



Dennis Span

Automating the virtual environment

Managing Windows Languages and Language Packs

Posted on **August 30, 2018**

Managing Windows languages and language packs on Windows 10 and Server 2016 can be a bit of a daunting task. This article aims to clarify most issues.

Important: as of Windows 10 version **1809**, Microsoft changed the way how to manage the installation of language packs. Please see Patrick van den Born's article [WVD and VDI automation: Change in language pack installation on Windows 10 1809 and higher](#) for more information.

Change Log

01.09.2018: added information on [Features on Demand](#) (languages for handwriting, speech, etc.).
27.12.2018: small error correction in the [installation directory](#) of language packs (I wrote `%SystemDrive%` instead of `%SystemRoot%`). Added [clarification](#) that changing the language for Windows Apps, Edge and Internet Explorer concerns the language displayed within the application and not the language of the application itself.

Great deals are waitin

Ad At our store or at your
get the latest tech quickly

Best Buy

Shop Now.



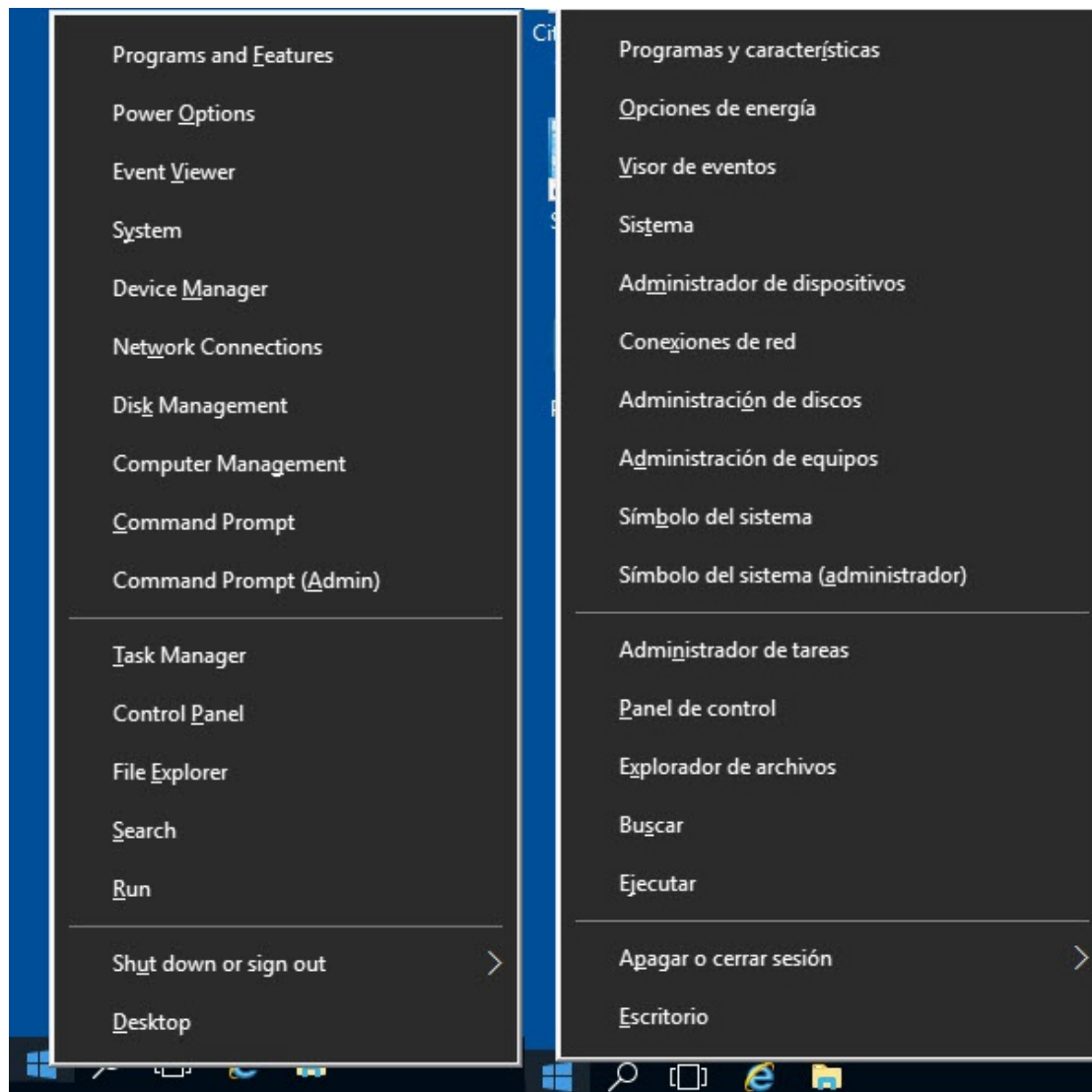
Table of Contents

- [An introduction to language packs](#)
 - [How to obtain a language pack \(for offline deployment\)](#)
 - [Windows Server 2016 Language Pack](#)
 - [Windows 10 version 1803 Language Pack](#)
 - [Windows 10 version 1803 Features on Demand](#)
 - [Adding a language pack to Windows](#)
 - [Complete installation script](#)
 - [How to change the Windows display language](#)
 - [Change the display language manually \(Windows Server 2016\)](#)
 - [Change the display language manually \(Windows 10\)](#)
 - [Change the display language using Microsoft Group Policy](#)
 - [Change the display language using a Group Policy Preference item](#)
 - [Using a logon script](#)
 - [How to change the display language for Windows Apps and Store, Edge and Internet Explorer](#)
 - [Manually \(Windows Server 2016\)](#)
 - [Manually \(Windows 10\)](#)
 - [Group Policy Preference](#)
 - [Using a logon script](#)
 - [How to prevent users from changing language settings](#)
 - [How to change the system code page for non unicode programs](#)
-

An introduction to language packs

A language pack allows you to display Windows in a different language besides English. All windows and menus are displayed in your locale language.





This is not the same as configuring local settings such as keyboard layout, date, time and number formatting. You can configure these locale settings without changing the Windows language. See also some of my other articles:

- [Configuring Regional Settings and Windows locales with Group Policy](#)
- [Configuring the time zone and code page with Group Policy](#)

As far as I can tell, language packs are available for all editions of Windows 10 and Windows Server 2016. To help you reduce the size of your image, language packs in Windows 10 are split into the following [language components](#) and [Features On Demand](#):

- **Language (interface) pack**
 - UI Text (this includes language packs, language interface packs and WinPE language packs which are all available in one download source)
- **Features on Demand (additional language packs)**
 - Basic (spell check, typing)
 - Fonts
 - Handwriting (for devices with pen inputs)



- Optical character recognition
- Text-to-speech (required for Cortana)
- Speech (required for Cortana)
- Retail Demo experience

Reference: [Language Packs in Windows 10 \(microsoft.com\)](https://dennisspan.com/managing-windows-languages-and-language-packs/)

A language (interface) pack, named *UI Text* in the above list, changes the display language for the following components:

- The operating system (all text in windows and menus)
- Windows apps and Windows Store
- Internet Explorer
- Edge (applies to Windows 10 only since Windows server has no Edge)
- WinPE (included in the same download source, but has to be used in a WinPE image)

Additional language packs, the so-called *Features on Demand*, are available for example for spelling, handwriting and speech recognition (Cortana) and more. These have to be **downloaded separately** and are available for Windows 10 only. They are NOT included within the source files of a language pack. More details are provided later on in this section.

Besides language packs for the operating system, Microsoft offer language packs for applications such as Microsoft Office. These have to be downloaded separately. The availability of language packs for third-party software depends on the vendor. Acrobat Reader for example support **more than 30 languages**. These are not within the scope of this article. In this article I only deal with language packs and the *Features on Demand*.

Please be aware that the list of available languages can differ for individual components. For example, the available language packs for Windows and for Office do not have to match. The same goes for third-party software. When planning your language strategy you should count with the fact that you will not find all languages for all products.

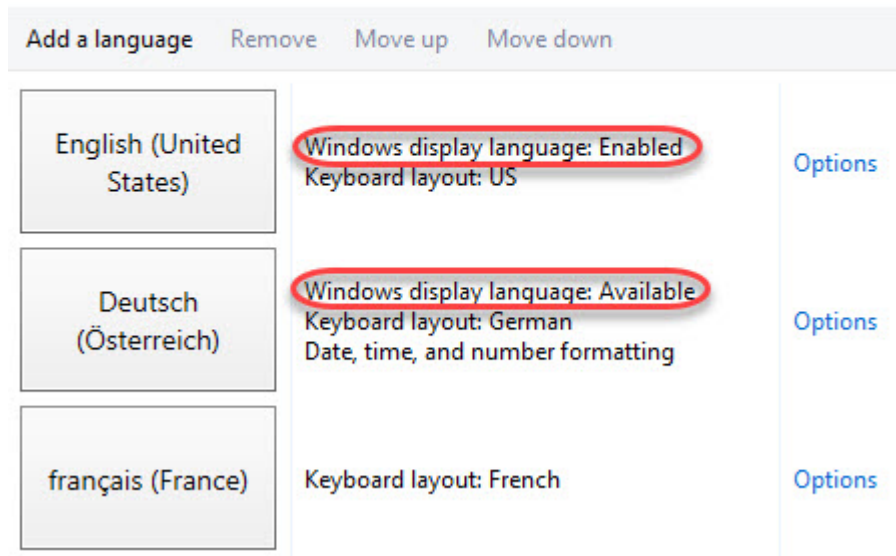
By default, when you add a language in Windows, only the keyboard layout is added. The actual Windows display language (the language (interface) pack) has to be downloaded separately. The same goes for additional language packs for handwriting, speech and so forth.

Take a look at the following screenshot. For English and German, the Windows display languages are available (English is currently enabled). For French however, only the keyboard layout is available, not the Windows display language.



Change your language preferences

You can type in any language you add to the list. Windows, apps and websites will appear in the first language in the list that they support.



Note: the screenshot above was taken from a Windows Server 2016 version 1607 machine. In Windows 10 version 1803, the menu and windows to manage languages look completely different (see the section [Change the display language manually \(Windows 10\)](#) for more information and screenshots).

