Windows Server 2022

After adding language packs, feature packs, cumulative updates, and other completions, components will have new changes, such as obsolete, pending deletion, etc. status. After adding language packs and feature packs to different versions, each language pack and feature pack has its corresponding version. There are not many changes in Windows 11 and Windows 10, but there are big changes when comparing Windows Server 2022 Datacenter and Windows Server 2022 Datacenter (Desktop Experience) versions. See the component analysis report for details.

1. **TABLE OF CONTENTS**
2. [Deployment image](#_Customize_the_deployment)
   * 1. [Prerequisites](#_PREREQUISITES)
     2. [Language package: Extract](#_LANGUAGE_PACKAGE:_EXTRACT)
     3. [Customize the deployment image](#_LANGUAGE_PACK:_INSTALL)
     4. [Generate ISO](#_4、生成_ISO)
3. [Common problem](#_COMMON_PROBLEM)
4. [Known issues](#_Known_issues_1)
5. [Component: unchanged](#_component:_unchanged)
6. [Component: changes after addition](#_component:_changes_after)

## **[deployment image](#_top)**

### **[PREREQUISITES](#_Customize_the_deployment)**

#### [**ISO tools**](#_PREREQUISITES)

Use a software that can edit ISO files, such as: PowerISO, DAEMON Tools, ISO Workshop;

#### **[Get related packages](#_PREREQUISITES)**

Learn "Attachment: OS" and view the "Windows Server 2022",

1. [**System installation package**](#_Get_related_packages)
   1. Prepare: en-us\_windows\_server\_2022\_x64\_dvd\_620d7eac.iso
   2. Unzip to: D:\en-us\_windows\_server\_2022\_x64\_dvd\_620d7eac
   3. After decompression is complete, change the directory en-us\_windows\_server\_2022\_x64\_dvd\_620d7eac to D:\OS2022
   4. All scripts and all paths have been set to D:\OS2022 by default as the image source.
2. [**Language Pack:**](#_Get_related_packages)

Download language pack: [20348.1.210507-1500.fe\_release\_amd64fre\_SERVER\_LOF\_PACKAGES\_OEM.iso](https://software-download.microsoft.com/download/sg/20348.1.210507-1500.fe_release_amd64fre_SERVER_LOF_PACKAGES_OEM.iso)

1. When running the PS command line:
2. Optional "Terminal" or "PowerShell ISE", if "Terminal" is not installed, please go to: <https://github.com/microsoft/terminal/releases> After downloading;
3. Open "Terminal" or "PowerShell ISE" as administrator, it is recommended to set the PowerShell execution policy: bypass, PS command line:

Set-ExecutionPolicy -ExecutionPolicy Bypass -Force

1. In this article, PS command line, green part, please copy it, paste it into the "Terminal" dialog box, press Enter and start running;
2. When there is .ps1, right-click the file and select Run with PowerShell, or copy the path and paste it into Terminal to run.

### **[LANGUAGE PACKAGE: EXTRACT](#_Customize_the_deployment)**

1. [**Prepare the language pack**](#_2、提取所需语言包)

Mounted 20348.1.210507-1500.fe\_release\_amd64fre\_SERVER\_LOF\_PACKAGES\_OEM.iso or unzip it to any location;

1. **Extract language pack scheme**

##### **Add**

* 1. Language name: Simplified Chinese - China, language tag: zh-CN, Scope of application: Install.Wim，Boot.Wim，WinRE.Wim

##### **Delete**

* 1. Language name: English - United States, language tag: en-US, Scope of application: Install.Wim，Boot.Wim，WinRE.Wim

1. **[Execute the extract command](#_2、提取所需语言包)**

* Auto = automatically search all local disks, default;
* Customize the path, for example, specify the E drive: $ISO = "E:\"

Copy the code or view the source file: Extract.ps1 ( [Local](Code/Extract.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Extract.ps1) )

$ISO = "Auto"

$SaveTo = "D:\OS2022\_Custom"

$Extract\_language\_Pack = @(

@{

Tag = "zh-CN"

Act = "Add"

Scope = @( "Install\Install"; "Install\WinRE"; "Boot\Boot" )

}

@{

Tag = "en-US"

Act = "Del"

Scope = @( "Install\Install"; "Install\WinRE"; "Boot\Boot" )

}

)

Function Extract\_Language

{

param

(

$Act, $NewLang, $Expand

)

Function Extract\_Process

{

param

(

$Package, $Name, $NewSaveTo

)

$NewSaveTo = "$($SaveTo)\$($NewSaveTo)\Language\$($Act)\$($NewLang)"

New-Item -Path $NewSaveTo -ItemType Directory -ErrorAction SilentlyContinue | Out-Null

if ($ISO -eq "Auto") {

Get-PSDrive -PSProvider FileSystem -ErrorAction SilentlyContinue | ForEach-Object {

ForEach ($item in $Package) {

$TempFilePath = Join-Path -Path $\_.Root -ChildPath $item -ErrorAction SilentlyContinue

if (Test-Path $TempFilePath -PathType Leaf) {

Write-host "`n Find: " -NoNewLine; Write-host $TempFilePath -ForegroundColor Green

Write-host " Copy to: " -NoNewLine; Write-host $NewSaveTo

Copy-Item -Path $TempFilePath -Destination $NewSaveTo -Force

}

}

}

} else {

ForEach ($item in $Package) {

$TempFilePath = Join-Path -Path $ISO -ChildPath $item -ErrorAction SilentlyContinue

Write-host "`n Find: " -NoNewline; Write-host $TempFilePath -ForegroundColor Green

if (Test-Path $TempFilePath -PathType Leaf) {

Write-host " Copy to: " -NoNewLine; Write-host $NewSaveTo

Copy-Item -Path $TempFilePath -Destination $NewSaveTo -Force

} else {

Write-host " Not found"

}

}

}

Write-host "`n Verify the language pack file"

ForEach ($item in $Package) {

$Path = "$($NewSaveTo)\$([IO.Path]::GetFileName($item))"

if (Test-Path $Path -PathType Leaf) {

Write-host " Discover: " -NoNewLine

Write-host $Path -ForegroundColor Green

} else {

Write-host " Not found: " -NoNewLine

Write-host $Path -ForegroundColor Red

}

}

}

$AdvLanguage = @(

@{

Path = "Install\Install"

Rule = @(

"LanguagesAndOptionalFeatures\Microsoft-Windows-LanguageFeatures-Fonts-{DiyLang}-Package~31bf3856ad364e35~amd64~~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-Server-Language-Pack\_x64\_{Lang}.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-LanguageFeatures-Basic-{Lang}-Package~31bf3856ad364e35~amd64~~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-LanguageFeatures-Handwriting-{Lang}-Package~31bf3856ad364e35~amd64~~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-LanguageFeatures-OCR-{Lang}-Package~31bf3856ad364e35~amd64~~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-LanguageFeatures-Speech-{Lang}-Package~31bf3856ad364e35~amd64~~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-LanguageFeatures-TextToSpeech-{Lang}-Package~31bf3856ad364e35~amd64~~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~{Lang}~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~wow64~{Lang}~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~amd64~{Lang}~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~wow64~{Lang}~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~{Lang}~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~{Lang}~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~{Lang}~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~{Lang}~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~{Lang}~.cab"

"LanguagesAndOptionalFeatures\Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~{Lang}~.cab"

)

}

@{

Path = "Install\WinRE"

Rule = @(

"Windows Preinstallation Environment\x64\WinPE\_OCs\WinPE-FontSupport-{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\lp.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-securestartup\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-atbroker\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-audiocore\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-audiodrivers\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-enhancedstorage\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-narrator\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-scripting\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-speech-tts\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-srh\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-srt\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-wds-tools\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-wmi\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-appxpackaging\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-storagewmi\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-wifi\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-rejuv\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-opcservices\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-hta\_{Lang}.cab"

)

}

@{

Path = "Boot\Boot"

Rule = @(

"Windows Preinstallation Environment\x64\WinPE\_OCs\WinPE-FontSupport-{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\lp.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\WinPE-Setup\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\WINPE-SETUP-Server\_{Lang}.CAB"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-securestartup\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-atbroker\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-audiocore\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-audiodrivers\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-enhancedstorage\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-narrator\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-scripting\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-speech-tts\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-srh\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-srt\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-wds-tools\_{Lang}.cab"

"Windows Preinstallation Environment\x64\WinPE\_OCs\{Lang}\winpe-wmi\_{Lang}.cab"

)

