

1. WinPE 的制作

WinPE 本身是一个最小的操作系统，有 2 个主要的用途：

- 在修复故障 Windows 的时候充当中间介质角色。
- 用来部署和安装操作系统。

WinPE 属于 RAM Disk，在 WinPE 启动后，对 WinPE 其中内容进行的改动会在下一次重启时还原。

更多信息可以参考：

[Windows PE \(WinPE\) | Microsoft Learn](#)

1.1 ADK 版本的选择

WinPE 的制作基于 Windows ADK 中的部署工具以及附带 WinPE 的文件。需要注意在 Windows 10, 1809 之前的 ADK 版本，ADK 和 WinPE 是集合在一个安装包中的。而较新的 ADK 版本则将 ADK 与 WinPE add-on for ADK 分开为 2 个不同的安装包。

在作为修复故障虚拟机的用途时，不同版本的之间的 WinPE 没有太大的区别，新版本的 WinPE 会多一些命令行工具。如果环境中有多版本，推荐选择环境中最新版本的 Windows 所对应的 ADK 来制作 WinPE。

选择适合你方案的合适的 ADK

如果可能，请使用与你正在使用的 Windows 版本相匹配的 ADK 版本。如果你的环境混合使用不同的 Windows 版本，请使用与环境中的最新操作系统匹配的 ADK 版本。

由于当前环境中最高的 Windows 版本为 Windows Server 2022，所以推荐选择 Windows Server 2022 的 ADK，本文中的大部分内容也是基于该 ADK 版本为基础。

更多信息的可以参考：

[Download and install the Windows ADK | Microsoft Learn](#)

1.2 ADK 的安装

a. 下载 Windows ADK 工具以及 WinPE add-on for ADK

联网下载安装：

[下载并安装 Windows ADK | Microsoft Docs](#)

分别点击以下 2 个链接进行下载 Windows Server 2022 版本的 ADK 以及 WinPE add-on for ADK：

- 下载适用于 Windows Server 2022 的 ADK：
适用于 Windows Server 2022 的 ADK 工具中的新增功能
- [适用于 Windows Server 2022 的 ADK](#) [↗](#)
 - [Windows Server 2022 ADK 的 Windows PE 加载项](#) [↗](#)
- 注意：**评估工具包在此版本的 ADK 中不可用，即使运行安装程序时选项可用。

b. 分别安装，先 ADK (仅勾选部署工具即可)，后 Windows PE：



Select the features you want to install

Click a feature name for more information.

☒ Windows 预安装环境 (Windows PE)

Windows 预安装环境 (Windows PE)

Size: 5.8 GB

设计为准备计算机以进行 Windows 安装和服务的最精简操作系统。

包括:

- Windows PE (x86)
- Windows PE (AMD64)
- Windows PE (ARM)
- Windows PE (ARM64)

Estimated disk space required: 5.8 GB

Disk space available: 38.5 GB

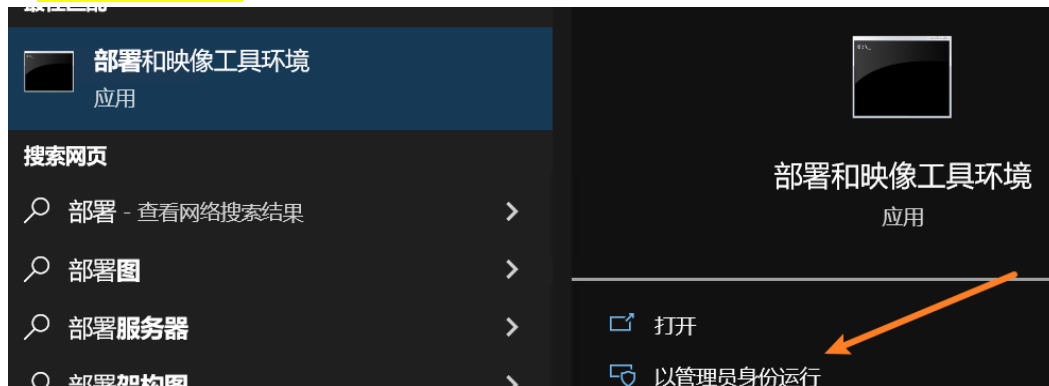
Back

 Install

Cancel

1.3 WinPE 的基本制作步骤

- a. 以管理员身份运行部署和映像工具环境：



- b. 在打开的窗口运行以下的命令，将 64 位的 PE 环境拷贝到 C:\WinPE_amd64：

```
copy /b amd64 C:\WinPE_amd64
```

```
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>copy /b amd64 C:\WinPE_amd64\
=====
Creating Windows PE customization working directory
C:\WinPE_amd64\
=====
```

- c. 在打开的部署和映像工具环境 CMD 窗口中通过以下的命令直接封装 ISO 文件：

```
MakeWinPEMedia /ISO C:\WinPE_amd64 C:\WinPE_amd64.iso
```

```
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>MakeWinPEMedia /ISO C:\WinPE_amd64
C:\WinPE_amd64.iso
Creating C:\WinPE_amd64.iso...
100% complete
Success
```

如果需要封装成 VHDX，可以用以下的 CMD 命令 (管理员权限) 先创建一个 VHDX：

```
diskpart
create vdisk file="C:\WinPE.vhdx" maximum=1000
attach vdisk
create partition primary
assign letter=V
format fs=ntfs quick
exit
```

```

C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>diskpart

Microsoft DiskPart 版本 10.0.19041.964

Copyright (C) Microsoft Corporation.
在计算机上: DESKTOP-SG7321T

DISKPART> create vdisk file="C:\WinPE.vhdx" maximum=1000

    100 百分比已完成

DiskPart 已成功创建虚拟磁盘文件。

DISKPART> attach vdisk

    100 百分比已完成

DiskPart 已成功连接虚拟磁盘文件。

DISKPART> create partition primary

DiskPart 成功地创建了指定分区。

DISKPART> assign letter=V

DiskPart 成功地分配了驱动器号或装载点。

DISKPART> format fs=ntfs quick

    100 百分比已完成

DiskPart 成功格式化该卷。

DISKPART> exit

退出 DiskPart...

```

然后在部署和映像工具环境 CMD 窗口中通过以下的命令将 WinPE 制作为 VHD:

```

MakeWinPEMedia /UFD C:\WinPE_amd64 V:
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>MakeWinPEMedia /UFD C:\WinPE_amd64 V:
WARNING, ALL DATA ON DISK DRIVE V: WILL BE LOST!
Proceed with Format [Y,N]?Y
Formatting V:...
Setting the boot code on V:...
Copying files to V:...
Success

```

最终，将 V 盘弹出：



1.4 WinPE 离线注入驱动文件

由于环境中的 Windows 是基于 KVM 的虚拟机，所以需要考虑为 WinPE 离线注入对应的驱动。**最重要的是**磁盘控制器的驱动，这是磁盘正常加载的前提，也是对问题机器修复的前提。**其次**需要考虑的是网卡驱动，网卡驱动可以使得 WinPE 网络联通，以访问一些共享路径进行文件拷贝动作。

因为离线安装驱动的 WinPE 内核版本是基于 Windows Server 2022，所以只需要为该版本的 WinPE 导入 Windows Server 2022 版本的磁盘控制器驱动以及网卡驱动。

参考步骤如下：

在我的电脑资源管理器中，打开以下的目录 C:\WinPE_amd64\media\sources，该目录下会有 boot.wim (WinPE 的本体)。将其 mount 起来，使用 DISM 离线打入驱动，然后再 unmount / commit 提交变更，具体步骤如下：

- a. 利用**管理员权限**新打开一个 CMD 窗口，
利用以下的命令将 C:\WinPE_amd64\media\sources\boot.wim 挂载到 C:\WinPE_amd64\mount 这个文件夹中

```
dism.exe /mount-image /imagefile:"C:\WinPE_amd64\media\sources\boot.wim" /index:1 /mountdir:"C:\WinPE_amd64\mount"
```



观察挂载后的目录，和平常操作系统的 C 盘目录结构相似：

电脑 > 本地磁盘 (C:) > WinPE_amd64 > mount >			
名称	修改日期	类型	
Program Files	2021/5/8 16:35	文件夹	
Program Files (x86)	2021/5/8 16:35	文件夹	
ProgramData	2021/5/8 16:35	文件夹	
Windows	2023/7/13 9:42	文件夹	
用户	2021/5/8 16:35	文件夹	

b. 通过 DISM 命令，离线安装驱动:

```
dism.exe /Image:"C:\WinPE_amd64\mount" /Add-Driver /Driver:"C:\drivers\mydriver.inf"
```

将上述标黄路径替换为实际的驱动路径，如下截图示例:

```
C:\Windows\system32>dism.exe /Image:"C:\WinPE_amd64\mount" /Add-Driver /Driver:"e:\NetKVM\2k22\amd64\netkvm.inf"
部署映像服务和管理工具
版本: 10.0.19041.844
映像版本: 10.0.20348.1
Found 1 driver package(s) to install.
Installing 1 of 1 - e:\NetKVM\2k22\amd64\netkvm.inf: The driver package was successfully installed.
操作成功完成。

C:\Windows\system32>dism.exe /Image:"C:\WinPE_amd64\mount" /Add-Driver /Driver:"e:\viosstor\2k22\amd64\viosstor.inf"
部署映像服务和管理工具
版本: 10.0.19041.844
映像版本: 10.0.20348.1
Found 1 driver package(s) to install.
Installing 1 of 1 - e:\viosstor\2k22\amd64\viosstor.inf: The driver package was successfully installed.
操作成功完成。
```

