```
local tab_list = {
  '10-ABC',
  '11-DEF',
  '12-XYZ',
  '13-123',
  '14-456',
  '15-789'
}

local tab_template = [===[
  <VerticalLayout height="40" >
    <Label float="true" pos="20,0,32,32" mosue="false" text="" font="sym16" width="32" />
    <Option float="true" pos="0,0,240,32" group="nav_item" name="¥Nav[{¥I}]" style="nav_item"
text="{¥ITEM}" />
  </VerticalLayout>
]===]

local tablayout_head = [===[
  <TabLayout name="¥TabLayoutMain" selectedid="-1" topbordersize="1" bordercolor="#ff000000"
padding="2,2,2,2" >

    <VerticalLayout padding="20,0,0,0" >
        <Label text="%{Home}" font="24" height="50" />
    </VerticalLayout>
]===]

local tablayout_template = [===[
    <VerticalLayout padding="20,0,0,0" >
        <Label text="{¥ITEM}" font="24" height="50" />
        <VerticalLayout name="¥Page[{¥I}]" />
    </VerticalLayout>
]===]

local tablayout_end = '</TabLayout>'


Nav:SetTabItemXML(tab_template, '.+-(.+)')
Nav:SetTabLayoutXML(tablayout_head, tablayout_end)
Nav:SetTabLayoutItemXML(tablayout_template, '.+-(.+)')

local appstore_path = APP_Path .. '\\AppStore'

tab_list = dir.folders(appstore_path)
--table.insert(tab_list, '10086-设置')
--排序测试
table.sort(tab_list)
Nav:SetTabList(tab_list)

-- load AppStore, CatalogInfo, AppInfo
dofile(appstore_path .. '\\_main.lua')
AppStore.path = appstore_path


--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\_navData.lua"
```

```
local tilelayout_head = [===[
  <TileLayout itemsize="132,108">
]===]

local tilelayout_item = [===[
  <VerticalLayout>
    <VerticalLayout float="true" pos="0,0,128,104" >
        <Label bkimage="{$ITEM.ICON}" width="32" height="32" padding="48,5,0,5" />
        <Label text="{$ITEM.NAME}" font="16" height="24" align="center" />
        <Label name="$App[{$ITEM.ID}]STAR" text="{$ITEM.STAR}" textcolor="#FFA4A4A4" align="center" />

        <Slider name="$App[{$ITEM.ID}]ProgressBar" imm="true" min="1" max="100" step="1" value="100"
height="24" padding="0,-10,0,0"
            thumbsize="1,1" thumbimage="file='themes\thumb_.png'"
            bkimage="file='themes\sliderbk.png'" foreimage="file='themes\sliderfr.png'" visible="false" />
    </VerticalLayout>

    <Button  float="true" pos="0,0,128,104" name="$App[{$ITEM.ID}]" tooltip="{$ITEM.DESC}"
hotimage="color='#206F6F6F'" />
  </VerticalLayout>
]===]

local tilelayout_end = '</TileLayout>'

TabPage:SetTileLayoutXML(tilelayout_head, tilelayout_end, tilelayout_item)

function TabPage:FillTileItem(id, name, template)
  local xml = {}
  local tmpl = ''
  local icon_path = string.format("%s\\%s\\", AppStore.path, name)
  local items = CatalogInfo[name].item
  local item = nil
  for i = 1, #items do
    item = items[i]
    tmpl = template:gsub('{$ITEM.ID}', item.id)
         if string.sub(item.icon, 2, 2) == ':' then
      tmpl = tmpl:gsub('{$ITEM.ICON}', item.icon)
    else
      tmpl = tmpl:gsub('{$ITEM.ICON}', icon_path .. item.icon)
    end

    tmpl = tmpl:gsub('{$ITEM.NAME}', item.name)
    if item.desc then
      tmpl = tmpl:gsub('{$ITEM.DESC}', item.desc)
    else
      tmpl = tmpl:gsub('{$ITEM.DESC}', '')
    end

    if item.star then
      tmpl = tmpl:gsub('{$ITEM.STAR}', item.star)
    else
      tmpl = tmpl:gsub('{$ITEM.STAR}', '-')
    end
    table.insert(xml, tmpl)
  end
  return table.concat(xml, "\r\n")
end
```

```
--"ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\_pageData.lua"
```

```lua
info.id = "vhdboot"

info.name = "添加VHD到系统"

info.icon = "vhd.png"

info.desc = "你可以一键重启到vhd(vhdx)，或添加vhd(vhdx)镜像到开机菜单"

info.star = "内置应用"

info.appmode = "vhdboot"

info.nbapp = "vhdboot"
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\00-vhdboot.lua"
```

```
fixed_out = {}
table.insert(
                 fixed_out,
                 string.format("<i ..\\UI_DL\\app_ico\\腾讯QQ.png><a qq><c #000000>本软件由史上最伟大网管耗电开发</c></a> ")
             )
table.insert(
                 fixed_out,
                 string.format("<i ..\\UI_DL\\app_ico\\qun.png><c #000000>所有脚本均从:\n(WINPE系统DIY交流群:343245050)复制、抄袭</c> \n \n \n \n                   已授权给:地球上所有PE")
             )
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\about.lua"
```

```
-- 提取固定值
local exitcode, stdout = winapi.execute([["bcdedit /copy {bootmgr} /d "8899.efi"]])
bootfile_id = string.match(stdout, "{(%x%x%x%x%x%x%x%x%-%x%x%x%x%-%x%x%x%x%-
%x%x%x%x%-%x%x%x%x%x%x%x%x%x%x%x%x)}")
                    exec("/hide /wait",
                    [[bcdedit /set "{]] .. bootfile_id .. [[}" device partition=B:]])

                    exec("/hide /wait",
           [[bcdedit /set "{]] .. bootfile_id .. [[}" path \EFI\Boot\grubfmx64.efi]])
                     exec("/hide /wait",
           [[bcdedit /deletevalue "{]] .. bootfile_id .. [[}" locale]])
                            exec("/hide /wait",
           [[bcdedit /deletevalue "{]] .. bootfile_id .. [[}" inherit]])
                            exec("/hide /wait",
           [[bcdedit /deletevalue "{]] .. bootfile_id .. [[}" default]])
                            exec("/hide /wait",
           [[bcdedit /deletevalue "{]] .. bootfile_id .. [[}}" displayorder]])
                            exec("/hide /wait",
           [[bcdedit /deletevalue "{]] .. bootfile_id .. [[}}" timeout]])
                            exec("/hide /wait",
           [[bcdedit /set {fwbootmgr} "{]] .. bootfile_id .. [[}" /addlast ]])



alert(bootfile_id)
--"чд·��":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\add_efifile_boot.lua"
```

WinXShell

- Project
- Documentation

Contents

- 属性
- 方法
- 事件函数

Classes

- **App**
- Shell
- System

Libraries

- Lua-Engine
- UI-Engine
- Windows-Shell

Program

- WinXShell

Manual

- CSIDL
- 更新历史记录

Class `App`

App 对象

属性

App.Version	程序版本信息
Lua.Version	Lua模块版本信息
App.Name	程序文件名
App.Path	程序所在目录
App.FullPath	程序的全路径名
App.CmdLine	程序的命令行参数

方法

App:Info (key)	获取 信息
App:CreateGUID ()	生成 一个GUID序列字符串
App:Exec ()	调用 子程序 (等待程序结束)
App:Run ()	启动 程序 (不等待)
App:Sleep (ms)	暂停 当前进程(线程)执行，用于等待或者延时处理
App:Pause ()	暂停 作为启动管理器时，可用此命令维持启动进程
App:SetTimer (id, interval)	注册 计时器
App:KillTimer (id)	结束 计时器
App:Print (...)	输出 信息
App:JCfg (section, key1[, key2[, ...]])	获取 WinXShell.jcfg中的配置参数

事件函数

App:OnLoad ()	程序启动事件
App:OnDaemon ()	作为常驻进程时，启动事件
App:OnTimer ()	计时器事件

属性

App.Version
程序版本信息

Lua.Version
Lua模块版本信息

App.Name
程序文件名

App.Path
程序所在目录

App.FullPath
程序的全路径名

App.CmdLine
程序的命令行参数

方法

App:Info (key)
获取 信息

Parameters:

- key string

  信息名

  - CmdLine - 命令行
  - Path - 程序所在目录
  - Name - 程序文件名
  - `FullName` - 程序完整路径名
  - `IsShell` - 是否WinXShell作为系统外壳程序

Returns:

请查看实例代码

Usage:

```
-- WinXShell.exe -script info.lua
-- info.lua
 App:Info('CmdLine')
 ---> '-script info.lua'

App:Info('Path')
 ---> C:\Program Files\WinXShell

App:Info('Name')
 ---> WinXShell.exe

App:Info('FullName')
 ---> C:\Program Files\WinXShell\WinXShell.exe

App:Info('IsShell')
 ---> 0
```

App:CreateGUID ()

生成 一个GUID序列字符串

App:Exec ()

调用 子程序 (等待程序结束)

App:Run ()

启动 程序 (不等待)

App:Sleep (ms)

暂停 当前进程(线程)执行，用于等待或者延时处理

Parameters:

- ms int 暂停时间 (毫秒)

Usage:

```
WinXShell.exe -console -code App:Sleep(5000)
```

App:Pause ()

暂停 作为启动管理器时，可用此命令维持启动进程

Usage:

```
App:Pause()
```

App:SetTimer (id, interval)

注册 计时器

Parameters:

- id int 计时器标识号 (请使用 **TID_USER** + 数字形式 设置，
  小于 **TID_USER (20000)** 以下的标识号作为程序内部计时器标识号)
- interval int 时间间隔 (毫秒)

Usage:

```
local TID_REFRESH_APP = TID_USER + 1        -- TID_USER = 20000
App:SetTimer(TID_REFRESH_APP, 2000)
```

App:KillTimer (id)

结束 计时器

Parameters:

- id int 计时器标识号

App:Print (...)

输出 信息

Parameters:

- ...

App:JCfg (section, key1[, key2[, ...]])

获取 WinXShell.jcfg中的配置参数

Parameters:

- section string 段名
- key1 string 项目名
- key2 string 项目名 (*optional*)
- ... (*optional*)

事件函数

App:OnLoad ()

程序启动事件

App:OnDaemon ()

作为常驻进程时，启动事件

App:OnTimer ()

计时器事件

generated by LDoc 1.4.6

```
require 'winapi'

winapi.set_encoding(winapi.CP_UTF8)

function winapi.temp_name() return os.tmpname() end -- overwrite winapi's bug temp_name()

function winapi.files(mask,subdirs,attrib)
    local flags = '/B '
    if subdirs then flags = flags..' /S' end
    if attrib then flags = flags..' /A:'..attrib end
    local ret, text = winapi.execute('dir '..flags..' "'..mask..'"','ansi') --unicode
    if ret ~= 0 then return nil,text end
    return text:gmatch('[^\r\n]+')
end

app_folders = {}

local function bind_to_folder(fpath, tbl)
    local root = tbl
    local paths = fpath:gmatch('[^\\]+')
    local fname
    print(fpath)
    for pname in paths do
      fname = pname
      if root[pname] then
        root = root[pname]
      end
    end
    root[fname] = fpath
end

local function load_programs(drv)
  app_folders[drv] = {}
  local t = app_folders[drv]

  -- create the dir struct first
  local root_path = drv .. [[:\ProgramData\Microsoft\Windows\Start Menu\Programs\]]
  local items, err = winapi.files(root_path .. '*.*', true, 'D')
  if not items then return print(err) end

  local rpath, paths
  local root = nil
  for f in items do
    rpath = f:gsub(root_path, '')
    root = t
    paths = rpath:gmatch('[^\\]+')
    for pname in paths do
      if root[pname] then
        root = root[pname]
      else
        root[pname] = {}
      end
    end
  end

  items, err = winapi.files(root_path .. '*.*', true, '-D')
  if not items then return print(err) end
  local lpath = ''
        table.print(t)
  for f in items do
    rpath = f:gsub(root_path, '')
    if not rpath:lower():match('desktop.ini$') then
      bind_to_folder(rpath, t)
    end
  end
  -- table.print(app_folders['C'])
end

local function pairsByKeys (t, f)
  local a = {}
  for n in pairs(t) do table.insert(a, n) end
  table.sort(a, f)
  local i = 0      -- iterator variable
  local iter = function ()   -- iterator function
    i = i + 1
    if a[i] == nil then return nil
    else return a[i], t[a[i]]
    end
  end
  return iter
end

local function space(n)
  return string.rep('  ', n)
end

local xml = ''
local function app_walk(t, level, fpath)
    local str = ''
    local vis = [[folderattr="selected=&quot;true&quot;"]]
    if level > 0 then vis = [[folderattr="selected=&quot;false&quot;" visible="false"]] end
    for k, v in pairsByKeys(t) do
        if type(v) == 'table' then
            --print(space(level) .. k)
            str = string.format([[<TreeNode text="%s" %s style="folder_node" >]] .. '\n', k, vis)
            xml = xml .. space(level) .. str
            app_walk(v, level + 1, fpath .. '\\' .. k)
            xml = xml .. space(level) .. '</TreeNode>\n'
        else
            --print(string.rep('  ', level) .. k)
            local text = string.match(k, '[^\\.]+')
            xml = xml .. space(level) ..
            string.format([[<TreeNode text="%s" %s style="item_node" itemattr="bkimage=&quot;file='%s,F'
dest='4,4,28,28'&quot;" />]] .. '\n', text, vis, fpath .. '\\' .. k)
        end
    end
end

function app_tree_xml(drv)
  xml = ''
  load_programs(drv)
  local tree = app_folders[drv]
  --table.print(tree)
  app_walk(tree, 0, drv .. [[:\ProgramData\Microsoft\Windows\Start Menu\Programs\]])
```

```
   -- print(xml)
  return xml
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Launcher\apps_tree.lua"
```

```
@echo off
set args1=%1
set args2=%2
set args3=%3
set args4=%4
set args5=%5
set args6=%6
set args7=%7
set args8=%8
set args9=%9
call :GetWinXShell
if exist "X:\Program Files\wxsUI\UI_AppStore\nbappstore.lua" start "" "X:\Program Files\wxsUI\UI_AppStore\nbappstore.lua"&&exit
cd /d %temp%
mode con: cols=22 lines=2
if not "%args1%" == "" (
start "" %WinXShell% -ui -jcfg wxsUI\\UI_Appstore\\main.jcfg %args1% %args2% %args3% %args4% %args5% %args6% %args7% %args8% %args9%
exit
)
if exist %temp%\appstore-master rd /s /q %temp%\appstore-master
if exist %temp%\appstore\wxsUI\UI_AppStore\loading.jcfg start "" %WINXSHELL% -ui -jcfg wxsUI\UI_AppStore\loading.jcfg -wait 8 -text "正在载入数据,稍候..."
appstore\aria2c.exe --check-certificate=false --user-agent=Wget/1.21.1 --allow-overwrite=true https://gitee.com/zwj4031/appstore/repository/archive/master.zip -d %temp% -o appstore.zip
appstore\bin\7zx64.exe x -y %temp%\appstore.zip
start "" /w %WINXSHELL%  -code "QuitWindow(nil,'AppStore')"
if exist %temp%\appstore-master call :checkrobocopy
start "" %WinXShell% -ui -jcfg wxsUI\\UI_Appstore\\main.jcfg
exit
:GetWinXShell
```

```
if exist %temp%\AppStore\WinXShell.exe set WINXSHELL=%temp%\AppStore\WinXShell.exe

if exist ..\..\WinXShell.exe set WINXSHELL=..\..\WinXShell.exe

exit /b

:checkrobocopy

if exist %systemdrive%\windows\system32\robocopy.exe (

robocopy %temp%\appstore-master %temp%\appstore

robocopy %temp%\appstore-master\AppStore %temp%\appstore\AppStore /mir

) else (

xcopy /s /e /y %temp%\appstore-master\AppStore %temp%\appstore\AppStore

)

if exist %systemdrive%\windows\system32\robocopy.exe (

robocopy /mir %temp%\appstore-master\wxsUI %temp%\appstore\wxsUI

) else (

xcopy /s /e /y %temp%\appstore-master\wxsUI %temp%\appstore\wxsUI

)


if exist %systemdrive%\windows\system32\robocopy.exe (

robocopy %temp%\appstore-master\bin %temp%\appstore\bin

) else (

xcopy /s /e /y %temp%\appstore-master\bin %temp%\appstore\bin

)

exit /b
```

```
@echo off
if exist "X:\Program Files\wxsUI\UI_AppStore\nbappstore.lua" start "" "X:\Program
Files\wxsUI\UI_AppStore\nbappstore.lua"&&exit
cd /d %temp%
mode con: cols=22 lines=2

if exist %temp%\appstore-master rd /s /q %temp%\appstore-master
call :GetWinXShell
if exist %temp%\appstore\wxsUI\UI_AppStore\loading.jcfg start "" %WINXSHELL% -ui -jcfg
wxsUI\UI_AppStore\loading.jcfg -wait 8 -text "正在载入数据,稍候..."
appstore\aria2c.exe --check-certificate=false --user-agent=Wget/1.21.1 --allow-overwrite=true
https://gitee.com/zwj4031/appstore/repository/archive/master.zip -d %temp% -o appstore.zip
appstore\bin\7zx64.exe x -y %temp%\appstore.zip
start "" /w %WINXSHELL%  -code "QuitWindow(nil,'AppStore')"
if exist %temp%\appstore-master call :checkrobocopy
start "" %WinXShell% -ui -jcfg wxsUI\\UI_Appstore\\main.jcfg
exit
:GetWinXShell
if exist %temp%\AppStore\WinXShell.exe set WINXSHELL=%temp%\AppStore\WinXShell.exe
if exist ..\..\WinXShell.exe set WINXSHELL=..\..\WinXShell.exe
exit /b
:checkrobocopy
if exist %systemdrive%\windows\system32\robocopy.exe (
robocopy %temp%\appstore-master %temp%\appstore
robocopy %temp%\appstore-master\AppStore %temp%\appstore\AppStore /mir
) else (
xcopy /s /e /y %temp%\appstore-master\AppStore %temp%\appstore\AppStore
)
if exist %systemdrive%\windows\system32\robocopy.exe (
robocopy /mir %temp%\appstore-master\wxsUI %temp%\appstore\wxsUI
) else (
xcopy /s /e /y %temp%\appstore-master\wxsUI %temp%\appstore\wxsUI
)

if exist %systemdrive%\windows\system32\robocopy.exe (
robocopy %temp%\appstore-master\bin %temp%\appstore\bin
) else (
xcopy /s /e /y %temp%\appstore-master\bin %temp%\appstore\bin
)
exit /b
```

```
cd /d "%~dp0"
::mode con: cols=36 lines=1
::@echo off
call :GetWinXShell

%WinXShell% -ui -jcfg wxsUI\\UI_Appstore\\main.jcfg
exit
:GetWinXShell
if exist WinXShell.exe set WINXSHELL=WinXShell.exe
if exist ..\..\WinXShell.exe set WINXSHELL=..\..\WinXShell.exe
exit /b
```

```
@ech@echo on
set root=%~dp0
cd /d "%~dp0..\.."
call :GetWinXShell
call :GetFolderName "%~p0"
if exist %temp%\AppStore rd /s /q %temp%\AppStore
mkdir %temp%\AppStore

cd /d %temp%\AppStore
git clone https://gitee.com/zwj4031/appstore.git
if errorlevel 128 call :timesync
start "" /w %WINXSHELL%  -code "QuitWindow(nil,'AppStore')"
call :checkrobocopy
exit
goto :EOF

:GetWinXShell
if exist x64\Debug\WinXShell.exe set WINXSHELL=x64\Debug\WinXShell.exe&&goto :EOF
if exist WinXShell.exe set WINXSHELL=WinXShell.exe&&goto :EOF
if exist WinXShell_%PROCESSOR_ARCHITECTURE%.exe set WINXSHELL=WinXShell_x86.exe&&goto :EOF
set WINXSHELL=WinXShell_x64.exe
goto :EOF

:GetFolderName
if "x%~2"=="x1" set "p0=%~n1" && goto :EOF
set "p0=%~1"
set p0=%p0:~0,-1%
call :GetFolderName  "%p0%" 1
goto :EOF

:checkrobocopy
```

```
if exist %systemdrive%\windows\system32\robocopy.exe (
robocopy %temp%\AppStore\appstore "X:\Program Files"
robocopy %temp%\AppStore\appstore\appstore "X:\Program Files\AppStore" /mir
robocopy  %temp%\AppStore\appstore\wxsUI\UI_AppStore "X:\Program Files\wxsUI\UI_AppStore"
/mir
robocopy  %temp%\AppStore\appstore\wxsUI\UI_NBGI "X:\Program Files\wxsUI\UI_NBGI" /mir
robocopy  %temp%\AppStore\appstore\wxsUI\UI_Tool "X:\Program Files\wxsUI\UI_Tool" /mir
) else (
xcopy /s /e /y %temp%\AppStore\appstore\*.* "X:\Program Files"
)
exit /b

:timesync
start "" /min "%root%timesync.exe" /auto
start "" /w %WINXSHELL% -ui -jcfg wxsUI\\UI_AppStore\\loading.jcfg -top -wait 3 -text "获取数
据失败，正在核准时间..."
start "" /w %WINXSHELL% -code "CloseWindow('WindowClass_0', 'TimeSync ')"
cd /d %temp%\AppStore
git clone https://gitee.com/zwj4031/appstore.git
exit /b
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style shared="true" name="ct-bkinfo" bkcolor="#FF2067B2" />
    <Default name="Label" textcolor="#FFFFFFFF" />
    <Default name="Button" textcolor="#FFFFFFFF" />
    <Style name="ct-bk" bkcolor="#FF005A9E" />
    <Style name="ct-txtinfo" textcolor="#FF429CE3" />
    <Style name="ct-bkhori" bkcolor="#FF337BB1" />
    <Style name="ct-today" textcolor="#FFFFFFFF" bkimage="file='themes\today-blue.png'" />
</ColorTheme>
```

```
        <VerticalLayout height="40" padding="0,20,0,0">
            <HorizontalLayout height="32">
                <Label font="sym" text="   " height="32" width="32" padding="10,0,0,0" />
                <Slider name="brightness_slider" min="20" max="100" step="10" value="100" height="24"
padding="5,4,10,0"
                    thumbsize="8,24" thumbimage="file='thumb.png'"
                    bkimage="file='sliderbk.png'" foreimage="file='sliderfr.png'" />
            </HorizontalLayout>
        </VerticalLayout>
```

```
    <VerticalLayout>
        <HorizontalLayout height="32" padding="0,5,0,0">
            <Label font="sym" text="   " height="32" width="32" padding="10,0,0,0" />
            <Slider name="brightness_slider" min="20" max="100" step="10" value="100" height="24" padding="5,4,10,0"
                thumbsize="8,24" thumbimage="file='thumb.png'"
                bkimage="file='sliderbk.png'" foreimage="file='sliderfr.png'" />
        </HorizontalLayout>
    </VerticalLayout>
```

```
{
 "name":"UI_Calendar",
 "class":"CDUICalendarWindow",
 "title":"Calendar",
 "entry":"Calendar10.xml",
 "lua":"Calendar.lua",
 "singleton":true,
 "position":"(auto)",
 "nobaricon":true,
 "trans":240,
 "OnDeactive":"hide",
 "#lunar":"none|auto"
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Calendar\Calendar.jcfg",
}
```

```
dofile(sui:info('uipath') .. 'lunar.lua')

p = print

TICK_TIMER_ID = 1000

local function get_month_day(year, mon)
  local days = {31,28,31,30,31,30,31,31,30,31,30,31}
  if mon == 2 and year % 4 == 0 then
    return 29
  else
    return days[mon]
  end
end

local function fillnum(num, c)
    if c == nil then c = '0' end
    return string.sub(c .. num, -2, -1)
end

function draw_cal(year, mon)
  -- draw title
  nav_mon.text = year .. '-' .. fillnum(mon)
  -- draw days
  local mdate_1 = os.time{year=year, month=mon, day=1, hour=0}
  local week = os.date('%w', mdate_1) -- type(week) => string
  local i, j
  week = math.tointeger(week) -- convert to integer
  suilib.print('week' .. week)
  for i = 1,week do
    d['1' .. i].text = ''
  end


  if last_day_elem ~= nil then
     --last_day_elem.style = 'day'
    last_day_elem.textcolor = normal_textcolor
    last_day_elem.bkimage = normal_bk
     last_day_elem = nil
  end

  local day_elem = nil
  i = week + 1;j = 1
  local md = get_month_day(year, mon)
  for k = 1,42 - week do
    if k <= md then
      d[j .. i].text = get_day_text(year, mon, k) -- fillnum(k, ' ')
      if k == cur_day then
        p("today:" .. k)
        day_elem = d[j .. i]
      end
    else
      d[j .. i].text = ''
    end
    i = i + 1
    if i > 7 then i = 1;j = j + 1 end
  end
  if year == cur_year and mon == cur_mon then
    --day_elem.style = 'today'
    day_elem.textcolor = today_textcolor
    day_elem.bkimage = today_bk
    last_day_elem = day_elem
  end

end

d = {}
local function get_elems()
  for i=1,6 do
    for j=1,7 do
      d[i ..j] = sui:find('d' .. i .. j)
    end
  end
  nav_mon = sui:find('nav_mon')
  now_year = sui:find('now_year')
  now_info = sui:find('now_info')
  now_time = sui:find('now_time')
  month_days_area = sui:find('month_days_area')
  local today_attr = sui:find('today_attr')
  today_textcolor = today_attr.textcolor
  p(today_textcolor)
  today_bk = today_attr.bkimage
  normal_textcolor = d['11'].textcolor
  normal_bk = d['11'].bkimage
end

local function var_init()
  today = os.time()
  cur_year = math.tointeger(os.date('%Y', today))
  cur_mon = math.tointeger(os.date('%m', today))
  cur_day = math.tointeger(os.date('%d', today))
  str_today = cur_year .. cur_mon .. cur_day
end

local function draw_now_info()
    now_info.text = string.format('%d-%d-%d, %%{info_w%s}',
        cur_year, cur_mon, cur_day, os.date('%w', today))
    now_time.text = os.date('%H:%M:%S', today)
end

function onload()
  get_elems()
  var_init()
  lunar_init()
  view_year = cur_year
  view_mon = cur_mon
  draw_now_info()
  last_day_elem = nil -- mark today
  draw_cal(cur_year, cur_mon)
  sui:find('datetime_settings').text = sui:find('datetime_settings').text
  suilib.call('SetTimer', TICK_TIMER_ID, 1000)
end
```

```lua
function  ontimer(id)
    if id == TICK_TIMER_ID then
        var_init()
        draw_now_info()
    end
end

function onclick(ctrl)
  if ctrl == 'pre_mon' then
    view_mon = view_mon - 1
    if view_mon == 0 then
        view_year = view_year - 1
        view_mon = 12
    end
    draw_cal(view_year, view_mon)
  elseif ctrl == 'next_mon' then
    view_mon = view_mon + 1
    if view_mon == 13 then
        view_year = view_year + 1
        view_mon = 1
    end
    draw_cal(view_year, view_mon)
  elseif ctrl == 'datetime_settings' then
    suilib.call("run", "timedate.cpl")
  end
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Calendar\Calendar.lua"
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="560,360" caption="0,0,0,32" sizebox="6,6,6,6"  >
    <Style imagerectstyle="l,t,w,h" />

    <Style name="week" height="40" width="40" font="18b" textcolor="#FFA0A0A0" />
    <Style name="day" height="40" width="40" font="18b" align="center" />
    <VerticalLayout bkcolor="#FFFFFFFF">
      <!--
        <HorizontalLayout height="32" bkcolor="#FFFF5151">
            <Label name="title" text="Calendar" textcolor="#FFFFFFFF" align="center" />
            <Button name="::closebtn" bkimage="file='close.png'" width="16" height="16"
padding="0,8,8,0" />
        </HorizontalLayout>
      -->
        <!-- get theme attribute value for lua -->
        <label name="today_attr" style="ct-today" visible="false" />
        <HorizontalLayout>
          <VerticalLayout width="240" style="ct-bkinfo" >
            <VerticalLayout padding="20,10,0,10">
            <label name="now_year" font="20" textcolor="#FFFFFFFF" />
            <label name="now_info" font="40b" textcolor="#FFFFFFFF" multiline="true" />
            <Control />
            <label name="now_time" font="40b" text="time" textcolor="#FFFFFFFF" />
            </VerticalLayout>
            <VerticalLayout height="45" style="ct-bkinfo-inverse" >
              <button name="more" font="20" text="More Settings" align="left" textpadding="20,10,0,0"
textcolor="#FFFFFFFF" />
            </VerticalLayout>
          </VerticalLayout>
          <VerticalLayout padding="20,0,20,0">
            <HorizontalLayout height="20" padding="0,10,0,0" >
            <Button name="pre_mon" text="   " font="i16" width="16" height="16" />
            <label name="nav_mon" text="2018-04" font="18b" align="center" />
            <Button name="next_mon" text="   " font="i16" width="16" height="16" />
            </HorizontalLayout>

            <HorizontalLayout padding="12,0,0,0">
              <label text="%{w7}" style="week" />
              <label text="%{w1}" style="week" />
              <label text="%{w2}" style="week" />
              <label text="%{w3}" style="week" />
              <label text="%{w4}" style="week" />
              <label text="%{w5}" style="week" />
              <label text="%{w6}" style="week" />
            </HorizontalLayout>

            <HorizontalLayout>
              <label name="d11" text="1" style="day" />
              <label name="d12" text="2" style="day" />
              <label name="d13" text="3" style="day" />
              <label name="d14" text="4" style="day" />
              <label name="d15" text="5" style="day" />
              <label name="d16" text="6" style="day" />
              <label name="d17" text="7" style="day" />
            </HorizontalLayout>

            <HorizontalLayout>
              <label name="d21" text="1" style="day" />
              <label name="d22" text="2" style="day" />
              <label name="d23" text="3" style="day" />
              <label name="d24" text="4" style="day" />
              <label name="d25" text="5" style="day" />
              <label name="d26" text="6" style="day" />
              <label name="d27" text="7" style="day" />
            </HorizontalLayout>

            <HorizontalLayout>
              <label name="d31" text="1" style="day" />
              <label name="d32" text="2" style="day" />
              <label name="d33" text="3" style="day" />
              <label name="d34" text="4" style="day" />
              <label name="d35" text="5" style="day" />
              <label name="d36" text="6" style="day" />
              <label name="d37" text="7" style="day" />
            </HorizontalLayout>

            <HorizontalLayout>
              <label name="d41" text="1" style="day" />
              <label name="d42" text="2" style="day" />
              <label name="d43" text="3" style="day" />
              <label name="d44" text="4" style="day" />
              <label name="d45" text="5" style="day" />
              <label name="d46" text="6" style="day" />
              <label name="d47" text="7" style="day" />
            </HorizontalLayout>

            <HorizontalLayout>
              <label name="d51" text="1" style="day" />
              <label name="d52" text="2" style="day" />
              <label name="d53" text="3" style="day" />
              <label name="d54" text="4" style="day" />
              <label name="d55" text="5" style="day" />
              <label name="d56" text="6" style="day" />
              <label name="d57" text="7" style="day" />
            </HorizontalLayout>

            <HorizontalLayout height="80" >
              <Control />
              <button name="::closebtn" text="OK" style="ct-textcolor" font="16b" />
              <button name="::closebtn" text="CANCEL" style="ct-textcolor" font="16b" />
            </HorizontalLayout>

          </VerticalLayout>
        </HorizontalLayout>
      </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="360,540" caption="0,0,0,32" sizebox="6,6,6,6"  >
    <Style imagerectstyle="l,t,w,h" />

    <Style name="week" width="46" font="18b" />
    <Style name="day" width="46" align="center" showhtml="true" />
    <VerticalLayout bkcolor="#FFFFFFFF">
      <!--
        <HorizontalLayout height="32" bkcolor="#FFFF5151">
            <Label name="title" text="Calendar" textcolor="#FFFFFFFF" align="center" />
            <Button name="::closebtn" bkimage="file='close.png'" width="16" height="16"
padding="0,8,8,0" />
        </HorizontalLayout>
      -->
        <!-- get theme attribute value for lua -->
        <Label name="today_attr" style="ct-today" visible="false" />
        <VerticalLayout style="ct-bk">
          <VerticalLayout height="120">
            <Label name="now_time" font="40b" text="time" padding="20,20,0,0" />
            <Label name="now_info" font="16" style="ct-txtinfo" padding="20,0,0,0" />
          </VerticalLayout>
          <Control height="1" style="ct-bkhori" />
          <VerticalLayout name="month_days_area">
            <HorizontalLayout height="40" padding="0,16,0,0" >
            <Label name="nav_mon" text="2018-04" align="left" textpadding="30,0,0,0" />
            <Button name="pre_mon" text="   " font="i16" width="46" height="40"  />
            <Button name="next_mon" text="   " font="i16" width="46" height="40" padding="6,0,0,0" />
            <Control width="20" />
            </HorizontalLayout>

            <HorizontalLayout height="40" padding="30,0,0,0">
              <Label text="%{w7}" style="week" />
              <Label text="%{w1}" style="week" />
              <Label text="%{w2}" style="week" />
              <Label text="%{w3}" style="week" />
              <Label text="%{w4}" style="week" />
              <Label text="%{w5}" style="week" />
              <Label text="%{w6}" style="week" />
            </HorizontalLayout>

            <VerticalLayout padding="20,0,0,0">
            <HorizontalLayout>
              <Label name="d11" text="1" style="day" />
              <Label name="d12" text="2" style="day" />
              <Label name="d13" text="3" style="day" />
              <Label name="d14" text="4" style="day" />
              <Label name="d15" text="5" style="day" />
              <Label name="d16" text="6" style="day" />
              <Label name="d17" text="7" style="day" />
            </HorizontalLayout>

            <HorizontalLayout>
              <Label name="d21" text="1" style="day" />
              <Label name="d22" text="2" style="day" />
              <Label name="d23" text="3" style="day" />
              <Label name="d24" text="4" style="day" />
              <Label name="d25" text="5" style="day" />
              <Label name="d26" text="6" style="day" />
              <Label name="d27" text="7" style="day" />
            </HorizontalLayout>
            <HorizontalLayout>
              <Label name="d31" text="1" style="day" />
              <Label name="d32" text="2" style="day" />
              <Label name="d33" text="3" style="day" />
              <Label name="d34" text="4" style="day" />
              <Label name="d35" text="5" style="day" />
              <Label name="d36" text="6" style="day" />
              <Label name="d37" text="7" style="day" />
            </HorizontalLayout>

            <HorizontalLayout>
              <Label name="d41" text="1" style="day" />
              <Label name="d42" text="2" style="day" />
              <Label name="d43" text="3" style="day" />
              <Label name="d44" text="4" style="day" />
              <Label name="d45" text="5" style="day" />
              <Label name="d46" text="6" style="day" />
              <Label name="d47" text="7" style="day" />
            </HorizontalLayout>

            <HorizontalLayout>
              <Label name="d51" text="1" style="day" />
              <Label name="d52" text="2" style="day" />
              <Label name="d53" text="3" style="day" />
              <Label name="d54" text="4" style="day" />
              <Label name="d55" text="5" style="day" />
              <Label name="d56" text="6" style="day" />
              <Label name="d57" text="7" style="day" />
            </HorizontalLayout>

            <HorizontalLayout>
              <Label name="d61" text="1" style="day" />
              <Label name="d62" text="2" style="day" />
              <Label name="d63" text="3" style="day" />
              <Label name="d64" text="4" style="day" />
              <Label name="d65" text="5" style="day" />
              <Label name="d66" text="6" style="day" />
              <Label name="d67" text="7" style="day" />
            </HorizontalLayout>

            </VerticalLayout>

            <Control height="1" style="ct-bkhori" />

            <VerticalLayout height="45">
              <Button name="datetime_settings" font="16b" text="#{@shell32.dll,24135}" align="left"
textpadding="20,10,0,0" />
            </VerticalLayout>
            <HorizontalLayout height="32">
              <Control />
              <Button name="::closebtn" text="%{OK}" font="16b" style="ct-txtinfo" />
              <Button name="::closebtn" text="%{CANCEL}" font="16b" style="ct-txtinfo" />
            </HorizontalLayout>

          </VerticalLayout>
        </VerticalLayout>
    </VerticalLayout>
</Window>
```

```
<VerticalLayout>
        <!-- get theme attribute value for lua -->
        <Label name="today_attr" style="ct-today" visible="false" height="24" />
        <VerticalLayout>
            <VerticalLayout height="100">
              <Button name="now_time" font="40b" text="time" align="left" padding="20,0,0,0" />
              <Label name="now_info" font="16" style="ct-txtinfo" padding="20,0,0,0" />
            </VerticalLayout>

            <Control height="1" style="ct-bkhori" />

            <VerticalLayout name="month_days_area" height="380" >
              <HorizontalLayout height="40" padding="0,16,0,0" >
                <Label name="nav_mon" text="2018-04" align="left" textpadding="30,0,0,0" />
                <Button name="pre_mon" text="   " font="i16" width="46" height="40"  />
                <Button name="next_mon" text="   " font="i16" width="46" height="40" padding="6,0,0,0"
/>
                <Control width="20" />
              </HorizontalLayout>

              <HorizontalLayout height="40" padding="30,0,0,0">
                <Label text="%{w7}" style="week" />
                <Label text="%{w1}" style="week" />
                <Label text="%{w2}" style="week" />
                <Label text="%{w3}" style="week" />
                <Label text="%{w4}" style="week" />
                <Label text="%{w5}" style="week" />
                <Label text="%{w6}" style="week" />
              </HorizontalLayout>

              <VerticalLayout padding="20,0,0,0">
                <HorizontalLayout>
                  <Label name="d11" text="1" style="day" />
                  <Label name="d12" text="2" style="day" />
                  <Label name="d13" text="3" style="day" />
                  <Label name="d14" text="4" style="day" />
                  <Label name="d15" text="5" style="day" />
                  <Label name="d16" text="6" style="day" />
                  <Label name="d17" text="7" style="day" />
                </HorizontalLayout>

                <HorizontalLayout>
                  <Label name="d21" text="1" style="day" />
                  <Label name="d22" text="2" style="day" />
                  <Label name="d23" text="3" style="day" />
                  <Label name="d24" text="4" style="day" />
                  <Label name="d25" text="5" style="day" />
                  <Label name="d26" text="6" style="day" />
                  <Label name="d27" text="7" style="day" />
                </HorizontalLayout>

                <HorizontalLayout>
                  <Label name="d31" text="1" style="day" />
                  <Label name="d32" text="2" style="day" />
                  <Label name="d33" text="3" style="day" />
                  <Label name="d34" text="4" style="day" />
                  <Label name="d35" text="5" style="day" />
                  <Label name="d36" text="6" style="day" />
                  <Label name="d37" text="7" style="day" />
                </HorizontalLayout>
                <HorizontalLayout>
                  <Label name="d41" text="1" style="day" />
                  <Label name="d42" text="2" style="day" />
                  <Label name="d43" text="3" style="day" />
                  <Label name="d44" text="4" style="day" />
                  <Label name="d45" text="5" style="day" />
                  <Label name="d46" text="6" style="day" />
                  <Label name="d47" text="7" style="day" />
                </HorizontalLayout>

                <HorizontalLayout>
                  <Label name="d51" text="1" style="day" />
                  <Label name="d52" text="2" style="day" />
                  <Label name="d53" text="3" style="day" />
                  <Label name="d54" text="4" style="day" />
                  <Label name="d55" text="5" style="day" />
                  <Label name="d56" text="6" style="day" />
                  <Label name="d57" text="7" style="day" />
                </HorizontalLayout>

                <HorizontalLayout>
                  <Label name="d61" text="1" style="day" />
                  <Label name="d62" text="2" style="day" />
                  <Label name="d63" text="3" style="day" />
                  <Label name="d64" text="4" style="day" />
                  <Label name="d65" text="5" style="day" />
                  <Label name="d66" text="6" style="day" />
                  <Label name="d67" text="7" style="day" />
                </HorizontalLayout>

              </VerticalLayout>

            </VerticalLayout>

          </VerticalLayout>
<VerticalLayout>
```

```
<VerticalLayout>
        <!-- get theme attribute value for lua -->
        <Label name="today_attr" style="ct-today" visible="false" height="24" />
        <VerticalLayout>
            <VerticalLayout height="100">
              <Button name="now_time" font="40b" text="time" align="left" padding="20,0,0,0" />
              <Label name="now_info" font="16" style="ct-txtinfo" padding="20,0,0,0" />
            </VerticalLayout>

            <Control height="1" style="ct-bkhori" />

            <VerticalLayout name="month_days_area" height="380" >
              <HorizontalLayout height="40" padding="0,16,0,0" >
                <Label name="nav_mon" text="2018-04" align="left" textpadding="30,0,0,0" />
                <Button name="pre_mon" text="   " font="i16" width="46" height="40"  />
                <Button name="next_mon" text="   " font="i16" width="46" height="40" padding="6,0,0,0"
/>
                <Control width="20" />
              </HorizontalLayout>

              <HorizontalLayout height="40" padding="30,0,0,0">
                <Label text="%{w7}" style="week" />
                <Label text="%{w1}" style="week" />
                <Label text="%{w2}" style="week" />
                <Label text="%{w3}" style="week" />
                <Label text="%{w4}" style="week" />
                <Label text="%{w5}" style="week" />
                <Label text="%{w6}" style="week" />
              </HorizontalLayout>

              <VerticalLayout padding="20,0,0,0">
                <HorizontalLayout>
                  <Label name="d11" text="1" style="day" />
                  <Label name="d12" text="2" style="day" />
                  <Label name="d13" text="3" style="day" />
                  <Label name="d14" text="4" style="day" />
                  <Label name="d15" text="5" style="day" />
                  <Label name="d16" text="6" style="day" />
                  <Label name="d17" text="7" style="day" />
                </HorizontalLayout>

                <HorizontalLayout>
                  <Label name="d21" text="1" style="day" />
                  <Label name="d22" text="2" style="day" />
                  <Label name="d23" text="3" style="day" />
                  <Label name="d24" text="4" style="day" />
                  <Label name="d25" text="5" style="day" />
                  <Label name="d26" text="6" style="day" />
                  <Label name="d27" text="7" style="day" />
                </HorizontalLayout>

                <HorizontalLayout>
                  <Label name="d31" text="1" style="day" />
                  <Label name="d32" text="2" style="day" />
                  <Label name="d33" text="3" style="day" />
                  <Label name="d34" text="4" style="day" />
                  <Label name="d35" text="5" style="day" />
                  <Label name="d36" text="6" style="day" />
                  <Label name="d37" text="7" style="day" />
                </HorizontalLayout>
                <HorizontalLayout>
                  <Label name="d41" text="1" style="day" />
                  <Label name="d42" text="2" style="day" />
                  <Label name="d43" text="3" style="day" />
                  <Label name="d44" text="4" style="day" />
                  <Label name="d45" text="5" style="day" />
                  <Label name="d46" text="6" style="day" />
                  <Label name="d47" text="7" style="day" />
                </HorizontalLayout>

                <HorizontalLayout>
                  <Label name="d51" text="1" style="day" />
                  <Label name="d52" text="2" style="day" />
                  <Label name="d53" text="3" style="day" />
                  <Label name="d54" text="4" style="day" />
                  <Label name="d55" text="5" style="day" />
                  <Label name="d56" text="6" style="day" />
                  <Label name="d57" text="7" style="day" />
                </HorizontalLayout>

                <HorizontalLayout>
                  <Label name="d61" text="1" style="day" />
                  <Label name="d62" text="2" style="day" />
                  <Label name="d63" text="3" style="day" />
                  <Label name="d64" text="4" style="day" />
                  <Label name="d65" text="5" style="day" />
                  <Label name="d66" text="6" style="day" />
                  <Label name="d67" text="7" style="day" />
                </HorizontalLayout>

              </VerticalLayout>

            </VerticalLayout>

          </VerticalLayout>
<VerticalLayout>
```

```
fixed_out = {}
table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\腾讯QQ.png><a qq><c #000000>腾讯QQ</c></a> \n")
            )
table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\微信.png><a wechat><c #000000>微信</c></a>\n")
            )                    --"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\chat.lua"
```

```
local panfu_list = {
    "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"
}

local fixed_out = {}

function findfile(file)
    for key, value in ipairs(panfu_list) do
        if File.exists(value .. [[:]] .. file) then
            havefile = "1"
            table.insert(fixed_out, string.format("盘符%s:", value))
        end
    end
    find_str = table.concat(fixed_out, "\n")
    if havefile == "1" then
        winapi.show_message("文件搜索",
                            [[找到]] .. file .. [[文件，在]] .. find_str)
    else
        winapi.show_message("文件搜索", "没有找到你要的文件!")
    end
end

findfile([[\boot\grubfm\diy.gz]])

--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\checkgrubfm.lua"
```