Microsoft separates between a **language pack** (LP) and a **language interface pack** (LIP). As [per Microsoft](#), a LIP is a "high-quality, localized "skin" for emerging or minority language markets. [...] A LIP provides the desktop user with an approximately 80% localized user experience by translating a reduced set of user interface (UI) elements. A LIP [...] has a dependency on a base language pack of Windows. The difference between an LP and a LIP is "the level of localization in comparison to language pack (LP) packages: LIP packages provide the desktop user with the most frequently accessed user interface and basic user assistance support (help files). In addition, a LIP is installed as a language add-on on top of an existing LP with base language dependency (Catalan LIP can only be installed on top of the Spanish or French LP, Welsh LIP can only be installed on top of the English LP). In addition, once a LIP is installed, switching the user interface between the LIP language and the LP base language is possible for users on all versions of Windows.

Also, as [per Microsoft](#); "Windows Server and Windows 10 language packs are not interchangeable. Windows Server language packs cannot be used on Windows 10, and Windows 10 language packs cannot be used on Windows Server."

In total, there are 38 language packs available for Windows 10 and Windows Server 2016. In comparison, for Windows Server 2012 R2, only 18 language packs are available.

In total, there are 72 language interface packs (LIPs) available for Windows 10. LIPs are not supported on Windows Server. So keep this in mind when you have both notebooks/desktops and for example Microsoft Remote Desktop Services or Citrix Virtual Apps and Desktops (previously known as XenApp or XenDesktop) in your organization. You may not be able to install the same languages on all your work



A language pack is installed in the directory %SystemRoot%\System32\%Language-ID%, so for example C:\Windows\System32\es-ES. The size of a language pack is about 50 MB. In case you want to install all 38 language packs, you increase the size of your image with about 2 GB. Language interface packs only use up around 10 MB of space. *Features on Demand* are between 5 and 70 MB in size. Most of them are around 10 to 20 MB though.

For a complete list of available LPs and LIPs for Windows Server 2012 R2, Windows 10 and Windows Server 2016 see the Microsoft article [Available Language Packs for Windows](#).

How to obtain a language pack

There are four ways how to get a language (interface) pack and the *Features on Demand*:

- **For offline use:**
 - OEMs and system builders with Microsoft Software License Terms can download language packs and LIPs from the [Microsoft OEM site](#) or the [OEM Partner Center](#).
 - IT professionals can download language packs from the [Microsoft Volume Licensing Site](#).
 - IT professionals can download language packs from [Visual Studio Subscriptions](#) (previously MSDN).
- **Manually:**
 - After Windows is installed, end users can download and install additional language packs in **Settings > Time & language > Region and language > Add a language** (under *Options*).

Reference: [Where to get language packs and LIPs \(microsoft.com\)](#)

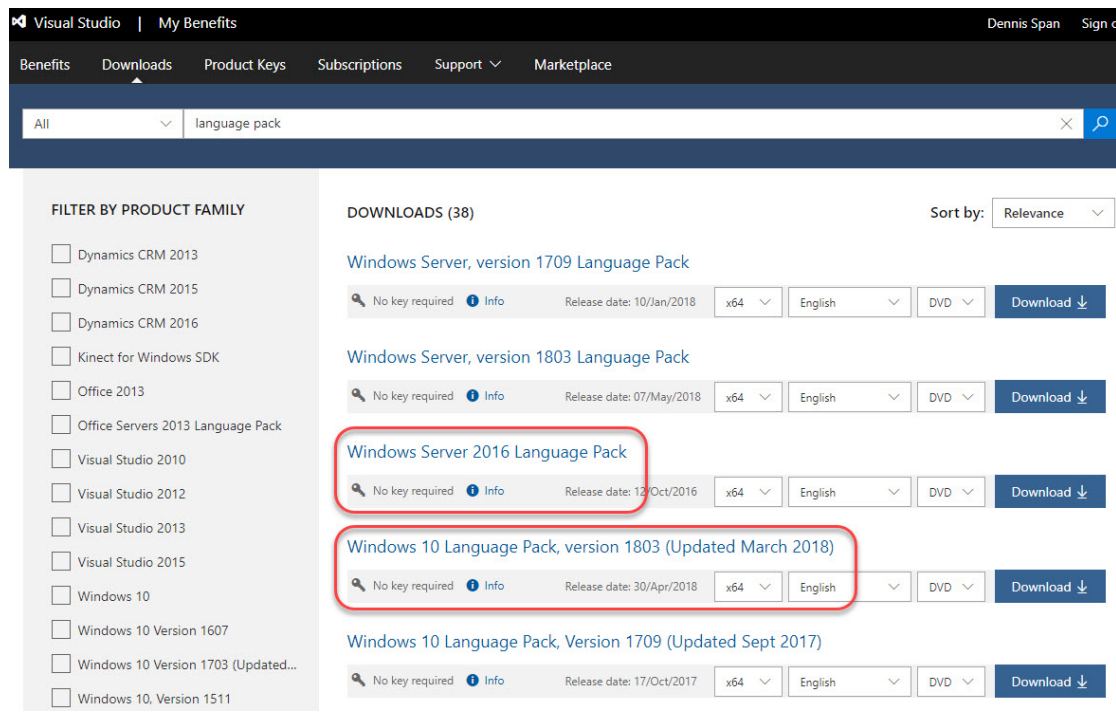
IT professionals require the offline installation sources to be able to deploy the language packs using the software deployment tool of their choice (e.g. MDT, SCCM, Citrix App Layering). To download the language packs I used my Visual Studio Subscription (MSDN). Select one of the links below for more details:

- [Windows Server 2016 Language Pack](#)
- [Windows 10 version 1803 Language Pack](#)
- [Windows 10 version 1803 Features on Demand](#)

Windows Server 2016 Language Pack

The download for *Windows Server 2016 Language Pack* is a single ISO file (about 2,6 GB in size) that contains all 38 languages. On the [Microsoft Volume License Site](#) or on the [Microsoft Visual Studio Subscription](#) website, search for *language pack* and you will find all language packs for all supported operating systems.



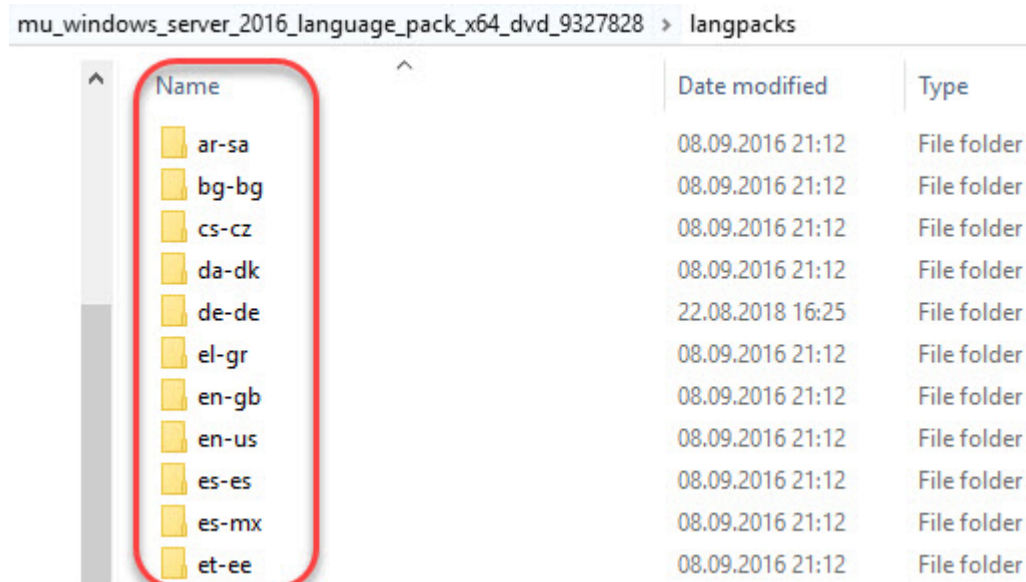


To extract the ISO file, use your favorite file archiving tool (e.g. 7-ZIP).

The ISO file for Windows Server 2016 contains two root folders:

- langpacks
- Windows Preinstallation Environment

The folder *langpacks* contains the language packs that need to be installed on Windows. The language files in the folder *Windows Preinstallation Environment* are used to localize your WinPE image. Within the folder *langpacks*, each language has its own subfolder.



These subfolders contain the source files we need for the unattended installation (deployment) of the language packs. The file name for each language pack is:

x64fre_Server_%language%_lp.cab

For example: **x64fre_Server_de-de_lp.cab**

Note: for past operating systems the name of a language pack file was *lp.cab*. This name is no longer used for Windows 10 and Windows Server 2016, except for the WinPE language packs.

Windows 10 version 1803 Language Pack

The language pack for Windows 10 is also a single ISO file, but much larger (more than 7 GB) compared to the one for **Windows Server 2016**. The reason for this is that the download for Windows 10 includes more source files. Besides the 38 language packs (LPs) also included in the ISO file for Windows Server 2016, the Windows 10 ISO file also includes 72 language interface packs (LIPs) and includes both 32-bit and 64-bit source files.

On the [Microsoft Volume License Site](#) or on the [Microsoft Visual Studio Subscription](#) website, search for *language pack* and you will find all language packs for all supported operating systems.

The screenshot shows the 'My Benefits' page on the Visual Studio website. The search bar contains 'language pack'. The results are filtered by 'All' and show 38 downloads. The results list several language packs, with the following details:

Product Name	Release Date	Architecture	Language	Media	Action
Windows Server, version 1709 Language Pack	10/Jan/2018	x64	English	DVD	Download
Windows Server, version 1803 Language Pack	07/May/2018	x64	English	DVD	Download
Windows Server 2016 Language Pack	12/Oct/2016	x64	English	DVD	Download
Windows 10 Language Pack, version 1803 (Updated March 2018)	30/Apr/2018	x64	English	DVD	Download
Windows 10 Language Pack, Version 1709 (Updated Sept 2017)	17/Oct/2017	x64	English	DVD	Download

The 'Windows Server 2016 Language Pack' and 'Windows 10 Language Pack, version 1803 (Updated March 2018)' are highlighted with red boxes.

To extract the ISO file, use your favorite file archiving tool (e.g. 7-ZIP).



Great deals are waitin

Ad At our store or at your
get the latest tech quickly

Best Buy

Shop Now.

The folder structure for the Windows 10 language packs is slightly different compared to the one for [Windows Server 2016](#):

- arm64
- Windows Preinstallation Environment
- x64
- x86

Each architecture has its own *langpacks* folder containing both the language packs and language interface packs. The file name for each language pack is:

Microsoft-Windows-Client-Language-Pack_%architecture%_%language%.cab

For example: **Microsoft-Windows-Client-Language-Pack_x64_de-de.cab**

The file name for each language interface pack is:

Microsoft-Windows-Client-Language-Interface-Pack_%architecture%_%language%.cab

For example: **Microsoft-Windows-Client-Language-Interface-Pack_x64_ca-es.cab**

Note: for past operating systems the name of a language pack file was *lp.cab*. This name is no longer used for Windows 10 and Windows Server 2016, except for the WinPE language packs.