}

)

switch -regex ($NewLang) {

"ar-AE|ar-BH|ar-DJ|ar-DZ|ar-EG|ar-ER|ar-IL|ar-IQ|ar-JO|ar-KM|ar-KW|ar-LB|ar-LY|ar-MA|ar-MR|ar-OM|ar-PS|ar-QA|ar-SA|ar-SD|ar-SO|ar-SS|ar-SY|ar-TD|ar-TN|ar-YE|arz-Arab|ckb-Arab|fa-AF|fa-IR|glk-Arab|ha-Arab|ks-Arab|ks-Arab-IN|ku-Arab|ku-Arab-IQ|mzn-Arab|pa-Arab|pa-Arab-PK|pnb-Arab|prs-AF|prs-Arab|ps-AF|sd-Arab|sd-Arab-PK|tk-Arab|ug-Arab|ug-CN|ur-IN|ur-PK|uz-Arab|uz-Arab-AF" { $NewDiyLanguage = "Arab" }

"as-IN|bn-BD|bn-IN|bpy-Beng" { $NewDiyLanguage = "Beng" }

"da-dk|iu-Cans|iu-Cans-CA" { $NewDiyLanguage = "Cans" }

"chr-Cher-US|chr-Cher" { $NewDiyLanguage = "Cher" }

"bh-Deva|brx-Deva|brx-IN|hi-IN|ks-Deva|mr-IN|ne-IN|ne-NP|new-Deva|pi-Deva|sa-Deva|sa-IN" { $NewDiyLanguage = "Deva" }

"am-ET|byn-ER|byn-Ethi|ti-ER|ti-ET|tig-ER|tig-Ethi|ve-Ethi|wal-ET|wal-Ethi" { $NewDiyLanguage = "Ethi" }

"gu-IN" { $NewDiyLanguage = "Gujr" }

"pa-Guru|pa-IN" { $NewDiyLanguage = "Guru" }

"cmn-Hans|gan-Hans|hak-Hans|wuu-Hans|yue-Hans|zh-CN|zh-gan-Hans|zh-hak-Hans|zh-Hans|zh-SG|zh-wuu-Hans|zh-yue-Hans" { $NewDiyLanguage = "Hans" }

"cmn-Hant|hak-Hant|lzh-Hant|zh-hak-Hant|zh-Hant|zh-HK|zh-lzh-Hant|zh-MO|zh-TW|zh-yue-Hant" { $NewDiyLanguage = "Hant" }

"he-IL|yi" { $NewDiyLanguage = "Hebr" }

"ja-JP" { $NewDiyLanguage = "Jpan" }

"km-KH" { $NewDiyLanguage = "Khmr" }

"kn-IN" { $NewDiyLanguage = "Knda" }

"ko-KR" { $NewDiyLanguage = "Kore" }

"de-de|lo-LA" { $NewDiyLanguage = "Laoo" }

"ml-IN" { $NewDiyLanguage = "Mlym" }

"or-IN" { $NewDiyLanguage = "Orya" }

"si-LK" { $NewDiyLanguage = "Sinh" }

"tr-tr|arc-Syrc|syr-SY|syr-Syrc" { $NewDiyLanguage = "Syrc" }

"ta-IN|ta-LK|ta-MY|ta-SG" { $NewDiyLanguage = "Taml" }

"te-IN" { $NewDiyLanguage = "Telu" }

"th-TH" { $NewDiyLanguage = "Thai" }

default { $NewDiyLanguage = "skip\_$($NewLanguage)\_" }

}

Foreach ($item in $Expand) {

$Language = @()

Foreach ($itemList in $AdvLanguage) {

if ($itemList.Path -eq $item) {

Foreach ($PrintLang in $itemList.Rule) {

$Language += "$($PrintLang)".Replace("{Lang}", $NewLang).Replace("{DiyLang}", $NewDiyLanguage)

}

Extract\_Process -NewSaveTo $itemList.Path -Package $Language -Name $item

}

}

}

}

ForEach ($item in $Extract\_language\_Pack) {

Extract\_Language -Act $item.Act -NewLang $item.Tag -Expand $item.Scope

}

### **[Customize the deployment image](#_Customize_the_deployment)**

#### [**Custom deployment image: Install.wim**](#_LANGUAGE_PACK:_INSTALL)

##### [**View Install.wim details**](#_3.1__)

Image name, image description, image size, architecture, version, index number, etc.;

$ViewFile = "D:\OS2022\Sources\Install.wim"

Get-WindowsImage -ImagePath $ViewFile | Foreach-Object {

Get-WindowsImage -ImagePath $ViewFile -index $\_.ImageIndex

}

CYCLIC OPERATION AREA, START,

##### **[Specify the path to mount install.wim](#_3.1__)**

New-Item -Path "D:\OS2022\_Custom\Install\Install\Mount" -ItemType directory -ea SilentlyContinue

1. [**Start mounting Install.wim**](#_Custom_deployment_image:)

Default index number: 1

$Index = "1"

Mount-WindowsImage -ImagePath "D:\OS2022\sources\install.wim" -Index $Index -Path "D:\OS2022\_Custom\Install\Install\Mount"

PROCESS FILES INSIDE THE INSTALL.WIM IMAGE, OPTIONALLY, START

###### **[Custom deployment image: WinRE.wim](#_3.1__)**

WARNING:

* WinRE.wim is a file within the Install.wim image;
* When Install.wim has multiple index numbers, only process any WinRE.wim;
* Synchronizing to all index numbers reduces the Install.wim volume; Learn "[How to bulk replace WinRE.wim in all index numbers in Install.wim](#_How_to_extract)".
  + 1. [**View WinRE.wim details**](#_3.1.2.1__)

Image name, image description, image size, architecture, version, index number, etc.;

$ViewFile = "D:\OS2022\_Custom\Install\Install\Mount\Windows\System32\Recovery\WinRE.wim"

Get-WindowsImage -ImagePath $ViewFile | Foreach-Object {

Get-WindowsImage -ImagePath $ViewFile -index $\_.ImageIndex

}

* + 1. [**Specify the path to mount WinRE.wim**](#_3.1.2.1__)

New-Item -Path "D:\OS2022\_Custom\Install\WinRE\Mount" -ItemType directory -ea SilentlyContinue

* + 1. [**Start mounting WinRE.wim**](#_3.1.2.1__)

Default index number: 1

$Index = "1"

$Mount =

$FileName = "D:\OS2022\_Custom\Install\Install\Mount\Windows\System32\Recovery\WinRE.wim"

Mount-WindowsImage -ImagePath $FileName -Index $Index -Path "D:\OS2022\_Custom\Install\WinRE\Mount"

* + 1. **Language pack**

Automatically install language packs: Match by component. After successful matching, install the corresponding local language pack file.

* + - 1. **Language pack: add**

Copy the code or view the source file: WinRE.Instl.lang.ps1 ( [Local](Code/Install/WinRE/WinRE.Instl.lang.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/WinRE/WinRE.Instl.lang.ps1) )

$Mount = "D:\OS2022\_Custom\Install\WinRE\Mount"

$Sources = "D:\OS2022\_Custom\Install\WinRE\Language\Add\zh-CN"

$Initl\_install\_Language\_Component = @()

Get-WindowsPackage -Path $Mount | ForEach-Object {

$Initl\_install\_Language\_Component += $\_.PackageName

}

Add-WindowsPackage -Path $Mount -PackagePath "$($Sources)\WinPE-FontSupport-zh-CN.cab"

$Language\_List = @(

@{ Match = "\*WinPE-LanguagePack-Package\*"; File = "lp.cab"; }

@{ Match = "\*SecureStartup\*"; File = "winpe-securestartup\_zh-CN.cab"; }

@{ Match = "\*ATBroker\*"; File = "winpe-atbroker\_zh-CN.cab"; }

@{ Match = "\*AudioCore\*"; File = "winpe-audiocore\_zh-CN.cab"; }

@{ Match = "\*AudioDrivers\*"; File = "winpe-audiodrivers\_zh-CN.cab"; }

@{ Match = "\*EnhancedStorage\*"; File = "winpe-enhancedstorage\_zh-CN.cab"; }

@{ Match = "\*Narrator\*"; File = "winpe-narrator\_zh-CN.cab"; }

@{ Match = "\*scripting\*"; File = "winpe-scripting\_zh-CN.cab"; }

@{ Match = "\*Speech-TTS\*"; File = "winpe-speech-tts\_zh-CN.cab"; }

@{ Match = "\*srh\*"; File = "winpe-srh\_zh-CN.cab"; }

@{ Match = "\*srt\*"; File = "winpe-srt\_zh-CN.cab"; }

@{ Match = "\*wds-tools\*"; File = "winpe-wds-tools\_zh-CN.cab"; }

@{ Match = "\*-WMI-Package\*"; File = "winpe-wmi\_zh-CN.cab"; }

@{ Match = "\*WinPE-AppxPackaging\*"; File = "winpe-appxpackaging\_zh-CN.cab"; }

@{ Match = "\*StorageWMI\*"; File = "winpe-storagewmi\_zh-CN.cab"; }

@{ Match = "\*WiFi\*"; File = "winpe-wifi\_zh-CN.cab"; }

@{ Match = "\*rejuv\*"; File = "winpe-rejuv\_zh-CN.cab"; }

@{ Match = "\*opcservices\*"; File = "winpe-opcservices\_zh-CN.cab"; }

@{ Match = "\*hta\*"; File = "winpe-hta\_zh-CN.cab"; }

)