WinXShell

- Project
- Documentation

Manual

- **CSIDL**
- 更新历史记录

Libraries

- Lua-Engine
- UI-Engine
- Windows-Shell

Program

- WinXShell

Classes

- App
- Shell
- System

CSIDL

CSIDL (Shlobj.h) 常量特殊项 ID 列表 - Win32 apps | Microsoft Docs

```
-- CSIDL (Shlobj.h) 常量特殊项 ID 列表
#define CSIDL_DESKTOP                   0x0000        // <desktop>
#define CSIDL_INTERNET                  0x0001        // Internet Explorer (icon on desktop)
#define CSIDL_PROGRAMS                  0x0002        // Start Menu\Programs
#define CSIDL_CONTROLS                  0x0003        // My Computer\Control Panel
#define CSIDL_PRINTERS                  0x0004        // My Computer\Printers
#define CSIDL_PERSONAL                  0x0005        // My Documents
#define CSIDL_FAVORITES                 0x0006        // <user name>\Favorites
#define CSIDL_STARTUP                   0x0007        // Start Menu\Programs\Startup
#define CSIDL_RECENT                    0x0008        // <user name>\Recent
#define CSIDL_SENDTO                    0x0009        // <user name>\SendTo
#define CSIDL_BITBUCKET                 0x000a        // <desktop>\Recycle Bin
#define CSIDL_STARTMENU                 0x000b        // <user name>\Start Menu
#define CSIDL_MYDOCUMENTS               CSIDL_PERSONAL //  Personal was just a silly name for My Documents
#define CSIDL_MYMUSIC                   0x000d        // "My Music" folder
#define CSIDL_MYVIDEO                   0x000e        // "My Videos" folder
#define CSIDL_DESKTOPDIRECTORY          0x0010        // <user name>\Desktop
#define CSIDL_DRIVES                    0x0011        // My Computer
#define CSIDL_NETWORK                   0x0012        // Network Neighborhood (My Network Places)
#define CSIDL_NETHOOD                   0x0013        // <user name>\nethood
#define CSIDL_FONTS                     0x0014        // windows\fonts
#define CSIDL_TEMPLATES                 0x0015
#define CSIDL_COMMON_STARTMENU          0x0016        // All Users\Start Menu
#define CSIDL_COMMON_PROGRAMS           0x0017        // All Users\Start Menu\Programs
#define CSIDL_COMMON_STARTUP            0x0018        // All Users\Startup
#define CSIDL_COMMON_DESKTOPDIRECTORY   0x0019        // All Users\Desktop
#define CSIDL_APPDATA                   0x001a        // <user name>\Application Data
#define CSIDL_PRINTHOOD                 0x001b        // <user name>\PrintHood
#define CSIDL_LOCAL_APPDATA             0x001c        // <user name>\Local Settings\Applicaiton Data (non roaming)
```

```
#define CSIDL_COMMON_FAVORITES          0x001f

#ifndef _SHFOLDER_H_
#define CSIDL_INTERNET_CACHE            0x0020
#define CSIDL_COOKIES                   0x0021
#define CSIDL_HISTORY                   0x0022
#define CSIDL_COMMON_APPDATA            0x0023        // All Users\Application Data
#define CSIDL_WINDOWS                   0x0024        // GetWindowsDirectory()
#define CSIDL_SYSTEM                    0x0025        // GetSystemDirectory()
#define CSIDL_PROGRAM_FILES             0x0026        // C:\Program Files
#define CSIDL_MYPICTURES                0x0027        // C:\Program Files\My Pictures
#endif // _SHFOLDER_H_

#define CSIDL_PROFILE                   0x0028        // USERPROFILE
#define CSIDL_SYSTEMX86                 0x0029        // x86 system directory on RISC
#define CSIDL_PROGRAM_FILESX86          0x002a        // x86 C:\Program Files on RISC

#ifndef _SHFOLDER_H_
#define CSIDL_PROGRAM_FILES_COMMON      0x002b        // C:\Program Files\Common
#endif // _SHFOLDER_H_

#define CSIDL_PROGRAM_FILES_COMMONX86   0x002c        // x86 Program Files\Common on RISC
#define CSIDL_COMMON_TEMPLATES          0x002d        // All Users\Templates

#ifndef _SHFOLDER_H_
#define CSIDL_COMMON_DOCUMENTS          0x002e        // All Users\Documents
#define CSIDL_COMMON_ADMINTOOLS         0x002f        // All Users\Start Menu\Programs\Administrative Tools
#define CSIDL_ADMINTOOLS                0x0030        // <user name>\Start Menu\Programs\Administrative Tools
#endif // _SHFOLDER_H_

#define CSIDL_CONNECTIONS               0x0031        // Network and Dial-up Connections
#define CSIDL_COMMON_MUSIC              0x0035        // All Users\My Music
#define CSIDL_COMMON_PICTURES           0x0036        // All Users\My Pictures
#define CSIDL_COMMON_VIDEO              0x0037        // All Users\My Video
#define CSIDL_RESOURCES                 0x0038        // Resource Direcotry

#define CSIDL_CDBURN_AREA               0x003b        // USERPROFILE\Local Settings\Application Data\Microsoft\CD Burning
```

generated by LDoc 1.4.6

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
  <Default shared="true" name="Label" textcolor="#ffffffff" />
  <Default shared="true" name="Button" textcolor="#ffffffff" />
  <Default shared="true" name="CheckBox" textcolor="#ffffffff" />
  <Default shared="true" name="Option" textcolor="#ffffffff" />
  <Style name="ct-bknav" bkcolor="#ff1f1f1f" textcolor="#ffff00ff" />
  <Style name="ct-bkcontent" bkcolor="#ff000000" textcolor="#ffffff00" />
  <Style name="ct-button" textcolor="#FF000000" bkcolor="#ffcccccc" pushedimage="color='#ff666666'"
width="100" height="32"
    bordersize="2" bordercolor="#ffcccccc" hotbordercolor="#ff858585" pushedbordercolor="#ff666666" />
  <Style name="ct-combo" bkimage="file='themes\ComboButton-dark.png' corner='2,2,36,2'"
    bordersize="2" bordercolor="#ff666666" focusbordercolor="#ff858585"
    itemselectedtextcolor="#ffffffff" itemselectedbkcolor="#ff000000" itemtextcolor="#ffffffff"
itembkcolor="#ff2b2b2b"
    itemhottextcolor="#ffffffff" itemhotbkcolor="#ff404040" />
  <Style name="switch" width="44" height="20" normalimage="file='themes\switch-dark.png'
source='0,0,44,20' dest='0,0,44,20'"
    selectedimage="file='themes\switch-dark.png' source='0,20,44,20' dest='0,0,44,20'" />
  <Style name="radio" height="20" align="left" textpadding="25,0,0,0"
    normalimage="file='themes\radio-dark.png' source='0,0,20,20' dest='0,0,20,20'"
    selectedimage="file='themes\radio-dark.png' source='0,20,20,20' dest='0,0,20,20'" />
  <Style name="check" height="20" align="left" textpadding="25,0,0,0"
    normalimage="file='themes\check-dark.png' source='0,0,20,20' dest='0,0,20,20'"
    selectedimage="file='themes\check-dark.png' source='0,20,20,20' dest='0,0,20,20'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style name="ct-bktitle" bkcolor="#FF333333" />
    <Style name="ct-bksubtitle" bkcolor="#FF555555" />
    <Style name="ct-bkcontent" bkcolor="#FFF9F9F9" />
    <Style name="ct-bkontent-darker1" bkcolor="#FFF2F2F2" />
    <Style name="attentionBtn" width="120" height="32" textcolor="#FF000000"
         hottextcolor="#FFFFFF" hotimage="color='#FFFF4081'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
  <Default shared="true" name="Label" textcolor="#ffffffff" />
  <Default shared="true" name="Button" textcolor="#ffffffff" />
  <Default shared="true" name="CheckBox" textcolor="#ffffffff" />
  <Default shared="true" name="Option" textcolor="#ffffffff" />
  <Style name="ct-bknav" bkcolor="#ff202020" textcolor="#ffff00ff" />
  <Style name="ct-bkcontent" bkcolor="#ff191919" textcolor="#ffffff00" />
  <Style name="ct-button" textcolor="#FF000000" bkcolor="#ffcccccc" pushedimage="color='#ff666666'"
width="100" height="32"
    bordersize="2" bordercolor="#ffcccccc" hotbordercolor="#ff858585" pushedbordercolor="#ff666666" />
  <Style name="ct-combo" bkimage="file='themes\ComboButton-dark.png' corner='2,2,36,2'"
    bordersize="2" bordercolor="#ff666666" focusbordercolor="#ff858585"
    itemselectedtextcolor="#ffffffff" itemselectedbkcolor="#ff000000" itemtextcolor="#ffffffff"
itembkcolor="#ff2b2b2b"
    itemhottextcolor="#ffffffff" itemhotbkcolor="#ff404040" />
  <Style name="switch" width="44" height="20" normalimage="file='themes\switch-dark.png'
source='0,0,44,20' dest='0,0,44,20'"
    selectedimage="file='themes\switch-dark.png' source='0,20,44,20' dest='0,0,44,20'" />
  <Style name="radio" height="20" align="left" textpadding="25,0,0,0"
    normalimage="file='themes\radio-dark.png' source='0,0,20,20' dest='0,0,20,20'"
    selectedimage="file='themes\radio-dark.png' source='0,20,20,20' dest='0,0,20,20'" />
  <Style name="check" height="20" align="left" textpadding="25,0,0,0"
    normalimage="file='themes\check-dark.png' source='0,0,20,20' dest='0,0,20,20'"
    selectedimage="file='themes\check-dark.png' source='0,20,20,20' dest='0,0,20,20'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
   <Default name="Label" textcolor="#FFFFFFFF" />
   <Default name="Button" textcolor="#FFFFFFFF" />
   <Style name="ct-bd" bordersize="1" bordercolor="#FF202020" />
   <Style name="ct-bk" bkcolor="#FF191919" />
   <Style name="ct-nav" bkcolor="#FF202020" />
   <Style name="ct-title" bkcolor="#FF202020" />
   <Style name="ct-bdtitle" bordercolor="#FF202020" />
   <Style name="ct-sysbutton" textcolor="#FF8B8B8B" hottextcolor="#FFFFFFFF" />
   <Style name="ct-txtinfo" textcolor="#FF0078D7" />
   <Style name="ct-bkhori" bkcolor="#FF5E5D5C" />
</ColorTheme>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
   <Style shared="true" name="ct-bkinfo" bkcolor="#FF2067B2" />
   <Default name="Label" textcolor="#FFFFFFFF" />
   <Default name="Button" textcolor="#FFFFFFFF" />
   <Style name="ct-bk" bkcolor="#FF363533" />
   <Style name="ct-txtinfo" textcolor="#FF6B6D6A" />
   <Style name="ct-bkhori" bkcolor="#FF5E5D5C" />
   <Style name="ct-today" textcolor="#FFFFFFFF" bkimage="file='themes\today-dark.png'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style name="ct-bkpanel" bkcolor="#FF515151" />
        <Style name="ct-cell" bkcolor="#FF3F3F3F" bordersize="2" bordercolor="#FF515151"
hotbordercolor="#FF777777" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
   <Default defaultfontcolor="#FFFFFFFF" />
   <Image shared="true" name="scrollbar.bmp" file="themes\scrollbar-dark.bmp" />
   <Image shared="true" name="rc\0.png" file="rc\default\0.png" />
   <Image shared="true" name="rc\25.png" file="rc\default\25.png" />
   <Image shared="true" name="rc\50.png" file="rc\default\50.png" />
   <Image shared="true" name="rc\75.png" file="rc\default\75.png" />
   <Image shared="true" name="rc\100.png" file="rc\default\100.png" />
   <Style name="ct-bkdarker1" bkcolor="#FF3F3F3F" />
   <Style name="ct-bkhori" bkcolor="#FF5E5D5C" />
   <Style name="ct-hotimage" hotimage="color='#FF6D6D6D'" />
   <Style name="ct-text-primary" textcolor="#FFFFFFFF" />
   <Style name="ct-text-secondary" textcolor="#FFBBBBBB" />
   <Style name="ct-text-secondary-url" textcolor="#FFBBBBBB" hottextcolor="#FFFFFFFF" />
   <Style name="ct-button" bkcolor="#FF6D6D6D" bordersize="2" bordercolor="#FF6D6D6D"
hotbordercolor="#FFA7A7A7" />
   <Style name="ct-option" bkcolor="#FF656565" selectedimage="color='#FF4A4A4A'" bordersize="2"
bordercolor="#FF3F3F3F" hotbordercolor="#FFA7A7A7" />
   <Style name="ct-list" itemhotbkcolor="#FF525252" itemselectedbkcolor="#FF484848" />
   <Style name="ct-combo"  itemselectedtextcolor="#FFFFFFFF" bkimage="file='themes\ComboButton-dark.png'
corner='2,2,36,2'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style name="ct-bknav" bkcolor="#ffe6e6e6" />
    <Style name="ct-bkcontent" bkcolor="#ffffffff" />
    <Style name="ct-button" textcolor="#FF000000" bkcolor="#ffcccccc"
pushedimage="color='#ff999999'" width="100" height="32"
        bordersize="2" bordercolor="#ffcccccc" hotbordercolor="#ff7a7a7a"
pushedbordercolor="#ff999999" />
    <Style name="ct-combo" bkimage="file='themes\ComboButton.png' corner='2,2,36,2'"
        bordersize="2" bordercolor="#ff999999" focusbordercolor="#ff666666"
        itemselectedtextcolor="#ff000000" itemselectedbkcolor="#ffffffff" itemtextcolor="#ff000000"
itembkcolor="#fff2f2f2"
        itemhottextcolor="#ff000000" itemhotbkcolor="#ffdadada" />
    <Style name="switch" width="44" height="20" normalimage="file='themes\switch.png'
source='0,0,44,20' dest='0,0,44,20'"
        selectedimage="file='themes\switch.png' source='0,20,44,20' dest='0,0,44,20'" />
    <Style name="radio" height="20" align="left" textpadding="25,0,0,0"
        normalimage="file='themes\radio.png' source='0,0,20,20' dest='0,0,20,20'"
        selectedimage="file='themes\radio.png' source='0,20,20,20' dest='0,0,20,20'" />
    <Style name="check" height="20" align="left" textpadding="25,0,0,0"
        normalimage="file='themes\check.png' source='0,0,20,20' dest='0,0,20,20'"
        selectedimage="file='themes\check.png' source='0,20,20,20' dest='0,0,20,20'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Default defaultfontcolor="#FFFFFFFF" />
    <Style name="ct-bk" bkcolor="#FF084F85" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>

   <Style name="ct-bktitle" bkcolor="#FF303F9F" />
   <Style name="ct-bksubtitle" bkcolor="#FF3F51B5" />
   <Style name="ct-bkcontent" bkcolor="#FFF9F9F9" />
   <Style name="ct-bkontent-darker1" bkcolor="#FFF2F2F2" />
   <Style name="attentionBtn" width="120" height="32" textcolor="#FF000000"
       hottextcolor="#FFFFFF" hotimage="color='#FFFF4081'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style name="ct-bkpanel" bkcolor="#FF005A9E" />
        <Style name="ct-cell" bkcolor="#FF0078D7" bordersize="2" bordercolor="#FF005A9E" hotbordercolor="#FF85BBE6" />
</ColorTheme>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style shared="true" name="ct-bkinfo" bkcolor="#FF2067B2" />
    <Default name="Label" textcolor="#FFFFFFFF" />
    <Default name="Button" textcolor="#FFFFFFFF" />
    <Style name="ct-bk" bkcolor="#FF363533" />
    <Style name="ct-txtinfo" textcolor="#FF6B6D6A" />
    <Style name="ct-bkhori" bkcolor="#FF5E5D5C" />
    <Style name="ct-today" textcolor="#FFFFFFFF" bkimage="file='themes\today-dark.png'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Default defaultfontcolor="#FFFFFFFF" />
    <Image shared="true" name="scrollbar.bmp" file="themes\scrollbar.bmp" />
    <Image shared="true" name="rc\0.png" file="rc\default\0.png" />
    <Image shared="true" name="rc\25.png" file="rc\default\25.png" />
    <Image shared="true" name="rc\50.png" file="rc\default\50.png" />
    <Image shared="true" name="rc\75.png" file="rc\default\75.png" />
    <Image shared="true" name="rc\100.png" file="rc\default\100.png" />
    <Style name="ct-bkdarker1" bkcolor="#FF005A9E" />
    <Style name="ct-bkhori" bkcolor="#FF989898" />
    <Style name="ct-hotimage" hotimage="color='#FF338ED6'" />
    <Style name="ct-text-primary" textcolor="#FFFFFFFF" />
    <Style name="ct-text-secondary" textcolor="#FF99C7EB" />
    <Style name="ct-text-secondary-url" textcolor="#FF99C7EB" hottextcolor="#FFFFFFFF" />
    <Style name="ct-button" bkcolor="#FF338ED6" bordersize="2" bordercolor="#FF338ED6"
hotbordercolor="#FF85BBE6" />
    <Style name="ct-option" bkcolor="#FF3378B1" selectedimage="color='#FF0078D7'" bordersize="2"
bordercolor="#FF005A9E" hotbordercolor="#FF85BBE6" />
    <Style name="ct-list" itemhotbkcolor="#FF196AA8" itemselectedbkcolor="#FF006CC0" />
    <Style name="ct-combo"  itemselectedtextcolor="#FFFFFFFF" bkimage="file='themes\ComboButton-
default.png' corner='2,2,36,2'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style name="ct-bknav" bkcolor="#fff3f3f3" />
    <Style name="ct-bkcontent" bkcolor="#ffffffff" />
    <Style name="ct-button" textcolor="#FF000000" bkcolor="#ffcccccc"
pushedimage="color='#ff999999'" width="100" height="32"
        bordersize="2" bordercolor="#ffcccccc" hotbordercolor="#ff7a7a7a"
pushedbordercolor="#ff999999" />
    <Style name="ct-combo" bkimage="file='themes\ComboButton.png' corner='2,2,36,2'"
        bordersize="2" bordercolor="#ff999999" focusbordercolor="#ff666666"
        itemselectedtextcolor="#ff000000" itemselectedbkcolor="#ffffffff" itemtextcolor="#ff000000"
itembkcolor="#fff2f2f2"
        itemhottextcolor="#ff000000" itemhotbkcolor="#ffdadada" />
    <Style name="switch" width="44" height="20" normalimage="file='themes\switch.png'
source='0,0,44,20' dest='0,0,44,20'"
        selectedimage="file='themes\switch.png' source='0,20,44,20' dest='0,0,44,20'" />
    <Style name="radio" height="20" align="left" textpadding="25,0,0,0"
        normalimage="file='themes\radio.png' source='0,0,20,20' dest='0,0,20,20'"
        selectedimage="file='themes\radio.png' source='0,20,20,20' dest='0,0,20,20'" />
    <Style name="check" height="20" align="left" textpadding="25,0,0,0"
        normalimage="file='themes\check.png' source='0,0,20,20' dest='0,0,20,20'"
        selectedimage="file='themes\check.png' source='0,20,20,20' dest='0,0,20,20'" />
</ColorTheme>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style shared="true" name="ct-bkinfo" bkcolor="#FF2067B2" />
    <Default name="Label" textcolor="#FFFFFFFF" />
    <Default name="Button" textcolor="#FFFFFFFF" />
    <Style name="ct-bk" bkcolor="#FF363533" />
    <Style name="ct-txtinfo" textcolor="#FF6B6D6A" />
    <Style name="ct-bkhori" bkcolor="#FF5E5D5C" />
    <Style name="ct-today" textcolor="#FFFFFFFF" bkimage="file='themes\today-dark.png'" />
</ColorTheme>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="14" size="14" name="Segoe UI" default="true" />
    <Font shared="true" id="18" size="18" name="Segoe UI" />
    <Font shared="true" id="24" size="24" name="Segoe UI" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
    <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="Settings" value="Settings" />
    <MultiLanguage id="Home" value="Home" />
    <MultiLanguage id="Customization" value="Customization" />

    <MultiLanguage id="Display" value="Display" />
    <MultiLanguage id="DPI" value="DPI" />
    <MultiLanguage id="Resolution" value="Resolution" />
    <MultiLanguage id="Rotate" value="Rotate" />
    <MultiLanguage id="Rotate_0" value="0 degree" />
    <MultiLanguage id="Rotate_90" value="90 degree" />
    <MultiLanguage id="Rotate_180" value="180 degree" />
    <MultiLanguage id="Rotate_270" value="270 degree" />

    <MultiLanguage id="Brightness" value="Brightness" />

    <MultiLanguage id="Apply" value="<u>A</u>pply" />
    <MultiLanguage id="Restore" value="<u>R</u>estore" />

    <MultiLanguage id="Background" value="Background" />

    <MultiLanguage id="Colors" value="Colors" />
    <MultiLanguage id="SelectSysColorTheme" value="Choose your default Windows mode" />
    <MultiLanguage id="SelectAppColorTheme" value="Choose your default app mode" />
    <MultiLanguage id="Light" value="Light" />
    <MultiLanguage id="Dark" value="Dark" />
    <MultiLanguage id="TransparencyEffects" value="Transparency effects" />
    <MultiLanguage id="ApplyAccentColorDesc" value="Show accent color on the following surfaces" />
    <MultiLanguage id="StartTaskbarActionCenter" value="Start, taskbar, and action center" />
    <MultiLanguage id="TitlebarsAndWinBorders" value="Title bars and window borders" />

    <MultiLanguage id="Taskbar" value="Taskbar" />
    <MultiLanguage id="AutoHideTaskbar" value="Automatically hide the taskbar" />
    <MultiLanguage id="UseSmallButtons" value="Use small taskbar buttons" />
    <MultiLanguage id="CombineButtons" value="Combine taskbar buttons" />
    <MultiLanguage id="Always" value="Always, hide the labels" />
    <MultiLanguage id="Auto" value="When taskbar is full" />
    <MultiLanguage id="Never" value="never" />

    <MultiLanguage id="On" value="On" />
    <MultiLanguage id="Off" value="Off" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="12" size="12" name="Segoe UI"  shared="true"  default="true" />
    <Font id="14" size="14" name="Segoe UI"  shared="true" />
    <Font id="16" size="16" name="Segoe UI"  shared="true" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="18" name="Segoe UI"  shared="true"  default="true" />
    <Font id="24" size="24" name="Segoe UI"  shared="true" />
    <Font shared="true" id="40b" size="40b" name="Segoe UI" />
    <Font shared="true" id="128" size="28" name="Segoe MDL2 Assets" />
    <Font shared="true" id="130" size="30" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="Sunday" />
    <MultiLanguage id="info_w1" value="Monday" />
    <MultiLanguage id="info_w2" value="Tuesday" />
    <MultiLanguage id="info_w3" value="Wednesday" />
    <MultiLanguage id="info_w4" value="Thursday" />
    <MultiLanguage id="info_w5" value="Friday" />
    <MultiLanguage id="info_w6" value="Saturday" />

    <MultiLanguage id="SignIn" value="Sign in" />
    <MultiLanguage id="ForgotPassword" value="Forgot password" />
    <MultiLanguage id="Logon" value="Logon" />

    <MultiLanguage id="will autologon with the $user account in $countdown second(s)" value="will
autologon with the $user account in $countdown second(s)" />
    <MultiLanguage id="Welcome to Windows ..." value="Welcome to Windows ..." />

   <MultiLanguage id="LogonInfo" value="Press Alt+S to logon with the SYSTEM account, and Alt+A to
logon with the Administrator account." />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="14" size="14" name="Segoe UI" default="true" />
    <Font shared="true" id="24" size="24" name="Segoe UI" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
    <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="Settings" value="Settings" />
    <MultiLanguage id="Home" value="Home" />
    <MultiLanguage id="Customization" value="Customization" />

    <MultiLanguage id="Display" value="Display" />
    <MultiLanguage id="DPI" value="DPI" />
    <MultiLanguage id="Resolution" value="Resolution" />
    <MultiLanguage id="Rotate" value="Rotate" />
    <MultiLanguage id="Rotate_0" value="0 degree" />
    <MultiLanguage id="Rotate_90" value="90 degree" />
    <MultiLanguage id="Rotate_180" value="180 degree" />
    <MultiLanguage id="Rotate_270" value="270 degree" />

    <MultiLanguage id="Brightness" value="Brightness" />

    <MultiLanguage id="Apply" value="<u>A</u>pply" />
    <MultiLanguage id="Restore" value="<u>R</u>estore" />

    <MultiLanguage id="Background" value="Background" />

    <MultiLanguage id="Colors" value="Colors" />
    <MultiLanguage id="SelectSysColorTheme" value="Choose your default Windows mode" />
    <MultiLanguage id="SelectAppColorTheme" value="Choose your default app mode" />
    <MultiLanguage id="Light" value="Light" />
    <MultiLanguage id="Dark" value="Dark" />
    <MultiLanguage id="TransparencyEffects" value="Transparency effects" />
    <MultiLanguage id="ApplyAccentColorDesc" value="Show accent color on the following surfaces" />
    <MultiLanguage id="StartTaskbarActionCenter" value="Start, taskbar, and action center" />
    <MultiLanguage id="TitlebarsAndWinBorders" value="Title bars and window borders" />

    <MultiLanguage id="Taskbar" value="Taskbar" />
    <MultiLanguage id="AutoHideTaskbar" value="Automatically hide the taskbar" />
    <MultiLanguage id="UseSmallButtons" value="Use small taskbar buttons" />
    <MultiLanguage id="CombineButtons" value="Combine taskbar buttons" />
    <MultiLanguage id="Always" value="Always, hide the labels" />
    <MultiLanguage id="Auto" value="When taskbar is full" />
    <MultiLanguage id="Never" value="never" />

    <MultiLanguage id="On" value="On" />
    <MultiLanguage id="Off" value="Off" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="16" name="Segoe UI"  shared="true"  default="true" />
    <Font id="24" size="24" name="Segoe UI"  shared="true" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="12" size="12" name="Segoe UI" default="true" />
    <Font shared="true" id="16" size="16" name="Segoe UI" />
    <Font shared="true" id="24" size="24" name="Segoe UI" />

    <MultiLanguage id="info_w0" value="Sunday" />
    <MultiLanguage id="info_w1" value="Monday" />
    <MultiLanguage id="info_w2" value="Tuesday" />
    <MultiLanguage id="info_w3" value="Wednesday" />
    <MultiLanguage id="info_w4" value="Thursday" />
    <MultiLanguage id="info_w5" value="Friday" />
    <MultiLanguage id="info_w6" value="Saturday" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
   <Font id="16" size="16" name="Segoe UI"  shared="true"  default="true" />
   <Font id="24" size="24" name="Segoe UI"  shared="true" />

<!-- The description of <Style imagerectstyle="l,t,w,h" /> in main.xml -->
   <!-- image's source, dest attributes will define with "left,top,width,height" style -->
   <!-- the origin style is "left,top,right,bottom". -->
   <!-- the xml file for "Include" tag is also use the same "imagerectstyle" -->

   <MultiLanguage id="Power" value="Power" />
   <MultiLanguage id="CloseComputer" value="Close PC" />
   <MultiLanguage id="Shutdown" value="Shutdown" />
   <MultiLanguage id="Restart" value="Restart" />
   <MultiLanguage id="Hibernate" value="Hibernate" />
   <MultiLanguage id="Standby" value="Standby" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="16" name="Segoe UI"  shared="true"  default="true" />
    <Font id="24" size="24" name="Segoe UI"  shared="true" />

    <MultiLanguage id="Settings" value="Settings" />
    <MultiLanguage id="Display Settings" value="Display Settings" />
    <MultiLanguage id="Resolution" value="Resolution" />
    <MultiLanguage id="Rotate" value="Rotate" />
    <MultiLanguage id="Rotate_0" value="0 degree" />
    <MultiLanguage id="Rotate_90" value="90 degree" />
    <MultiLanguage id="Rotate_180" value="180 degree" />
    <MultiLanguage id="Rotate_270" value="270 degree" />
    <MultiLanguage id="Brightness" value="Brightness" />
    <MultiLanguage id="Apply" value="<u>A</u>pply" />
    <MultiLanguage id="Close" value="<u>C</u>lose" />
    <MultiLanguage id="Restore" value="<u>R</u>estore" />
    <MultiLanguage id="Cancel" value="<u>C</u>ancel" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="16" name="Segoe UI"  shared="true"  default="true" />
    <Font id="24" size="24" name="Segoe UI"  shared="true" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
    <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="10" size="10" name="Segoe UI" shared="true" />
    <Font id="16" size="16" name="Segoe UI" shared="true" default="true" />
    <Font id="12" size="12" name="Segoe UI" shared="true" />
    <Font id="24" size="24" name="Segoe UI" shared="true" />
    <Font id="16u" size="16" name="Segoe UI" shared="true" underline="true" />

    <MultiLanguage id="::Connected" value="Connected" />
    <MultiLanguage id="::Secured" value="Secured" />
    <MultiLanguage id="::Open" value="Open" />
    <MultiLanguage id="::Connect" value="Connect" />
    <MultiLanguage id="::Disconnect" value="Disconnect" />
    <MultiLanguage id="msg_hidden_ssid" value="Hidden Network" />
    <MultiLanguage id="msg_enter_ssid" value="Enter the name(SSID) for the network" />
    <MultiLanguage id="msg_enter_password" value="Enter the network security key" />
    <MultiLanguage id="Cancel" value="Cancel" />
    <MultiLanguage id="NetworkSettings" value="Network & Internet settings" />
    <MultiLanguage id="NetworkSettingsMsg" value="Change settings, such as setting a static IP address." />
    <MultiLanguage id="Firewall" value="Firewall" />
    <MultiLanguage id="Proxy" value="Proxy" />
    <MultiLanguage id="::msg_select_device" value="Select the WLAN device" />
    <MultiLanguage id="::msg_service_unavailable" value="WLAN service is unavailable" />
    <MultiLanguage id="::msg_device_unavailable" value="WLAN device is unavailable" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="16" size="16" name="Segoe UI" default="true" />
    <Font shared="true" id="16b" size="16" name="Segoe UI" bold="true" />
    <Font shared="true" id="18" size="18" name="Segoe UI" />
    <Font shared="true" id="18b" size="18" name="Segoe UI" bold="true" />
    <Font shared="true" id="20" size="20" name="Segoe UI" />
    <Font shared="true" id="40" size="40b" name="Segoe UI" />
    <Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="Sunday" />
    <MultiLanguage id="info_w1" value="Monday" />
    <MultiLanguage id="info_w2" value="Tuesday" />
    <MultiLanguage id="info_w3" value="Wednesday" />
    <MultiLanguage id="info_w4" value="Thursday" />
    <MultiLanguage id="info_w5" value="Friday" />
    <MultiLanguage id="info_w6" value="Saturday" />

    <MultiLanguage id="w7" value="Su" />
    <MultiLanguage id="w1" value="Mo" />
    <MultiLanguage id="w2" value="Tu" />
    <MultiLanguage id="w3" value="We" />
    <MultiLanguage id="w4" value="Th" />
    <MultiLanguage id="w5" value="Fr" />
    <MultiLanguage id="w6" value="Sa" />

    <MultiLanguage id="OK" value="OK" />
    <MultiLanguage id="CANCEL" value="CANCEL" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
   <Font id="16" size="16" name="Segoe UI"  shared="true"  default="true" />
   <Font id="24" size="24" name="Segoe UI"  shared="true" />

   <MultiLanguage id="Settings" value="Ajustes" />
   <MultiLanguage id="Display Settings" value="Ajustes de Pantalla" />
   <MultiLanguage id="Resolution" value="Resolución" />
   <MultiLanguage id="Rotate" value="Rotar" />
   <MultiLanguage id="Rotate_0" value="0 grados" />
   <MultiLanguage id="Rotate_90" value="90 grados" />
   <MultiLanguage id="Rotate_180" value="180 grados" />
   <MultiLanguage id="Rotate_270" value="270 grados" />
   <MultiLanguage id="Brightness" value="Brillo" />
   <MultiLanguage id="Apply" value="<u>A</u>plicar" />
   <MultiLanguage id="Close" value="<u>C</u>lose" />
   <MultiLanguage id="Restore" value="<u>R</u>estore" />
   <MultiLanguage id="Cancel" value="<u>C</u>ancelar" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
   <Font id="16" size="16" name="Segoe UI"  shared="true"  default="true" />
   <Font id="24" size="24" name="Segoe UI"  shared="true" />

   <MultiLanguage id="Power" value="Apagar" />
   <MultiLanguage id="CloseComputer" value="Cerrar Ordenador" />
   <MultiLanguage id="Shutdown" value="Apagar" />
   <MultiLanguage id="Restart" value="Reiniciar" />
   <MultiLanguage id="Hibernate" value="Hibernar" />
   <MultiLanguage id="Standby" value="Suspender" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="10" size="10" name="Segoe UI" shared="true" />
    <Font id="16" size="16" name="Segoe UI" shared="true" default="true" />
    <Font id="12" size="12" name="Segoe UI" shared="true" />
    <Font id="24" size="24" name="Segoe UI" shared="true" />
    <Font id="16u" size="16" name="Segoe UI" shared="true" underline="true" />

    <MultiLanguage id="::Connected" value="Conectado" />
    <MultiLanguage id="::Secured" value="Segura" />
    <MultiLanguage id="::Open" value="Abierta" />
    <MultiLanguage id="::Connect" value="Conectar" />
    <MultiLanguage id="::Disconnect" value="Desconectar" />
    <MultiLanguage id="msg_hidden_ssid" value="Hidden Network" />
    <MultiLanguage id="msg_enter_ssid" value="Enter the name(SSID) for the network" />
    <MultiLanguage id="msg_enter_password" value="Introducir contraseña" />
    <MultiLanguage id="Cancel" value="Cancelar" />
    <MultiLanguage id="NetworkSettings" value="Configuración de red e Internet" />
    <MultiLanguage id="NetworkSettingsMsg" value="Cambia los ajustes de configuración y tarjeta red."
/>
    <MultiLanguage id="Firewall" value="Firewall" />
    <MultiLanguage id="Proxy" value="Proxy" />
    <MultiLanguage id="::msg_select_device" value="Seleccionar tarjeta red" />
    <MultiLanguage id="::msg_service_unavailable" value="Servicio WLAN no disponible" />
    <MultiLanguage id="::msg_device_unavailable" value="Tarjeta WLAN no disponible" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="14" size="14" name="Segoe UI" default="true" />
    <Font shared="true" id="24" size="24" name="Segoe UI" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
    <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="Settings" value="Ajustes" />
    <MultiLanguage id="Home" value="Inicio" />
    <MultiLanguage id="Customization" value="Personalizar" />

    <MultiLanguage id="Display" value="Pantalla" />
    <MultiLanguage id="Resolution" value="ResoluciÃ³n" />
    <MultiLanguage id="Rotate" value="Rotar" />
    <MultiLanguage id="Rotate_0" value="0 grados" />
    <MultiLanguage id="Rotate_90" value="90 grados" />
    <MultiLanguage id="Rotate_180" value="180 grados" />
    <MultiLanguage id="Rotate_270" value="270 grados" />

    <MultiLanguage id="Brightness" value="Brillo" />

    <MultiLanguage id="Apply" value="<u>A</u>plicar" />
    <MultiLanguage id="Restore" value="<u>R</u>estore" />

    <MultiLanguage id="Background" value="Fondo de pantalla" />

    <MultiLanguage id="Colors" value="Tema" />
    <MultiLanguage id="SelectAppColorTheme" value="Elegir el color por defecto" />
    <MultiLanguage id="Light" value="Claro" />
    <MultiLanguage id="Dark" value="Oscuro" />

    <MultiLanguage id="Taskbar" value="Barra de tareas" />
    <MultiLanguage id="AutoHideTaskbar" value="Esconder automÃ¡ticamente la barra de tareas" />
    <MultiLanguage id="UseSmallButtons" value="Utilizar iconos pequeÃ±os en la barra de tareas" />
    <MultiLanguage id="CombineButtons" value="Combinar botones en la barra de tareas" />
    <MultiLanguage id="Always" value="Siempre" />
    <MultiLanguage id="Auto" value="Cuando estÃ© llena" />
    <MultiLanguage id="Never" value="Nunca" />

    <MultiLanguage id="On" value="Activar" />
    <MultiLanguage id="Off" value="Desactivar" />
</Locale>
```

```
{
 "name":"UI_Shutdown",
 "title":"%{Power}",
 "nobaricon":true,
 "baricon":"power.ico",
 "style":0,
 "exstyle":8,
 "entry":"full.xml",
 "lua":"full.lua"
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\full.jcfg",
}
```

```
{
 "name":"UI_Launcher",
 "singleton":true,
 "lua":"main-full.lua",
 "nobaricon":true,
 "position":"leftbottom",
 "startup_win":"max",
 "#OnDeactive":"hide"
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Launcher\full.jcfg",
}
```

```
local power_clicked = false

function onload()
    local wp = ''
    local param = App.CmdLine
    local blur = string.match(param, '-blur%s([0-9\\.]+)%s?')
    if blur == nil then blur = '0.0' end

    wp = string.match(param, '-bk%s(.+)$')
    if wp == nil then wp = Desktop:GetWallpaper() end

    local win = sui:find('full_win')
    win.bkimage = "file='" .. wp .. "' blur='" .. blur .. "'"
    sui:moveto(0, 0, 2000, 2000, -1, 64)
end

function onclick(ctrl)
  if power_clicked then return end
  if ctrl == "restartbtn" then
    power_clicked = true
    System:Reboot()
  elseif ctrl == "shutdownbtn" then
    power_clicked = true
    System:Shutdown()
  end
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\full.lua"
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="320,300" caption="0,0,0,32" defaultfontcolor="#FFE8EAF6">

    <Style imagerectstyle="l,t,w,h" />

    <Style name="PowerBtn" value="font=&quot;24&quot; textpadding=&quot;0,136,0,0&quot;
width=&quot;128&quot; bkimage=&quot;file='$1B.png' dest='32,32,64,64'&quot;
hotimage=&quot;file='$1.png' dest='32,32,64,64'&quot; textcolor=&quot;#FF000000&quot;
hottextcolor=&quot;#FFFFFF&quot;" />
    <!-- <VerticalLayout bkimage="file='bk.jpg' blur='20.0'"> -->
    <VerticalLayout name="full_win" bkimage="file='blur.jpg' blur='0.0'">
        <VerticalLayout>
            <Control />
                <HorizontalLayout height="180">
                    <Control />
                    <HorizontalLayout bkimage="file='cbtn128.png' fade='176' dest='0,0,128,128'"
width="128">
                        <Button name="restartbtn" text="%{Restart}" style="PowerBtn:reboot" />
                    </HorizontalLayout>
                        <Control width="50" />
                    <HorizontalLayout bkimage="file='cbtn128.png' fade='176' dest='0,0,128,128'"
width="128">
                        <Button name="shutdownbtn" text="%{Shutdown}" style="PowerBtn:shutdown" />
                    </HorizontalLayout>
                    <Control />
                </HorizontalLayout>
            <Control />
        </VerticalLayout>
    </VerticalLayout>
</Window>
```

```
temp = os.getenv("temp")
systemdrive = os.getenv("systemdrive")
local fwbootmgr_menu_table = {}
local exitcode, stdout = winapi.execute("cmd /c bcdedit /enum all")
stdout = stdout:gsub("\r\n", "\n")
--一次性启动命令Bcdedit /set {fwbootmgr} bootsequence {d0a72d68-494c-11ed-badf-00505626a58c} /addlast
--第一项命令Bcdedit /set {fwbootmgr} displayorder {d0a72d68-494c-11ed-badf-00505626a58c} /addfirst

-- 正则匹配获取多行
function string.gfind(stdout, patten)
    local i, j = 0, 0
    return function()
        i, j = string.find(stdout, patten, j + 1)
        if (i == nil) then -- end find
            return nil
        end
        return string.sub(stdout, i, j)
    end
end


-- 提取固定值
function bcdinit()

--初始化一些东西
for def_str in string.gfind(stdout, "标识符                    {default}*.-\n\n") do
def_stdout = def_str
end
def_stdout = def_stdout:gsub("\r\n", "\n")
```

```
for bootmgr_str in string.gfind(stdout, "标识符                    {bootmgr}*.-\n\n") do
bootmgr_stdout = bootmgr_str
end
bootmgr_stdout = bootmgr_stdout:gsub("\r\n", "\n")


for fwbootmgr_str in string.gfind(stdout, "{fwbootmgr}*.-\n\n") do
fwbootmgr_menu = fwbootmgr_str
end
end

function get_fwboot_description(fwboot_guid_regex, stdout)
if guid_table.id:find("{bootmgr}") then
desc_regex = bootmgr_stdout:match("{bootmgr}[^*]+\ndescription([^\n]+)\n")
else
desc_regex = fwboot_guid_regex .. "\ndescription([^\n]+)\n"
end
for desc_str in stdout:gmatch(desc_regex) do
if desc_str ¯= nil then
guid_table.desc = desc_str

table.insert(fwbootmgr_menu_table, string.format("UEFI序列%s \n GUID:%s \n 设备%s\n",
guid_table.item, guid_table.id, guid_table.desc))
end
end
end

function get_bootmgr_def()
bootmgr_def = def_stdout:match("{default}[^*]+\ndescription([^\n]+)\n")
if bootmgr_def ¯= nil then
guid_table.def = bootmgr_def
```