Windows 10 version 1803 Features on Demand

The *Features on Demand* consists of two ISO files (between 3,5 and 4,5 GB in size). You can download either the 32-bit or 64-bit source files. For each version of Windows 10, the download consists of two parts (one ISO file per part). The languages for handwriting, speech, etc. are included in Part 1.

On the [Microsoft Volume License Site](#) or on the [Microsoft Visual Studio Subscription](#) website, search for *features on demand* and you will find the available downloads for all Windows 10 versions.



Visual Studio | My Benefits Dennis Span Sign out

Benefits Downloads Product Keys Subscriptions Support Marketplace

All features on demand

FILTER BY PRODUCT FAMILY

- ☐ Windows 10
- ☐ Windows 10 Version 1607
- ☐ Windows 10 Version 1703 (Updated...
- ☐ Windows 10, Version 1511
- ☐ Windows 10, Version 1709 (Updated...
- ☐ Windows 10, version 1803 (Updated...

DOWNLOADS (6) Sort by: Release date

Windows 10 Features on Demand, version 1803 (Updated March 2018)

No key required Info Release date: 30/Apr/2018 x64 English DVD Download

Windows 10 Features on Demand, Version 1709 (Updated Sept 2017)

No key required Info Release date: 17/Oct/2017 x64 English DVD Download

Windows 10 Features on Demand, Version 1703 (Updated Mar 2017)

No key required Info Release date: 05/Apr/2017 x64 English DVD Download

Windows 10 Features on Demand, Version 1607 (Updated Jul 2016)

No key required Info Release date: 02/Aug/2016 x64 English DVD Download

Windows 10 Features on Demand, Version 1511

No key required Info Release date: 12/Nov/2015 x64 English DVD Download

Windows 10 Features on Demand

No key required Info Release date: 28/Jul/2015 x64 English DVD Download

To obtain the language files you only need to download Part 1.

Windows 10 Features on Demand, version 1803 (Updated March 2018)

No key required Info Release date: 30/Apr/2018 x64 English DVD Download

Windows 10 Features on Demand, Version 1709 (Updated Sept 2017)

Windows 10 Features on Demand Part 1, version 1803 (Updated March 2018) (x64) - DVD (English)

Windows 10 Features on Demand Part 2, version 1803 (Updated March 2018) (x64) - DVD (English)

To extract the ISO file, use your favorite file archiving tool (e.g. 7-ZIP).

Within the folder structure you find multiple CAB files that represent a *Feature on Demand*. The name of the CAB files that contain language specific features start with *Microsoft-Windows-LanguageFeatures*. The following *categories* exist:

- Microsoft-Windows-LanguageFeatures-**Basic**.....cab
- Microsoft-Windows-LanguageFeatures-**Fonts**.....cab
- Microsoft-Windows-LanguageFeatures-**Handwriting**.....cab
- Microsoft-Windows-LanguageFeatures-**OCR**.....cab
- Microsoft-Windows-LanguageFeatures-**Speech**.....cab
- Microsoft-Windows-LanguageFeatures-**TextToSpeech**.....cab

Select the feature(s) in the language(s) that you require. The next section explains how to install (add) a language pack to the operating system.

Adding a language pack to Windows



There are a couple of different ways how to install a language (interface) pack:

- You can add a language pack to Windows by using the **DISM / Add-WindowsPackage** tool. *Add-WindowsPackage* is the PowerShell equivalent of the DISM executable. How to use **DISM / Add-WindowsPackage** is described in detail in this article. Just keep on reading. I also recommend you to check out the following Microsoft articles on this matter:
 - [Add and Remove Language Packs on a Running Windows Installation](#)
 - [Add and Remove Language Packs Offline Using DISM](#)
- To deploy a multilingual version of Windows by using Windows Setup (for example, a corporate image Windows DVD or a set of images available on a corporate network), you can add language resources to the installation program. See [Add Multilingual Support to a Windows Distribution](#). For corporate or network-based deployments, you may also need to update the Windows Preinstallation Environment (Windows PE) that users see when they choose how and where to install Windows to their PC. For more information, see [WinPE: Mount and Customize](#).
- After Windows is installed, end users can download and install additional language packs and LIPs from the Language Control Panel. For more information, see one of the following sections in this article:
 - [Change the display language manually \(Windows Server 2016\)](#)
 - [Change the display language manually \(Windows 10\)](#)

Reference: Section *Add languages to Windows* in the article [Language Packs](#) (microsoft.com)

My preferred silent installation method is to use **DISM** or the PowerShell equivalent **Add-WindowsPackage** (used in the [installation script](#) below).

The basic DISM command to install a language pack is:

```
dism.exe /online /add-package /packagepath:"C:\LP\de-de\lx64fre_Server_de-de_lp.cab" /quiet /norestart /logpath:"C:\Logs\MUI-de-de.log"
```

Also *Features on Demand* can be installed like that. Take for example the German handwriting package:

```
dism.exe /online /add-package /packagepath:"C:\LP\de-de\Microsoft-Windows-LanguageFeatures-Handwriting-de-de-Package~31bf3856ad364e35~amd64~~.cab" /quiet /norestart /logpath:"C:\Logs\MUI-de-de_Handwriting.log"
```

Note: the parameter */online* refers to the currently active Windows machine, as opposed to "slip streaming" source files in an offline Windows image file (WIM). The term "online" has nothing to do with being online on the Internet.

On operating systems from Windows 8.1/Windows Server 2012 R2 and newer with PowerShell 5.0 and higher installed, the cmdlet *Add-WindowsPackage* can be used instead of using *DISM.exe*:

```
Add-WindowsPackage -Online -PackagePath "C:\LP\de-de\lx64fre_Server_de-de_lp.cab" -LogPath "C:\Logs\MUI-de-de.log" -NoRestart
```



Please be aware that both **DISM** and the PowerShell equivalent **Add-WindowsPackage** always generate a log file. If you do not specify a custom log file and log file location, the default log path `C:\Windows\Logs\DISM` is used.

In the following section, [Complete Installation Script](#), you find a custom PowerShell script to install language packs including logging and error handling.

Complete installation script

The script below installs the Windows language pack using my [installation template](#), which includes detailed logging and error handling. Also, the functions used in the scripts require my [PowerShell Functions Library](#) to be present on the local system. This means that before you can use the script, you need to copy the PowerShell module file [DS_PowerShell_Function_Library.psm1](#) to the local system first.

In order to use this script, please follow these steps:

- Create an installation directory on the local computer or on a file share (UNC path). For example: `C:\Temp\Microsoft\Language Pack Spanish`.
- Create a subdirectory called *Files*.
- [Download](#) the language pack to the folder *Files* in the installation directory. Include the subfolder for the language pack. The structure should be `Files\%LanguagePackDir%`, for example `Files\es-es`. The CAB file, in this case for the Spanish language, is located in the directory `es-es`.
- Download and copy the PowerShell module file [DS_PowerShell_Function_Library.psm1](#) to a directory on your local system.
- Copy the complete PowerShell script below to a new PS1 file (e.g. `Install_Microsoft_Windows_MUI_Spanish.ps1`) and add this file to the root of the installation directory (not in the subdirectory *Files*).
- Optional: modify the location of the log directory and log file name in lines 43 and 44.
- Enter the correct path to the PowerShell module file in line 60.
- Modify the file name in line 70. In case you want to use a different directory name instead of *Files*, change it in line 71.
- Execute the PowerShell script as follows:
`powershell.exe -executionpolicy bypass -file "C:\Temp\Microsoft\Language Pack Spanish\Install_Microsoft_Windows_MUI_Spanish.ps1"`

Install Microsoft Windows MUI Spanish	PowerShell
1	<code>#=====</code>
2	<code>#</code>
3	<code># Install a Microsoft Language Pack</code>
4	<code>#</code>
5	<code># AUTHOR: Dennis Span</code>
6	<code># DATE : 27.08.2018</code>
7	<code>#</code>
8	<code># COMMENT: This script installs a Microsoft language pack (MUI) on the local machine</code>
9	<code>#</code>
10	<code># Note: see the article 'https://dennispan.com/powershell-scripting-template-fc</code>
11	<code>#</code>
12	<code># Note: for an overview of all functions in the PowerShell function library 'DS_</code>
13	<code># -Windows functions: https://dennispan.com/powershell-function-library/power</code>
14	<code># -Citrix functions: https://dennispan.com/powershell-function-library/power</code>

```

15 #
16 # Change log:
17 # -----
18 # <DATE> <NAME>: <CHANGE DESCRIPTION>
19 #=====
20
21 # Get the script parameters if there are any
22 param
23 (
24     # The only parameter which is really required is 'Uninstall'
25     # If no parameters are present or if the parameter is not
26     # 'uninstall', an installation process is triggered
27     [string]$Installationtype
28 )
29
30 # define Error handling
31 # note: do not change these values
32 $Global:ErrorActionPreference = "Stop"
33 if($verbose){ $Global:VerbosePreference = "Continue" }
34
35 #####
36 # Preparation #
37 #####
38
39 # Disable File Security
40 $env:SEE_MASK_NOZONECHECKS = 1
41
42 # Custom variables [edit]
43 $BaseLogDir = "C:\Logs" # [edit] add the log
44 $PackageName = "Microsoft Language Pack Spanish" # [edit] enter the d
45
46 # Global variables
47 $StartDir = $PSScriptRoot # the directory path of the script currently being execut
48 if (!(($Installationtype -eq "Uninstall")) { $Installationtype = "Install" }
49 $LogDir = (Join-Path $BaseLogDir $PackageName).Replace(" ", "_")
50 $LogFileName = "($($Installationtype))_($($PackageName).log"
51 $LogFile = Join-path $LogDir $LogFileName
52
53 # Create the log directory if it does not exist
54 if (!(Test-Path $LogDir)) { New-Item -Path $LogDir -ItemType directory | Out-Null }
55
56 # Create new log file (overwrite existing one)
57 New-Item $LogFile -ItemType "file" -force | Out-Null
58
59 # Import the Dennis Span PowerShell Function Library
60 Import-Module "C:\Scripts\DS_PowerShell_Function_Library.psm1"
61
62 DS_WriteLog "I" "START SCRIPT - $Installationtype $PackageName" $LogFile
63 DS_WriteLog "-" "" $LogFile
64
65 #####
66 # Installation #
67 #####
68
69 # Set variables
70 $FileName = "x64fre_Server-es-es_lp.cab" # [edit] en
71 $FileSubfolder = "Files" # [edit] en
72
73 # Install the language pack (using the log file defined in the variable $LogFile)
74 $Language = $FileName.Substring(14,5) # Retrieve
75 $FileSubfolder = "Files\$($Language)" # Add th
76 $FullPath = Join-Path $StartDir $FileSubfolder # Concat
77 $FileFullPath = Join-Path $FullPath $FileName # Concat
78 $LogFileMUI = Join-Path $LogDir "DISM_Log_LanguagePack_($($Language.ToUpper()).log"
79 DS_WriteLog "I" "Install the Windows Language Pack $($Language.ToUpper())" $LogFile

