Yi' Solutions

Yi's Soultions

由多部分组成：封装脚本、封装教程、视频教程、部署引擎：全自动添加 Windows 已安装语言、Yi's 优化脚本等。

为你解决封装多语版的疑难问题，通过 Yi 提供的制作方法，搭配部署引擎可完美的解决此问题，你可任意的发起这一场封装之旅，从此结束这场“不可封装之旅”。

组成部分

A. 封装教程

由 Yi 编写的封装教程，可选开启 Windows 11 24H2、23H2、22H2、Windows 10 22H2、Windows Server 2025、Windows Server 2022 的封装之旅，有不同的封装版本可选。

B. 视频教程

视频教程包含了不同的封装方法：自定义分配封装事件、自动驾驶、手动封装，以及封装脚本介绍等。

C. 封装脚本

使用 PowerShell 语言开发，遵循开源协议，可任意分发，不受版权限制。

了解更多：https://github.com/ilikeyi/Solutions

D. 本地语言体验包（LXPs）下载器

解决批量下载“本地语言体验包（LXPs）”安装包，可筛选或下载全部。

了解更多：https://github.com/ilikeyi/LXPs，已包含在完整版内：_Encapsulation_Custom\Engine\LXPs

部署引擎

E. 全自动添加 Windows 已安装语言

拥有部署引擎基本功能，不包含其它。

了解更多：https://github.com/ilikeyi/Multilingual，已包含在完整版内：_Encapsulation_Custom\Engine\Multilingual

F. Yi's 优化脚本

拥有部署引擎基本功能，包含：优化脚本、常用软件安装、软件安装、系统优化、服务优化、UWP 卸载、更改文件夹位置等。

了解更多：https://github.com/ilikeyi/Yi.Optimiz.Private，已包含在完整版内：_Encapsulation_Custom\Engine\Yi.Optimiz.Private

目录

前言

感谢您使用由 Yi 开发的解决方案，从开发到公开发行，耗时了 4 年多时间，在这 4 年里，一个人独自开发、解决疑难杂症等问题，非常艰难，还有漫长的挫折和煎熬，即便是这样，未来，还有很多工作要做。

Yi’s Solutions 的起源：

因自己习惯使用首选英文，但是又想有中文语言包，随时可切换不同的语言。当初只想做一个最简单的“Windows 10 多语言版：United States - English, 中国 - 简体中文”，一个版本里集成 2 种语言。

你想问的：不知道安装中文版后添加英文语言包？或安装英文版后添加中文语言包？

答：我要的是安装完成后一次拥有多种不同的语言，而不是安装系统后手动添加。

当成功离线封装添加了多语后，这一晃仅是 4 年了，当回顾这 4 年来，得出的结论是：

- 该方法通用制作 Windows 11、Windows 10、Windows Server 2022、Windows Server vNext
- 在与微软官方提供的学习制作方法不同，不遵守由官方提供的方案，既来之，则安之。

制作方法千变万化，如同兵者，诡道也（出自：孙子兵法）。

其中，还要不断的完善封装脚本、部署引擎、本地语言体验包（LXPs）下载器的开发工作，最难解决和开发的：

1. PowerShell

1.1. 批量处理已挂载和未挂载等任务，开发批量和解决批量的问题，足足耗时了几个月时间才完成，那一段时间经常看着屏幕发呆。

1.2. 图形界面，已超过 50 个以上，最难开发的有：

1.2.1.1. 映像来源，包含了不同的嵌入页面：挂载到、语言选择、详细信息、提取 ISO 文件、设置界面等开发，单页代码接近 8000 行，纯手工编写。

1.2.1.2. 解决方案生成，包含了：添加部署引擎，添加部署标记、软件包添加，不同的架构、添加 Office 安装包、添加字体等功能，代码接过 7400 行，部署到映像源里，安装到系统后还需要测试脚本。