```
else
guid_table.def = "没有安装操作系统"
end
if stdout:find("{19260817--6666--8888--f00d--caffee000000}") then
NewBeePE_install = "       NewBeePE 已安装"
else
NewBeePE_install = "     当前没有安装NewBeePE"
end
end

function get_bcdedit()
fwboot_item = 0
guid_table = {}
for token in string.gmatch(fwbootmgr_menu, " {([^\n]+)\n") do
fwboot_item = fwboot_item + 1
guid_table.item = fwboot_item
fwboot_guid = string.format("{%s", token)
guid_table.id = fwboot_guid
fwboot_guid_regex = "" .. fwboot_guid
if fwboot_guid:find("-") then
fwboot_guid_regex = fwboot_guid:gsub("-", "--")
end
get_fwboot_description(fwboot_guid_regex, stdout)
end
end

bcdinit()
get_bcdedit()
get_bootmgr_def()

fwbootmgr_menu = table.concat(fwbootmgr_menu_table, "\n")
fwbootmgr_menu = string.format(fwbootmgr_menu .. "\n本机离线系统状态:\n默认启动操作系统:" ..
guid_table.def .. "\nNewBeePE安装状态:" .. NewBeePE_install)
alert(fwbootmgr_menu)
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_bcdedit.lua"
```

```
regkey=[[HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\]]

local subkeys = Reg:GetSubKeys(regkey)
for i, v in ipairs(subkeys ) do
  v = reg_read(regkey .. v, {'DhcpIPAddress', 'DhcpServer'})
alert("获得IP地址" .. v['DhcpIPAddress'])

end

--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_getofflineip.lua"
```

```
--下载相关添加
local getfullpath_func = type(File.GetFullPath)
if getfullpath_func == "function" then
temp = File.GetFullPath(os.getenv("temp"))
else
temp = os.getenv("temp")
end
SystemDrive = os.getenv("SystemDrive")
local fixed_speed_out = {}
local pid = winapi.get_current_pid()
local logfile = pid .. ".log"
TIMER_ID_QUIT = 1002
TIMER_ID_LOADING = 10088
TIMER_ID_CHECK_INSTALLED = 10089
eta_str = 0
dl_str = 0
app_runpath = "temp"
downpath = temp
set_downpath = temp
app_setup_args = "/S"
process_val = 0
function winntsetup()
    install_esd = [[X:\\Program Files\\WinNTSetup\\winntsetup.exe]]
    if File.exists("" .. install_esd .. "") then
        exec("/hide",
            [[cmd /c ftype .esd="]] ..
            install_esd .. [[" nt6 /source:"%1" /unattend:"unattend\内置管理员用户
\X64\Panther\Unattend.xml"]])
        exec("/hide",
            [[cmd /c ftype .wim="]] ..
            install_esd .. [[" nt6 /source:"%1" /unattend:"unattend\内置管理员用户
\X64\Panther\Unattend.xml"]])
        exec("/hide",
            [[cmd /c ftype 7-Zip.wim="]] ..
            install_esd .. [[" nt6 /source:"%1" /unattend:"unattend\内置管理员用户
\X64\Panther\Unattend.xml"]])
    end
end

function ckadmin()
    --if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
        if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
        toadmin = winapi.show_message("受限!没有以管理员身份运行!功能无法正常使用", "请右键管
理员身份运行本程序!,\n是否现在就尝试管理员身份重新运行本程序?"
            , "yes-no", "warning")
    end
    if toadmin == "yes" then
        --exec('/hide', [[cmd /c shutdown -a]])
        --winapi.show_message('注意', "执行管理员提升权限操作")
        exec("/admin", [["]] .. APP_Path .. [[\WinXshell.exe" -ui -jcfg
wxsUI\UI_AppStore\main.jcfg]])

        sui:close()
        return
    elseif toadmin == "no" then
        return
    end
end

--other
--下载相关
```

```
function ontimer(id)
    if id == TIMER_ID_QUIT then
        sui:close()
    elseif id == TIMER_ID_LOADING then
        loading()
    elseif id == TIMER_ID_CHECK_INSTALLED then
        check_installed()
    end
end

--下载开始
function string.gfind(stdout, patten)
    local i, j = 0, 0
    return function()
        i, j = string.find(stdout, patten, j + 1)
        if (i == nil) then -- end find
            return nil
        end
        return string.sub(stdout, i, j)
    end
end

function loading()
    local path = temp .. "\\" .. logfile
    local file = io.open(path, "r")
    if file == nil then
        App_STAR.text = "准备就绪!"
        App_ProgressBar.value = 0
    else
        local text = file:read("*a")
        file:close()
        local regex = "([^\n]+)[\n]*￥" -- 匹配最后一行有效文本 ￥表示匹配字符串结尾位置
        complete = text:match("Download complete: ([^] %s]+)")
        for all_str in string.gfind(text, "[[][#].-[]]") do
            table.insert(fixed_speed_out, all_str)
        end
        down_str = table.concat(fixed_speed_out, "\n\n")
        --winapi.show_message("最后一个", down_str)
        --文件判断语句备份
        if complete ‾= nil then
            eta_str = "0S"
            App_STAR.text = "完成!"
            App_ProgressBar.value = "100"
            process_val = "100"
            check_app()
        end
        for loading_str in string.gmatch(down_str, regex) do
            dl_str = loading_str:match("DL:([^] %s]+)")
            eta_str = loading_str:match("ETA:([^]]+)")
            process_val = loading_str:match("[((]([^%%]+)")
            if process_val ‾= nil and eta_str ‾= nil and process_val < "95" then
                App_STAR.text = "已经完成:" .. process_val .. "% "
                --App_STAR.text = "已经完成:" .. process_val .. "% 速度:" .. dl_str .. "/S 剩
余时间:" .. eta_str
                App_ProgressBar.value = process_val
            end
        end
    end
    --local regex = "([^\n]+)￥"
end

function check_app()
```

```
    App_ProgressBar.value = 100
    process_val = 100
    suilib.call("KillTimer", 10088)
    if File.exists(downpath .. "\\" .. app_setup) and app_runpath ~= "temp" and winpe == nil
then
        App_STAR.text = "正在安装... "
        --App_STAR.text= "正在安装... [" .. app_setup .. "]"
        exec("/hide", [[cmd /c start "" ]] .. downpath .. [[\]] .. app_setup .. " " ..
app_setup_args)
        check_installed()
    elseif File.exists(downpath .. "\\" .. app_setup) and app_runpath == "temp" and winpe ==
nil then
        --winapi.show_message("check_app", "完成")
        App_STAR.text = "执行程序.. "
        --App_STAR.text= "请安装[" .. app_setup .. "]"
        exec("/hide", [[cmd /c start "" ]] .. downpath .. [[\]] .. app_setup .. " " ..
app_setup_args)
        --完成后关闭sui:close()
                         suilib.call("KillTimer", 1002)
                  suilib.call("KillTimer", 10089)
    elseif winpe ~= nil then
        suilib.call("KillTimer", 10088)
        pepath = downpath .. [[\]] .. app_setup
        pepath = pepath:gsub(":\\\\", ":\\")
        addfile = "wim"
        ntboot_ask(pepath)
        --完成后关闭 sui:close()
                  suilib.call("KillTimer", 1002)
                  suilib.call("KillTimer", 10089)
    end
end
```

```
function check_installed()

    if File.exists(app_runpath) and app_runpath ~= "temp" and file == nil and winpe == nil then
        App_STAR.text = "安装完成"
        exec("/hide", [[cmd /c start "" "]] .. app_runpath .. [["]])
        suilib.call("KillTimer", 10088)
        Desktop:Refresh()
        exec("/hide", [[cmd /c del /q ]] .. downpath .. [[\]] .. app_setup .. [["]])
        --suilib.call("KillTimer", 10089)
        --完成后关闭 sui:close()
                         suilib.call("KillTimer", 1002)
                 suilib.call("KillTimer", 10089)
    elseif File.exists(app_runpath) and app_runpath ~= "temp" and file ~= nil and winpe == nil then
        App_STAR.text = "安装完成"
        --winapi.show_message("下载完成后播放文件:", app_file)
        suilib.call("KillTimer", 10088)
        Desktop:Refresh()
        exec("/hide", [[cmd /c del /q ]] .. downpath .. [[\]] .. app_setup .. [["]])
        exec("/hide", [[cmd /c start "" "]] .. file .. [["]])
                --完成后关闭
        --sui:close()
                         suilib.call("KillTimer", 1002)
                 suilib.call("KillTimer", 10089)

    end
end

function app_init()
    App_ProgressBar = sui:find("App_ProgressBar")
```

```
--如果是pe，调用ntboot


--不安装的程序如果存在直接运行
if File.exists(downpath .. [[\]] .. app_setup) and app_runpath == "temp" and winpe == nil or app_runpath == nil then
    exec("/hide", [[cmd /c start "" ]] .. downpath .. [[\]] .. app_setup)
elseif File.exists(downpath .. [[\]] .. app_setup) and winpe ‾= nil then
        pepath = downpath .. [[\]] .. app_setup
    pepath = pepath:gsub(":\\\\", ":\\")
    addfile = "wim"
    ntboot_ask(pepath)
    --完成后关闭sui:close()
                    suilib.call("KillTimer", 1002)
            suilib.call("KillTimer", 10089)
end
--sui:close()


if app_url ‾= nil or user_agent ‾= nil then
    exec(
        "/hide",
 [[cmd /c aria2c --file-allocation=falloc -x 16 -c --allow-overwrite=true ]] ..
        aria2c_args ..
        [[ ]] ..
        user_agent .. [[ "]] .. app_url .. [[" -d ]] .. downpath .. [[ -o ]] .. app_setup .. [[>%temp%\]] .. logfile
    )
end
App_ProgressBar = sui:find("¥App[" .. ITEMID .. "]ProgressBar")
App_STAR.text = "正在下载"
App_ProgressBar.visible = "1"
```

```lua
    App_ProgressBar.value = 0
    suilib.call("SetTimer", TIMER_ID_CHECK_INSTALLED, 2000)
    suilib.call("SetTimer", TIMER_ID_LOADING, 1000)
end

---下载初始化结束

function storedl(app_url, user_agent, app_setup, app_setup_args, app_runpath, aria2c_args)
    app_init()
end

function storedl_analysis(
    app_id,
    direct_url,
    user_agent,
    app_setup,
    app_setup_args,
    app_runpath,
    aria2c_args,
    download_page)
    exec(
        "/wait /hide",
        [[cmd /c aria2c ]] ..
        user_agent .. [[ --allow-overwrite=true ]] .. download_page .. [[ -d %temp% -o ]] .. app_id .. ".dl"
    )
    local path = temp .. "\\" .. app_id .. ".dl"
    tmpfile = io.open(path, "r")
    html = tmpfile:read("*a")
    --winapi.show_message("解析下载", "html代码" .. [["]] .. direct_url .. [["]])
    tmpfile:close()
    --下面的解析是RustDesk特有的，如果有影响就删除!!!!!!!!!!
    html = html:gsub('\\/', '/')
    app_url = html:match(direct_url)

        --如果是匹配了不存在http前缀的类型，加上https://前缀
    if direct_url_prefix ‾= nil then
        app_url = direct_url_prefix .. app_url
        end
    if app_url == nil then
        winapi.show_message("错误", "解析失败")
    else
        app_init()
    end
end

function onlick_app(app_click, appmode)

    ITEMID = AppInfo[app_click].id
    App_STAR = sui:find("¥App[" .. ITEMID .. "]STAR")
    pemode = AppInfo[app_click].pemode
    winpe = AppInfo[app_click].winpe
    adminmode = AppInfo[app_click].adminmode
    --pemode值如果不为空 当前又不是pe，则不运行
    if pemode ‾= nil and SystemDrive ‾= "X:" then
        msg("当前环境不支持运行本软件，本工具只能在PE下运行!")
        return
    end
    if adminmode ‾= nil then
        ckadmin()
         end
        if toadmin == "yes" then return end
    --如果是nbapp
    if appmode == "nbapp" then
        nbapp = AppInfo[app_click].nbapp
        exec("/show", [[WinXShell.exe -ui -jcfg wxsUI\\UI_DL\\main.jcfg -nbapp ]] .. nbapp)
    elseif appmode == "nbapp_file" then
        --如果是本地应用
        App_STAR.text = "安装后打开" .. file
        exec("/show", [[WinXShell.exe -ui -jcfg wxsUI\\UI_DL\\main.jcfg -nbapp ]] .. nbapp .. [[ -file ]] .. file)
    elseif appmode == "runapp" then
        --如果是商店直链
        runapp = AppInfo[app_click].app_runpath
        App_STAR.text = "正在运行"
        exec(runapp)
        --  if AppStore.config["close_runapp_clicked"] == 1 then
        --     sui:close()
        -- elseif AppStore.config["hide_runapp_clicked"] == 1 then
        --    sui:hide()
        -- end
    elseif appmode == "StoreApp_direct_url" then
        --  --如果是商店解析链
        app_url = AppInfo[app_click].direct_url
        app_runpath = AppInfo[app_click].app_runpath
        app_setup_args = AppInfo[app_click].app_setup_args
        app_setup = AppInfo[app_click].app_setup
        user_agent = AppInfo[app_click].user_agent
        aria2c_args = AppInfo[app_click].aria2c_args
        storedl(app_url, user_agent, app_setup, app_setup_args, app_runpath, aria2c_args)
    elseif appmode == "StoreApp_analysis_url" then
        App_STAR.text = "正在解析"
        app_id = AppInfo[app_click].id
        direct_url = AppInfo[app_click].direct_url
                direct_url_prefix = AppInfo[app_click].direct_url_prefix
        app_runpath = AppInfo[app_click].app_runpath
        app_setup_args = AppInfo[app_click].app_setup_args
        app_setup = AppInfo[app_click].app_setup
        user_agent = AppInfo[app_click].user_agent
        aria2c_args = AppInfo[app_click].aria2c_args
        download_page = AppInfo[app_click].download_page
        storedl_analysis(
            app_id,
            direct_url,
            user_agent,
            app_setup,
            app_setup_args,
            app_runpath,
            aria2c_args,
            download_page
        )
    elseif appmode == "wimboot" then
        addfile = "raw"
                Add_Boot_Option(addfile)

    elseif appmode == "vhdboot" then
        --什么都不是
                addfile = "vhd"
                Add_Boot_Option(addfile)
        elseif appmode == "StoreApp_BaiPiao" then
        BaiPiaoname = AppInfo[app_click].name
        BaiPiaoid = AppInfo[app_click].id
        BaiPiaourl = AppInfo[app_click].download_page
    dofile ("wxsUI\\UI_AppStore\\nbtool_nbdl_BaiPiao.lua")
```

```
    elseif app_click ~= nil and appmode == nil then
        msg("无法使用，缺少info.appmode相关项")
    end
end

function helper_nbdl_onload()
    winntsetup()
    --onclick('¥App[qq]')
          if File.exists([[diy_nav.lua]]) then
        dofile ("diy_nav.lua")
          else
        onclick("¥Nav[5]")

    end

end
function helper_nbdl_onclick(ctrl)

    if ctrl == "set_button" then
        set_downpath = Dialog:BrowseFolder('选择下载保存位置， 当前： ' ..
            downpath .. "\n 下载系统镜像时尽量修改保存位置或设置虚拟内存 ", 15)
    end
    if set_downpath == "" then
        downpath = temp
    else
        downpath = set_downpath
    end
    --for _dlhelper_start
    local app_click = ctrl:match("¥App%[(.+)%]")
    if app_click ~= nil then
        appmode = AppInfo[app_click].appmode
        onlick_app(app_click, appmode)
    end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_nbdl.lua"
```

```
local ntbootfile_ext = ""

local bootfile = ""

local getfullpath_func = type(File.GetFullPath)

if getfullpath_func == "function" then

temp = File.GetFullPath(os.getenv("temp"))

else

temp = os.getenv("temp")

end

systemdrive = os.getenv("systemdrive")


-- 提取固定值

local exitcode, stdout = winapi.execute("cmd /c mountvol")

stdout = stdout:gsub("\r\n", "\n")

get_BIOS_str = stdout:match("EFI ([^\n]+)\n")




-- 获取系统版本

function get_winver()

    local exitcode, stdout = winapi.execute([[ver]])

    local winver_str = stdout:match("10.0.")

    -- stdout = stdout:gsub("\r\n", "\n")

    -- winapi.show_message('错误', winver_str)

    if winver_str ~= nil then

        winver = "Win10/Win11"
    else
        winver = "win7"
    end
end

-- 检测环境
function ck_BIOS()
    if get_BIOS_str == nil then
        BIOS_str = "传统BIOS"

        boot_platform = "BIOS"
        wimwinload = [[\Windows\system32\boot\winload.exe]]
        vhdwinload = [[\Windows\system32\winload.exe]]

    else
            bcd_drv = "B:"
        BIOS_str = "UEFI"

        boot_platform = "UEFI"
        wimwinload = [[\Windows\system32\boot\winload.efi]]
        vhdwinload = [[\Windows\system32\winload.efi]]
    end
end
```

```
ck_BIOS()

App_platform = sui:find("boot_platform")
App_bt_text = sui:find("bt_text")
App_platform.text = "当前环境:" .. BIOS_str

--执行批处理代码
function execlines(cmds)

for line in cmds:gmatch("([^\n]*)\n?") do

            --alert(line)

            exec("/hide /wait", line)

end
end



function setvartocmd()
os.setenv('addmode', addmode)
os.setenv('ramdiskid', ramdiskID)
os.setenv('guid', ID)
```

```
local boot_file_name = string.match(boot_file_path, "([^\\]+)¥")
os.setenv(′bootapp_name′, boot_file_name)
os.setenv(′boot_file_path′, boot_file_path)
os.setenv(′boot_file_partition′, boot_file_partition)
os.setenv(′boot_sdi_partition′, boot_sdi_partition)
os.setenv(′bootapp_file′, bootfile)
end
function createBCD(boot_platform)
   if boot_platform == "BIOS" then



            exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\Boot]])


            exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] .. bcd_drv .. [[\]])
                exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv .. [[\]])


                exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\Boot\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
                end
 if boot_platform == "UEFI" then
        local bcd_drv = "B:"
            exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Microsoft\\Boot]])
                exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Boot]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv .. [[\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv .. [[\\EFI\\Boot\\bootx64.efi]])


                exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
                end
end

function cs_patition()


    if boot_platform == "BIOS" then
        bcd_drv = Dialog:BrowseFolder(
                  "当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区",
                  17)
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] ..
                 bcd_drv .. [[\Boot\BCD]])
    elseif boot_platform == "UEFI" and File.exists([[B:\EFI\Microsoft\Boot\BCD]]) then

           exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    elseif boot_platform == "UEFI" and
        not File.exists([[B:\EFI\Microsoft\Boot\BCD]]) or not File.exists([[B:\EFI\Boot\Bootx64.efi]]) then
        createBCD(boot_platform)
        exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    end
end


function check_offline_os()
    if exec("/wait", [[bcdedit]]) == 1 then

        cs_patition()


        boot_bcd_var()
    else
        boot_bcd_var()
    end
```

```
end

function boot_bcd_var()

    if boot_platform == "BIOS" then

        local exitcode, boot_drv_stdout = winapi.execute(
                                               "cmd /c bcdedit /enum {bootmgr}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        bcd_store = boot_drv .. [[\Boot\BCD]]
        bcd_drv = boot_drv
    elseif boot_platform == "UEFI" then
        exec("/show /wait", "cmd /c mountvol B: /s")
                local bcd_drv = "B:"
        local exitcode, boot_drv_stdout = winapi.execute(
                                               "cmd /c bcdedit /enum {default}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        local exitcode, bcd_drv_stdout = winapi.execute(
                                              "cmd /c bcdedit /enum {bootmgr}")
        bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
        bcd_drv = bcd_drv_stdout:match(
                      "device                  partition=([^\n]+)\n")
        bcd_store = bcd_drv .. [[\EFI\Microsoft\Boot\BCD]]
    end
end




function ntboot_init()
    if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:" then

        exec("/hide /wait", [[cmd /c mkdir "X:\program files\ms" ]])
        exec("/hide",
             [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])

        get_winver()

ck_BIOS()

        ntboot_addmode = "pe"

        check_offline_os()
        ntboot_runmode = "WinPE环境模式!(离线系统[启动]盘符->" .. bcd_drv ..
                             "] [启动环境->" .. BIOS_str .. "]"
        APP_Path = os.getenv("WINXSHELL_MODULEPATH")
    else
```

```
        get_winver()

                ck_BIOS()



        boot_drv = os.getenv("systemdrive")
        bcd_drv = os.getenv("systemdrive")
        ntboot_addmode = "os"
        ntboot_runmode = "正常系统环境!(系统[启动]盘符)->" .. boot_drv ..
                             "] [启动环境->" .. BIOS_str .. "]"
    end
end

function getBootfileExt(bootfile)
    if bootfile ‾= nil then
        bootfile_ext = bootfile:match(".+%.(%w+)¥")
    bootfile_ext = string.lower(bootfile_ext) -- 转小写
        end

    -- 转大写 bootfile_ext = string.upper(bootfile_ext)
    if bootfile_ext == "wim" then
        addfile = "wim"
        bootok = "1"

    elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
```

```
        addfile = "vhd"
        bootok = "1"
        elseif bootfile_ext == "efi" and boot_platform == "BIOS" then
        alert("当前环境不支持添加'EFI应用程序',请在'UEFI环境'执行此操作")
                 sui:close()
        return
    elseif bootfile_ext == "efi" and boot_platform == "UEFI" then
        addfile = "efi"
        bootok = "1"
    end
end

function ntboot_args(bootfile)
   ntboot_init()
    getBootfileExt(bootfile)
    if bootok ~= "1" then
        winapi.show_message("错误", "暂未支持启动这种格式")
        return
    end
    if bootnow == "1" then

        bootsdi(bootfile)
        ntbootnow(bootfile)
    else
        -- vhd
        bootsdi(bootfile)
        ntbootadd(bootfile)
    end
end

-- 分离boot.sdi路径
function bootsdi(bootfile)
    -- wim或vhd文件 所在分区相对路径
    boot_file_path = bootfile:match(":([^] %s]+)")
    -- wim或vhd文件 所在分区
    boot_file_partition = bootfile:match("([^] %s]+):")
    sdi = [[]] .. bcd_drv .. [[\pe\boot.sdi]]
    -- boot.sdi 所在分区相对路径
    -- alert(sdi)
    boot_sdi_path = sdi:match(":([^] %s]+)")
    -- boot.sdi 所在分区
    boot_sdi_partition = sdi:match("([^] %s]+):")
end

function win7_replacebootmgr()
    exec("/show /wait", "cmd /c mountvol B: /s")
    if addfile == "vhd" then
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                  [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
    end
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
    exec("/hide /wait", "cmd /c mountvol B: /d")
end

function Add_Boot(addfile)
    if winver == "win7" then win7_replacebootmgr() end
    if addfile == "wim" then
        math.randomseed(os.time())
        local random_number = string.format("%06d", math.random(0, 999999))

        ID = [["{19260817-6666-8888-f00d-caffee]] .. random_number .. [[}"]]
                ramdiskID = "{19260817-6666-8888-f00d-caffee000010}"
local cmdlines = [[
bcdedit.exe /create %ramdiskid% /device
bcdedit.exe /set %ramdiskid%  ramdisksdidevice partition=%boot_sdi_partition%:
bcdedit.exe /set %ramdiskid%  ramdisksdipath \pe\boot.sdi
bcdedit.exe /create %ramdiskid% /device
bcdedit.exe /create %guid% /d %bootapp_name% /application osloader
bcdedit.exe /set %guid% Description %bootapp_name%
bcdedit.exe /set %guid% Device ramdisk=[%boot_file_partition%:]%boot_file_path%,%ramdiskid%
bcdedit.exe /set %guid% OSDevice ramdisk=[%boot_file_partition%:]%boot_file_path%,%ramdiskid%
bcdedit.exe /set %guid% systemroot \windows
bcdedit.exe /set %guid% detecthal Yes
bcdedit.exe /set %guid% winpe Yes
bcdedit.exe /set %guid% highestmode Yes
bcdedit.exe /set %guid% bootmenupolicy legacy
bcdedit %addmode% %guid% /addlast
]]
setvartocmd()
cmdlines = cmdlines:envstr()
execlines(cmdlines)
    elseif addfile == "vhd" then
        math.randomseed(os.time())
        local random_number = string.format("%06d", math.random(0, 999999))
    ID = [["{19260817-6666-8888-f00d-caffee]] .. random_number .. [[}"]]
local cmdlines = [[
bcdedit /create {bootmgr} /d "Windows Boot Manager"
bcdedit /set {bootmgr} Locale zh-CN
bcdedit /set {bootmgr} nointegritychecks yes
bcdedit /set {bootmgr} testsigning yes
bcdedit /set {bootmgr} timeout 8
bcdedit -create  %guid% /d %bootapp_name% /application osloader
bcdedit /set device vhd=[%boot_file_partition%:]%boot_file_path%
bcdedit /set osdevice vhd=[%boot_file_partition%:]%boot_file_path%
bcdedit.exe /set %guid% systemroot \windows
bcdedit.exe /set %guid% Locale zh-CN
bcdedit.exe /set %guid% detecthal Yes
bcdedit.exe /set %guid% highestmode Yes
bcdedit %addmode% %guid% /addlast
]]
setvartocmd()
cmdlines = cmdlines:envstr()
execlines(cmdlines)


          elseif addfile == "efi" then
                local cmdlines = [[
cmd /c copy "%bootapp_file%" B:\EFI\Boot\
bcdedit /set %guid% path \EFI\Boot\%bootapp_name%
bcdedit /deletevalue %guid% locale
bcdedit /deletevalue %guid% inherit
bcdedit /deletevalue %guid% default
bcdedit /deletevalue %guid% displayorder
bcdedit /deletevalue %guid% timeout
bcdedit /deletevalue %guid% nointegritychecks
bcdedit /deletevalue %guid% testsigning
bcdedit %addmode% %guid% /addlast
]]
```

```
setvartocmd()
cmdlines = cmdlines:envstr()
execlines(cmdlines)




local exitcode, stdout = winapi.execute([["bcdedit /copy {bootmgr} /d "%bootapp_name%"]])
ID = string.match(stdout, "{[0-9a-fA-F%-]+}")
os.setenv(′guid′, ID)
cmdlines = cmdlines:envstr()
execlines(cmdlines)

    end
end

function ntbootnow(bootfile)
    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
                 [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/bootsequence"
```

```
say = "添加一次性启动完成，系统将立即重启!"
Add_Boot(addfile)

if ntboot_addmode == "pe" then

    exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
    local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum all|find "]] ..
                                               ID .. [["]])
    bcdedit_r = bcdedit_enum:match("{.+}")
    dojob = "wpeutil reboot"
else
    local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum all|find "]] ..
                                               ID .. [["]])
    bcdedit_r = bcdedit_enum:match("{.+}")
    exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
    dojob = "shutdown -r -t 10"
end

if bcdedit_r == nil then

    winapi.show_message("注意",
                        "添加失败，可能没有权限(需要右键->管理员运行)")
 --如果是appstore运行就不关闭    sui:close()
```

```
    else
        exec("/hide", [[cmd /c ]] .. dojob)
        r = winapi.show_message("系统即将重启",
                                "系统即将重启并进入" .. bootfile ..
                                    "\n是否取消？【如果是PE环境取消无效】",
                                "yes-no", "warning")
    end
    if r == "yes" then
        exec("/hide", [[cmd /c shutdown -a]])
        winapi.show_message("注意",
                            "您虽然取消了立即启动到镜像，\n但重启后还是会启动一次的，\n再重启就好了，莫慌!")
         --如果是appstore运行就不关闭  sui:close()
    end
end


function ntboot_ask(bootfile)
    if BIOS_str == nil then ck_BIOS() end
    if bcd_drv == nil then
        -- boot_drv = os.getenv("systemdrive")
        ntboot_init()
    end
    if winver == nil then get_winver() end
    bootsdi(bootfile)
    local r = winapi.show_message("操作镜像",
                                  "［是］ 立即重启到镜像并只启动一次\n\n ［否］ 添加到菜单以后启动\n\n ［取消］ 不操作\n\n 当前启动文件：" ..
```

```
                                     bootfile .. "\n\n 启动环境：" ..
                                     BIOS_str .. "\n 当前系统为：" ..
                                     winver .. "\n\n boot.sdi所在分区：" ..
                                     boot_sdi_partition ..
                                     "\n boot.sdi相对路径：" ..
                                     boot_sdi_path ..
                                     "\n\n 启动镜像盘符：" ..
                                     boot_file_partition .. "\n 启动镜像：" ..
                                     boot_file_path, "yes-no-cancel", "warning")
    if r == "yes" then
        ntbootnow(bootfile)
    elseif r == "no" then
        ntbootadd(bootfile)
    elseif r == "cancel" then
            App_bt_text.text = [[   注意:<c #ff0000>添加失败，你取消了操作!</c>]]

        return

    -- winapi.show_message("您取消了操作镜像", bootfile)
    end
end

function ntbootadd(bootfile)

    -- 创建pe目录
```

```
if addfile == "wim" then
    exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
             [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
     exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
             [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
end
addmode = "/displayorder"
say = "添加永久启动项完成!"
Add_Boot(addfile)
if ntboot_addmode == "pe" then

    -- alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID .. [["]])

    local exitcode, bcdedit_enum = winapi.execute(
                                      [[bcdedit /enum all |find "]] .. ID ..
                                          [["]])
    bcdedit_r = bcdedit_enum:match("{.+}")
    exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)

else
    local exitcode, bcdedit_enum = winapi.execute(
                                      [[bcdedit /enum all |find "]] .. ID ..
                                          [["]])
    bcdedit_r = bcdedit_enum:match("{.+}")
    end
    -- alert([[bcdedit /enum |find "]] .. ID .. [["]])
    if bcdedit_r == nil then
                   App_bt_text.text = "   注意:<c #ff0000>添加失败，可能没有权限(需要右键->管理员运行)</c>"
     -- winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
    else

    App_bt_text.text = "  成功添加<c #ff0000>" .. bootfile .. "</c>"
        if appmode == "wimboot" then
        winapi.show_message("选择了添加到启动菜单", say)
                end

        --如果是appstore运行就不关闭   sui:close()
    end
end

function Add_Boot_Option(addfile)
    ntboot_init()
    -- if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
    if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
        toadmin = winapi.show_message("没有以管理员身份运行!",
                                      "请右键管理员身份运行本程序!，\n是否现在就尝试管理员身份重新运行本程序？",
                                      "yes-no", "warning")
    end
    if toadmin == "yes" then
        -- exec('/hide', [[cmd /c shutdown -a]])
        -- winapi.show_message('注意', "执行管理员提升权限操作")
        exec("/admin", [["]] .. APP_Path ..
                 [[\WinXshell.exe" -ui -jcfg wxsUI\UI_AppStore\main.jcfg]])
        sui:close()
        return
    elseif toadmin == "no" then
        return
    end
    -- winapi.show_message('添加文件类型', addfile)
    local filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
    if addfile == "wim" then filter = "启动镜像|*wim|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                                 "文件 当前为:" .. ntboot_runmode, filter)
    elseif addfile == "vhd" then filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                                 "文件 当前为:" .. ntboot_runmode, filter)
    elseif addfile == "raw" then
        local filter = "启动镜像|*wim;*vhd;*vhdx;*efi|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的文件 当前为:" .. ntboot_runmode, filter)
        getBootfileExt(bootfile)
        bootsdi(bootfile)
        end
    if bootfile == "" then
        -- winapi.show_message("你犯了个错误!", "你没有选择镜像!")
```

```
        return
    else

        ntboot_ask(bootfile)
    end
end

function helper_ntboot_onload()

    -- addwim()
    if has_option("-ntboot") then
        bootfile = get_option("-ntboot")
        ntboot_args(bootfile)
    end
    if has_option("-now") then
        bootnow = "1"
    else
        bootnow = "0"
    end
end


function check_boot()
    if has_option("-now") or has_option("-r")  then
                bootnow = "1"
        else
                bootnow = "0"
        end
end
        --addwim()


        if has_option("-ntboot") then
        bootfile = get_option("-ntboot")
                check_boot()
        ntboot_args(bootfile)
   elseif has_option("-f")  then
        bootfile = get_option("-f")
        check_boot()
        ntboot_args(bootfile)
        else

        return

        end--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_ntboot.lua"
```

```
local ntbootfile_ext = ""
local bootfile = ""
local getfullpath_func = type(File.GetFullPath)
if getfullpath_func == "function" then
temp = File.GetFullPath(os.getenv("temp"))
else
temp = os.getenv("temp")
end
systemdrive = os.getenv("systemdrive")

-- 提取固定值
local exitcode, stdout = winapi.execute("cmd /c mountvol")
stdout = stdout:gsub("\r\n", "\n")
bios_str = stdout:match("EFI ([^\n]+)\n")

-- 获取系统版本
function get_winver()
    local exitcode, stdout = winapi.execute([[ver]])
    local winver_str = stdout:match("10.0.")
    -- stdout = stdout:gsub("\r\n", "\n")
    -- winapi.show_message(′错误′, winver_str)
    if winver_str ‾= nil then
        winver = "Win10/Win11"
    else
        winver = "win7"
```

```
  end
end

function createBCD(boot_platform)
  if boot_platform == "bios" then

      exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\Boot]])

      exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] .. bcd_drv .. [[\]])
        exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv .. [[\]])

        exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\Boot\BCD.tmp]])
        exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\\Boot\\BCD.tmp]])
        exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
        end
 if boot_platform == "uefi" then
    local bcd_drv = "B:"
      exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Microsoft\\Boot]])
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Boot]])
      exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
      exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv .. [[\\EFI\\Boot\\bootx64.efi]])



                exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows
Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
                end
end

function cs_patition()



    if boot_platform == "bios" then
        bcd_drv = Dialog:BrowseFolder(
                 "当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区",
                 17)
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] ..
                 bcd_drv .. [[\Boot\BCD]])
    elseif boot_platform == "uefi" and File.exists([[B:\EFI\Microsoft\Boot\BCD]]) then

           exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    elseif boot_platform == "uefi" and
        not File.exists([[B:\EFI\Microsoft\Boot\BCD]]) or not File.exists([[B:\EFI\Boot\Bootx64.efi]]) then
        createBCD(boot_platform)
        exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    end
end



function check_offline_os()
    if exec("/wait", [[bcdedit]]) == 1 then
        cs_patition()
        boot_bcd_var()
    else
        boot_bcd_var()
    end
end

function boot_bcd_var()
    if boot_platform == "bios" then
        local exitcode, boot_drv_stdout = winapi.execute(
                                              "cmd /c bcdedit /enum {bootmgr}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        bcd_store = boot_drv .. [[\Boot\BCD]]
        bcd_drv = boot_drv
    elseif boot_platform == "uefi" then
        exec("/show /wait", "cmd /c mountvol B: /s")
                local bcd_drv = "B:"
        local exitcode, boot_drv_stdout = winapi.execute(
                                              "cmd /c bcdedit /enum {default}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        local exitcode, bcd_drv_stdout = winapi.execute(
                                             "cmd /c bcdedit /enum {bootmgr}")
        bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
        bcd_drv = bcd_drv_stdout:match(
                      "device                  partition=([^\n]+)\n")
        bcd_store = bcd_drv .. [[\EFI\Microsoft\Boot\BCD]]
    end
end

-- 检测环境
function ck_bios()
    if bios_str == nil then
        bios_str = "传统BIOS"
        boot_platform = "bios"
        wimwinload = [[\Windows\system32\boot\winload.exe]]
        vhdwinload = [[\Windows\system32\winload.exe]]
    else
            bcd_drv = "B:"
        bios_str = "UEFI"
        boot_platform = "uefi"
        wimwinload = [[\Windows\system32\boot\winload.efi]]
        vhdwinload = [[\Windows\system32\winload.efi]]
    end
end
ck_bios()
App_platform = sui:find("boot_platform")
App_platform.text = "当前环境:" .. bios_str

function ntboot_init()
    if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:" then
        exec("/hide /wait", [[cmd /c mkdir "X:\program files\ms" ]])
        exec("/hide",
             [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])
        get_winver()
        ntboot_addmode = "pe"

        check_offline_os()
        ntboot_runmode = "WinPE环境模式!(离线系统[启动]盘符->" .. bcd_drv ..
                             "] [启动环境->" .. bios_str .. "]"
        APP_Path = os.getenv("WINXSHELL_MODULEPATH")
    else
        get_winver()
        boot_drv = os.getenv("systemdrive")
        bcd_drv = os.getenv("systemdrive")
        ntboot_addmode = "os"
        ntboot_runmode = "正常系统环境!(系统[启动]盘符)->" .. boot_drv ..
                             "] [启动环境->" .. bios_str .. "]"
    end
end

function getBootfileExt(bootfile)
    bootfile_ext = bootfile:match(".+%.(%w+)¥")
    bootfile_ext = string.lower(bootfile_ext) -- 转小写

    -- 转大写 bootfile_ext = string.upper(bootfile_ext)
    if bootfile_ext == "wim" then
        addfile = "wim"
        bootok = "1"

    elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
        addfile = "vhd"
        bootok = "1"
    end
end
```

```
function ntboot_args(bootfile)

    getBootfileExt(bootfile)
    if bootok ~= "1" then
        winapi.show_message("错误", "暂未支持启动这种格式")
        return
    end
    if bootnow == "1" then

        bootsdi(bootfile)
        ntbootnow(bootfile)
    else
        -- vhd
        bootsdi(bootfile)
        ntbootadd(bootfile)
    end
end

-- 分离boot.sdi路径
function bootsdi(bootfile)
    -- wim或vhd文件 所在分区相对路径
    boot_file_path = bootfile:match(":([^] %s]+)")
    -- wim或vhd文件 所在分区
    boot_file_partition = bootfile:match("([^] %s]+):")
    sdi = [[]] .. bcd_drv .. [[\pe\boot.sdi]]
    -- boot.sdi 所在分区相对路径
    -- alert(sdi)
    boot_sdi_path = sdi:match(":([^] %s]+)")
    -- boot.sdi 所在分区
    boot_sdi_partition = sdi:match("([^] %s]+):")
end

function win7_replacebootmgr()
    exec("/show /wait", "cmd /c mountvol B: /s")
    if addfile == "vhd" then
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
    end
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
             [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
             [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
             [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
    exec("/hide /wait", "cmd /c mountvol B: /d")
end
```

```
function Add_Boot(addfile)
    if winver == "win7" then win7_replacebootmgr() end
    if addfile == "wim" then
        ID = "{19260817-6666-8888-f00d-caffee000000}"
        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait",
            [[bcdedit /create {ramdiskoptions} /d "ntboot ramdisk"]])
        exec("/hide /wait",
            [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait",
            [[bcdedit /set {ramdiskoptions} ramdisksdidevice  partition=]] ..
                boot_sdi_partition .. [[:]])
        exec("/hide /wait",
            [[bcdedit /set {ramdiskoptions} ramdisksdipath ]] .. boot_sdi_path)
        exec("/hide /wait", [[bcdedit -create ]] .. ID ..
                [[ /d "BootWim With NTBOOT 6.x" /application osloader]])
        exec("/hide /wait",
            [[bcdedit -set ]] .. ID .. " device ramdisk=[" ..
                boot_file_partition .. ":]" .. boot_file_path ..
                [[,{ramdiskoptions}]])
        exec("/hide /wait",
            [[bcdedit -set ]] .. ID .. " osdevice ramdisk=[" ..
                boot_file_partition .. ":]" .. boot_file_path ..
                [[,{ramdiskoptions}]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. wimwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
        exec("/hide /wait",
            [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
    elseif addfile == "vhd" then
        ID = "{19260817-6666-8888-f00d-caffee000001}"
        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait",
            [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
        -- exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
        exec("/hide /wait", [[bcdedit -create ]] .. ID ..
                [[ /d "Boot from VHD With NTBOOT 6.x" /application osloader]])
        exec("/hide /wait",
            [[bcdedit -set ]] .. ID .. " device vhd=[" .. boot_file_partition ..
                ":]" .. boot_file_path)
        exec("/hide /wait",
            [[bcdedit -set ]] .. ID .. " osdevice vhd=[" .. boot_file_partition ..
```

```
                ":]" .. boot_file_path)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
        exec("/hide /wait",
            [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
    end
end

function ntbootnow(bootfile)
    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
               exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
                [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/bootsequence"
    say = "添加一次性启动完成，系统将立即重启!"
    Add_Boot(addfile)
    if ntboot_addmode == "pe" then
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        local exitcode, bcdedit_enum = winapi.execute(
                                            [[bcdedit /enum all|find "]] ..
                                                boot_file_path .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        dojob = "wpeutil reboot"
    else
        local exitcode, bcdedit_enum = winapi.execute(
                                            [[bcdedit /enum all|find "]] ..
                                                boot_file_path .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        dojob = "shutdown -r -t 10"
    end

    if bcdedit_r == nil then
        winapi.show_message("注意",
                            "添加失败，可能没有权限(需要右键->管理员运行)")
        sui:close()
    else
        exec("/hide", [[cmd /c ]] .. dojob)
        r = winapi.show_message("系统即将重启",
                                "系统即将重启并进入" .. bootfile ..
                                    "\n是否取消？【如果是PE环境取消无效】",
                                "yes-no", "warning")
    end
    if r == "yes" then
        exec("/hide", [[cmd /c shutdown -a]])
        winapi.show_message("注意",
                            "您虽然取消了立即启动到镜像，\n但重启后还是会启动一次的，\n再重启就好了，莫慌!")
        sui:close()
    end
end

function ntboot_ask(bootfile)
    if bios_str == nil then ck_bios() end
    if bcd_drv == nil then
        -- boot_drv = os.getenv("systemdrive")
        ntboot_init()
    end
    if winver == nil then get_winver() end
    bootsdi(bootfile)
    local r = winapi.show_message("操作镜像",
                                  "［是］立即重启到镜像并只启动一次\n\n［否］添加到菜单以后启动\n\n［取消］不操作\n\n当前启动文件：" ..
                                      bootfile .. "\n\n启动环境：" ..
                                      bios_str .. "\n当前系统为：" ..
                                      winver .. "\n\nboot.sdi所在分区：" ..
                                      boot_sdi_partition ..
                                      "\nboot.sdi相对路径：" ..
                                      boot_sdi_path ..
```

