Windows Server 2022

不同的系统版本，有着不同的封装方式，封装过程中包含：“语言包：添加、关联、删除”、“驱动：添加、删除”、“累积更新：添加、删除”、“InBox Appx：添加、更新、标记”等。

在这背后藏着很多的隐藏故事，想解开这些，你准备好开始尝试封装了吗？

1. 目录
2. [部署映像](#_自定义部署映像)
   * 1. [先决条件](#_先决条件)
     2. [语言包：提取](#_语言包：提取)
     3. [自定义部署映像](#_语言包：安装)
     4. [ISO](#_生成_ISO)
3. [常见问题](#_常见问题)
4. [已知问题](#_已知问题)
5. [组件：映像中已安装的所有包，未变更](#_组件：未变更)
6. [组件：映像中已安装的所有包，新增后变化](#_组件：新增后变化)

## **[部署映像](#_top)**

### **[先决条件](#_如何添加语言包到映像里)**

#### [ISO 工具](#_PREREQUISITES)

使用一款可编辑 ISO 文件的软件，例如：PowerISO、DAEMON Tools、ISO Workshop；

#### [**获取相关包**](#_PREREQUISITES)

学习“附件1：获取 Windows 安装包”，查看“Windows Server 2022”项，

1. [**系统安装包**](#_PREREQUISITES)
   1. 准备：en-us\_windows\_server\_2022\_x64\_dvd\_620d7eac.iso
   2. 解压到：D:\en-us\_windows\_server\_2022\_x64\_dvd\_620d7eac
   3. 解压完成后，将目录en-us\_windows\_11\_business\_editions\_version\_22h2\_x64\_dvd\_17a08c 更改为 D:\OS2022
   4. 所有脚本、所有路径，已默认设置为 D:\OS2022 为映像来源。

##### [**语言包**](#_PREREQUISITES)

###### **学习**

* + 1. [将语言添加到 Windows 11 映像](https://learn.microsoft.com/en-us/azure/virtual-desktop/windows-11-language-packs)
    2. [语言和区域按需功能 (FOD)](https://learn.microsoft.com/zh-cn/windows-hardware/manufacture/desktop/features-on-demand-language-fod?view=windows-11)
       1. **语言标记关联**

语言标记关联是什么？

* 映像语言仅英文版时，添加zh-HK 语言包后，映像语言不会新增语言标记，应先安装 zh-TW 后，再安装zh-HK 即可获得对应的关联。
* 可参阅微软官方原版：Windows 10、Windows 11 繁体版。

已知语言标记关联：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 语言标记 | 可选语言关联标记 | 备 注 |
|  | zh-TW | Zh-HK |  |

* + - 1. **字体**

添加语言包时，触发了对应的语言标记时，需添加所需的字体功能，下载“[所有可用语言 FOD 的列表](https://download.microsoft.com/download/7/6/0/7600F9DC-C296-4CF8-B92A-2D85BAFBD5D2/Windows-10-1809-FOD-to-LP-Mapping-Table.xlsx)”了解更多。

在“[语言包：提取](#_2、提取所需语言包)”时，已加入自动识别功能，可了解函数：Function Match\_Required\_Fonts

###### **语言包：下载**

[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. 运行 PS 命令行时：
2. 可选“Terminal ”或“PowerShell ISE”，未安装“Terminal”，请前往 <https://github.com/microsoft/terminal/releases> 后下载；
3. 以管理员身份打开“Terminal”或“PowerShell ISE”，建议设置 PowerShell 执行策略：绕过，PS 命令行：

Set-ExecutionPolicy -ExecutionPolicy Bypass -Force

1. 在本文中，PS 命令行，绿色部分，请复制后，粘贴到“Terminal”对话框，按回车键（Enter）后开始运行；
2. 有 .ps1 时，点击文件右键，选择以 PowerShell 运行，或复制路径，粘贴到 Terminal 里运行。

### [**语言包：提取**](#_2、提取所需语言包)

1. [**准备语言包**](#_2、提取所需语言包)

挂载 20348.1.210507-1500.fe\_release\_amd64fre\_SERVER\_LOF\_PACKAGES\_OEM.iso 或解压到任意位置；

1. 提取语言包方案

##### 添加

* 1. 语言名称：简体中文 - 中国，语言标记：zh-CN，适用范围：Install.Wim，Boot.Wim，WinRE.Wim

##### 删除

* 1. 语言名称：英语 - 美国，语言标记：en-US，适用范围：Install.Wim，Boot.Wim，WinRE.Wim

1. [**执行提取命令**](#_2、提取所需语言包)