安装完成后，可以使用以下的命令查看安装的驱动:

```
dism.exe /Image:"C:\WinPE_amd64\mount" /Get-Drivers
```

```
C:\Windows\system32>dism.exe /Image:"C:\WinPE_amd64\mount" /Get-Drivers
部署映像服务和管理工具
版本: 10.0.19041.844
映像版本: 10.0.20348.1
Obtaining list of 3rd party drivers from the driver store...
Driver packages listing:
Published Name : oem0.inf
Original File Name : netkvm.inf ←
Inbox : No
Class Name : Net
Provider Name : Red Hat, Inc.
Date : 2021/12/2
Version : 100.90.104.21500

Published Name : oem1.inf
Original File Name : viosstor.inf ←
Inbox : No
Class Name : SCSIAdapter
Provider Name : Red Hat, Inc.
Date : 2021/12/2
Version : 100.90.104.21500
```

c. 提交变更，并 unmount-image

(建议操作前关闭所有的文件夹窗口，以避免报错文件被占用)

```
dism.exe /unmount-image /mountdir:"C:\WinPE_amd64\mount" /commit
C:\Windows\system32>dism.exe /unmount-image /mountdir:"C:\WinPE_amd64\mount" /commit

部署映像服务和管理工具
版本: 10.0.19041.844

正在保存映像
[=====100.0%=====]
正在卸载映像
[=====100.0%=====]
操作成功完成。
```

注：如果提示报错 0xc1420117，请关闭所有与 mount 路径相关的窗口以及应用，并重新运行上述命令，若报错变更为 0xc142011d，请运行以下的命令：

```
dism.exe /unmount-image /mountdir:"C:\WinPE_amd64\mount" /discard
```

参考 1.3 中的后面的 c 步骤利用 [MakeWinPEMedia](#) 将 WinPE 制作成 ISO 或者 VHDX。

1.5 WinPE 引导验证

根据以往的经验，某些高版本内核的 Windows 启动是需要一定的固件版本基础的 (在 VMware 平台上遇到过 Windows Server 2019 无法在低版本的固件上启动的案例的)，换一句话说，在该 WinPE 制作完成后，需要测试其能在不同版本 Windows Server 所在的固件平台上正确引导。引导进入 WinPE 后，需要确认能否识别 disk，网卡等相关信息。

1.5.1 查看磁盘状态

```
diskpart
list disk
list volume
```

```
X:\windows\system32>diskpart

Microsoft DiskPart version 10.0.20348.1

Copyright (C) Microsoft Corporation.
On computer: MININT-4ELPIOI

DISKPART> list disk

   Disk ###    Status         Size      Free      Dyn  Gpt
   -----    -
   Disk 0      Online            127 GB     0 B
   Disk 1      Online            127 GB     0 B

DISKPART> list volume

   Volume ###  Ltr  Label          Fs      Type          Size      Status       Info
   -----  -
   Volume 0     G    DVD_ROM        UDF     DVD-ROM        328 MB    Healthy
   Volume 1     C    System Rese    NTFS     Partition      549 MB    Healthy
   Volume 2     E    System Rese    NTFS     Partition      126 GB    Healthy
   Volume 3     D    System Rese    NTFS     Partition      549 MB    Healthy
   Volume 4     F    System Rese    NTFS     Partition      126 GB    Healthy
```

查询完成后，使用 exit 退出 diskpart 工具。

1.5.2 查看网卡状态

Ipconfig

```
X:\windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::89fe:5053:dc31:c049%3
    IPv4 Address. . . . . : 172.24.151.178
    Subnet Mask . . . . . : 255.255.240.0
    Default Gateway . . . . . : 172.24.144.1
```

1.5.3 查看设备状态

pnputil.exe /enum-devices

```
Instance ID:          ROOT\SYSTEM\0000
Device Description:    Plug and Play Software Device Enumerator
Class Name:            System
Class GUID:            {4d36e97d-e325-11ce-bfc1-08002be10318}
Manufacturer Name:    (Standard system devices)
Status:                Started
Driver Name:           swenum.inf
```

该命令用以查看 WinPE 中设备加载的状态。

1.5.4 查看驱动状态

pnputil.exe /enum-drivers

```
X:\windows\system32>pnputil /enum-drivers
Microsoft PnP Utility

Published Name:        oem0.inf
Original Name:         viostor.inf
Provider Name:         Red Hat, Inc.
Class Name:            Storage controllers
Class GUID:            {4d36e97b-e325-11ce-bfc1-08002be10318}
Driver Version:        12/02/2021 100.90.104.21500
Signer Name:           Microsoft Windows Hardware Compatibility Publisher

Published Name:        oem1.inf
Original Name:         netkvm.inf
Provider Name:         Red Hat, Inc.
Class Name:            Network adapters
Class GUID:            {4d36e972-e325-11ce-bfc1-08002be10318}
Driver Version:        12/02/2021 100.90.104.21500
Signer Name:           Microsoft Windows Hardware Compatibility Publisher
```

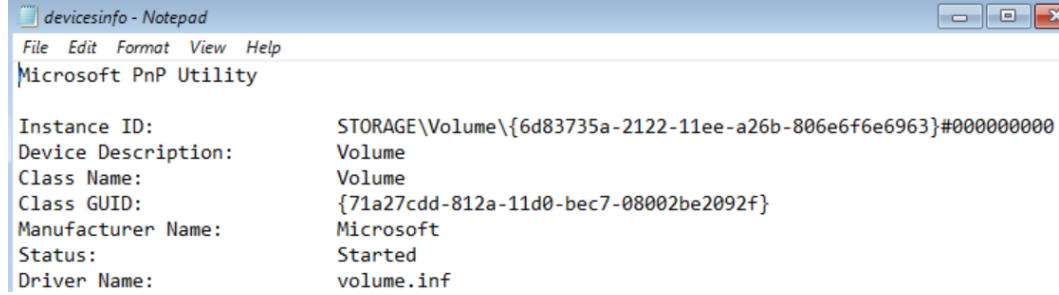
该命令用以查看 WinPE 中安装的驱动信息。

使用上述的命令查看信息时，可以将信息导出为 txt 文件，然后使用 notepad 打开查看或者是拷贝到其他地方进行查看，如下示例：

```
pnputil.exe /enum-devices > X:\Windows\System32\devicesinfo.txt  
notepad.exe X:\Windows\System32\devicesinfo.txt
```

```
X:\windows\system32>pnputil.exe /enum-devices > X:\Windows\System32\devicesinfo.txt
```

```
X:\windows\system32>notepad.exe X:\Windows\System32\devicesinfo.txt
```



devicesinfo - Notepad

File Edit Format View Help

Microsoft PnP Utility

Instance ID: STORAGE\Volume\{6d83735a-2122-11ee-a26b-806e6f6e6963}#000000000

Device Description: Volume

Class Name: Volume

Class GUID: {71a27cdd-812a-11d0-bec7-08002be2092f}

Manufacturer Name: Microsoft

Status: Started

Driver Name: volume.inf

2. WinPE 的定制

正如文档开头所述，WinPE 属于 RAM Disk，在 WinPE 启动后，对 WinPE 其中内容进行的改动会在下一次重启时还原。

对 WinPE 的永久性修改以及定制，需要通过离线挂载 boot.wim，对其进行自定义的变更。

2.1 为 WinPE 拷贝对应的文件

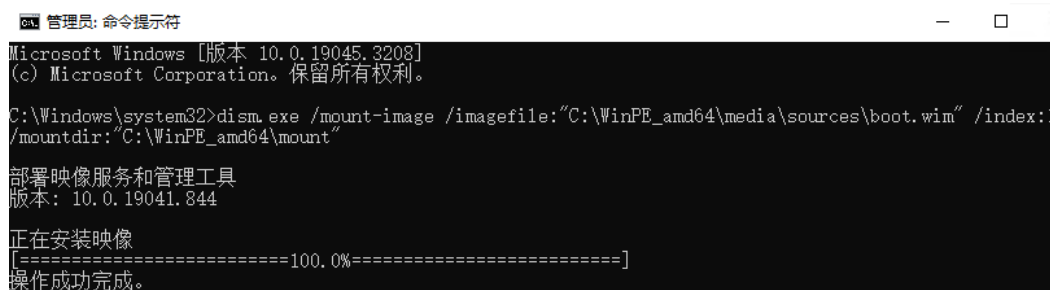
参考 1.4 章节中开始的步骤：

在我的电脑资源管理器中，打开以下的目录 `C:\WinPE_amd64\media\sources`，该目录下会有 `boot.wim` (WinPE 的本体)。将其 mount 起来，并将对应文件夹拷贝至挂载路径的 `Windows\System32` 或者是其他路径下，然后再 `unmount /commit` 提交变更，具体步骤如下：

a. **管理员权限**新打开一个 CMD 窗口，

利用以下的命令将 `C:\WinPE_amd64\media\sources\boot.wim` 挂载到 `C:\WinPE_amd64\mount` 这个文件夹中：

```
dism.exe /mount-image /imagefile:"C:\WinPE_amd64\media\sources\boot.wim" /index:1 /mountdir:"C:\WinPE_amd64\mount"
```



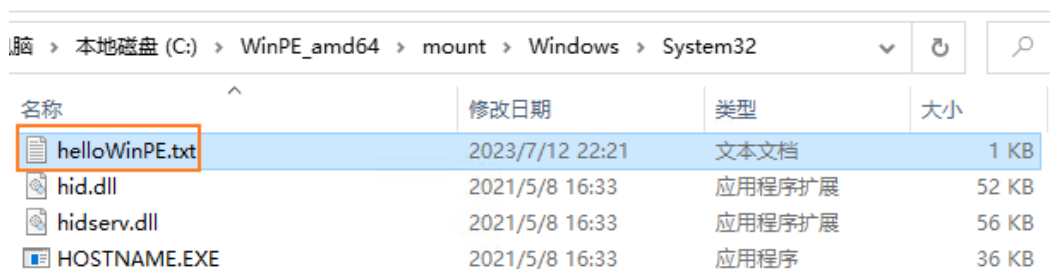
```
管理员: 命令提示符
Microsoft Windows [版本 10.0.19045.3208]
(c) Microsoft Corporation。保留所有权利。

C:\Windows\system32>dism.exe /mount-image /imagefile:"C:\WinPE_amd64\media\sources\boot.wim" /index:1 /mountdir:"C:\WinPE_amd64\mount"

部署映像服务和管理工具
版本: 10.0.19041.844

正在安装映像
[=====100.0%=====]
操作成功完成。
```

b. 将目标的文件/文件夹拷贝至 `C:\WinPE_amd64\mount\Windows\System32` 下 (或者是 `mount` 的路径下的其他文件夹中)：



