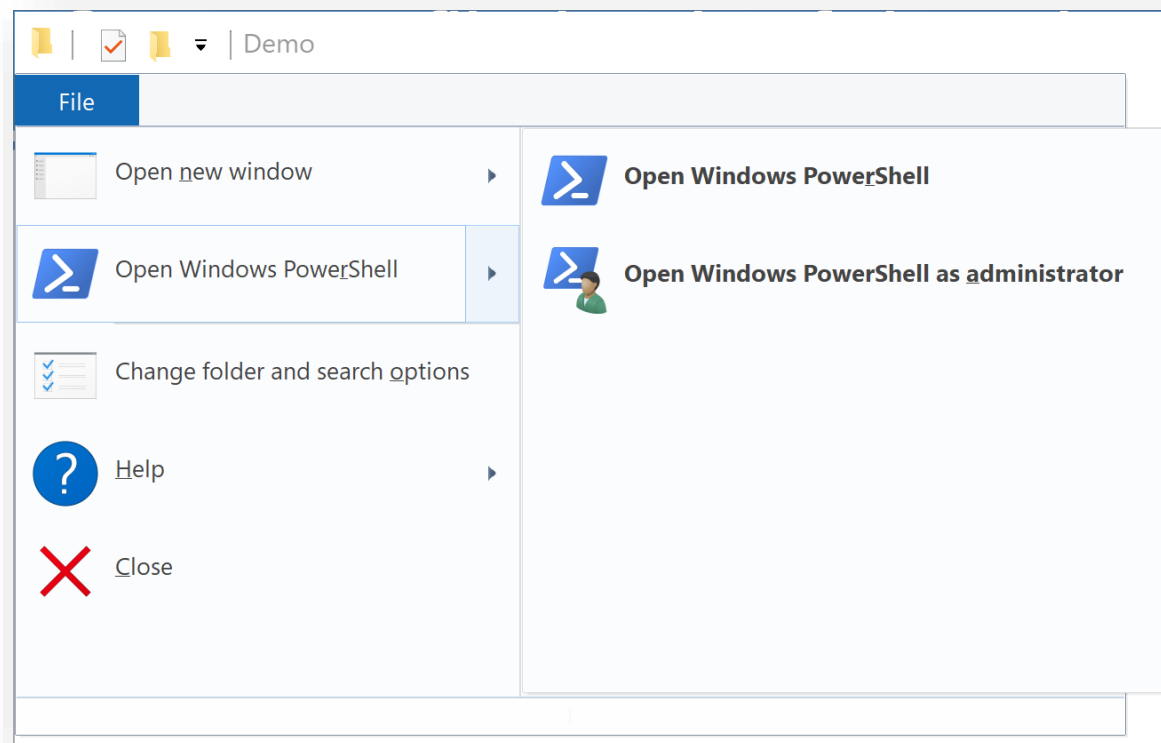
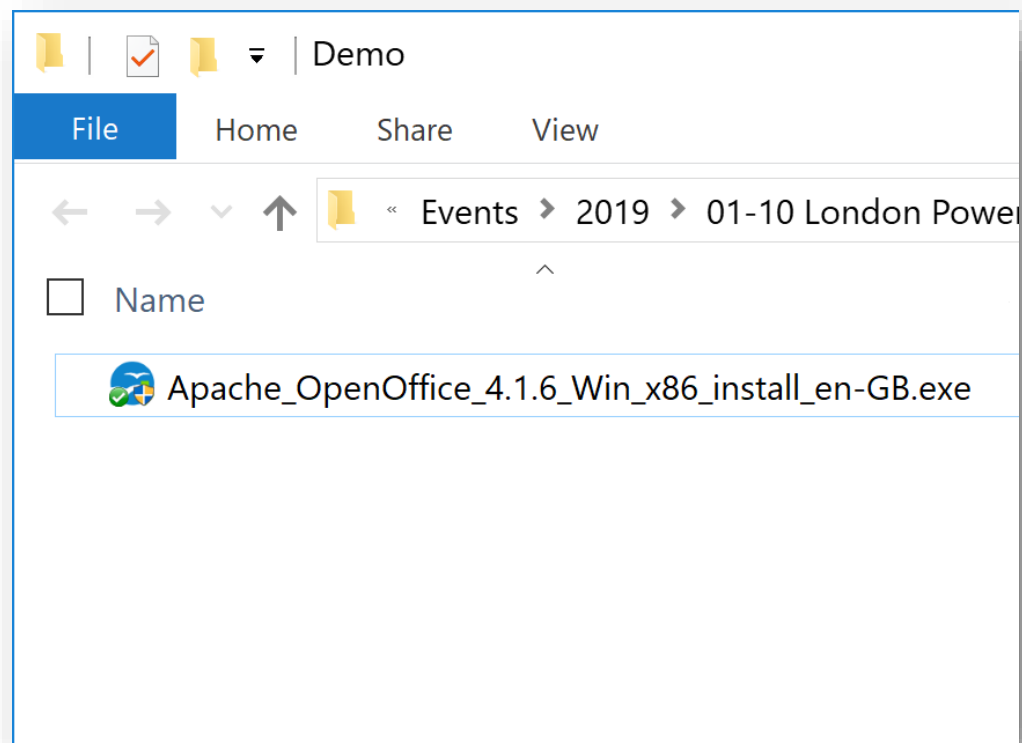
**Full installation:** File size ~ 127 MByte | Signatures and hashes: [KEYS](#) , [ASC](#) , [SHA256](#) , [SHA512](#)