* Auto = 自动搜索本地所有磁盘，默认；
* 自定义路径，例如指定为 E 盘：$ISO = "E:\"

复制代码或查看源文件：Extract.ps1 ( [本地](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 Match\_Required\_Fonts

{

param( $Lang )

$Fonts = @(

@{ Match = @("as", "ar-SA", "ar", "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-SD", "ar-SO", "ar-SS", "ar-SY", "ar-TD", "ar-TN", "ar-YE", "arz-Arab", "ckb-Arab", "fa", "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", "prs-AF", "prs-Arab", "ps", "ps-AF", "sd-Arab", "sd-Arab-PK", "tk-Arab", "ug", "ug-Arab", "ug-CN", "ur", "ur-IN", "ur-PK", "uz-Arab", "uz-Arab-AF"); Name = "Arab"; }

@{ Match = @("bn-IN", "as-IN", "bn", "bn-BD", "bpy-Beng"); Name = "Beng"; }

@{ Match = @("da-dk", "iu-Cans", "iu-Cans-CA"); Name = "Cans"; }

@{ Match = @("chr-Cher-US", "chr-Cher"); Name = "Cher"; }

@{ Match = @("hi-IN", "bh-Deva", "brx", "brx-Deva", "brx-IN", "hi", "ks-Deva", "mai", "mr", "mr-IN", "ne", "ne-IN", "ne-NP", "new-Deva", "pi-Deva", "sa", "sa-Deva", "sa-IN"); Name = "Deva"; }

@{ Match = @("am", "am-ET", "byn", "byn-ER", "byn-Ethi", "ti", "ti-ER", "ti-ET", "tig", "tig-ER", "tig-Ethi", "ve-Ethi", "wal", "wal-ET", "wal-Ethi"); Name = "Ethi"; }

@{ Match = @("gu", "gu-IN"); Name = "Gujr"; }

@{ Match = @("pa", "pa-IN", "pa-Guru"); Name = "Guru"; }

@{ Match = @("zh-CN", "cmn-Hans", "gan-Hans", "hak-Hans", "wuu-Hans", "yue-Hans", "zh-gan-Hans", "zh-hak-Hans", "zh-Hans", "zh-SG", "zh-wuu-Hans", "zh-yue-Hans"); Name = "Hans"; }

@{ Match = @("zh-TW", "cmn-Hant", "hak-Hant", "lzh-Hant", "zh-hak-Hant", "zh-Hant", "zh-HK", "zh-lzh-Hant", "zh-MO", "zh-yue-Hant"); Name = "Hant"; }

@{ Match = @("he", "he-IL", "yi"); Name = "Hebr"; }

@{ Match = @("ja", "ja-JP"); Name = "Jpan"; }

@{ Match = @("km", "km-KH"); Name = "Khmr"; }

@{ Match = @("kn", "kn-IN"); Name = "Knda"; }

@{ Match = @("ko", "ko-KR"); Name = "Kore"; }

@{ Match = @("de-de", "lo", "lo-LA"); Name = "Laoo"; }

@{ Match = @("ml", "ml-IN"); Name = "Mlym"; }

@{ Match = @("or", "or-IN"); Name = "Orya"; }

@{ Match = @("si", "si-LK"); Name = "Sinh"; }

@{ Match = @("tr-tr", "arc-Syrc", "syr", "syr-SY", "syr-Syrc"); Name = "Syrc"; }

@{ Match = @("ta", "ta-IN", "ta-LK", "ta-MY", "ta-SG"); Name = "Taml"; }

@{ Match = @("te", "te-IN"); Name = "Telu"; }

@{ Match = @("th", "th-TH"); Name = "Thai"; }

)

ForEach ($item in $Fonts) {

if (($item.Match) -Contains $Lang) {

return $item.Name

}

}

return "Not\_matched"

}

Function Match\_Other\_Region\_Specific\_Requirements

{

param( $Lang )

$Fonts = @(

@{ Match = @("zh-TW"); Name = "Taiwan"; }

)

ForEach ($item in $Fonts) {

if (($item.Match) -Contains $Lang) {

return $item.Name

}

}

return "Skip\_specific\_packages"

}

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"

)

}

)