```
                                        "\n\n 启动镜像盘符: " ..
                                        boot_file_partition .. "\n 启动镜像: " ..
                                        boot_file_path, "yes-no-cancel", "warning")
    if r == "yes" then
        ntbootnow(bootfile)
    elseif r == "no" then
        ntbootadd(bootfile)
    elseif r == "cancel" then
        return
        -- winapi.show_message("您取消了操作镜像", bootfile)
    end
end

function ntbootadd(bootfile)
    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
         exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
                 [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/displayorder"
    say = "添加永久启动项完成!"
    Add_Boot(addfile)
    if ntboot_addmode == "pe" then
        -- alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID .. [["]])
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum |find "]] .. ID ..
                                               [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
    else
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum |find "]] .. ID ..
                                               [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
    end

    -- alert([[bcdedit /enum |find "]] .. ID .. [["]])
    if bcdedit_r == nil then
        winapi.show_message("注意",
                            "添加失败，可能没有权限(需要右键->管理员运行)")
    else
        winapi.show_message("选择了添加到启动菜单", say)
        sui:close()
    end
end

function Add_Boot_Option(addfile)
```

```
ntboot_init()
-- if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
    toadmin = winapi.show_message("没有以管理员身份运行!",
                                  "请右键管理员身份运行本程序!, \n是否现在就尝试管理员身份重新运行本程序？",
                                  "yes-no", "warning")
end
if toadmin == "yes" then
    -- exec('/hide', [[cmd /c shutdown -a]])
    -- winapi.show_message('注意', "执行管理员提升权限操作")
    exec("/admin", [["]] .. APP_Path ..
             [[\WinXshell.exe" -ui -jcfg wxsUI\UI_AppStore\main.jcfg]])
    sui:close()
    return
elseif toadmin == "no" then
    return
end
-- winapi.show_message('添加文件类型', addfile)
local filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
if addfile == "wim" then filter = "启动镜像|*wim|所有文件|*.*"
    bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                              "文件 当前为:" .. ntboot_runmode, filter)
elseif addfile == "vhd" then filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
    bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                              "文件 当前为:" .. ntboot_runmode, filter)
    elseif addfile == "raw" then
        local filter = "启动镜像|*wim;*vhdx;*vhdx|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的文件[支持WIM/VHD/VHDX]当前为:" .. ntboot_runmode, filter)
        getBootfileExt(bootfile)
        bootsdi(bootfile)
        end
    if bootfile == "" then
        -- winapi.show_message("你犯了个错误!", "你没有选择镜像!")
        return
    else

        ntboot_ask(bootfile)
    end
end

function helper_ntboot_onload()
    -- addwim()
    if has_option("-ntboot") then
        bootfile = get_option("-ntboot")
        ntboot_args(bootfile)
    end
    if has_option("-now") then
        bootnow = "1"
    else
        bootnow = "0"
    end
end



function check_boot()
    if has_option("-now") or has_option("-r")  then
                bootnow = "1"
        else
                bootnow = "0"
        end
end
        --addwim()



--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\20230301\helper_ntboot.lua"
```

```
local ntbootfile_ext = ""
local bootfile = ""
local getfullpath_func = type(File.GetFullPath)
if getfullpath_func == "function" then
temp = File.GetFullPath(os.getenv("temp"))
else
temp = os.getenv("temp")
end
systemdrive = os.getenv("systemdrive")
APP_Path = app:info('path')
UI_Path = sui:info('uipath')
-- APP_Path = os.getenv("temp") .. [[\NTBoot]] --ntboot用
-- 提取固定值
local exitcode, stdout = winapi.execute("cmd /c mountvol")
stdout = stdout:gsub("\r\n", "\n")
get_bios_str = stdout:match("EFI ([^\n]+)\n")

-- 获取系统版本
function get_winver()
    local exitcode, stdout = winapi.execute([[ver]])
    local winver_str = stdout:match("10.0.")
    -- stdout = stdout:gsub("\r\n", "\n")
    -- winapi.show_message('错误', winver_str)
    if winver_str ‾= nil then
        winver = "Win10/Win11"
    else
        winver = "win7"
    end
end

-- 检测环境
```

```
function ck_bios()
    if get_bios_str == nil then
        bios_str = "传统BIOS"
        boot_platform = "bios"
        wimwinload = [[\Windows\system32\boot\winload.exe]]
        vhdwinload = [[\Windows\system32\winload.exe]]
    else
            bcd_drv = "B:"
        bios_str = "UEFI"
        boot_platform = "uefi"
        wimwinload = [[\Windows\system32\boot\winload.efi]]
        vhdwinload = [[\Windows\system32\winload.efi]]
    end
end
ck_bios()
App_platform = sui:find("boot_platform") --diskfire用
App_platform.text = "当前环境:" .. bios_str --diskfire用

function createBCD(boot_platform)
   if boot_platform == "bios" then


            exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\Boot]])


            exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] ..
bcd_drv .. [[\]])


                exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv ..
[[\Boot\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\\Boot\\BCD.tmp
/create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv ..
[[\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
bcd_drv)
                end
 if boot_platform == "uefi" then
        local bcd_drv = "B:"
            exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Microsoft\\Boot]])
                exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Boot]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. bcd_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. bcd_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. bcd_drv .. [[\\EFI\\Boot\\bootx64.efi]])



                exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
bcd_drv)
                end
end

function cs_patition()



    if boot_platform == "bios" then
```

```
        bcd_drv = Dialog:BrowseFolder(
                  "当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区",
                  17)
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] ..
                 bcd_drv .. [[\Boot\BCD]])
    elseif boot_platform == "uefi" and File.exists([[B:\EFI\Microsoft\Boot\BCD]]) then

           exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    elseif boot_platform == "uefi" and
        not File.exists([[B:\EFI\Microsoft\Boot\BCD]]) or not
File.exists([[B:\EFI\Boot\Bootx64.efi]]) then
        createBCD(boot_platform)
        exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    end
end


function check_offline_os()
    if exec("/wait", [[bcdedit]]) == 1 then
        cs_patition()
        boot_bcd_var()
    else
        boot_bcd_var()
    end
end

function boot_bcd_var()
    if boot_platform == "bios" then
```

```
        local exitcode, boot_drv_stdout = winapi.execute(
                                               "cmd /c bcdedit /enum {bootmgr}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        bcd_store = boot_drv .. [[\Boot\BCD]]
        bcd_drv = boot_drv
    elseif boot_platform == "uefi" then
        exec("/show /wait", "cmd /c mountvol B: /s")
                local bcd_drv = "B:"
        local exitcode, boot_drv_stdout = winapi.execute(
                                               "cmd /c bcdedit /enum {default}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        local exitcode, bcd_drv_stdout = winapi.execute(
                                              "cmd /c bcdedit /enum {bootmgr}")
        bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
        bcd_drv = bcd_drv_stdout:match(
                      "device                  partition=([^\n]+)\n")
        bcd_store = bcd_drv .. [[\EFI\Microsoft\Boot\BCD]]
    end
end


function ntboot_init()

    if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:" then
        exec("/hide /wait", [[cmd /c mkdir "X:\program files\ms" ]])
        exec("/hide",
             [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])
        get_winver()
        ntboot_addmode = "pe"

        check_offline_os()
        ntboot_runmode = "WinPE环境模式!(离线系统[启动]盘符->" .. bcd_drv ..
                             "] [启动环境->" .. bios_str .. "]"
        APP_Path = os.getenv("WINXSHELL_MODULEPATH")
    else
        get_winver()
        boot_drv = os.getenv("systemdrive")
        bcd_drv = os.getenv("systemdrive")
        ntboot_addmode = "os"
        ntboot_runmode = "正常系统环境!(系统[启动]盘符)->" .. boot_drv ..
                             "] [启动环境->" .. bios_str .. "]"
    end
end

function getBootfileExt(bootfile)
    bootfile_ext = bootfile:match(".+%.(%w+)¥")
    bootfile_ext = string.lower(bootfile_ext) -- 转小写
    -- 转大写 bootfile_ext = string.upper(bootfile_ext)
    if bootfile_ext == "wim" then
        addfile = "wim"
        bootok = "1"
    elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
        addfile = "vhd"
        bootok = "1"
    end
end

function ntboot_args(bootfile)
```

```
    ntboot_init()
    getBootfileExt(bootfile)

    if bootok ~= "1" then
        winapi.show_message("错误", "暂未支持启动这种格式")
        return
    end
    if bootnow == "1" then
        bootsdi(bootfile)
        ntbootnow(bootfile)
    else
        -- vhd
        bootsdi(bootfile)
        ntbootadd(bootfile)
    end
end

-- 分离boot.sdi路径
function bootsdi(bootfile)
    -- wim或vhd文件 所在分区相对路径
    boot_file_path = bootfile:match(":([^] %s]+)")
    -- wim或vhd文件 所在分区
    boot_file_partition = bootfile:match("([^] %s]+):")
    sdi = [[]] .. bcd_drv .. [[\pe\boot.sdi]]
    -- boot.sdi 所在分区相对路径
    -- alert(sdi)
    boot_sdi_path = sdi:match(":([^] %s]+)")
    -- boot.sdi 所在分区
    boot_sdi_partition = sdi:match("([^] %s]+):")
end
```

```
function win7_replacebootmgr()
    exec("/show /wait", "cmd /c mountvol B: /s")
    if addfile == "vhd" then
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
    end
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
    exec("/hide /wait", "cmd /c mountvol B: /d")
end

function Add_Boot(addfile)
    if winver == "win7" then win7_replacebootmgr() end
    if addfile == "wim" then
        ID = "{19260817-6666-8888-f00d-caffee000000}"
        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait",
             [[bcdedit /create {ramdiskoptions} /d "ntboot ramdisk"]])
        exec("/hide /wait",
             [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait",
             [[bcdedit /set {ramdiskoptions} ramdisksdidevice  partition=]] ..
                 boot_sdi_partition .. [[:]])
        exec("/hide /wait",
             [[bcdedit /set {ramdiskoptions} ramdisksdipath ]] .. boot_sdi_path)
        exec("/hide /wait", [[bcdedit -create ]] .. ID ..
                 [[ /d "BootWim With NTBOOT 6.x" /application osloader]])
        exec("/hide /wait",
             [[bcdedit -set ]] .. ID .. " device ramdisk=[" ..
                 boot_file_partition .. ":]" .. boot_file_path ..
                 [[,{ramdiskoptions}]])
        exec("/hide /wait",
             [[bcdedit -set ]] .. ID .. " osdevice ramdisk=[" ..
                 boot_file_partition .. ":]" .. boot_file_path ..
                 [[,{ramdiskoptions}]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. wimwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
        exec("/hide /wait",
             [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
    elseif addfile == "vhd" then
        ID = "{19260817-6666-8888-f00d-caffee000001}"
        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait",
             [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
        -- exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
        exec("/hide /wait", [[bcdedit -create ]] .. ID ..
                 [[ /d "Boot from VHD With NTBOOT 6.x" /application osloader]])
```

```
        exec("/hide /wait",
             [[bcdedit -set ]] .. ID .. " device vhd=[" .. boot_file_partition ..
                 ":]" .. boot_file_path)
        exec("/hide /wait",
             [[bcdedit -set ]] .. ID .. " osdevice vhd=[" .. boot_file_partition ..
                 ":]" .. boot_file_path)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
        exec("/hide /wait",
             [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
    end
end

function ntbootnow(bootfile)
    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/bootsequence"
    say = "添加一次性启动完成，系统将立即重启!"
    Add_Boot(addfile)
    if ntboot_addmode == "pe" then
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum all|find "]] ..
```

```
                                                boot_file_path .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        dojob = "wpeutil reboot"
    else
        local exitcode, bcdedit_enum = winapi.execute(
                                            [[bcdedit /enum all|find "]] ..
                                                boot_file_path .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        dojob = "shutdown -r -t 10"
    end

    if bcdedit_r == nil then
        winapi.show_message("注意",
                            "添加失败，可能没有权限(需要右键->管理员运行)")
        sui:close()
    else
        exec("/hide", [[cmd /c ]] .. dojob)
        r = winapi.show_message("系统即将重启",
                                "系统即将重启并进入" .. bootfile ..
                                    "\n是否取消？【如果是PE环境取消无效】",
                                "yes-no", "warning")
    end
    if r == "yes" then
        exec("/hide", [[cmd /c shutdown -a]])
        winapi.show_message("注意",
                            "您虽然取消了立即启动到镜像，\n但重启后还是会启动一次的，\n再重启
就好了，莫慌!")
        sui:close()
    end
end
```

```
function ntboot_ask(bootfile)
    if bios_str == nil then ck_bios() end
    if bcd_drv == nil then
        -- boot_drv = os.getenv("systemdrive")
        ntboot_init()
    end
    if winver == nil then get_winver() end
        getBootfileExt(bootfile)
    bootsdi(bootfile)
    local r = winapi.show_message("操作镜像",
                                    " [是] 立即重启到镜像并只启动一次\n\n [否] 添加到菜单以后启动\n\n [取消] 不操作\n\n 当前启动文件：" ..
                                        bootfile .. "\n\n 启动环境：" ..
                                        bios_str .. "\n 当前系统为：" ..
                                        winver .. "\n\n boot.sdi所在分区：" ..
                                        boot_sdi_partition ..
                                        "\n boot.sdi相对路径：" ..
                                        boot_sdi_path ..
                                        "\n\n 启动镜像盘符：" ..
                                        boot_file_partition .. "\n 启动镜像：" ..
                                        boot_file_path, "yes-no-cancel", "warning")
    if r == "yes" then
        ntbootnow(bootfile)
    elseif r == "no" then
        ntbootadd(bootfile)
    elseif r == "cancel" then
        return
        -- winapi.show_message("您取消了操作镜像", bootfile)
    end
end
```

```
function ntbootadd(bootfile)
    -- 创建pe目录
        --alert("你的系统目录是" .. bcd_drv)
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/displayorder"
    say = "添加永久启动项完成!"
    Add_Boot(addfile)
    if ntboot_addmode == "pe" then
        -- alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID .. [["]])
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum |find "]] .. ID ..
                                               [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
    else
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum |find "]] .. ID ..
                                               [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
    end

    -- alert([[bcdedit /enum |find "]] .. ID .. [["]])
    if bcdedit_r == nil then
        winapi.show_message("注意",
                            "添加失败，可能没有权限(需要右键->管理员运行)")
    else
        winapi.show_message("选择了添加到启动菜单", say)
        sui:close()
    end
end

function Add_Boot_Option(addfile)

    ntboot_init()

    -- if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
    if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
        toadmin = winapi.show_message("没有以管理员身份运行!",
                                       "请右键管理员身份运行本程序!, \n是否现在就尝试管理员身
份重新运行本程序? ",
                                       "yes-no", "warning")
    end
    if toadmin == "yes" then
        -- exec('/hide', [[cmd /c shutdown -a]])
        -- winapi.show_message('注意', "执行管理员提升权限操作")
        exec("/admin", [["%temp%\NTBoot\WinXshell.exe" -script ntboot.lua]])
        return
    elseif toadmin == "no" then

        return
    end

  --winapi.show_message('添加文件类型', addfile)
        local filter = "启动镜像|*wim;*vhdx;*vhdx|所有文件|*.*"
                bootfile = Dialog:OpenFile(ntboot_runmode, filter )

        if bootfile == "" then
                --winapi.show_message("你犯了个错误!", "你没有选择镜像!")
                return
        else
                ntboot_ask(bootfile)
        end
end



function helper_ntboot_onload()
     --注释的是ntboot或bootfire用的
        if has_option("-ntboot") then
        alert("ntboot")
        bootfile = get_option("-ntboot")
                check_boot()
        ntboot_args(bootfile)
   --elseif has_option("-f")  then
   --alert("-f")
        --bootfile = get_option("-f")
        --check_boot()
        --ntboot_args(bootfile)
        else

        --Add_Boot_Option(addfile)

        end

  end



function check_boot()
    if has_option("-now") or has_option("-r")  then
                bootnow = "1"
        else
```

```
            bootnow = "0"
      end
end
      --addwim()
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot.lua"
```

```
local ntbootfile_ext = ""
local bootfile = ""
SystemDrive = os.getenv("SystemDrive")
temp = os.getenv("temp")
-- 提取固定值
local exitcode, stdout = winapi.execute("cmd /c mountvol")
stdout = stdout:gsub("\r\n", "\n")
local bios_str = stdout:match("EFI ([^\n]+)\n")
--检测环境
if bios_str == nil then
        bios_str = "传统BIOS"
        wimwinload = [[\Windows\system32\boot\winload.exe]]
        vhdwinload = [[\Windows\system32\winload.exe]]
else
        bios_str = "UEFI"
        wimwinload = [[\Windows\system32\boot\winload.efi]]
        vhdwinload = [[\Windows\system32\winload.efi]]
end

--获取系统版本
function winver()
        local exitcode, stdout = winapi.execute([[ver]])
        local winver_str = stdout:match("10.0.")
        --stdout = stdout:gsub("\r\n", "\n")
        --winapi.show_message('错误', winver_str)
        if winver_str ~= nil then
                winver = "Win10/Win11"
        else
                winver = "win7"
        end
end
```

```
winver()


function getBootfileExt(bootfile)
        bootfile_ext = bootfile:match(".+%.(%w+)¥")
        if bootfile_ext == "wim" then
                addfile = "wim"
                bootok = "1"
        elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
                addfile = "vhd"
                bootok = "1"
        end
end


function ntboot_args(bootfile)
        getBootfileExt(bootfile)
        if bootok ￣= "1" then
                winapi.show_message("错误", "暂未支持启动这种格式")
                return
        end
        if bootnow == "1" then
                bootsdi(bootfile)
                ntbootnow(bootfile)
        else
                bootsdi(bootfile)
                ntbootadd(bootfile)
                --vhd
        end
end


--分离boot.sdi路径
function bootsdi(bootfile)
```

```
        --wim或vhd文件 所在分区相对路径
        boot_file_path = bootfile:match(":([^] %s]+)")
        --wim或vhd文件 所在分区
        boot_file_partition = bootfile:match("([^] %s]+):")
        sdi = [[]] .. SystemDrive .. [[\pe\boot.sdi]]
        --boot.sdi 所在分区相对路径
        boot_sdi_path = sdi:match(":([^] %s]+)")
        --boot.sdi 所在分区
        boot_sdi_partition = sdi:match("([^] %s]+):")
end

function win7_replacebootmgr()
        exec("/show /wait", "cmd /c mountvol B: /s")
        if addfile == "vhd" then
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
        end
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
        exec("/hide /wait", "cmd /c mountvol B: /d")
end

function Add_Boot(addfile)
        if winver == "win7" then
                win7_replacebootmgr()
        end
        if addfile == "wim" then
                ID = "{19260817-6666-8888-f00d-caffee000000}"
                exec("/hide /wait", [[bcdedit /delete ]] .. ID)
                exec("/hide /wait", [[bcdedit /create {ramdiskoptions} /d "ntboot ramdisk"]])
                exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdidevice
partition=]] .. boot_sdi_partition .. [[:]])
                exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdipath ]] ..
boot_sdi_path)
                exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "BootWim With NTBOOT
6.x" /application osloader]])
                exec(
                        "/hide /wait",
                        [[bcdedit -set ]] ..
                        ID .. " device ramdisk=[" .. boot_file_partition .. ":]" ..
boot_file_path .. [[,{ramdiskoptions}]]
                )
                exec(
                        "/hide /wait",
                        [[bcdedit -set ]] ..
                        ID .. " osdevice ramdisk=[" .. boot_file_partition .. ":]" ..
boot_file_path .. [[,{ramdiskoptions}]]
                )
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. wimwinload)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ winpe yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[
/addlast]])
        elseif addfile == "vhd" then
                ID = "{19260817-6666-8888-f00d-caffee000001}"
                exec("/hide /wait", [[bcdedit /delete ]] .. ID)
                exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
                --exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
                exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "Boot from VHD With
NTBOOT 6.x" /application osloader]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " device vhd=[" ..
boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " osdevice vhd=[" ..
boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[
/addlast]])
        end
end

function ntbootnow(bootfile)
        --创建pe目录
        if addfile == "wim" then
                winapi.execute([[mkdir %systemdrive%\pe"]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi"
%systemdrive%\pe]])
        end
        addmode = "/bootsequence"
        dojob = "shutdown -r -t 10"
        say = "添加一次性启动完成，系统将立即重启!"
        Add_Boot(addfile)
        local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID ..
[["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        if bcdedit_r == nil then
                winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
        else
                exec("/hide", [[cmd /c ]] .. dojob)
                r = winapi.show_message("系统即将重启", "系统即将重启并进入" .. bootfile ..
"\n是否取消？",
                        "yes-no", "warning")
        end
        if r == "yes" then
                exec("/hide", [[cmd /c shutdown -a]])
                winapi.show_message("注意", "您虽然取消了立即启动到镜像，\n但重启后还是会启动
一次的，\n再重启就好了，莫慌!")
        end
end


function ntboot_ask(bootfile)
        bootsdi(bootfile)
        local r =
        winapi.show_message(
                "操作镜像",
                "［是］立即重启到镜像并只启动一次\n\n［否］添加到菜单以后启动\n\n［取消］不操
作\n\n 当前启动文件："
```

```
        ..
        bootfile ..
        "\n\n 启动环境：" ..
        bios_str ..
        "\n 当前系统为：" ..
        winver ..
        "\n\n boot.sdi所在分区：" ..
        boot_sdi_partition ..
        "\n boot.sdi相对路径：" ..
        boot_sdi_path .. "\n\n 启动镜像盘符：" .. boot_file_partition .. "\n 启动镜像：" .. boot_file_path,
        "yes-no-cancel",
        "warning"
    )
    if r == "yes" then
        ntbootnow(bootfile)
    elseif r == "no" then
        ntbootadd(bootfile)
    elseif r == "cancel" then
        winapi.show_message("您取消了操作镜像", bootfile)
    end
end

function ntbootadd(bootfile)
    --创建pe目录
    if addfile == "wim" then
        winapi.execute([[mkdir %systemdrive%\pe"]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi" %systemdrive%\pe]])
    end
    addmode = "/displayorder"
    say = "添加永久启动项完成!"
    Add_Boot(addfile)
    local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID .. [["]])
    bcdedit_r = bcdedit_enum:match("{.+}")
    if bcdedit_r == nil then
        winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
    else
        winapi.show_message("选择了添加到启动菜单", say)
    end
end

function Add_Boot_Option(addfile)
    if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
        toadmin = winapi.show_message("没有以管理员身份运行!", "请右键管理员身份运行本程序!，\n是否现在就尝试管理员身份重新运行本程序？"
            , "yes-no", "warning")
    end
    if toadmin == "yes" then
        --exec('/hide', [[cmd /c shutdown -a]])
        --winapi.show_message('注意', "执行管理员提升权限操作")
        exec("/admin", [["]] .. APP_Path .. [[\WinXshell.exe" -ui -jcfg wxsUI\UI_AppStore\main.jcfg]])
        sui:close()
        return
    elseif toadmin == "no" then
        return
    end
    --winapi.show_message('添加文件类型', addfile)
    local filter = "启动镜像|*vhdx;*.vhdx|所有文件|*.*"
    if addfile == "wim" then
```

```
                filter = "启动镜像|*wim|所有文件|*.*"
        end
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile .. "文件 ", filter)

        if bootfile == "" then
                winapi.show_message("你犯了个错误!", "你没有选择镜像!")
        else
                ntboot_ask(bootfile)

        end
end

function helper_ntboot_onload()
        --addwim()
        if has_option("-ntboot") then
                bootfile = get_option("-ntboot")
                ntboot_args(bootfile)
        end
        if has_option("-now") then
                bootnow = "1"
        else
                bootnow = "0"
        end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_ntboot_.lua"
```

```
local ntbootfile_ext = ""
local bootfile = ""
local getfullpath_func = type(File.GetFullPath)
if getfullpath_func == "function" then
temp = File.GetFullPath(os.getenv("temp"))
else
temp = os.getenv("temp")
end
systemdrive = os.getenv("systemdrive")

-- 提取固定值
local exitcode, stdout = winapi.execute("cmd /c mountvol")
stdout = stdout:gsub("\r\n", "\n")
get_bios_str = stdout:match("EFI ([^\n]+)\n")

-- 获取系统版本
function get_winver()
    local exitcode, stdout = winapi.execute([[ver]])
    local winver_str = stdout:match("10.0.")
    -- stdout = stdout:gsub("\r\n", "\n")
    -- winapi.show_message('错误', winver_str)
    if winver_str ~= nil then
        winver = "Win10/Win11"
    else
```

```
        winver = "win7"
    end
end

-- 检测环境
function ck_bios()
    if get_bios_str == nil then
        bios_str = "传统BIOS"
        boot_platform = "bios"
        wimwinload = [[\Windows\system32\boot\winload.exe]]
        vhdwinload = [[\Windows\system32\winload.exe]]
    else
            bcd_drv = "B:"
        bios_str = "UEFI"
        boot_platform = "uefi"
        wimwinload = [[\Windows\system32\boot\winload.efi]]
        vhdwinload = [[\Windows\system32\winload.efi]]
    end
end

ck_bios()

App_platform = sui:find("boot_platform")
App_platform.text = "当前环境:" .. bios_str
```

```
function createBCD(boot_platform)
   if boot_platform == "bios" then

            exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\Boot]])

            exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] .. bcd_drv .. [[\]])
                exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv .. [[\]])

                exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\Boot\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
                end
 if boot_platform == "uefi" then
        local bcd_drv = "B:"
            exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Microsoft\\Boot]])
                exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Boot]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
```

```
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv .. [[\\EFI\\Boot\\bootx64.efi]])


                exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows
Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
                end
end

function cs_patition()


    if boot_platform == "bios" then
        bcd_drv = Dialog:BrowseFolder(
                  "当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区",
                  17)
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] ..
                 bcd_drv .. [[\Boot\BCD]])
    elseif boot_platform == "uefi" and File.exists([[B:\EFI\Microsoft\Boot\BCD]]) then

           exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    elseif boot_platform == "uefi" and
        not File.exists([[B:\EFI\Microsoft\Boot\BCD]]) or not File.exists([[B:\EFI\Boot\Bootx64.efi]]) then
        createBCD(boot_platform)
        exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    end
end


function check_offline_os()
    if exec("/wait", [[bcdedit]]) == 1 then
        cs_patition()
        boot_bcd_var()
    else
        boot_bcd_var()
    end
end

function boot_bcd_var()
    if boot_platform == "bios" then
        local exitcode, boot_drv_stdout = winapi.execute(
                                               "cmd /c bcdedit /enum {bootmgr}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        bcd_store = boot_drv .. [[\Boot\BCD]]
        bcd_drv = boot_drv
    elseif boot_platform == "uefi" then
        exec("/show /wait", "cmd /c mountvol B: /s")
                local bcd_drv = "B:"
        local exitcode, boot_drv_stdout = winapi.execute(
                                              "cmd /c bcdedit /enum {default}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        local exitcode, bcd_drv_stdout = winapi.execute(
                                             "cmd /c bcdedit /enum {bootmgr}")
        bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
        bcd_drv = bcd_drv_stdout:match(
                      "device                  partition=([^\n]+)\n")
        bcd_store = bcd_drv .. [[\EFI\Microsoft\Boot\BCD]]
    end
end



function ntboot_init()
    if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:" then

        exec("/hide /wait", [[cmd /c mkdir "X:\program files\ms" ]])
        exec("/hide",
             [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])

        get_winver()

ck_bios()

        ntboot_addmode = "pe"

        check_offline_os()
        ntboot_runmode = "WinPE环境模式!(离线系统[启动]盘符->" .. bcd_drv ..
                              "] [启动环境->" .. bios_str .. "]"
        APP_Path = os.getenv("WINXSHELL_MODULEPATH")
    else
        get_winver()
                ck_bios()
        boot_drv = os.getenv("systemdrive")
        bcd_drv = os.getenv("systemdrive")
        ntboot_addmode = "os"
        ntboot_runmode = "正常系统环境!(系统[启动]盘符)->" .. boot_drv ..
                              "] [启动环境->" .. bios_str .. "]"
    end
end

function getBootfileExt(bootfile)
```

```
    bootfile_ext = bootfile:match(".+%.(%w+)¥")
    bootfile_ext = string.lower(bootfile_ext) -- 转小写

    -- 转大写 bootfile_ext = string.upper(bootfile_ext)
    if bootfile_ext == "wim" then
        addfile = "wim"
        bootok = "1"

    elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
        addfile = "vhd"
        bootok = "1"
    end
end

function ntboot_args(bootfile)

    getBootfileExt(bootfile)
    if bootok ‾= "1" then
        winapi.show_message("错误", "暂未支持启动这种格式")
        return
    end
    if bootnow == "1" then

        bootsdi(bootfile)
        ntbootnow(bootfile)
    else
        -- vhd
        bootsdi(bootfile)
        ntbootadd(bootfile)
    end
end

-- 分离boot.sdi路径
function bootsdi(bootfile)
    -- wim或vhd文件 所在分区相对路径
    boot_file_path = bootfile:match(":([^ %s]+)")
    -- wim或vhd文件 所在分区
    boot_file_partition = bootfile:match("([^ %s]+):")
    sdi = [[]] .. bcd_drv .. [[\pe\boot.sdi]]
    -- boot.sdi 所在分区相对路径
    -- alert(sdi)
    boot_sdi_path = sdi:match(":([^ %s]+)")
    -- boot.sdi 所在分区
    boot_sdi_partition = sdi:match("([^ %s]+):")
end

function win7_replacebootmgr()
    exec("/show /wait", "cmd /c mountvol B: /s")
    if addfile == "vhd" then
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
    end
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
    exec("/hide /wait", "cmd /c mountvol B: /d")
end

function Add_Boot(addfile)
    if winver == "win7" then win7_replacebootmgr() end
    if addfile == "wim" then
        ID = "{19260817-6666-8888-f00d-caffee000000}"
        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait",
            [[bcdedit /create {ramdiskoptions} /d "ntboot ramdisk"]])
        exec("/hide /wait",
            [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait",
            [[bcdedit /set {ramdiskoptions} ramdisksdidevice  partition=]] ..
                boot_sdi_partition .. [[:]])
        exec("/hide /wait",
            [[bcdedit /set {ramdiskoptions} ramdisksdipath ]] .. boot_sdi_path)
        exec("/hide /wait", [[bcdedit -create ]] .. ID ..
                [[ /d "BootWim With NTBOOT 6.x" /application osloader]])
        exec("/hide /wait",
            [[bcdedit -set ]] .. ID .. " device ramdisk=[" ..
                boot_file_partition .. ":]" .. boot_file_path ..
                [[,{ramdiskoptions}]])
        exec("/hide /wait",
            [[bcdedit -set ]] .. ID .. " osdevice ramdisk=[" ..
                boot_file_partition .. ":]" .. boot_file_path ..
                [[,{ramdiskoptions}]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. wimwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
        exec("/hide /wait",
            [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
    elseif addfile == "vhd" then
        ID = "{19260817-6666-8888-f00d-caffee000001}"
        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait",
             [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
        -- exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
        exec("/hide /wait", [[bcdedit -create ]] .. ID ..
                 [[ /d "Boot from VHD With NTBOOT 6.x" /application osloader]])
        exec("/hide /wait",
             [[bcdedit -set ]] .. ID .. " device vhd=[" .. boot_file_partition ..
                 ":]" .. boot_file_path)
        exec("/hide /wait",
             [[bcdedit -set ]] .. ID .. " osdevice vhd=[" .. boot_file_partition ..
                 ":]" .. boot_file_path)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
        exec("/hide /wait",
             [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
    end
end
```

```
function ntbootnow(bootfile)
    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
                 [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/bootsequence"
    say = "添加一次性启动完成，系统将立即重启!"
    Add_Boot(addfile)
    if ntboot_addmode == "pe" then
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum all|find "]] ..
                                               boot_file_path .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        dojob = "wpeutil reboot"
    else
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum all|find "]] ..
                                               boot_file_path .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        dojob = "shutdown -r -t 10"
    end

    if bcdedit_r == nil then
        winapi.show_message("注意",
                            "添加失败，可能没有权限(需要右键->管理员运行)")
     --如果是appstore运行就不关闭   sui:close()
    else
        exec("/hide", [[cmd /c ]] .. dojob)
        r = winapi.show_message("系统即将重启",
                                "系统即将重启并进入" .. bootfile ..
                                    "\n是否取消？【如果是PE环境取消无效】",
                                "yes-no", "warning")
    end
    if r == "yes" then
        exec("/hide", [[cmd /c shutdown -a]])
        winapi.show_message("注意",
                            "您虽然取消了立即启动到镜像，\n但重启后还是会启动一次的，\n再重启就好了，莫慌!")
         --如果是appstore运行就不关闭  sui:close()
    end
end

function ntboot_ask(bootfile)
    if bios_str == nil then ck_bios() end
    if bcd_drv == nil then
```

```
        -- boot_drv = os.getenv("systemdrive")
        ntboot_init()
    end
    if winver == nil then get_winver() end
    bootsdi(bootfile)
    local r = winapi.show_message("操作镜像",
                                  " [是] 立即重启到镜像并只启动一次\n\n [否] 添加到菜单以后启动\n\n [取消] 不操作\n\n 当前启动文件：" ..
                                      bootfile .. "\n\n 启动环境：" ..
                                      bios_str .. "\n 当前系统为：" ..
                                      winver .. "\n\n boot.sdi所在分区：" ..
                                      boot_sdi_partition ..
                                      "\n boot.sdi相对路径：" ..
                                      boot_sdi_path ..
                                      "\n\n 启动镜像盘符：" ..
                                      boot_file_partition .. "\n 启动镜像：" ..
                                      boot_file_path, "yes-no-cancel", "warning")
    if r == "yes" then
        ntbootnow(bootfile)
    elseif r == "no" then
        ntbootadd(bootfile)
    elseif r == "cancel" then
        return
        -- winapi.show_message("您取消了操作镜像", bootfile)
    end
end
```

```
function ntbootadd(bootfile)
    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
         exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
                 [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/displayorder"
    say = "添加永久启动项完成!"
    Add_Boot(addfile)
    if ntboot_addmode == "pe" then
        -- alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID .. [["]])
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum |find "]] .. ID ..
                                               [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
    else
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum |find "]] .. ID ..
                                               [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
    end

    -- alert([[bcdedit /enum |find "]] .. ID .. [["]])
    if bcdedit_r == nil then
        winapi.show_message("注意",
                            "添加失败，可能没有权限(需要右键->管理员运行)")
    else
        winapi.show_message("选择了添加到启动菜单", say)
        --如果是appstore运行就不关闭    sui:close()
    end
end

function Add_Boot_Option(addfile)
    ntboot_init()
    -- if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
    if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
        toadmin = winapi.show_message("没有以管理员身份运行!",
                                      "请右键管理员身份运行本程序!, \n是否现在就尝试管理员身份重新运行本程序？",
                                      "yes-no", "warning")
    end
    if toadmin == "yes" then
        -- exec('/hide', [[cmd /c shutdown -a]])
        -- winapi.show_message('注意', "执行管理员提升权限操作")
        exec("/admin", [["]] .. APP_Path ..
                 [[\WinXshell.exe" -ui -jcfg wxsUI\UI_AppStore\main.jcfg]])
        sui:close()
        return
    elseif toadmin == "no" then
        return
    end
    -- winapi.show_message('添加文件类型', addfile)
    local filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
    if addfile == "wim" then filter = "启动镜像|*wim|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                                  "文件 当前为:" .. ntboot_runmode, filter)
    elseif addfile == "vhd" then filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                                  "文件 当前为:" .. ntboot_runmode, filter)
    elseif addfile == "raw" then
        local filter = "启动镜像|*wim;*vhdx;*vhdx|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的文件 当前为:" .. ntboot_runmode, filter)
        getBootfileExt(bootfile)
        bootsdi(bootfile)
        end
    if bootfile == "" then
        -- winapi.show_message("你犯了个错误!", "你没有选择镜像!")
        return
    else

        ntboot_ask(bootfile)
    end
end

function helper_ntboot_onload()

    -- addwim()
    if has_option("-ntboot") then
        bootfile = get_option("-ntboot")
        ntboot_args(bootfile)
    end
    if has_option("-now") then
        bootnow = "1"
    else
        bootnow = "0"
    end
end


function check_boot()
    if has_option("-now") or has_option("-r")  then
                bootnow = "1"
        else
                bootnow = "0"
        end
end
```

```
        --addwim()



--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot_20230327.lua"
```

```
local ntbootfile_ext = ""
local bootfile = ""
temp = os.getenv("temp")
systemdrive = os.getenv("systemdrive")
APP_Path = os.getenv("WINXSHELL_MODULEPATH")

-- 提取固定值
local exitcode, stdout = winapi.execute("cmd /c mountvol")
stdout = stdout:gsub("\r\n", "\n")
bios_str = stdout:match("EFI ([^\n]+)\n")

--获取系统版本
function get_winver()
        local exitcode, stdout = winapi.execute([[ver]])
        local winver_str = stdout:match("10.0.")
        --stdout = stdout:gsub("\r\n", "\n")
        --winapi.show_message('错误', winver_str)
        if winver_str ~= nil then
                winver = "Win10/Win11"
        else
                winver = "win7"
        end
end

function createBCD(boot_platform)
       if boot_platform == "bios" then

            exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\\Boot]])

            exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] .. boot_drv .. [[\]])
```

```
                exec("/hide /wait", [[bcdedit /createstore ]] .. boot_drv ..
[[\Boot\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. boot_drv .. [[\\Boot\\BCD.tmp
/create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. boot_drv ..
[[\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
boot_drv)
                end
                if boot_platform == "uefi" then
            exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\\EFI\\Microsoft\\Boot]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Boot\\bootx64.efi]])



                exec("/hide /wait", [[bcdedit /createstore ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
boot_drv)
                end


end
```

```
function cs_patition()
        boot_drv = Dialog:BrowseFolder("当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区", 17)
        if boot_platform == "bios" and File.exists(boot_drv .. [[\Boot\BCD]]) then
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\Boot\BCD]])

        elseif boot_platform == "bios" and not File.exists(boot_drv .. [[\Boot\BCD]]) then
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\Boot\BCD]])

        elseif boot_platform == "uefi" and File.exists(boot_drv .. [[\Boot\BCD]]) then
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\EFI\Microsoft\Boot\BCD]])

        elseif boot_platform == "uefi" and not File.exists(boot_drv .. [[\EFI\Microsoft\Boot\BCD]]) then
        createBCD(boot_platform)
                exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\EFI\Microsoft\Boot\BCD]])
end
end

function check_offline_os()
        if exec("/wait", [[bcdedit]]) == 1 then
                cs_patition()
                boot_bcd_var()
        else
                boot_bcd_var()
        end
```

```
end


function boot_bcd_var()
        if boot_platform == "bios" then
                local exitcode, boot_drv_stdout = winapi.execute("cmd /c bcdedit /enum {bootmgr}")
                boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
                boot_drv = boot_drv_stdout:match("device                  partition=([^\n]+)\n")
                bcd_store = boot_drv .. [[\Boot\BCD]]
        elseif boot_platform == "uefi" then
                local exitcode, boot_drv_stdout = winapi.execute("cmd /c bcdedit /enum {default}")
                boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
                boot_drv = boot_drv_stdout:match("device                  partition=([^\n]+)\n")
                local exitcode, bcd_drv_stdout = winapi.execute("cmd /c bcdedit /enum {bootmgr}")
                bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
                bcd_drv = bcd_drv_stdout:match("device                  partition=([^\n]+)\n")
                bcd_store = boot_drv .. [[\EFI\Microsoft\Boot\BCD]]
        end
end


--检测环境
function ck_bios()
        if bios_str == nil then
                bios_str = "传统BIOS"
                boot_platform = "bios"
                wimwinload = [[\Windows\system32\boot\winload.exe]]
                vhdwinload = [[\Windows\system32\winload.exe]]
        else
                bios_str = "UEFI"
                boot_platform = "uefi"
                wimwinload = [[\Windows\system32\boot\winload.efi]]
                vhdwinload = [[\Windows\system32\winload.efi]]
        end
end

function ntboot_init()
        if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:"  then
         exec("/hide", [[cmd /c mkdir "X:\program files\ms" ]])
          exec("/hide", [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])
            get_winver()
                ntboot_addmode = "pe"
                ck_bios()
                check_offline_os()
                ntboot_runmode = "WinPE环境模式![离线系统盘符->" .. boot_drv .. "] [启动环境->" .. bios_str .. "]"
        else
                get_winver()
                boot_drv = os.getenv("systemdrive")
                ck_bios()
                ntboot_addmode = "os"
                ntboot_runmode = "正常系统环境![系统盘符->" .. boot_drv .. "] [启动环境->" .. bios_str .. "]"
        end
end

function getBootfileExt(bootfile)
        bootfile_ext = bootfile:match(".+%.(%w+)¥")
        bootfile_ext = string.lower(bootfile_ext) --转小写
```

```
    --转大写 bootfile_ext = string.upper(bootfile_ext)
    if bootfile_ext == "wim" then
        addfile = "wim"
        bootok = "1"
    elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
        addfile = "vhd"
        bootok = "1"
    end
end

function ntboot_args(bootfile)
    getBootfileExt(bootfile)
    if bootok ‾= "1" then
        winapi.show_message("错误", "暂未支持启动这种格式")
        return
    end
    if bootnow == "1" then
        bootsdi(bootfile)
        ntbootnow(bootfile)
    else
        --vhd
        bootsdi(bootfile)
        ntbootadd(bootfile)
    end
end

--分离boot.sdi路径
function bootsdi(bootfile)
    --wim或vhd文件 所在分区相对路径
    boot_file_path = bootfile:match(":([^] %s]+)")
    --wim或vhd文件 所在分区
```

```
        boot_file_partition = bootfile:match("([^ %s]+):")
        sdi = [[]] .. boot_drv .. [[\pe\boot.sdi]]
        --boot.sdi 所在分区相对路径
        --alert(sdi)
        boot_sdi_path = sdi:match(":([^ %s]+)")
        --boot.sdi 所在分区
        boot_sdi_partition = sdi:match("([^ %s]+):")
end


function win7_replacebootmgr()
        exec("/show /wait", "cmd /c mountvol B: /s")
        if addfile == "vhd" then
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll"
B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll"
B:\\EFI\\Boot\\bootvhd.dll]])
        end
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi"
B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi"
B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi"
B:\\EFI\\Boot\\bootx64.efi]])
        exec("/hide /wait", "cmd /c mountvol B: /d")
end

function Add_Boot(addfile)
        if winver == "win7" then
        win7_replacebootmgr()
    end
    if addfile == "wim" then
        ID = "{19260817-6666-8888-f00d-caffee000000}"
        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait", [[bcdedit /create {ramdiskoptions} /d "ntboot ramdisk"]])
        exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdidevice partition=]] .. boot_sdi_partition .. [[:]])
        exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdipath ]] .. boot_sdi_path)
        exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "BootWim With NTBOOT 6.x" /application osloader]])
        exec(
            "/hide /wait",
            [[bcdedit -set ]] ..
            ID .. " device ramdisk=[" .. boot_file_partition .. ":]" .. boot_file_path .. [[,{ramdiskoptions}]]
        )
        exec(
            "/hide /wait",
            [[bcdedit -set ]] ..
            ID .. " osdevice ramdisk=[" .. boot_file_partition .. ":]" .. boot_file_path .. [[,{ramdiskoptions}]]
        )
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. wimwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
```