```

```

80 DS_WriteLog "I" "-File full path: $($FileFullPath)" $LogFile
81 DS_WriteLog "I" "-DISM log file: $($LogFileMUI)" $LogFile
82 Try {
83     DS_WriteLog "I" "-Command: Add-WindowsPackage -Online -PackagePath ""$FileFullP
84     Add-WindowsPackage -Online -PackagePath $FileFullPath -LogPath $LogFileMUI -NoR
85     DS_WriteLog "S" "The language pack $($Language.ToUpper()) was installed success
86 } Catch {
87     DS_WriteLog "E" "Failed to install the language pack $($Language.ToUpper())" $L
88     Exit 1
89 }
90
91 #####
92 # Finalize #
93 #####
94
95 # Enable File Security
96 Remove-Item env:\SEE_MASK_NOZONECHECKS
97
98 DS_WriteLog "-" "" $LogFile
99 DS_WriteLog "I" "End of script" $LogFile

```

If you want to go one step further and create an SCCM package as well, please follow the step-by-step explanation in the article [Deep dive creating SCCM packages for Citrix](#).

How to change the Windows display language

The Windows display language is configured per user, not per machine. There are a couple of methods how to change the display language. You can choose one of the following options:

- [Change the display language manually \(Windows Server 2016\)](#)
- [Change the display language manually \(Windows 10\)](#)
- [Change the display language using Microsoft Group Policy](#)
- [Change the display language using a Group Policy Preference item](#)

Most methods apply to both Windows 10 and Windows Server 2016 unless stated otherwise.

Change the display language manually (Windows Server 2016)

Note: in an enterprise environment, I strongly recommend installing the required language packs [unattended](#) during the initial installation. This goes for both client and server operating systems.

To manually change the Windows display language on a machine with Windows Server 2016 version 1607, go to *Control Panel \ Language*. In the menu the entry *Add a language* adds the keyboard layout for a particular language. Under *Options* you can enable the Windows display language if it has been installed.



Change your language preferences

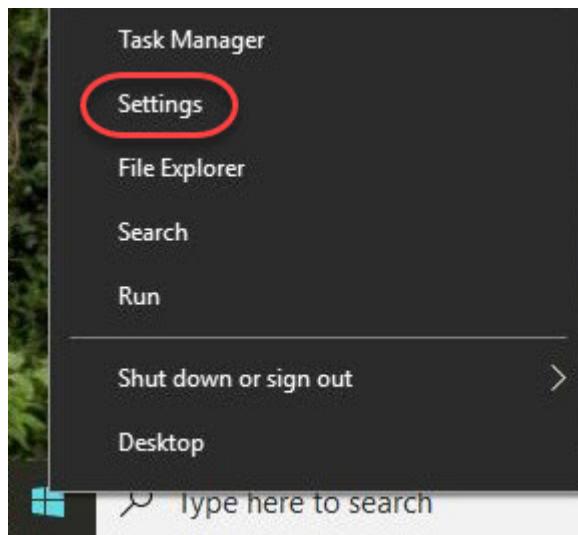
You can type in any language you add to the list. Windows, apps and websites will appear in the first language in the list that they support.

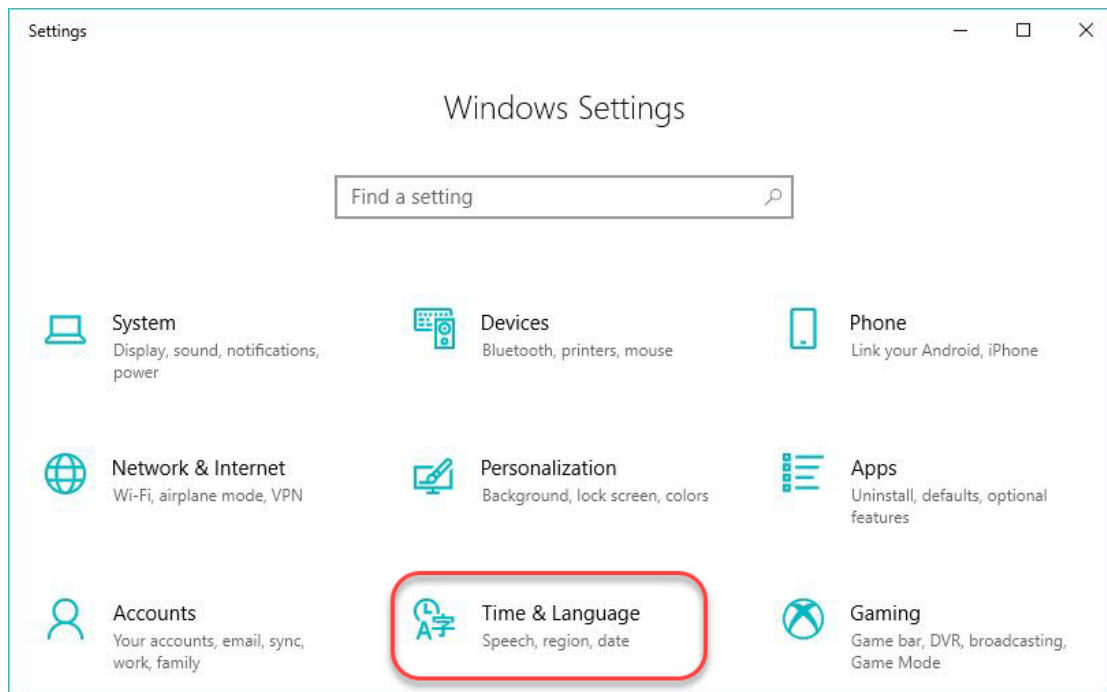
[Add a language](#) [Remove](#) [Move up](#) [Move down](#)

English (United States)	Windows display language: Enabled Keyboard layout: US	Options
Deutsch (Österreich)	Windows display language: Available Keyboard layout: German Date, time, and number formatting	Options
français (France)	Keyboard layout: French	Options

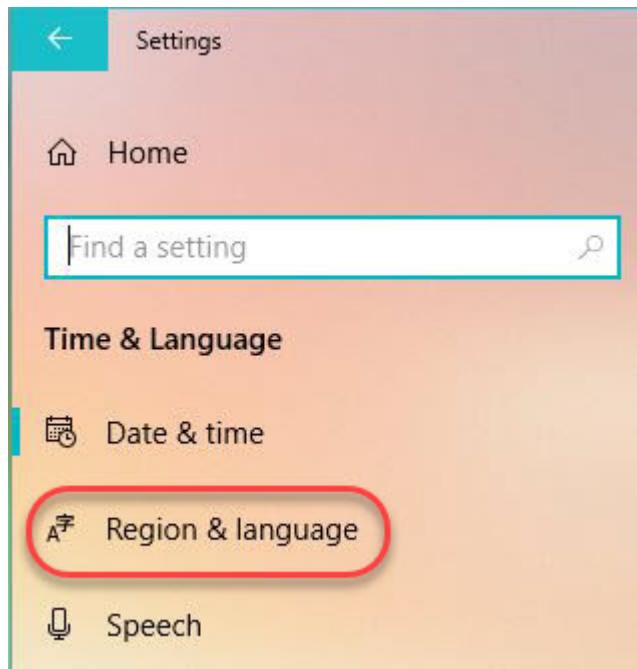
Change the display language manually (Windows 10)

In Windows 10 version 1803, changing languages is handled differently. Go to *Settings \ Time & Language*.





On the left side select *Region & Language*.



In this window you can add a language. After adding a language, Windows checks if there is a language pack available for it. In case a language pack is found, you have the option to download and install it. Click the *Options* button.



Languages


Windows display language


Windows features like Settings and File Explorer will appear in this language.



English (United States) 

Preferred languages

Apps and websites will appear in the first language in the list that they support.


 Add a language





 Deutsch (Österreich)
Language pack available

Options

Remove

 English (United States)
Windows display language

Select the *Download* button beneath the language options to start the download.



Deutsch (Österreich)

Language options

Downloading



Use post-reform rules



On

Handwriting (8 MB)

Download

Speech (5 MB)

Download

After installing the language pack, Windows will automatically install it.

Deutsch (Österreich)

Language options

Installing



Use post-reform rules



On

Handwriting (8 MB)