最最艰难的是人工调整控件大小，调整一次耗时几天，不低于 5 次大型调整。

1.3. 自动驾驶

自动驾驶最难开发，相当于考古工程，足足考古了 3 个月才完成该功能。

2. 封装系统

2.1. 封装脚本需要解决添加中遇到的问题和批量的问题，找问题与修复问题真的艰难；

2.2. 添加 InBox Apps 应用程序后，需要在断网的情况下测试所有重新安装的应用程序，是否能正常运行；

2.3. 了解 InBox Apps 应用程序依赖性，创建封装脚本规则等。

2.4. 封装完成后测试是否有新的问题存在等；

3. 文档编写

3.1. 8 个封装教程编写及修正，代码测试，排版，累计 400 页、超过 3 万字审核等。

章节 1　部分组成介绍

A.　封装教程

提供了不同的版本：有完整版本、精简版，提供的格式：.Docx 文档格式，.Pdf 文档格式，版本区别：

1. 完整版本，无删减内容；
2. 精简版，不包含：报告、注意事项等；

可前往封装之旅的教程有，

可选语言版本：简体中文版、美国英文版（Google 翻译：中文译英文），下载完整包可获得所有文档：[压缩包]:_Learn\Packaging.tutorial，或前往 https://github.com/ilikeyi/solutions/tree/main/_Learn/Packaging.tutorial 后选择。

B.　视频教程

1.　封装教程

1.1.　Windows 11 23H2：实战封装教程

- Youtube | https://youtu.be/e6mzybgMHF0
- 哔哩哔哩 | https://www.bilibili.com/video/BV1sj421R7uj/
- 腾讯视频 | https://v.qq.com/x/page/j3543gs3pv7.html
- 西瓜视频 | https://www.ixigua.com/7348909159569424946?utm_source=Readme

2.　自定义封装事件

2.1.　Windows 11 23H2：自定义封装事件

- Youtube | https://youtu.be/L1AxyoAhNMY
- 哔哩哔哩 | https://www.bilibili.com/video/BV1HK421e7AV/
- 腾讯视频 | https://v.qq.com/x/page/c3543pyggh0.html
- 西瓜视频 | https://www.ixigua.com/7348904251847868966?utm_source=Readme

3.　自动驾驶

3.1.　Windows 11 23H2：自动驾驶封装

- Youtube | https://youtu.be/BbS_T2d9lfc
- 哔哩哔哩 | https://www.bilibili.com/video/BV1Mt421G7Uf/
- 腾讯视频 | https://v.qq.com/x/page/l35436bsird.html