电脑 > 本地磁盘 (C:) > WinPE_amd64 > mount > Windows > System32				
名称	修改日期	类型	大小	
helloWinPE.txt	2023/7/12 22:21	文本文档	1 KB	
hid.dll	2021/5/8 16:33	应用程序扩展	52 KB	
hidserv.dll	2021/5/8 16:33	应用程序扩展	56 KB	
HOSTNAME.EXE	2021/5/8 16:33	应用程序	36 KB	

c. 也可以对其中的一些文件进行更改和替换，

比如 WinPE 的桌面壁纸，位于挂载路径的 `Windows\System32` 下的 `winpe.jpg`：

« mount » Windows » System32

在 System32 中搜索

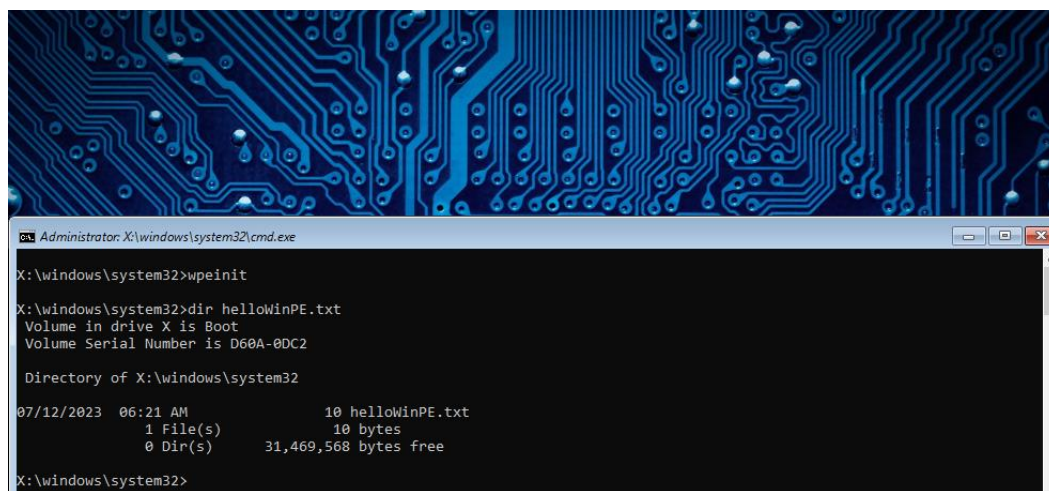
名称	修改日期	类型	大小
winntslui	2022/5/7 13:27	应用程序扩展	264 KB
winpe.jpg	2023/7/12 22:28	JPG 文件	426 KB
winpeshl.exe	2022/5/7 13:27	应用程序	52 KB
winre.jpg	2022/5/7 13:36	JPG 文件	1 KB
WinSCard.dll	2022/5/7 13:27	应用程序扩展	264 KB

拷贝替换完成上述文件后，先参考 1.4 中的 c 步将 boot.wim unmount 并保存更改。

最后参考 1.3 中的后面的 c 步骤利用 [MakeWinPEMedia](#) 将 WinPE 制作成 ISO 或者 VHDX。

测试效果：

新的 WinPE 启动后，会在 X:\Windows\System32 下找到添加的文件，同时 WinPE 壁纸也发生了变化。



上述的更改为永久性的更改。

2.2 为 WinPE 拷贝常用的工具

可以为 WinPE 中添加一些常用的实用工具，包括一些 UI 工具，这样可以大大提升使用体验。

添加的工具需要考虑以下几点：

1. 64 位编译的应用程序，WinPE 只支持 64 位/32 位其中一种位数应用的运行：
通常在制作 WinPE 时，选用的是 64 位的源文件，参考 1.3 章节的命令 `copyype amd64`。所以当前的 WinPE 是不支持 32 位应用运行的：

```
X:\Windows>cd "Process Monitor"

X:\Windows\Process Monitor>Procmon.exe
This version of X:\Windows\Process Monitor\Procmon.exe is not compatible with the version of Windows you're running. Check your computer's system information and then contact the software publisher.
```

2. 添加的工具应为免安装版本：
通过 2.1 章节中拷贝文件的方式，直接拷贝到 WinPE 对应目录中。当修改后的 WinPE 启动后，就可以直接运行对应应用。
3. 具体应用在 WinPE 中的运行兼容性仍然需要后续的测试验证。

以下为添加 2 个工具的示例步骤：

微软官方的工具 Process Monitor
[Process Monitor - Sysinternals | Microsoft Learn](#)

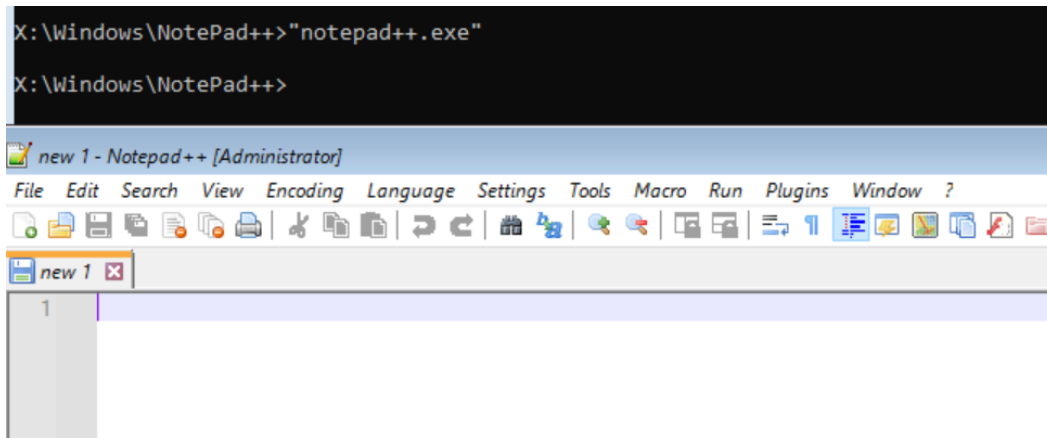
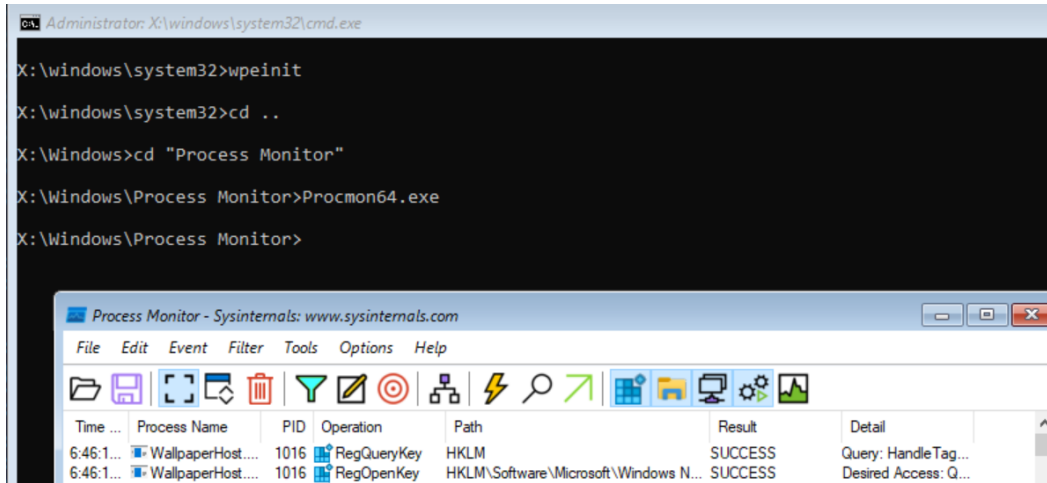
其他的工具，比如 Notepad ++

通过 2.1 章节中拷贝文件的方式，将对应文件的执行程序的路径拷贝到至挂载路径的 Windows 目录下：

脑 > 本地磁盘 (C:) > WinPE_amd64 > mount > Windows						
名称	修改日期	类型	大小			
system.ini	2021/5/8 16:34	配置设置	1 KB			
win.ini	2021/5/8 16:34	配置设置	1 KB			
WindowsShell.Manifest	2021/5/8 16:33	MANIFEST 文件	1 KB			
bfsvc.exe	2021/5/8 16:33	应用程序	96 KB			
regedit.exe	2021/5/8 16:33	应用程序	388 KB			
notepad.exe	2021/5/8 16:33	应用程序	220 KB			
INF	2023/7/15 12:40	文件夹				
NotePad++	2023/7/13 9:42	文件夹				
Process Monitor	2023/7/13 9:42	文件夹				

测试效果：

新的 WinPE 启动后，切换到 `X:\Windows` 中对应工具的目录，即可运行对应的应用程序：

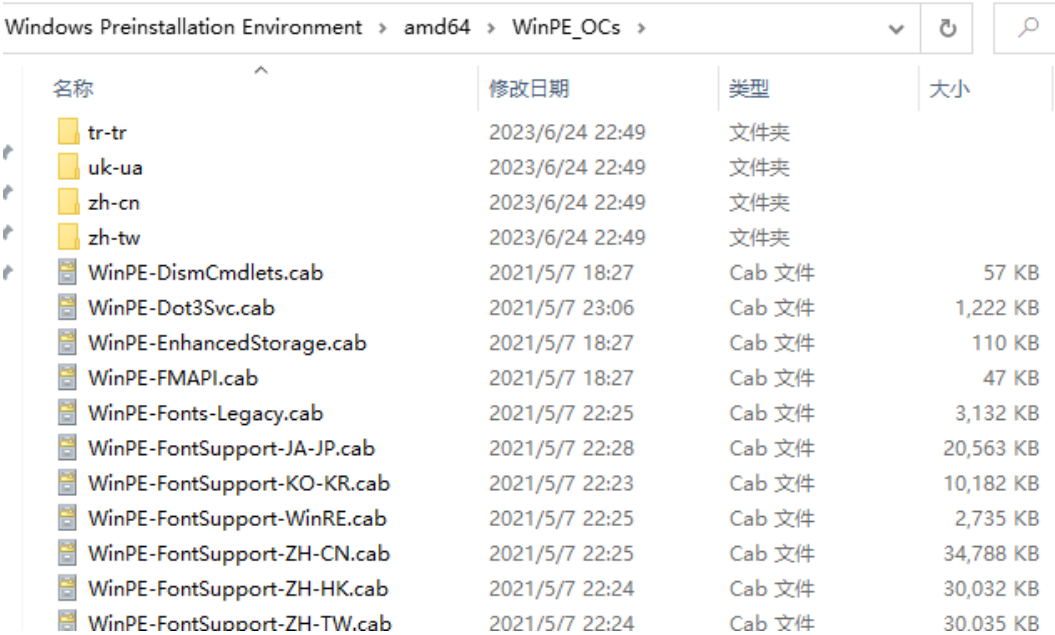


2.3 为 WinPE 添加系统组件

在上一章节介绍了将一些免安装的 64 位应用拷贝到至 WinPE 中直接运行的方法。这一章节会介绍另一种方式。

WinPE 自身也有一些官方制作好的.cab 文件，这些文件负责安装一些常见功能，比如 WMI，.NET Framework 等等。这些安装包位于 ADK WinPE 的目录：