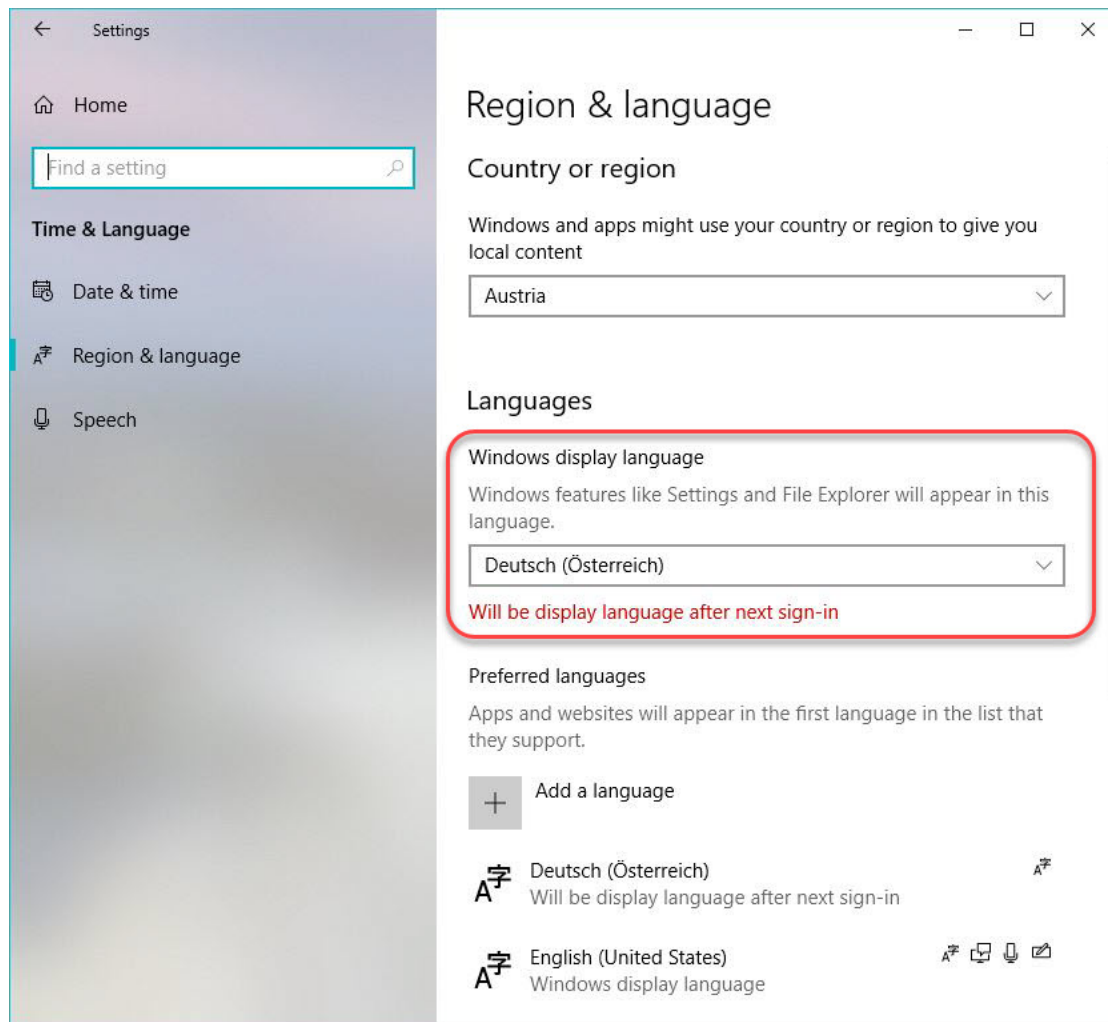
Download

Speech (5 MB)

Download

After installing the language pack, a new popup box is available where you can select the preferred Windows display language. As mentioned before, this is for the current user only.



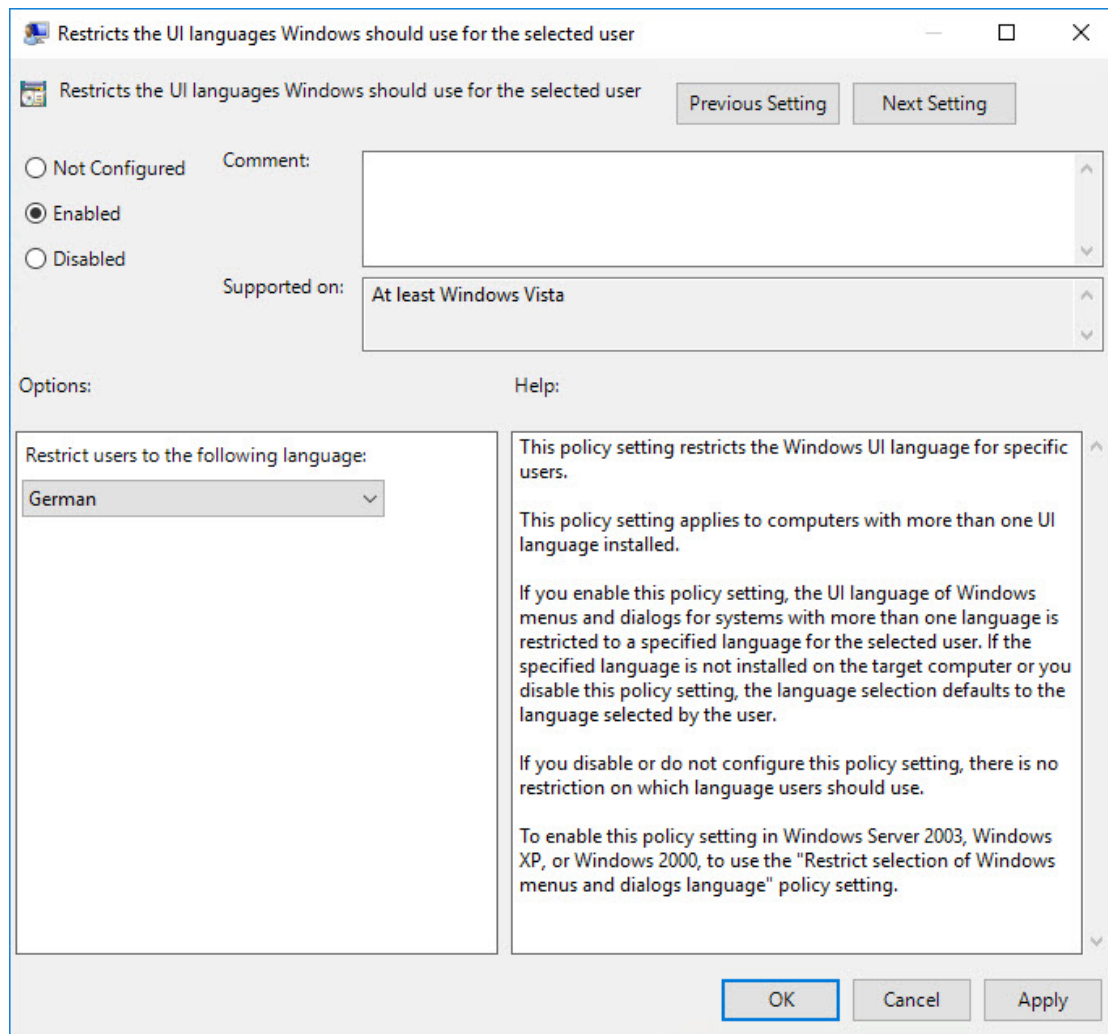


After changing the Windows display language you have to sign-out (and sign-in again) for the changes to apply.

Change the display language using Microsoft Group Policy

The Windows display language for a specific user or group can be forced by enabling the Group Policy setting *Restricts the UI language Windows should use for the selected user* in the Group Policy *User Configuration \ Policies \ Administrative Templates \ Control Panel \ Regional and Language Options*.





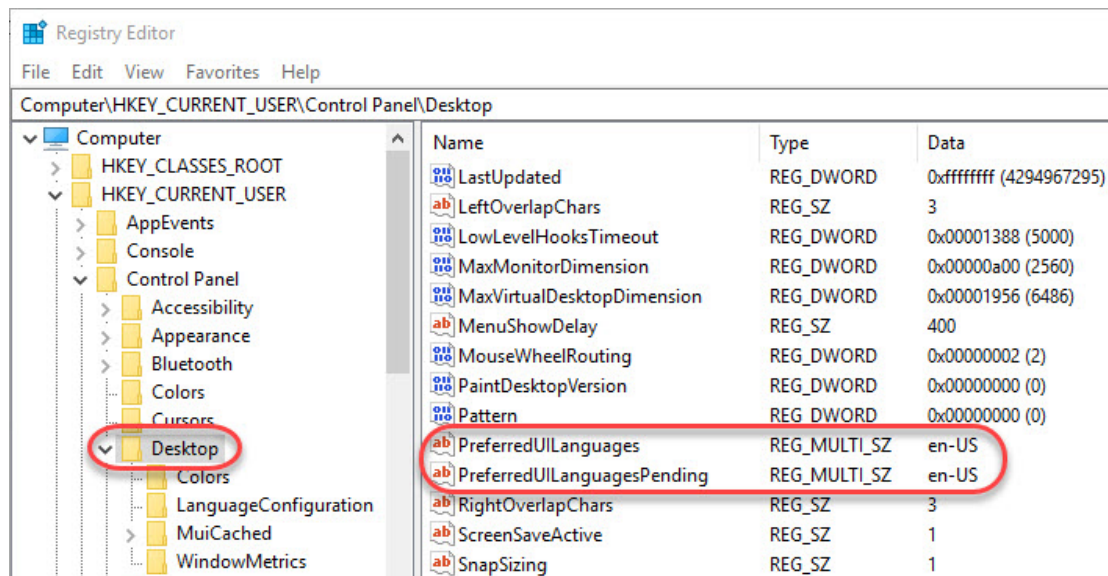
It is important to be aware that this setting prevents a user from selecting a different display language, even if more languages are installed on the local system. In case you want to be less restrictive towards your users, use a Group Policy Preference item instead of a Group Policy item. How to use a Group Policy Preference item is explained in the [section below](#).

Change the display language using a Group Policy Preference item

In case you want users to be able to change their Windows display language, you cannot use a Group Policy setting. As mentioned in the previous paragraph, configuring a Group Policy setting to define the user's display language results in the user not being able to change the Windows language. The solution is to use a Group Policy Preference item instead.

The preferred Windows display language is stored in the registry value *PreferredUILanguages* in the registry key *HKEY_CURRENT_USER\Control Panel\Desktop*. The value *PreferredUILanguagesPending* is only used to store the language that will be active at next login in case the user changed it manually and did not sign-out yet.

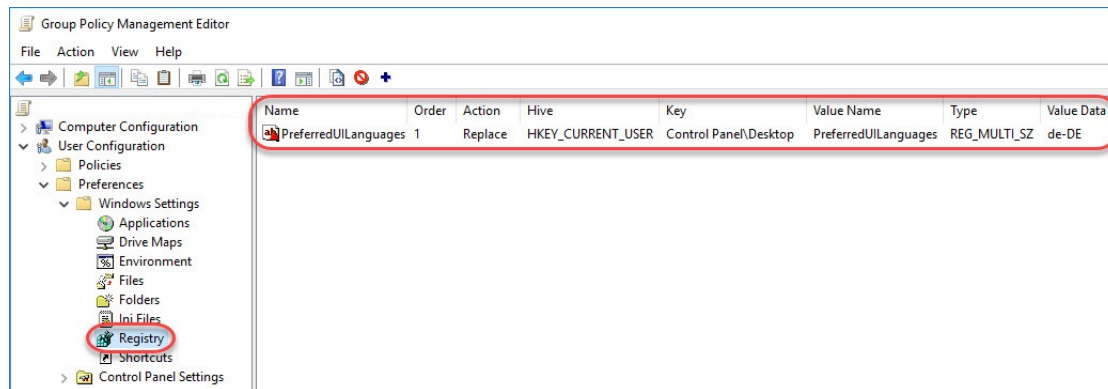




Changing the language in this multi-string registry value accomplishes the same result as when changing the language manually using the GUI. Of course, for everything to work the language pack has to **already be installed** on the local system.

