

WINDOWS SERVER 2022

添加语言包、功能包后、累积更新及其它完成后，组件会有新的变化，例如过时的、待删除等状态。向不同的版本添加语言包、功能包后，每一个语言包、功能包有对应的。在 Windows 11、Windows 10 里并无多少变化，但是在 Windows Server 2022 Datacenter 和 Windows Server 2022 Datacenter (Desktop Experience) 版对比就有很大的变化，详见组件分析报告。

目录

- 一、部署映像
 - A. 先决条件
 - B. 语言包：提取
 - C. 自定义部署映像
 - D. 生成 ISO
- 二、常见问题
- 三、已知问题
- 四、组件：未变更
- 五、组件：新增后变化

一、部署映像

A. 先决条件

II ISO 工具

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

III 获取相关包

学习“[附件 1：获取 Windows 安装包](#)”，查看“[Windows Server 2022](#)”项，

1. 系统安装包

- 1.1. 准备：[en-us_windows_server_2022_x64_dvd_620d7eac.iso](#)
- 1.2. 解压到：[D:\en-us_windows_server_2022_x64_dvd_620d7eac](#)
- 1.3. 解压完成后，将目录 [en-us_windows_11_business_editions_version_22h2_x64_dvd_17a08c](#) 更改为 [D:\OS2022](#)
- 1.4. 所有脚本、所有路径，已默认设置为 [D:\OS2022](#) 为映像来源。

2. 语言包：

下载：[20348.1.210507-1500.fe_release_amd64fre_SERVER_LOF_PACKAGES_OEM.iso](#)

II 运行 PS 命令行时：

1. 可选 “Terminal” 或 “PowerShell ISE”，未安装 “Terminal”，请前往 <https://github.com/microsoft/terminal/releases> 后下载；
2. 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`

3. 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；
4. 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。

B. 语言包：提取

II 准备语言包

挂载 [20348.1.210507-1500.fe_release_amd64fre_SERVER_LOF_PACKAGES_OEM.iso](#) 或解压到任意位置；

III 提取语言包方案

1. 添加

1.1. 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](#)

2. 删除

2.1. 语言名称：[英语 - 美国](#)，语言标记：[en-US](#)，适用范围：[Install.Wim](#)，[Boot.Wim](#)，[WinRE.Wim](#)

IV 执行提取命令

- [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](#), [Github](#))

`$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_OC\WinPE-FontSupport-{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\lp.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-securestartup_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-atbroker_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-audiocore_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-audiodrivers_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-enhancedstorage_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-narrator_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-scripting_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-speech-tts_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-srh_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-srt_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-wds-tools_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-wmi_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-appxpackaging_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{Lang}\winpe-storagewmi_{Lang}.cab"
            "Windows Preinstallation Environment\x64\WinPE_OC\{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 = "Lao" }
    "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
}
```

C. 自定义部署映像

II 自定义部署映像：Install.wim

1. 查看 Install.wim 详细信息

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](#)，

2. 指定挂载 Install.wim 路径

```
New-Item -Path "D:\OS2022_Custom\Install\Install\Mount" -ItemType directory -ea SilentlyContinue
```

3. 开始挂载 Install.wim

默认索引号：1

```
$Index = "1"
Mount-WindowsImage -ImagePath "D:\OS2022\sources\install.wim" -Index $Index -Path "D:\OS2022_Custom\Install\Install\Mount"
```

[处理 Install.wim 映像内的文件](#)，[可选项](#)，[开始](#)，

3.1. 自定义部署映像：WinRE.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”](#)。

3.1.1. 查看 WinRE.wim 详细信息

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
}
```

3.1.2. 指定挂载 WinRE.wim 路径

```
New-Item -Path "D:\OS2022_Custom\Install\WinRE\Mount" -ItemType directory -ea SilentlyContinue
```

3.1.3. 开始挂载 WinRE.wim

默认索引号：1

```
$Index = "1"
Mount-WindowsImage -ImagePath $FileName -Index $Index -Path "D:\OS2022_Custom\Install\WinRE\Mount"
```

3.1.4. 语言包

自动安装语言包：按组件匹配，匹配成功后，再安装本地对应的语言包文件。

3.1.4.1. 语言包：添加

复制代码或查看源文件：[WinRE.Instl.lang.ps1](#) ([本地](#), [Github](#))

```
$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
        }
    }
}
```

3.1.4.2. 脱机映像语言：更改

3.1.4.2.1. 更改默认语言、区域设置和其他国际设置

```
Dism /Image:"D:\OS2022_Custom\Install\WinRE\Mount" /Set-AllIntl:zh-CN
```