ForEach ($Rule in $Language\_List) {

Write-host "`n Rule name: $($Rule.Match)" -ForegroundColor Yellow; Write-host " $('-' \* 80)"

ForEach ($Component in $Initl\_install\_Language\_Component) {

if ($Component -like "\*$($Rule.Match)\*") {

Write-host " Component name: " -NoNewline

Write-host $Component -ForegroundColor Green

Write-host " Language pack file: " -NoNewline

Write-host "$($Sources)\$($Rule.File)" -ForegroundColor Green

Write-Host " Installing ".PadRight(22) -NoNewline

try {

Add-WindowsPackage -Path $Mount -PackagePath "$($Sources)\$($Rule.File)" | Out-Null

Write-host "Finish" -ForegroundColor Green

} catch {

Write-host "Failed" -ForegroundColor Red

}

break

}

}

}

* + - 1. **Offline image language: change**
         1. **Change default language, regional settings, and other international settings**

Dism /Image:"D:\OS2022\_Custom\Install\WinRE\Mount" /Set-AllIntl:zh-CN

* + - * 1. **View available language settings**

Dism /Image:"D:\OS2022\_Custom\Install\WinRE\Mount" /Get-Intl

* + - 1. [**Language packs: Removed, optional**](#_执行安装命令后安装语言包到：WinRE)
* After you add languages, if you want to deploy to a non-English locale, you can save space by removing the English language component. When you remove a language, uninstall the language components in the reverse order in which you added them.
* After adding the Chinese, delete "English - United States" in reverse, the language tag: en-US, you need to extract the language pack in advance

Copy the code or view the source files: WinRE.Del.Specified.lang.Tag.ps1 ( [Local](Code/Install/WinRE/WinRE.Del.Specified.lang.Tag.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/WinRE/WinRE.Del.Specified.lang.Tag.ps1) )

$Lang = "en-US"

$Mount = "D:\OS2022\_Custom\Install\WinRE\Mount"

$Sources = "D:\OS2022\_Custom\Install\WinRE\Language\Del\en-US"

$Initl\_install\_Language\_Component = @()

Get-WindowsPackage -Path $Mount | ForEach-Object {

$Initl\_install\_Language\_Component += $\_.PackageName

}

$Language = @(

@{ Match = "\*windowsupdate\*"; File = "winpe-windowsupdate\_$($Lang).cab"; }

@{ Match = "\*appxdeployment\*"; File = "winpe-appxdeployment\_$($Lang).cab"; }

@{ Match = "\*hta\*"; File = "winpe-hta\_$($Lang).cab"; }

@{ Match = "\*opcservices\*"; File = "winpe-opcservices\_$($Lang).cab"; }

@{ Match = "\*rejuv\*"; File = "winpe-rejuv\_$($Lang).cab"; }

@{ Match = "\*WiFi\*"; File = "winpe-wifi\_$($Lang).cab"; }

@{ Match = "\*StorageWMI\*"; File = "winpe-storagewmi\_$($Lang).cab"; }

@{ Match = "\*WinPE-AppxPackaging\*"; File = "winpe-appxpackaging\_$($Lang).cab"; }

@{ Match = "\*-WMI-Package\*"; File = "winpe-wmi\_$($Lang).cab"; }

@{ Match = "\*wds-tools\*"; File = "winpe-wds-tools\_$($Lang).cab"; }

@{ Match = "\*srt\*"; File = "winpe-srt\_$($Lang).cab"; }

@{ Match = "\*srh\*"; File = "winpe-srh\_$($Lang).cab"; }

@{ Match = "\*Speech-TTS\*"; File = "winpe-speech-tts\_$($Lang).cab"; }

@{ Match = "\*scripting\*"; File = "winpe-scripting\_$($Lang).cab"; }

@{ Match = "\*Narrator\*"; File = "winpe-narrator\_$($Lang).cab"; }

@{ Match = "\*EnhancedStorage\*"; File = "winpe-enhancedstorage\_$($Lang).cab"; }

@{ Match = "\*AudioDrivers\*"; File = "winpe-audiodrivers\_$($Lang).cab"; }

@{ Match = "\*AudioCore\*"; File = "winpe-audiocore\_$($Lang).cab"; }

@{ Match = "\*ATBroker\*"; File = "winpe-atbroker\_$($Lang).cab"; }

@{ Match = "\*SecureStartup\*"; File = "winpe-securestartup\_$($Lang).cab"; }

@{ Match = "\*WinPE-LanguagePack-Package\*"; File = "lp.cab"; }

)

ForEach ($Rule in $Language) {

Write-host "`n Rule name: $($Rule.Match)" -ForegroundColor Yellow; Write-host " $('-' \* 80)"

ForEach ($Component in $Initl\_install\_Language\_Component) {

if ($Component -like "\*$($Rule.Match)\*$($Lang)\*") {

Write-host " Component name: " -NoNewline

Write-host $Component -ForegroundColor Green

Write-host " Language pack file: " -NoNewline

Write-host "$($Sources)\$($Rule.File)" -ForegroundColor Green

Write-Host " Deleting ".PadRight(22) -NoNewline

try {

Remove-WindowsPackage -Path $Mount -PackagePath "$($Sources)\$($Rule.File)" -ErrorAction SilentlyContinue | Out-Null

Write-host "Finish" -ForegroundColor Green

} catch {

Write-host "Failed" -ForegroundColor Red

Write-host " $($\_)" -ForegroundColor Red

}

break

}

}

}

$InitlClearLanguagePackage = @()

Get-WindowsPackage -Path $Mount | ForEach-Object {

if ($\_.PackageName -like "\*$($Lang)\*") {

$InitlClearLanguagePackage += $\_.PackageName

}

}

if ($InitlClearLanguagePackage.count -gt 0) {

ForEach ($item in $InitlClearLanguagePackage) {

Write-Host "`n $($item)" -ForegroundColor Green

Write-Host " Deleting ".PadRight(22) -NoNewline

try {

Remove-AppxProvisionedPackage -Path $Mount -PackageName $item -ErrorAction SilentlyContinue | Out-Null

Write-host "Finish" -ForegroundColor Green

} catch {

Write-host "Failed" -ForegroundColor Red

Write-host " $($\_)" -ForegroundColor Red

}

}

}

* + - 1. **Components**
         1. **View**

Get-WindowsPackage -Path "D:\OS2022\_Custom\Install\WinRE\Mount" | Out-GridView

* + - * 1. **Export to Csv**

$SaveTo = "D:\OS2022\_Custom\Install\WinRE\Report.$(Get-Date -Format "yyyyMMddHHmmss").csv"

Get-WindowsPackage -Path "D:\OS2022\_Custom\Install\WinRE\Mount" | Export-CSV -NoType -Path $SaveTo

Write-host $SaveTo -ForegroundColor Green

* + 1. [**Update, optional**](#_3.1.2.1__)

To prepare the update file available, change the example file name: KB\_WinRE.cab

* + - 1. **Add**

$KBPath = "D:\OS2022\_Custom\Install\WinRE\Update\KB\_Boot.cab"

Add-WindowsPackage -Path "D:\OS2022\_Custom\Install\WinRE\Mount" -PackagePath $KBPath

* + - 1. **Delete**

$KBPath = "D:\OS2022\_Custom\Install\WinRE\Update\KB\_Boot.cab"

Remove-WindowsPackage -Path "D:\OS2022\_Custom\Install\WinRE\Mount" -PackagePath $KBPath

* + - 1. **Solid update, optional**

It cannot be uninstalled after curing, which cleans the recovery image and resets the basis of any superseded components.

Dism /image:"D:\OS2022\_Custom\Install\WinRE\Mount" /cleanup-image /StartComponentCleanup /ResetBase

* + - * 1. [**Clean components after curing and updating, optional**](#_固化更新，可选项)

$Mount = "D:\OS2022\_Custom\Install\WinRE\Mount"

Get-WindowsPackage -Path $Mount -ErrorAction SilentlyContinue | ForEach-Object {

if ($\_.PackageState -eq "Superseded") {

Write-Host " $($\_.PackageName)" -ForegroundColor Green

Remove-WindowsPackage -Path $Mount -PackageName $\_.PackageName | Out-Null

}

}

* + 1. [**Driver, optional**](#_3.1.2.1__)
    2. [**Save image**](#_3.1.2.1__)

Save-WindowsImage -Path "D:\OS2022\_Custom\Install\WinRE\Mount"

* + 1. [**Unmount image**](#_3.1.2.1__)

Close any applications that may be accessing files in the image, including File Explorer.

