Yi' Solutions

Yi's Soultions

Windows installed languages, Yi's optimization scripts, etc.

To solve the difficult problem of packaging multi-lingual versions for you, through this, you can initiate this packaging journey at will and end this "unpackageable journey".

Component

A. Packaging Tutorial

The packaging tutorial written by Yi can optionally start the packaging journey of Windows 11 23H2, 22H2, Windows 10, and Windows Server 2022. Different packaging versions are available.

B. Video tutorial

The video tutorial includes different packaging methods: custom allocation of packaging events, automatic driving, manual packaging, and introduction to packaging scripts.

C. Package script

Developed using the PowerShell language, it follows an open source license and can be distributed arbitrarily without copyright restrictions.

D. Local Language Experience Packs (LXPs) Downloader

Solve the problem of batch downloading of "Local Language Experience Packages (LXPs)" installation packages, and you can filter or download all.

Learn more: https://github.com/ilikeyi/LXPs, included in the full version: _Encapsulation_Custom\Engine\LXPs

DEPLOYMENT ENGINE

E. Fully automatically add Windows installed languages

Has the basic functions of a deployment engine and does not include others.

Learn more: https://github.com/ilikeyi/Multilingual, included in the full version: _Encapsulation_Custom\Engine\Multilingual

F. Yi's optimization script

It has the basic functions of a deployment engine, including: optimization scripts, common software installation, software installation, system optimization, service optimization, UWP uninstallation, changing folder location, etc.

Learn more: https://github.com/ilikeyi/Yi.Optimiz.Private, included in the full version: _Encapsulation_Custom\Engine\Yi.Optimiz.Private

Table of contents

Preface

Thank you for using the solution developed by Yi. It took more than 4 years from development to public release. During these 4 years, it was very difficult for one person to develop and solve difficult and complicated problems alone, and there were also long setbacks. And suffering, even so, there is still a lot of work to be done in the future.

Origin of Yi's Solutions:

Because I am accustomed to using English as my preferred choice, but I also want to have a Chinese language pack, I can switch to different languages at any time. At first, I just wanted to make the simplest "Windows 10 Multilingual Edition: United States - English, China - Simplified Chinese", integrating 2 languages in one version.

What you want to ask: I don’t know how to add the English language pack after installing the Chinese version? Or add the Chinese language pack after installing the English version?

Answer: What I want is to have multiple different languages at once after the installation is completed, rather than manually adding them after installing the system.

It has only been 4 years since multilingualism was successfully added to offline packaging. When looking back on these 4 years, the conclusion is:

- This method is generally applicable to Windows 11, Windows 10, Windows Server 2022, and Windows Server vNext.
- It is different from the learning and production methods officially provided by Microsoft. If you do not follow the plans provided by the official, you will have to settle for it.

The production methods are ever-changing, just like soldiers, there are also deceptions (from: Sun Tzu's Art of War).

Among them, we need to continuously improve the development of packaging scripts, deployment engines, and local language experience packages (LXPs) downloaders. The most difficult ones to solve and develop are:

1. PowerShell

    1.1. It took several months to complete batch processing of mounted and unmounted tasks, batch development and batch problems. During that time, I often looked at the screen in a daze.

    1.2. 50 graphical interfaces . The most difficult ones to develop are:

        1.2.1.1. The image source includes different embedded pages: mounting, language selection, detailed information, ISO file extraction, interface setting and other development. A single page of code is close to 8000 lines, written entirely by hand.

        1.2.1.2. Solution generation includes: adding deployment engines, adding deployment tags, adding software packages, different architectures, adding Office installation packages, adding fonts and other functions. The code takes over 7400 lines and is deployed to the image source. After being installed in the system, Test script is required.

        The most difficult thing is to manually adjust the control size, which takes several days and no less than 5 large adjustments.

    1.3. Autopilot

    Autonomous driving is the most difficult to develop. It is equivalent to an archaeological project. It took a full three months to complete this function.