c2db89f490bed788c2010bf29f90b39b4b86e61524b3a6ecedb0b54df113d4c7 \*Apache\_OpenOffice\_4.1.6\_Win\_x86\_install\_en-GB.exe



# Compare file hash of downloads

B



# Compare file hash of downloads

B

```
$hash = 'c2db89f490bed788c2010bf29f90b39b4b86e61524b3a6ecedb0b54df113d4c7'  
  
Get-FileHash -Path .\Apache_OpenOffice_4.1.6_Win_x86_install_en-GB.exe |  
    Where-Object Hash -eq $hash
```

Algorithm	Hash	Path
-----	----	----
SHA256	C2DB89F490BED788C2010BF29F90B39B4B86E61524B3A6ECEDB0B54DF113D4C7	C:\Users\daniel.krebs\UK DevOps Collective Dropbox\Events\2019\01-10 London PowerShell\Demo>



# String variable expansion

B

Single quotes

PSHOME: \$path

Double quotes - Variant 1a

PSHOME: C:\Windows\System32\WindowsPowerShell\v1.0

Double quotes - Variant 2a

PSHOME: C:\Windows\System32\WindowsPowerShell\v1.0

Double quotes - Variant 1b

PSHOME: text

Double quotes - Variant 3a

PSHOME: C:\Windows\System32\WindowsPowerShell\v1.0

Double quotes - Variant 2b

PSHOME: C:\Windows\System32\WindowsPowerShell\v1.0\_other text

Double quotes - Variant 3b

PSHOME: C:\Windows\System32\WindowsPowerShell\v1.0\_other text



# DEMO







# DEMO

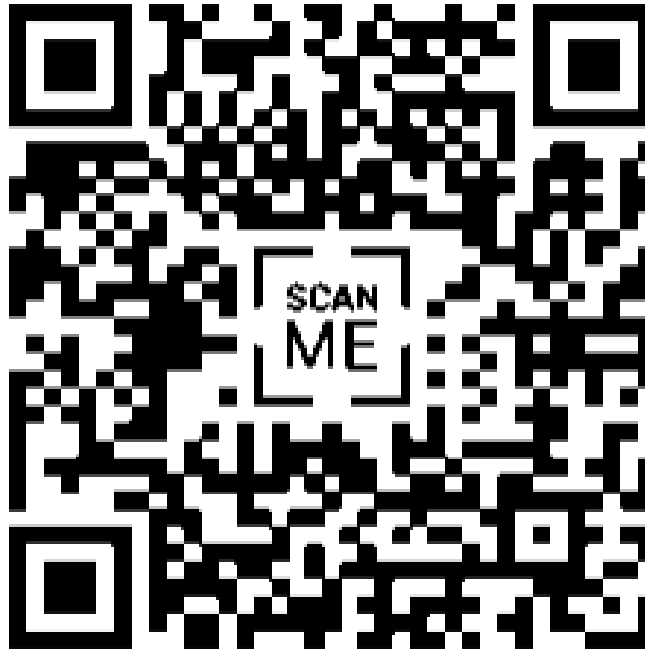


# DEMO

# THE END



JOIN US ON SLACK 



<https://slofile.com/slack/get-psuguk>