3.1.4.2.2. 查看可用的语言设置

```
Dism /Image:"D:\OS2022_Custom\Install\WinRE\Mount" /Get-Intl
```

3.1.4.3. 语言包：删除，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](#), [Github](#))

```
$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

    } catch {

        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

}

}

}
```

3.1.4.4. 组件

3.1.4.4.1. 查看

```
Get-WindowsPackage -Path "D:\OS2022_Custom\Install\WinRE\Mount" | Out-GridView
```

3.1.4.4.2. 导出到 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
```

3.1.5. 更新，可选

准备可用的更新文件，请更改示例文件名： KB_WinRE.cab

3.1.5.1. 添加

```
$KBPath = "D:\OS2022_Custom\Install\WinRE\Update\KB_Boot.cab"

Add-WindowsPackage -Path "D:\OS2022_Custom\Install\WinRE\Mount" -PackagePath $KBPath
```

3.1.5.2. 删除

```
$KBPath = "D:\OS2022_Custom\Install\WinRE\Update\KB_Boot.cab"

Remove-WindowsPackage -Path "D:\OS2022_Custom\Install\WinRE\Mount" -PackagePath $KBPath
```

3.1.5.3. **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
```

3.1.5.3.1. **固化更新后清理组件，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
    }
}
```

3.1.6. **驱动，可选**

3.1.7. **保存映像**

```
Save-WindowsImage -Path "D:\OS2022_Custom\Install\WinRE\Mount"
```

3.1.8. **卸载映像**

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
```

3.1.9. **重建 WinRE.wim 后，可缩小文件大小**

复制代码或查看源文件：[WinRE.Rebuild.ps1](#) ([本地](#), [Github](#))

```
$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
}
```


3.1.10. 备份 Install.wim

复制代码或查看源文件：[WinRE.Backup.ps1](#) ([本地](#), [Github](#))

```
$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
```

3.1.11. 替换 Install.wim 映像内的 WinRE.wim

- 每次挂载 Install.wim 后 “[替换 WinRE.wim](#)” ；
- 学习 “[获取 Install.wim 所有索引号后并替换旧的 WinRE.wim](#)” 。

[处理 Install.wim 映像内的文件](#)，结束。

4. 语言包

自动安装语言包：按组件匹配，匹配成功后，再安装本地对应的语言包文件。

4.1. 语言包：添加

复制代码或查看源文件：[Install.Instl.lang.ps1](#) ([本地](#), [Github](#))

```
$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

    }

}

}
```

4.2. 脱机映像语言：更改

- 从 Windows 11 开始，DISM 设置的默认系统 UI 语言在所有版本中保持不变（家庭版除外）。对于所有商业版，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，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。

4.2.1. 更改默认语言、区域设置和其他国际设置

```
Dism /Image:"D:\OS2022_Custom\Install\Install\Mount" /Set-AllIntl:zh-CN
```

4.2.2. 查看可用的语言设置

```
Dism /Image:"D:\OS2022_Custom\Install\Install\Mount" /Get-Intl
```

4.3. 语言包：删除，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](#), [Github](#))

```
$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

    }

}

}
```

4.4. 组件

4.4.1. 查看

```
Get-WindowsPackage -Path "D:\OS2022_Custom\Install\Install\Mount" | Out-GridView
```

4.4.2. 导出到 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
```

5. 更新, 可选项

5.1. 下载

查阅 “[Windows Server 2022 更新历史](#)” , 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](#)" (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
```

5.2. 添加

```
$KBPath = "D:\OS2022_Custom\Install\Install\Update\windows10.0-kb5030216-  
x64_cbe587155f9818548b75f65d5cd41d341ed2fc61.msu"  
Add-WindowsPackage -Path "D:\OS2022_Custom\Install\Install\Mount" -PackagePath $KBPath
```

5.3. 固化更新, 可选项

```
Dism /Image:"D:\OS2022_Custom\Install\Install\Mount" /cleanup-image /StartComponentCleanup /ResetBase
```

5.3.1. 固化更新后清理组件

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

```
$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  
    }  
}
```

6. 驱动, 可选

7. 健康

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
```

8. 替换 WinRE.wim

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
```

9. 保存映像

```
Save-WindowsImage -Path "D:\OS2022_Custom\Install\Install\Mount"
```

10. 卸载映像

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。

11. 如何批量替换 Install.wim 里的所有索引号里的 WinRE.wim

11.1. 获取 WimLib

前往 <https://wimlib.net> 官方网站后，选择不同的版本：[arm64](#), [x64](#), [x86](#)，下载完成后解压到：[D:\WimLib](#)