2. Packaging system

    2.1. The encapsulation script needs to solve the problems encountered during addition and batch problems. It is really difficult to find and fix problems;

    2.2. adding the InBox Apps application, you need to test whether all reinstalled applications can run normally while disconnecting from the Internet;

    2.3. Understand InBox Apps application dependencies, create wrapper script rules, and more.

2.4. After packaging is completed, test whether there are any new problems, etc.;

3. Documentation

3.1. Writing and revising 8 packaging tutorials, code testing, typesetting, review of a total of 400 pages and more than 30,000 words, etc.

CHAPTER 1 Part introduction

A. Packaging tutorial

Different versions are provided: a full version and a simplified version. The formats provided are: .Docx document format, .Pdf document format. Version differences:

1. Complete version, no deleted content;
2. The streamlined version does not include: reports, notes, etc.;

Tutorials available for the packaging journey include:

Optional language versions: Simplified Chinese version, English version

(Google Translate: Chinese to English), download the complete package to get all documents: [Compressed package]:_Learn\Packaging.tutorial, or go to https://github.com/ilikeyi/solutions/tree/main/_Learn/Packaging.tutorial and select .

B. Video tutorial

1. Packaging tutorial

1.1. Windows 11 23H2: Practical packaging tutorial

- Youtube | https://youtu.be/BWttnewLv-s
- Bilibili | https://www.bilibili.com/video/BV1sj421R7uj/
- Tencent Video | https://v.qq.com/x/page/j3543gs3pv7.html
- Watermelon video | https://www.ixigua.com/7348909159569424946?utm_source=Readme

2. Custom encapsulated events

2.1. Windows 11 23H2: Custom encapsulated events

- Youtube | https://youtu.be/e6mzybgMHF0
- Bilibili | https://www.bilibili.com/video/BV1HK421e7AV/
- Tencent Video | https://v.qq.com/x/page/c3543pyggh0.html
- Watermelon video | https://www.ixigua.com/7348904251847868966?utm_source=Readme

3. Autopilot

3.1. Windows 11 23H2: Autopilot encapsulation

- Youtube | https://youtu.be/BbS_T2d9Ifc
- Bilibili | https://www.bilibili.com/video/BV1Mt421G7Uf/
- Tencent Video | https://v.qq.com/x/page/l35436bsird.html
- Watermelon video | https://www.ixigua.com/7348896802180956683?utm_source=Readme

C. Package script

1. The main functions of the encapsulated script

1.1. Check for updates: In order to better stay up to date with the latest version, you can check whether the latest version is available at any time.

1.2. Hot refresh: After changing the script, enter R in the main interface and execute "reload module" to complete the hot refresh.

1.3. Language pack: United States - English, 中文 (简体), 中文 (繁体), 대한민국 - 한국어, 日本 - 日本語

1.4. Event Pattern

1.4.1. Autopilot

1.4.1.1. Prerequisites

1.4.1.2. Import public libraries

Different rules can be set: cumulative updates, drivers

1.4.1.3. Others: Import from different configuration files, you can customize the selection of import items, associate ISO schemes, etc.

1.4.2. Custom allocation events

When assigning, you can customize the items to be assigned

1.4.3. Manual operation

1.4.3.1. All operations can be interrupted, and only the current task can be interrupted.

1.4.3.2. Perception function

1.5. Descending order: Automatically identify ARM64, x64, and x86 architectures, and automatically select dependent programs in descending order according to the architecture.

1.6. 1.6. ISO: Automatically identify ISO tag names and initialize rules (supports inclusion class matching), decompress, mount, pop up, verify hash, display corresponding ISO files according to rules, search, automatically classify: files, language packages, function packages , InBox Apps

1.7. Perception function

1.7.1. Add language pack, combo: add language pack, add cumulative update, save mounted, generate ISO

1.7.2. Add cumulative update, combo: add cumulative update, save mounted, generate ISO

This is the perception function, which can be "specified global perception" or "customized current perception" in the settings interface. How to customize the sensing options, please refer to: Encapsulation Script Development Guide.

1.8. Fix

1.8.1. Delete DISM mount records saved in the registry

1.8.2. Delete all resources associated with the corrupted mounted image

1.8.3. After adding the routing function, you can run: Yi -Fix, or select it in the setting interface.

1.9. Mount points

1.9.1. Can be customized and specified to be mounted to

1.9.2. Automatically search all local disks and automatically select the disk with the volume name: RAMDISK. The initial volume name can be modified. This function is enabled by default.