C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs



Windows Preinstallation Environment > amd64 > WinPE_OCs >				
名称	修改日期	类型	大小	
tr-tr	2023/6/24 22:49	文件夹		
uk-ua	2023/6/24 22:49	文件夹		
zh-cn	2023/6/24 22:49	文件夹		
zh-tw	2023/6/24 22:49	文件夹		
WinPE-DismCmdlets.cab	2021/5/7 18:27	Cab 文件	57 KB	
WinPE-Dot3Svc.cab	2021/5/7 23:06	Cab 文件	1,222 KB	
WinPE-EnhancedStorage.cab	2021/5/7 18:27	Cab 文件	110 KB	
WinPE-FMAPI.cab	2021/5/7 18:27	Cab 文件	47 KB	
WinPE-Fonts-Legacy.cab	2021/5/7 22:25	Cab 文件	3,132 KB	
WinPE-FontSupport-JA-JP.cab	2021/5/7 22:28	Cab 文件	20,563 KB	
WinPE-FontSupport-KO-KR.cab	2021/5/7 22:23	Cab 文件	10,182 KB	
WinPE-FontSupport-WinRE.cab	2021/5/7 22:25	Cab 文件	2,735 KB	
WinPE-FontSupport-ZH-CN.cab	2021/5/7 22:25	Cab 文件	34,788 KB	
WinPE-FontSupport-ZH-HK.cab	2021/5/7 22:24	Cab 文件	30,032 KB	
WinPE-FontSupport-ZH-TW.cab	2021/5/7 22:24	Cab 文件	30.035 KB	

可以使用 `dism.exe /add-package` 的命令，将这些功能添加进入 WinPE，参考链接如下：

[Add or Remove Packages Offline Using DISM | Microsoft Learn](#)

2.3.1 为 WinPE 添加 PowerShell 模块

a. 参考通过 2.1 章节中的 a 步骤,

将 C:\WinPE_amd64\media\sources\boot.wim 挂载到 C:\WinPE_amd64\mount 这个文件夹中。

b. 使用以下的命令为 WinPE 离线安装 PowerShell 组件:

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\WinPE-WMI.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\en-us\WinPE-WMI_en-us.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\WinPE-NetFX.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\en-us\WinPE-NetFX_en-us.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\WinPE-Scripting.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\en-us\WinPE-Scripting_en-us.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\WinPE-PowerShell.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\en-us\WinPE-PowerShell_en-us.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\WinPE-StorageWMI.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\en-us\WinPE-StorageWMI_en-us.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\WinPE-DismCmdlets.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\en-us\WinPE-DismCmdlets_en-us.cab"
```

```
选择 管理员: 命令提示符
C:\Windows\system32>Dism /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\WinPE-DismCmdlets.cab"

部署映像服务和管理工具
版本: 10.0.19041.844

映像版本: 10.0.20348.1

Processing 1 of 1 - Adding package WinPE-DismCmdlets-Package~31bf3856ad364e35~amd64~10.0.20348.1
[=====100.0%=====]
操作成功完成。

C:\Windows\system32>Dism /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\en-us\WinPE-DismCmdlets_en-us.cab"

部署映像服务和管理工具
版本: 10.0.19041.844

映像版本: 10.0.20348.1

Processing 1 of 1 - Adding package WinPE-DismCmdlets-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1
[=====100.0%=====]
操作成功完成。
```

可以使用以下的命令查看之前安装的组件信息：

```
dism.exe /Image:"C:\WinPE_amd64\mount" /get-packages /format:table
C:\Windows\system32>dism.exe /Image:"C:\WinPE_amd64\mount" /get-packages /format:table

部署映像服务和管理工具
版本: 10.0.19041.844

映像版本: 10.0.20348.1

Packages listing:

-----
Package Identity                                     State      Release Type   Install Time
-----
Microsoft-Windows-WinPE-LanguagePack-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1  Installed  Language Pack  2021/5/8 8:52
Microsoft-Windows-WinPE-Package~31bf3856ad364e35~amd64~10.0.20348.1                    Installed  Foundation      2021/5/8 8:35
WinPE-DismCmdlets-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1                    Installed  Language Pack  2023/7/15 4:55
WinPE-DismCmdlets-Package~31bf3856ad364e35~amd64~10.0.20348.1                        Installed  Feature Pack   2023/7/15 4:54
WinPE-NetFx-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1                        Installed  Language Pack  2023/7/15 4:54
WinPE-NetFx-Package~31bf3856ad364e35~amd64~10.0.20348.1                            Installed  Feature Pack   2023/7/15 4:54
WinPE-PowerShell-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1                    Installed  Language Pack  2023/7/15 4:54
WinPE-PowerShell-Package~31bf3856ad364e35~amd64~10.0.20348.1                        Installed  Feature Pack   2023/7/15 4:54
WinPE-Scripting-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1                    Installed  Language Pack  2023/7/15 4:54
WinPE-Scripting-Package~31bf3856ad364e35~amd64~10.0.20348.1                        Installed  Feature Pack   2023/7/15 4:54
WinPE-StorageWMI-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1                    Installed  Language Pack  2023/7/15 4:54
WinPE-StorageWMI-Package~31bf3856ad364e35~amd64~10.0.20348.1                        Installed  Feature Pack   2023/7/15 4:54
WinPE-WMI-Package~31bf3856ad364e35~amd64~en-US~10.0.20348.1                         Installed  Language Pack  2023/7/15 4:53
WinPE-WMI-Package~31bf3856ad364e35~amd64~10.0.20348.1                             Installed  Feature Pack   2023/7/15 4:53

操作成功完成。
```

- c. 参考之前的步骤，先 unmount commit 更改，然后利用 [MakeWinPEMedia](#) 将 WinPE 制作成 ISO 或者 VHDX。

测试效果如下：

```
Administrator: X:\windows\system32\cmd.exe - powershell

X:\windows\system32>wpeinit

X:\windows\system32>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

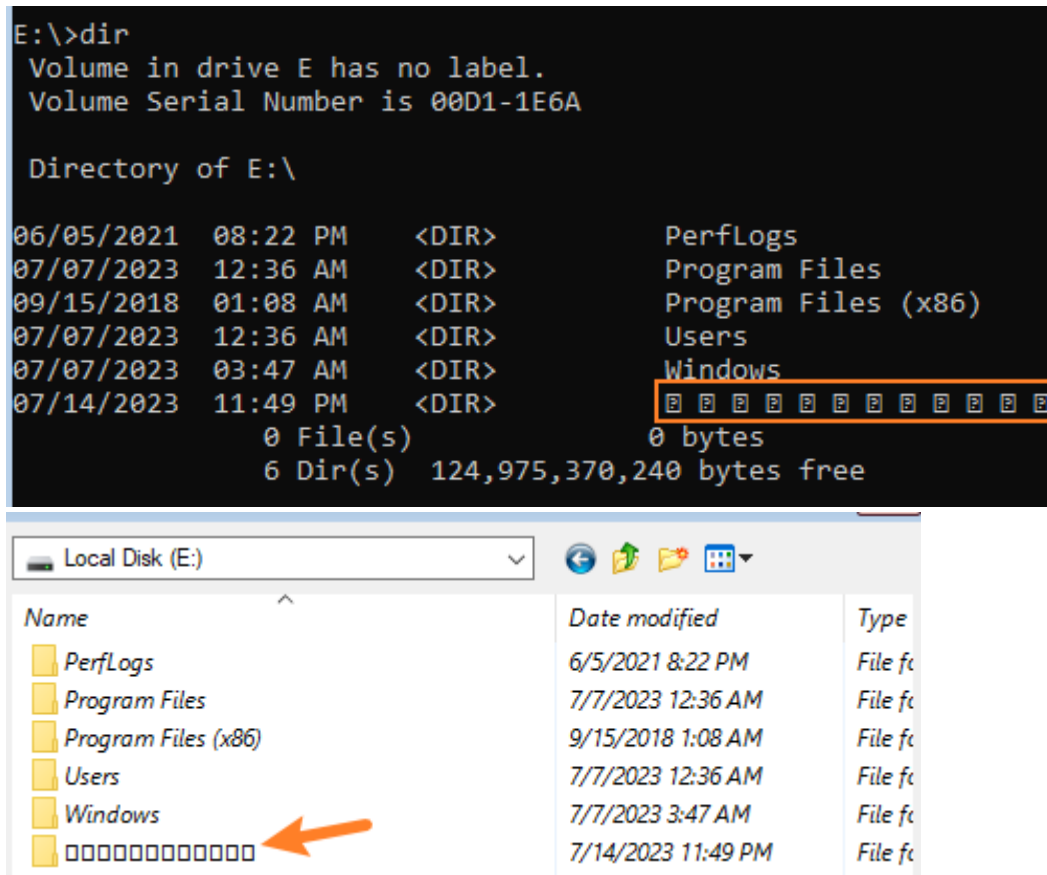
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS X:\windows\system32> get-process