```
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[
/addlast]])
        elseif addfile == "vhd" then
                ID = "{19260817-6666-8888-f00d-caffee000001}"
                exec("/hide /wait", [[bcdedit /delete ]] .. ID)
                exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
                --exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
                exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "Boot from VHD With
NTBOOT 6.x" /application osloader]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " device vhd=[" ..
boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " osdevice vhd=[" ..
boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[
/addlast]])
        end
end


function ntbootnow(bootfile)
        --创建pe目录
        if addfile == "wim" then
                exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi" ]] ..
boot_drv .. [[\pe]])
        end
        addmode = "/bootsequence"
        say = "添加一次性启动完成，系统将立即重启!"
        Add_Boot(addfile)
        if ntboot_addmode == "pe" then
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
                        local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum
all|find "]] .. boot_file_path .. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                dojob = "wpeutil reboot"
        else
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum all|find "]] ..
boot_file_path .. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
                dojob = "shutdown -r -t 10"
        end

        if bcdedit_r == nil then
                winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
                sui:close()
        else
                exec("/hide", [[cmd /c ]] .. dojob)
                r = winapi.show_message("系统即将重启", "系统即将重启并进入" .. bootfile ..
"\n是否取消？【如果是PE环境取消无效】",
                        "yes-no", "warning")
        end
        if r == "yes" then
                exec("/hide", [[cmd /c shutdown -a]])
                winapi.show_message("注意", "您虽然取消了立即启动到镜像，\n但重启后还是会启动一次的，\n再重启就好了，莫慌!")
                sui:close()
        end
end

function ntboot_ask(bootfile)
alert(bios_str)
    ck_bios()
        if boot_drv == nil then
        --boot_drv = os.getenv("systemdrive")
        ntboot_init()

        end
        if winver == nil then
        get_winver()
        end
        bootsdi(bootfile)
        local r =
        winapi.show_message(
                "操作镜像",
                " [是] 立即重启到镜像并只启动一次\n\n [否] 添加到菜单以后启动\n\n [取消] 不操作\n\n 当前启动文件："
                ..
                bootfile ..
                "\n\n 启动环境：" ..
                bios_str ..
                "\n 当前系统为：" ..
                winver ..
                "\n\n boot.sdi所在分区: " ..
                boot_sdi_partition ..
                "\n boot.sdi相对路径: " ..
                boot_sdi_path .. "\n\n 启动镜像盘符: " .. boot_file_partition .. "\n 启动镜像: " .. boot_file_path,
                "yes-no-cancel",
                "warning"
        )
        if r == "yes" then
                ntbootnow(bootfile)
        elseif r == "no" then
                ntbootadd(bootfile)
        elseif r == "cancel" then
                return
                --winapi.show_message("您取消了操作镜像", bootfile)
        end
end

function ntbootadd(bootfile)
        --创建pe目录
        if addfile == "wim" then
                exec("/hide /wait", [[mkdir ]] .. boot_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi" ]] .. boot_drv .. [[\pe]])
        end
        addmode = "/displayorder"
        say = "添加永久启动项完成!"
        Add_Boot(addfile)
        if ntboot_addmode == "pe" then
                --alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID .. [["]])
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID
.. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        else
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID
.. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
        end

        --alert([[bcdedit /enum |find "]] .. ID .. [["]])
        if bcdedit_r == nil then
                winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
        else
                winapi.show_message("选择了添加到启动菜单", say)
                sui:close()
        end
end

function Add_Boot_Option(addfile)
        ntboot_init()
        --if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
        if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
                toadmin = winapi.show_message("没有以管理员身份运行!", "请右键管理员身份运行
本程序!，\n是否现在就尝试管理员身份重新运行本程序？"
                        , "yes-no", "warning")
        end
        if toadmin == "yes" then
                --exec('/hide', [[cmd /c shutdown -a]])
                --winapi.show_message('注意', "执行管理员提升权限操作")
                exec("/admin", [["]] .. APP_Path .. [[\WinXshell.exe" -ui -jcfg
wxsUI\UI_AppStore\main.jcfg]])
                sui:close()
                return
        elseif toadmin == "no" then
                return
        end
        --winapi.show_message('添加文件类型', addfile)
        local filter = "启动镜像|*vhdx;*.vhdx|所有文件|*.*"
        if addfile == "wim" then
                filter = "启动镜像|*wim|所有文件|*.*"
        end
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile .. "文件 当前为:" ..
ntboot_runmode, filter)

        if bootfile == "" then
                --winapi.show_message("你犯了个错误!", "你没有选择镜像!")
                return
        else
                ntboot_ask(bootfile)
        end
end

function helper_ntboot_onload()
        --addwim()
        if has_option("-ntboot") then
                bootfile = get_option("-ntboot")
                ntboot_args(bootfile)
        end
        if has_option("-now") then
                bootnow = "1"
        else
```

```
                bootnow = "0"
        end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot20221021.lua"
```

```
local ntbootfile_ext = ""
local bootfile = ""
temp = os.getenv("temp")
systemdrive = os.getenv("systemdrive")
APP_Path = os.getenv("WINXSHELL_MODULEPATH")

-- 提取固定值
local exitcode, stdout = winapi.execute("cmd /c mountvol")
stdout = stdout:gsub("\r\n", "\n")
bios_str = stdout:match("EFI ([^\n]+)\n")

--获取系统版本
function get_winver()
        local exitcode, stdout = winapi.execute([[ver]])
        local winver_str = stdout:match("10.0.")
        --stdout = stdout:gsub("\r\n", "\n")
        --winapi.show_message('错误', winver_str)
        if winver_str ~= nil then
                winver = "Win10/Win11"
        else
                winver = "win7"
        end
end

function createBCD(boot_platform)
       if boot_platform == "bios" then

            exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\\Boot]])

            exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] ..
boot_drv .. [[\]])
```

```
                exec("/hide /wait", [[bcdedit /createstore ]] .. boot_drv ..
[[\Boot\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. boot_drv .. [[\\Boot\\BCD.tmp
/create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. boot_drv ..
[[\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
boot_drv)
                end
                if boot_platform == "uefi" then
            exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\\EFI\\Microsoft\\Boot]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Boot\\bootx64.efi]])



                exec("/hide /wait", [[bcdedit /createstore ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
boot_drv)
                end


end
```

```
function cs_patition()
        boot_drv = Dialog:BrowseFolder("当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区", 17)
        if boot_platform == "bios" and File.exists(boot_drv .. [[\Boot\BCD]]) then
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\Boot\BCD]])

        elseif boot_platform == "bios" and not File.exists(boot_drv .. [[\Boot\BCD]]) then
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\Boot\BCD]])

        elseif boot_platform == "uefi" and File.exists(boot_drv .. [[\Boot\BCD]]) then
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\EFI\Microsoft\Boot\BCD]])

        elseif boot_platform == "uefi" and not File.exists(boot_drv .. [[\EFI\Microsoft\Boot\BCD]]) then
        createBCD(boot_platform)
                exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\EFI\Microsoft\Boot\BCD]])
end
end

function check_offline_os()
        if exec("/wait", [[bcdedit]]) == 1 then
                cs_patition()
                boot_bcd_var()
        else
                boot_bcd_var()
        end
```

```
end

function boot_bcd_var()
        if boot_platform == "bios" then
                local exitcode, boot_drv_stdout = winapi.execute("cmd /c bcdedit /enum {bootmgr}")
                boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
                boot_drv = boot_drv_stdout:match("device                  partition=([^\n]+)\n")
                bcd_store = boot_drv .. [[\Boot\BCD]]
        elseif boot_platform == "uefi" then
                local exitcode, boot_drv_stdout = winapi.execute("cmd /c bcdedit /enum {default}")
                boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
                boot_drv = boot_drv_stdout:match("device                  partition=([^\n]+)\n")
                local exitcode, bcd_drv_stdout = winapi.execute("cmd /c bcdedit /enum {bootmgr}")
                bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
                bcd_drv = bcd_drv_stdout:match("device                  partition=([^\n]+)\n")
                bcd_store = boot_drv .. [[\EFI\Microsoft\Boot\BCD]]
        end
end

--检测环境
function ck_bios()
        if bios_str == nil then
                bios_str = "传统BIOS"
                boot_platform = "bios"
                wimwinload = [[\Windows\system32\boot\winload.exe]]
                vhdwinload = [[\Windows\system32\winload.exe]]
        else
                bios_str = "UEFI"
                boot_platform = "uefi"
                wimwinload = [[\Windows\system32\boot\winload.efi]]
                vhdwinload = [[\Windows\system32\winload.efi]]
        end
end

function ntboot_init()
        if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:"  then
         exec("/hide", [[cmd /c mkdir "X:\program files\ms" ]])
          exec("/hide", [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])
            get_winver()
                ntboot_addmode = "pe"
                ck_bios()
                check_offline_os()
                ntboot_runmode = "WinPE环境模式![离线系统盘符->" .. boot_drv .. "] [启动环境->" .. bios_str .. "]"
        else
                get_winver()
                boot_drv = os.getenv("systemdrive")
                ck_bios()
                ntboot_addmode = "os"
                ntboot_runmode = "正常系统环境![系统盘符->" .. boot_drv .. "] [启动环境->" .. bios_str .. "]"
        end
end

function getBootfileExt(bootfile)
        bootfile_ext = bootfile:match(".+%.(%w+)¥")
        bootfile_ext = string.lower(bootfile_ext) --转小写
        --转大写 bootfile_ext = string.upper(bootfile_ext)
        if bootfile_ext == "wim" then
                addfile = "wim"
                bootok = "1"
        elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
                addfile = "vhd"
                bootok = "1"
        end
end

function ntboot_args(bootfile)
        getBootfileExt(bootfile)
        if bootok ~= "1" then
                winapi.show_message("错误", "暂未支持启动这种格式")
                return
        end
        if bootnow == "1" then
                bootsdi(bootfile)
                ntbootnow(bootfile)
        else
                --vhd
                bootsdi(bootfile)
                ntbootadd(bootfile)
        end
end

--分离boot.sdi路径
function bootsdi(bootfile)
        --wim或vhd文件 所在分区相对路径
        boot_file_path = bootfile:match(":([^] %s]+)")
        --wim或vhd文件 所在分区
        boot_file_partition = bootfile:match("([^] %s]+):")
        sdi = [[]] .. boot_drv .. [[\pe\boot.sdi]]
        --boot.sdi 所在分区相对路径
        --alert(sdi)
        boot_sdi_path = sdi:match(":([^] %s]+)")
        --boot.sdi 所在分区
        boot_sdi_partition = sdi:match("([^] %s]+):")
end




function win7_replacebootmgr()
        exec("/show /wait", "cmd /c mountvol B: /s")
        if addfile == "vhd" then
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
        end
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
        exec("/hide /wait", "cmd /c mountvol B: /d")
end

function Add_Boot(addfile)
        if winver == "win7" then
```

```
            win7_replacebootmgr()
        end
        if addfile == "wim" then
            ID = "{19260817-6666-8888-f00d-caffee000000}"
            exec("/hide /wait", [[bcdedit /delete ]] .. ID)
            exec("/hide /wait", [[bcdedit /create {ramdiskoptions} /d "ntboot ramdisk"]])
            exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
            exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdidevice
partition=]] .. boot_sdi_partition .. [[:]])
            exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdipath ]] ..
boot_sdi_path)
            exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "BootWim With NTBOOT
6.x" /application osloader]])
            exec(
                "/hide /wait",
                [[bcdedit -set ]] ..
                ID .. " device ramdisk=[" .. boot_file_partition .. ":]" ..
boot_file_path .. [[,{ramdiskoptions}]]
            )
            exec(
                "/hide /wait",
                [[bcdedit -set ]] ..
                ID .. " osdevice ramdisk=[" .. boot_file_partition .. ":]" ..
boot_file_path .. [[,{ramdiskoptions}]]
            )
            exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. wimwinload)
            exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
            exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
            exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
            exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ winpe yes]])
            exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
```

```
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
        elseif addfile == "vhd" then
                ID = "{19260817-6666-8888-f00d-caffee000001}"
                exec("/hide /wait", [[bcdedit /delete ]] .. ID)
                exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
                --exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
                exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "Boot from VHD With NTBOOT 6.x" /application osloader]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " device vhd=[" .. boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " osdevice vhd=[" .. boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
        end
end


function ntbootnow(bootfile)
        --创建pe目录
```

```
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi" ]] .. boot_drv .. [[\pe]])
    end
    addmode = "/bootsequence"
    say = "添加一次性启动完成，系统将立即重启!"
    Add_Boot(addfile)
    if ntboot_addmode == "pe" then
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
            local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum all|find "]] .. boot_file_path .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        dojob = "wpeutil reboot"
    else
        local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum all|find "]] .. boot_file_path .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        dojob = "shutdown -r -t 10"
    end

    if bcdedit_r == nil then
        winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
        sui:close()
    else
        exec("/hide", [[cmd /c ]] .. dojob)
        r = winapi.show_message("系统即将重启", "系统即将重启并进入" .. bootfile .. "\n是否取消？【如果是PE环境取消无效】",
            "yes-no", "warning")
    end
```

```
        if r == "yes" then
                exec("/hide", [[cmd /c shutdown -a]])
                winapi.show_message("注意", "您虽然取消了立即启动到镜像，\n但重启后还是会启动一次的，\n再重启就好了，莫慌!")
                sui:close()
        end
end

function ntboot_ask(bootfile)
alert(bios_str)
    ck_bios()
        if boot_drv == nil then
        --boot_drv = os.getenv("systemdrive")
        ntboot_init()

        end
        if winver == nil then
        get_winver()
        end
        bootsdi(bootfile)
        local r =
        winapi.show_message(
                "操作镜像",
                "［是］立即重启到镜像并只启动一次\n\n［否］添加到菜单以后启动\n\n［取消］不操作\n\n 当前启动文件："
                ..
                bootfile ..
                "\n\n 启动环境：" ..
                bios_str ..
                "\n 当前系统为：" ..
                winver ..
```

```
            "\n\n boot.sdi所在分区：" ..
            boot_sdi_partition ..
            "\n boot.sdi相对路径：" ..
            boot_sdi_path .. "\n\n 启动镜像盘符：" .. boot_file_partition .. "\n 启动镜像：" .. boot_file_path,
            "yes-no-cancel",
            "warning"
        )
        if r == "yes" then
            ntbootnow(bootfile)
        elseif r == "no" then
            ntbootadd(bootfile)
        elseif r == "cancel" then
            return
            --winapi.show_message("您取消了操作镜像", bootfile)
        end
end

function ntbootadd(bootfile)
        --创建pe目录
        if addfile == "wim" then
            exec("/hide /wait", [[mkdir ]] .. boot_drv .. [[\pe]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi" ]] .. boot_drv .. [[\pe]])
        end
        addmode = "/displayorder"
        say = "添加永久启动项完成!"
        Add_Boot(addfile)
        if ntboot_addmode == "pe" then
            --alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID .. [["]])
```

```
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID .. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        else
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID .. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
        end

        --alert([[bcdedit /enum |find "]] .. ID .. [["]])
        if bcdedit_r == nil then
                winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
        else
                winapi.show_message("选择了添加到启动菜单", say)
                sui:close()
        end
end

function Add_Boot_Option(addfile)
        ntboot_init()
        --if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
        if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
                toadmin = winapi.show_message("没有以管理员身份运行!", "请右键管理员身份运行本程序!, \n是否现在就尝试管理员身份重新运行本程序？"
                          , "yes-no", "warning")
        end
        if toadmin == "yes" then
                --exec('/hide', [[cmd /c shutdown -a]])
                --winapi.show_message('注意', "执行管理员提升权限操作")
                exec("/admin", [["]] .. APP_Path .. [[\WinXshell.exe" -ui -jcfg
wxsUI\UI_AppStore\main.jcfg]])
                sui:close()
                return
        elseif toadmin == "no" then
                return
        end
        --winapi.show_message('添加文件类型', addfile)
        local filter = "启动镜像|*vhdx;*.vhdx|所有文件|*.*"
        if addfile == "wim" then
                filter = "启动镜像|*wim|所有文件|*.*"
        end
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile .. "文件 当前为:" ..
ntboot_runmode, filter)

        if bootfile == "" then
                --winapi.show_message("你犯了个错误!", "你没有选择镜像!")
                return
        else
                ntboot_ask(bootfile)
        end
end

function helper_ntboot_onload()
        --addwim()
        if has_option("-ntboot") then
                bootfile = get_option("-ntboot")
                ntboot_args(bootfile)
        end
        if has_option("-now") then
                bootnow = "1"
        else
```

```
                    bootnow = "0"
          end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_ntboot20221021.lua"
```

```
local ntbootfile_ext = ""
local bootfile = ""
local getfullpath_func = type(File.GetFullPath)
if getfullpath_func == "function" then
temp = File.GetFullPath(os.getenv("temp"))
else
temp = os.getenv("temp")
end
systemdrive = os.getenv("systemdrive")

-- 提取固定值
local exitcode, stdout = winapi.execute("cmd /c mountvol")
stdout = stdout:gsub("\r\n", "\n")
bios_str = stdout:match("EFI ([^\n]+)\n")

-- 获取系统版本
function get_winver()
    local exitcode, stdout = winapi.execute([[ver]])
    local winver_str = stdout:match("10.0.")
    -- stdout = stdout:gsub("\r\n", "\n")
    -- winapi.show_message('错误', winver_str)
    if winver_str ~= nil then
        winver = "Win10/Win11"
    else
        winver = "win7"
    end
end

-- 检测环境
function ck_bios()
    if bios_str == nil then
        bios_str = "传统BIOS"
        boot_platform = "bios"
        wimwinload = [[\Windows\system32\boot\winload.exe]]
        vhdwinload = [[\Windows\system32\winload.exe]]
    else
            bcd_drv = "B:"
        bios_str = "UEFI"
        boot_platform = "uefi"
        wimwinload = [[\Windows\system32\boot\winload.efi]]
        vhdwinload = [[\Windows\system32\winload.efi]]
    end
end

--执行批处理代码
function execlines(cmds)

for line in cmds:gmatch("([^\n]*)\n?") do

             --alert(line)
             exec("/hide /wait", line)

end
end


ck_bios()

App_platform = sui:find("boot_platform")
App_platform.text = "当前环境:" .. bios_str



function createBCD(boot_platform)
  if boot_platform == "bios" then

            exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\Boot]])

            exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] .. bcd_drv .. [[\]])
               exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv .. [[\]])

               exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\Boot\BCD.tmp]])
               exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
               exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\\Boot\\BCD.tmp]])
```

```
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
                end
 if boot_platform == "uefi" then
        local bcd_drv = "B:"
            exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Microsoft\\Boot]])
                exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Boot]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv .. [[\\EFI\\Boot\\bootx64.efi]])



                exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows
Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
                end
end

function cs_patition()
```

```
    if boot_platform == "bios" then
        bcd_drv = Dialog:BrowseFolder(
                  "当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区",
                  17)
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] ..
                 bcd_drv .. [[\Boot\BCD]])
    elseif boot_platform == "uefi" and File.exists([[B:\EFI\Microsoft\Boot\BCD]]) then

           exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    elseif boot_platform == "uefi" and
        not File.exists([[B:\EFI\Microsoft\Boot\BCD]]) or not File.exists([[B:\EFI\Boot\Bootx64.efi]]) then
        createBCD(boot_platform)
        exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    end
end


function check_offline_os()
    if exec("/wait", [[bcdedit]]) == 1 then
        cs_patition()
        boot_bcd_var()
    else
        boot_bcd_var()
    end
end

function boot_bcd_var()
    if boot_platform == "bios" then
        local exitcode, boot_drv_stdout = winapi.execute(
                                               "cmd /c bcdedit /enum {bootmgr}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        bcd_store = boot_drv .. [[\Boot\BCD]]
        bcd_drv = boot_drv
    elseif boot_platform == "uefi" then
        exec("/show /wait", "cmd /c mountvol B: /s")
                local bcd_drv = "B:"
        local exitcode, boot_drv_stdout = winapi.execute(
                                               "cmd /c bcdedit /enum {default}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        local exitcode, bcd_drv_stdout = winapi.execute(
                                              "cmd /c bcdedit /enum {bootmgr}")
        bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
        bcd_drv = bcd_drv_stdout:match(
                      "device                  partition=([^\n]+)\n")
        bcd_store = bcd_drv .. [[\EFI\Microsoft\Boot\BCD]]
    end
end




function ntboot_init()
    if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:" then

        exec("/hide /wait", [[cmd /c mkdir "X:\program files\ms" ]])
        exec("/hide",
             [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])

        get_winver()

ck_bios()

        ntboot_addmode = "pe"

        check_offline_os()
        ntboot_runmode = "WinPE环境模式!(离线系统[启动]盘符->" .. bcd_drv ..
                             "] [启动环境->" .. bios_str .. "]"
        APP_Path = os.getenv("WINXSHELL_MODULEPATH")
    else
        get_winver()
                ck_bios()
        boot_drv = os.getenv("systemdrive")
        bcd_drv = os.getenv("systemdrive")
        ntboot_addmode = "os"
        ntboot_runmode = "正常系统环境!(系统[启动]盘符)->" .. boot_drv ..
                            "] [启动环境->" .. bios_str .. "]"
    end
end

function getBootfileExt(bootfile)
    bootfile_ext = bootfile:match(".+%.(%w+)¥")
    bootfile_ext = string.lower(bootfile_ext) -- 转小写

    -- 转大写 bootfile_ext = string.upper(bootfile_ext)
    if bootfile_ext == "wim" then
        addfile = "wim"
        bootok = "1"

    elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
        addfile = "vhd"
        bootok = "1"

    elseif bootfile_ext == "efi" and boot_platform == "uefi"then
```

```
        addfile = "efi"
        bootok = "1"
    end
end

function ntboot_args(bootfile)
   ntboot_init()
    getBootfileExt(bootfile)
    if bootok ~= "1" then
        winapi.show_message("错误", "暂未支持启动这种格式")
        return
    end
    if bootnow == "1" then

        bootsdi(bootfile)
        ntbootnow(bootfile)
    else
        -- vhd
        bootsdi(bootfile)
        ntbootadd(bootfile)
    end
end

-- 分离boot.sdi路径
function bootsdi(bootfile)
```

```
    -- wim或vhd文件 所在分区相对路径
    boot_file_path = bootfile:match(":([^] %s]+)")
    -- wim或vhd文件 所在分区
    boot_file_partition = bootfile:match("([^] %s]+):")
    sdi = [[]] .. bcd_drv .. [[\pe\boot.sdi]]
    -- boot.sdi 所在分区相对路径
    -- alert(sdi)
    boot_sdi_path = sdi:match(":([^] %s]+)")
    -- boot.sdi 所在分区
    boot_sdi_partition = sdi:match("([^] %s]+):")
end

function win7_replacebootmgr()
    exec("/show /wait", "cmd /c mountvol B: /s")
    if addfile == "vhd" then
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
    end
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
            [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
             [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
    exec("/hide /wait", "cmd /c mountvol B: /d")
end

function Add_Boot(addfile)
    if winver == "win7" then win7_replacebootmgr() end
    if addfile == "wim" then
        ID = "{19260817-6666-8888-f00d-caffee000000}"
                ramdiskID = "{19260817-6666-8888-f00d-caffee000010}"
        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait",
             [[bcdedit /create ]] .. ramdiskID .. [[ /device]])
                 exec("/hide /wait",
             [[bcdedit /set  ]] .. ramdiskID .. [[ ramdisksdidevice  partition=]] ..
                 boot_sdi_partition .. [[:]])
        exec("/hide /wait",
             [[bcdedit /set  ]] .. ramdiskID ..  [[ ramdisksdipath ]] .. boot_sdi_path)

        exec("/hide /wait",
             [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])

        exec("/hide /wait", [[bcdedit -create ]] .. ID ..
                 [[ /d "BootWim With NTBOOT 6.x" /application osloader]])
        exec("/hide /wait",
             [[bcdedit -set ]] .. ID .. " device ramdisk=[" ..
```

```
                boot_file_partition .. ":]" .. boot_file_path ..
                [[,]] .. ramdiskID)
        exec("/hide /wait",
            [[bcdedit -set ]] .. ID .. " osdevice ramdisk=[" ..
                boot_file_partition .. ":]" .. boot_file_path ..
                [[,]] .. ramdiskID)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. wimwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
        exec("/hide /wait",
            [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
    elseif addfile == "vhd" then
        ID = "{19260817-6666-8888-f00d-caffee000003}"

        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait",
            [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
        -- exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
        exec("/hide /wait", [[bcdedit -create ]] .. ID ..
                 [[ /d "Boot from VHD With NTBOOT 6.x" /application osloader]])
        exec("/hide /wait",
             [[bcdedit -set ]] .. ID .. " device vhd=[" .. boot_file_partition ..
                 ":]" .. boot_file_path)
        exec("/hide /wait",
             [[bcdedit -set ]] .. ID .. " osdevice vhd=[" .. boot_file_partition ..
                 ":]" .. boot_file_path)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait",
             [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
          elseif addfile == "efi" then
                local cmdlines = [[
cmd /c copy "%bootapp_file%" B:\EFI\Boot\
bcdedit /set %guid% path \EFI\Boot\%bootapp_name%
bcdedit /deletevalue %guid% locale
bcdedit /deletevalue %guid% inherit
bcdedit /deletevalue %guid% default
bcdedit /deletevalue %guid% displayorder
bcdedit /deletevalue %guid% timeout
bcdedit /deletevalue %guid% nointegritychecks
bcdedit /deletevalue %guid% testsigning
bcdedit %addmode% %guid% /addlast
]]
os.setenv('addmode', addmode)
local boot_file_path = string.match(boot_file_path, "([^\\]+)¥")
os.setenv('bootapp_name', boot_file_path)
os.setenv('bootapp_file', bootfile)



local exitcode, stdout = winapi.execute([["bcdedit /copy {bootmgr} /d "%bootapp_name%"]])
ID = string.match(stdout, "{[0-9a-fA-F%-]+}")
os.setenv('guid', ID)
cmdlines = cmdlines:envstr()
execlines(cmdlines)

    end
end

function ntbootnow(bootfile)
    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
                 [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/bootsequence"
    say = "添加一次性启动完成，系统将立即重启!"
    Add_Boot(addfile)

    if ntboot_addmode == "pe" then

        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum all|find "]] ..
                                               ID .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        dojob = "wpeutil reboot"
    else
        local exitcode, bcdedit_enum = winapi.execute(
                                           [[bcdedit /enum all|find "]] ..
                                               ID .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        dojob = "shutdown -r -t 10"
    end
```

```
    if bcdedit_r == nil then

        winapi.show_message("注意",
                           "添加失败，可能没有权限(需要右键->管理员运行)")
     --如果是appstore运行就不关闭   sui:close()
    else
        exec("/hide", [[cmd /c ]] .. dojob)
        r = winapi.show_message("系统即将重启",
                               "系统即将重启并进入" .. bootfile ..
                                   "\n是否取消？【如果是PE环境取消无效】",
                               "yes-no", "warning")
    end
    if r == "yes" then
        exec("/hide", [[cmd /c shutdown -a]])
        winapi.show_message("注意",
                           "您虽然取消了立即启动到镜像，\n但重启后还是会启动一次的，\n再重启就好了，莫慌!")
         --如果是appstore运行就不关闭  sui:close()
    end
end

function ntboot_ask(bootfile)
    if bios_str == nil then ck_bios() end
    if bcd_drv == nil then
        -- boot_drv = os.getenv("systemdrive")
        ntboot_init()
    end
    if winver == nil then get_winver() end
    bootsdi(bootfile)
    local r = winapi.show_message("操作镜像",
                                  "[是] 立即重启到镜像并只启动一次\n\n[否] 添加到菜单以后启动\n\n[取消] 不操作\n\n当前启动文件：" ..
                                      bootfile .. "\n\n启动环境：" ..
                                      bios_str .. "\n当前系统为：" ..
                                      winver .. "\n\nboot.sdi所在分区：" ..
                                      boot_sdi_partition ..
                                      "\nboot.sdi相对路径：" ..
                                      boot_sdi_path ..
                                      "\n\n启动镜像盘符：" ..
                                      boot_file_partition .. "\n启动镜像：" ..
                                      boot_file_path, "yes-no-cancel", "warning")
    if r == "yes" then
        ntbootnow(bootfile)
    elseif r == "no" then
        ntbootadd(bootfile)
    elseif r == "cancel" then
        return
        -- winapi.show_message("您取消了操作镜像", bootfile)
    end
end
```

```
function ntbootadd(bootfile)


    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
         exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
                 [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/displayorder"
    say = "添加永久启动项完成!"
    Add_Boot(addfile)
    if ntboot_addmode == "pe" then

        -- alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID .. [["]])

        local exitcode, bcdedit_enum = winapi.execute(
                                          [[bcdedit /enum all |find "]] .. ID ..
                                              [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)

    else
        local exitcode, bcdedit_enum = winapi.execute(
                                            [[bcdedit /enum all |find "]] .. ID ..
                                                [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
    end
    -- alert([[bcdedit /enum |find "]] .. ID .. [["]])
    if bcdedit_r == nil then

        winapi.show_message("注意",
                            "添加失败，可能没有权限(需要右键->管理员运行)")
    else
        winapi.show_message("选择了添加到启动菜单", say)
        --如果是appstore运行就不关闭    sui:close()
    end
end

function Add_Boot_Option(addfile)
    ntboot_init()
    -- if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
    if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
        toadmin = winapi.show_message("没有以管理员身份运行!",
                                      "请右键管理员身份运行本程序!，\n是否现在就尝试管理员身份重新运行本程序？",
                                      "yes-no", "warning")
    end
    if toadmin == "yes" then
        -- exec('/hide', [[cmd /c shutdown -a]])
        -- winapi.show_message('注意', "执行管理员提升权限操作")
        exec("/admin", [["]] .. APP_Path ..
                 [[\WinXshell.exe" -ui -jcfg wxsUI\UI_AppStore\main.jcfg]])
        sui:close()
        return
    elseif toadmin == "no" then
        return
    end
    -- winapi.show_message('添加文件类型', addfile)
    local filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
    if addfile == "wim" then filter = "启动镜像|*wim|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                                 "文件 当前为:" .. ntboot_runmode, filter)
    elseif addfile == "vhd" then filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                                 "文件 当前为:" .. ntboot_runmode, filter)
    elseif addfile == "raw" then
        local filter = "启动镜像|*wim;*vhd;*vhdx;*efi|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的文件 当前为:" .. ntboot_runmode, filter)
        getBootfileExt(bootfile)
        bootsdi(bootfile)
        end
    if bootfile == "" then
        -- winapi.show_message("你犯了个错误!", "你没有选择镜像!")
        return
    else

        ntboot_ask(bootfile)
    end
end

function helper_ntboot_onload()

    -- addwim()
    if has_option("-ntboot") then
        bootfile = get_option("-ntboot")
        ntboot_args(bootfile)
    end
    if has_option("-now") then
        bootnow = "1"
    else
        bootnow = "0"
    end
end



function check_boot()
    if has_option("-now") or has_option("-r")  then
                bootnow = "1"
        else
```

```
                bootnow = "0"
        end
end
        --addwim()


        if has_option("-ntboot") then
        bootfile = get_option("-ntboot")
                check_boot()
        ntboot_args(bootfile)
   elseif has_option("-f")  then
        bootfile = get_option("-f")
        check_boot()
        ntboot_args(bootfile)
        else

        return

        end--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot20240423.lua"
```

```
local ntbootfile_ext = ""

local bootfile = ""

local getfullpath_func = type(File.GetFullPath)

if getfullpath_func == "function" then

temp = File.GetFullPath(os.getenv("temp"))

else

temp = os.getenv("temp")

end

systemdrive = os.getenv("systemdrive")


-- 提取固定值

local exitcode, stdout = winapi.execute("cmd /c mountvol")

stdout = stdout:gsub("\r\n", "\n")

bios_str = stdout:match("EFI ([^\n]+)\n")


-- 获取系统版本

function get_winver()

    local exitcode, stdout = winapi.execute([[ver]])

    local winver_str = stdout:match("10.0.")

    -- stdout = stdout:gsub("\r\n", "\n")

    -- winapi.show_message('错误', winver_str)

    if winver_str ‾= nil then

        winver = "Win10/Win11"

    else

        winver = "win7"
    end
end

-- 检测环境
function ck_bios()
    if bios_str == nil then
        bios_str = "传统BIOS"
        boot_platform = "bios"
        wimwinload = [[\Windows\system32\boot\winload.exe]]
        vhdwinload = [[\Windows\system32\winload.exe]]
    else
            bcd_drv = "B:"
        bios_str = "UEFI"
        boot_platform = "uefi"
        wimwinload = [[\Windows\system32\boot\winload.efi]]
        vhdwinload = [[\Windows\system32\winload.efi]]
    end
end

--执行批处理代码
function execlines(cmds)

for line in cmds:gmatch("([^\n]*)\n?") do

             --alert(line)
            exec("/hide /wait", line)

end
end


ck_bios()

App_platform = sui:find("boot_platform")
App_platform.text = "当前环境:" .. bios_str



function createBCD(boot_platform)
   if boot_platform == "bios" then

           exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\Boot]])

           exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] .. bcd_drv .. [[\]])
               exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv .. [[\]])

               exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\Boot\BCD.tmp]])
               exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
               exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\\Boot\\BCD.tmp]])
```

```
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
                end
 if boot_platform == "uefi" then
        local bcd_drv = "B:"
            exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Microsoft\\Boot]])
                exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\\EFI\\Boot]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootvhd.dll" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv ..
[[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]] .. bcd_drv .. [[\\EFI\\Boot\\bootx64.efi]])



                exec("/hide /wait", [[bcdedit /createstore ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows
Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. bcd_drv .. [[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] .. bcd_drv)
                end
end

function cs_patition()
```

```
    if boot_platform == "bios" then
        bcd_drv = Dialog:BrowseFolder(
                  "当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区",
                  17)
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] ..
                 bcd_drv .. [[\Boot\BCD]])
    elseif boot_platform == "uefi" and File.exists([[B:\EFI\Microsoft\Boot\BCD]]) then


           exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    elseif boot_platform == "uefi" and
        not File.exists([[B:\EFI\Microsoft\Boot\BCD]]) or not File.exists([[B:\EFI\Boot\Bootx64.efi]]) then
        createBCD(boot_platform)
        exec("/hide /wait",
             [[reg load hklm\BCD_]] .. boot_platform .. [[ B:\EFI\Microsoft\Boot\BCD]])
    end
end



function check_offline_os()
    if exec("/wait", [[bcdedit]]) == 1 then
        cs_patition()
        boot_bcd_var()
    else
        boot_bcd_var()
    end
end


function boot_bcd_var()
    if boot_platform == "bios" then
        local exitcode, boot_drv_stdout = winapi.execute(
                                               "cmd /c bcdedit /enum {bootmgr}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        bcd_store = boot_drv .. [[\Boot\BCD]]
        bcd_drv = boot_drv
    elseif boot_platform == "uefi" then
        exec("/show /wait", "cmd /c mountvol B: /s")
                local bcd_drv = "B:"
        local exitcode, boot_drv_stdout = winapi.execute(
                                               "cmd /c bcdedit /enum {default}")
        boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
        boot_drv = boot_drv_stdout:match(
                       "device                  partition=([^\n]+)\n")
        local exitcode, bcd_drv_stdout = winapi.execute(
                                              "cmd /c bcdedit /enum {bootmgr}")
        bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
        bcd_drv = bcd_drv_stdout:match(
                    "device                  partition=([^\n]+)\n")
        bcd_store = bcd_drv .. [[\EFI\Microsoft\Boot\BCD]]
    end
end



function ntboot_init()
    if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:" then

        exec("/hide /wait", [[cmd /c mkdir "X:\program files\ms" ]])
        exec("/hide",
             [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])

        get_winver()

ck_bios()

        ntboot_addmode = "pe"

        check_offline_os()
        ntboot_runmode = "WinPE环境模式!(离线系统[启动]盘符->" .. bcd_drv ..
                          "] [启动环境->" .. bios_str .. "]"
        APP_Path = os.getenv("WINXSHELL_MODULEPATH")
    else
        get_winver()
                ck_bios()
        boot_drv = os.getenv("systemdrive")
        bcd_drv = os.getenv("systemdrive")
        ntboot_addmode = "os"
        ntboot_runmode = "正常系统环境!(系统[启动]盘符)->" .. boot_drv ..
                             "] [启动环境->" .. bios_str .. "]"
    end
end

function getBootfileExt(bootfile)
    bootfile_ext = bootfile:match(".+%.(%w+)¥")
        if bootfile_ext ‾= nil then
    bootfile_ext = string.lower(bootfile_ext) -- 转小写
        end

    -- 转大写 bootfile_ext = string.upper(bootfile_ext)
    if bootfile_ext == "wim" then
        addfile = "wim"
        bootok = "1"

    elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
        addfile = "vhd"
        bootok = "1"
```

```lua
    elseif bootfile_ext == "efi" and boot_platform == "uefi"then
        addfile = "efi"
        bootok = "1"
    end
end

function ntboot_args(bootfile)
   ntboot_init()
    getBootfileExt(bootfile)
    if bootok ~= "1" then
        winapi.show_message("错误", "暂未支持启动这种格式")
        return
    end
    if bootnow == "1" then

        bootsdi(bootfile)
        ntbootnow(bootfile)
    else
        -- vhd
        bootsdi(bootfile)
        ntbootadd(bootfile)
    end
end
```

```
-- 分离boot.sdi路径
function bootsdi(bootfile)
    -- wim或vhd文件 所在分区相对路径
    boot_file_path = bootfile:match(":([^] %s]+)")
    -- wim或vhd文件 所在分区
    boot_file_partition = bootfile:match("([^] %s]+):")
    sdi = [[]] .. bcd_drv .. [[\pe\boot.sdi]]
    -- boot.sdi 所在分区相对路径
    -- alert(sdi)
    boot_sdi_path = sdi:match(":([^] %s]+)")
    -- boot.sdi 所在分区
    boot_sdi_partition = sdi:match("([^] %s]+):")
end


function win7_replacebootmgr()
    exec("/show /wait", "cmd /c mountvol B: /s")
    if addfile == "vhd" then
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
    end
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
             [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
             [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
    exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
             [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
    exec("/hide /wait", "cmd /c mountvol B: /d")
end


function Add_Boot(addfile)
    if winver == "win7" then win7_replacebootmgr() end
    if addfile == "wim" then
        math.randomseed(os.time())
        local random_number = string.format("%06d", math.random(0, 999999))

        ID = [["{19260817-6666-8888-f00d-caffee]] .. random_number .. [[}"]]
                ramdiskID = "{19260817-6666-8888-f00d-caffee000010}"
local cmdlines = [[
bcdedit.exe /create %ramdiskid% /device
bcdedit.exe /set %ramdiskid%  ramdisksdidevice partition=%boot_sdi_partition%:
bcdedit.exe /set %ramdiskid%  ramdisksdipath \pe\boot.sdi
bcdedit.exe /create %ramdiskid% /device
bcdedit.exe /create %guid% /d %bootapp_name% /application osloader
bcdedit.exe /set %guid% Description %bootapp_name%
bcdedit.exe /set %guid% Device ramdisk=[%boot_file_partition%:]%boot_file_path%,%ramdiskid%
bcdedit.exe /set %guid% OSDevice ramdisk=[%boot_file_partition%:]%boot_file_path%,%ramdiskid%
bcdedit.exe /set %guid% systemroot \windows
bcdedit.exe /set %guid% detecthal Yes
bcdedit.exe /set %guid% winpe Yes
bcdedit.exe /set %guid% highestmode Yes
bcdedit.exe /set %guid% bootmenupolicy legacy
bcdedit.exe -displayorder %guid% -addlast
]]
os.setenv(′ramdiskid′, ramdiskID)
os.setenv(′guid′, ID)
local boot_file_name = string.match(boot_file_path, "([^\\]+)¥")
os.setenv(′bootapp_name′, boot_file_name)
os.setenv(′boot_file_path′, boot_file_path)
os.setenv(′boot_file_partition′, boot_file_partition)
os.setenv(′boot_sdi_partition′, boot_sdi_partition)
os.setenv(′bootapp_file′, bootfile)
os.setenv(′guid′, ID)
cmdlines = cmdlines:envstr()
execlines(cmdlines)
    elseif addfile == "vhd" then
        ID = "{19260817-6666-8888-f00d-caffee000003}"

        exec("/hide /wait", [[bcdedit /delete ]] .. ID)
        exec("/hide /wait",
            [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
        exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
        -- exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
        exec("/hide /wait", [[bcdedit -create ]] .. ID ..
                  [[ /d "Boot from VHD With NTBOOT 6.x" /application osloader]])
        exec("/hide /wait",
              [[bcdedit -set ]] .. ID .. " device vhd=[" .. boot_file_partition ..
                  ":]" .. boot_file_path)
        exec("/hide /wait",
              [[bcdedit -set ]] .. ID .. " osdevice vhd=[" .. boot_file_partition ..
                  ":]" .. boot_file_path)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
        exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
        -- exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                 exec("/hide /wait",
              [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[ /addlast]])
          elseif addfile == "efi" then
                 local cmdlines = [[
cmd /c copy "%bootapp_file%" B:\EFI\Boot\
bcdedit /set %guid% path \EFI\Boot\%bootapp_name%
bcdedit /deletevalue %guid% locale
bcdedit /deletevalue %guid% inherit
bcdedit /deletevalue %guid% default
bcdedit /deletevalue %guid% displayorder
bcdedit /deletevalue %guid% timeout
bcdedit /deletevalue %guid% nointegritychecks
bcdedit /deletevalue %guid% testsigning
bcdedit %addmode% %guid% /addlast
]]
os.setenv('addmode', addmode)
local boot_file_name= string.match(boot_file_path, "([^\\]+)¥")
os.setenv('bootapp_name', boot_file_name)
os.setenv('bootapp_file', bootfile)



local exitcode, stdout = winapi.execute([["bcdedit /copy {bootmgr} /d "%bootapp_name%"]])
ID = string.match(stdout, "{[0-9a-fA-F%-]+}")
os.setenv('guid', ID)
cmdlines = cmdlines:envstr()
execlines(cmdlines)

    end
end

function ntbootnow(bootfile)
    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
                 [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/bootsequence"
    say = "添加一次性启动完成，系统将立即重启!"
    Add_Boot(addfile)

    if ntboot_addmode == "pe" then

        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        local exitcode, bcdedit_enum = winapi.execute(
                                            [[bcdedit /enum all|find "]] ..
                                                ID .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        dojob = "wpeutil reboot"
    else
        local exitcode, bcdedit_enum = winapi.execute(
                                            [[bcdedit /enum all|find "]] ..
                                                ID .. [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        dojob = "shutdown -r -t 10"
    end

    if bcdedit_r == nil then

        winapi.show_message("注意",
                    "添加失败，可能没有权限(需要右键->管理员运行)")
     --如果是appstore运行就不关闭   sui:close()
    else
        exec("/hide", [[cmd /c ]] .. dojob)
        r = winapi.show_message("系统即将重启",
                    "系统即将重启并进入" .. bootfile ..
                        "\n是否取消？【如果是PE环境取消无效】",
                    "yes-no", "warning")
    end
    if r == "yes" then
        exec("/hide", [[cmd /c shutdown -a]])
        winapi.show_message("注意",
                    "您虽然取消了立即启动到镜像，\n但重启后还是会启动一次的，\n再重启就好了，莫慌!")
         --如果是appstore运行就不关闭  sui:close()
    end
end

function ntboot_ask(bootfile)
    if bios_str == nil then ck_bios() end
    if bcd_drv == nil then
```