$NewFonts = Match\_Required\_Fonts -Lang $NewLang

$SpecificPackage = Match\_Other\_Region\_Specific\_Requirements -Lang $NewLang

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}", $NewFonts).Replace("{Specific}", $SpecificPackage)

}

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

}

### **[自定义部署映像](#_如何添加语言包到映像里)**

#### **[自定义部署映像：Install.wim](#_INSTALL_LANGUAGE_PACK)**

##### [**查看 Install.wim 详细信息**](#_3.1__)

映像名称、映像描述、映像大小、架构、版本、索引号等；

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

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

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

}

循环操作区域，开始，

##### [**指定挂载 Install.wim 路径**](#_3.1__)

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

1. **开始挂载 Install.wim**

默认索引号：1

Mount-WindowsImage -ImagePath "D:\OS2022\sources\install.wim" -Index "1" -Path "D:\OS2022\_Custom\Install\Install\Mount"

处理 Install.wim 映像内的文件，可选项，开始，

###### **[自定义部署映像：WinRE.wim](#_安装语言包到：Install.wim)**

注意：

* WinRE.wim 属于 Install.wim 映像内的文件；
* Install.wim 有多个索引号时，仅处理任意一个 WinRE.wim 即可；
* 同步至所有索引号即可减少 Install.wim 体积；学习“[如何批量替换 Install.wim 里的所有索引号里的 WinRE.wim](#_如何批量替换_Install.wim_里的所有索引号里的)”。
  + 1. [**查看 WinRE.wim 详细信息**](#_3.1.2.1__)

映像名称、映像描述、映像大小、架构、版本、索引号等；

$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. [**指定挂载 WinRE.wim 路径**](#_3.1.2.1__)

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

* + 1. [**开始挂载 WinRE.wim**](#_3.1.2.1__)

默认索引号：1

Mount-WindowsImage -ImagePath $FileName -Index "1" -Path "D:\OS2022\_Custom\Install\WinRE\Mount"

* + 1. **[语言包](#_自定义部署映像：WinRE.wim)**

自动安装语言包：获取“组件：映像中已安装的所有包”后进行匹配，匹配到对应的名称后，再安装本地对应的语言包文件。

* + - 1. [**语言包：添加**](#_语言包)

复制代码或查看源文件：WinRE.Instl.lang.ps1 ( [本地](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. **脱机映像语言：更改**
         1. **更改默认语言、区域设置和其他国际设置**

语言标记：zh-CN

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

* + - * 1. **查看可用的语言设置**

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

* + - 1. [**语言包：删除**](#_执行安装命令后安装语言包到：WinRE)**，可选**
* 添加语言后，如果要部署到非英语区域，可通过删除英语语言组件来节省空间。 删除语言时，请按与添加语言组件相反的顺序卸载语言组件。
* 添加中文后，反向删除“英语 - 美国”，语言标记：en-US，需提前提取语言包

复制代码或查看源文件：WinRE.Del.Specified.lang.Tag.ps1 ( [本地](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. **组件：**[**映像中已安装的所有包**](#_语言包)
         1. [**查看**](#_组件)

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

* + - * 1. [**导出到 Csv**](#_组件)

$SaveTo = "D:\OS2022\_Custom\Install\WinRE\Report.$(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. [**累积更新，可选**](#_3.1.2.1__)

准备可用的累积更新文件，请更改示例文件名：KB\_WinRE.cab

* + - 1. **添加**

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

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

* + - 1. **删除**

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

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

* + - 1. **固化更新，可选项**

固化后不可卸载，固化时将清理恢复映像并重置任何被取代的组件的基础。

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

* + - * 1. [**固化更新后清理组件，可选项**](#_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. [**驱动，可选**](#_3.1.2.1__)
    2. [**保存映像**](#_3.1.2.1__)

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

* + 1. [**卸载映像**](#_3.1.2.1__)

关闭所有可能正在访问映像中文件的应用程序，包括文件资源管理器。

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

* + 1. [**重建 WinRE.wim 后，可缩小文件大小**](#_3.1.2.1__)

复制代码或查看源文件：WinRE.Rebuild.ps1 ( [本地](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 Rebuild".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. [**备份 Install.wim**](#_3.1.2.1__)

复制代码或查看源文件：WinRE.Backup.ps1 ( [本地](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. [**替换 Install.wim 映像内的 WinRE.wim**](#_安装语言包到：WinRE.wim)
* 每次挂载 Install.wim 后“[替换 WinRE.wim](#_替换_WinRE.wim)”；
* 学习“[获取 Install.wim 所有索引号后并替换旧的 WinRE.wim](#_获取_Install.wim_所有索引号后并替换旧的)”。