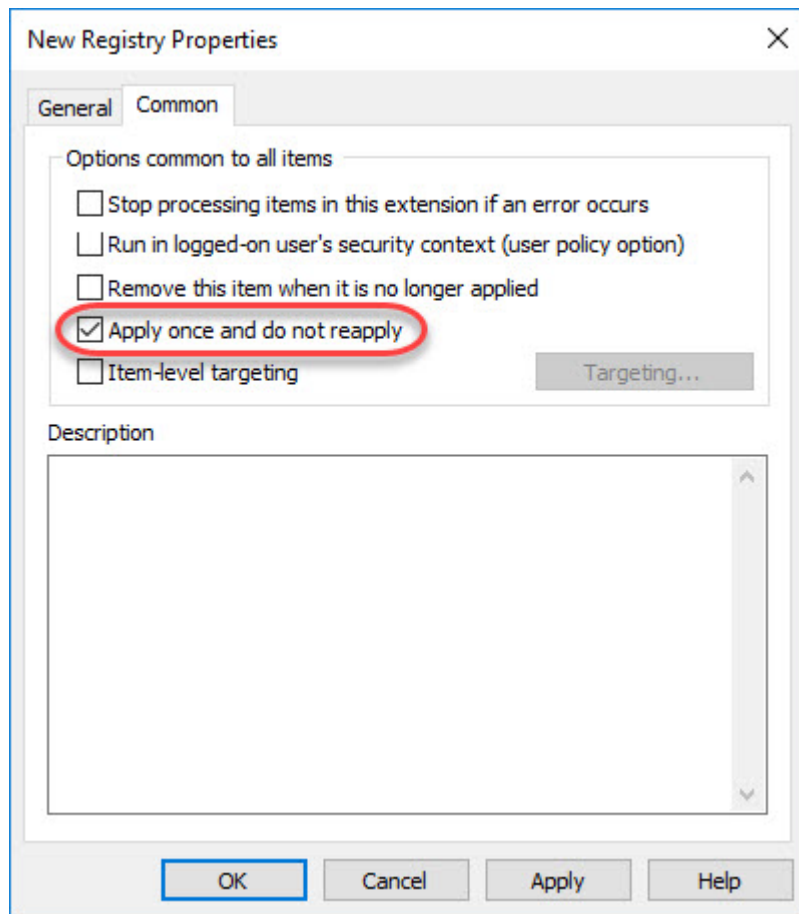
This solution applies to both Windows 10 and Windows Server 2016 (and higher).

Using a Group Policy Preference registry item allows an administrator to configure the preferred Windows display language while still allowing a user to change the language manually.



Configuring the option "run one time" ensures that the preference item is not enforced at next login.





Applying the preference item only once ensures that the display language is never overwritten should the user decide to change it.

Using [Item-level Targeting](#) (ILT) you can assign preferred Windows languages to different users based on a number of criteria (see the section *Configuring regional settings using Group Policy* in my article [Configure Regional Settings and Windows locales with Group Policy](#) for an example how to use ILT).

Using a logon script

As explained in the [previous paragraph](#), the setting to configure the Windows display language is a registry value. Therefore, it is possible to use a logon script to configure this setting.

In case you use a PowerShell script I invite you to use my [PowerShell Function Library](#). You can configure the registry value using the function [DS_SetRegistryValue](#).

To prevent a user from having to sign-out in order to apply the language settings, make sure to run your logon script *synchronously*. This means that your logon script needs to run before the user's desktop is started (so before the initial *explorer.exe* process is started). Enable the Group Policy item *Run logon scripts synchronously* in the Group Policy *Computer or User Configuration \ Policies \ Administrative Templates \ System \ Scripts*.



How to change the display language for Windows Apps and Store, Edge and Internet Explorer

As stated in the [introduction](#) of this article, the Windows display language is applied to all Windows components including Windows Apps and Store, Edge and Internet Explorer. However, it is possible to use a different language for the Windows Apps and Store, Edge and Internet Explorer.

Note: the language settings I am referring to here is the language used *inside* a Windows App, Edge or Internet Explorer, not the menu language of the actual application! For example, when the language of Internet Explorer is set to German and you visit Microsoft.com, the content is displayed in German.

This is possible by changing the order of the languages. You can choose one of the following options:

- [Manually \(Windows Server 2016\)](#)
- [Manually \(Windows 10\)](#)
- [Group Policy Preference](#)

Most methods apply to both Windows 10 and Windows Server 2016 unless stated otherwise.

Great deals are waiting

Ad At our store or at your
get the latest tech quickly

Best Buy

Shop Now.

Manually (Windows Server 2016)

To manually change the Windows display language for Windows Apps and Store, Edge and Internet Explorer, on a machine with Windows Server 2016 version 1607 installed, go to *Control Panel \ Language*. In the menu, move the preferred language to the top of the list.



Change your language preferences

You can type in any language you add to the list. Windows, apps and websites will appear in the first language in the list that they support.

[Add a language](#) [Remove](#) [Move up](#) [Move down](#)

Deutsch (Deutschland)	Keyboard layout: German	Options
English (United States)	Windows display language: Enabled Keyboard layout: US	Options

Manually (Windows 10)

In Windows 10 version 1803, changing languages is handled differently. Go to *Settings \ Time & Language \ Region and Language*. Use the arrows to move the preferred language for Windows Apps and Store, Edge and Internet Explorer to the top of the list.



Languages

Windows display language

Windows features like Settings and File Explorer will appear in this language.

English (United States) ▼

Preferred languages

Apps and websites will appear in the first language in the list that they support.

+ Add a language

Deutsch (Österreich)
Language pack installed

English (United States)
Windows display language (override)

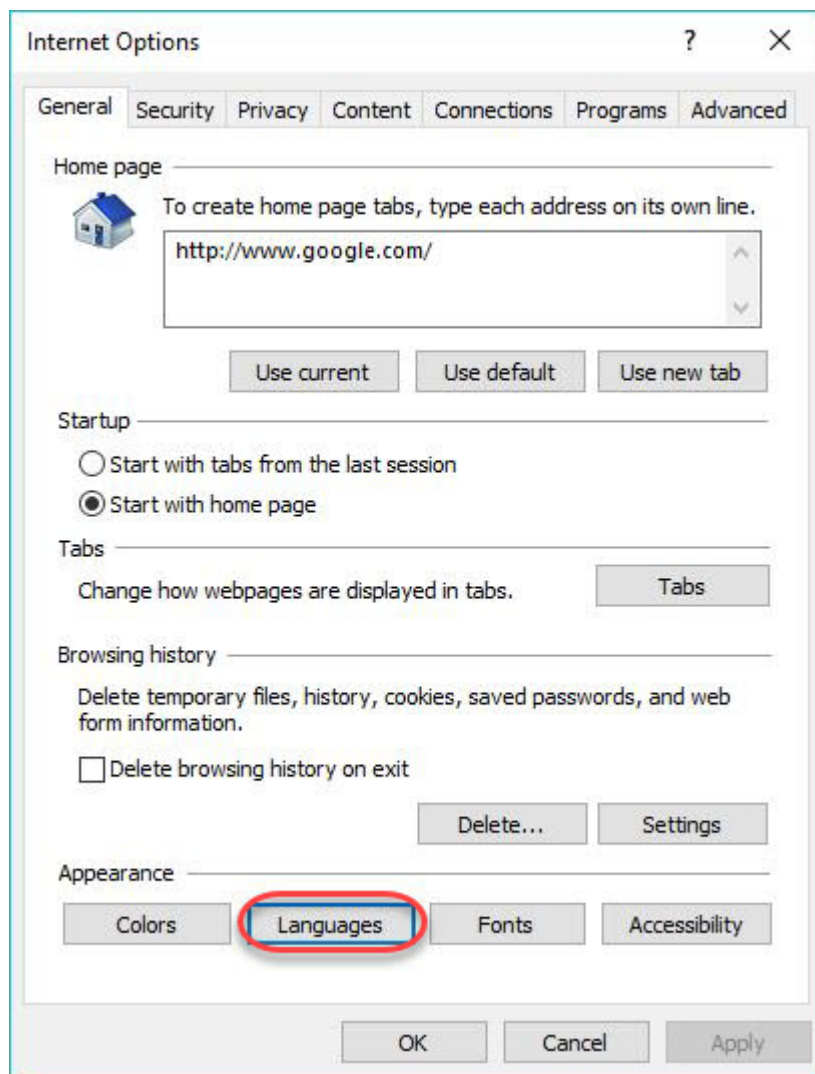


Options

Remove

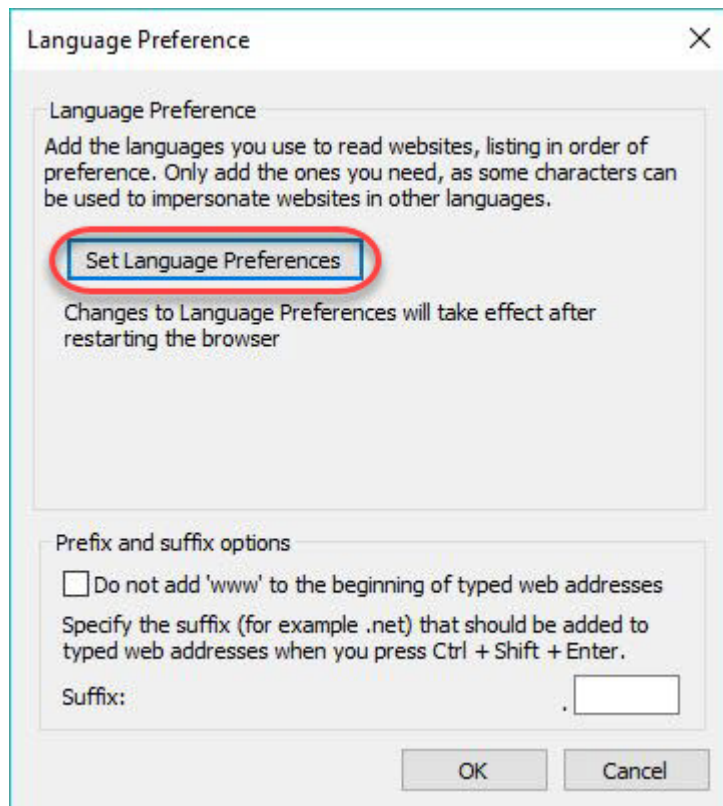
It is also possible to change the display language for Windows Apps and Store, Edge and Internet Explorer from within Internet Explorer. Go to *Internet Options* and on the tab *General* go to *Languages*.