```
        -- boot_drv = os.getenv("systemdrive")
        ntboot_init()
    end
    if winver == nil then get_winver() end
    bootsdi(bootfile)
    local r = winapi.show_message("操作镜像",
                                 " [是] 立即重启到镜像并只启动一次\n\n [否] 添加到菜单以后启动\n\n [取消] 不操作\n\n 当前启动文件：" ..
                                     bootfile .. "\n\n 启动环境：" ..
                                     bios_str .. "\n 当前系统为：" ..
                                     winver .. "\n\n boot.sdi所在分区：" ..
                                     boot_sdi_partition ..
                                     "\n boot.sdi相对路径：" ..
                                     boot_sdi_path ..
                                     "\n\n 启动镜像盘符：" ..
                                     boot_file_partition .. "\n 启动镜像：" ..
                                     boot_file_path, "yes-no-cancel", "warning")
    if r == "yes" then
        ntbootnow(bootfile)
    elseif r == "no" then
        ntbootadd(bootfile)
    elseif r == "cancel" then
        return
        -- winapi.show_message("您取消了操作镜像", bootfile)
    end
end
```

```
function ntbootadd(bootfile)

    -- 创建pe目录
    if addfile == "wim" then
        exec("/hide /wait", [[cmd /c mkdir ]] .. bcd_drv .. [[\pe]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path ..
                 [[\ms\boot.sdi" ]] .. bcd_drv .. [[\pe]])
         exec("/hide /wait", [[cmd /c copy /y "]] .. bcd_drv ..
                 [[\windows\system32\boot.sdi" ]] .. bcd_drv .. [[\pe]])
    end
    addmode = "/displayorder"
    say = "添加永久启动项完成!"
    Add_Boot(addfile)
    if ntboot_addmode == "pe" then

        -- alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID .. [["]])

        local exitcode, bcdedit_enum = winapi.execute(
                                          [[bcdedit /enum all |find "]] .. ID ..
                                              [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
        exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)

    else
```

```
        local exitcode, bcdedit_enum = winapi.execute(
                                            [[bcdedit /enum all |find "]] .. ID ..
                                                [["]])
        bcdedit_r = bcdedit_enum:match("{.+}")
    end
    -- alert([[bcdedit /enum |find "]] .. ID .. [["]])
    if bcdedit_r == nil then

        winapi.show_message("注意",
                            "添加失败，可能没有权限(需要右键->管理员运行)")
    else
        winapi.show_message("选择了添加到启动菜单", say)
        --如果是appstore运行就不关闭   sui:close()
    end
end

function Add_Boot_Option(addfile)
    ntboot_init()
    -- if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
    if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
        toadmin = winapi.show_message("没有以管理员身份运行!",
                                      "请右键管理员身份运行本程序!，\n是否现在就尝试管理员身份重新运行本程序？",
                                      "yes-no", "warning")
    end
    if toadmin == "yes" then
        -- exec('/hide', [[cmd /c shutdown -a]])
        -- winapi.show_message('注意', "执行管理员提升权限操作")
        exec("/admin", [["]] .. APP_Path ..
                 [[\WinXshell.exe" -ui -jcfg wxsUI\UI_AppStore\main.jcfg]])
        sui:close()
        return
    elseif toadmin == "no" then
        return
    end
    -- winapi.show_message('添加文件类型', addfile)
    local filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
    if addfile == "wim" then filter = "启动镜像|*wim|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                                  "文件 当前为:" .. ntboot_runmode, filter)
    elseif addfile == "vhd" then filter = "启动镜像|*vhd;*vhdx|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile ..
                                  "文件 当前为:" .. ntboot_runmode, filter)
    elseif addfile == "raw" then
        local filter = "启动镜像|*wim;*vhd;*vhdx;*efi|所有文件|*.*"
        bootfile = Dialog:OpenFile("选择你要启动的文件 当前为:" .. ntboot_runmode, filter)
        getBootfileExt(bootfile)
        bootsdi(bootfile)
        end
    if bootfile == "" then
        -- winapi.show_message("你犯了个错误!", "你没有选择镜像!")
```

```
        return
    else

        ntboot_ask(bootfile)
    end
end

function helper_ntboot_onload()

    -- addwim()
    if has_option("-ntboot") then
        bootfile = get_option("-ntboot")
        ntboot_args(bootfile)
    end
    if has_option("-now") then
        bootnow = "1"
    else
        bootnow = "0"
    end
end


function check_boot()
    if has_option("-now") or has_option("-r")  then
                bootnow = "1"
        else
                bootnow = "0"
        end
end
        --addwim()



        if has_option("-ntboot") then
        bootfile = get_option("-ntboot")
                check_boot()
        ntboot_args(bootfile)
   elseif has_option("-f")  then
        bootfile = get_option("-f")
        check_boot()
        ntboot_args(bootfile)
        else

        return

        end--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot20240424.lua"
```

```
local ntbootfile_ext = ""
local bootfile = ""
temp = os.getenv("temp")
systemdrive = os.getenv("systemdrive")
APP_Path = os.getenv("WINXSHELL_MODULEPATH")

-- 提取固定值
local exitcode, stdout = winapi.execute("cmd /c mountvol")
stdout = stdout:gsub("\r\n", "\n")
bios_str = stdout:match("EFI ([^\n]+)\n")

--获取系统版本
function get_winver()
        local exitcode, stdout = winapi.execute([[ver]])
        local winver_str = stdout:match("10.0.")
        --stdout = stdout:gsub("\r\n", "\n")
        --winapi.show_message('错误', winver_str)
        if winver_str ~= nil then
                winver = "Win10/Win11"
        else
                winver = "win7"
        end
end

function createBCD(boot_platform)
       if boot_platform == "bios" then

            exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\\Boot]])

            exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] .. boot_drv .. [[\]])
```

```
                exec("/hide /wait", [[bcdedit /createstore ]] .. boot_drv ..
[[\Boot\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. boot_drv .. [[\\Boot\\BCD.tmp
/create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. boot_drv ..
[[\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
boot_drv)
                end
                if boot_platform == "uefi" then
            exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\\EFI\\Microsoft\\Boot]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Boot\\bootx64.efi]])



                exec("/hide /wait", [[bcdedit /createstore ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
boot_drv)
                end


end
```

```
function cs_patition()
        boot_drv = Dialog:BrowseFolder("当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区", 17)
        if boot_platform == "bios" and File.exists(boot_drv .. [[\Boot\BCD]]) then
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\Boot\BCD]])

        elseif boot_platform == "bios" and not File.exists(boot_drv .. [[\Boot\BCD]]) then
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\Boot\BCD]])

        elseif boot_platform == "uefi" and File.exists(boot_drv .. [[\Boot\BCD]]) then
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\EFI\Microsoft\Boot\BCD]])

        elseif boot_platform == "uefi" and not File.exists(boot_drv .. [[\EFI\Microsoft\Boot\BCD]]) then
        createBCD(boot_platform)
                exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\EFI\Microsoft\Boot\BCD]])
end
end

function check_offline_os()
        if exec("/wait", [[bcdedit]]) == 1 then
                cs_patition()
                boot_bcd_var()
        else
                boot_bcd_var()
        end
```

```
end


function boot_bcd_var()
        if boot_platform == "bios" then
                local exitcode, boot_drv_stdout = winapi.execute("cmd /c bcdedit /enum {bootmgr}")
                boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
                boot_drv = boot_drv_stdout:match("device                  partition=([^\n]+)\n")
                bcd_store = boot_drv .. [[\Boot\BCD]]
        elseif boot_platform == "uefi" then
                local exitcode, boot_drv_stdout = winapi.execute("cmd /c bcdedit /enum {default}")
                boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
                boot_drv = boot_drv_stdout:match("device                  partition=([^\n]+)\n")
                local exitcode, bcd_drv_stdout = winapi.execute("cmd /c bcdedit /enum {bootmgr}")
                bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
                bcd_drv = bcd_drv_stdout:match("device                  partition=([^\n]+)\n")
                bcd_store = boot_drv .. [[\EFI\Microsoft\Boot\BCD]]
        end
end

--检测环境
function ck_bios()


        if bios_str == nil then
                bios_str = "传统BIOS"
                boot_platform = "bios"
                wimwinload = [[\Windows\system32\boot\winload.exe]]
                vhdwinload = [[\Windows\system32\winload.exe]]
        else
                bios_str = "UEFI"
                boot_platform = "uefi"
                wimwinload = [[\Windows\system32\boot\winload.efi]]
                vhdwinload = [[\Windows\system32\winload.efi]]
        end


end
ck_bios()
App_platform = sui:find("boot_platform")
App_platform.text = "当前环境:" .. bios_str

function ntboot_init()
        if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:"  then
         exec("/hide", [[cmd /c mkdir "X:\program files\ms" ]])
          exec("/hide", [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])
            get_winver()
                ntboot_addmode = "pe"

                check_offline_os()
                ntboot_runmode = "WinPE环境模式![离线系统盘符->" .. boot_drv .. "] [启动环境->" .. bios_str .. "]"
        else
                get_winver()
                boot_drv = os.getenv("systemdrive")

                ntboot_addmode = "os"
                ntboot_runmode = "正常系统环境![系统盘符->" .. boot_drv .. "] [启动环境->" .. bios_str .. "]"
        end
end

function getBootfileExt(bootfile)
        bootfile_ext = bootfile:match(".+%.(%w+)¥")
        bootfile_ext = string.lower(bootfile_ext) --转小写
        --转大写 bootfile_ext = string.upper(bootfile_ext)
        if bootfile_ext == "wim" then
                addfile = "wim"
                bootok = "1"
        elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
                addfile = "vhd"
                bootok = "1"
        end
end

function ntboot_args(bootfile)
        getBootfileExt(bootfile)
        if bootok ‾= "1" then
                winapi.show_message("错误", "暂未支持启动这种格式")
                return
        end
        if bootnow == "1" then
                bootsdi(bootfile)
                ntbootnow(bootfile)
        else
                --vhd
                bootsdi(bootfile)
                ntbootadd(bootfile)
        end
end
```

```
--分离boot.sdi路径
function bootsdi(bootfile)
        --wim或vhd文件 所在分区相对路径
        boot_file_path = bootfile:match(":([^] %s]+)")
        --wim或vhd文件 所在分区
        boot_file_partition = bootfile:match("([^] %s]+):")
        sdi = [[]] .. boot_drv .. [[\pe\boot.sdi]]
        --boot.sdi 所在分区相对路径
        --alert(sdi)
        boot_sdi_path = sdi:match(":([^] %s]+)")
        --boot.sdi 所在分区
        boot_sdi_partition = sdi:match("([^] %s]+):")
end




function win7_replacebootmgr()
        exec("/show /wait", "cmd /c mountvol B: /s")
        if addfile == "vhd" then
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
        end
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
        exec("/hide /wait", "cmd /c mountvol B: /d")
end


function Add_Boot(addfile)
        if winver == "win7" then
                win7_replacebootmgr()
        end
        if addfile == "wim" then
                ID = "{19260817-6666-8888-f00d-caffee000000}"
                exec("/hide /wait", [[bcdedit /delete ]] .. ID)
                exec("/hide /wait", [[bcdedit /create {ramdiskoptions} /d "ntboot ramdisk"]])
                exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdidevice
partition=]] .. boot_sdi_partition .. [[:]])
                exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdipath ]] ..
boot_sdi_path)
                exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "BootWim With NTBOOT
6.x" /application osloader]])
                exec(
                        "/hide /wait",
                        [[bcdedit -set ]] ..
                        ID .. " device ramdisk=[" .. boot_file_partition .. ":]" ..
boot_file_path .. [[,{ramdiskoptions}]]
                )
                exec(
                        "/hide /wait",
                        [[bcdedit -set ]] ..
                        ID .. " osdevice ramdisk=[" .. boot_file_partition .. ":]" ..
boot_file_path .. [[,{ramdiskoptions}]]
                )
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. wimwinload)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ winpe yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[
/addlast]])
        elseif addfile == "vhd" then
                ID = "{19260817-6666-8888-f00d-caffee000001}"
                exec("/hide /wait", [[bcdedit /delete ]] .. ID)
                exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
                --exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
                exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "Boot from VHD With
NTBOOT 6.x" /application osloader]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " device vhd=[" ..
boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " osdevice vhd=[" ..
boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[
/addlast]])
        end
```

```
end


function ntbootnow(bootfile)
        --创建pe目录
        if addfile == "wim" then
                exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi" ]] ..
boot_drv .. [[\pe]])
        end
        addmode = "/bootsequence"
        say = "添加一次性启动完成，系统将立即重启!"
        Add_Boot(addfile)
        if ntboot_addmode == "pe" then
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
                        local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum
all|find "]] .. boot_file_path .. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                dojob = "wpeutil reboot"
        else
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum all|find "]] ..
boot_file_path .. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
                dojob = "shutdown -r -t 10"
        end

        if bcdedit_r == nil then
                winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
                sui:close()
        else
                exec("/hide", [[cmd /c ]] .. dojob)
                r = winapi.show_message("系统即将重启", "系统即将重启并进入" .. bootfile ..
"\n是否取消? 【如果是PE环境取消无效】",
                        "yes-no", "warning")
        end
        if r == "yes" then
                exec("/hide", [[cmd /c shutdown -a]])
                winapi.show_message("注意", "您虽然取消了立即启动到镜像, \n但重启后还是会启动
一次的, \n再重启就好了, 莫慌!")
                sui:close()
        end
end

function ntboot_ask(bootfile)

    if bios_str == nil then
        ck_bios()

        end
        if boot_drv == nil then
        --boot_drv = os.getenv("systemdrive")
        ntboot_init()



        end
        if winver == nil then
        get_winver()
        end
        bootsdi(bootfile)
        local r =
        winapi.show_message(
                "操作镜像",
                " [是] 立即重启到镜像并只启动一次\n\n [否] 添加到菜单以后启动\n\n [取消] 不操作\n\n 当前启动文件："
                ..
                bootfile ..
                "\n\n 启动环境：" ..
                bios_str ..
                "\n 当前系统为：" ..
                winver ..
                "\n\n boot.sdi所在分区：" ..
                boot_sdi_partition ..
                "\n boot.sdi相对路径：" ..
                boot_sdi_path .. "\n\n 启动镜像盘符：" .. boot_file_partition .. "\n 启动镜像：" .. boot_file_path,
                "yes-no-cancel",
                "warning"
        )
        if r == "yes" then
                ntbootnow(bootfile)
        elseif r == "no" then
                ntbootadd(bootfile)
        elseif r == "cancel" then
                return
                --winapi.show_message("您取消了操作镜像", bootfile)
        end
end

function ntbootadd(bootfile)
        --创建pe目录
        if addfile == "wim" then
                exec("/hide /wait", [[mkdir ]] .. boot_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi" ]] ..
boot_drv .. [[\pe]])
        end
        addmode = "/displayorder"
        say = "添加永久启动项完成!"
        Add_Boot(addfile)
        if ntboot_addmode == "pe" then
                --alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID ..
[["]])
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID
.. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        else
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID
.. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
        end

        --alert([[bcdedit /enum |find "]] .. ID .. [["]])
        if bcdedit_r == nil then
                winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
        else
                winapi.show_message("选择了添加到启动菜单", say)
                sui:close()
        end
end

function Add_Boot_Option(addfile)
        ntboot_init()
        --if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
        if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
        toadmin = winapi.show_message("没有以管理员身份运行!", "请右键管理员身份运行
本程序!,\n是否现在就尝试管理员身份重新运行本程序? "
            , "yes-no", "warning")
    end
    if toadmin == "yes" then
        --exec('/hide', [[cmd /c shutdown -a]])
        --winapi.show_message('注意', "执行管理员提升权限操作")
        exec("/admin", [["]] .. APP_Path .. [[\WinXshell.exe" -ui -jcfg
wxsUI\UI_AppStore\main.jcfg]])
        sui:close()
        return
    elseif toadmin == "no" then
        return
    end
    --winapi.show_message('添加文件类型', addfile)
    local filter = "启动镜像|*vhdx;*.vhdx|所有文件|*.*"
    if addfile == "wim" then
        filter = "启动镜像|*wim|所有文件|*.*"
    end
    bootfile = Dialog:OpenFile("选择你要启动的" .. addfile .. "文件 当前为:" ..
ntboot_runmode, filter)

    if bootfile == "" then
        --winapi.show_message("你犯了个错误!", "你没有选择镜像!")
        return
    else
        ntboot_ask(bootfile)
    end
end

function helper_ntboot_onload()
```

```
        --addwim()
        if has_option("-ntboot") then
                bootfile = get_option("-ntboot")
                ntboot_args(bootfile)
        end
        if has_option("-now") then
                bootnow = "1"
        else
                bootnow = "0"
        end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_ntboot202210242032.lua"
```

```
local ntbootfile_ext = ""
local bootfile = ""
temp = os.getenv("temp")
systemdrive = os.getenv("systemdrive")
APP_Path = os.getenv("WINXSHELL_MODULEPATH")

-- 提取固定值
local exitcode, stdout = winapi.execute("cmd /c mountvol")
stdout = stdout:gsub("\r\n", "\n")
bios_str = stdout:match("EFI ([^\n]+)\n")

--获取系统版本
function get_winver()
        local exitcode, stdout = winapi.execute([[ver]])
        local winver_str = stdout:match("10.0.")
        --stdout = stdout:gsub("\r\n", "\n")
        --winapi.show_message('错误', winver_str)
        if winver_str ~= nil then
                winver = "Win10/Win11"
        else
                winver = "win7"
        end
end

function createBCD(boot_platform)
       if boot_platform == "bios" then

            exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\\Boot]])

            exec("/show /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgr" ]] ..
boot_drv .. [[\]])
```

```
                exec("/hide /wait", [[bcdedit /createstore ]] .. boot_drv ..
[[\Boot\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. boot_drv .. [[\\Boot\\BCD.tmp
/create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. boot_drv ..
[[\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
boot_drv)
                end
                if boot_platform == "uefi" then
            exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\\EFI\\Microsoft\\Boot]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
            exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\\ms\\bootmgfw.efi" ]]
.. boot_drv .. [[\\EFI\\Boot\\bootx64.efi]])



                exec("/hide /wait", [[bcdedit /createstore ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /store ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit.exe /import ]] .. boot_drv ..
[[\EFI\Microsoft\\Boot\\BCD.tmp]])
                exec("/hide /wait", [[bcdedit.exe /set {bootmgr} device partition=]] ..
boot_drv)
                end


end
```

```
function cs_patition()
        boot_drv = Dialog:BrowseFolder("当前为WinPE环境，但无法找到离线系统，请选择离线系统启动分区", 17)
        if boot_platform == "bios" and File.exists(boot_drv .. [[\Boot\BCD]]) then
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\Boot\BCD]])

        elseif boot_platform == "bios" and not File.exists(boot_drv .. [[\Boot\BCD]]) then
        createBCD(boot_platform)
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\Boot\BCD]])

        elseif boot_platform == "uefi" and File.exists(boot_drv .. [[\Boot\BCD]]) then
        exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\EFI\Microsoft\Boot\BCD]])

        elseif boot_platform == "uefi" and not File.exists(boot_drv .. [[\EFI\Microsoft\Boot\BCD]]) then
        createBCD(boot_platform)
                exec("/hide /wait", [[reg load hklm\BCD_]] .. boot_platform .. [[ ]] .. boot_drv .. [[\EFI\Microsoft\Boot\BCD]])
end
end

function check_offline_os()
        if exec("/wait", [[bcdedit]]) == 1 then
                cs_patition()
                boot_bcd_var()
        else
                boot_bcd_var()
        end
```

```
end


function boot_bcd_var()
        if boot_platform == "bios" then
                local exitcode, boot_drv_stdout = winapi.execute("cmd /c bcdedit /enum
{bootmgr}")
                boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
                boot_drv = boot_drv_stdout:match("device                  partition=
([^\n]+)\n")
                bcd_store = boot_drv .. [[\Boot\BCD]]
        elseif boot_platform == "uefi" then
                local exitcode, boot_drv_stdout = winapi.execute("cmd /c bcdedit /enum
{default}")
                boot_drv_stdout = boot_drv_stdout:gsub("\r\n", "\n")
                boot_drv = boot_drv_stdout:match("device                  partition=
([^\n]+)\n")
                local exitcode, bcd_drv_stdout = winapi.execute("cmd /c bcdedit /enum
{bootmgr}")
                bcd_drv_stdout = bcd_drv_stdout:gsub("\r\n", "\n")
                bcd_drv = bcd_drv_stdout:match("device                  partition=
([^\n]+)\n")
                bcd_store = boot_drv .. [[\EFI\Microsoft\Boot\BCD]]
        end
end

--检测环境
function ck_bios()


        if bios_str == nil then
                bios_str = "传统BIOS"
                boot_platform = "bios"
                wimwinload = [[\Windows\system32\boot\winload.exe]]
                vhdwinload = [[\Windows\system32\winload.exe]]
        else
                bios_str = "UEFI"
                boot_platform = "uefi"
                wimwinload = [[\Windows\system32\boot\winload.efi]]
                vhdwinload = [[\Windows\system32\winload.efi]]
        end


end
ck_bios()
App_platform = sui:find("boot_platform")
App_platform.text = "当前环境:" .. bios_str

function ntboot_init()
        if File.exists("X:\\ipxefm\\ipxeboot.txt") or systemdrive == "X:"  then
         exec("/hide", [[cmd /c mkdir "X:\program files\ms" ]])
          exec("/hide", [[cmd /c copy /y X:\ipxefm\app\wimboot\*.* "X:\program files\ms"]])
            get_winver()
                ntboot_addmode = "pe"

                check_offline_os()
                ntboot_runmode = "WinPE环境模式![离线系统盘符->" .. boot_drv .. "] [启动环境->" .. bios_str .. "]"
        else
                get_winver()
                boot_drv = os.getenv("systemdrive")

                ntboot_addmode = "os"
                ntboot_runmode = "正常系统环境![系统盘符->" .. boot_drv .. "] [启动环境->" .. bios_str .. "]"
        end
end


function getBootfileExt(bootfile)
        bootfile_ext = bootfile:match(".+%.(%w+)¥")
        bootfile_ext = string.lower(bootfile_ext) --转小写
        --转大写 bootfile_ext = string.upper(bootfile_ext)
        if bootfile_ext == "wim" then
                addfile = "wim"
                bootok = "1"
        elseif bootfile_ext == "vhd" or bootfile_ext == "vhdx" then
                addfile = "vhd"
                bootok = "1"
        end
end


function ntboot_args(bootfile)
        getBootfileExt(bootfile)
        if bootok ‾= "1" then
                winapi.show_message("错误", "暂未支持启动这种格式")
                return
        end
        if bootnow == "1" then
                bootsdi(bootfile)
                ntbootnow(bootfile)
        else
                --vhd
                bootsdi(bootfile)
                ntbootadd(bootfile)
        end
end
```

```
--分离boot.sdi路径
function bootsdi(bootfile)
        --wim或vhd文件 所在分区相对路径
        boot_file_path = bootfile:match(":([^] %s]+)")
        --wim或vhd文件 所在分区
        boot_file_partition = bootfile:match("([^] %s]+):")
        sdi = [[]] .. boot_drv .. [[\pe\boot.sdi]]
        --boot.sdi 所在分区相对路径
        --alert(sdi)
        boot_sdi_path = sdi:match(":([^] %s]+)")
        --boot.sdi 所在分区
        boot_sdi_partition = sdi:match("([^] %s]+):")
end




function win7_replacebootmgr()
        exec("/show /wait", "cmd /c mountvol B: /s")
        if addfile == "vhd" then
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll" B:\\EFI\\Microsoft\\Boot\\bootvhd.dll]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootvhd.dll" B:\\EFI\\Boot\\bootvhd.dll]])
        end
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgfw.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Microsoft\\Boot\\bootmgr.efi]])
        exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\bootmgfw.efi" B:\\EFI\\Boot\\bootx64.efi]])
        exec("/hide /wait", "cmd /c mountvol B: /d")
end


function Add_Boot(addfile)
        if winver == "win7" then
                win7_replacebootmgr()
        end
        if addfile == "wim" then
                ID = "{19260817-6666-8888-f00d-caffee000000}"
                exec("/hide /wait", [[bcdedit /delete ]] .. ID)
                exec("/hide /wait", [[bcdedit /create {ramdiskoptions} /d "ntboot ramdisk"]])
                exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdidevice partition=]] .. boot_sdi_partition .. [[:]])
                exec("/hide /wait", [[bcdedit /set {ramdiskoptions} ramdisksdipath ]] .. boot_sdi_path)
                exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "BootWim With NTBOOT 6.x" /application osloader]])
                exec(
                        "/hide /wait",
                        [[bcdedit -set ]] ..
                        ID .. " device ramdisk=[" .. boot_file_partition .. ":]" .. boot_file_path .. [[,{ramdiskoptions}]]
                )
                exec(
                        "/hide /wait",
                        [[bcdedit -set ]] ..
                        ID .. " osdevice ramdisk=[" .. boot_file_partition .. ":]" .. boot_file_path .. [[,{ramdiskoptions}]]
                )
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. wimwinload)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ winpe yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[
/addlast]])
        elseif addfile == "vhd" then
                ID = "{19260817-6666-8888-f00d-caffee000001}"
                exec("/hide /wait", [[bcdedit /delete ]] .. ID)
                exec("/hide /wait", [[bcdedit /create {bootmgr} /d "Windows Boot Manager"]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} Locale zh-CN]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} nointegritychecks yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} testsigning yes]])
                exec("/hide /wait", [[bcdedit /set {bootmgr} timeout 5]])
                --exec('/hide', [[bcdedit /set {bootmgr} timeout 30]])
                exec("/hide /wait", [[bcdedit -create ]] .. ID .. [[ /d "Boot from VHD With
NTBOOT 6.x" /application osloader]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " device vhd=[" ..
boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. " osdevice vhd=[" ..
boot_file_partition .. ":]" .. boot_file_path)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ path ]] .. vhdwinload)
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ Locale zh-CN]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ systemroot \Windows]])
                exec("/hide /wait", [[bcdedit -set ]] .. ID .. [[ detecthal yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ winpe yes]])
                --exec('/hide', [[bcdedit -set ]] .. ID .. [[ pae forceenable]])
                exec("/hide /wait", [[cmd /c bcdedit ]] .. addmode .. [[ ]] .. ID .. [[
/addlast]])
        end
```

```
end


function ntbootnow(bootfile)
        --创建pe目录
        if addfile == "wim" then
                exec("/hide /wait", [[cmd /c mkdir ]] .. boot_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi" ]] ..
boot_drv .. [[\pe]])
        end
        addmode = "/bootsequence"
        say = "添加一次性启动完成，系统将立即重启!"
        Add_Boot(addfile)
        if ntboot_addmode == "pe" then
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
                        local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum
all|find "]] .. boot_file_path .. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                dojob = "wpeutil reboot"
        else
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum all|find "]] ..
boot_file_path .. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
                dojob = "shutdown -r -t 10"
        end

        if bcdedit_r == nil then
                winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
                sui:close()
        else
                exec("/hide", [[cmd /c ]] .. dojob)
                r = winapi.show_message("系统即将重启", "系统即将重启并进入" .. bootfile ..
"\n是否取消？【如果是PE环境取消无效】",
                        "yes-no", "warning")
        end
        if r == "yes" then
                exec("/hide", [[cmd /c shutdown -a]])
                winapi.show_message("注意", "您虽然取消了立即启动到镜像，\n但重启后还是会启动
一次的，\n再重启就好了，莫慌!")
                sui:close()
        end
end

function ntboot_ask(bootfile)

    if bios_str == nil then
        ck_bios()

        end
        if boot_drv == nil then
        --boot_drv = os.getenv("systemdrive")
        ntboot_init()



        end
        if winver == nil then
        get_winver()
        end
        bootsdi(bootfile)
        local r =
        winapi.show_message(
                "操作镜像",
                "［是］立即重启到镜像并只启动一次\n\n［否］添加到菜单以后启动\n\n［取消］不操作\n\n 当前启动文件："
                ..
                bootfile ..
                "\n\n 启动环境：" ..
                bios_str ..
                "\n 当前系统为：" ..
                winver ..
                "\n\n boot.sdi所在分区：" ..
                boot_sdi_partition ..
                "\n boot.sdi相对路径：" ..
                boot_sdi_path .. "\n\n 启动镜像盘符：" .. boot_file_partition .. "\n 启动镜像：" .. boot_file_path,
                "yes-no-cancel",
                "warning"
        )
        if r == "yes" then
                ntbootnow(bootfile)
        elseif r == "no" then
                ntbootadd(bootfile)
        elseif r == "cancel" then
                return
                --winapi.show_message("您取消了操作镜像", bootfile)
        end
end

function ntbootadd(bootfile)
        --创建pe目录
        if addfile == "wim" then
                exec("/hide /wait", [[mkdir ]] .. boot_drv .. [[\pe]])
                exec("/hide /wait", [[cmd /c copy /y "]] .. APP_Path .. [[\ms\boot.sdi" ]] ..
boot_drv .. [[\pe]])
        end
        addmode = "/displayorder"
        say = "添加永久启动项完成!"
        Add_Boot(addfile)
        if ntboot_addmode == "pe" then
                --alert([[bcdedit -store ]] .. bcd_store .. [[ /enum |find "]] .. ID ..
[["]])
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID
.. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
                exec("/hide /wait", [[reg unload hklm\BCD_]] .. boot_platform)
        else
                local exitcode, bcdedit_enum = winapi.execute([[bcdedit /enum |find "]] .. ID
.. [["]])
                bcdedit_r = bcdedit_enum:match("{.+}")
        end

        --alert([[bcdedit /enum |find "]] .. ID .. [["]])
        if bcdedit_r == nil then
                winapi.show_message("注意", "添加失败，可能没有权限(需要右键->管理员运行)")
        else
                winapi.show_message("选择了添加到启动菜单", say)
                sui:close()
        end
end

function Add_Boot_Option(addfile)
        ntboot_init()
        --if exec("/wait", APP_Path .. "\\bin\\isadmin.exe") == 0 then
        if exec("/wait", [[REG QUERY "HKU\S-1-5-19"]]) == 1 then
```

```
                toadmin = winapi.show_message("没有以管理员身份运行!", "请右键管理员身份运行本程序!, \n是否现在就尝试管理员身份重新运行本程序？"
                        , "yes-no", "warning")
        end
        if toadmin == "yes" then
                --exec('/hide', [[cmd /c shutdown -a]])
                --winapi.show_message('注意', "执行管理员提升权限操作")
                exec("/admin", [["]] .. APP_Path .. [[\WinXshell.exe" -ui -jcfg wxsUI\UI_AppStore\main.jcfg]])
                sui:close()
                return
        elseif toadmin == "no" then
                return
        end
        --winapi.show_message('添加文件类型', addfile)
        local filter = "启动镜像|*vhdx;*.vhdx|所有文件|*.*"
        if addfile == "wim" then
                filter = "启动镜像|*wim|所有文件|*.*"
        end
        bootfile = Dialog:OpenFile("选择你要启动的" .. addfile .. "文件 当前为:" .. ntboot_runmode, filter)

        if bootfile == "" then
                --winapi.show_message("你犯了个错误!", "你没有选择镜像!")
                return
        else
                ntboot_ask(bootfile)
        end
end

function helper_ntboot_onload()
```

```
        --addwim()
        if has_option("-ntboot") then
                bootfile = get_option("-ntboot")
                ntboot_args(bootfile)
        end
        if has_option("-now") then
                bootnow = "1"
        else
                bootnow = "0"
        end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot202210242032.lua"
```

```
dofile("wxsUI\\UI_AppStore\\types\\media")
dofile("wxsUI\\UI_AppStore\\types\\work")
local file_type = "未知类型"
local file_ext = ""

-- 获取扩展名
function getExtension(str)
        file_ext = str:match(".+%.(%w+)¥")

        for key, value in ipairs(media_list) do
                if file_ext:find("" .. value .. "") then
                        soft_list = "media"
                        file_type = "多媒体类"
                end
        end

        for key, value in ipairs(work_list) do
                if file_ext:find("" .. value .. "") then
                        soft_list = "work"
                        file_type = "文档办公类"
                end
        end

        if soft_list == "media" then
                onclick("¥Nav[3]")
        elseif soft_list == "work" then
                onclick("¥Nav[4]")
        elseif file_ext == "vmx" then
                exec("/hide", [[cmd /c ftype .vmx="%ProgramFiles%\vmware\vmware.exe" %1]])
                soft_list = "vm"
                file_type = "虚拟机"
                onclick("¥Nav[6]")
        elseif file_ext == "wim" then
                exec("/show", [[cmd /k echo %1]])
                soft_list = "wimfile"
                file_type = "镜像"
                onclick("¥Nav[6]")
        else
            openwithsoft = Dialog:OpenFile("未知类型，请选择一个程序打开" .. str .. "文件:", filter)
                exec("/show", openwithsoft .. [[ "]] .. str .. [["]])
                 sui:close()
                --onclick("¥Nav[7]")
                --return str
        end
end

function helper_openfile_onload()
        if has_option("-file") then
                file = get_option("-file")
        end
        if file ~= nil then
                getExtension(file)
        end
end

function helper_openfile_onclick(ctrl)
        -- for _helper_openfile_start
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_openfile.lua"
```

```
temp = os.getenv("temp")
desk = [[X:\Users\Default\DeskTop\]]

exitcode, nwinfo_stdout = winapi.execute("nwinfo --cp=UTF8 --sys --format=lua")
nwinfo_stdout =  nwinfo_stdout:gsub("#!lua", "")
load(nwinfo_stdout)()
sys_cmdline = _NWINFO.System["Start Options"]



--alert(_NWINFO.System["Start Options"])
function dojob(job)
if job == "NETCOPY" then
--alert("本次执行网络同传")
LINK(desk .. [[netcopy网络同传.lnk]], [[netcopyx64.exe]], nil, nil)
 exec("/min",
            [[cmd /c netcopyx64.exe]])
elseif job == "NETGHOST" then
--alert("本次执行GHOST网克")
LINK(desk .. [[ghostx64.lnk]], [[ghostx64.exe]], nil, nil)
LINK(desk .. [[Ghost自动网克.lnk]], [[startup.bat]], [[netghost]], [[ico\ghost32.ico]])
exec("/min",
            [[startup.bat netghost]])
else
alert("不支持的任务")
end
end

function cp_ipxefm_file()

        exec("/hide /wait",
            [[cmd /c copy /y X:\ipxefm\app\inject\default X:\windows]])
        exec("/hide /wait",
            [[sysx64.exe]])

end



function init()
if sys_cmdline:find("@") then
cp_ipxefm_file()
local my_cmdline = string.match(sys_cmdline, ′MININT(.*)RDIMAGEOFFSET′)
my_cmdline = string.gsub(my_cmdline, "^%s*(.-)%s*¥", "%1")  -- 去除字符串两端的空格
local cmdlines = {}
for k, v in string.gmatch(my_cmdline, "([^@=]+)=([^@=]+)") do
    cmdlines[k] = v
end
dojob(cmdlines["JOB"])
else
return
end
end
init()



--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_StartOptions.lua"
```

```
RUNVERY_TIMER_ID = 11111


USERPROFILE = os.getenv("USERPROFILE")
temp = os.getenv("temp")
SystemDrive = os.getenv("SystemDrive")
if File.exists("X:\\ipxefm\\ipxeboot.txt") then
  very_runmode = "pe"
  zexe = [[X:\\Program Files\\7-zip\\7z.exe]]
else
  very_runmode = "system"
  zexe = "%temp%\\AppStore\\bin\\7zx64.exe"
end


function dl_verysync()
  if not File.exists(temp .. [[\very.zip]]) then
    -- winapi.show_message("添加传输目录", "不存在")
    exec(
      "/wait",
      [[aria2c -c --user-agent="Chrome/94.0.4606.71" http://www.verysync.com/download.php?
platform=windows-amd64 -d %temp% -o very.zip]]
    )
  end
  exec("/wait", zexe .. [[ -y x %temp%\very.zip -o%temp%\verysync]])
  local exitcode, veryappdir = winapi.execute([[dir /b %temp%\verysync]])
  veryappdir = string.gsub(veryappdir, "\r\n", "")
  verysync = string.format("%s\\verysync\\%s\\verysync.exe", temp, veryappdir)
  -- verydata = string.format("%s\\verysync\\%s\\data", temp, veryappdir)
  verylog = string.format("%s\\AppData\\Local\\Verysync\\verysync.log", USERPROFILE)
  if very_runmode == "pe" then
    verylog = "X:\\Program Files\\Verysync\\verysync.log"
  else
```

```
    verylog = string.format("%s\\AppData\\Local\\Verysync\\verysync.log", USERPROFILE)
  end
  exec("/hide", [[cmd /c del /q /f ]] .. verylog)
  suilib.call("SetTimer", RUNVERY_TIMER_ID, 8000)
  exec("/show", "cmd /c " .. verysync)
end

function check_ready()
  if File.exists(verylog) then
    local veryfile = io.open(verylog, "r")
    local text = veryfile:read("*a")
    veryok = text:match("My name is")
    veryfile:close()
  end
  if veryok ~= nil then
    suilib.call("KillTimer", 11111)
    local sendpath = Dialog:BrowseFolder("微力同步初始化完毕，请选择要需要传输的目录", 17)
    local exitcode, share_pass = winapi.execute([[cmd /c ]] .. verysync .. [[ folder add ]]
.. sendpath)
    share_pass = string.gsub(share_pass, "\r\n", "")
    veryro = share_pass:match("RO:([^\n]+)\n")
    veryrw = share_pass:match("RW:([^\n]+)\n")

    winapi.show_message("微力密钥", "共享目录为:" .. sendpath .. "\n只读密钥" .. veryro ..
"\n读写密钥" .. veryrw)
  end
end

function get_share()
  if File.exists(verylog) then
    local veryfile = io.open(verylog, "r")
    local text = veryfile:read("*a")
    veryro = text:match("RO:([^\n]+)\n")
    veryrw = text:match("RW:([^\n]+)\n")
    veryfile:close()
    winapi.show_message("微力只读密钥", veryro)
    winapi.show_message("添加传输目录", "目录  " .. sendpath .. "  已经共享")
  end
end

function ontimer(tid)
  if tid == RUNVERY_TIMER_ID then
    check_ready()
  end
end

dl_verysync()
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_verysync.lua"
```

```
RUNVERY_TIMER_ID = 11111

USERPROFILE = os.getenv("USERPROFILE")
temp = os.getenv("temp")
cd = os.getenv("cd")
SystemDrive = os.getenv("SystemDrive")
if File.exists("X:\\ipxefm\\ipxeboot.txt") then
  very_runmode = "pe"
  zexe = [[X:\\Program Files\\7-zip\\7z.exe]]
else
  very_runmode = "system"
  zexe = "%temp%\\AppStore\\bin\\7zx64.exe"
end

function dl_verysync()
  if not File.exists(temp .. [[\very.zip]]) then

    -- winapi.show_message("添加传输目录", "不存在")
    exec(
      "/wait",
      [[cmd /c start /w "下载微力中……" aria2c -c --user-agent="Chrome/94.0.4606.71" http://www.verysync.com/download.php?platform=windows-amd64 -d %temp% -o very.zip]]
    )
  end
  exec("/wait", zexe .. [[ -y x %temp%\very.zip -o%temp%\verysync]])
  local exitcode, veryappdir = winapi.execute([[dir /b %temp%\verysync]])
  veryappdir = string.gsub(veryappdir, "\r\n", "")
  verysync = string.format("%s\\verysync\\%s\\verysync.exe", temp, veryappdir)
  verylog = string.format("%s\\AppData\\Local\\Verysync\\verysync.log", USERPROFILE)
  veryhome = string.format("%s\\verysync\\%s\\verysync", temp, veryappdir)
  if very_runmode == "pe" then
  verylog = "X:\\Program Files\\Verysync\\verysync.log"
   verypath = string.format("%s\\verysync\\%s", temp, veryappdir)
        verypedata = "X:\\Program Files\\Verysync"
    exec("/wait /show", [[cmd /c md "X:\\Program Files\\Verysync"]])
   exec("/wait /show", [[cmd /c mklink /d ]] .. veryhome .. [[ ]] .. [["]] .. verypedata ..
[["]])
   --alert(verypath)
        --verylog = string.format("%s\\verysync\\%s\\data\\verysync.log", temp, veryappdir)
        --verylog = string.format("%s\\verydata\\verysync.log", temp, veryappdir)




  else
    verylog = string.format("%s\\AppData\\Local\\Verysync\\verysync.log", USERPROFILE)
  end
os.putenv("path", verypath)
  exec("/wait /hide", [[cmd /c del /q /f ]] .. verylog)
  exec("/hide", "cmd /c verysync")
  App:SetTimer('微力检测', 8000)
  alert("初始化微力中……")
  --App:Sleep(5088)
  end

function check_ready()

  if File.exists(verylog) then
    local veryfile = io.open(verylog, "r")
    local text = veryfile:read("*a")
    veryok = text:match("My name is")
    veryfile:close()
  end
  App:Sleep(5088)
  if veryok ~= nil then
     App:KillTimer('微力检测')
    local sendpath = Dialog:BrowseFolder("微力同步初始化完毕，请选择要需要传输的目录", 17)
 exec("/show /wait", [[cmd /c verysync folder add ]] .. sendpath)
 exec("/show", [[cmd /k verysync folder list]])

        share_pass = string.gsub(share_pass, "\r\n", "")
         alert(share_pass)
    veryro = share_pass:match("RO:([^\n]+)\n")
    veryrw = share_pass:match("RW:([^\n]+)\n")

    winapi.show_message("微力密钥", "共享目录为:" .. sendpath .. "\n只读密钥" .. veryro ..
"\n读写密钥" .. veryrw)
  end
end

function get_share()
  if File.exists(verylog) then
    local veryfile = io.open(verylog, "r")
    local text = veryfile:read("*a")
    veryro = text:match("RO:([^\n]+)\n")
    veryrw = text:match("RW:([^\n]+)\n")
    veryfile:close()
    winapi.show_message("微力只读密钥", veryro)
    winapi.show_message("添加传输目录", "目录   " .. sendpath .. "   已经共享")
  end
end
```