2. For the main functions of the image source

Aimed at encapsulating the main functions of the Windows operating system, it supports batch operations of main items and extensions.

2.1. Event

For example, when operating WinRE.wim , you need to mount Install.wim before mounting WinRe.wim so that you can perform the corresponding tasks for WinRE.

What are the files within the image? For example, Install.wim contains the WinRE.wim file. After mounting install.wim , events can be assigned to process WinRe.wim .

Main functions: Mounted or unmounted events can be assigned. The main trigger events can be assigned:

- Main item: Boot.wim
- Main item: Install.wim , files within the image (extension items): WinRE.wim

2.2. Event handling

Event processing is divided into several options: no need to mount the image, item mode that requires the image to be mounted, and support for main image and batch processing within the image.

2.2.1. No need to mount image

2.2.1.1. Add, delete, update files within the image, extract, rebuild, apply

2.2.1.2. Extract language pack

2.2.1.3. Convert Esd and Wim to each other

2.2.1.4. Split Install.wim into Install.swm

2.2.1.5. Merge install.swm to install.wim

2.2.1.6. Generate ISO

- Synchronize all known local languages
- Generate tags: multilingual tags, monolingual tags, calculate image version, calculate installed languages, release year and month, version code
- Customization: ISO volume label name, ISO file name, specified save to

2.2.2. You need to mount the image before you can operate the item

2.2.2.1. Language pack

Add languages, reversely delete languages, change image default language, clean up obsolete components.

- During installation: Automatically classify language packages and function packages according to regional tags, and automatically match all installed packages in the image.
- Extraction: Extract language packages according to rules, custom-select language tags, classified known associations;
- Language pack: sync to ISO installer

- Regenerate Lang.ini
- Cumulative update: After installing the language pack, you must add a cumulative update (you can install the same version number as the initial version or the latest cumulative update), because before the cumulative update is added, there will be no changes to the components. It will not happen until you install the cumulative update. New changes, such as component status: obsolete, to be deleted;

2.2.2.2. Local Language Experience Packages (LXPs)

Mark, add, delete, delete by matching rules

2.2.2.3. InBox Apps

Mark, add, delete, delete according to matching rules, infinitely customize different image versions to pre-install applications, etc. For details, see: Package Script Development Guide.

2.2.2.3.1. Step 1: Install Local Language Experience Packs (LXPs), region tags

- InBox Apps applications, there are two methods: one is to add language packages; the other is to add local language experience packages (LXPs) for marking.
- After the two methods are added, the offline image language will be added, and the added InBox Apps application will be matched according to the installed language of the offline image; this is the so-called region tagging function.

2.2.2.3.2. Step 2: Customize the InBox Apps application and check the dependencies before adding it

2.2.2.3.3. Step 3: Clean up local language experience packages (LXPs)

2.2.2.3.4. After mounting: installed items can be managed and deleted

When batched or not mounted, you can fuzzy match the application name and delete it.

2.2.2.4. Cumulative updates

Language packs, Local Language Experience Packs (LXPs), InBox Apps, and cumulative updates are combos and combination punches.

2.2.2.5. Drive

Add, delete

2.2.2.6. Windows features

Enable, disable, support mounted post-processing enable, disable

2.2.2.7. Run a PowerShell function

- What is a function? You can create custom functions, write custom code, and obtain all variable names, global parameters, etc. available in PowerShell.
- Distributable: Run the PowerShell function before the task, and run the PowerShell function after the task is completed.