- 西瓜视频 | https://www.ixigua.com/7348896802180956683?utm_source=Readme

C. 封装脚本

1. 封装脚本主要功能

1.1. 检查更新：为更好的保持至最新版本，可随时检查是否有最新版可用

1.2. 热刷新：更改脚本后，在主界面里输入 R 后执行“重新加载模块”即可完成热刷新

1.3. 语言包：United States - English、中文（简体）、中文（繁体）、대한민국 - 한국어、日本 - 日本語

1.4. 事件模式

1.4.1. 自动驾驶

1.4.1.1. 先决条件

1.4.1.2. 导入公共库

可设置不同的规则：累积更新、驱动

1.4.1.3. 其它：从不同的配置文件里导入，可自定义选择导入项、关联 ISO 方案等

1.4.2. 自定义分配事件

分配时，可自定义选择待分配的项

1.4.3. 手动操作

1.4.3.1. 操作时可全部中断、仅中断当前任务

1.4.3.2. 感知功能

1.4.3.3. 添加语言包，连招：添加语言包、添加累积更新、保存已挂载、生成 ISO

1.4.3.4. 添加累积更新，连招：添加累积更新、保存已挂载、生成 ISO

这就是感知功能，可在设置界面“指定全局感知”或“自定义当前感知”功能。如何自定义感知选项，请参阅：封装脚本开发指南。

1.5. 降序：自动识别 ARM64、x64、x86 架构，根据架构自动降序选择依赖性程序

1.6. ISO：自动识别 ISO 标签名并初始化规则（支持包含类匹配）、解压、挂载、弹出、校验哈稀、按规则显示对应 ISO 文件、搜索，自动分类：文件、语言包、功能包、InBox Apps

1.7. 修复

1.7.1. 删除保存在注册表里的 DISM 挂载记录

1.7.2. 删除与已损坏的已装载映像关联的所有资源

1.7.3. 添加路由功能后，可运行：Yi -Fix，或在设置界面里选择。

1.8. 挂载点

1.8.1. 可自定义指定挂载到

1.8.2. 自动搜索本地所有磁盘，自动选择卷标名：RAMDISK 的磁盘，可修改初始卷标名，默认已启用该功能

2. 面向于映像源主要功能

面向于封装 Windows 操作系统的主要功能，支持批量操作主要项、扩展项。

2.1. 事件

比如操作 WinRE.wim 时，需要挂载 Install.wim 后才能再挂载 WinRe.wim，才可以执行针对 WinRE 的相应任务。

什么是映像内的文件？例如 Install.wim 里包含了 WinRE.wim 文件，挂载 install.wim 后，可以分配事件去处理 WinRe.wim。

主要功能：可分配已挂载或未挂载事件，主要触发事件，可分配：

- 主要项：Boot.wim
- 主要项：Install.wim，映像内的文件（扩展项）：WinRE.wim

2.2. 事件处理

事件处理分为几种方案：无需挂载映像、需要挂载映像后才能操作项方式，支持主映像、映像内批量处理。

2.2.1. 无需挂载映像

2.2.1.1. 添加、删除、更新映像内的文件、提取、重建、应用

2.2.1.2. 提取语言包

2.2.1.3. 互转 Esd、Wim

2.2.1.4. 拆分 Install.wim 为 Install.swm

2.2.1.5. 合并 install.swm 到 install.wim

2.2.1.6. 生成 ISO

- 同步本地所有已知语言

- 生成标记：多语标签、单语标签、计算映像版本、计算已安装语言、发行年月、版本代码
- 自定义：ISO 卷标名、ISO 文件名、指定保存到

2.2.2. 需要挂载映像后才能操作项

2.2.2.1. 语言包

添加语言、反向删除语言、更改映像默认语言、清理已过时的组件。

- 安装时：按区域标记自动归类语言包、功能包，自动从映像中已安装的所有包匹配。
- 提取：按规则提取语言包，自定义选择语言标标记，已分类已知关联；
- 语言包：同步到 ISO 安装程序
- 重新生成 Lang.ini
- 累积更新：安装语言包后必须添加累积更新（可安装初始版本相同版本号或最新的累积更新），因为未添加累积更新之前，组件不会有任何变化，至到你安装累积更新后才会发生新的变化，例如组件状态：已过时、待删除；

2.2.2.2. 本地语言体验包（LXPs）

标记、添加、删除、按匹配规则删除

2.2.2.3. InBox Apps

标记、添加、删除、按匹配规则删除、可无限自定义不同的映像版本预安装应用程序等，详见：封装脚本开发指南。

2.2.2.3.1. 第一步：安装本地语言体验包（LXPs），区域标记

- InBox Apps 应用程序想获得多语言包，有二种方法：一是添加语言包；二是添加本地语言体验包（LXPs）来进行标记。
- 二种方法添加后，脱机映像语言会有新增，添加 InBox Apps 应用程序根据脱机映像已安装语言来进行匹配；这就是所谓的区域标记功能。

2.2.2.3.2. 第二步：自定义添加 InBox Apps 应用程序，添加前可检查依赖性

2.2.2.3.3. 第三步：清理本地语言体验包（LXPs）

2.2.2.3.4. 挂载后：可管理已安装项并删除

批量或未挂载时，可模糊按应用程序名称匹配到后并删除

2.2.2.4. 累积更新

语言包、本地语言体验包（LXPs）、InBox Apps、累积更新是连招、组合拳。

2.2.2.5. 驱动

添加、删除

2.2.2.6. Windows 功能

启用、禁用、支持已挂载后处理启用、禁用

2.2.2.7. 运行 PowerShell 函数

- 什么是函数？可以创建自定义函数，写入自定义代码，可以获取可用的 PowerShell 内的所有变量名、全局参数等。
- 可分配：有任务前运行 PowerShell 函数，运行任务完成后运行 PowerShell 函数

