



Llywodraeth Cymru
Welsh Government

Windows 11 – One Click for Cymraeg

Mae'r ddogfen yma hefyd ar gael yn Gymraeg.
This document is also available in Welsh.



Digital ISBN **978-1-80853-498-0**
© Crown copyright October 2026
WG55454

Table of Contents

Introduction	4
Purpose of document	4
Entra ID Groups	5
File Structure	5
Set-Region-cyGB	5
Set-Region-enGB	6
Intune Applications	7
Cymraeg mewn clic	7
English in one click	7
Code Signing	9
Logging	9
Appendix	10
Set-Region-cyGB	10
Set-Region-cyGB.ps1	10
UserConfig-cyGB.ps1	14
Set-Region-enGB	16
Set-Region-enGB.ps1	16
UserConfig-enGB.ps1	20

Introduction

Purpose of document

This document is all about one simple ambition: making it easier for more people to use more Cymraeg in the technology they rely on every day. Cymraeg mewn Clic does exactly what it says on the tin - giving users a quick, straightforward way to switch their Windows device to Welsh, without needing to be a technical wizard.

By removing a small friction, the solution helps Welsh become an easy, practical choice at work and supports our wider goal of increasing the everyday use of the language across digital technology. This document is for technical architects and specialists, but Cymraeg mewn Clic is for everyone. The purpose of this document therefore is to describe the Cymraeg Mewn Clic solution which has been developed by the Welsh Government. The solution, driven by PowerShell scripts and Intune deployed applications, can be easily installed by end users through the Company Portal.

Equally, if users prefer to switch their devices back to English, this can be achieved by installing the 'English in One Click' app available through the Company Portal.

Entra ID Groups

The Entra ID group below is suggested to be created for a pilot rollout.

Group Name	Users – Welsh Language
Group Description	Users added to this Entra ID Group will have the ability to install the 'Cymraeg mewn clic' app in the Company Portal which converts many aspects of their device into Welsh. In addition, users will have the ability to install the app 'English – in One Click' in the Company Portal which converts many aspects of their device into English.

File Structure

Set-Region-cyGB

The 'Set-Region-cyGB' folder should contain the following files (all required source code is included within the appendix of this document):

Filename	Description
Set-Region-cyGB.ps1	This script (executed first) runs under the system context and is used to perform the following tasks: - Remove any existing scheduled tasks named 'LXP-UserSession-Config-V2*' as a cleanup action. - Copy 'UserConfig-cyGB.ps1' locally. - Install the cyGB language pack. - Set the system preferred UI Language. - Set the Windows UI Language override for regional changes. - Set the Windows User Language List. - Set the User Culture. - Set the Windows home location. - Copy the User International Settings from the current user to System, including Welcome screen and new users. - Create a scheduled task named 'LXP-UserSession-Config-V2-CYGB' to execute the 'UserConfig-cyGB.ps1' PowerShell script under the user context.
UserConfig-cyGB.ps1	This script executes at next logon for all users (user context script). It performs the following tasks: - Set the Windows UI Language override for regional changes. - Set the Windows User Language List. - Set the User Culture. - Set the Windows home location.

Set-Region-enGB

The 'Set-Region-enGB' folder contains the following files:

Filename	Description
Set-Region-enGB.ps1	This script (executed first) runs under the system context and is used to perform the following tasks: - Remove any existing scheduled tasks named 'LXP-UserSession-Config-V2*' as a cleanup action. - Copy 'UserConfig-enGB.ps1' locally. - Install the enGB language pack. - Set the system preferred UI Language. - Set the Windows UI Language override for regional changes. - Set the Windows User Language List. - Set the User Culture. - Set the Windows home location. - Copy the User International Settings from the current user to System, including Welcome screen and new users. - Create a scheduled task named 'LXP-UserSession-Config-V2-ENGB' to execute the 'UserConfig-cyGB.ps1' PowerShell script under the user context.
UserConfig-enGB.ps1	This script executes at next logon for all users (user context script). It performs the following tasks: - Set the Windows UI Language override for regional changes. - Set the Windows User Language List. - Set the User Culture. - Set the Windows home location.

Intune Applications

The Microsoft Win32 Content Prep Tool must be used to create '.intunewin' files for each respective application using the file structures described above. Once a '.intunewin' file has been created for each language application, the respective Intune applications below can be created.