2.2.2.8. Solution: Generate

It can be generated into the image source and mounted offline items. It can generate: deployment engine, response pre-answer, software package, customized collection package, adding Microsoft Office installation package, etc. When deploying software, it supports arm64, x64, x86 in descending order. Added, supports single language and multi-language deployment

2.2.2.8.1. Deployment engine

2.2.2.8.1.1. First experience, in the process of deploying prerequisites

- Allow global search and synchronization of deployment tags
- Allow automatic updates
- Add home directory to Defend exclude directory
- Disable network location wizard
- System disk volume label: The home directory name is the same
- When encountering multiple languages
  - Block Appx cleanup maintenance tasks
  - Prevent cleanup of unused on-demand feature language packs
  - Prevent cleaning of unused language packs
- Add a personalized "context menu"
- Change system locale

2.2.2.8.1.2. First time experience, after completing prerequisites

- Pop up the deployment engine main interface
- Allow first preview, as planned
- Restore Powershell execution policy: restricted
- Delete the entire solution
- Delete the deployment engine and keep the others

2.2.2.8.2. Should answer in advance

2.2.2.8.2.1. Optional preset architecture core version: 11, 10

2.2.2.8.2.2. Specifies the command that should be pre-deployed

2.2.2.8.2.3. Specify single language or multiple languages

2.2.2.8.2.4. Specify Autounattend.xml scheme

- Semi-automatic is valid for all installation methods
- EFI automatic installation
- Egacy automatic installation

2.2.2.8.2.5. Installation interface: Hide product key, hide selection of operating system to install, hide acceptance of license terms

2.2.2.8.2.6. For server version:

- Server Manager does not start automatically on login
- Internet Explorer Enhanced Security Configuration: Shut down administrators, shut down users

2.2.2.8.2.7. Specify time zone

2.2.2.8.3. Add collection

2.2.2.8.3.1. After selecting the preferred architecture package, it will be added automatically in descending order.

2.2.2.8.3.2. Choose to deploy the Microsoft Office installation package: specify language, specify add to, optional versions: Office 365, Office 2024, Office 2021, Office 2019

2.2.2.8.3.3. Add package

2.2.2.8.3.4. Add font

2.2.2.9. Generate report

Can generate: health status, installed application packages, offline installed languages, installed InBox Apps applications, drivers

2.2.2.10. pop up

Main functions of pop-up: save, not save, support pop-up extensions, use WimLib to update files in the image after pop-up.

CHAPTER 2 Start the packaging journey

A. Prerequisites

I. Require

PowerShell version

- PowerShell 5.1

  Requires Windows 11, Windows 10, Windows Server 2022, Windows Server vNext or the 5.1 version that comes with the system by default. You can optionally upgrade to the latest version of PowerShell 7.

- PowerShell 7

  To get the latest version, go to https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows After that, select the version you want to download, download and install it.

II. Command Line

1. "Terminal" or "PowerShell ISE" is optional. If "Terminal" is not installed, please go to https://github.com/microsoft/terminal/releases then download;

2. Open "Terminal" or "PowerShell ISE" as an administrator, set the PowerShell execution policy: Bypass, PS command line:

   Set- ExecutionPolicy - ExecutionPolicy Bypass-Force

3. In this article, the green part belongs to the PS command line. Please copy it, paste it into the "Terminal" dialog box, and press Enter to start running;

4. there is .ps1 , right-click the file and select Run as PowerShell, or copy the path and paste it into " Terminal" or "PowerShell ISE" to run. For the path with a colon, add the & character in the command line, example: & "D:\ YiSolutions_Encapsulation_SIP.ps1"

III. Get Yi's Solutions

1. Project address

1.1. Official website

1.1.1. Automatic download

Set-ExecutionPolicy -ExecutionPolicy Bypass -Force

irm https://fengyi.tel/get.ps1 | iex

Prioritize downloading from the official website. After the download is completed: add routing function. Run the wrapper script.

1.1.2. Manual download

Go to https://fengyi.tel/solutions Then check the download items, or open https://fengyi.tel/go/solutions and download directly.

1.2. Gihtub

1.2.1. Automatic download

Set-ExecutionPolicy -ExecutionPolicy Bypass -Force

irm https://github.com/ilikeyi/Solutions/raw/main/get.ps1 | iex

Prioritize downloading from the Github website. After the download is completed: add the routing function and run the packaging script.

1.2.2. Manual download

Go to https://github.com/ilikeyi/solutions Then select " Code" and then select Download ZIP.

Or go to https://github.com/ilikeyi/solutions/releases , select the available version you want to download, and click to download the source code (zip, tar.gz).