2.2.2.8. 解决方案：生成

可生成到映像源里、已挂载脱机项里，可生成：部署引擎、应预答，软件包，自定义合集包、添加 Microsoft Office 安装包等，部署软件时支持 arm64、 x64、 x86 降序添加，支持单语言、多语言部署

2.2.2.8.1. 部署引擎

2.2.2.8.1.1. 首次体验，部署先决条件过程中

- 允许全盘搜索并同步部署标记
- 允许自动更新
- 添加主目录到 Defend 排除目录
- 禁用网络位置向导
- 系统盘卷标：主目录名相同
- 遇到多语言时
  - 阻止 Appx 清理维护任务
  - 阻止清理未使用的按需功能语言包
  - 阻止清理未使用的语言包
- 添加个性化“上下文菜单”

- 更改系统区域设置

2.2.2.8.1.2. 首次体验，完成先决条件后

- 弹出部署引擎主界面
- 允许首次预体验，按计划
- 恢复 Powershell 执行策略：受限
- 删除整个解决方案
- 删除部署引擎，保留其它

2.2.2.8.2. 应预答

2.2.2.8.2.1. 可选预置架构核心版本：11、10

2.2.2.8.2.2. 指定应预答部署时命令

2.2.2.8.2.3. 指定单语言、多语言

2.2.2.8.2.4. 指定 Autounattend.xml 方案

- 半自动对所有安装方式有效
- EFI 自动安装
- egacy 自动安装

2.2.2.8.2.5. 安装界面：隐藏产品密钥、隐藏选择要安装的操作系统、隐藏接受许可条款

2.2.2.8.2.6. 服务器版本时：

- 登录时不自动启动服务器管理器
- Internet Explorer 增强的安全配置：关闭管理员、关闭用户

2.2.2.8.2.7. 指定时间区域

2.2.2.8.3. 添加合集

2.2.2.8.3.1. 选择首选架构包后，自动降序添加

2.2.2.8.3.2. 选择部署 Microsoft Office 安装包：指定语言、指定添加到、可选版本：Office 365、 Office 2024、 Office 2021、 Office 2019

2.2.2.8.3.3. 添加软件包

2.2.2.8.3.4. 添加字体

2.2.2.9. 生成报告

可生成：健康状态、已安装的应用程序包、脱机已安装语言、已安装 InBox Apps 应用、驱动

2.2.2.10. 弹出

弹出主要功能：保存、不保存，支持弹出扩展项，弹出后使用 WimLib 更新映像内的文件。

章节 2 开启封装之旅

A. 先决条件

I. 要求

PowerShell 版本

- PowerShell 5.1

  需要 Windows 11、Windows 10、Windows Server 2022、Windows Server vNext 或系统默认自带的 5.1 版本，可选升级最新版 PowerShell 7。

- PowerShell 7

  获取最新版，前往 https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows 后，选择需要下载的版本，下载后并安装。

II. 命令行

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”，设置 PowerShell 执行策略：绕过，PS 命令行：

   ```
   Set-ExecutionPolicy -ExecutionPolicy Bypass -Force
   ```

3. 在本文中，绿色部分属于 PS 命令行，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 ”或“PowerShell ISE”里运行，带冒号的路径，在命令行添加 & 字符，示例：

   ```
   & "D:\YiSolutions\_Encapsulation\_SIP.ps1"
   ```

III. 获取 Yi’s Solutions

1. 官方网站

1.1. 自动下载

Set-ExecutionPolicy -ExecutionPolicy Bypass -Force

irm https://fengyi.tel/gs | iex

优先从官方网站下载，下载完成后：添加路由功能。运行封装脚本。

1.2. 手动下载

前往 https://fengyi.tel/solutions 后查看下载项，或打开 https://fengyi.tel/go/solutions 后直接下载。

2. Gihtub

2.1. 自动下载

Set-ExecutionPolicy -ExecutionPolicy Bypass -Force

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

优先从 Github 网站下载，下载完成后：添加路由功能，运行封装脚本。

2.2. 手动下载

前往 https://github.com/ilikeyi/solutions 后选择“代码”，再选择下载 ZIP。

或前往 https://github.com/ilikeyi/solutions/releases 后，选择需要下载的可用版本，点击下载源代码（zip、tar.gz）。