Handles  NPM(K)    PM(K)      WS(K)      CPU(s)    Id  SI ProcessName
-----
71        6      4276      4164        0.00     700  1 cmd
143       10      1952     10976        0.08     620  1 conhost
142       10      1880      8784        0.06    1020  1 conhost
194       12      1804      5740        0.23     460  0 csrss
100       12      1024      5816        0.16     560  1 csrss
```

2.3.2 为 WinPE 添加中文支持

如果 WinPE 里面集成了一些中文工具或者包含中文字符的目录，可能会因为编码问题而显示异常：



可以通过为 WinPE 安装中文语言包，与 2.3.1 章节中的步骤类似，mount，unmount 以及制作成 ISO 的步骤这里不再赘述。

在 mount boot.wim 成功后，利用以下的命令安装中文的语言包：

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs\WinPE-FontSupport-ZH-CN.cab"
```

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs\zh-cn\lp.cab"
```

为 WinPE 离线设置 locale：

```
dism.exe /Image:"C:\WinPE_amd64\mount" /Set-UILang:zh-CN
dism.exe /Image:"C:\WinPE_amd64\mount" /Set-SysLocale:zh-CN
dism.exe /Image:"C:\WinPE_amd64\mount" /Set-UserLocale:zh-CN
dism.exe /Image:"C:\WinPE_amd64\mount" /Set-InputLocale:zh-CN
dism.exe /Image:"C:\WinPE_amd64\mount" /Set-TimeZone:"China Standard Time"
```

设置完成后，可以使用以下的命令，检查设置：

```
dism.exe /Image:"C:\WinPE_amd64\mount" /get-intl
C:\Windows\system32>dism.exe /Image:"C:\WinPE_amd64\mount" /get-intl

部署映像服务和管理工具
版本: 10.0.19041.844

映像版本: 10.0.20348.1

Reporting offline international settings.

Default system UI language : zh-CN
The UI language fallback is : en-US
System locale : zh-CN
Default time zone : China Standard Time
User locale for default user : zh-CN
Location : 中国 (GE0ID = 45)
Active keyboard(s) : 0804:[81D4E9C9-1D3B-41BC-9E6C-4B40BF79E35E] {FA550B04-5AD7-411F-A5AC-CA038EC515D7}, 0804:[6A498709-E00B-4C45-A018-8F9E4081AE40] {82590C13-F4DD-44F4-BA1D-8667246FDF8E}
Keyboard layered driver : PC/AT Enhanced Keyboard (101/102-Key)

Installed language(s): en-US
Type : Fully localized language.
Installed language(s): zh-CN
Type : Partially localized language, MUI type.
Fallback Languages en-US

操作成功完成。
```

注意，之前 PowerShell 组件只安装了英文的语言包，此时将对应组件的中文语言包补齐。

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\zh-cn\WinPE-WMI_zh-cn.cab"

dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\zh-cn\WinPE-NetFX_zh-cn.cab"

dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\zh-cn\WinPE-Scripting_zh-cn.cab"

dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\zh-cn\WinPE-PowerShell_zh-cn.cab"

dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\zh-cn\WinPE-StorageWMI_zh-cn.cab"

dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\zh-cn\WinPE-DismCmdlets_zh-cn.cab"
```

确保每个组件都有对应的语言包：

WinPE-DismCmdlets-Package_31bf3856ad364e35_amd64_en-US_10.0.20348.1	Installed	Language Pack	2023/7/15 4:55
WinPE-DismCmdlets-Package_31bf3856ad364e35_amd64_zh-CN_10.0.20348.1	Installed	Language Pack	2023/7/15 8:20
WinPE-DismCmdlets-Package_31bf3856ad364e35_amd64_10.0.20348.1	Installed	Feature Pack	2023/7/15 4:54

测试效果：

```
2023/07/15 17:08 <DIR> Program Files (x86)
2023/07/07 16:36 <DIR> Users
2023/07/07 19:47 <DIR> Windows
2023/07/15 15:49 <DIR> 管理是否允许允许百度网盘
```

2.3.3 为 WinPE 添加补丁更新

一般情况下，无需考虑为 WinPE 安装补丁。

但在安全漏洞的变更影响或者是因为一些已知问题 bug 的影响下，需要考虑为 WinPE 安装补丁包进行更新。

补丁的选择基于当前 WinPE 内核的版本，也就是基于制作 WinPE ADK 的版本。

在这篇文档中的 WinPE 制作基于 Windows Server 2022 的内核，即选择 Windows Server 2022 版本的相关更新。

以 Windows Server 2022 的累积更新为例：

[July 11, 2023—KB5028171 \(OS Build 20348.1850\) - Microsoft Support](#)

a. 先将下载后的补丁包 (.msu 后缀) 进行解压：

使用 Windows 自带的工具 expand，如下示例：

```
管理员: 命令提示符
Microsoft Windows [版本 10.0.19045.3208]
(c) Microsoft Corporation。保留所有权利。

C:\Windows\system32>cd /d d:\KB

d:\KB>expand -f:* d:\kb\windows10.0-kb5028171-x64_df2198a9a9ac5cf4b2d60af6b2c14d5902df0594.msu d:\kb\
Microsoft (R) 文件扩展实用程序
版权所有 (c) Microsoft Corporation。保留所有权利。

正在将 d:\kb\Windows10.0-KB5028171-x64.cab 添加到提取队列
正在将 d:\kb\Windows10.0-KB5028171-x64-pkgProperties.txt 添加到提取队列
正在将 d:\kb\Windows10.0-KB5028171-x64_uup.xml 添加到提取队列
正在将 d:\kb\SSU-20348.1846-x64.cab 添加到提取队列
正在将 d:\kb\WSUSSCAN.cab 添加到提取队列

正在展开文件 ...
进程: 0 超出 5 文件
完成展开文件 ...
总共 5 个文件。
```

名称	修改日期	类型	大小
windows10.0-kb5028171-x64_df2198a9a9ac5cf4b2d60af6b2c14d5902df0594.msu	2023/7/15 16:34	Microsoft 更新独...	350,141 KB
WSUSSCAN.cab	2023/7/7 19:43	Cab 文件	150 KB
Windows10.0-KB5028171-x64_uup.xml	2023/7/7 19:42	XML 文档	1 KB
Windows10.0-KB5028171-x64-pkgProperties.txt	2023/7/7 19:42	文本文档	1 KB
Windows10.0-KB5028171-x64.cab	2023/7/7 19:41	Cab 文件	333,827 KB
SSU-20348.1846-x64.cab	2023/7/7 19:33	Cab 文件	16,017 KB

b. 解压后，有 2 个包是需要离线安装进入 WinPE 的：

先安装 SSU-20348.1846-x64.cab (体积小的)，再安装 Windows10.0-KB5028171-x64.cab (体积大的)，使用指令和上一章节类似，**mount**，**unmount** 以及制作成 ISO 的步骤这里不再赘述：

```
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"D:\KB\SSU-20348.1846-x64.cab"
dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"D:\KB\Windows10.0-KB5028171-x64.cab"
```

运行效果如下图：

```
C:\Windows\system32>dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"D:\KB\SSU-20348.1846-x64.cab"
部署映像服务和管理工具
版本: 10.0.19041.844
映像版本: 10.0.20348.1
Processing 1 of 1 - Adding package Package_for_ServicingStack_1846~31bf3856ad364e35~amd64~~20348.1846.1.1
[=====100.0%=====]
操作成功完成。

C:\Windows\system32>dism.exe /Add-Package /Image:"C:\WinPE_amd64\mount" /PackagePath:"D:\KB\Windows10.0-KB5028171-x64.cab"
部署映像服务和管理工具
版本: 10.0.19041.844
映像版本: 10.0.20348.1
Processing 1 of 1 - 操作成功完成。
```

3. WinPE 的操作示例

3.1 快速确认原系统盘符

通常来说，使用 diskpart 指令来查看是最保险以及稳妥的方式，若要快速在 WinPE 中确认原系统盘符，可以通过以下方式：

即通过一个简单的 bat 脚本来判断，脚本示例内容如下：

```
@echo Find a drive that is origin C drive.  
@for %a in (C D E F G H I J K L M N O P Q R S T U V W X Y Z) do @if exist %a:\Windows\System32  
  \logman.exe set IMAGESDRIVE=%a  
@echo The Origin C drive in WinPE is: %IMAGESDRIVE%
```

脚本的逻辑也比较简单，就是遍历 26 个盘符的指定路径中的指定文件，该文件一定是原系统有，但 WinPE 没有的一个文件。

运行效果如下：

```
X:\Windows\System32>echo Find a drive that is origin C drive.  
Find a drive that is origin C drive.  
The Origin C drive in WinPE is: F
```

局限性：

一种典型的情况就是盘符变成了 RAW 格式，此时数据无法访问了，脚本自然就无法判断了。

该脚本局限性相对较小，可以考虑测试效果，然后集成进入 WinPE。相关步骤可以参考 2.1 章节。

3.2 通过 WinPE 修改原系统注册表

注册表本身是以 Hive 文件的形式存于操作系统中的，大部分关键注册表 Hive 文件位于以下的位置：

`C:\Windows\System32\Config`

对于 Windows Server，会存在 RegIdleBackup 这个计划任务 (位于 `Library\Microsoft\Windows\Registry`) 将这些重要注册表自动备份到 `C:\Windows\System32\Config\Backup` 文件夹中。

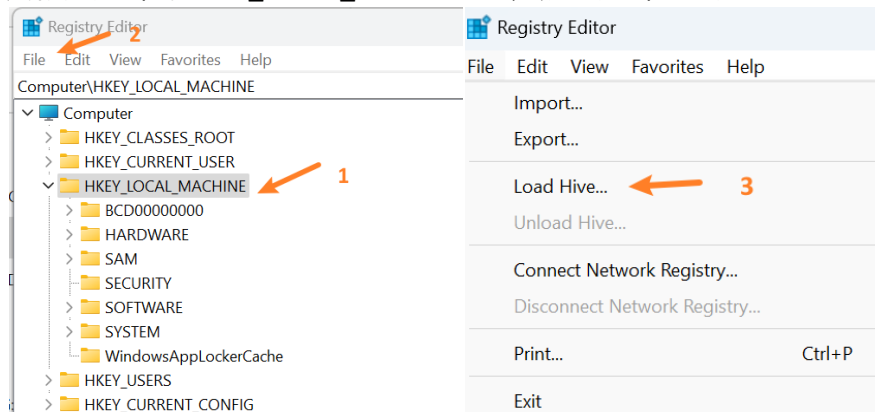
由于 WinPE 本身是一个操作系统，所以利用 `regedit.exe` 打开注册表编辑器看到的注册表项是 WinPE 自身的 Hive 文件。在 WinPE 中需要通过 Load Hive 的方式实现对原系统的注册表设置的变动。