Cymraeg mewn clic

Application Name	Cymraeg mewn clic
Description	This application will replace any existing language configuration you have set on your device with the Welsh (CY-GB) language. It will require a manual reboot once installation is complete.
Install Command	Powershell.exe -ExecutionPolicy ByPass -File .\Set-Region-cyGB.ps1
Install behaviour	System
Rules format	Manually configure detection rules
Detection Rules	(Registry) Key Path: HKLM\Software\<CompanyName>\LanguageXPWIN11\ModifiedLanguage Value Name: SetLanguage-cy-GB Detection Method: Integer Comparison Operator: Equals Value: 1
Assignment	Users – Welsh Language (for pilot), All Users (for mass rollout)

English in one click

Application Name	English in one click
Description	This application will replace any existing language configuration you have set on your device with the English (EN-GB) language. It will require a manual reboot once installation is complete.
Install Command	Powershell.exe -ExecutionPolicy ByPass -File .\Set-Region-enGB.ps1
Install behaviour	System
Rules format	Manually configure detection rules
Detection Rules	(Registry) Key Path: HKLM\Software\<CompanyName>\LanguageXPWIN11\ModifiedLanguage

	Value Name: SetLanguage-en-GB Detection Method: Integer Comparison Operator: Equals Value: 1
Assignment	Users – Welsh Language (for pilot), All Users (for mass rollout)

Code Signing

The UK National Cyber Security Center recommends that PowerShell execution Policies are set to 'All Signed'. As a result PowerShell scripts that execute in the user context within this solution (UserConfig-cyGB.ps1) must be code signed.

Logging

Logs relating to the initial PowerShell script (executed in the system context) are written locally on the device to the following location:

C:\ProgramData\Microsoft\IntuneManagementExtension\Logs

The log files will be called 'Set-Region-cygb' and 'Set-Region-engb'.

In addition, logs relating to the scheduled task (executed in the user context) are written to the following location:

%LOCALAPPDATA%\Temp

Appendix

Set-Region-cyGB

Set-Region-cyGB.ps1

```
<#
.SYNOPSIS
    Script to set the region settings to cy-GB.
.DESCRIPTION
    This script (executed first) runs under the system context and is used to perform
    the following tasks:
    - Remove any existing scheduled tasks named 'LXP-UserSession-Config-V2*'
    as a cleanup action.
    - Copy 'UserConfig-cyGB.ps1' locally.
    - Install the cyGB language pack.
    - Set the system preferred UI Language.
    - Set the Windows UI Language override for regional changes.
    - Set the Windows User Language List.
    - Set the User Culture.
    - Set the Windows home location.
    - Copy the User International Settings from the current user to System,
    including Welcome screen and new users.
    - Create a scheduled task named 'LXP-UserSession-Config-V2-CYGB' to
    execute the 'UserConfig-cyGB.ps1' PowerShell script under the user context.

.NOTES
    File Name   : Set-Region-cyGB.ps1
    Author      : Gavin James
    Version     : v4
    Last Updated : 28/01/2025
#>

#Create the EventLog Source if it does not already exist
If (![system.diagnostics.eventlog]::SourceExists('Set_Region')){
    New-EventLog -LogName Application -Source "Set_Region"
}

Write-EventLog -LogName Application -Source "Set_Region" -EventID 9681 -
EntryType Information -Message "The PowerShell script to set Region settings to
cy-GB was executed."

# Microsoft Intune Management Extension might start a 32-bit PowerShell
instance. If so, restart as 64-bit PowerShell
If ($ENV:PROCESSOR_ARCHITEW6432 -eq "AMD64") {
    Try {
        &"$ENV:WINDIR\SysNative\WindowsPowershell\v1.0\PowerShell.exe" -File
        $PSCOMMANDPATH
    }
    Catch {
```

```

    Throw "Failed to start $PSCOMMANDPATH"
}
Exit
}

#Set variables:
#Company name
$CompanyName = "<CompanyName>"
# The language we want as new default. Language tag can be found here:
https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/available-language-packs-for-windows
$language = "cy-GB"

#Second language added to accomodate the inconsistent languages returned from
Get-SystemPreferredUILanguage
$language2 = "cy"

# Geographical ID we want to set. GeoID can be found here:
https://learn.microsoft.com/en-us/windows/win32/intl/table-of-geographical-locations?redirectedfrom=MSDN
$geoid = "242" # United Kingdom

# Start Transcript
Start-Transcript -Path
"$env:ProgramData\Microsoft\IntuneManagementExtension\Logs\$($Split-Path
$PSCommandPath -Leaf).ToLower().Replace(".ps1",".log"))" | Out-Null

#Remove existing Language pack scheduled tasks.
"...Remove existing Language pack scheduled tasks."
Get-ScheduledTask | Where-Object TaskName -like "LXP-UserSession-Config-V2*"
| Unregister-ScheduledTask -Confirm:$false

# custom folder for temp scripts
"...creating custom temp script folder"
$scriptFolderPath =
"$env:SystemDrive\ProgramData\$CompanyName\CustomTempScripts"
New-Item -ItemType Directory -Force -Path $scriptFolderPath
""

$UserConfigScriptPath = $(Join-Path -Path $scriptFolderPath -ChildPath
"UserConfig-cyGB.ps1")
"...moving userconfig scripts"
# Due to the code-signing restrictions I have pre-created the UserConfig.ps1 script
and signed using a code signing certificate.
# This script must be moved to the required folder.
$UserConfigScript = "UserConfig-cyGB.ps1"
$SourceFolder = $PSScriptRoot
$TargetFolder = $UserConfigScript