2. Netdisk download

2.1. Alibaba Cloud Cloud Disk | https://www.alipan.com/s/sFU4uaJ6uV3

2.2. 123 Netdisk | https://www.123pan.com/s/zitA-QU9l.html

2.3. Google Drive | https://drive.google.com/drive/folders/1qTgFvbETlk23v_RGw_rXQPcZvlVirO-O?usp=sharing

3. After the download is complete , unzip the downloaded file to: D:\YiSolutions

IV. PowerShell script

1. Prerequisites: Once met, run the package main script

   D:\YiSolutions_Encapsulation_SIP.ps1

   After entering the main interface of the packaging script, you can add the routing function to the system variable. After adding it, run Yi in the PowerShell terminal next time to enter the boot interface, or enter Yi -sip to directly enter the packaging interface without entering the full path of the script. run.

2. Other items

   2.1. Backup

   D:\YiSolutions_Encapsulation_Unpack.ps1, when routing function is available: Yi - unpack

   2.2. Create upgrade package

   D:\YiSolutions_Encapsulation_Create.Upgrade.Package.ps1, when routing function is available: Yi -CU

   2.3. Create a deployment engine upgrade package

   D:\YiSolutions_Encapsulation_Create.Custom.Engine.upgrade.package.ps1, when routing function is available: Yi -CEUP

   2.4. Convert all software into compressed packages

   D:\YiSolutions_Encapsulation_Zip.ps1, when routing function is available: Yi -Zip

   2.5. Create a template: cumulative updates, drivers

   D:\YiSolutions_Encapsulation_Ct.ps1, when routing function is available: Yi -CT

B. Deploy software

1. Microsoft Office

   When you need to deploy the ODT version of Microsoft Office 2024, Microsoft Office 2021, Microsoft Office 2019, or Microsoft 2016, please download it according to the downloaded architecture version, for example, download:

   1.1. Download Microsoft Office 365

   x64: D:\YiSolutions_Encapsulation_Custom\Office\365\amd64\Download.x64.ps1

   x86: D:\YiSolutions_Encapsulation_Custom\Office\365\amd64\Download.x86.ps1

   1.2. Download Microsoft Office 2021

   x64: D:\YiSolutions_Encapsulation_Custom\Office\2021\amd64\Download.x64.ps1

   x86: D:\YiSolutions_Encapsulation_Custom\Office\2021\amd64\Download.x86.ps1

2. Custom packages

When generating a solution, if a region tag is specified, the matched software packages will be copied automatically. When not found, en -US is copied as the default. If there is no corresponding architecture, please create x86 as the default. When generating, add in descending order: arm64, x64, x86.

2.1. 2.1. Custom package: Create

2.1.1. 7zip

7zip Known available languages: One installation package already contains multiple languages, with different architecture versions: arm64, x64, x86

2.1.1.1. en-US

Creating the directory:schema\ en -US, copy the application package to that directory

Arm64: D:\YiSolutions_Encapsulation_Custom\Software\00\7z\arm64\en-US , installation package: 7z2301-arm64.exe

x64: D:\YiSolutions_Encapsulation_Custom\Software\00\7z\AMD64\en-US , installation package: 7z2301-x64.exe

x86: D:\YiSolutions_Encapsulation_Custom\Software\00\7z\x86\en-US , installation package: 7z2301.exe

2.1.2. WinRAR

WinRAR language is only monolingual, divided according to different language areas, and has different architecture versions: x64, x86

2.1.2.1. en-US

Creating the directory:schema\ en -US, copy the application package to that directory

x64: D:\YiSolutions_Encapsulation_Custom\Software\00\WinRAR\AMD64\en-US , installation package: winrar-x64-624.exe

x86: D:\YiSolutions_Encapsulation_Custom\Software\00\WinRAR\x86\en-US , installation package: winrar-x32-624.exe

2.1.2.2. zh-CN

Creating the directory:schema\ en -US, copy the application package to that directory

x64: D:\YiSolutions_Encapsulation_Custom\Software\00\WinRAR\AMD64\en-US , installation package: winrar-x64-624sc.exe

x86: D:\YiSolutions_Encapsulation_Custom\Software\00\WinRAR\x86\en-US , installation package: winrar-x32-624sc.exe

2.1.2.3. Others are not listed. Please refer to the above directory structure and create it when making it. For other versions, please refer to the official website.