参考步骤如下：

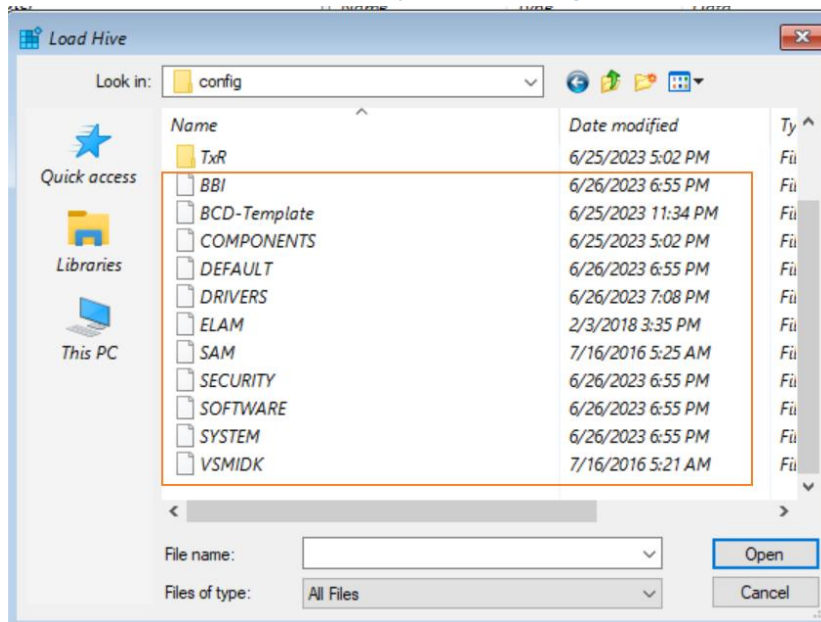
1. 引导进入 WinPE，在 CMD 窗口中运行 `regedit.exe`：

打开注册表编辑器，注意此时注册表编辑器中的内容均是 WinPE 自身相关信息。

2. 鼠标先点击到 `HKEY_LOCAL_MACHINE` 上，然后点击左上角 `File -> Load Hive...`：



选择原系统盘符下的 `Windows\System32\config` 下对应的 Hive 文件：



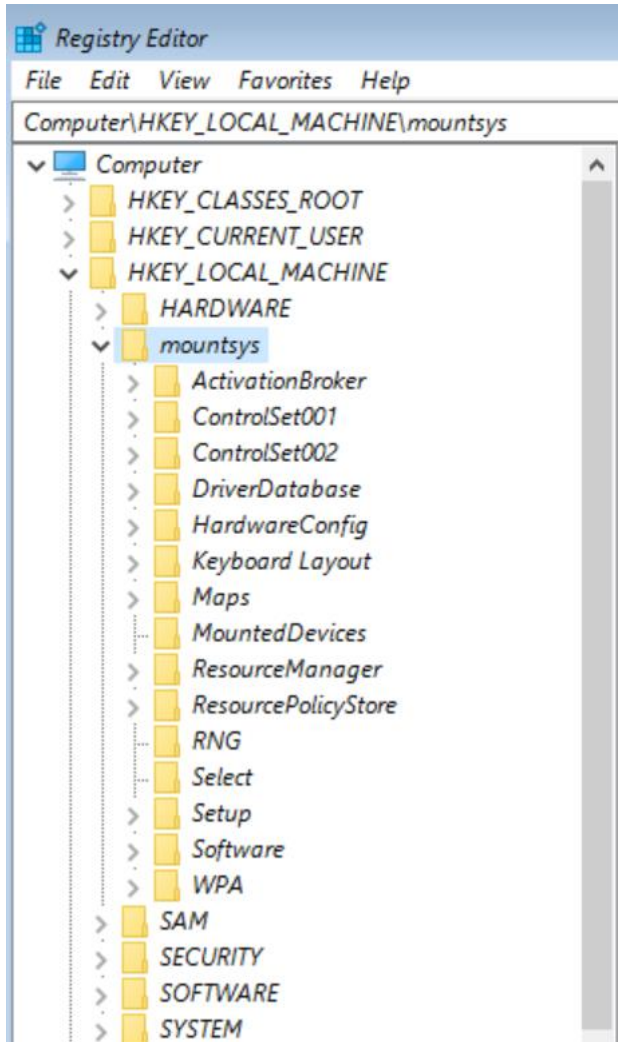
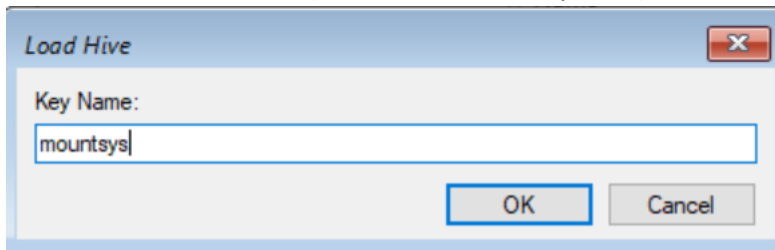
其中经常需要被挂载的就是 SOFTWARE, SYSTEM 以及 Security HIVE，它们分别对应原系统以下的注册表项：

`HKEY_LOCAL_MACHINE\SOFTWARE`

`HKEY_LOCAL_MACHINE\SYSTEM`

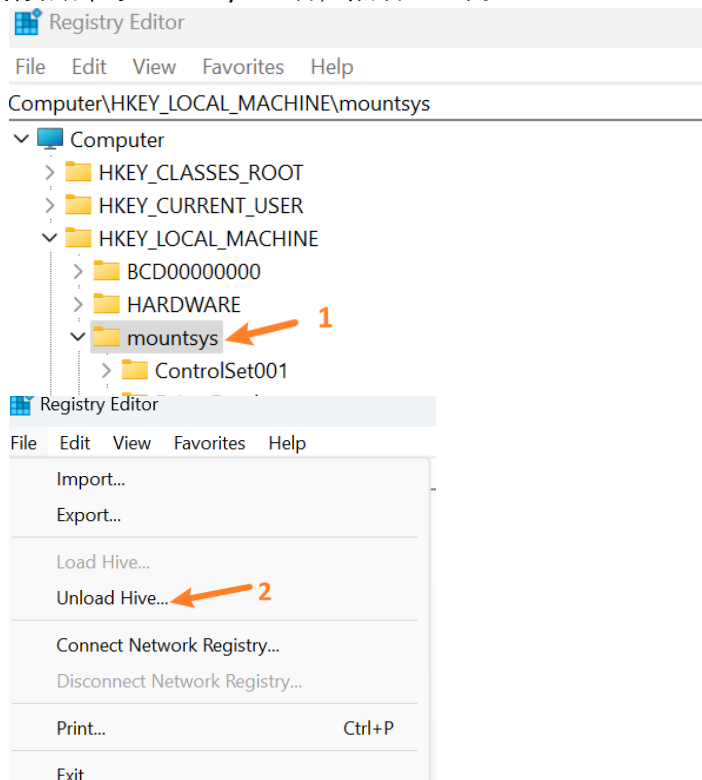
`HKEY_LOCAL_MACHINE\SECURITY`

3. 以挂载 SYSTEM Hive 为例,
可以给它随意取个名称 (但不要与已有的 Key 重名), 例如 mountsys:

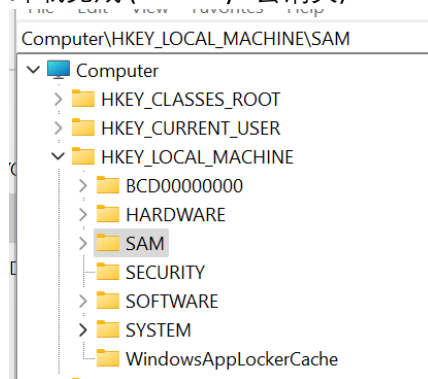


如上图, 这样就将原系统 `HKEY_LOCAL_MACHINE\SYSTEM` 挂载到 WinPE 下的注册表编辑器上了, 此时可以在 mountsys 下进行相应的改动。

4. 改动完成后，需要 **Unload Hive**，
鼠标点击到 mountsys 之后，然后左上角 File -> Unload Hive:



卸载完成 (mountsys 会消失):



若出现 ACCESS DENIED 的报错，可以尝试关闭重新打开 regedit.exe。

3.3 通过 WinPE 修改原系统的密码

关于重置密码，目标官方仅有以下的公开方案：

[Reset a local Windows password without Azure agent - Virtual Machines | Microsoft Learn](#)

请参考以下的具体步骤：

1. 准备三个文件，文件名以及内容如下：

文件名：

gpt.ini

内容：

```
[General]
gPCFunctionalityVersion=2
gPCMachineExtensionNames=[{42B5FAAE-6536-11D2-AE5A-0000F87571E3}{40B6664F-4972-11D1-A7CA-0000F87571E3}]
Version=1
```

文件名：

scripts.ini

内容：

```
[Startup]
0CmdLine=ResetVMPwd.cmd
0Parameters=
```

文件名：

ResetVMPwd.cmd

内容：

```
net user administrator Password01!
```

ResetVMPwd.cmd 中是真正执行的密码，这里预留的命令是修改内置管理员 Administrator 的密码为 Password01!，也可以参考原 KB 中的命令来新建一个管理员用户。

将这些文件拷贝到一个 ResetPwd 文件夹中：

ResetPwd					在 ResetPwd 中搜索	
	名称	修改日期	类型	大小		
	gpt.ini	2023/6/27 11:33	配置设置	1 KB		
	scripts.ini	2023/6/27 11:33	配置设置	1 KB		
	ResetVMPwd.cmd	2023/6/27 11:33	Windows 命令脚本	1 KB		