Dismount-WindowsImage -Path "D:\OS2022\_Custom\Install\WinRE\Mount" -Discard

* + 1. [**After rebuilding WinRE.wim, the file size can be reduced**](#_3.1.2.1__)

Copy the code or view the source file: WinRE.Rebuild.ps1 ( [Local](Code/Install/WinRE/WinRE.Rebuild.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/WinRE/WinRE.Rebuild.ps1) )

$FileName = "D:\OS2022\_Custom\Install\Install\Mount\Windows\System32\Recovery\WinRE.wim"

Get-WindowsImage -ImagePath $Filename -ErrorAction SilentlyContinue | ForEach-Object {

Write-Host " Image name: " -NoNewline

Write-Host $\_.ImageName -ForegroundColor Yellow

Write-Host " The index number: " -NoNewline

Write-Host $\_.ImageIndex -ForegroundColor Yellow

Write-Host "`n Rebuilding ".PadRight(28) -NoNewline

Export-WindowsImage -SourceImagePath $Filename -SourceIndex $\_.ImageIndex -DestinationImagePath "$($FileName).New" -CompressionType max

Write-Host "Finish`n" -ForegroundColor Green

}

if (Test-Path "$($FileName).New" -PathType Leaf) {

Remove-Item -Path $Filename

Move-Item -Path "$($FileName).New" -Destination $Filename

Write-Host "Finish" -ForegroundColor Green

} else {

Write-host "Failed" -ForegroundColor Red

}

* + 1. [**Backup WinRE.wim**](#_3.1.2.1__)

Copy the code or view the source file: WinRE.Backup.ps1 ( [Local](Code/Install/WinRE/WinRE.Backup.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/WinRE/WinRE.Backup.ps1) )

$WimLibPath = "D:\OS2022\_Custom\Install\Install\Update\Winlib"

$FileName = "D:\OS2022\_Custom\Install\Install\Mount\Windows\System32\Recovery\WinRE.wim"

New-Item -Path $WimLibPath -ItemType Directory -ea SilentlyContinue

Copy-Item -Path $FileName -Destination $WimLibPath -Force

* + 1. [**Replace WinRE.wim within the Install.wim image**](#_3.1.2.1__)
* After each installation of Install.wim, use item "[Replace the WinRE.wim](#_Replace_the_WinRE.wim)";
* Learning "[After obtaining all the index numbers of Install.wim and replace the old WinRE.wim](#_After_getting_all)".

PROCESS FILES INSIDE THE INSTALL.WIM IMAGE, END

##### **Language pack**

Automatically install language packs: Match by component. After successful matching, install the corresponding local language pack file.

###### **Language pack: add**

Copy the code or view the source file: Install.Instl.lang.ps1 ( [Local](Code/Install/Install.Instl.lang.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/Install.Instl.lang.ps1) )

$Mount = "D:\OS2022\_Custom\Install\Install\Mount"

$Sources = "D:\OS2022\_Custom\Install\Install\Language\Add\zh-CN"

$Initl\_install\_Language\_Component = @()

Get-WindowsPackage -Path $Mount | ForEach-Object {

$Initl\_install\_Language\_Component += $\_.PackageName

}

Add-WindowsPackage -Path $Mount -PackagePath "$($Sources)\Microsoft-Windows-LanguageFeatures-Fonts-Hans-Package~31bf3856ad364e35~amd64~~.cab"

$Language\_List = @(

@{ Match = "\*Server-LanguagePack-Package\*"; File = "Microsoft-Windows-Server-Language-Pack\_x64\_zh-CN.cab"; }

@{ Match = "\*LanguageFeatures-Basic\*"; File = "Microsoft-Windows-LanguageFeatures-Basic-zh-CN-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*Handwriting\*"; File = "Microsoft-Windows-LanguageFeatures-Handwriting-zh-CN-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*OCR\*"; File = "Microsoft-Windows-LanguageFeatures-OCR-zh-CN-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*LanguageFeatures-Speech\*"; File = "Microsoft-Windows-LanguageFeatures-Speech-zh-CN-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*TextToSpeech\*"; File = "Microsoft-Windows-LanguageFeatures-TextToSpeech-zh-CN-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*MSPaint\*amd64\*"; File = "Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~zh-CN~.cab"; }

@{ Match = "\*MSPaint\*wow64\*"; File = "Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~wow64~zh-CN~.cab"; }

@{ Match = "\*Notepad\*amd64\*"; File = "Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~amd64~zh-CN~.cab"; }

@{ Match = "\*Notepad\*wow64\*"; File = "Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~wow64~zh-CN~.cab"; }

@{ Match = "\*PowerShell\*amd64\*"; File = "Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~zh-CN~.cab"; }

@{ Match = "\*PowerShell\*wow64\*"; File = "Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~zh-CN~.cab"; }

@{ Match = "\*StepsRecorder\*amd64\*"; File = "Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~zh-CN~.cab"; }

@{ Match = "\*StepsRecorder\*wow64\*"; File = "Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~zh-CN~.cab"; }

@{ Match = "\*WordPad\*amd64\*"; File = "Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~zh-CN~.cab"; }

@{ Match = "\*WordPad\*wow64\*"; File = "Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~zh-CN~.cab"; }

)

ForEach ($Rule in $Language\_List) {

Write-host "`n Rule name: $($Rule.Match)" -ForegroundColor Yellow; Write-host " $('-' \* 80)"

ForEach ($Component in $Initl\_install\_Language\_Component) {

if ($Component -like "\*$($Rule.Match)\*") {

Write-host " Component name: " -NoNewline

Write-host $Component -ForegroundColor Green

Write-host " Language pack file: " -NoNewline

Write-host "$($Sources)\$($Rule.File)" -ForegroundColor Green

Write-Host " Installing ".PadRight(22) -NoNewline

try {

Add-WindowsPackage -Path $Mount -PackagePath "$($Sources)\$($Rule.File)" | Out-Null

Write-host "Finish" -ForegroundColor Green

break

} catch {

Write-host "Failed" -ForegroundColor Red

}

}

}

}

###### **Offline image language: change**

* tarting Windows 11, the [default System UI Language](https://learn.microsoft.com/en-us/windows/win32/intl/user-interface-language-management) set by DISM is left unaltered on all editions except for Home edition. For all [commercial editions](https://www.microsoft.com/windowsforbusiness/compare) the language chosen during the Out-of-Box Experience (OOBE) is set as the [System Preferred UI language](https://learn.microsoft.com/en-us/windows/win32/intl/user-interface-language-management) and Windows will be displayed in this language and for Home edition the language chosen at OOBE will continue to be the default System UI Language.
* As of Windows 10, version 2004, if an .appx-based Language Experience Pack (LXP) backed language is passed as an argument then the language will be set as the System Preferred UI language and its parent language will be set as the Default System UI language. In prior versions only .cab based language packs were supported.
  + 1. **Change default language, regional settings, and other international settings**

Dism /Image:"D:\OS2022\_Custom\Install\Install\Mount" /Set-AllIntl:zh-CN

* + 1. **View available language settings**

Dism /Image:"D:\OS2022\_Custom\Install\Install\Mount" /Get-Intl

###### [**Language packs: Removed, optional**](#_执行安装命令后安装语言包到：WinRE)

* After you add languages, if you want to deploy to a non-English locale, you can save space by removing the English language component. When you remove a language, uninstall the language components in the reverse order in which you added them.
* After adding the Chinese, delete "English - United States" in reverse, the language tag: en-US, you need to extract the language pack in advance

Copy the code or view the source files: Install.Del.Specified.lang.Tag.ps1 ( [Local](Code/Install/Install.Del.Specified.lang.Tag.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/Install/WinRE.Del.Specified.lang.Tag.ps1) )

$Lang = "en-US"

$Mount = "D:\OS2022\_Custom\Install\Install\Mount"

$Sources = "D:\OS2022\_Custom\Install\Install\Language\Del\en-US"

$Initl\_install\_Language\_Component = @()

Get-WindowsPackage -Path $Mount | ForEach-Object {

$Initl\_install\_Language\_Component += $\_.PackageName

}

$Language = @(

@{ Match = "\*WordPad\*wow64\*"; File = "Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*WordPad\*amd64\*"; File = "Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~$($Lang)~.cab"; }

@{ Match = "\*WMIC\*FoD\*Package\*wow64\*"; File = "Microsoft-Windows-WMIC-FoD-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*WMIC\*FoD\*Package\*amd64\*"; File = "Microsoft-Windows-WMIC-FoD-Package~31bf3856ad364e35~AMD64~$($Lang)~.cab"; }