处理 Install.wim 映像内的文件，结束。

##### **[语言包](#_自定义部署映像：Install.wim)**

自动安装语言包：获取“组件：映像中已安装的所有包”后进行匹配，匹配到对应的名称后，再安装本地对应的语言包文件。

###### [**语言包：添加**](#_语言包_1)

复制代码或查看源文件：Install.Instl.lang.ps1 ( [本地](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"; }

@{ Match = "\*Client\*LanguagePack\*zh-TW\*"; File = "Microsoft-Windows-InternationalFeatures-Taiwan-Package~31bf3856ad364e35~amd64~~.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

}

}

}

}

###### **脱机映像语言：更改**

* 从 Windows 11 开始，DISM 设置的[默认系统 UI 语言](https://learn.microsoft.com/zh-cn/windows/win32/intl/user-interface-language-management)在所有版本中保持不变（家庭版除外）。 对于[所有商业版](https://www.microsoft.com/windowsforbusiness/compare)，在开箱即用体验 (OOBE) 期间选择的语言会设置为[系统首选 UI 语言](https://learn.microsoft.com/zh-cn/windows/win32/intl/user-interface-language-management)，Windows 将以此语言显示；对于家庭版，在 OOBE 期间选择的语言将继续用作默认系统 UI 语言。
* 从 Windows 10 版本 2004 开始，如果将基于 .appx 的语言体验包 (LXP) 支持的语言作为参数传递，则该语言将设置为系统首选 UI 语言，其父语言将设置为默认系统 UI 语言。 在以前的版本中，仅支持基于 .cab 的语言包。
  + 1. **更改默认语言、区域设置和其他国际设置**

语言标记：zh-CN

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

* + 1. **查看可用的语言设置**

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

###### [**语言包：删除**](#_执行安装命令后安装语言包到：WinRE)**，可选**

* 添加语言后，如果要部署到非英语区域，可通过删除英语语言组件来节省空间。 删除语言时，请按与添加语言组件相反的顺序卸载语言组件。
* 添加中文后，反向删除“英语 - 美国”，语言标记：en-US，需提前提取语言包

复制代码或查看源文件：Install.Del.Specified.lang.Tag.ps1 ( [本地](Code/Install/WinRE/WinRE.Del.Specified.lang.Tag.ps1), [Github](https://github.com/ilikeyi/solutions/blob/main/_Documents/Attachment/OS.11/22H2/Code/Install/Install/Install.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

}

}

}

###### **组件：**[**映像中已安装的所有包**](#_语言包_1)

* + 1. [**查看**](#_组件_1)

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

* + 1. [**导出到 Csv**](#_组件_1)

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

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

Write-host $SaveTo -ForegroundColor Green

##### **[累积更新，可选项](#_3.1__)**

###### [**下载**](#_更新，可选项)

查阅“[Windows Server 2022 更新历史](https://support.microsoft.com/en-gb/topic/windows-server-2022-update-history-e1caa597-00c5-4ab9-9f3e-8212fe80b2ee)”，例如安装累积更新：KB5030216

前往下载页面：<https://www.catalog.update.microsoft.com/Search.aspx?q=Kb5030216> 或 "[直连下载](https://catalog.s.download.windowsupdate.com/c/msdownload/update/software/secu/2023/09/windows10.0-kb5030216-x64_cbe587155f9818548b75f65d5cd41d341ed2fc61.msu)" （无法下载时请进入下载页面），保存到：

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

###### [**添加**](#_更新，可选项)

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

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

###### **[固化更新，可选项](#_更新，可选项)**

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

* + 1. [**固化更新后清理组件**](#_固化更新，可选项)

复制代码或查看源文件：Install.Update.Curing.ps1 ( [本地](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

}

}

##### [**驱动，可选**](#_3.1__)

##### **添加部署引擎，可选**

* 了解“[部署引擎](#_添加部署引擎，可选)”，如果添加到 ISO 安装介质，可跳过添加到已挂载。
* 如果添加部署引擎到已挂载里，请继续在当前位置执行下一步。