2. 将该文件夹离线拷贝到 WinPE 中，相关步骤可以参考 2.1 章节。

3. 测试效果：

a. 进入 WinPE，会在 X:\Windows\System32 下找到刚刚添加的对应文件夹：

```
X:\windows\system32>dir ResetPwd
Volume in drive X is Boot
Volume Serial Number is D60A-0DC2

Directory of X:\windows\system32\ResetPwd

06/26/2023  07:38 PM    <DIR>          .
06/26/2023  07:38 PM    <DIR>          ..
06/26/2023  07:33 PM                152 gpt.ini
06/26/2023  07:33 PM                34 ResetVMPwd.cmd
06/26/2023  07:33 PM                50 scripts.ini
                3 File(s)                236 bytes
                2 Dir(s)      533,716,992 bytes free
```

将 **gpt.ini** 拷贝至原系统盘符\Windows\System32\GroupPolicy\

注：若该目录已经存在 gpt.ini，请重命名原文件为 gpt.ini.bak。

将 **scripts.ini** 拷贝至原系统盘符\Windows\System32\GroupPolicy\Machine\Scripts\

注：若文件夹不存在请自行创建，若该目录已经存在 scripts.ini，请重命名原文件为 scripts.ini.bak。

将 **ResetVMPwd.cmd** 拷贝至原系统盘符的\Windows\System32\GroupPolicy\Machine\Scripts\Startup\

注：若文件夹不存在请自行创建。

命令示例参考如下：

```
X:\windows\system32>dir e:\Windows\System32\GroupPolicy
Volume in drive E has no label.
Volume Serial Number is 8A22-0480

Directory of e:\Windows\System32\GroupPolicy

06/26/2023  09:07 PM    <DIR>          .
05/15/2023  06:49 AM    <DIR>          ..
                0 File(s)                0 bytes
                2 Dir(s)  119,367,782,400 bytes free

X:\windows\system32>copy x:\Windows\System32\ResetPwd\gpt.ini e:\Windows\System32\GroupPolicy\
1 file(s) copied.

X:\windows\system32>md e:\Windows\System32\GroupPolicy\Machine\Scripts\

X:\windows\system32>copy x:\Windows\System32\ResetPwd\scripts.ini e:\Windows\System32\GroupPolicy\Machine\Scripts\
1 file(s) copied.

X:\windows\system32>md e:\Windows\System32\GroupPolicy\Machine\Scripts\Startup\

X:\windows\system32>copy x:\Windows\System32\ResetPwd\ResetVMPwd.cmd e:\Windows\System32\GroupPolicy\Machine\Scripts\Startup\
1 file(s) copied.
```

b. 重启进入正常系统，使用修改后的密码进行登录。

c. 登录成功后，手动删除

`C:\Windows\System32\GroupPolicy\Machine\Scripts\scripts.ini` (若有备份文件，将 `scripts.ini.bak` 改回原名 `scripts.ini`)

`C:\Windows\System32\GroupPolicy\Machine\Scripts\Startup\ResetVMPwd.cmd`

`C:\Windows\System32\GroupPolicy\gpt.ini` (若有备份文件，将 `gpt.ini.bak` 改回原名 `gpt.ini`)

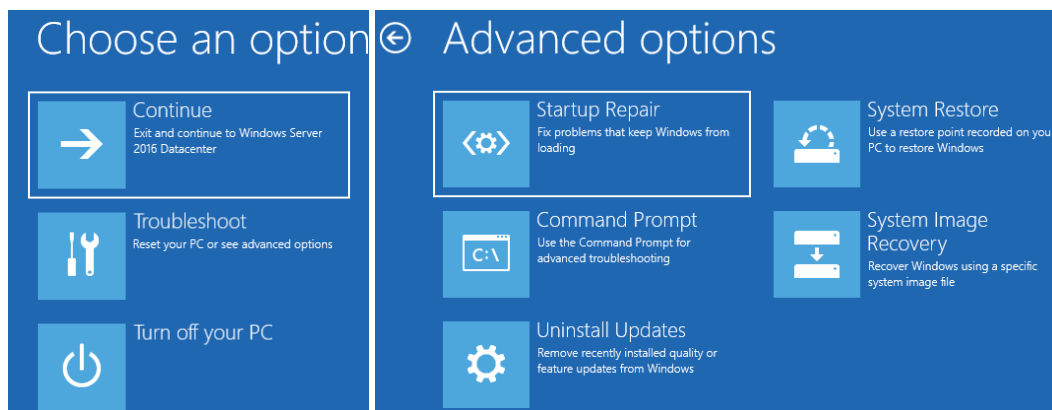
4. WinRE 与 WinPE

WinRE 是 Windows ISO 以及系统自带的 Windows Recovery 模式，它也是基于 WinPE 制作，在 WinPE 的基础上集成了一些 UI 界面以及一些工具。

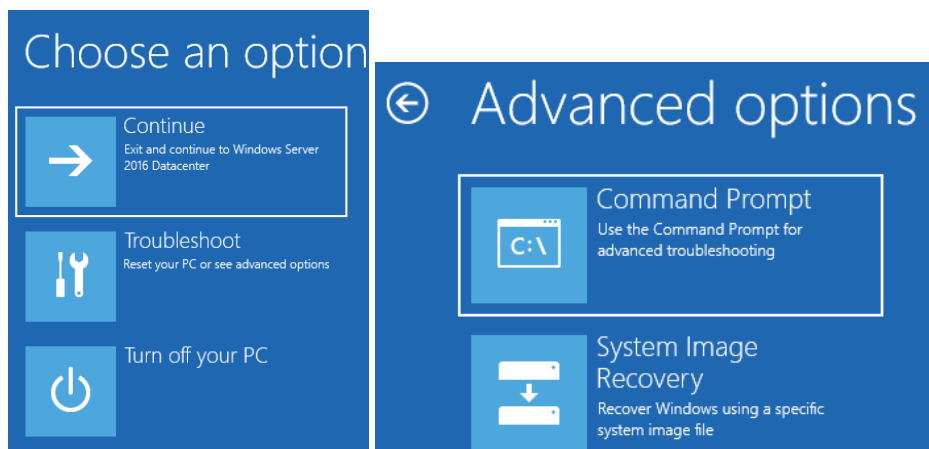
4.1 WinRE 说明

Windows Client (比如 Windows 10 Windows 11) 的 WinRE 相比 Windows Server 在 Advanced Options 中会多几个选项。

下图是 Windows 11 ISO 自带的 WinRE：



下图是 Windows Server 2022 自带的 WinRE：



4.2 集成 WinRE 功能

将上述功能界面集成到之前制作的 WinPE 中目前来看不太现实，WinPE 比 WinRE 少很多相关组件，没有现成的安装包，也并非通过拷贝一两个文件就能实现上述的 UI 显示。拿上述 UI 为例，需要从 WinRE 中拷贝以下的文件到 WinPE 才能实现正常实现功功能：

```
X:\windows\system32\bootrec.exe
X:\windows\system32\reagent.dll
X:\windows\system32\bootux.dll
X:\windows\system32\BootMenuUX.dll
X:\windows\system32\tbs.dll
X:\windows\system32\BCP47Langs.dll
X:\windows\system32\bmrui.exe
X:\windows\system32\spp.dll
X:\windows\system32\rstrui.exe
X:\windows\system32\reset*

X:\windows\system32\en-US\bootrec.exe.mui
X:\windows\system32\en-US\bootux.dll.mui
X:\windows\system32\en-US\reagent.dll.mui
X:\windows\system32\en-US\bmrui.exe.mui
X:\windows\system32\en-US\spp.dll.mui
X:\windows\system32\en-US\rstrui.exe.mui

X:\Program Files\Common Files\Microsoft Shared\ink
X:\sources
```

拷贝上述文件仅能实现在 WinPE 中的 UI 显示以及 Startup Repair 的功能，像 System Restore 等功能需要补齐更多文件和注册表才能在 WinPE 中工作。

综上，通过拷贝文件的方式将 WinRE 功能集成到 WinPE 是不现实的。

换一种思路，可以考虑将 WinRE 的 wim 文件作为 boot.wim 制作成 WinPE，然后为其注入驱动和工具，应该就能实现利用 WinRE 修复功能的需求，同时大大减少镜像的体积，参考步骤如下：

1. 选择一个安装镜像的 ISO

比如 Windows 10/11 或者 Windows Server 的安装镜像，若需要更多的功能选项，建议选择 Windows Client，双击挂载到文件目录中。

2. 切换到挂载目录 sources 路径，会发现有两个较大的文件 boot.wim 和 install.wim。

3. 将 boot.wim 替换之前制作步骤中的 boot.wim (C:\WinPE_amd64\media\sources\boot.wim)。

4. 参考之前的步骤对其进行驱动导入等定制后制作成镜像。注意点如下：

a. 系统安装镜像里面的 boot.wim 里面有 2 个 index.