@{ Match = "\*StepsRecorder\*wow64\*"; File = "Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*StepsRecorder\*amd64\*"; File = "Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~$($Lang)~.cab"; }

@{ Match = "\*Printing\*PMCPPC\*amd64\*"; File = "Microsoft-Windows-Printing-PMCPPC-FoD-Package~31bf3856ad364e35~AMD64~$($Lang)~.cab"; }

@{ Match = "\*PowerShell\*wow64\*"; File = "Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*PowerShell\*amd64\*"; File = "Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~$($Lang)~.cab"; }

@{ Match = "\*MediaPlayer\*wow64\*"; File = "Microsoft-Windows-MediaPlayer-Package-wow64-$($Lang).cab"; }

@{ Match = "\*MediaPlayer\*amd64\*"; File = "Microsoft-Windows-MediaPlayer-Package-AMD64-$($Lang).cab"; }

@{ Match = "\*Notepad\*wow64\*"; File = "Microsoft-Windows-Notepad-System-FoD-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*Notepad\*amd64\*"; File = "Microsoft-Windows-Notepad-System-FoD-Package~31bf3856ad364e35~AMD64~$($Lang)~.cab"; }

@{ Match = "\*InternetExplorer\*"; File = "Microsoft-Windows-InternetExplorer-Optional-Package~31bf3856ad364e35~AMD64~$($Lang)~.cab"; }

@{ Match = "\*TextToSpeech\*"; File = "Microsoft-Windows-LanguageFeatures-TextToSpeech-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*LanguageFeatures-Speech\*"; File = "Microsoft-Windows-LanguageFeatures-Speech-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*OCR\*"; File = "Microsoft-Windows-LanguageFeatures-OCR-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*Handwriting\*"; File = "Microsoft-Windows-LanguageFeatures-Handwriting-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*LanguageFeatures-Basic\*"; File = "Microsoft-Windows-LanguageFeatures-Basic-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*Client-LanguagePack-Package\*"; File = "Microsoft-Windows-Client-Language-Pack\_x64\_$($Lang).cab"; }

)

ForEach ($Rule in $Language) {

Write-host "`n Rule name: $($Rule.Match)" -ForegroundColor Yellow; Write-host " $('-' \* 80)"

ForEach ($Component in $Initl\_install\_Language\_Component) {

if ($Component -like "\*$($Rule.Match)\*$($Lang)\*") {

Write-host " Component name: " -NoNewline

Write-host $Component -ForegroundColor Green

Write-host " Language pack file: " -NoNewline

Write-host "$($Sources)\$($Rule.File)" -ForegroundColor Green

Write-Host " Deleting ".PadRight(22) -NoNewline

try {

Remove-WindowsPackage -Path $Mount -PackagePath "$($Sources)\$($Rule.File)" -ErrorAction SilentlyContinue | Out-Null

Write-host "Finish" -ForegroundColor Green

} catch {

Write-host "Failed" -ForegroundColor Red

Write-host " $($\_)" -ForegroundColor Red

}

break

}

}

}

$InitlClearLanguagePackage = @()

Get-WindowsPackage -Path $Mount | ForEach-Object {

if ($\_.PackageName -like "\*$($Lang)\*") {

$InitlClearLanguagePackage += $\_.PackageName

}

}

if ($InitlClearLanguagePackage.count -gt 0) {

ForEach ($item in $InitlClearLanguagePackage) {

Write-Host "`n $($item)" -ForegroundColor Green

Write-Host " Deleting ".PadRight(22) -NoNewline

try {

Remove-AppxProvisionedPackage -Path $Mount -PackageName $item -ErrorAction SilentlyContinue | Out-Null

Write-host "Finish" -ForegroundColor Green

} catch {

Write-host "Failed" -ForegroundColor Red

Write-host " $($\_)" -ForegroundColor Red

}

}

}

* 1. **Components**
     1. **View**

Get-WindowsPackage -Path "D:\OS2022\_Custom\Install\Install\Mount" | Out-GridView

* + 1. **Export to Csv**

$SaveTo = "D:\OS2022\_Custom\Install\Install\Report.$(Get-Date -Format "yyyyMMddHHmmss").csv"

Get-WindowsPackage -Path "D:\OS2022\_Custom\Install\Install\Mount" | Export-CSV -NoType -Path $SaveTo

Write-host $SaveTo -ForegroundColor Green

##### **[Updates, optional](#_3.1__)**

###### [**Download**](#_Updates,_optional)

Check the "[Windows Server 2022 Update History](https://support.microsoft.com/en-gb/topic/windows-server-2022-update-history-e1caa597-00c5-4ab9-9f3e-8212fe80b2ee) ", for example, install the cumulative update: KB5030216

Go to the download page: <https://www.catalog.update.microsoft.com/Search.aspx?q=Kb5030216> Or "[Direct download](https://catalog.s.download.windowsupdate.com/c/msdownload/update/software/secu/2023/09/windows10.0-kb5030216-x64_cbe587155f9818548b75f65d5cd41d341ed2fc61.msu)" (If you cannot download, please go to the download page), save to

D:\OS2022\_Custom\Install\Install\Update\windows10.0-kb5030216-x64\_cbe587155f9818548b75f65d5cd41d341ed2fc61.msu

###### [**Add**](#_Updates,_optional)

$KBPath = "D:\OS2022\_Custom\Install\Install\Update\windows10.0-kb5030216-x64\_cbe587155f9818548b75f65d5cd41d341ed2fc61.msu"

Add-WindowsPackage -Path "D:\OS2022\_Custom\Install\Install\Mount" -PackagePath $KBPath

###### **[Solid update, optional](#_Updates,_optional)**

It cannot be uninstalled after curing, which cleans the recovery image and resets the basis of any superseded components.

Dism /Image:"D:\OS2022\_Custom\Install\Install\Mount" /cleanup-image /StartComponentCleanup /ResetBase

* + 1. [**Clean up components after curing updates**](#_Solid_update,_optional)

Copy the code or view the source file: Install.Update.Curing.ps1 ( [Local](Code/Install/Install.Update.Curing.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/Install.Update.Curing.ps1) )

$Mount = "D:\OS2022\_Custom\Install\Install\Mount"

Get-WindowsPackage -Path $Mount -ErrorAction SilentlyContinue | ForEach-Object {

if ($\_.PackageState -eq "Superseded") {

Write-Host " $($\_.PackageName)" -ForegroundColor Green

Remove-WindowsPackage -Path $Mount -PackageName $\_.PackageName | Out-Null

}

}

##### [**Drive, optional**](#_3.1__)

##### [**Health**](#_3.1__)

Check whether there is any damage before saving. When the health status is abnormal, abort saving

Repair-WindowsImage -Path "D:\OS2022\_Custom\Install\Install\Mount" -ScanHealth

##### **[Replace the WinRE.wim](#_3.1__)**

WinRE.wim in all index numbers in Install.wim has been replaced in batches. Please skip this step.

$WinRE = "D:\OS2022\_Custom\Install\Install\Update\Winlib\WinRE.wim"

$CopyTo = "D:\OS2022\_Custom\Install\Install\Mount\Windows\System32\Recovery"

Copy-Item -Path $WinRE -Destination $CopyTo -Force

##### [**Save image**](#_3.1__)

Save-WindowsImage -Path "D:\OS2022\_Custom\Install\Install\Mount"

##### [**Unmount image**](#_3.1__)

Close any applications that may be accessing files in the image, including File Explorer.

Dismount-WindowsImage -Path "D:\OS2022\_Custom\Install\Install\Mount" -Discard

CYCLIC OPERATION AREA, END.

##### [**Replace WinRE.wim in all index numbers in Install.wim**](#_3.1__)

###### [**Get WimLib**](#_Replace_WinRE.wim_in)

After going to the official website of <https://wimlib.net>, select a different version: [arm64](https://wimlib.net/downloads/wimlib-1.14.3-windows-aarch64-bin.zip), [x64](https://wimlib.net/downloads/wimlib-1.14.3-windows-x86_64-bin.zip), [x86](https://wimlib.net/downloads/wimlib-1.14.3-windows-i686-bin.zip), and extract it to: D:Wimlib after downloading.