11.2. 如何在 Install.wim 里提取和更新 WinRE.wim

11.2.1. 从 Install.wim 里提取 WinRE.wim 文件 Install.wim

复制代码或查看源文件：[Install.WinRE.Extract.ps1](#) ([本地](#), [Github](#))

```
$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-imageex.exe" -ArgumentList $Arguments -wait -nonewwindow
```

11.2.2. 获取 Install.wim 所有索引号后并替换旧的 WinRE.wim

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

```
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
    }
}
```

12. 重建 Install.wim 后可缩小文件大小

复制代码或查看源文件：[Install.Rebuild.wim.ps1](#) ([本地](#), [Github](#))

```
$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
}
```

III 自定义部署映像：boot.wim

1. 查看 Boot.wim 文件信息

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
}
```

2. 指定挂载 Boot.wim 路径

```
New-Item -Path "D:\OS2022_Custom\Boot\Boot\Mount" -ItemType directory -ea SilentlyContinue
```

3. 开始挂载 Boot.wim

默认索引号：2

```
$Index = "2"
Mount-WindowsImage -ImagePath "D:\OS2022\sources\boot.wim" -Index $Index -Path "D:\OS2022_Custom\Boot\Boot\Mount"
```

4. 语言包

自动安装语言包：按组件匹配，匹配成功后，再安装本地对应的语言包文件。

4.1. 语言包：添加

复制代码或查看源文件：[Boot.Instl.lang.ps1](#) ([本地](#), [Github](#))

```
$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
    }
}
```

```
}
```

4.2. 脱机映像语言：更改

4.2.1. 更改默认语言、区域设置和其他国际设置

```
Dism /Image:"D:\OS2022_Custom\Boot\Boot\Mount" /Set-AllIntl:zh-CN
```

4.2.2. 查看可用的语言设置

```
Dism /Image:"D:\OS2022_Custom\Boot\Boot\Mount" /Get-Intl
```

4.3. 语言包：删除，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](#), [Github](#))

```
$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

    }

}

}

```

4.4. 组件

4.4.1. 查看

```
Get-WindowsPackage -Path "D:\OS2022_Custom\Boot\Boot\Mount" | Out-GridView
```

4.4.2. 导出到 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
```

4.5. 语言包：同步到 ISO 安装程序

```
Copy-Item -Path "D:\OS2022_Custom\Boot\Boot\Mount\sources\zh-CN" -Destination "D:\OS2022\sources\zh-CN" -Recurse -Force
```

4.6. 重新生成 Lang.ini

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。

4.6.1. 重新生成已挂载目录 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"
```

4.6.2. 重新生成 lang.ini 后，同步到安装程序

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"
```

5. 更新，可选项

准备可用的更新文件，请更改示例文件名：[KB_Boot.cab](#)

5.1. 添加

```
$KBPath = "D:\OS2022_Custom\Boot\Boot\Update\KB_Boot.cab"
Add-WindowsPackage -Path "D:\OS2022_Custom\Boot\Boot\Mount" -PackagePath $KBPath
```

5.2. 删除

```
$KBPath = "D:\OS2022_Custom\Boot\Boot\Update\KB_Boot.cab"
```

```
Remove-WindowsPackage -Path "D:\OS2022_Custom\Boot\Boot\Mount" -PackagePath $KBPath
```

5.3. 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
```

5.3.1. 固化更新后清理组件，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
    }
}
```

6. 驱动，可选项

7. 保存映像

```
Save-WindowsImage -Path "D:\OS2022_Custom\Boot\Boot\Mount"
```

8. 卸载映像

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
```

D. 生成 ISO

- II 下载 oscding 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；
- III 使用 oscding 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](#), [Github](#))

```
$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
```

二、 常见问题

- II 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
```

III 修复挂载出现异常的问题

- 1. 查看已挂载

```
Get-WindowsImage -Mounted
```

- 2. 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
```

- 3. 删除与已损坏的已装载映像关联的所有资源。