In the window *Language Preference*, click on the button *Set Language Preferences*.





This will lead you to the [Region and Language](#) settings page.

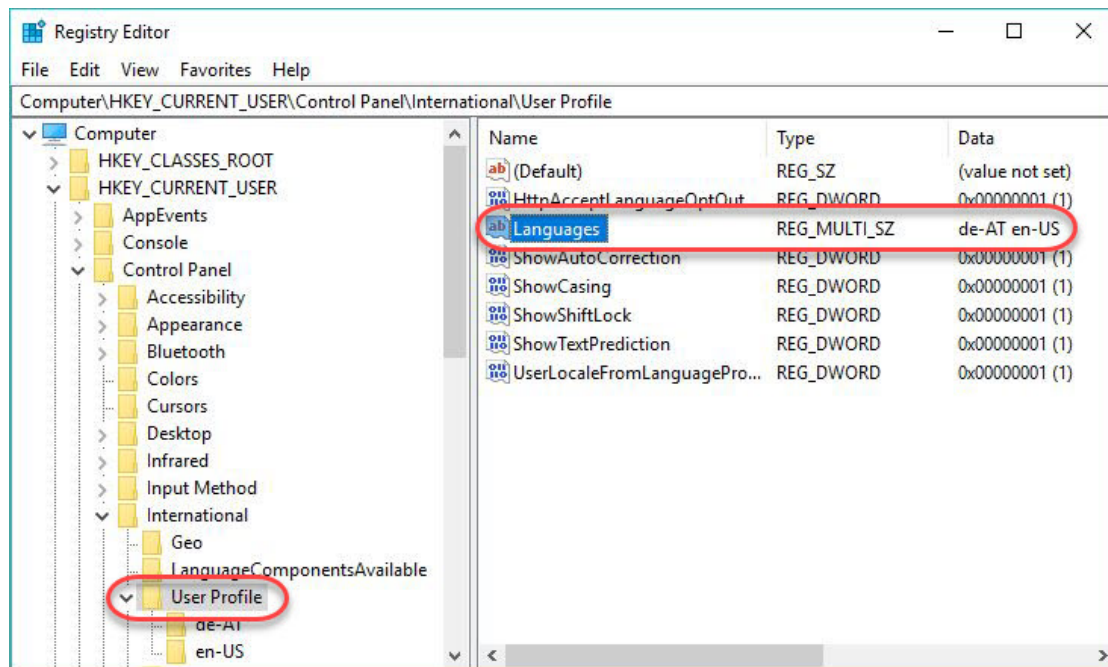
Note: those of you who have experience with older operating systems will quickly realize that Microsoft made some significant changes concerning the language configuration in Internet Explorer.

Group Policy Preference

As far as I am aware, there is no Group Policy setting to configure the display language for Windows Apps and Store, Edge and Internet Explorer. But don't worry; you can use a Group Policy Preference instead.

The preferred Windows display language for Windows Apps and Store, Edge and Internet Explorer is stored in the registry value *Languages* in the registry key *HKEY_CURRENT_USER\Control Panel\International\User Profile*.





Changing the order of languages in this multi-string registry value accomplishes the same result as when changing the order using the GUI.

This solution applies to both Windows 10 and Windows Server 2016 (and higher).

See the section [Change the display language using a Group Policy Preference item](#) in this article for more information how to configure a preference item.

Using a logon script

As explained in the [previous paragraph](#), the setting to configure the display language for Windows Apps and Store, Edge and Internet Explorer is a registry value. Therefore, it is possible to use a logon script to configure this setting.

In case you use a PowerShell script I invite you to use my [PowerShell Function Library](#). You can configure the registry value using the function [DS_SetRegistryValue](#).

To prevent a user from having to sign-out in order to apply the language settings, make sure to run your logon script *synchronously*. This means that your logon script needs to run before the user's desktop is started (so before the initial *explorer.exe* process is started). To accomplish this, enable the Group Policy item *Run logon scripts synchronously* in the Group Policy *Computer or User Configuration \ Policies \ Administrative Templates \ System \ Scripts*.

How to prevent users from changing language settings

By configuring the Windows display language [using a Group Policy](#), you basically prevent a user from changing the language, at least until the next Group Policy refresh cycle (this will likely be every 30 to 60 minutes in most organizations).



In case this is not enough and you want to prevent users from changing language settings in any of the menus, the following policies are available:

- **User Configuration \ Policies \ Administrative Templates \ Control Panel \ Regional and Language Options**
Setting: Hide the select language group options
- **User Configuration \ Policies \ Administrative Templates \ Windows Components \ Internet Explorer \ Internet Control Panel**
Setting: Disable the general page
- **User Configuration \ Policies \ Administrative Templates \ Windows Components \ Internet Explorer**
Setting: Disable changing language setting

Please be aware that configuring the above settings does not prevent the user from changing these settings programmatically or by using tools such as *regedit.exe*. After all, language settings are user specific settings and by default, users have permission to change their own user-specific settings.

How to change the system code page for non-unicode programs

A Windows language (interface) pack does not modify the system specific code page for non-Unicode programs. Please see the article [Configuring the time zone and code page with Group Policy](#) for detailed information on how to change the system's code page.

Conclusion

I honestly thought that writing this article would be simpler and less time consuming. There are still some smaller open issues I am working on and I will update this article when I am done. In the mean time, I hope the information in this article was of some help to you.

Related

[SOLVED - Slow installation of MSU files using the WUSA.exe](#)

June 2, 2016

In "Other"

[Deep dive creating SCCM packages for Citrix](#)

February 9, 2017

In "Citrix"

[Configuring the time zone and code page with Group Policy](#)

October 7, 2017

In "Other"

[Dennis Span](#)



Dennis Span works as a Lead Sales Engineer at Citrix in Vienna, Austria. He holds multiple certifications such as CCE-V, CCIA and CCEA. In 2017, Dennis became a Citrix Technology Advocate (CTA). In 2019, he became a Citrix Technology Professional (CTP). Besides his interest in virtualization technologies and blogging, he loves



spending time with his family as well as snowboarding, playing basketball and rowing. He is fluent in Dutch, English, German and Slovak and speaks some Spanish.

This entry was posted in **Other** and tagged **Group Policy**, **PowerShell**, **Regional Settings**, **Scripting**, **Windows locale** by **Dennis Span**. Bookmark the **permalink** [<https://dennisspan.com/managing-windows-languages-and-language-packs/>].

35 THOUGHTS ON "MANAGING WINDOWS LANGUAGES AND LANGUAGE PACKS"



RVJ Callanan

on **October 22, 2018 at 14:46** said:

Super-informative article cutting through all the noise on this thorny subject.

In view of your extensive experience on this subject, I was hoping you might have come up against the following problem which is widely reported but doesn't affect users in the United States (which may explain why it is not getting due attention from Microsoft).

Even though I have English (GB) installed and operational on my Google Cloud-based Windows Server 2016, the user's Display Language reverts to the originally installed Language, i.e. English (United States), after every user sign off/sign on (via RDP). The Keyboard Language is not affected by this problem and I can change the Display Language to English (United Kingdom) via the task bar during a logon session. However it reverts back to English (United States) on next sign-on.

I have tried all of the stated approached in your article but to no avail.



Dennis Span

on **October 22, 2018 at 21:20** said:

Hi Roger,

I am happy you enjoyed the article. You say that this problem is widely reported. Can you send me a couple of links perhaps? The user display language is a current user setting that can only be overwritten automatically in two situations:

1) The registry value

HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\MUI\Settings ->

'PreferredUILanguages' with the value 'en-US' is configured (as I state in my article this can be set via Group Policy as well).

2) The user profile is not persistent: the user gets a new, clean profile at each logon.

Please check if any of the above situations apply. If not, please get back to me.



In an RDP session, the local client keyboard is parsed to the RDP session (that is why the keyboard settings work).

Bye,

Dennis



RVJ Callanan

on **October 23, 2018 at 00:16** said:

Actually Dennis, I just found the solution not long after commenting on your article.

See:

https://answers.microsoft.com/en-us/windows/forum/windows_10-other_settings-winpc/windows-keeps-automatically-adding-en-us-keyboard/69aaea0d-f588-4d54-89fa-a632ae7bcd3

I am gobsmacked by Microsoft's continuing efforts to destabilise their operating system by introducing crazy off-the-cuff design decisions with every new release.

To quote the relevant section:

« For some reason, Windows seems to regularly check to see if I'm using the EN-US keyboard layout, and adds it if I'm not »

Actually Microsoft in 2008 added to Windows an "improvement" where a PC when booting, changes the Language and Keyboard Layout set on this PC, to the one of a nearby other PC on the network, which of course is often US, but also can be US-International, French, German, or any of the ~150 others. Of course this behavior, being full unexpected and full hidden, puts the user in a long period of doubt and fruitless search for solution.

Diego gave on 07 Oct 2012 a Total protection against these Keyboard Layout changes from nearby computers, which is immediately, totally, and reliably effective:

In RegEdit, navigate to

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Keyboard Layout, in right pane right-click and choose "New" then "DWORD Value", name it "IgnoreRemoteKeyboardLayout", then double-click it and under "Value data" type "1" (don't mind "Hexadecimal" or "Decimal", either will do).



Dennis Span

on **October 23, 2018 at 08:17** said:

Hi Roger,



I am slightly confused. I thought you had an issue with the user's Windows display language changing, not the keyboard layout. I guess I misunderstood. Sorry. Concerning the keyboard layout, it is correct that the registry value "IgnoreRemoteKeyboardLayout" prevents that the keyboard of the local client device is parsed in an RDP session. This is also officially documented by Microsoft (<https://support.microsoft.com/en-us/help/322042/input-language-of-terminal-server-client-does-not-match-that-of-termin>). This is also what I meant to say with my phrase "In an RDP session, the local client keyboard is parsed to the session" in my previous comment.

The fact that the keyboard reverts back to English (US) tells me that the keyboard on your local client is set to English (US) and not English (UK). It is actually best practice to NOT create and set the registry value "IgnoreRemoteKeyboardLayout" to "1". Under most circumstances, it is a good thing that each user logging on to the remote machine gets his or her local keyboard redirected.