##### [**健康**](#_3.1__)

保存前应检查是否损坏，健康状态异常时，中止保存

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

##### [**替换 WinRE.wim**](#_3.1__)

已批量替换 Install.wim 里的所有索引号里的 WinRE.wim 请跳过该步骤。

$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

##### [**保存映像**](#_3.1__)

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

##### [**卸载映像**](#_3.1__)

关闭所有可能正在访问映像中文件的应用程序，包括文件资源管理器。

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

循环操作区域，结束。

##### **[如何批量替换 Install.wim 里的所有索引号里的 WinRE.wim](#_3.1__)**

###### [**获取WimLib**](#_如何批量替换_Install.wim_里的所有索引号里的)

前往<https://wimlib.net> 官方网站后，选择不同的版本：[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)，下载完成后解压到：D:\Wimlib

###### **[如何在 Install.wim 里提取和更新 WinRE.wim](#_如何批量替换_Install.wim_里的所有索引号里的)**

* + 1. [从 Install.wim 里提取 WinRE.wim 文件Install.wim](#_如何在_Install.wim_里提取和更新)

复制代码或查看源文件：Install.WinRE.Extract.ps1 ( [本地](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. [**获取 Install.wim 所有索引号后并替换旧的 WinRE.wim**](#_如何在_Install.wim_里提取和更新)

复制代码或查看源文件：Install.WinRE.Replace.wim.ps1 ( [本地](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

}

##### [**重建 Install.wim 后可缩小文件大小**](#_3.1__)

复制代码或查看源文件：Install.Rebuild.wim.ps1 ( [本地](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 Rebuild".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

}

#### **[自定义部署映像：boot.wim](#_INSTALL_LANGUAGE_PACK)**

##### [**查看 Boot.wim 文件信息**](#_安装语言包到：boot.wim)

映像名称、映像描述、映像大小、架构、版本、索引号等；

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

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

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

}

##### [**指定挂载 Boot.wim 路径**](#_安装语言包到：boot.wim)

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

##### [**开始挂载 Boot.wim**](#_安装语言包到：boot.wim)

默认索引号：2

Mount-WindowsImage -ImagePath "D:\OS2022\sources\boot.wim" -Index "2" -Path "D:\OS2022\_Custom\Boot\Boot\Mount"

##### **[语言包](#_自定义部署映像：boot.wim)**

自动安装语言包：获取“组件：映像中已安装的所有包”后进行匹配，匹配到对应的名称后，再安装本地对应的语言包文件。

###### [**语言包：添加**](#_语言包_2)

复制代码或查看源文件：Boot.Instl.lang.ps1 ( [本地](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\_zh\*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

}

}

}

###### **脱机映像语言：更改**

* + 1. **更改默认语言、区域设置和其他国际设置**

语言标记：zh-CN

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

* + 1. **查看可用的语言设置**

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

###### [**语言包：删除**](#_执行安装命令后安装语言包到：WinRE)**，可选**

* 添加语言后，如果要部署到非英语区域，可通过删除英语语言组件来节省空间。 删除语言时，请按与添加语言组件相反的顺序卸载语言组件。
* 添加中文后，反向删除“英语 - 美国”，语言标记：en-US，需提前提取语言包

复制代码或查看源文件：Boot.Del.Specified.lang.Tag.ps1 ( [本地](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

}

}

}

###### [**组件：映像中已安装的所有包**](#_语言包_2)

* + 1. [**查看**](#_组件_2)

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

* + 1. [**导出到 Csv**](#_组件_2)

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

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

Write-host $SaveTo -ForegroundColor Green

###### **[语言包：同步到 ISO 安装程序](#_安装语言包到：boot.wim)**

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

###### **[重新生成 Lang.ini](#_安装语言包到：boot.wim)**

重新生成后，可调整“安装界面”，选择“语言”时的顺序，打开 lang.ini，默认首选值 = 3，非默认值 = 2。

* + 1. [**重新生成已挂载目录 lang.ini**](#_重新生成_Lang.ini)

重新生成的Lang.ini 文件位置：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. [**重新生成 lang.ini 后，同步到安装程序**](#_重新生成_Lang.ini)

重新生成的Lang.ini 文件位置：D:\OS2022\Sources\lang.ini

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

##### [**累积更新，可选项**](#_安装语言包到：boot.wim)

准备可用的累积更新文件，请更改示例文件名：KB\_Boot.cab

* 1. **添加**

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

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

* 1. **删除**

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

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

* 1. **固化更新，可选项**

固化后不可卸载，固化时将清理恢复映像并重置任何被取代的组件的基础。

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

* + 1. [**固化更新后清理组件，可选项**](#_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