```
AppTimer['微力检测'] = function(tid)
check_ready()

end


dl_verysync()
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_verysync.lua"
```

WinXShell

- Project
- Documentation

Contents

Manual

- CSIDL
- 更新历史记录

Libraries

- Lua-Engine
- UI-Engine
- Windows-Shell

Program

- WinXShell

Classes

- App
- Shell
- System

更新历史记录

中文 English

中文

WinXShell RC5.1.4 beta1 (2024.08.08)

这是一个主要更新。添加和改善了以下内容:

- [App] 添加 ru-RU 俄语信息资源。
- [App] 增加-regist_only 选项，仅注册程序路径，执行命令时可省略程序路径(与-regist -noaction效果相同)。
- [Daemon] 增加"JS_DAEMON":{"disable_showdesktop": boolean} 选项来关闭显示桌面处理行为。
- [Daemon] 增加连续按下2次 CAPSLOCK 大写字母锁定键响应函数。
- [wxsUI] 修复UI_Calendar组件内存使用量持续增长的问题。
- [Shell] 改善 新增 *FileExpRefresh* 扩展，其他文件管理器也支持自动刷新。
- [Shell] 新增任务栏透明效果相关设置(需要DWM组件支持)。
- [FileExplorer] 改善文件资源管理器在高分屏下的显示效果。
- [FileExplorer] 改善驱动器打开动作。当双击BitLocker加密驱动器时，将自动弹出解锁对话框。
- [Lua] 增加App.Version, Lua.Version 属性，可查看当前程序版本信息。
- [Lua] 增加App:CreateGUID() 方法，可用来生成GUID字符串。
- [Lua] 增加System:NetJoin() 方法，可用来加入工作组或域。
- [Lua] 增加System:EnableEUDC() 方法，开启用户自定义外字支持。
- [Lua] 增加Proc:IsVisable() 方法，可以判断程序窗口是否处于显示或隐藏状态。
- [Lua] 增加Proc:Activate() 方法，可以激活程序窗口。
- [Lua] 增加Disk.IsLocked() 方法，可以判断磁盘分区是否被BitLocker加密。
- [Lua] 增加 *WinXShellC.exe* 控制台程序，执行-code 代码或者-script 脚本文件时，可通过Cmd:Echo() 函数输出内容到控制台进行交互。

```
for /f %i in ('WinXShellC.exe -code Cmd:Echo^(App.Version^)') do set Ver=%i
echo %Ver%
for /f %i in ('WinXShellC.exe -code Cmd:Echo^(Screen:Get^('X'^)^)') do set ScreenX=%i
echo %ScreenX%
```

WinXShell 5.1.2 (2024.02.02)

这是一个主要更新。添加和改善了以下内容:

- [Daemon] 修复Explorer.exe高版本显示桌面按钮崩溃问题。
- [wxsUI] UI_Logon增加用户按钮热键，可通过Alt+A直接登录Admin账户, Alt+S登录SYSTEM账户。
- [wxsUI] 任务栏在屏幕上方时，UI_WIFI将显示到右上角。
- [wxsUI] 修复当启动后再创建PPPoE时，UI_WIFI界面显示高度不正确的问题。
- [Shell] 改善 其他文件管理器也支持自动刷新。
- [Shell] *WinXShell.lua* 中可以定义2个热键函数， WinXShell作为外壳时可自定义WIN+S，WIN+F热键动作。

```
Shell.onHotKey['WIN+S'] = function()
  App:Debug("WIN+S hotkey is pressed.")
  App:Run('everything.exe')
end

Shell.onHotKey['WIN+F'] = function()
  App:Debug("WIN+F hotkey is pressed.")
  Alert('F')
end
```

- [Lua] 增加App:Pause() 命令，作为启动管理器时，可用此命令维持启动进程。
- [Lua] 增加System:CreatePageFile(file, min, max) 方法，可创建页面文件。
- [Lua] 增加System:ReloadCursors() 方法，可刷新鼠标指针式样。

[Lua] 增加Reg:GetSubKeys() 方法，可以获取注册表的子项目集合。 用法参考 LUA_Tests.bat 中的示例。

```
print("GetSubKeys for [HKEY_CLASSES_ROOT\\Folder]:")
local subkeys = Reg:GetSubKeys([[HKEY_CLASSES_ROOT\Folder]])
for i, v in ipairs(subkeys ) do
  print(str.fmt("%d:%s", i, v))
end
```

WinXShell 5.0 (2022.11.11)

这是一个主要更新。添加和改善了以下内容:

- 新增 使用说明文档(WinXShell_Docs)
- 重构 Lua 代码接口， WinXShell.lua 事件响应函数等
- 改进 当作为外壳(Shell)时，以下操作可分别定义打开不同的文件资源管理器。
- 桌面双击打开文件夹
- Windows + E 组合键
- 任务栏快速启动栏中点击文件资源管理器图标
- 改进 UI组件
- UI_WIFI 支持连接 WPA2/WPA3 密码类型
- UI_WIFI 添加显示密码按钮
- ~~UI_WIFI 支持连接同名SSID~~
- UI_WIFI 修复不显示连接界面时，网络状态指示处理未回收内存的问题
- UI_WIFI 添加 Windows 11 风格托盘图标
- UI_WIFI 支持自定义托盘图标
- UI_WIFI 浅色主题文本框显示效果
- UI_Calendar , UI_TrayPanel 修正农历显示不正确的问题，更新农历数据(~2025 年)
- 改进 Lua 接口
- 新增 Sui:onClick() 点击事件支持文本中的URL超链接响应
- 新增 Sui:onHover() 响应鼠标悬停事件
- 新增 Dialog:Show() 弹出对话框
- 新增 Dialog:OpenFile() , Dialog:SaveFile() 弹出打开文件/保存文件窗口
- 新增 Dialog:BrowseFolder() 浏览文件夹窗口
- 新增 File.GetShortPath() 获取 8.3格式的短路径
- 新增 File.GetFullPath() 获取 完整路径格式
- 新增 日志功能(-log 选项)
- 改善 自动识别是否是 Windows PE 环境运行， -winpe 选项 已废弃
- 修复 有时打开菜单时，导致程序无法正常工作的问题
- 更新 适配 Windows 11 新版本系统
- 其他细节更新

本次更新功能增加不多，主要补充了说明文档，对**Lua**接口进行了重新设计，代码重构。

WinXShell 4.6 (2021.11.11)

这是一个主要更新。添加和改善了以下内容:

- 修复 亮度调节功能占用256MB内存的问题。
- 改进 任务栏按钮显示风格。(适配Windows主题，解决Windows 11按钮高亮显示不自然的问题)
- 新增 任务栏程序快速关闭按钮(可通过配置文件进行设定)。
- 新增 UI组件
- UI_LED 屏幕提示信息(可滚动)。

- UI_TrayPanel 显示系统信息，日历，调度调节控件。
- 改进 UI组件
- 当显示设置发生变化时，将调用 ondisplaychanged() 函数，可用来调整窗口位置或更新数据。
- UI_Settings DPI设置新增225%, 250%, 275%, 300%选项。
- UI_SystemInfo 适配Windows 11。
- UI_Calendar 改善界面，新增亮度调节控件(可通过-brightness=true|false参数控制)。
- 改进 lua接口
- 新增 app:info('FirmwareType') 方法
- 新增 app:info('IsUEFIMode') 方法
- 新增 sui:title(str) 方法
- 新增 sui:info('rect') 方法
- 新增 sui:info('wh') 方法
- 新增 FolderOptions:Toggle(opt) 方法
- 其他细节更新

WinXShell 4.5 (2021.04.04)

这是一个主要更新。添加和改善了以下内容:

- 新增 任务栏窗口预览功能(thumbnail)。
- 改进 lua接口

```
可使用Desktop对象，可通过此对象更改壁纸，实时改变桌面图标大小，
布局等表示样式，可直接刷新桌面。
```

- 改进 UI组件
- UI_Settings 支持设置屏幕亮度
- UI_SystemInfo 支持读取OEM信息
- UI_SystemInfo 修复无法正确显示机器名的问题
- UI_SystemInfo 调整界面字体，布局等细节
- UI_WIFI 可直接输入回车键进行连接
- UI_WIFI 【网络和Internet 设置】可打开【网络连接】页面(更改适配器选项)
- UI_WIFI 修复启动窗口闪烁问题
- UI_WIFI 修复在外壳启动前运行时，弹出连接窗口会覆盖任务栏的问题

将支持的Lua函数和对象信息书写LUA_TEST.bat测试脚本(UTF-8编码，中文说明)。

WinXShell 4.4 (2020.10.10)

这是一个常规更新。添加和改善了以下内容:

- 改善 lua_helper扩展

```
将winapi和lua扩展库编译到应用程序本身。
减小程序体积，简化结构，仅应用程序就可支持运行lua代码。
```

- 强化 UI组件
- UI_WIFI 支持连接隐藏网络
- UI_WIFI 支持多无线网卡进行网络连接
- UI_Calendar 支持显示农历信息
- UI_Settings 支持修改显示DPI

WinXShell 4.3 (2020.04.04)

这是一个主要更新。添加和改善了以下内容:

- 改进 编译选项设置 “随机基址”和“数据执行保护(DEP)”。
- 新增 开始菜单项目，新增【网络连接】项目。
- 新增 UI_WIFI 新增 PPPoE拨号按钮，有拨号连接时显示。
- 新增 UI*WIFI*，*UI*Volume支持 -notrayicon 参数，不创建托盘图标。
- 改进 直接接管系统 声音，网络 托盘图标，点击调用 UI*Volume* 和 *UI*WIFI 组件。
- 原生状态指示图标
- 支持图标右键菜单
- 菜单支持全部语言
- 改善 lua_helper 函数库，WinXShell.lua 仅保留用户自定义函数和接口。
- 新增 wxs-ui:xxx, wxs-open:xxx 短命令协议接口。 例如: wxs-ui:wifi, wxs-open:system, wxs-open:networkconnections,

  wxs-open:devices, wxs-open:printers

- 改善 控制面板中点击【连接到网络】，将打开 UI_WIFI 连接界面。
- 改善 ADSL拨号连接，将打开 UI_WIFI 连接界面。(正常系统下有效，PE下无效)
- 修复 我的电脑右键【属性】菜单没有菜单热键问题。
- 修复 点击控制面板无限弹出资源管理器问题。
- 改善 我的电脑右键【属性】默认显示系统属性界面。
- 新增 浅色主题支持。
- 任务栏 添加浅色主题
- 声音,WIFI,日历 添加浅色主题
- 改善 个性化设定界面，增加更多颜色相关设定选项。
- 改善 声音设定界面滑块移动流畅度提高，提示音不再阻塞。
- 修复 信息通知栏无法显示的问题。

English

WinXShell 5.0 (2022.11.11) EN

Not translated yet. Please use the translation tool to view this update instructions.

WinXShell 4.6 (2021.11.11) EN

Not translated yet. Please use the translation tool to view this update instructions.

WinXShell 4.5 (2021.04.04) EN

This is a major update.The following was added and improved:

- Added task thumbnail feature.
- Improved the lua interface.

```
Add the Desktop object, through which you can change the wallpaper, change the size of the
desktop icons, the layout and other presentation styles in real time,
and refresh the desktop directly.
```

- Improved UI components
- UI_Settings supports setting screen brightness
- UI_SystemInfo supports reading OEM information
- UI_SystemInfo fix the issue that the computer name cannot be displayed correctly
- UI_SystemInfo adjust interface font, layout and other details
- UI_WIFI add the enter key to connect
- UI_WIFI [network and Internet settings] link will open [Network connection] control panel window(change adapter options)
- UI_WIFI fix the flickering problem of the startup window
- UI_WIFI fix the problem that the connection window will cover the taskbar when the this is running before the shell starts.

Write the supported Lua functions and objects into the LUA_TEST.bat test script (UTF-8 encoding, in Chinese).

WinXShell 4.4 (2020.10.10) EN

This is a regular update. The following were updated or improved:

- Improved the lua_helper extension.

```
Compile winapi.dll and lua extension helpers into the application itself.
Reduce the size of the program, simplify the structure, and only the application program can
support running Lua code.
```

- Improved UI components.
- UI_WIFI supports connection to hidden network
- UI_WIFI supports multiple wireless network adapters for network connection
- UI_Calendar supports display of lunar calendar for Chinese OS
- UI_Settings supports modification and display of DPI

generated by LDoc 1.4.6

# 更新历史记录

[中文](#中文) [English](#English)

中文
==
==

## WinXShell RC5.1.4 beta5 (2024.11.11)
这是一个主要更新。添加和改善了以下内容:

* [App] 添加 ru-RU 俄语信息资源。
* [App] 增加`-regist_only`选项，仅注册程序路径，执行命令时可省略程序路径(与`-regist -noaction`效果相同)。
* [Daemon] 增加`"JS_DAEMON":{"disable_showdesktop": boolean}`选项来关闭显示桌面处理行为。
* [Daemon] 增加连续按下2次 CAPSLOCK 大写字母锁定键响应函数。
* [wxsUI] 在 Windows 11 环境下，窗口支持圆角显示。
* [wxsUI] 改进UI_SystemInfo组件，支持显示OEMLogo图片。
* [wxsUI] 修复UI_Calendar组件内存使用量持续增长的问题。
* [wxsUI] 修复UI_Settings组件在24H2系统PE下分辨率调整时，任务栏错位问题。
* [Shell] 改善 新增 _FileExpRefresh_ 扩展，其他文件管理器也支持自动刷新。
* [Shell] 新增任务栏透明效果相关设置(需要DWM组件支持)。
* [Shell] 支持使用自定义ico文件来配置开始菜单图标。
* [FileExplorer] 改善文件资源管理器在高分屏下的显示效果。
* [FileExplorer] 改善驱动器打开动作。当双击BitLocker加密驱动器时，将自动弹出解锁对话框。
* [Lua] 增加`App.Version`，`Lua.Version`属性，可查看当前程序版本信息。
* [Lua] 增加`App:CreateGUID()`方法，可用来生成GUID字符串。
* [Lua] 增加`System:NetJoin()`方法，可用来加入工作组或域。
* [Lua] 增加`System:EnableEUDC()`方法，开启用户自定义外字支持。
* [Lua] 增加`Proc:IsVisable()`方法，可以判断程序窗口是否处于显示或隐藏状态。
* [Lua] 增加`Proc:Activate()`方法，可以激活程序窗口。
* [Lua] 增加`Disk.IsLocked()`方法，可以判断磁盘分区是否被BitLocker加密。
* [Lua] 增加 _WinXShellC.exe_ 控制台程序，执行`-code 代码`或者`-script 脚本文件`时，可通过`Cmd:Echo()`函数输出内容到控制台进行交互。

```bat
for /f %i in ('WinXShellC.exe -code Cmd:Echo^(App.Version^)') do set Ver=%i
echo %Ver%
for /f %i in ('WinXShellC.exe -code Cmd:Echo^(Screen:Get^('X'^)^)') do set ScreenX=%i
echo %ScreenX%
```

## WinXShell 5.1.2 (2024.02.02)
这是一个主要更新。添加和改善了以下内容:

* [Daemon] 修复Explorer.exe高版本显示桌面按钮崩溃问题。
* [wxsUI] UI_Logon增加用户按钮热键，可通过Alt+A直接登录Admin账户, Alt+S登录SYSTEM账户。
* [wxsUI] 任务栏在屏幕上方时，UI_WIFI将显示到右上角。
* [wxsUI] 修复当启动后再创建PPPoE时，UI_WIFI界面显示高度不正确的问题。
* [Shell] 改善 其他文件管理器也支持自动刷新。
* [Shell] _WinXShell.lua_ 中可以定义2个热键函数， WinXShell作为外壳时可自定义WIN+S，WIN+F热键动作。

```lua
Shell.onHotKey['WIN+S'] = function()
  App:Debug("WIN+S hotkey is pressed.")
  App:Run('everything.exe')
end
```

```lua
Shell.onHotKey['WIN+F'] = function()
  App:Debug("WIN+F hotkey is pressed.")
  Alert('F')
end
```

* [Lua] 增加`App:Pause()`命令，作为启动管理器时，可用此命令维持启动进程。
* [Lua] 增加`System:CreatePageFile(file, min, max)`方法，可创建页面文件。
* [Lua] 增加`System:ReloadCursors()`方法，可刷新鼠标指针式样。
* [Lua] 增加`Reg:GetSubKeys()`方法，可以获取注册表的子项目集合。
用法参考 LUA_Tests.bat 中的示例。

```lua
print("GetSubKeys for [HKEY_CLASSES_ROOT\\Folder]:")
local subkeys = Reg:GetSubKeys([[HKEY_CLASSES_ROOT\Folder]])
for i, v in ipairs(subkeys ) do
  print(str.fmt("%d:%s", i, v))
end
```

## WinXShell 5.0 (2022.11.11)
这是一个主要更新。添加和改善了以下内容：

* 新增 使用说明文档(`WinXShell_Docs`)
* 重构 Lua 代码接口，`WinXShell.lua` 事件响应函数等
* 改进 当作为外壳(Shell)时，以下操作可分别定义打开不同的文件资源管理器。
  * 桌面双击打开文件夹
  * Windows + E 组合键
  * 任务栏快速启动栏中点击文件资源管理器图标
* 改进 UI组件
  * `UI_WIFI` 支持连接 WPA2/WPA3 密码类型
  * `UI_WIFI` 添加显示密码按钮
  * ~~`UI_WIFI` 支持连接同名SSID~~
  * `UI_WIFI` 修复不显示连接界面时，网络状态指示处理未回收内存的问题
  * `UI_WIFI` 添加 Windows 11 风格托盘图标
  * `UI_WIFI` 支持自定义托盘图标
  * `UI_WIFI` 浅色主题文本框显示效果
  * `UI_Calendar`, `UI_TrayPanel` 修正农历显示不正确的问题，更新农历数据(~2025 年)
* 改进 Lua 接口
  * 新增 `Sui:onClick()` 点击事件支持文本中的URL超链接响应
  * 新增 `Sui:onHover()` 响应鼠标悬停事件
  * 新增 `Dialog:Show()` 弹出对话框
  * 新增 `Dialog:OpenFile()`, `Dialog:SaveFile()` 弹出打开文件/保存文件窗口
  * 新增 `Dialog:BrowseFolder()` 浏览文件夹窗口
  * 新增 `File.GetShortPath()` 获取 8.3格式的短路径
  * 新增 `File.GetFullPath()` 获取 完整路径格式
* 新增 日志功能(`-log` 选项)
* 改善 自动识别是否是 Windows PE 环境运行，`-winpe` 选项 已废弃
* 修复 有时打开菜单时，导致程序无法正常工作的问题
* 更新 适配 Windows 11 新版本系统
* 其他细节更新

本次更新功能增加不多，主要补充了说明文档，对Lua接口进行了重新设计，代码重构。

## WinXShell 4.6 (2021.11.11)
这是一个主要更新。添加和改善了以下内容：

* 修复 亮度调节功能占用256MB内存的问题。
* 改进 任务栏按钮显示风格。(适配Windows主题，解决Windows 11按钮高亮显示不自然的问题)
* 新增 任务栏程序快速关闭按钮(可通过配置文件进行设定)。
* 新增 UI组件
  * UI_LED 屏幕提示信息(可滚动)。
  * UI_TrayPanel 显示系统信息，日历，调度调节控件。
* 改进 UI组件
  * 当显示设置发生变化时，将调用 ondisplaychanged() 函数，可用来调整窗口位置或更新数据。
  * UI_Settings DPI设置新增225%, 250%, 275%, 300%选项。
  * UI_SystemInfo 适配Windows 11。
  * UI_Calendar 改善界面，新增亮度调节控件(可通过-brightness=true|false参数控制)。
* 改进 lua接口
  * 新增 app:info('FirmwareType') 方法
  * 新增 app:info('IsUEFIMode') 方法
  * 新增 sui:title(str) 方法
  * 新增 sui:info('rect') 方法
  * 新增 sui:info('wh') 方法
  * 新增 FolderOptions:Toggle(opt) 方法
* 其他细节更新

## WinXShell 4.5 (2021.04.04)
这是一个主要更新。添加和改善了以下内容:

* 新增 任务栏窗口预览功能(thumbnail)。
* 改进 lua接口
    可使用Desktop对象，可通过此对象更改壁纸，实时改变桌面图标大小，
    布局等表示样式，可直接刷新桌面。
* 改进 UI组件
  * UI_Settings 支持设置屏幕亮度
  * UI_SystemInfo 支持读取OEM信息
  * UI_SystemInfo 修复无法正确显示机器名的问题
  * UI_SystemInfo 调整界面字体，布局等细节
  * UI_WIFI 可直接输入回车键进行连接
  * UI_WIFI 【网络和Internet 设置】可打开【网络连接】页面(更改适配器选项)
  * UI_WIFI 修复启动窗口闪烁问题
  * UI_WIFI 修复在外壳启动前运行时，弹出连接窗口会覆盖任务栏的问题

将支持的Lua函数和对象信息书写LUA_TEST.bat测试脚本(UTF-8编码，中文说明)。

## WinXShell 4.4 (2020.10.10)
这是一个常规更新。添加和改善了以下内容:

* 改善 lua_helper扩展
   将winapi和lua扩展库编译到应用程序本身。
   减小程序体积，简化结构，仅应用程序就可支持运行lua代码。

* 强化 UI组件
  * UI_WIFI 支持连接隐藏网络
  * UI_WIFI 支持多无线网卡进行网络连接
  * UI_Calendar 支持显示农历信息
  * UI_Settings 支持修改显示DPI

## WinXShell 4.3 (2020.04.04)
这是一个主要更新。添加和改善了以下内容:

* 改进 编译选项设置 “随机基址”和“数据执行保护(DEP)”。
* 新增 开始菜单项目，新增【网络连接】项目。
* 新增 UI_WIFI 新增 PPPoE拨号按钮，有拨号连接时显示。
* 新增 UI_WIFI，UI_Volume支持 -notrayicon 参数，不创建托盘图标。
* 改进 直接接管系统 声音，网络 托盘图标，点击调用 UI_Volume 和 UI_WIFI 组件。
  * 原生状态指示图标
  * 支持图标右键菜单
  * 菜单支持全部语言
* 改善 lua_helper 函数库，WinXShell.lua 仅保留用户自定义函数和接口。
* 新增 wxs-ui:xxx, wxs-open:xxx 短命令协议接口。
  例如: wxs-ui:wifi, wxs-open:system, wxs-open:networkconnections,
        wxs-open:devices, wxs-open:printers
* 改善 控制面板中点击【连接到网络】，将打开 UI_WIFI 连接界面。
* 改善 ADSL拨号连接，将打开 UI_WIFI 连接界面。（正常系统下有效，PE下无效）
* 修复 我的电脑右键【属性】菜单没有菜单热键问题。
* 修复 点击控制面板无限弹出资源管理器问题。
* 改善 我的电脑右键【属性】默认显示系统属性界面。
* 新增 浅色主题支持。
  * 任务栏   添加浅色主题
  * 声音,WIFI,日历   添加浅色主题
* 改善 个性化设定界面，增加更多颜色相关设定选项。
* 改善 声音设定界面滑块移动流畅度提高，提示音不再阻塞。
* 修复 信息通知栏无法显示的问题。

____

## English
===

## WinXShell 5.0 (2022.11.11) EN
Not translated yet.
Please use the translation tool to view this update instructions.

## WinXShell 4.6 (2021.11.11) EN
Not translated yet.
Please use the translation tool to view this update instructions.

## WinXShell 4.5 (2021.04.04) EN
This is a major update.The following was added and improved:

* Added task thumbnail feature.
* Improved the lua interface.
    Add the Desktop object, through which you can change the wallpaper, change the size of the desktop icons, the layout and other presentation styles in real time,
    and refresh the desktop directly.
* Improved UI components
  * UI_Settings   supports setting screen brightness
  * UI_SystemInfo supports reading OEM information
  * UI_SystemInfo fix the issue that the computer name cannot be displayed correctly
  * UI_SystemInfo adjust interface font, layout and other details
  * UI_WIFI   add the enter key to connect
  * UI_WIFI   [network and Internet settings] link will open [Network connection] control panel window(change adapter options)
  * UI_WIFI   fix the flickering problem of the startup window
  * UI_WIFI   fix the problem that the connection window will cover the taskbar when the this is

running before the shell starts.

Write the supported Lua functions and objects into the LUA_TEST.bat test script (UTF-8 encoding, in Chinese).

## WinXShell 4.4 (2020.10.10) EN

This is a regular update. The following were updated or improved:

* Improved the lua_helper extension.
    Compile winapi.dll and lua extension helpers into the application itself.
    Reduce the size of the program, simplify the structure, and only the application program can support running Lua code.
* Improved UI components.
  * UI_WIFI   supports connection to hidden network
  * UI_WIFI   supports multiple wireless network adapters for network connection
  * UI_Calendar supports display of lunar calendar for Chinese OS
  * UI_Settings supports modification and display of DPI

WinXShell

- Project
- Documentation

Libraries

- Lua-Engine
- UI-Engine
- Windows-Shell

Program

- WinXShell

Classes

- App
- Shell
- System

Manual

- CSIDL
- 更新历史记录

WinXShell 实用程序

WinXShell 一个轻量的外壳程序，一个启动管理程序，一个现代UI组件引擎。

Libraries

Lua-Engine	Lua 解释程序(Lua Engine)
UI-Engine	UI Engine
Windows-Shell	Windows 外壳程序

Program

WinXShell	WinXShell 实用程序。

Classes

App	App 对象
Shell	Shell 相关对象
System	系统 相关对象

Manual

csidl.md

history.md

generated by LDoc 1.4.6

```
{
 "name":"UI_Settings",
 "title":"Layout",
 "entry":"KB_Layout.xml",
 "lua":"KB_Layout.lua",
 "nobaricon":true
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Settings\KB_Layout.jcfg",
}
```

```
local localeid = nil
function onload()
  localeid = Reg:Read([[HKEY_USERS\.DEFAULT\Control Panel\International]], 'Locale')
  if localeid == nil then return end
  App:Print('LCID' .. localeid)
  localeid = string.sub(localeid, -4, -1)
  --:: Update registry keyboard layouts in HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\WinPE\KeyboardLayouts
  App:Run('Wpeutil.exe', 'ListKeyboardLayouts 0x' .. localeid)
end

function onclick(ctrl)
  App:Run('Wpeutil.exe', string.format('SetKeyboardLayout 0x%s:%s', localeid, ctrl))
  sui:hide()
  App:Sleep(3000) -- ?
  sui:close()
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Settings\KB_Layout.lua"
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="500,360" caption="0,0,0,32" sizebox="6,6,6,6" defaultfontcolor="#FFFFFFFF" >
    <Style imagerectstyle="l,t,w,h" />

    <Style name="layout_btn" align="left" hotimage="color='#FF3D7BBC'" />
    <VerticalLayout bkcolor="#FF2067B2">

        <HorizontalLayout height="40">
            <Label text="Choose your keyboard layout" font="18b" align="center" />
            <Button name="::closebtn" text="   " font="i16" width="32" height="32" padding="0,8,8,0" />
        </HorizontalLayout>

        <HorizontalLayout>
          <VerticalLayout padding="20,0,20,0">
          <button name="0000080c" text="Belgian French (fr-BE)" style="layout_btn" />
          <button name="00000804" text="Chinese PRC (zh-CN)" style="layout_btn" />
          <button name="00000404" text="Chinese Taiwan (zh-TW)" style="layout_btn" />
          <button name="00000413" text="Dutch (nl-NL)" style="layout_btn" />
          <button name="00000809" text="English UK (en-GB)" style="layout_btn" />
          <button name="00000409" text="English US (en-US)" style="layout_btn" />
          <button name="0000040c" text="French (fr-FR)" style="layout_btn" />
          <button name="00000407" text="German (de-DE)" style="layout_btn" />
          <button name="00000408" text="Greek (el-GR)" style="layout_btn" />
          <button name="0000040d" text="Hebrew (he-IL)" style="layout_btn" />
          </VerticalLayout>
          <VerticalLayout padding="20,0,20,0">
          <button name="0000040e" text="Hungarian (hu-HU)" style="layout_btn" />
          <button name="00000410" text="Italian (it-IT)" style="layout_btn" />
          <button name="00000412" text="Korean (ko-KR)" style="layout_btn" />
          <button name="00000816" text="Portuguese (pt-PT)" style="layout_btn" />
          <button name="00000416" text="Portuguese Brazil (pt-BR)" style="layout_btn" />
          <button name="00000419" text="Russian (ru-RU)" style="layout_btn" />
          <button name="0000040a" text="Spanish (es-ES)" style="layout_btn" />
          <button name="0000041d" text="Swedish (sv-SE)" style="layout_btn" />
          <button name="0000041f" text="Turkish (tr-TR)" style="layout_btn" />
          <button text="" style="layout_btn" enabled="false" />
          </VerticalLayout>
        </HorizontalLayout>
    </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="16" size="16" name="Segoe UI" default="true" />
    <Font shared="true" id="16b" size="16" name="Segoe UI" bold="true" />
    <Font shared="true" id="18" size="18" name="Segoe UI" />
    <Font shared="true" id="18b" size="18" name="Segoe UI" bold="true" />
    <Font shared="true" id="20" size="20" name="Segoe UI" />
    <Font shared="true" id="40" size="40b" name="Segoe UI" />
    <Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="일요일" />
    <MultiLanguage id="info_w1" value="월요일" />
    <MultiLanguage id="info_w2" value="화요일" />
    <MultiLanguage id="info_w3" value="수요일" />
    <MultiLanguage id="info_w4" value="목요일" />
    <MultiLanguage id="info_w5" value="금요일" />
    <MultiLanguage id="info_w6" value="토요일" />

    <MultiLanguage id="w7" value="일" />
    <MultiLanguage id="w1" value="월" />
    <MultiLanguage id="w2" value="화" />
    <MultiLanguage id="w3" value="수" />
    <MultiLanguage id="w4" value="목" />
    <MultiLanguage id="w5" value="금" />
    <MultiLanguage id="w6" value="토" />

    <MultiLanguage id="OK" value="확인" />
    <MultiLanguage id="CANCEL" value="취소" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
   <Font id="10" size="10" name="Segoe UI" shared="true" />
   <Font id="16" size="16" name="Segoe UI" shared="true" default="true" />
   <Font id="12" size="12" name="Segoe UI" shared="true" />
   <Font id="24" size="24" name="Segoe UI" shared="true" />
   <Font id="16u" size="16" name="Segoe UI" shared="true" underline="true" />

   <MultiLanguage id="::Connected" value="연결됨" />
   <MultiLanguage id="::Secured" value="보안" />
   <MultiLanguage id="::Open" value="열기" />
   <MultiLanguage id="::Connect" value="연결" />
   <MultiLanguage id="::Disconnect" value="연결 끊기" />
   <MultiLanguage id="msg_hidden_ssid" value="Hidden Network" />
   <MultiLanguage id="msg_enter_ssid" value="Enter the name(SSID) for the network" />
   <MultiLanguage id="msg_enter_password" value="네트워크 보안 키 입력" />
   <MultiLanguage id="Cancel" value="취소" />
   <MultiLanguage id="NetworkSettings" value="네트워크 및 인터넷 설정" />
   <MultiLanguage id="NetworkSettingsMsg" value="정적 IP 주소 설정과 같은 설정을 변경." />
   <MultiLanguage id="Firewall" value="방화벽" />
   <MultiLanguage id="Proxy" value="프록시" />
   <MultiLanguage id="::msg_select_device" value="WLAN 장치 선택" />
   <MultiLanguage id="::msg_service_unavailable" value="WLAN 서비스를 사용할 수 없음" />
   <MultiLanguage id="::msg_device_unavailable" value="WLAN 장치를 사용할 수 없음" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="16" name="Segoe UI"  shared="true"  default="true" />
    <Font id="24" size="24" name="Segoe UI"  shared="true" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="16" name="Segoe UI"  shared="true"  default="true" />
    <Font id="24" size="24" name="Segoe UI"  shared="true" />

    <MultiLanguage id="Settings" value="설정" />
    <MultiLanguage id="Display Settings" value="디스플레이 설정" />
    <MultiLanguage id="Resolution" value="해상도" />
    <MultiLanguage id="Rotate" value="회전" />
    <MultiLanguage id="Rotate_0" value="0 도" />
    <MultiLanguage id="Rotate_90" value="90 도" />
    <MultiLanguage id="Rotate_180" value="180 도" />
    <MultiLanguage id="Rotate_270" value="270 도" />
    <MultiLanguage id="Brightness" value="밝기" />
    <MultiLanguage id="Apply" value="적용(<u>A</u>)" />
    <MultiLanguage id="Close" value="닫기(<u>C</u>)" />
    <MultiLanguage id="Restore" value="복구(<u>R</u>)" />
    <MultiLanguage id="Cancel" value="취소(<u>C</u>)" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
   <Font id="16" size="18" name="Segoe UI"  shared="true"  default="true" />
   <Font id="24" size="24" name="Segoe UI"  shared="true" />
   <Font shared="true" id="40b" size="40b" name="Segoe UI" />
   <Font shared="true" id="128" size="28" name="Segoe MDL2 Assets" />
   <Font shared="true" id="130" size="30" name="Segoe MDL2 Assets" />

   <MultiLanguage id="info_w0" value="일요일" />
   <MultiLanguage id="info_w1" value="월요일" />
   <MultiLanguage id="info_w2" value="화요일" />
   <MultiLanguage id="info_w3" value="수요일" />
   <MultiLanguage id="info_w4" value="목요일" />
   <MultiLanguage id="info_w5" value="금요일" />
   <MultiLanguage id="info_w6" value="토요일" />

   <MultiLanguage id="SignIn" value="서명" />
   <MultiLanguage id="ForgotPassword" value="암호 분실" />
   <MultiLanguage id="Logon" value="로그온" />

   <MultiLanguage id="will autologon with the $user account in $countdown second(s)"
value="$countdown 초 뒤에 $user 계정으로 자동 로그온할 것입니다." />
   <MultiLanguage id="Welcome to Windows ..." value="Windows 시작 ..." />

</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="14" size="14" name="Segoe UI" default="true" />
    <Font shared="true" id="24" size="24" name="Segoe UI" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
    <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="Settings" value="설정" />
    <MultiLanguage id="Home" value="홈" />
    <MultiLanguage id="Customization" value="사용자 지정" />

    <MultiLanguage id="Display" value="디스플레이" />
    <MultiLanguage id="Resolution" value="해상도" />
    <MultiLanguage id="Rotate" value="회전" />
    <MultiLanguage id="Rotate_0" value="0 도" />
    <MultiLanguage id="Rotate_90" value="90 도" />
    <MultiLanguage id="Rotate_180" value="180 도" />
    <MultiLanguage id="Rotate_270" value="270 도" />

    <MultiLanguage id="Brightness" value="밝기" />

    <MultiLanguage id="Apply" value="적용(<u>A</u>)" />
    <MultiLanguage id="Restore" value="복구(<u>R</u>)" />

    <MultiLanguage id="Background" value="배경" />

    <MultiLanguage id="Colors" value="색상" />
    <MultiLanguage id="SelectAppColorTheme" value="기본 앱 모드 선택" />
    <MultiLanguage id="Light" value="밝은 라이트" />
    <MultiLanguage id="Dark" value="어두운 다크" />

    <MultiLanguage id="Taskbar" value="작업 표시줄" />
    <MultiLanguage id="AutoHideTaskbar" value="작업 표시줄 자동으로 숨기기" />
    <MultiLanguage id="UseSmallButtons" value="작은 작업 표시줄 단추 사용" />
    <MultiLanguage id="CombineButtons" value="작업 표시줄 단추 하나로 표시" />
    <MultiLanguage id="Always" value="항상, 레이블 숨기기" />
    <MultiLanguage id="Auto" value="작업표시줄이 꽉 찼을 때" />
    <MultiLanguage id="Never" value="안함" />

    <MultiLanguage id="On" value="켬" />
    <MultiLanguage id="Off" value="끔" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="16" name="Segoe UI"  shared="true"  default="true" />
    <Font id="24" size="24" name="Segoe UI"  shared="true" />

<!-- The description of <Style imagerectstyle="l,t,w,h" /> in main.xml -->
    <!-- image's source, dest attributes will define with "left,top,width,height" style -->
    <!-- the origin style is "left,top,right,bottom". -->
    <!-- the xml file for "Include" tag is also use the same "imagerectstyle" -->

    <MultiLanguage id="Power" value="전원" />
    <MultiLanguage id="CloseComputer" value="PC 끄기" />
    <MultiLanguage id="Shutdown" value="종료" />
    <MultiLanguage id="Restart" value="재시작" />
    <MultiLanguage id="Hibernate" value="최대절전모드" />
    <MultiLanguage id="Standby" value="절전모드" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="16" size="16" name="Segoe UI" default="true" />
    <Font shared="true" id="16b" size="16" name="Segoe UI" bold="true" />
    <Font shared="true" id="18" size="18" name="Segoe UI" />
    <Font shared="true" id="18b" size="18" name="Segoe UI" bold="true" />
    <Font shared="true" id="20" size="20" name="Segoe UI" />
    <Font shared="true" id="40" size="40b" name="Segoe UI" />
    <Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="일요일" />
    <MultiLanguage id="info_w1" value="월요일" />
    <MultiLanguage id="info_w2" value="화요일" />
    <MultiLanguage id="info_w3" value="수요일" />
    <MultiLanguage id="info_w4" value="목요일" />
    <MultiLanguage id="info_w5" value="금요일" />
    <MultiLanguage id="info_w6" value="토요일" />

    <MultiLanguage id="w7" value="일" />
    <MultiLanguage id="w1" value="월" />
    <MultiLanguage id="w2" value="화" />
    <MultiLanguage id="w3" value="수" />
    <MultiLanguage id="w4" value="목" />
    <MultiLanguage id="w5" value="금" />
    <MultiLanguage id="w6" value="토" />

    <MultiLanguage id="OK" value="확인" />
    <MultiLanguage id="CANCEL" value="취소" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
   <Font shared="true" id="12" size="12" name="Segoe UI" default="true" />
   <Font shared="true" id="16" size="16" name="Segoe UI" />
   <Font shared="true" id="24" size="24" name="Segoe UI" />

   <MultiLanguage id="info_w0" value="일요일" />
   <MultiLanguage id="info_w1" value="월요일" />
   <MultiLanguage id="info_w2" value="화요일" />
   <MultiLanguage id="info_w3" value="수요일" />
   <MultiLanguage id="info_w4" value="목요일" />
   <MultiLanguage id="info_w5" value="금요일" />
   <MultiLanguage id="info_w6" value="토요일" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style shared="true" name="ct-bkinfo" bkcolor="#FF2067B2" />
    <Default name="Label" textcolor="#FF000000" />
    <Default name="Button" textcolor="#FF000000" />
    <Style name="ct-bk" bkcolor="#FFE4E4E4" />
    <Style name="ct-txtinfo" textcolor="#FF004275" />
    <Style name="ct-bkhori" bkcolor="#FFB6B6B6" />
    <Style name="ct-today" textcolor="#FFFFFFFF" bkimage="file='themes\today-light.png'" />
</ColorTheme>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Default defaultfontcolor="#FF000000" />
    <Default name="Edit" shared="true" bordersize="0,0,0,1" bordercolor="#FF0078D7"
bkcolor="#FFF4F4F4" nativebkcolor="#FFF4F4F4" />
    <Image shared="true" name="scrollbar.bmp" file="themes\scrollbar-light.bmp" />
    <Image shared="true" name="rc\0.png" file="rc\light\0.png" />
    <Image shared="true" name="rc\25.png" file="rc\light\25.png" />
    <Image shared="true" name="rc\50.png" file="rc\light\50.png" />
    <Image shared="true" name="rc\75.png" file="rc\light\75.png" />
    <Image shared="true" name="rc\100.png" file="rc\light\100.png" />
    <Style name="ct-bkdarker1" bkcolor="#FFE4E4E4" />
    <Style name="ct-bkhori" bkcolor="#FFB6B6B6" />
    <Style name="ct-hotimage" hotimage="color='#FFF4F4F4'" />
    <Style name="ct-text-primary" textcolor="#FF000000" />
    <Style name="ct-text-secondary" textcolor="#FF666666" />
    <Style name="ct-text-secondary-url" textcolor="#FF1874CD" hottextcolor="#FF000000" />
    <Style name="ct-button" bkcolor="#FFCCCCCC" bordersize="2" bordercolor="#FFD3D3D3"
hotbordercolor="#FFADADAD" />
    <Style name="ct-option" bkcolor="#FFF4F4F4" selectedimage="color='#FF0078D7'"
selectedtextcolor="#FFFFFFFF" bordersize="1" bordercolor="#FFE4E4E4" hotbordercolor="#FF929292"
/>
    <Style name="ct-list" itemhotbkcolor="#FFF4F4F4" itemselectedbkcolor="#FFFFFFFF" />
    <Style name="ct-combo" bkimage="file='themes\ComboButton-light.png' corner='2,2,36,2'" />
</ColorTheme>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Style shared="true" name="ct-bkinfo" bkcolor="#FF2067B2" />
    <Default name="Label" textcolor="#FF000000" />
    <Default name="Button" textcolor="#FF000000" />
    <Style name="ct-bk" bkcolor="#FFE4E4E4" />
    <Style name="ct-txtinfo" textcolor="#FF004275" />
    <Style name="ct-bkhori" bkcolor="#FFB6B6B6" />
    <Style name="ct-today" textcolor="#FFFFFFFF" bkimage="file='themes\today-light.png'" />
</ColorTheme>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ColorTheme>
    <Default defaultfontcolor="#FF000000" />
    <Image shared="true" name="snd_mute.ico" file="rc\light\snd_mute.ico" />
    <Image shared="true" name="snd_1.ico" file="rc\light\snd_1.ico" />
    <Image shared="true" name="snd_2.ico" file="rc\light\snd_2.ico" />
    <Image shared="true" name="snd_3.ico" file="rc\light\snd_3.ico" />
    <Style name="ct-bk" bkcolor="#FFE4E4E4" />
    <Image name="thumb.png" file="thumb-light.png" />
</ColorTheme>
```

```
{
 "name":"UI_AppStore",
 "lua":"loading.lua",
 "entry":"loading.xml",
 "nobaricon":true
}
{
 "ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\loading.jcfg",
}
```