```

```

Write-host "The PSScriptRoot variable is $PSScriptRoot"
Copy-Item -Path "$SourceFolder\$UserConfigScript" -Destination
$UserConfigScriptPath -Recurse -Force -ErrorAction Stop

# Install an additional language pack including FODs
"Installing languagepack"
Install-Language $language -CopyToSettings

#Set System Preferred UI Language
"Set SystemPreferredUILanguage"
Set-SystemPreferredUILanguage $language

#Check status of the installed language pack
"Checking installed languagepack status"
$installedLanguage = (Get-InstalledLanguage).LanguageId

if ($installedLanguage -like $language){
    Write-Host "Language $language installed"
}
else {
    Write-Host "Failure! Language $language NOT installed"
    Exit 1
}

#Check status of the System Preferred Language
$SystemPreferredUILanguage = Get-SystemPreferredUILanguage

if (($SystemPreferredUILanguage -like $language) -or
($SystemPreferredUILanguage -like $language2)){
    Write-Host "System Preferred UI Language set to $language. OK"
}
else {
    Write-Host "Failure! System Preferred UI Language NOT set to $language.
System Preferred UI Language is $SystemPreferredUILanguage"
    Exit 1
}

# Configure new language defaults under current user (system account) after
which it can be copied to the system
#Set Win UI Language Override for regional changes
"Set WinUILanguageOverride"
Set-WinUILanguageOverride -Language $language

# Set Win User Language List, sets the current user language settings
"Set WinUserLanguageList"
$OldList = Get-WinUserLanguageList
$UserLanguageList = New-WinUserLanguageList -Language $language

```

```

$UserLanguageList += $OldList | where { $_.LanguageTag -ne $language }
$UserLanguageList | select LanguageTag
Set-WinUserLanguageList -LanguageList $UserLanguageList -Force

# Set Culture, sets the user culture for the current user account.
"Set culture"
Set-Culture -CultureInfo $language

# Set Win Home Location, sets the home location setting for the current user
"Set WinHomeLocation"
Set-WinHomeLocation -Geold $geold

# Copy User International Settings from current user to System, including
Welcome screen and new user
"Copy UserInternationalSettingsToSystem "
Copy-UserInternationalSettingsToSystem -WelcomeScreen $True -NewUser $True

$taskName = "LXP-UserSession-Config-V2-$language"
$action = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-
ExecutionPolicy Bypass -File ""$userConfigScriptPath"
$trigger = New-ScheduledTaskTrigger -AtLogOn
$principal = New-ScheduledTaskPrincipal -UserId (Get-CimInstance -ClassName
Win32_ComputerSystem | Select-Object -expand UserName)
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries
$task = New-ScheduledTask -Action $action -Trigger $trigger -Principal $principal -
Settings $settings
Register-ScheduledTask $taskName -InputObject $task
Start-ScheduledTask -TaskName $taskName

Start-Sleep -Seconds 30

Unregister-ScheduledTask -TaskName $taskName -Confirm:$false

# trigger 'LanguageComponentsInstaller\ReconcileLanguageResources' otherwise
'Windows Settings' need a long time to change finally
"Trigger ScheduledTask =
LanguageComponentsInstaller\ReconcileLanguageResources"
Start-ScheduledTask -TaskName
"\Microsoft\Windows\LanguageComponentsInstaller\ReconcileLanguageResource
s"

Start-Sleep 10

#Remove previous language pack UserConfig Scripts
Get-Childitem $scriptFolderPath | Where-Object Name -ne $userConfigScript |
Remove-Item

#Remove previous language pack registry keys.