}

}

##### [**驱动，可选项**](#_安装语言包到：boot.wim)

##### [**保存映像**](#_安装语言包到：boot.wim)

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

##### [**卸载映像**](#_安装语言包到：boot.wim)

关闭所有可能正在访问映像中文件的应用程序，包括文件资源管理器。

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

#### **[部署引擎](#_3、安装语言包)**

* + - * 了解“自动添加 Windows 系统已安装的语言”，学习：<https://github.com/ilikeyi/Multilingual>，如何下载：
      * 进入网站后，点击 “代码”，“下载压缩包”，下载完成后得到 main.zip 压缩包文件。
      * 前往 <https://github.com/ilikeyi/Multilingual/releases> 下载页面，选择可用版本：1.1.0.4，选择下载源代码格式：zip，下载完成后得到 Multilingual-1.1.0.4.zip 压缩包文件；
      * 将已下载的 main.zip 或 Multilingual-1.1.0.4.zip，解压到：D:\Multilingual-1.1.0.4，重命名：D:\Multilingual
      * 学习“[无人值守 Windows 安装参考](https://learn.microsoft.com/en-us/windows-hardware/customize/desktop/unattend/)”，通过无人值守来干预安装过程。

##### **添加方式**

###### **添加到 ISO 安装介质**

* + 1. **无人值守**
       1. **添加到：[ISO]:\Autounattend.xml**

引导 ISO 安装时，Autounattend.xml 干预 WinPE 安装程序。

复制 D:\Multilingual\Unattend\Mul.Unattend.xml 到 D:\OS2022\Autounattend.xml

Copy-Item "D:\Multilingual\Unattend\Mul.Unattend.xml" -Destination "D:\OS2022\Autounattend.xml" -Force

* + - 1. **添加到：[ISO]:\Sources\Unattend.xml**

挂载或解压 ISO 时，运行 [ISO]:\Setup.exe 安装程序后，[ISO]:\Sources\Unattend.xml 将干预安装过程。

复制 D:\Multilingual\Unattend\Mul.Unattend.xml 到 D:\OS2022\Sources\Unattend.xml

Copy-Item "D:\Multilingual\Unattend\Mul.Unattend.xml" -Destination "D:\OS2022\Sources\Unattend.xml" -Force

* + - 1. **添加到：[ISO]:\sources\$OEM$\$$\Panther\unattend.xml**

安装过程中复制到系统盘里，复制到：{系统盘}:\Windows\Panther\unattend.xml

* + - * 1. **创建 $OEM$ 路径**

New-Item -Path "D:\OS2022\sources\`$OEM$\`$$\Panther" -ItemType Directory

* + - * 1. **复制**

复制 D:\Multilingual\Unattend\Mul.Unattend.xml 到 D:\OS2022\Sources\$OEM$\Panther\Unattend.xml

Copy-Item "D:\Multilingual\Unattend\Mul.Unattend.xml" -Destination "D:\OS2022\sources\`$OEM$\`$$\Panther\Unattend.xml" -Force

* + 1. **部署引擎：添加**

添加“自动添加 Windows 系统已安装的语言”到 D:\OS2022\sources\$OEM$\$1\Yi\Engine 目录里。

* + - 1. **部署引擎：复制**

复制 D:\Multilingual\Engine 到 D:\OS2022\Sources\$OEM$\$1\Yi\Engine

Copy-Item "D:\Multilingual\Engine" -Destination "D:\OS2022\sources\`$OEM$\`$1\Yi\Engine" -Recurse -Force

* + - 1. **部署引擎：自定义部署标记**

$Flag = @(

"Is\_Mark\_Sync" # 允许全盘搜索并同步部署标记

# 先决部署

# "Auto\_Update" # 允许自动更新

# "Use\_UTF8" # Beta 版：使用 Unicode UTF-8 提供全球语言支持

"Disable\_Network\_Location\_Wizard" # 网络位置向导

"Disable\_Cleanup\_Appx\_Tasks" # Appx 清理维护任务

"Disable\_Cleanup\_On\_Demand\_Language" # 阻止清理未使用的按需功能语言包

"Disable\_Cleanup\_Unsed\_Language" # 阻止清理未使用的语言包

"Prerequisites\_Reboot" # 重新启动计算机

# 完成首次部署

# "Popup\_Engine" # 允许首次弹出部署引擎主界面

# "Allow\_First\_Pre\_Experience" # 允许首次预体验，按计划

"Reset\_Execution\_Policy" # 恢复 PowerShell 执行策略：受限

"Clear\_Solutions" # 删除整个解决方案

"Clear\_Engine" # 删除部署引擎，保留其它

# "First\_Experience\_Reboot" # 重新启动计算机

)