###### **[How to extract and update WinRE.wim in Install.wim](#_Replace_WinRE.wim_in)**

* + 1. [**Extract the WinRE.wim file from Install.wim**](#_How_to_extract_1)

Copy the code or view the source file: Install.WinRE.Extract.ps1 ( [Local](Code/Install/Install.WinRE.Extract.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/Install.WinRE.Extract.ps1) )

$Arguments = @(

"extract",

"D:\OS2022\sources\install.wim", "1",

"\Windows\System32\Recovery\Winre.wim",

"--dest-dir=""D:\OS2022\_Custom\Install\Install\Update\Winlib"""

)

New-Item -Path "D:\OS2022\_Custom\Install\Install\Update\Winlib" -ItemType Directory -ea SilentlyContinue

Start-Process -FilePath "d:\wimlib\wimlib-imagex.exe" -ArgumentList $Arguments -wait -nonewwindow

* + 1. **[After getting all index numbers of Install.wim and replacing the old WinRE.wim](#_How_to_extract_1)**

Copy the code or view the source file: Install.WinRE.Replace.wim.ps1 ( [Local](Code/Install/Install.WinRE.Replace.wim.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/Install.WinRE.Replace.wim.ps1) )

Get-WindowsImage -ImagePath "D:\OS2022\sources\install.wim" -ErrorAction SilentlyContinue | ForEach-Object {

Write-Host " Image name: " -NoNewline

Write-Host $\_.ImageName -ForegroundColor Yellow

Write-Host " The index number: " -NoNewline

Write-Host $\_.ImageIndex -ForegroundColor Yellow

Write-Host "`n Replacement "

$Arguments = @(

"update",

"D:\OS2022\sources\install.wim",

$\_.ImageIndex,

"--command=""add 'D:\OS2022\_Custom\Install\Install\Update\Winlib\WinRE.wim' '\Windows\System32\Recovery\WinRe.wim'"""

)

Start-Process -FilePath "d:\wimlib\wimlib-imagex.exe" -ArgumentList $Arguments -wait -nonewwindow

Write-Host " Finish`n" -ForegroundColor Green

}

##### [**Rebuilding Install.wim reduces file size**](#_Install_language_pack)

Copy the code or view the source file: Install.Rebuild.wim.ps1 ( [Local](Code/Install/Install.Rebuild.wim.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Install/Install.Rebuild.wim.ps1) )

$InstallWim = "D:\OS2022\sources\install.wim"

Get-WindowsImage -ImagePath $InstallWim -ErrorAction SilentlyContinue | ForEach-Object {

Write-Host " Image name: " -NoNewline

Write-Host $\_.ImageName -ForegroundColor Yellow

Write-Host " The index number: " -NoNewline

Write-Host $\_.ImageIndex -ForegroundColor Yellow

Write-Host "`n Rebuilding".PadRight(28) -NoNewline

Export-WindowsImage -SourceImagePath $InstallWim -SourceIndex $\_.ImageIndex -DestinationImagePath "$($InstallWim).New" -CompressionType max | Out-Null

Write-Host "Finish`n" -ForegroundColor Green

}

if (Test-Path "$($InstallWim).New" -PathType Leaf) {

Remove-Item -Path $InstallWim

Move-Item -Path "$($InstallWim).New" -Destination $InstallWim

Write-Host "Finish" -ForegroundColor Green

} else {

Write-host "Failed" -ForegroundColor Red

}

#### [**Custom deployment image: boot.wim**](#_INSTALL_LANGUAGE_PACK)

##### [**View Boot.wim details**](#_Install_language_pack_1)

Image name, image description, image size, architecture, version, index number, etc.;

$ViewFile = "D:\OS2022\Sources\Boot.wim"

Get-WindowsImage -ImagePath $ViewFile | Foreach-Object {

Get-WindowsImage -ImagePath $ViewFile -index $\_.ImageIndex

}

##### [**Specify the path to mount Boot.wim**](#_Install_language_pack_1)

New-Item -Path "D:\OS2022\_Custom\Boot\Boot\Mount" -ItemType directory -ea SilentlyContinue

##### [**Start mounting Boot.wim**](#_Install_language_pack_1)

Default index number: 2

$Index = "2"

Mount-WindowsImage -ImagePath "D:\OS2022\sources\boot.wim" -Index $Index -Path "D:\OS2022\_Custom\Boot\Boot\Mount"

##### **Language pack**

Automatically install language packs: Match by component. After successful matching, install the corresponding local language pack file.

###### **Language pack: add**

Copy the code or view the source file: Boot.Instl.lang.ps1 ( [Local](Code/Boot/Boot.Instl.lang.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Boot/Boot.Instl.lang.ps1) )

$Mount = "D:\OS2022\_Custom\Boot\Boot\Mount"

$Sources = "D:\OS2022\_Custom\Boot\Boot\Language\Add\zh-CN"

$Initl\_install\_Language\_Component = @()

Get-WindowsPackage -Path $Mount | ForEach-Object {

$Initl\_install\_Language\_Component += $\_.PackageName

}

Add-WindowsPackage -Path $Mount -PackagePath "$($Sources)\WinPE-FontSupport-zh-CN.cab"

$Language = @(

@{ Match = "\*WinPE\*Setup\*Server\*Package\*"; File = "WINPE-SETUP-Server\_zh-CN.CAB"; }

@{ Match = "\*WinPE\*Setup\*Package\*"; File = "WinPE-Setup\_zh-CN.cab"; }

@{ Match = "\*WinPE-LanguagePack-Package\*"; File = "lp.cab"; }

@{ Match = "\*SecureStartup\*"; File = "winpe-securestartup\_zh-CN.cab"; }

@{ Match = "\*ATBroker\*"; File = "winpe-atbroker\_zh-CN.cab"; }

@{ Match = "\*AudioCore\*"; File = "winpe-audiocore\_zh-CN.cab"; }

@{ Match = "\*AudioDrivers\*"; File = "winpe-audiodrivers\_zh-CN.cab"; }

@{ Match = "\*EnhancedStorage\*"; File = "winpe-enhancedstorage\_zh-CN.cab"; }

@{ Match = "\*Narrator\*"; File = "winpe-narrator\_zh-CN.cab"; }

@{ Match = "\*scripting\*"; File = "winpe-scripting\_zh-CN.cab"; }

@{ Match = "\*Speech-TTS\*"; File = "winpe-speech-tts\_zh-CN.cab"; }

@{ Match = "\*srh\*"; File = "winpe-srh\_zh-CN.cab"; }

@{ Match = "\*srt\*"; File = "winpe-srt\_zh-CN.cab"; }

@{ Match = "\*wds-tools\*"; File = "winpe-wds-tools\_zh-CN.cab"; }

@{ Match = "\*-WMI-Package\*"; File = "winpe-wmi\_zh-CN.cab"; }

)

ForEach ($Rule in $Language) {

Write-host "`n Rule name: $($Rule.Match)" -ForegroundColor Yellow; Write-host " $('-' \* 80)"

ForEach ($Component in $Initl\_install\_Language\_Component) {

if ($Component -like "\*$($Rule.Match)\*") {

Write-host " Component name: " -NoNewline

Write-host $Component -ForegroundColor Green

Write-host " Language pack file: " -NoNewline

Write-host "$($Sources)\$($Rule.File)" -ForegroundColor Green

Write-Host " Installing ".PadRight(22) -NoNewline

try {

Add-WindowsPackage -Path $Mount -PackagePath "$($Sources)\$($Rule.File)" | Out-Null

Write-host "Finish" -ForegroundColor Green

} catch {

Write-host "Failed" -ForegroundColor Red

}

break

}

}

}

###### **Offline image language: change**

* + 1. **Change default language, regional settings, and other international settings**

Dism /Image:"D:\OS2022\_Custom\Boot\Boot\Mount" /Set-AllIntl:zh-CN

* + 1. **View available language settings**

Dism /Image:"D:\OS2022\_Custom\Boot\Boot\Mount" /Get-Intl

###### [**Language packs: Removed, optional**](#_执行安装命令后安装语言包到：WinRE)

* After you add languages, if you want to deploy to a non-English locale, you can save space by removing the English language component. When you remove a language, uninstall the language components in the reverse order in which you added them.
* After adding the Chinese, delete "English - United States" in reverse, the language tag: en-US, you need to extract the language pack in advance

Copy the code or view the source files: Boot.Del.Specified.lang.Tag.ps1 ( [Local](Code/Boot/Boot.Del.Specified.lang.Tag.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/Boot/Boot/Boot.Del.Specified.lang.Tag.ps1) )

$Lang = "en-US"

$Mount = "D:\OS2022\_Custom\Boot\Boot\Mount"

$Sources = "D:\OS2022\_Custom\Boot\Boot\Language\Del\en-US"

$Initl\_install\_Language\_Component = @()

Get-WindowsPackage -Path $Mount | ForEach-Object {

$Initl\_install\_Language\_Component += $\_.PackageName

}

$Language = @(

@{ Match = "\*WordPad\*wow64\*"; File = "Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*WordPad\*amd64\*"; File = "Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~$($Lang)~.cab"; }