IV. PowerShell 脚本

1. 封装脚本

D:\YiSolutions_Encapsulation_SIP.ps1

进入封装脚本主界面后，您可将路由功能添加到系统变量，添加后，下次在 PowerShell 终端里运行 Yi 进入引导界面，或输入 Yi -sip 直接进入封装界面，无需再输入脚本完整路径才可运行。

2. 其它项

2.1. 备份

D:\YiSolutions_Encapsulation_Unpack.ps1，路由功能可用时：Yi -unpack

在封装脚本执行检查更新时，可将备份好的文件作为升级包使用。

2.2. 创建部署引擎升级包

D:\YiSolutions_Encapsulation_Create.Custom.Engine.upgrade.package.ps1，路由功能可用时：Yi -CEUP

2.3. 将所有软件转换为压缩包

D:\YiSolutions_Encapsulation_Zip.ps1，路由功能可用时：Yi -Zip

2.4. 创建模板：累积更新

D:\YiSolutions_Encapsulation_Create.Template.ps1，路由功能可用时：Yi -CT

B. 部署软件

1. Microsoft Office

需要部署 ODT 版本的 Microsoft Office 2024、Microsoft Office 2021、Microsoft Office 2019、Microsoft 2016 时，请根据下载的架构版本下载，例如下载：

1.1. 下载 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. 下载 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. 自定义软件包

生成解决方案时，指定了区域标记时，将自动按匹配到的软件包进行复制。未发现时，en-US 作为默认复制，没有相对应的架构请创建 x86 为默认，生成时，按降序顺序添加：arm64、x64、x86。

2.1. 自定义软件包：创建

2.1.1. 7zip

7zip 已知可用语言：一个安装包已包含了多语言，有不同的架构版本：arm64、x64、x86

2.1.1.1. en-US

创建目录：架构 \ en-US 后，复制应用程序包到该目录

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

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

x86: D:\YiSolutions_Encapsulation_Custom\Software\00\7z\x86\en-US，安装包：7z2301.ex

2.1.2. WinRAR

WinRAR 语言仅单语，根据不同的语言区域来进行划分，有不同的架构版本：x64、x86

2.1.2.1. en-US

创建目录：架构 \ en-US 后，复制应用程序包到该目录

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

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

2.1.2.2. zh-CN

创建目录：架构 \ en-US 后，复制应用程序包到该目录

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

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

2.1.2.3. 其它未列出，制作时请参阅以上目录结构并创建，其它版本请参阅官方网站

2.2. 自定义软件：转换为压缩包

转换为 zip 后，将缩小文件体积，Yi's 优化脚本已包含了：首次体验时，自动解压所有 zip 压缩包。

3. 字体

可添加不同的字体到：D:\YiSolutions_Encapsulation_Custom\Fonts，首次体验部署时，自动安装字体。

C. 虚拟内存盘

内存盘是什么？内存盘也被称为虚拟内存盘，它是一种可以提高电脑内存和文件快速访问的技术。但是内存盘会导致电脑在关闭之后会出现数据丢失，内存盘是比较不安全是一种设置。