2.2. Custom software: Convert to compressed package

After converting to zip, the file size will be reduced. Yi 's optimization script has been included: during the first experience, all zip archives will be automatically decompressed.

3. Fonts

Different fonts can be added to: D:\YiSolutions_Encapsulation_Custom\Fonts . When you experience deployment for the first time, the fonts will be automatically installed.

C. Virtual memory disk

What is a memory disk? Memory disk, also known as virtual memory disk, is a technology that improves quick access to computer memory and files. However, the memory disk will cause data loss after the computer is shut down. The memory disk is a relatively unsafe setting.

Even so, I don't think so. During the encapsulation process, installation package files will be frequently released, logs will be generated, etc. When mounting to a virtual disk, this has many benefits, including quick formatting.

1. Matching suggestions

When adding language packs, cumulative updates, and InBox Apps, the installation package is stored in a memory virtual disk, which will occupy a large amount of memory. It is recommended that you store it in a non-virtual memory disk.

2. Software recommendation

1.1. Ultra RAMDisk | http://ultraramdisk.com

1.2. ImDisk | https://sourceforge.net/projects/imdisk-toolkit

1.3. AMD Radeon RAMDisk | http://www.radeonramdisk.com

1.4. Primo Ramdisk | https://www.romexsoftware.com/en-us/primo-ramdisk/index.html

1.5. SoftPerfect RAM Disk | https://www.softperfect.com/products/ramdisk

1.6. StarWind RAM Disk | https://www.starwindsoftware.com/high-performance-ram-disk-emulator

3. How to create

When creating a memory disk, you should calculate the unused rate of physical memory, open "Task Manager", "Performance", and check the remaining memory rate. Suggestions:

3.1. When the physical memory is 16G and the system has 9G remaining, it is recommended to divide it into: 6G memory + 40G swap file, leaving more than 3G of remaining memory;

3.2. Physical memory 32G: When the system has 26G remaining, it is recommended to divide it into: 22 memory + 40G swap file, and keep the remaining memory above 4G;

3.3. Physical memory 64G: When the system has 54G remaining, only 50G memory is divided, there is no need to create a swap file, and the remaining memory is reserved for more than 4G;

3.4. Physical memory 128G: When the system has 115G remaining, divide the memory between 40-110G. There is no need to create a swap file and keep the remaining memory above 8G.

Note: Insufficient memory can cause problems during the encapsulation process.

CHAPTER 3 Wrapping scripts: developer's guide

1. Essential tools

1.1. Visual Studio Code | https://code.visualstudio.com/Download

1.2. Sublime Text | https://www.sublimetext.com

When there are some operations that Visual Studio Code cannot complete, use Sublime Text to achieve the best results. Optional

2. Import the project into developer tools

   2.1. Install Visual Studio Code and open the software

   2.2. Please obtain the packaging script first and extract it to: D:\YiSolutions

   2.3. Select Visual Studio Code to browse, select "Open Directory", select the D:\ YiSolutions directory and import it to the project list

3. Custom rules

Before creating custom rules, please refer to the inclusion and exclusion of InBox Apps application rules. Note that after online update, the enabled custom rules will not be synchronized to the new version. After the update is completed, please manually copy the custom rules to the new version. version, online updates using the repair function will reset all files.

3.1. Learn

3.1.1. Learn about preset rules

3.1.1.1. Contains InBox Apps

{Compressed package }:_Encapsulation\Modules\1.0.0.0\Functions\Custom\Solutions.Custom.With.InBox.Apps.psm1

3.1.1.2. Does not include InBox Apps

{Compressed package }:_Encapsulation\Modules\1.0.0.0\Functions\Custom\Solutions.Custom.Only.Language.psm1

3.1.2. Find dependencies from InBox Apps applications

the pre-rule Solutions. Custom.With.InBox.Apps .psm1 configuration file, all application dependencies are found from the installation package list. The installation package is found in different files. Please install the 7zip software first. What is the example? Find:

3.1.2.1. Microsoft.WindowsAlarms_8wekyb3d8bbwe.msixbundle

After selecting the file, use 7z to open the compressed package, find TimeUniversal_11.2304.0.0_x64.msix and double-click to open it. Find AppxManifest.xml, save it in any directory or double-click it to open it. After opening:

Look between <Dependencies> and </Dependencies> for details:

```
<Dependencies>
  <PackageDependency Name="Microsoft.UI.Xaml.2.8" MinVersion="8.2207.14002.0"/>
  <PackageDependency Name="Microsoft.NET.Native.Framework.2.2" MinVersion="2.2.29512.0"/>
  <PackageDependency Name="Microsoft.NET.Native.Runtime.2.2" MinVersion="2.2.28604.0"/>
  <PackageDependency Name="Microsoft.VCLibs.140.00" MinVersion="14.0.30704.0"/>
  <PackageDependency Name="Microsoft.VCLibs.140.00.UWPDesktop" MinVersion="14.0.30704.0"/>
 </Dependencies>
```

Dependent frameworks and minimum version numbers: Microsoft.UI.Xaml.2.8, Microsoft.NET.Native.Framework.2.2, Microsoft.NET.Native.Runtime.2.2, Microsoft.VCLibs.140.00, Microsoft.VCLibs.140.00.UWPDesktop

3.1.2.2. Microsoft.GetHelp_8wekyb3d8bbwe.appxbundle

After selecting the file, use 7z to open the compressed package, find GetHelpApp_10.2201.421.0_x64.appx and double-click to open it. Find AppxManifest.xml, save it in any directory or double-click it to open it. After opening:

Look between <Dependencies> and </Dependencies> for details:

```
<Dependencies>

  <PackageDependency Name="Microsoft.UI.Xaml.2.7" MinVersion="7.2109.13004.0"/>

  <PackageDependency Name="Microsoft.NET.Native.Framework.2.2" MinVersion="2.2.29512.0"/>

  <PackageDependency Name="Microsoft.NET.Native.Runtime.2.2" MinVersion="2.2.28604.0"/>

  <PackageDependency Name="Microsoft.VCLibs.140.00" MinVersion="14.0.27810.0"/>

 </Dependencies>
```

Dependent frameworks and minimum version numbers: Microsoft.UI.Xaml .2.7, Microsoft.NET.Native.Framework.2.2, Microsoft.NET.Native.Runtime.2.2, Microsoft.VCLibs.140.00

3.1.2.3. Other

Each application is different and located in different locations. Some are under the main application package, and some are installed in different architectures.

3.2. Create custom rules

Path: D:\YiSolutions_Encapsulation\Modules\1.0.0.0\Functions\Custom

3.2.1. Edit

Edit using tool: Solutions.Custom.Extension.psm 1

3.2.2. Rename

Rename Solutions.Custom.Extension.psd1.Template to Solutions.Custom.Extension.psd1 and delete .Template .

3.2.3. Verify

Verify that there are no error entries after running the wrapper script.

3.3. Precautions

reinstalling the InBox Apps: You must perform an installation test; you must test whether all known InBox Apps can be opened normally while disconnected from the Internet .

When Windows 11 23H2 was produced, items that needed to be fixed were listed in the prerequisites. Due to the InBox Apps installation package officially provided by Microsoft :

1. Short of a pound or two

2. Supplied application is damaged etc.

4. Regular development

4.1. Quickly locate " Image source is no longer processed"

Development skills, such as quickly locating " Image source no longer processed" and searching for text, the search results are:

D:\YiSolutions_Encapsulation\Modules\1.0.0.0\langpacks\zh-CN\Events.psd1

2,2: AssignSkip = Image source is no longer processed

After copying and naming, search again: $ lang.AssignSkip , search results:

D:\YiSolutions_Encapsulation\Modules\1.0.0.0\Functions\Events\Assign\Solutions.Image.Assign.psm1

1976,20: Text = $ lang.AssignSkip

In this way, positioning is completed quickly.

5. Cooperation

If you encounter new problems during the development process, please contact us through the following contact information:

Author: Yi

Website: https://fengyi.tel

Suggestions or feedback: https://github.com/ilikeyi/solutions/issues

Email: 775159955@qq.com , ilikeyi@outlook.com

instant messaging

- QQ: 775159955
- WeChat: FengYi