```

```

Remove-Item
"Registry::HKLM\Software\$CompanyName\LanguageXPWIN11\ModifiedLanguage" -Recurse -Force

# Add registry key for Intune detection
"Add registry key for Intune detection"
REG add
"HKLM\Software\$CompanyName\LanguageXPWIN11\ModifiedLanguage" /v
"SetLanguage-$language" /t REG_DWORD /d 1

# trigger store updates, there might be new app versions due to the language
change
$UpdateCommand = "-command {Get-CimInstance -Namespace
`"root\cimv2\mdm\dmmap`" -ClassName
`"MDM_EnterpriseModernAppManagement_AppManagement01`" | Invoke-
CimMethod -MethodName `"UpdateScanMethod`"}"

#Ref: https://learn.microsoft.com/en-
us/powershell/module/scheduledtasks/new-
scheduledtaskprincipal?view=windowsserver2025-ps

# Scheduled Task name to run the language pack update
$UpdateTaskName = "LXP-Update-Config-$language"

Unregister-ScheduledTask -TaskName $UpdatetaskName -Confirm:$false

$UpdateAction = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument
$UpdateCommand
$UpdateTrigger = New-ScheduledTaskTrigger -AtLogOn
$UpdatePrincipal = New-ScheduledTaskPrincipal -UserId "LOCALSERVICE" -
LogonType ServiceAccount
$UpdateSettings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries
$UpdateTask = New-ScheduledTask -Action $UpdateAction -Trigger
$UpdateTrigger -Principal $UpdatePrincipal -Settings $UpdateSettings
Register-ScheduledTask $UpdateTaskName -InputObject $UpdateTask
Start-ScheduledTask -TaskName $UpdateTaskName

Exit 0
Stop-Transcript

```

UserConfig-cyGB.ps1

```

$language = "cy-GB"

Start-Transcript -Path "$env:TEMP\LXP-UserSession-Config-$language.log" | Out-
Null

$geold = 242

```

```
# important for regional change like date and time...
"Set-WinUILanguageOverride = $language"
Set-WinUILanguageOverride -Language $language

"Set-WinUserLanguageList = $language"

$OldList = Get-WinUserLanguageList
$UserLanguageList = New-WinUserLanguageList -Language $language
$UserLanguageList += $OldList | where { $_.LanguageTag -ne $language }
"Setting new user language list:"
$UserLanguageList | select LanguageTag
""
"Set-WinUserLanguageList -LanguageList ..."
Set-WinUserLanguageList -LanguageList $UserLanguageList -Force

"Set-Culture = $language"
Set-Culture -CultureInfo $language

"Set-WinHomeLocation = $geold"
Set-WinHomeLocation -Geold $geold

Stop-Transcript -Verbose
```