But I am happy that your problem was solved!

Bye,

Dennis



RobotJones

on **October 25, 2018 at 15:27** said:

I've recently been tasked installing an additional language pack in our environment (multiple versions of windows 10) and quickly found things were not as simple as MS make them out to be. This is by far the clearest article I've found on the subject, exactly what I was looking for. Thank you for taking the time to write such an informative piece.



Dennis Span

on **October 30, 2018 at 12:12** said:

Thank you for your very kind feedback! I am happy that the article was of some help to you.





Keith Eves

on **December 24, 2018 at 19:11** said:

Is there a command to check which language pack(s) is/are installed?
I'm looking for something that can be used in a batch file preferably.

Thanks in advance

**Dennis Span**

on **December 27, 2018 at 15:26** said:

Hi Keith,

```
$OSInfo = Get-WmiObject -Class Win32_OperatingSystem  
$languagePacks = $OSInfo.MUILanguages  
$languagePacks
```

For a batch file you can use the command "dism.exe": **dism /online /Get-Intl**

Reference: <https://stackoverflow.com/questions/32561430/powershell-command-to-get-all-the-language-packs-installed-on-a-server-running-w>



NVR

on **March 21, 2019 at 21:59** said:

Hi there,

I want to add addition language on Windows 10 1809 domain computers using Group policy 2012 R2. I dont want to restrict any language on Windows 10 computers. Above script is not working fine.

Could you please guide me how to do that?



**Dennis Span**on **March 28, 2019 at 15:23** said:

Hi,

Could you please be more specific and tell me what exactly is not working for you? Are you referring the the script to install the language pack, or are you talking about the Group Policy (Preference) to configure the default language for the user?

**oger frederic**on **April 17, 2019 at 00:21** said:

During installation of your Windows 10 OS, You can change your OS language. Beside of If you've bought retail activation product license key, then you can enter it on Microsoft official site to download French language pack, For which I recommend you to buy cheap and legit license keys from ODosta Store at: <http://products.odosta.com> They're offering cheap license with lifetime warranty but before buying just search for their coupon codes too.

**Fintan**on **May 19, 2019 at 07:19** said:

Excellent article – thanks for posting (and for keeping it updated).

I notice that if Windows Display Language is set to “English (Australia)” then the registry key PreferredUILanguages is set to en-GB, so it seems that the Australia language pack is part of the GB/UK language pack.

If this is correct, and the registry key only sets the Language Pack, how do you distinguish which form of English (in this case UK or Australian) you want used for the Display Language?

**Dennis Span**on **June 13, 2019 at 21:30** said:

Hi Fintan. The question is whether or not you can actually set the Windows language to Australian or if it is 100% the same as setting the language to British. I can check if you want to.



shlomi

on **May 27, 2019 at 13:27** said:

thanks you!!

**Dennis Span**

on **June 4, 2019 at 20:21** said:

You're welcome!



Eduard Herdt

on **June 6, 2019 at 10:18** said:

I have Windows 10 1607 with several installed language packs if I start Windows update will be update installed, but if I switch to another language will install the same update again. Do you have any idea how I can update multiple languages image?

**Dennis Span**

on **June 13, 2019 at 20:48** said:

Hi Eduard. I will need a bit more information to fully understand your situation. So you are saying that after installing certain Windows updates and afterwards switching the Windows language this triggers the same update once again? Did I understand it correctly?





Jürgen

on **July 1, 2019 at 17:33** said:

Hello, we need to install the language packs offline – all downloads (bussinesscenter/MSDN) works only with new installed servers (means without any update just ISO). If the server get all microsoft-updates the language packs form the download don't work.

Where can i get the actual language pack?

Kind regards

**Dennis Span**

on **July 18, 2019 at 21:32** said:

Hi Jürgen,

My apologies for the late reply. I have been extremely busy lately. I am not sure I understand your problem. What do you mean that all downloads only work with newly installed servers? When you download a language pack as described in the section <https://dennisspan.com/managing-windows-languages-and-language-packs/#ObtainWindows10LP>, you are not able to install this on an already installed server or W10 machine? And in your second remark, if I understand it correctly, you say that after you install Windows updates the already installed language packs do not work? Could you please elaborate on these issues? Which Windows version are you using? The more details you can provide the better.



Raj

on **October 26, 2019 at 21:56** said:

Hi Dennis,

Thank you for the excellent article. I am facing an issue. I am preparing Citrix golden Image which requires me to install all 18 additional languages that supports windows server 2012 R2. I did install them successfully on the golden image. But after installation – the golden image is taking 12mins to power on/ restart takes. Anything abnormal with additional language packs installation ? Any recommendation from MS ?

**Dennis Span**



on **October 31, 2019 at 17:16** said:

Hi Raj,

I am happy you like the article. Hmm, I never came across the issue you are describing. Are you sure that the language packs are the cause? What does the event log say? You could use ProcMon to find the root cause, in particular by enabling the Boot Logging option (see this article by Mark Russinovich for more information:

<https://blogs.technet.microsoft.com/markrussinovich/2012/07/01/the-case-of-the-veeerrry-slow-logons/>).



Shahid Mustafa

on **November 14, 2019 at 18:02** said:

Very informative article, I must say Dennis! Good work!

Now what I am facing issue with is Microsoft Edge after installing English OS and then installing a certain Language Pack after it, via SCCM.

The issue I am facing is when I start Edge on that client with the language pack (Swedish) that I have installed, I get this weird kind of box rectangles on Edge where I can not see home, favorites and many other icons, they are all shown as a box. However when I point at them, I get the information in right language that what it is.

When I see the languages I can only see Swedish in "Windows Display Languages" I have to install English language all of sudden first and then I can see the icons of Edge.

It's Windows 10 I am working with SCCM Current Branch. I know it can be a bit out of topic but I wonder if you know what can be the root cause?

Regards

Shahid



Dennis Span

on **November 18, 2019 at 09:42** said:

Hi Shahid,

Sorry, but I have no suggestions for you at this moment. I have not seen this issue myself.



**Russell**on **December 2, 2019 at 14:24** said:

Thanks very much Dennis. This is the best write-up on the current state of Language Pack installs I have come across. After several hours of reading and research on the subject you have answered all of my remaining questions and filled in some gaps in my understanding of Powershell and Language Pack installation.

One update to your article – there was no new LP media released for Windows 1009 – it uses 1903 version LPs.

Many thanks for the great contribution to the community.

Russell

**Dennis Span**on **December 2, 2019 at 20:50** said:

Thanks for the kind feedback Russell. And thanks for the information concerning the Windows 1009 LPs.

**Rem**on **February 10, 2020 at 15:07** said:

Hi Dennis,

Great article but unfortunately it doesn't work for our situation or I'm doing something wrong. We have only 1 language pack installed, nl-NL (checked that in powershell). The thing is, preferred language is set to en-US for each new user which logs onto a machine.

More funny is that we have no en-US installed, only the keyboard layout is US English but all other regional settings are Dutch. I think the en-US comes from the keyboard layout, logically, since there's really no other LP installed besides nl-NL.

Now I'm trying to change the preferred language order through the registry (GPO simulation). I'm talking about changing Control Panel\International\User Profile "Languages". I can do this for the current user, but only when I add the Control Panel\Desktop "PreferredUILanguages". Otherwise it simply won't work.



Also, when I do this manually I can remove en-US from the list after changing the preferred language order. Through GPO/registry not a chance.

I'm going to try to go into audit mode and generalize the settings with correct information and see if after that it will do what we want. We simply do NOT want for the user to see Eng near the date/time or for them to have the possibility to change anything to English regional settings. Not sure where to go from here yet. I do remember that for Windows 7 it was a similar tricky situation to get things right with languages. I just don't remember how that was done and probably won't work the same way on Win 10 anyways.

**Dennis Span**

on **March 1, 2020 at 12:33** said:

Hi Rem,

My apologies for the late reply. I was on holiday. Just to be sure; what exactly is the issue? Is it the Windows language, the keyboard language or the date/time settings? The keyboard language has nothing to do with the language pack. You can use any keyboard layout you like independent of the Windows display language. The same goes for regional settings. I wrote an article about regional settings which may clarify things:

<https://dennisspan.com/configuring-regional-settings-and-windows-locales-with-group-policy/>.

**Bhargav**

on **May 16, 2020 at 02:54** said:

Hi Dennis,

Thank you for the excellent article.

Unfortunately I am not having professional licence of either MSDN nor Visual studio.

But I need the iso for Windows 10 LP. Can you share iso on like google drive or one drive or any other source.

I have tons of PC at my office those are not connected to the internet and are not in any kind of domain.

It would be great help if you can please provide me the iso file for Windows 10.

Many thanks for the great contribution to the community.

Bhargav



**Dennis Span**on **May 16, 2020 at 10:04** said:

Hi Bhargav. You can actually create your own ISO file using the [Microsoft Media Creation Tool](#).

**Kiran**on **June 7, 2020 at 08:33** said:

Hi Dennis,

Great Article, I have a question though.

I have language pack installed and I have created an Image of that.

I have deployed that image in another machine.

But when I run "Set-WinUserLanguageList en-US" in that machine in powershell, It is automatically deleting German language pack at times(not always).

Since there is no interent connetion in that machine.

The language pack gets delete permanantly.(Only way is to reimage it).

Is there any way to prevent the OS from deleting any language pack?

**Dennis Span**on **June 8, 2020 at 22:47** said:

Hi Kiran,

You are right, the command "Set-WinUserLanguageList en-US" is used to (re)set the language list. The command you execute basically states that the language list should only consist of "en-US". Thus any other language packs will be removed. What exactly are you trying to do? Why are you executing the command? Do you want to set a default language for the current user? If yes, I recommend using a Group Policy (Preference) for this.

**Annika**

on **June 15, 2020 at 11:21** said:

Hi Dennis,

Thank you for the excellent article.

I have one problem, when trying to create a gpo with Regional Options one country and language is missing from the list.

Do you know how I can add missing Regions?



Dennis Span

on **September 10, 2020 at 08:48** said:

Hi Annika. I do not think it is possible to add regions to the Group Policy Preference. These are pre-defined. In this particular situation you will have to configure the regional settings via registry.



Venkat

on **October 13, 2020 at 20:35** said:

Hi Dennis,

Excellent article.

I am searching since 2 days to know end to end about multi language. By God's grace today I found your article which saved customer escalations on delay.



Dennis Span

on **October 13, 2020 at 22:41** said:

I am glad the article helped you Venkat! Thanks for the kind feedback!

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)