ForEach ($item in $Flag) {

Write-host " $($item)" -ForegroundColor Green

New-Item -Path "D:\OS2022\sources\`$OEM$\`$1\Yi\Engine\Deploy\Allow" -ItemType Directory -ErrorAction SilentlyContinue | Out-Null

Out-File -FilePath "D:\OS2022\sources\`$OEM$\`$1\Yi\Engine\Deploy\Allow\$($item)" -Encoding utf8 -ErrorAction SilentlyContinue

}

###### **添加到已挂载**

通过“[自定义部署映像：Install.wim](#_3.1__)”，执行“[开始挂载 Install.wim](#_开始挂载_Install.wim)”，挂载到：D:\OS2022\_Custom\Install\Install\Mount

* + 1. **无人值守**

复制 D:\Multilingual\Unattend\Mul.Unattend.xml 到 D:\OS2022\_Custom\Install\Install\Mount\Panther\Unattend.xml

Copy-Item "D:\Multilingual\Unattend\Mul.Unattend.xml" -Destination "D:\OS2022\_Custom\Install\Install\Mount\Panther" -Force

* + 1. **部署引擎**

添加“自动添加 Windows 系统已安装的语言”到 D:\OS2022\_Custom\Install\Install\Mount\Yi\Engine 目录里。

* + - 1. **部署引擎：复制**

复制 D:\Multilingual\Engine 到 D:\OS2022\_Custom\Install\Install\Mount\Yi\Engine

Copy-Item "D:\Multilingual\Engine" -Destination "D:\OS2022\_Custom\Install\Install\Mount\Yi\Engine" -Recurse -Force

* + - 1. **部署引擎：自定义部署标记**

$Flag = @(

"Is\_Mark\_Sync" # 允许全盘搜索并同步部署标记

# 先决部署

# "Auto\_Update" # 允许自动更新

# "Use\_UTF8" # Beta 版：使用 Unicode UTF-8 提供全球语言支持

"Disable\_Network\_Location\_Wizard" # 网络位置向导

"Disable\_Cleanup\_Appx\_Tasks" # Appx 清理维护任务

"Disable\_Cleanup\_On\_Demand\_Language" # 阻止清理未使用的按需功能语言包

"Disable\_Cleanup\_Unsed\_Language" # 阻止清理未使用的语言包

"Prerequisites\_Reboot" # 重新启动计算机

# 完成首次部署

# "Popup\_Engine" # 允许首次弹出部署引擎主界面

# "Allow\_First\_Pre\_Experience" # 允许首次预体验，按计划

"Reset\_Execution\_Policy" # 恢复 PowerShell 执行策略：受限

"Clear\_Solutions" # 删除整个解决方案

"Clear\_Engine" # 删除部署引擎，保留其它

# "First\_Experience\_Reboot" # 重新启动计算机

)

ForEach ($item in $Flag) {

Write-host " $($item)" -ForegroundColor Green

New-Item -Path "D:\OS2022\_Custom\Install\Install\Mount\Yi\Engine\Deploy\Allow" -ItemType Directory -ErrorAction SilentlyContinue | Out-Null

Out-File -FilePath "D:\OS2022\_Custom\Install\Install\Mount\Yi\Engine\Deploy\Allow\$($item)" -Encoding utf8 -ErrorAction SilentlyContinue

}

##### **部署引擎：进阶**

###### **部署引擎：添加过程中**

在复制部署引擎后，可添加部署标记来干预安装过程。

###### **无人值守方案**

自定义无人值守时，以下文件存在时请同步修改：

* D:\OS2022\Autounattend.xml
* D:\OS2022\Sources\Unattend.xml
* D:\OS2022\sources\$OEM$\$$\Panther\unattend.xml
* D:\OS2022\_Custom\Install\Install\Mount\Panther\Unattend.xml
  + 1. **多语言或单语**