对其更改需要指定 index 2，而非之前步骤中的 index 1。

可以通过以下的命令来查看 WIM 文件中的 index 信息：

```
dism.exe /get-wiminfo /wimfile:"<location of wim>"
```

将<location of wim>替换成 WIM 文件的实际路径，如下为运行示例：

```
C:\Windows\System32>dism /get-wiminfo /wimfile:"f:\boot.wim"

Deployment Image Servicing and Management tool
Version: 10.0.22621.1

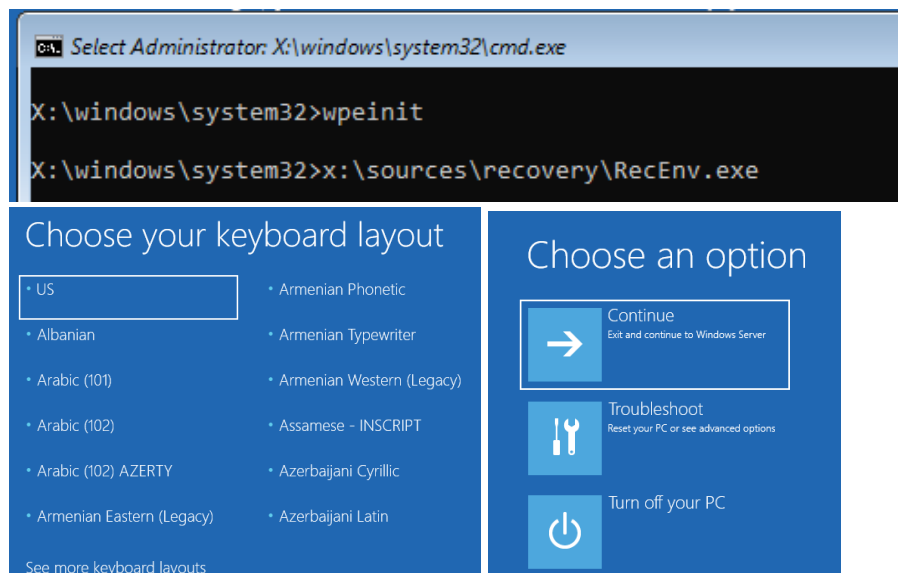
Details for image : f:\boot.wim

Index : 1
Name : Microsoft Windows PE (amd64)
Description : Microsoft Windows PE (amd64)
Size : 2,032,164,171 bytes

Index : 2
Name : Microsoft Windows Setup (amd64)
Description : Microsoft Windows Setup (amd64)
Size : 2,131,654,360 bytes

The operation completed successfully.
```

- b. 若选择 WinRE 的 boot.wim 基于不同的 Windows 内核，
在进行 1.4 章节添加驱动的时候，需要更改成对应的驱动版本。
在进行 2.3 章节添加系统组件操作时要选择安装对应版本的 ADK 来添加对应版本的系统组件。
- c. 可选择将 WinRE 的 boot.wim mount 起来，将其中 `sources\setup.exe` 删除。
这样制作成 WinPE 启动后，会直接启动熟悉的 CMD 窗口，若需要使用 WinRE 功能，
只需在 CMD 窗口中运行 `X:\sources\recovery\RecEnv.exe`：



5. WinPE 的未来定制方向

WinPE 未来的定制方向应该朝着集成化以及智能化发展，基于各项功能完善的基础上，可开发自定义的脚本功能，增加其易用性。

市面上有很多高度定制化的 WinPE，微软自身也有产品依赖于高度定制化的 WinPE。比如 MDT (Microsoft Deployment Toolkit)，其定制目的为自定义部署操作系统，在 WinPE 中实现了自动化分区，安装驱动，离线设置操作系统等一系列功能。

在这之前，需要对各项功能进行测试，并进行集成化的测试，同时也需要了解 WinPE 的一些逻辑原理流程。

5.1 WinPE 的启动流程

当 WinPE 启动后，省略 winlogon.exe 之前的部分。

winlogon.exe 会先去读 `HKLM\SYSTEM\Setup\CmdLine` 注册表，以决定启动什么进程。通常该注册表里面的值为 `Winpeshl.exe`。`Winpeshl.exe` 会尝试先去启动 `%SYSTEMDRIVE%\sources\setup.exe`。如果 `setup.exe` 不存在，会去尝试执行 `%SYSTEMROOT%\system32\winpeshl.ini` 中指定的应用。如果 `winpeshl.ini` 也没有指定应用，`Winpeshl.exe` 会去执行 `cmd /k %SYSTEMROOT%\system32\startnet.cmd`。默认情况下 `startnet.cmd` 里面仅包含一条命令即 `Wpeinit.exe`。

上述原理也是 4.2 章节中的 4.c 步骤删除 `setup.exe` 的理论依据。

5.2 简单的定制化示例

在这个示例中，将为 WinPE 添加一个自动启动项，该启动项显示一些常用的工具和指令选项，用户只需要根据屏幕提示，即可进行相应的操作。

1. 打开在 Mount 路径 `\Windows\System32\startnet.cmd`，在其中添加一行 `index.bat`。

2. 新建 Mount 路径 `\Windows\System32\index.bat`，内容如下：

```
@echo off
color 02
Title WIM 镜像部署工具
cls

rem 开启高性能模式
powercfg -s 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
for %a in (a b c d e f g h i j k l m n o p q r s t u v w x y z) do (if
exist %a:\WimFiles\setup.exe set DiskT=%a:\)
```

```

rem 设置半透明
"X:\Windows\System32\Cmdotc.vbs"

rem 判断 PE 启动模式
for /f "tokens=2* delims= " %%A in ('reg query
HKLM\System\CurrentControlSet\Control /v PEFirmwareType') DO SET Firmware=%%B
if %Firmware%==0x1 set BootMode=BIOS
if %Firmware%==0x2 set BootMode=UEFI
echo #####
echo #当前启动类型为: %BootMode% 启动

:start
echo #####
echo #输入"1" 将手动格式化磁盘并安装操作系统_
echo #输入"2" 将启动 Process Monitor 工具_
echo #输入"3" 将退出安装程序并关机_
echo #输入"4" 将退出安装程序并重启_
echo #####
set /p A=请输入:
cls

If "%A%"=="1" (
    "%DiskT%\WimFiles\setup.exe"
) else If "%A%"=="2" (
    "X:\Windows\Process Monitor\Procmon64.exe"
) else If "%A%"=="3" (
    goto shutdown
) else If "%A%"=="4" (
    goto reboot
) else goto error

goto start

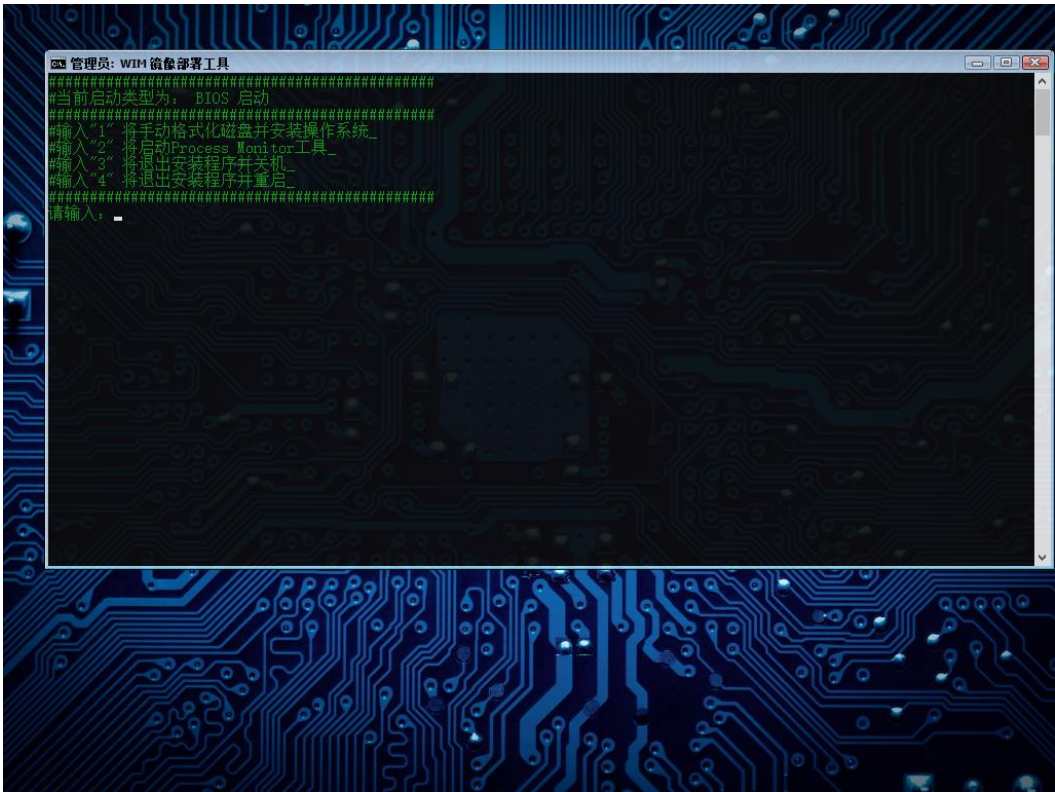
:error
echo 输入错误! 请重新输入!
goto start

:reboot
echo #####
echo #感谢使用, 即将重启电脑!
echo #####
wpeutil reboot

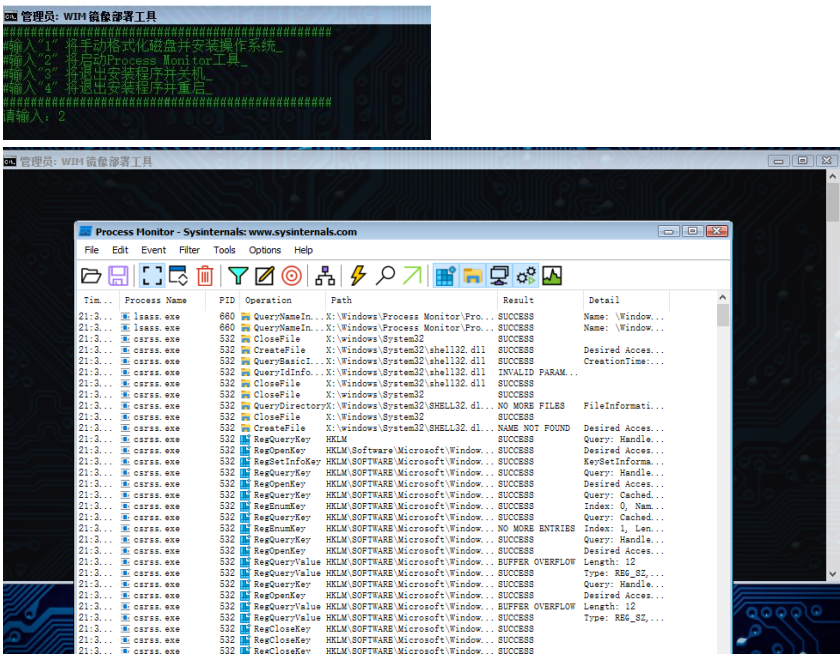
:shutdown
echo #####
echo #感谢使用, 即将关闭电脑!
echo #####
wpeutil shutdown

```

测试效果如下：



根据提示输入选择，运行指定的程序：



上述为一个简单的示例，在 WinPE 集成了 PowerShell 之后，类似的自定义开发将变得更简单和更灵活。