虽然如此，但是我不这样认为，在封装过程中会频繁的释放安装包文件、生成日志等，挂载到虚拟盘时，这有很多好处，快速格式化。

1. 搭配建议

添加语言包、添加累积更新、添加 InBox Apps 时，安装包存放于内存虚拟盘里，这样会占用大量的内存，建议您存放于非虚拟内存盘。

2. 软件推荐

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. 如何创建

创建内存盘时应计算物理内存未使用率，打开“任务管理器”，“性能”，查看内存剩余率，建议：

3.1. 物理内存 16G，系统剩余 10G 时，建议划分：6G 内存 + 40G 交换文件，保留剩余内存 4G 以上；

3.2. 物理内存 32G：系统剩余 26G 时，建议划分：20 内存 + 40G 交换文件，保留剩余内存 6G 以上；

3.3. 物理内存 64G：系统剩余 54G 时，仅划分 40G 内存，无需创建交换文件，保留剩余内存 8G 以上；

3.4. 物理内存 128G：系统剩余 115G 时，划分 40 内存之间，无需创建交换文件。

注意：内存不足时会导致在封装过程中出现问题。

章节 3 封装脚本：开发人员指南

1. 必备工具

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

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

有些操作时，Visual Studio Code 完成不了时，使用 Sublime Text 以达到最好。可选

2. 导入项目到开发者工具

2.1. 安装 Visual Studio Code 并打开软件

2.2. 请先获取封装脚本后并解压到：D:\YiSolutions

2.3. 选择 Visual Studio Code 浏览，选择“打开目录”，选择 D:\YiSolutions 目录后并导入到项目列表

3. 自定义规则

创建自定义规则前，请参考包含和不包含 InBox Apps 应用程序规则，注意，在线更新后，启用的自定义规则不会同步到新版本里，更新完成后，请手动复制自定义规则到新的版本里，在线更新使用修复功能将重置所有文件。

3.1. 学习

3.1.1. 了解预置规则

3.1.1.1. 包含 InBox Apps

{压缩包}:_Encapsulation\Modules\1.0.0.0\Functions\Custom\Solutions.Custom.With.InBox.Apps.psm1

3.1.1.2. 不包含 InBox Apps

{压缩包}:_Encapsulation\Modules\1.0.0.0\Functions\Custom\Solutions.Custom.Only.Language.psm1

3.1.2. 从 InBox Apps 应用程序里查找依赖性

预规则 Solutions.Custom.With.InBox.Apps.psm1 配置文件里，所有应用程序依赖性，是从安装包清单里找到的，安装包里在不同的文件里查找，请先安装 7zip 软件，示例如何查找：

3.1.2.1. Microsoft.WindowsAlarms_8wekyb3d8bbwe.msixbundle

选择文件后，使用 7z 打开压缩包，找到 TimeUniversal_11.2304.0.0_x64.msix 并双击打开，找到 AppxManifest.xml，保存任意目录或双击打开，打开后：

查看 <Dependencies> 到 </Dependencies> 之间，详细内容：

```
<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>
```

所依赖框架和最低版本号：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

选择文件后，使用 7z 打开压缩包，找到 GetHelpApp_10.2201.421.0_x64.appx 并双击打开，找到 AppxManifest.xml，保存任意目录或双击打开，打开后：

查看 <Dependencies> 到 </Dependencies> 之间，详细内容：

```
<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>
```

所依赖框架和最低版本号：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. 其它

每款应用程序不同，所在的位置不同，有些在主应用包下，有些在不同的架构安装内。

3.2. 创建自定义规则

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

3.2.1. 编辑

使用工具编辑：Solutions.Custom.Extension.psm1

3.2.2. 重命名

重命名 Solutions.Custom.Extension.psd1.Template 为 Solutions.Custom.Extension.psd1，删除 .Template。

3.2.3. 验证

运行封装脚本后验证是否有错误项。

3.3. 注意事项

重新安装 InBox Apps 应用程序后：必须进行安装测试；必须在断网的情况下测试所有已知 InBox Apps 应用程序是否能正常打开。

在制作 Windows 11 23H2 时，在先决条件里已列出需修复的项。由于微软官方提供的 InBox Apps 安装包：

1. 缺斤少两
2. 提供的应用程序损坏等

4. 常规开发

4.1. 快速定位到“不再处理映像源”

开发技巧，比如快速定位到“不再处理映像源”，搜索文字后，搜索结果：

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

2,2: AssignSkip = 不再处理映像源

复制命名后，再次搜索：$lang.AssignSkip，搜索结果：

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

1976,20: Text = $lang.AssignSkip

这样就快速定位完成。

5. 协作

如果您在开发过程中遇到新的问题，请通过以下联系方式：

作者：Yi

网站：https://fengyi.tel

建议或反馈：https://github.com/ilikeyi/solutions/issues

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

即时通讯

- QQ：775159955
- 微信：FengYi