多语言时，单语时，可互相切换，替换时，请替换文件里所有相同的。

* + - 1. **多语言**

<UILanguage>%OSDUILanguage%</UILanguage>

<InputLocale>%OSDInputLocale%</InputLocale>

<SystemLocale>%OSDSystemLocale%</SystemLocale>

<UILanguage>%OSDUILanguage%</UILanguage>

<UILanguageFallback>%OSDUILanguageFallback%</UILanguageFallback>

<UserLocale>%OSDUserLocale%</UserLocale>

* + - 1. **单语**

单语需指定语言标记，例如指定语言标记：zh-CN

<UILanguage>zh-CN</UILanguage>

<InputLocale>zh-CN</InputLocale>

<SystemLocale>zh-CN</SystemLocale>

<UILanguage>zh-CN</UILanguage>

<UILanguageFallback>zh-CN</UILanguageFallback>

<UserLocale>zh-CN</UserLocale>

* + 1. **用户方案**

默认使用自建用户Administrator 并自动登录，可通过修改以下配置切换：自建、自定义用户。

* + - 1. **自建用户 Administrator**

默认使用自建用户：Administrator 并自动登录，插入到 <OOBE> 和 </OOBE> 之间。

<UserAccounts>

<LocalAccounts>

<LocalAccount wcm:action="add">

<Password>

<Value></Value>

<PlainText>true</PlainText>

</Password>

<Description>Administrator</Description>

<DisplayName>Administrator</DisplayName>

<Group>Administrators</Group>

<Name>Administrator</Name>

</LocalAccount>

</LocalAccounts>

</UserAccounts>

<AutoLogon>

<Password>

<Value></Value>

<PlainText>true</PlainText>

</Password>

<Enabled>true</Enabled>

<Username>Administrator</Username>

</AutoLogon>

* + - 1. **自定义用户**

设置自定义用户后，安装系统完成后，在 OOBE 里，可选择本地、在线用户等设置。

* + - * 1. **删除**

用户名：从开始处删除 <UserAccounts> 到 </UserAccounts>

自动登录：从开始处删除 <AutoLogon> 到 </AutoLogon>

* + - * 1. **替换**

从开始处 <OOBE> 到 </OOBE>

<OOBE>

<ProtectYourPC>3</ProtectYourPC>

<HideEULAPage>true</HideEULAPage>

<HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>

</OOBE>

### [**生成 ISO**](#_如何添加语言包到映像里)

1. 下载 oscdimg 后，保存到：C:\Windows\System32 目录里，保存在其它路径，或请输入 OScdimg.exe 绝对位置；
2. 使用 oscdimg命令行生成一个 ISO 文件，保存到：D:\WS2022.iso

复制代码或查看源文件：ISO.ps1 ( [本地](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, $FileName)

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

## **[常见问题](file:///D:\\Yi.Solutions\\_Documents\\Attachment\\OS.11\\22H2\\Windows.11.22H2.zh-CN.docx" \l "_top)**

1. 清理所有挂载到

关闭所有可能正在访问映像中文件的应用程序，包括文件资源管理器。

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. **修复挂载出现异常的问题**
2. 查看已挂载

Get-WindowsImage -Mounted

1. 删除保存在注册表里的 DISM 挂载记录

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

1. 删除与已损坏的已装载映像关联的所有资源。

Clear-WindowsCorruptMountPoint

Dism /cleanup-wim

1. 清理目录

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

Remove-Item "D:\OS2022\Sources\`$OEM$" -Force -Recurse

Remove-Item -Path "D:\OS2022\Autounattend.xml" -Force

Remove-Item -Path "D:\OS2022\Sources\Unattend.xml" -Force

## **[已知问题](#_top)**

1. 添加 Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~zh-CN~.cab 到 Windows Server 2022 Standard Core, Windows Server 2022 Datacenter Core后会新增 Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1，执行删除会报错，暂时不建议操作。

## [**组件：映像中已安装的所有包，未变更**](#_top)

系统安装包：en-us\_windows\_server\_2022\_x64\_dvd\_620d7eac.iso，不同版本，组件列表：

| 序号 | 类型 | 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 |

## [**组件：映像中已安装的所有包，新增后变化**](#_top)

| 序号 | 类型 | 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 |  |
|  | 语音 | 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 |
|  | 文本和语音 | 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 |
|  | 功能包 | 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 |
|  | 功能包 | 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 |
|  | 功能包 | 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 |
|  | 功能包 | 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 |
|  | 功能包 | 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 |  |
|  | 功能包 | Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~zh-CN~.cab | |
| Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 |  |
|  | 按需包 | Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~zh-CN~.cab | |
| Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1 |  |
|  | 按需包 | Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~zh-CN~.cab | |
| Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1 |  |
|  | 功能包 | 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 |
|  | 功能包 | 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 |