```
Clear-WindowsCorruptMountPoint
```

IV 清理自定义目录

```
Remove-Item "D:\OS2022_Custom" -Force -Recurse -ErrorAction SilentlyContinue
```

三、 已知问题

- 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。

四、 组件：未变更

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
1		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
2		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
3		Microsoft-OneCore-DirectX-Database-FOD-Package~31bf3856ad364e35~amd64~~10.0.20348.1	Microsoft-OneCore-DirectX-Database-FOD-Package~31bf3856ad364e35~amd64~~10.0.20348.1
4		Microsoft-OneCore-RasSstp-Api-Package~31bf3856ad364e35~amd64~~10.0.20348.1	Microsoft-OneCore-RasSstp-Api-Package~31bf3856ad364e35~amd64~~10.0.20348.1
5		Microsoft-Windows-FodMetadata-Package~31bf3856ad364e35~amd64~~10.0.20348.1	Microsoft-Windows-FodMetadata-Package~31bf3856ad364e35~amd64~~10.0.20348.1
6		Microsoft-Windows-Foundation-Package~31bf3856ad364e35~amd64~~10.0.20348.1	Microsoft-Windows-Foundation-Package~31bf3856ad364e35~amd64~~10.0.20348.1
7		Microsoft-Windows-InternetExplorer-Optional-Package~31bf3856ad364e35~amd64~~11.0.20348.1	
8		Microsoft-Windows-InternetExplorer-Optional-Package~31bf3856ad364e35~amd64~~11.0.20348.75	
9		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

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
10		Microsoft-Windows-LanguageFeatures-Handwriting-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
11		Microsoft-Windows-LanguageFeatures-OCR-en-us-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
12		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
13		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
14		Microsoft-Windows-MediaPlayer-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
15		Microsoft-Windows-MediaPlayer-Package~31bf3856ad364e35~amd64~~10.0.20348.169	
16		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
17		Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
18		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
19		Microsoft-Windows-MSPaint-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1	
20		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
21		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
22		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
23		Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
24		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
25		Microsoft-Windows-Notepad-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1	
26		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
27		Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1	
28		Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
29		Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1	
30		Microsoft-Windows-PowerShell-ISE-FOD-Package~31bf3856ad364e35~wow64~~10.0.20348.1	
31		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
32		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
33		Microsoft-Windows-ServerCore-Package~31bf3856ad364e35~amd64~~10.0.20348.1	Microsoft-Windows-ServerCore-Package~31bf3856ad364e35~amd64~~10.0.20348.1
34		Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1	
35		Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
36		Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~en-US~10.0.20348.1	
37		Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~~10.0.20348.1	
38		Microsoft-Windows-TabletPCMath-Package~31bf3856ad364e35~amd64~~10.0.20348.1	

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
39		Microsoft-Windows-TabletPCMath-Package~31bf3856ad364e35~amd64~~10.0.20348.143	
40		Microsoft-Windows-UserExperience-Desktop-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
41		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
42		Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1	Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~amd64~~10.0.20348.1
43		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
44		Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1	Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~~10.0.20348.1
45		Microsoft-Windows-Xps-Xps-Viewer-Opt-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
46		OpenSSH-Client-Package~31bf3856ad364e35~amd64~~10.0.20348.1	OpenSSH-Client-Package~31bf3856ad364e35~amd64~~10.0.20348.1
47		Package_for_DotNetRollup~31bf3856ad364e35~amd64~~10.0.4400.	Package_for_DotNetRollup~31bf3856ad364e35~amd64~~10.0.4400.1
48		Package_for_RollupFix~31bf3856ad364e35~amd64~~20348.169.1.7	Package_for_RollupFix~31bf3856ad364e35~amd64~~20348.169.1.7
49		Package_for_ServicingStack~31bf3856ad364e35~amd64~~20348.169.1.1	Package_for_ServicingStack~31bf3856ad364e35~amd64~~20348.169.1.1

五、 组件：新增后变化

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
1	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
2	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
3	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
4	Handwriting	Microsoft-Windows-LanguageFeatures-Handwriting-zh-CN-Package~31bf3856ad364e35~AMD64~~.cab	
		Microsoft-Windows-LanguageFeatures-Handwriting-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
5	OCR	Microsoft-Windows-LanguageFeatures-OCR-zh-CN-Package~31bf3856ad364e35~AMD64~~.cab	
		Microsoft-Windows-LanguageFeatures-OCR-zh-cn-Package~31bf3856ad364e35~amd64~~10.0.20348.1	
6	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
7	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
8	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
9	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
10	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
11	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
12	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	
13	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	
14	按需包	Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~zh-CN~.cab	
		Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~amd64~zh-CN~10.0.20348.1	
15	按需包	Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~zh-CN~.cab	
		Microsoft-Windows-StepsRecorder-Package~31bf3856ad364e35~wow64~zh-CN~10.0.20348.1	
16	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
17		Microsoft-Windows-WordPad-FoD-Package~31bf3856ad364e35~wow64~zh-CN~.cab	

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
	功能包	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