Set-Region-enGB

Set-Region-enGB.ps1

```
<#
.SYNOPSIS
    Script to set the region settings to EN-GB.
.DESCRIPTION
    This script (executed first) runs under the system context and is used to perform
    the following tasks:
    - Remove any existing scheduled tasks named 'LXP-UserSession-Config-V2*'
    as a cleanup action.
    - Copy 'UserConfig-enGB.ps1' locally.
    - Install the enGB language pack.
    - Set the system preferred UI Language.
    - Set the Windows UI Language override for regional changes.
    - Set the Windows User Language List.
    - Set the User Culture.
    - Set the Windows home location.
    - Copy the User International Settings from the current user to System,
    including Welcome screen and new users.
    - Create a scheduled task named 'LXP-UserSession-Config-V2-ENGB' to
    execute the 'UserConfig-cyGB.ps1' PowerShell script under the user context.

.NOTES
    File Name   : Set-Region-engb.ps1
    Author      : Gavin James
    Version     : v4
    Last Updated : 29/01/2025
#>

#Create the EventLog Source if it does not already exist
If (![system.diagnostics.eventlog]::SourceExists('Set_Region')){
    New-EventLog -LogName Application -Source "Set_Region"
}

Write-EventLog -LogName Application -Source "Set_Region" -EventID 9681 -
EntryType Information -Message "The PowerShell script to set Region settings to
en-gb was executed."

# Microsoft Intune Management Extension might start a 32-bit PowerShell
instance. If so, restart as 64-bit PowerShell
If ($ENV:PROCESSOR_ARCHITEW6432 -eq "AMD64") {
    Try {
        &"$ENV:WINDIR\SysNative\WindowsPowershell\v1.0\PowerShell.exe" -File
$PSCOMMANDPATH
    }
    Catch {
        Throw "Failed to start $PSCOMMANDPATH"
    }
}
```

```

Exit
}

#Set variables:
#Company name
$CompanyName = "<CompanyName>"
# The language we want as new default. Language tag can be found here:
https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/available-language-packs-for-windows
$language = "en-GB"
# Geographical ID we want to set. GeoID can be found here:
https://learn.microsoft.com/en-us/windows/win32/intl/table-of-geographical-locations?redirectedfrom=MSDN
$geold = "242" # United Kingdom

# Start Transcript
Start-Transcript -Path
"$env:ProgramData\Microsoft\IntuneManagementExtension\Logs\$(($($Split-Path
$PSCmdPath -Leaf).ToLower().Replace(".ps1",".log"))" | Out-Null

#Remove existing Language pack scheduled tasks.
"...Remove existing Language pack scheduled tasks."
Get-ScheduledTask | Where-Object TaskName -like "LXP-UserSession-Config-V2*"
| Unregister-ScheduledTask -Confirm:$false

# custom folder for temp scripts
"...creating custom temp script folder"
$scriptFolderPath =
"$env:SystemDrive\ProgramData\$CompanyName\CustomTempScripts"
New-Item -ItemType Directory -Force -Path $scriptFolderPath
""

$userConfigScriptPath = $(Join-Path -Path $scriptFolderPath -ChildPath
"userConfig.ps1")
"...moving userconfig scripts"
# Due to the code-signing restrictions I have pre-created the UserConfig.ps1 script
and signed using a code signing certificate.
# This script must be moved to the required folder.
$userConfigScript = "UserConfig-engb.ps1"
$sourceFolder = $PSScriptRoot
$targetFolder = $userConfigScript

Write-host "The PSScriptRoot variable is $PSScriptRoot"
Copy-Item -Path "$sourceFolder\$userConfigScript" -Destination
$userConfigScriptPath -Recurse -Force -ErrorAction Stop

# Install an additional language pack including FODs
"Installing languagepack"

```

```

Install-Language $language -CopyToSettings

#Set System Preferred UI Language
"Set SystemPreferredUILanguage"
Set-SystemPreferredUILanguage $language

#Check status of the installed language pack
"Checking installed languagepack status"
$installedLanguage = (Get-InstalledLanguage).LanguageId

if ($installedLanguage -like $language){
    Write-Host "Language $language installed"
}
else {
    Write-Host "Failure! Language $language NOT installed"
    Exit 1
}

#Check status of the System Preferred Language
$SystemPreferredUILanguage = Get-SystemPreferredUILanguage

if ($SystemPreferredUILanguage -like $language){
    Write-Host "System Preferred UI Language set to $language. OK"
}
else {
    Write-Host "Failure! System Preferred UI Language NOT set to $language.
System Preferred UI Language is $SystemPreferredUILanguage"
    Exit 1
}

# Configure new language defaults under current user (system account) after
which it can be copied to the system
#Set Win UI Language Override for regional changes
"Set WinUILanguageOverride"
Set-WinUILanguageOverride -Language $language

# Set Win User Language List, sets the current user language settings
"Set WinUserLanguageList"
$OldList = Get-WinUserLanguageList
$UserLanguageList = New-WinUserLanguageList -Language $language
$UserLanguageList += $OldList | where { $_.LanguageTag -ne $language }
$UserLanguageList | select LanguageTag
Set-WinUserLanguageList -LanguageList $UserLanguageList -Force

# Set Culture, sets the user culture for the current user account.
"Set culture"
Set-Culture -CultureInfo $language