@{ Match = "\*WMIC\*FoD\*Package\*wow64\*"; File = "Microsoft-Windows-WMIC-FoD-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*WMIC\*FoD\*Package\*amd64\*"; File = "Microsoft-Windows-WMIC-FoD-Package~31bf3856ad364e35~AMD64~$($Lang)~.cab"; }

@{ Match = "\*StepsRecorder\*wow64\*"; File = "Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*StepsRecorder\*amd64\*"; File = "Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~$($Lang)~.cab"; }

@{ Match = "\*Printing\*PMCPPC\*amd64\*"; File = "Microsoft-Windows-Printing-PMCPPC-FoD-Package~31bf3856ad364e35~AMD64~$($Lang)~.cab"; }

@{ Match = "\*PowerShell\*wow64\*"; File = "Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*PowerShell\*amd64\*"; File = "Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~$($Lang)~.cab"; }

@{ Match = "\*MediaPlayer\*wow64\*"; File = "Microsoft-Windows-MediaPlayer-Package-wow64-$($Lang).cab"; }

@{ Match = "\*MediaPlayer\*amd64\*"; File = "Microsoft-Windows-MediaPlayer-Package-AMD64-$($Lang).cab"; }

@{ Match = "\*Notepad\*wow64\*"; File = "Microsoft-Windows-Notepad-System-FoD-Package~31bf3856ad364e35~wow64~$($Lang)~.cab"; }

@{ Match = "\*Notepad\*amd64\*"; File = "Microsoft-Windows-Notepad-System-FoD-Package~31bf3856ad364e35~AMD64~$($Lang)~.cab"; }

@{ Match = "\*InternetExplorer\*"; File = "Microsoft-Windows-InternetExplorer-Optional-Package~31bf3856ad364e35~AMD64~$($Lang)~.cab"; }

@{ Match = "\*TextToSpeech\*"; File = "Microsoft-Windows-LanguageFeatures-TextToSpeech-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*LanguageFeatures-Speech\*"; File = "Microsoft-Windows-LanguageFeatures-Speech-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*OCR\*"; File = "Microsoft-Windows-LanguageFeatures-OCR-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*Handwriting\*"; File = "Microsoft-Windows-LanguageFeatures-Handwriting-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*LanguageFeatures-Basic\*"; File = "Microsoft-Windows-LanguageFeatures-Basic-$($Lang)-Package~31bf3856ad364e35~amd64~~.cab"; }

@{ Match = "\*Client-LanguagePack-Package\*"; File = "Microsoft-Windows-Client-Language-Pack\_x64\_$($Lang).cab"; }

@{ Match = "\*Fonts-Hans\*"; File = "Microsoft-Windows-LanguageFeatures-Fonts-Hans-Package~31bf3856ad364e35~amd64~~.cab"; }

)

ForEach ($Rule in $Language) {

Write-host "`n Rule name: $($Rule.Match)" -ForegroundColor Yellow; Write-host " $('-' \* 80)"

ForEach ($Component in $Initl\_install\_Language\_Component) {

if ($Component -like "\*$($Rule.Match)\*$($Lang)\*") {

Write-host " Component name: " -NoNewline

Write-host $Component -ForegroundColor Green

Write-host " Language pack file: " -NoNewline

Write-host "$($Sources)\$($Rule.File)" -ForegroundColor Green

Write-Host " Deleting ".PadRight(22) -NoNewline

try {

Remove-WindowsPackage -Path $Mount -PackagePath "$($Sources)\$($Rule.File)" -ErrorAction SilentlyContinue | Out-Null

Write-host "Finish" -ForegroundColor Green

} catch {

Write-host "Failed" -ForegroundColor Red

Write-host " $($\_)" -ForegroundColor Red

}

break

}

}

}

$InitlClearLanguagePackage = @()

Get-WindowsPackage -Path $Mount | ForEach-Object {

if ($\_.PackageName -like "\*$($Lang)\*") {

$InitlClearLanguagePackage += $\_.PackageName

}

}

if ($InitlClearLanguagePackage.count -gt 0) {

ForEach ($item in $InitlClearLanguagePackage) {

Write-Host "`n $($item)" -ForegroundColor Green

Write-Host " Deleting ".PadRight(22) -NoNewline

try {

Remove-AppxProvisionedPackage -Path $Mount -PackageName $item -ErrorAction SilentlyContinue | Out-Null

Write-host "Finish" -ForegroundColor Green

} catch {

Write-host "Failed" -ForegroundColor Red

Write-host " $($\_)" -ForegroundColor Red

}

}

}

###### **Components**

* + 1. **View**

Get-WindowsPackage -Path "D:\OS2022\_Custom\Boot\Boot\Mount" | Out-GridView

* + 1. **Export to Csv**

$SaveTo = "D:\OS2022\_Custom\Boot\Boot\Report.$(Get-Date -Format "yyyyMMddHHmmss").csv"

Get-WindowsPackage -Path "D:\OS2022\_Custom\Boot\Boot\Mount" | Export-CSV -NoType -Path $SaveTo

Write-host $SaveTo -ForegroundColor Green

###### [**Language packs: sync to ISO installer**](#_Install_language_pack_1)

Copy-Item -Path "D:\OS2022\_Custom\Boot\Boot\Mount\sources\zh-CN" -Destination "D:\OS2022\sources\zh-CN" -Recurse -Force

###### **[Regenerate Lang.ini](#_Install_language_pack_1)**

After regeneration, you can adjust the "Installation Interface", the order when selecting "Language", open lang.ini, the default preferred value = 3, non-default value = 2.

* + 1. [**Regenerate the mounted directory lang.ini**](#_Regenerate_Lang.ini)

Re-generated Lang.ini file location: D:\OS2022\_Custom\Boot\Boot\Mount\Sources\lang.ini

Dism /image:"D:\OS2022\_Custom\Boot\Boot\Mount" /gen-langini /distribution:"D:\OS2022\_Custom\Boot\Boot\Mount"

* + 1. [**After regenerating lang.ini, synchronize to the installer**](#_Regenerate_Lang.ini)

Re-generated Lang.ini file location: D:\OS2022\Sources\lang.ini

Dism /image:"D:\OS2022\_Custom\Boot\Boot\Mount" /gen-langini /distribution:"D:\OS2022"

##### [**Update, optional**](#_安装语言包到：boot.wim)

To prepare the update file available, change the example file name: KB\_Boot.cab

* 1. **Add**

$KBPath = "D:\OS2022\_Custom\Boot\Boot\Update\KB\_Boot.cab"

Add-WindowsPackage -Path "D:\OS2022\_Custom\Boot\Boot\Mount" -PackagePath $KBPath

* 1. **Delete**

$KBPath = "D:\OS2022\_Custom\Boot\Boot\Update\KB\_Boot.cab"

Remove-WindowsPackage -Path "D:\OS2022\_Custom\Boot\Boot\Mount" -PackagePath $KBPath

* 1. **Solid update, optional**

It cannot be uninstalled after curing, which cleans the recovery image and resets the basis of any superseded components.

Dism /image:"D:\OS2022\_Custom\Boot\Boot\Mount" /cleanup-image /StartComponentCleanup /ResetBase

* + 1. [**Clean components after curing and updating, optional**](#_固化更新，可选项)

$Mount = "D:\OS2022\_Custom\Boot\Boot\Mount"

Get-WindowsPackage -Path $Mount -ErrorAction SilentlyContinue | ForEach-Object {

if ($\_.PackageState -eq "Superseded") {

Write-Host " $($\_.PackageName)" -ForegroundColor Green

Remove-WindowsPackage -Path $Mount -PackageName $\_.PackageName | Out-Null

}

}

##### [**Drive, optional**](#_Install_language_pack_1)

##### [**Save image**](#_Install_language_pack_1)

Save-WindowsImage -Path "D:\OS2022\_Custom\Boot\Boot\Mount"

##### [**Unmount image**](#_Install_language_pack_1)

Close any applications that may be accessing files in the image, including File Explorer.

Dismount-WindowsImage -Path "D:\OS2022\_Custom\Boot\Boot\Mount" -Discard

### [Generate](#_Customize_the_deployment) **[ISO](#_Customize_the_deployment)**

1. After downloading oscdimg, save it to: C:\Windows\System32 directory, save it in another path, or please enter the absolute location of OScdimg.exe;
2. Use the oscdimg command line to generate an ISO file and save it to: D:\WS2022.iso

Copy the code or view the source file: ISO.ps1 ( [Local](Code/ISO.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.2022/Code/ISO.ps1) )

$ISO = "D:\Win2022"

$Volume = "Win2022"

$SaveTo = "D:\Win2022.iso"

$Arguments = @("-m", "-o", "-u2", "-udfver102", "-l""$($Volume)""", "-bootdata:2#p0,e,b""$($ISO)\boot\etfsboot.com""#pEF,e,b""$($ISO)\efi\microsoft\boot\efisys.bin""", $ISO, $SaveTo)

Start-Process -FilePath "OSCdimg.exe" -ArgumentList $Arguments -wait -nonewwindow

## **COMMON PROBLEM**

1. Clean all mounts to

Close any applications that may be accessing files in the image, including File Explorer.