```

```

# Set Win Home Location, sets the home location setting for the current user
"Set WinHomeLocation"
Set-WinHomeLocation -Geoid $geoid

# Copy User International Settings from current user to System, including
Welcome screen and new user
"Copy UserInternationalSettingsToSystem "
Copy-UserInternationalSettingsToSystem -WelcomeScreen $True -NewUser $True

$taskName = "LXP-UserSession-Config-V2-$language"
$action = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-
ExecutionPolicy Bypass -File ""$userConfigScriptPath"
$trigger = New-ScheduledTaskTrigger -AtLogOn
$principal = New-ScheduledTaskPrincipal -UserId (Get-CimInstance -ClassName
Win32_ComputerSystem | Select-Object -expand UserName)
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries
$task = New-ScheduledTask -Action $action -Trigger $trigger -Principal $principal -
Settings $settings
Register-ScheduledTask $taskName -InputObject $task
Start-ScheduledTask -TaskName $taskName

Start-Sleep -Seconds 30

Unregister-ScheduledTask -TaskName $taskName -Confirm:$false

# trigger 'LanguageComponentsInstaller\ReconcileLanguageResources' otherwise
'Windows Settings' need a long time to change finally
"Trigger ScheduledTask =
LanguageComponentsInstaller\ReconcileLanguageResources"
Start-ScheduledTask -TaskName
"\Microsoft\Windows\LanguageComponentsInstaller\ReconcileLanguageResource
s"

Start-Sleep 10

#Remove previous language pack UserConfig Scripts
Get-ChildItem $scriptFolderPath | Where-Object Name -ne $userConfigScript |
Remove-Item

#Remove previous language pack registry keys.
Remove-Item
"Registry::HKLM\Software\$CompanyName\LanguageXPWIN11\ModifiedLanguag
e" -Recurse -Force

# Add registry key for Intune detection
"Add registry key for Intune detection"

```

```

REG add
"HKLM\Software\$CompanyName\LanguageXPWIN11\ModifiedLanguage" /v
"SetLanguage-$language" /t REG_DWORD /d 1

# trigger store updates, there might be new app versions due to the language
change
$UpdateCommand = "-command {Get-CimInstance -Namespace
`"root\cimv2\mdm\dmmap`" -ClassName
`"MDM_EnterpriseModernAppManagement_AppManagement01`" | Invoke-
CimMethod -MethodName `"UpdateScanMethod`"}"

#Ref: https://learn.microsoft.com/en-
us/powershell/module/scheduledtasks/new-
scheduledtaskprincipal?view=windowsserver2025-ps

# Scheduled Task name to run the language pack update
$UpdateTaskName = "LXP-Update-Config-$language"

Unregister-ScheduledTask -TaskName $UpdatetaskName -Confirm:$false

$UpdateAction = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument
$UpdateCommand
$UpdateTrigger = New-ScheduledTaskTrigger -AtLogOn
$UpdatePrincipal = New-ScheduledTaskPrincipal -UserId "LOCALSERVICE" -
LogonType ServiceAccount
$UpdateSettings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries
$UpdateTask = New-ScheduledTask -Action $UpdateAction -Trigger
$UpdateTrigger -Principal $UpdatePrincipal -Settings $UpdateSettings
Register-ScheduledTask $UpdateTaskName -InputObject $UpdateTask
Start-ScheduledTask -TaskName $UpdateTaskName

Exit 0
Stop-Transcript

```

UserConfig-enGB.ps1

```

$language = "en-GB"

Start-Transcript -Path "$env:TEMP\LXP-UserSession-Config-$language.log" | Out-
Null

$geoid = 242

# important for regional change like date and time...
"Set-WinUILanguageOverride = $language"
Set-WinUILanguageOverride -Language $language

"Set-WinUserLanguageList = $language"

```

```
$OldList = Get-WinUserLanguageList
$UserLanguageList = New-WinUserLanguageList -Language $language
$UserLanguageList += $OldList | where { $_.LanguageTag -ne $language }
"Setting new user language list:"
$UserLanguageList | select LanguageTag
""
"Set-WinUserLanguageList -LanguageList ..."
Set-WinUserLanguageList -LanguageList $UserLanguageList -Force

"Set-Culture = $language"
Set-Culture -CultureInfo $language

"Set-WinHomeLocation = $geold"
Set-WinHomeLocation -Geold $geold

Stop-Transcript -Verbose
```