Dismount-WindowsImage -Path "D:\OS2022\_Custom\Install\Install\Mount" -Discard

Dismount-WindowsImage -Path "D:\OS2022\_Custom\Install\WinRE\Mount" -Discard

Dismount-WindowsImage -Path "D:\OS2022\_Custom\Boot\Boot\Mount" -Discard

1. **Fix the problem of abnormal mounting**
2. **View mounted**

Get-WindowsImage -Mounted

1. **Delete the DISM mount record saved in the registry**

Remove-Item -Path "HKLM:\SOFTWARE\Microsoft\WIMMount\Mounted Images\\*" -Force -Recurse -ErrorAction SilentlyContinue | Out-Null

1. **Delete all resources associated with the corrupted mounted image**

Clear-WindowsCorruptMountPoint

1. Clean custom directories

Remove-Item "D:\OS2022\_Custom" -Force -Recurse -ErrorAction SilentlyContinue

## **[Known issues](#_top)**

1. Add Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~zh-CN~.cab to Windows Server 2022 Standard Core, Windows Server 2022 Datacenter Core will add Microsoft-Windows-PowerShell-ISE-FOD -Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1, an error will be reported when deleting it, and the operation is not recommended for the time being.

## **[component: unchanged](#_top)**

System installation package: en-us\_windows\_server\_2022\_x64\_dvd\_620d7eac.iso, different versions, component list:

| Sn | Type | Windows Server 2022 Standard (Desktop Experience), Windows Server 2022 Datacenter (Desktop Experience) | Windows Server 2022 Standard Core, Windows Server 2022 Datacenter Core |
| --- | --- | --- | --- |
|  |  | Downlevel-NLS-Sorting-Versions-Server-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Downlevel-NLS-Sorting-Versions-Server-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Downlevel-NLS-Sorting-Versions-Server-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1 | Downlevel-NLS-Sorting-Versions-Server-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1 |
|  |  | Microsoft-OneCore-DirectX-Database-FOD-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-OneCore-DirectX-Database-FOD-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-OneCore-RasSstp-Api-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-OneCore-RasSstp-Api-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-Windows-FodMetadata-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-FodMetadata-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-Windows-Foundation-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-Foundation-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-Windows-InternetExplorer-Optional-Package~31bf3856ad364e35~amd64~~11.0.20348.1 |  |
|  |  | Microsoft-Windows-InternetExplorer-Optional-Package~31bf3856ad364e35~amd64~~11.0.20348.75 |  |
|  |  | Microsoft-Windows-LanguageFeatures-Basic-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-LanguageFeatures-Basic-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-Windows-LanguageFeatures-Handwriting-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  |  | Microsoft-Windows-LanguageFeatures-OCR-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  |  | Microsoft-Windows-LanguageFeatures-Speech-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-LanguageFeatures-Speech-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-Windows-LanguageFeatures-TextToSpeech-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-LanguageFeatures-TextToSpeech-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-Windows-MediaPlayer-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  |  | Microsoft-Windows-MediaPlayer-Package~31bf3856ad364e35~amd64~~10.0.20348.169 |  |
|  |  | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 |
|  |  | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | |
|  |  | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1 | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1 |
|  |  | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1 | |
|  |  | Microsoft-Windows-Networking-RemoteAccess-PowerShell-Base-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 | Microsoft-Windows-Networking-RemoteAccess-PowerShell-Base-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 |
|  |  | Microsoft-Windows-Networking-RemoteAccess-PowerShell-Base-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-Networking-RemoteAccess-PowerShell-Base-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-Windows-Networking-RemoteAccess-PowerShell-Base-Package~31bf3856ad364e35~amd64~~10.0.20348.143 | Microsoft-Windows-Networking-RemoteAccess-PowerShell-Base-Package~31bf3856ad364e35~amd64~~10.0.20348.143 |
|  |  | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | |
|  |  | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 |
|  |  | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1 | |
|  |  | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1 | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1 |
|  |  | Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 |  |
|  |  | Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  |  | Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1 |  |
|  |  | Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~~10.0.20348.1 |  |
|  |  | Microsoft-Windows-Server-LanguagePack-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 | Microsoft-Windows-Server-LanguagePack-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 |
|  |  | Microsoft-Windows-Server-LanguagePack-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.169 | Microsoft-Windows-Server-LanguagePack-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.169 |
|  |  | Microsoft-Windows-ServerCore-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-ServerCore-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 |  |
|  |  | Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  |  | Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1 |  |
|  |  | Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~~10.0.20348.1 |  |
|  |  | Microsoft-Windows-TabletPCMath-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  |  | Microsoft-Windows-TabletPCMath-Package~31bf3856ad364e35~amd64~~10.0.20348.143 |  |
|  |  | Microsoft-Windows-UserExperience-Desktop-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  |  | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1 |
|  |  | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1 | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1 |
|  |  | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1 | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1 |
|  |  | Microsoft-Windows-Xps-Xps-Viewer-Opt-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  |  | OpenSSH-Client-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | OpenSSH-Client-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  |  | Package\_for\_DotNetRollup~31bf3856ad364e35~amd64~~10.0.4400. | Package\_for\_DotNetRollup~31bf3856ad364e35~amd64~~10.0.4400.1 |
|  |  | Package\_for\_RollupFix~31bf3856ad364e35~amd64~~20348.169.1.7 | Package\_for\_RollupFix~31bf3856ad364e35~amd64~~20348.169.1.7 |
|  |  | Package\_for\_ServicingStack~31bf3856ad364e35~amd64~~20348.169.1.1 | Package\_for\_ServicingStack~31bf3856ad364e35~amd64~~20348.169.1.1 |

## **[component: changes after addition](#_top)**

| Sn | Type | Windows Server 2022 Standard (Desktop Experience), Windows Server 2022 Datacenter (Desktop Experience) | Windows Server 2022 Standard Core, Windows Server 2022 Datacenter Core |
| --- | --- | --- | --- |
|  | Fonts | Microsoft-Windows-LanguageFeatures-Fonts-Hans-Package~31bf3856ad364e35~AMD64~~.cab | |
| Microsoft-Windows-LanguageFeatures-Fonts-Hans-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-LanguageFeatures-Fonts-Hans-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  | Language pack | Microsoft-Windows-Server-Language-Pack\_x64\_zh-CN.cab | |
| Microsoft-Windows-Server-LanguagePack-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 | Microsoft-Windows-Server-LanguagePack-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 |
|  | Basic | Microsoft-Windows-LanguageFeatures-Basic-zh-CN-Package~31bf3856ad364e35~AMD64~~.cab | |
| Microsoft-Windows-LanguageFeatures-Basic-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-LanguageFeatures-Basic-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  | Handwriting | Microsoft-Windows-LanguageFeatures-Handwriting-zh-CN-Package~31bf3856ad364e35~AMD64~~.cab | |
| Microsoft-Windows-LanguageFeatures-Handwriting-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  | OCR | Microsoft-Windows-LanguageFeatures-OCR-zh-CN-Package~31bf3856ad364e35~AMD64~~.cab | |
| Microsoft-Windows-LanguageFeatures-OCR-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |  |
|  | Speech | Microsoft-Windows-LanguageFeatures-Speech-zh-CN-Package~31bf3856ad364e35~AMD64~~.cab | |
| Microsoft-Windows-LanguageFeatures-Speech-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-LanguageFeatures-Speech-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  | Text to speech | Microsoft-Windows-LanguageFeatures-TextToSpeech-zh-CN-Package~31bf3856ad364e35~AMD64~~.cab | |
| Microsoft-Windows-LanguageFeatures-TextToSpeech-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1 | Microsoft-Windows-LanguageFeatures-TextToSpeech-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1 |
|  | Feature pack | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~zh-CN~.cab | |
| Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 |
|  | Feature pack | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~wow64~zh-CN~.cab | |
| Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 | Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 |
|  | Feature pack | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~amd64~zh-CN~.cab | |
| Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 |
|  | Feature pack | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~wow64~zh-CN~.cab | |
| Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 | Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 |
|  | Feature pack | Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~zh-CN~.cab | |
| Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 |  |
|  | Feature pack | Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~zh-CN~.cab | |
| Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 |  |
|  | On demand package | Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~zh-CN~.cab | |
| Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 |  |
|  | On demand package | Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~zh-CN~.cab | |
| Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 |  |
|  | Feature pack | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~zh-CN~.cab | |
| Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 |
|  | Feature pack | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~zh-CN~.cab | |
| Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 | Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 |