```
show_text = "热烈欢迎各位领导莅临指导。"
lshift_index = 0

TIMER_ID_FLUSH_TEXT = 1001
TIMER_ID_QUIT = 1002

function lshift_text()
    local new_text = show_text:sub(lshift_index + 1)
    lshift_index = lshift_index + 2 -- 全汉字，一次移动2个字节
    if lshift_index > show_text:len() then lshift_index = 0 end
    elem_label_info.text = "<b>" .. new_text .. "</b>"
    elem_label_info.bkcolor = "#FEFFFFFF"
end

function onload()
    local text = get_option("-text")
    local wait = get_option("-wait")
    if text then
        if text:sub(1, 1) == '"' then text = text:sub(2, -2) end
        show_text = text
    end
    elem_label_info = sui:find("label_info")
    elem_label_info.text = "<b>" .. show_text .. "</b>"

    -- 按文字内容调整窗体宽度
    local def_w, def_h, w
    def_w = 580 -- 默认宽度
    def_h = 152 -- 默认高度

    if has_option("-top") then
        -- 移动到顶部
        sui:move(0, -((Screen:GetY() - def_h) // 2), 0, 0)
    end

    w = #show_text // 2 * 42 -- 全汉字，一个汉字占2个字节
    if w > Screen:GetX() then w = Screen:GetX() end
    if w > def_w then
        w = w - def_w
        sui:move(-w // 2, 0, w, 0)
    end

    if has_option("-scroll") then
        -- 滚动计时器，文字滚动效果
        suilib.call("SetTimer", TIMER_ID_FLUSH_TEXT, 300)
    end

    if wait then suilib.call("SetTimer", TIMER_ID_QUIT, wait * 1000) end
end

function ontimer(id)
    if id == TIMER_ID_QUIT then
        sui:close()
    elseif id == TIMER_ID_FLUSH_TEXT then
        lshift_text()
    end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\loading.lua"
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="580,152" caption="0,0,0,32" defaultfontcolor="#FFE8EAF6"  layeredimage="bk.png"
layeredopacity="255" >

    <Style imagerectstyle="l,t,w,h" />

    <VerticalLayout bkcolor="#FEFFFFFF">
          <HorizontalLayout height="32" visible="false" >
             <Control />
             <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
             <Button name="::closebtn" text="  " font="24" textcolor="#ff8b8b8b" hottextcolor="#ffffffff"
hotimage="color='#ff007ACC'" width="48" height="32" />
          </HorizontalLayout>
        <VerticalLayout height="120"  inset="1,1,1,1" >
           <Label name="label_info" text="<b>热烈欢迎各位领导莅临指导。</b>" font="24"
showhtml="true" textcolor="#FFF8D714" textpadding="40,20,0,0" />
        </VerticalLayout>

    </VerticalLayout>
</Window>
```

```
proc = winapi.spawn_process([[X:\Program Files\PENetwork\PENetwork.exe]])
a = proc:get_pid()
winapi.show_message("ČĄľĂpid", a)--
"Ô- Âˇžś":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\Lua_Getpid_Demo.lua"
```

```
--[=[ 2>nul
rem -- ===================== batch script =====================
@echo off
set "f0=%~f0"&&set "dp0=%~dp0"

cd /d "%~dp0"

set WINXSHELL=WinXShell.exe
set x8664=x64
if not "x%PROCESSOR_ARCHITECTURE%"=="xAMD64" set x8664=x86
if not exist %WINXSHELL% set WINXSHELL=WinXShell_%x8664%.exe

rem one line code
%WINXSHELL% -code "winapi.show_message('title', 'message')"
pause
rem %WINXSHELL% -code "Taskbar:Pin('cmd.exe', nil, '/k echo lua_test', 'shell32.dll', 27)"
rem pause

rem code file
start /wait %WINXSHELL% -script "%f0%"
pause
goto :EOF
]=]


--- -- =====================  lua script  =====================
-- comment, started with --

local dp0 = os.getenv('dp0')
winapi.show_message('CD', dp0)
winapi.show_message('#{shell32.dll,21769}', app:call('resstr', '#{@shell32.dll,21769}'))

winapi.show_message('app', app:info('winver'))

winapi.show_message('Screen', string.format('%dx%d, DPI:%d, brightness:%d',
    Screen:GetX(), Screen:GetY(), Screen:GetDPI(), Screen:Get('brightness')))

-- Screen:Set('brightness', 60)

winapi.show_message('Desktop', Desktop:GetWallpaper())
Desktop:ShowIcons(0)
app:sleep(2000)
Desktop:ShowIcons(1)
app:sleep(2000)
Desktop:SetIconSize('Large') -- 'L'
app:sleep(2000)
Desktop:SetIconSize('M')
-- app:sleep(2000)
-- Desktop:SetIconSize('S')

exec('/min', 'notepad.exe')

-- More Information

--[==[
WinXShell 公开参数一览:

wxs-open:system | 打开我的电脑属性
wxs-open:printers | 打开打印机设置窗口
wxs-open:networkconnections | 打开网络连接设置窗口
```

wxs-open:devicesandprinters | 打开设备和打印机设置窗口
wxs-open:wifi | 打开无线网络连接界面
wxs-open:volume | 打开音量控制界面

ms-settings:taskbar | 打开任务栏设置界面
ms-settings:dateandtime | 打开时间和日期设置界面ms-settings:display | 打开显示设置界面
ms-settings:personalization | 打开个性化设置界面
ms-settings:network | 打开网络连接设置界面(高版本Windows有效)
-console | 调试窗口
-ui -jcfg XXXXX\YYYY\ZZZ.jcfg | 启动UI组件
-winpe | Windows PE 环境标识
-code luacode | 执行单行lua代码
-file luafile | 执行lua脚本文件

WinXShell 公开对象一览(-code, -script 时使用):

app 程序对象。
app:info(opt) | 获取程序信息
app:call(...) | 调用WinXShell内置函数
app:jcfg(opt) | 获取jcfg配置文件设定信息
app:print(str) | 打印信息到调试窗口

对象方法参数说明:
app:info(opt) | 获取程序信息
opt可以指定以下值。
'cmdline' | 命令行参数(字符串)
'winver' | Windows版本号(字符串)
'langid | 系统语言编号(字符串)
'locale' | 系统语言名称(字符串)
'path' | 程序模块路径(字符串)
'FirmwareType' | 启动模式[BIOS或UEFI](字符串)
'IsUEFIMode' | 是否是UEFI启动模式(布尔值)

对象方法参数说明:
app:call(...) | 调用WinXShell内置函数
... 可指定函数及参数如下。
'getresolutionlist' | 获取可用分辨率列表(字符串)
'cd', path | 切换当前工作目录到指定位置
'desktop::updatewallpaper' | 刷新系统壁纸
'desktop::getwallpaper' | 获取系统壁纸文件路径
'desktop::setwallpaper', file | 更改系统壁纸
'volume::mute', 0or1 | 禁音/解除禁音
'volume::ismuted' | 获取禁音状态
'volume::getlevel' | 获取当前音量大小
'volume::setlevel', 0to100 | 更改当前音量大小
'volume::getname' | 获取当前音频设备名称
'gettickcount' | 获取系统启动时间(秒单位)
'sleep', imsec | 等待imsec毫秒
'play', wavfile | 播放WAV格式音频文件
'putenv', var, val | 设置环境变量
'envstr', str | 展开字符串中的环境变量(字符串)
'resstr', resstr | 展开字符串中的系统字符串资源
'run', file | 启动程序或者文件
'exitcode', code | 指定程序的退出码
'exit', code | 直接退出程序

System 系统对象。
System:GetSettings(key) | 获取系统设定
System:SetSettings(key, val) | 更改系统设定
System:SysColorTheme(mode) | (mode = 'light' | 'dark')

更改系统颜色主题
System:AppsColorTheme(mode) | 更改应用颜色主题(mode = 'light' | 'dark')
System:Reboot() | 重启计算机
System:Shutdown() | 关闭计算机

对象方法参数说明:
System:GetSettings(key) | 获取系统设定
key可以指定以下值。
'AppsColorTheme' | 应用程序颜色主题
'SysColorTheme' | 系统颜色主题
'ShellColorPrevalence' | 外壳外观颜色
'WindowColorPrevalence' | 系统外观颜色
'Colors.Transparency' | 使用透明色

System:SetSettings(key, val) | 更改系统设定
key可以指定以下值。
'ShellColorPrevalence' | 外壳外观颜色
'WindowColorPrevalence' | 系统外观颜色
'Colors.Transparency' | 使用透明色

Screen 屏幕对象。
Screen:Get(...) | 获取屏幕信息
Screen:Set(...) | 设置屏幕参数
Screen:GetX() | 获取屏幕分辨率宽度
Screen:GetY() | 获取屏幕分辨率高度
Screen:GetRotation() | 获取屏幕旋转状态
Screen:GetDPI() | 获取屏幕DPI
Screen:Disp([w, h]) | 设置屏幕分辨率。不指定宽度和高度时，自动使用最大分辨率。
Screen:DispTest(arr) | 按照指定的分辨率列表逐个尝试设置屏幕分辨率，直到设置成功为止。

对象方法参数说明:
Screen:Get(...) | 获取屏幕信息
Screen:Set(...) | 设置屏幕参数
参数可以指定以下值。
'dpi' | 屏幕DPI
'brightness' | 屏幕亮度

Desktop 桌面对象。
Desktop:Refresh() | 刷新桌面
Desktop:GetWallpaper() | 获取壁纸路径
Desktop:SetWallpaper(wallpaper) | 设置壁纸路径
Desktop:SetIconSize(size) | 设置桌面图标大小('S', 'M', 'L')
Desktop:AutoArrange(checked) | 设置 自动排列图标 选项 (0 | 1)
Desktop:SnapToGrid(checked) | 设置 将图标与网格对齐 选项 (0 | 1)
Desktop:ShowIcons(checked) | 设置 显示桌面图标 选项 (0 | 1)

Taskbar 任务栏对象。
Taskbar:IsReady(sec) | 指定sec秒内判断任务栏是否准备就绪(sec为-1时，直到任务栏准备好为止)
Taskbar:WaitForReady() | 等待任务栏加载完毕(内部调用Taskbar:IsReady(-1)方法)
Taskbar:GetSetting(key) | 获取任务栏设定
Taskbar:SetSetting(key, value, type) | 更改任务栏设定
Taskbar:CombineButtons(value[, update]) | 更改任务栏合并按钮选项(value ='always' | 'auto'),
update为0时，不立即生效
Taskbar:UseSmallIcons(value[,update]) | 更改任务栏使用小图标选项(value ='always' | 'auto'),
update为0时，不立即生效
Taskbar:AutoHide(value) | 更改任务栏自动隐藏选项(value = 0 | 1)。
Taskbar:Pin(target[, name[, param[, icon[, index[, showcmd]]]]]) |

```
                                                                    固定命令到任务栏

--------------------------------------------------------
Startmenu 开始菜单对象。
Startmenu:Pin(target[, name[, param[, icon[, index[, showcmd]]]]]) | 固定命令到开始菜单

--------------------------------------------------------
FolderOptions 文件夹选项对象。
FolderOptions:Get(opt) | 获取文件夹选项。
FolderOptions:Set(opt, val) | 更改文件夹选项。
FolderOptions:Toggle(opt) | 切换文件夹选项。

对象方法参数说明:
opt可以指定以下值。
'ShowAll'          | 显示隐藏文件
'ShowExt'          | 显示隐藏已知文件类型的扩展名
'ShowSuperHidden'  | 显示系统隐藏文件

WinXShell 可使用的库:

Lua5.4内置库: os, math, io等
https://www.lua.org/manual/5.4/

winapi 扩展库
https://github.com/stevedonovan/winapi

WinXShell 公开函数一览(-code, -script 时使用):

MinimizeWindow(class, title) | 隐藏指定窗口。
HideWindow(class, title) | 隐藏指定窗口。
CloseWindow(class, title) | 关闭指定窗口。
CloseShellWindow() | 关闭外壳窗口(外壳窗口不存在时，将等待外壳窗口出现后再关闭)。
os_ver_info() | 获取 ProductName, CSDVersion 信息(注册表)。
cpu_info() | 获取 CPU 信息(注册表)。
mem_info() | 获取系统可用内存(硬件以外)，总内存大小，当前可用内存。
res_str(file, id) | 获取指定字符串资源。
win_copyright() | 获取 Windows 许可证说明。
call_dll(...) | 调用 dll 中的公开函数。
reg_read(key, values) | 读取注册表信息。
reg_write(key, name, value[, type]) | 写入注册表信息。未指定type时, 当作 REG_SZ 类型。
exec(option, cmd) | 执行命令。 option 可指定 /wait - 等待执行命令结束, /hide - 隐藏，/min - 最小化。
link(lnk, target, param, icon, index, showcmd) | 创建快捷方式。
wxsUI(ui[, jcfg[, opt[, app_path]]]) | 打开指定的UI组件。
PinToStartMenu(target[, name[, param[, icon[, index[, showcmd]]]]]) | 别名。固定命令到开始菜单。

]==]
```

```
--[=[ 2>nul

set WINXSHELL_LUASCRIPT=WinXShellTest.lua
set WINXSHELL_LOGFILE=C:\Windows\Temp\WinXShell.log

set WINXSHELL=WinXShell.exe
set x8664=x64
if not "x%PROCESSOR_ARCHITECTURE%"=="xAMD64" set x8664=x86
if not exist %WINXSHELL% set WINXSHELL=WinXShell_%x8664%.exe
if not exist %WINXSHELL%  set WINXSHELL=x64\Debug\WinXShell.exe

rem one line code
%WINXSHELL% -code "winapi.show_message('title', 'message')"

rem code file
:LOOP
%WINXSHELL% -console -script "%~dpnx0"
type "%WINXSHELL_LOGFILE%"
pause
goto :LOOP

goto :EOF
]=]

--- -- ====================  lua script  ====================

-- Alias


MsgBox = winapi.show_message
```

```
-- Downward Compatibility

print(app)

-- Tests

local Tester = {}
local r
local v

-- Functions

-- run()
-- exec()
-- link()

-- New for v5.0.0

--[[
Alert('abc', 123, App:Version())
r = MsgBox('title', 'message', 'yes-no', 'warn')
print('You clicked the [' .. r .. '] button.')
]]

function Tester.BuiltinLib()
-- os lib
local v = os.info('WinVer')
local mem = os.info('Mem')

p(
"======================os.info======================",
"\n",
′v.ver=′ .. v.ver, "v[3]=" .. v[3], "v[4]="  .. v[4], "v[′1.2′]="  .. v["1.2"],
"\n",
os.info(′Copyright′),
"\n",
os.info(′CPU′)[′name′], os.info(′CPU′)[′¯MHz′],
"\n",
 mem[1], mem[2], mem[3],
"\n",
mem[′total′], mem[′used′], mem[′avail′], mem[′used%′],
"\n",
string.format("Total: %d GB, Used: %.2f GB, Avail: %.2f GB, Used: %.1f %%",
    mem[′total_gb′], mem[′used_gb′], mem[′avail_gb′], mem[′used%′]),
"\n",
os.info(′Tickcount′), os.info(′LangId′), os.info(′Locale′),
os.info(′FirmwareType′), os.info(′IsUEFIMode′),
os.info(′IsWinPE′),
"\n",
math.band(4, 5),
"\n"
)

p(
"====================os.exists====================",
"\nos.exists([[%HOMEDRIVE%\\Windows\\]])",
os.exists([[%HOMEDRIVE%\Windows\]]),
"\nos.exists([[%HOMEDRIVE%\\Windows\\Explorer.exe]])",
os.exists([[%HOMEDRIVE%\Windows\Explorer.exe]]),
"\nos.exists([[C:\\WindowsPE\\]])",
os.exists([[C:\WindowsPE\]]),
"\nos.exists([[C:\\App.exe]])",
os.exists([[C:\App.exe]])
)

p("=====================string/math extends=====================\n")

local str1 = [[%SystemRoot%\System32\regedit.exe]]
local str2 = '#{@shell32.dll,9316}'

p(
"string.envstr:",
string.envstr([[%ProgramFiles%\WinXShell\WinXShell.exe]]),
"\nstr1:envstr()",
str1:envstr(),
"\nstring.resstr:",
string.resstr('#{@shell32.dll,9315}'),
"\nstr2:resstr()",
str2:resstr(),
"\nmath.band(4, 5):",
math.band(4, 5),
"\n"
)

end



-- App Object

function Tester.App()

p(
"=====================App Object=====================",
"\nCmdLine:", App.CmdLine,
"\nPath:", App.Path,
"\nName:", App.Name,
"\nFullPath:", App.FullPath,
"\n",
"\nApp:HasOption(\'-console\')", App:HasOption('-console'),
"\nApp:HasOption(\'-debug\')", App:HasOption('-debug'),
"\nApp:GetOption(\'-script\')", App:GetOption('-script'),
"\nApp:GetOption(\'-theme\')", App:GetOption('-theme'),
"\n"
)
App:Sleep(50)
App:Error("[ERROR] Failed to do something.")
App:Warn("CAUTION !!!")
App:InfoLog("notice message")
App:Debug("message in one line")
App:Debug("one", "two")

end

--- Reg Object

function Tester.Reg()

p("\n\n\n")
p(
"=====================Reg Object====================="
)
local regkey=[[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion]]
v = Reg:Read(regkey, {'CurrentBuild', 'UBR'})
```

```
---> v[1] = v['CurrentBuild'] = '19044'
---> v[2] = v['UBR'] = 1645
print(
  str.fmt("v[1] = %s, v[2] = %d", v[1], v[2])
)
print(
  str.fmt("v['CurrentBuild'] = %s, v['UBR'] = %d", v['CurrentBuild'], v['UBR'])
)

regkey = [[HKEY_CURRENT_USER\SOFTWARE\WinXShell\Tests]]

Reg:Write(regkey, '', 'DefaultItemValue')
Reg:Write(regkey, 'Test', 'Test String')
Reg:Write(regkey, 'number', 123, REG_DWORD)

Reg:Write(regkey .. '\\SubKey', '', 'Default')

print(Reg:Read(regkey, ''))
print(Reg:Read(regkey, 'Test'))

-- Reg:Delete([[HKEY_CURRENT_USER\SOFTWARE\WinXShell\Tests]], 'Test')
-- Reg:Delete([[HKEY_CURRENT_USER\SOFTWARE\WinXShell\Tests]])

print("GetSubKeys for [HKEY_CLASSES_ROOT\\Folder]:")
local subkeys = Reg:GetSubKeys([[HKEY_CLASSES_ROOT\Folder]])
for i, v in ipairs(subkeys ) do
  print(str.fmt("%d:%s", i, v))
end

-- Downward Compatibility
```

```
app:call('sleep', 50)

regkey=[[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion]]
v = reg_read(regkey, {'CurrentBuild', 'UBR'})
print(v['UBR'])

end




-- Proc/Window lib
function Tester.Proc()

p(
"======================Proc/Window lib======================"
)
App:Run("notepad.exe")

App:Sleep(500)
local proc = Window.Find("无标题 - 记事本")

p(
  proc:GetClassName(),
  proc:GetFileName(),
  proc:GetHandle(),
  str.fmt("Handle = 0x%x", proc:GetHandle())
)
p(proc)
proc:Close()
```

```
App:Sleep(500)
obj = Window.Find("无标题 - 记事本")
p(obj, str.fmt("Handle = 0x%x", obj:GetHandle()))
p("Proc End")

end

Tester.BuiltinLib()
Tester.App()
Tester.Reg()
Tester.Proc()
```

```
--[=[ 2>nul

set WINXSHELL_LUASCRIPT=WinXShellTest.lua
set WINXSHELL_LOGFILE=C:\Windows\Temp\WinXShell.log

set WINXSHELL=WinXShell.exe
set x8664=x64
if not "x%PROCESSOR_ARCHITECTURE%"=="xAMD64" set x8664=x86
if not exist %WINXSHELL% set WINXSHELL=WinXShell_%x8664%.exe
if not exist %WINXSHELL%  set WINXSHELL=x64\Debug\WinXShell.exe


%WINXSHELL% -console -script "%~dpnx0"
type "%WINXSHELL_LOGFILE%"

goto :EOF
]=]

--- -- ====================  lua script  ====================



Alert('abc', 123, App:Version())



App:Sleep(50)
App:Error("[ERROR] Failed to do something.")
App:Warn("CAUTION !!!")
App:InfoLog("notice message")
App:Debug("message in one line")
App:Debug("one", "two")
```

```
App:SetTimer('测试App计时器', 3000)


AppTimer['测试App计时器'] = function(tid)
    Alert('时间到')
end


Alert('OK to Finish')
```

WinXShell

- Project
- Documentation

Contents

Libraries

- **Lua-Engine**
- UI-Engine
- Windows-Shell

Program

- WinXShell

Classes

- App
- Shell
- System

Manual

- CSIDL
- 更新历史记录

Module `Lua-Engine`

Lua 解释程序(Lua Engine)

用于 `WinXShell.lua`, `-code`, `-script` 选项时使用的代码，函数。

Lua 与 Lua 库

Overview 概览

os

扩展 库

os.putenv (var, value)	设置 环境变量
os.info (key)	获取 系统信息
os.exec ([opt], cmd)	执行外部命令
os.link (lnk, target[, param[, icon[, index[, showcmd]]]])	创建 快捷方式
os.rundll (dllfile, funcname[, param1[, param2[, ...]]])	调用 dll 中的公开函数

扩展 string 库

string.envstr (str)	展开 环境变量
string.resstr (str)	展开 字符串资源

扩展 math 库

math.band (a, b)	进行 位与运算

扩展 lua 函数

exec ()	执行外部命令
link ()	创建 快捷方式
wxsUI (ui[, jcfg[, param]])	打开 指定的UI组件

扩展 lua 对象

App	App
Shell	Shell
System	System

Deprecated 已弃用

win_copyright ()	已弃用。获取 Windows 许可证说明
cpu_info ()	已弃用。获取 CPU 信息(注册表)
mem_info ()	已弃用。获取 内存信息
res_str (file, id)	已弃用。获取 资源文件中的字符串信息
call_dll ()	已弃用。调用 dll 中的公开函数
CloseShellWindow ()	已弃用。关闭 外壳窗口
MinimizeWindow (class, title)	已弃用。最小化 指定窗口
HideWindow (class, title)	已弃用。隐藏 指定窗口
CloseWindow (class, title)	已弃用。关闭 指定窗口

Lua 与 Lua 库

Overview

概览

WinXShell 将Lua解释器嵌入程序中，支持调用Lua代码，还提供了实用的扩展Lua对象，功能函数。

Lua语言自带的功能，模块可直接使用, 比如字符串操作，数学函数，系统功能函数等。Lua标准库

详细信息请访问 Lua官方网站 https://www.lua.org/ 查看。

另外还引入了 winapi 库，可以使用很多实用的面向Windows系统的功能。

简单使用Lua指令时，请了解以下**Lua**基本语法:

- 函数说明中的类型主要有 `int` 整数类型, `bool` 布尔型(true - 真, false - 假), string 字符串类型。
- 字符串使用一对单引号(`'`)或者双引号(`"`)包起来。特殊的字符需要使用转义字符代替。比如:
  - `\t` => 制表符， `\\` => `\`, `\n` => 换行符
- 注册表或者文件路径存在很多反斜杠(`\`)， 使用双中括号抱起来，可直接使用`[[path\to\file]]`。
- 使用双点符号(`..`)可以将字符串进行拼接(也可以使用string.format函数)。

还可通过编辑 `WinXShell.lua` 加载自定义扩展模块，功能函数。

扩展 os 库

os.putenv (var, value)

设置 环境变量

Parameters:

- var string 变量名
- value string 变量值

os.info (key)

获取 系统信息

Parameters:

- key string

  信息名

  - `Version` - 获取 系统版本信息
  - 从注册表获取 ProductName, CSDVersion 信息
  - `WinVer` - 获取 系统版本信息
    - 返回 包含多个系统版本数组
    - `['ver']` - 包含4级别的完整版本信息
    - `[1]` - 1级版本号 (整数)
    - `[2]` - 2级版本号 (整数)
    - `[3]` - 3级版本号 (整数)
    - `[4]` - 4级版本号 (整数)
    - `['1.2']` - 主次版本号 (字符串)
  - `CPU` - 处理器信息
    - 返回 包含多个处理器信息的数组
    - `['name']` - 名称
    - `['~MHz']` - 速度 (整数, 单位 MHz)
  - `Mem` - 内存信息
    - 返回 包含多个内存信息的数组
    - 系统可用内存(硬件以外); 总内存大小; 当前可用内存

        - `['total']` - 总内存大小 (整数, 单位 字节)
        - `['used']` - 当前已使用内存 (整数, 单位 字节)
        - `['avail']` - 当前可用内存 (整数, 单位 字节)
        - `['total_gb']` - 总内存大小 (整数, 单位 GB)
        - `['used_gb']` - 当前已使用内存 (浮点数, 单位 GB)
        - `['avail_gb']` - 当前可用内存 (浮点数, 单位 GB)
        - `['used%']` - 当前已使用内存 (浮点数, 百分比)
    - `Copyright` - 版权信息
    - `LangId` - 系统语言识别号
    - `LocaleName` - 系统语言名
    - `TickCount` - 系统启动后经历的时间 (毫秒)
    - `FirmwareType` - 启动模式
    - `IsUEFIMode` - 是否是 UEFI 启动模式
    - `IsWinPE` - 是否是在 Windows PE 环境

Usage:

```
-- WinXShell.exe -script info.lua
-- info.lua

local v = os.info('WinVer')
v.ver ---> '10.0.19044.1586'
v['ver'] ---> '10.0.19044.1586'
v[1] ---> 10
v[2] ---> 0
v[3] ---> 19044
v[4] ---> 1586
v["1.2"]  ---> '10.0'


local cpu_info = os.info('CPU')
cpu_info['name']
---> 'Intel(R) Core(TM) i5-6600K CPU @ 3.50GHz'
cpu_info['~MHz']
---> 3504


local mem = os.info('Mem')
print(mem['total'], mem['used'], mem['avail'], mem['used%'])
---> 17129684992     3752480768      13377204224     22.0

print(string.format("Total: %d GB, Used: %.2f GB, Avail: %.2f GB, Used: %.1f %%",
    mem['total_gb'], mem['used_gb'], mem['avail_gb'], mem['used%']))

---> Total: 16 GB, Used: 3.49 GB, Avail: 12.46 GB, Used: 22.0 %

os.info('Copyright')
---> (c) Microsoft Corporation。保留所有权利。

os.info('LangId')
---> '2052'

os.info('Locale')
---> 'zh-CN'

os.info('FirmwareType')
---> 'UEFI'

os.info('IsUEFIMode')
---> 1

os.info('IsWinPE')
---> 0
```

os.exec ([opt], cmd)

执行外部命令

Parameters:

- opt string

  启动选项 可以是以下启动选项中的一个或多个的组合:

    - `/hide` 隐藏窗口启动
    - `/wait` 等待程序执行结束
    - `/min` 最小化窗口启动
    - `/max` 最大化窗口启动
    - `/admin` 要求管理员权限启动

  (*optional*)
- cmd string 命令行(含参数)

Usage:

```
exec('cmd.exe /k echo hello')
exec('/hide /wait', 'PETools.cmd PETools.7z')
```

os.link (lnk, target[, param[, icon[, index[, showcmd]]]])

创建 快捷方式

Parameters:

- lnk string 指定要创建的快捷方式的路径
- target string 快捷方式的目标路径
- param string 程序执行参数 (*optional*)
- icon string 指定任务栏图标(ico文件，或者包含图标的资源文件) (*optional*)
- index int 图标索引号 (*optional*)
- showcmd string 启动参数 (*optional*)

Usage:

```
os.link([[D:\Tools\Everything.lnk]], [[%ProgramFile%\Everything\Everything.exe]])
os.link(([[D:\Tools\Hello.lnk]], 'cmd.exe', '/k echo hello', 'shell32.dll', 27)
```

os.rundll (dllfile, funcname[, param1[, param2[, ...]]])

调用 dll 中的公开函数

Parameters:

- dllfile string dll文件路径
- funcname string 函数名
- param1 函数参数 (*optional*)
- param2 函数参数 (*optional*)
- ... 函数参数 (*optional*)

Usage:

```
-- SHEmptyRecycleBinA - 清空指定驱动器上的回收站。

-- SHEmptyRecycleBinA(HWND hwnd, LPCSTR pszRootPath, DWORD dwFlags);

-- 参数
-- DWORD dwFlags
SHERB_NOCONFIRMATION = 1
-- 不会显示任何确认删除对象的对话框。
SHERB_NOPROGRESSUI = 2
-- 不显示指示进度的对话框。
SHERB_NOSOUND = 4
-- 操作完成后不会播放任何声音。
-- https://learn.microsoft.com/en-us/windows/win32/api/shellapi/nf-shellapi-shemptyrecyclebina

-- 清空所有驱动器上的所有回收站，第三个参数 dwFlags 为 0, 不进行特别设置，
-- 将弹出是否清空回收站提示，清空时显示进度对话框，且清空时有清空操作音效。
os.rundll('shell32.dll','SHEmptyRecycleBinA',0,0,0)
```

扩展 string 库

string.envstr (str)

展开 环境变量

Parameters:

- str string 包含环境变量引用的字符串

Usage:

```
string.envstr('%ProgramFiles%\WinXShell\WinXShell.exe')
---> X:\Program Files\WinXShell\WinXShell.exe
local str1 = [[%SystemRoot%\System32\regedit.exe]]
str1:envstr() ---> C:\Windows\System32\regedit.exe
```

string.resstr (str)

展开 字符串资源

Parameters:

- str string 包含字符串资源引用的字符串

Usage:

```
string.resstr('#{@shell32.dll,9315}') ---> 本地磁盘
local str2 = '#{@shell32.dll,9316}'
str2:resstr() ---> DVD 驱动器
```

扩展 math 库

math.band (a, b)

进行 位与运算

Parameters:

- a int 操作数 a
- b int 操作数 b

Usage:

```
math.band(5, 6) ---> 4
```

扩展 lua 函数

exec ()

执行外部命令

See also:

os.exec

link ()

创建 快捷方式

See also:

os.link

wxsUI (ui[, jcfg[, param]])

打开 指定的UI组件

Parameters:

- ui string 组件名
- jcfg string 启动配置文件 (*optional*)
- param string 启动参数 (*optional*)

扩展 lua 对象

App

App

Shell

Shell

System

System

Deprecated 已弃用

win_copyright ()

已弃用。获取 Windows 许可证说明

See also:

os.info

cpu_info ()

已弃用。获取 CPU 信息(注册表)

See also:

os.info

mem_info ()

已弃用。获取 内存信息

See also:

os.info

res_str (file, id)

已弃用。获取 资源文件中的字符串信息

Parameters:

- file string 资源文件
- id int 资源号

See also:

string.resstr

call_dll ()

已弃用。调用 dll 中的公开函数

See also:

os.rundll

CloseShellWindow ()

已弃用。关闭 外壳窗口 外壳窗口不存在时，将等待外壳窗口出现后再关闭

See also:

Shell:WaitAndClose

MinimizeWindow (class, title)

已弃用。最小化 指定窗口

Parameters:

- class
- title

See also:

System.Proc

HideWindow (class, title)

已弃用。隐藏 指定窗口

Parameters:

- class
- title

See also:

System.Proc

CloseWindow (class, title)

已弃用。关闭 指定窗口

Parameters:

- class
- title

See also:

System.Proc

generated by LDoc 1.4.6

```
--[[
      Leap Month            L
      14 13 12 11 | 10 F E D | C B A 9 | 8 7 6 5 | 4 3 2 1
2018   0  0  0  0 |  1 1 0 0 | 1 1 0 1 | 0 1 0 0 | 1 0 1 0
2019   0  0  0  0 |  1 1 0 0 | 1 1 0 0 | 1 0 0 1 | 0 1 0 1
2020   0  1  0  0 |  1 1 0 0 | 1 0 1 0 | 1 0 0 1 | 1 1 1 0
2021   0  0  0  0 |  1 0 0 0 | 0 1 0 1 | 0 1 0 1 | 0 1 1 0
2022   0  0  0  0 |  0 1 0 0 | 1 0 1 0 | 1 0 1 1 | 0 1 0 1
2023   0  0  1  0 |  1 0 0 0 | 1 0 1 0 | 1 1 0 1 | 1 0 1 0
2024   0  0  0  0 |  1 0 0 0 | 0 1 1 0 | 1 1 0 1 | 0 0 1 0
2025   0  1  1  0 |  0 1 0 0 | 0 1 1 1 | 0 1 1 0 | 0 1 0 1
2026   0  0  0  0 |  0 1 0 0 | 0 1 1 1 | 0 0 1 0 | 0 1 0 1
]]

lunar_data = {
    ["2018"] = {0,11,15,0x0CD4A},
    ["2019"] = {0,11,26,0x0CC95},
    ["2020"] = {0,12, 7,0x4CA9E},
    ["2021"] = {0,11,18,0x08956},
    ["2022"] = {0,11,29,0x04AB5},
    ["2023"] = {0,12,10,0x28ADA},
    ["2024"] = {0,11,20,0x086D2},
    ["2025"] = {0,12, 2,0x64765},
    ["2026"] = {0,11,13,0x04725},
}

lunar_cache = {}

local function get_month_day(year, mon)
  local days = {31,28,31,30,31,30,31,31,30,31,30,31}
  if mon == 2 and year % 4 == 0 then  -- enough for 1901~2099
    return 29
  else
    return days[mon]
  end
end

local lunar = {}
local lunar_enabled = false

function lunar_init()
    local locale =  sui:jcfg('lunar')
    if locale == 'none' then return end
    if locale == nil or locale == 'auto' then locale = os.info('locale') end

    local lf = sui:info('uipath') .. 'lunar.' .. locale .. '.lua'
    if not File.Exists(lf) then return end

    dofile(lf)
    -- month_days_area.height = 482
    for i=1,6 do
        for j=1,7 do
          d[i ..j].multiline = "true"
        end
    end
    sui:move(0, -72, 0, 72)
    lunar_enabled = true
```

```
end

function lunar.get_disp_text(m, d, leap)
   local key_md = string.format("%02d%02d", m, d)
    if lunar_specs[key_md] ~= nil then return lunar_specs[key_md] end
    if d == 1 then
        local str_m = lunar_months[m]
        if leap == 1 then str_m =  '<f 12>' .. lunar_leap_name .. str_m .. '</f>' end
        return str_m
    end
    return lunar_days[d]
end

function lunar.init_month_text(y, m, lm, ld, leap, lbits, lm_shift)
    local ym = string.format("%02d%02d", y, m)

    local mon_text = {}
    local n = get_month_day(y, m)

    local leap_month = lbits & 0xF0000
    leap_month = leap_month >> 16

    local be30d = lbits & (1 << (lm + lm_shift - 1))
    if leap == 1 then be30d = lbits & (1 << 12) end
    if be30d > 0 then be30d = 1 end
    App:Print(ym .. ":[" .. lm .. "," .. ld .. "] " .. be30d)
    for d = 1, n do
        mon_text[d] = lunar.get_disp_text(lm, ld, leap)
        if be30d == 0 and lm == 12 and ld == 29 then
            mon_text[d] = lunar.get_disp_text(12, 30, 0)
        end
        -- App:Print(d .. ":" .. mon_text[d])
        -- next lunar day
        ld = ld + 1
        if ld > 29 + be30d then
            ld = 1
            -- next month
            if (leap == 0 and leap_month == lm) then
                leap = 1
            else
                lm = lm + 1
                if leap == 1 then leap = 0 end
            end
            if lm == 13 then
                lm_shift = 0
                lm = 1
            end
            if leap == 1 then
                be30d = lbits & (1 << 12)
            else
                be30d = lbits & (1 << (lm + lm_shift - 1))
            end
            if be30d > 0 then be30d = 1 end
        end
    end
    lunar_cache[ym] = mon_text
    return lm, ld, leap, lm_shift
```

```lua
end

function lunar.get_year_text(y)
    local str_y = string.format("%d", y)
    -- return from lunar_cache
    if lunar_cache[str_y] ~= nil then return lunar_cache[str_y] end

    local ldata = lunar_data[str_y]
    if ldata == nil then return nil end

    local mon_text = {}
    local lm = ldata[2]  -- lunar month
    local ld = ldata[3]  -- lunar day
    local leap = 0
    local lbits = ldata[4] -- lunar bits
    local lm_shift = 4
    local m = 1
    for m = 1, 12 do
        lm, ld, leap, lm_shift = lunar.init_month_text(y, m, lm, ld, leap, lbits, lm_shift)
    end
    lunar_cache[str_y] = 1
    return 1
end

function lunar.get_month_text(y, m)
    local ym = string.format("%02d%02d", y, m)
    -- return from lunar_cache
    if lunar_cache[ym] ~= nil then return lunar_cache[ym] end

    if lunar.get_year_text(y) == nil then return nil end
    return lunar_cache[ym]
end

function get_day_text(y, m, d)
    if not lunar_enabled then return d end

    local ldata = lunar.get_month_text(y, m)
    if  ldata == nil then return d end
    return d .. '\n' .. ldata[d]
    -- return d .. '\n' .. d
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\lunar.lua"
```

```
--[[
      Leap Month           L
      14 13 12 11 | 10 F E D | C B A 9 | 8 7 6 5 | 4 3 2 1
2018   0  0  0  0 |  1 1 0 0 | 1 1 0 1 | 0 1 0 0 | 1 0 1 0
2019   0  0  0  0 |  1 1 0 0 | 1 1 0 0 | 1 0 0 1 | 0 1 0 1
2020   0  1  0  0 |  1 1 0 0 | 1 0 1 0 | 1 0 0 1 | 1 1 1 0
2021   0  0  0  0 |  1 0 0 0 | 0 1 0 1 | 0 1 0 1 | 0 1 1 0
2022   0  0  0  0 |  0 1 0 0 | 1 0 1 0 | 1 0 1 1 | 0 1 0 1
2023   0  0  1  0 |  1 0 0 0 | 1 0 1 0 | 1 1 0 1 | 1 0 1 0
2024   0  0  0  0 |  1 0 0 0 | 0 1 1 0 | 1 1 0 1 | 0 0 1 0
2025   0  1  1  0 |  0 1 0 0 | 0 1 1 1 | 0 1 1 0 | 0 1 0 1
]]

lunar_data = {
    ["2018"] = {0,11,15,0x0CD4A},
    ["2019"] = {0,11,26,0x0CC95},
    ["2020"] = {0,12, 7,0x4CA9E},
    ["2021"] = {0,11,18,0x08956},
    ["2022"] = {0,11,29,0x04AB5},
    ["2023"] = {0,12,10,0x28ADA},
    ["2024"] = {0,11,20,0x086D2},
    ["2025"] = {0,12, 2,0x64765},
}

lunar_cache = {}

local function get_month_day(year, mon)
  local days = {31,28,31,30,31,30,31,31,30,31,30,31}
  if mon == 2 and year % 4 == 0 then  -- enough for 1901~2099
    return 29
  else
    return days[mon]
  end
end

local lunar = {}
local lunar_enabled = false

function lunar_init()
    local locale =  sui:jcfg('lunar')
    if locale == 'none' then return end
    if locale == nil or locale == 'auto' then locale = os.info('locale') end

    local lf = sui:info('uipath') .. 'lunar.' .. locale .. '.lua'
    if not File.Exists(lf) then return end

    dofile(lf)
    -- month_days_area.height = 482
    for i=1,6 do
        for j=1,7 do
          d[i ..j].multiline = "true"
        end
    end
    sui:move(0, -72, 0, 72)
    lunar_enabled = true
end
```

```
function lunar.get_disp_text(m, d, leap)
   local key_md = string.format("%02d%02d", m, d)
    if lunar_specs[key_md] ~= nil then return lunar_specs[key_md] end
    if d == 1 then
        local str_m = lunar_months[m]
        if leap == 1 then str_m =  '<f 12>' .. lunar_leap_name .. str_m .. '</f>' end
        return str_m
    end
    return lunar_days[d]
end

function lunar.init_month_text(y, m, lm, ld, leap, lbits, lm_shift)
    local ym = string.format("%02d%02d", y, m)

    local mon_text = {}
    local n = get_month_day(y, m)

    local leap_month = lbits & 0xF0000
    leap_month = leap_month >> 16

    local be30d = lbits & (1 << (lm + lm_shift - 1))
    if leap == 1 then be30d = lbits & (1 << 12) end
    if be30d > 0 then be30d = 1 end
    App:Print(ym .. ":[" .. lm .. "," .. ld .. "] " .. be30d)
    for d = 1, n do
        mon_text[d] = lunar.get_disp_text(lm, ld, leap)
        if be30d == 0 and lm == 12 and ld == 29 then
            mon_text[d] = lunar.get_disp_text(12, 30, 0)
        end
        -- App:Print(d .. ":" .. mon_text[d])
        -- next lunar day
        ld = ld + 1
        if ld > 29 + be30d then
            ld = 1
            -- next month
            if (leap == 0 and leap_month == lm) then
                leap = 1
            else
                lm = lm + 1
                if leap == 1 then leap = 0 end
            end
            if lm == 13 then
                lm_shift = 0
                lm = 1
            end
            if leap == 1 then
                be30d = lbits & (1 << 12)
            else
                be30d = lbits & (1 << (lm + lm_shift - 1))
            end
            if be30d > 0 then be30d = 1 end
        end
    end
    lunar_cache[ym] = mon_text
    return lm, ld, leap, lm_shift
end
```

```
function lunar.get_year_text(y)
   local str_y = string.format("%d", y)
   -- return from lunar_cache
   if lunar_cache[str_y] ~= nil then return lunar_cache[str_y] end

   local ldata = lunar_data[str_y]
   if ldata == nil then return nil end

   local mon_text = {}
   local lm = ldata[2]  -- lunar month
   local ld = ldata[3]  -- lunar day
   local leap = 0
   local lbits = ldata[4] -- lunar bits
   local lm_shift = 4
   local m = 1
   for m = 1, 12 do
      lm, ld, leap, lm_shift = lunar.init_month_text(y, m, lm, ld, leap, lbits, lm_shift)
   end
   lunar_cache[str_y] = 1
   return 1
end

function lunar.get_month_text(y, m)
   local ym = string.format("%02d%02d", y, m)
   -- return from lunar_cache
   if lunar_cache[ym] ~= nil then return lunar_cache[ym] end

   if lunar.get_year_text(y) == nil then return nil end
   return lunar_cache[ym]
end

function get_day_text(y, m, d)
   if not lunar_enabled then return d end

   local ldata = lunar.get_month_text(y, m)
   if  ldata == nil then return d end
   return d .. '\n' .. ldata[d]
   -- return d .. '\n' .. d
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Calendar\lunar.lua"
```

```
lunar_leap_name = "闰"
lunar_months = {"正月","二月","三月","四月","五月","六月", "七月","八月","九月","十月","<f12>十一月</f>","腊月"}

lunar_days = {
    "初一","初二","初三","初四","初五",
    "初六","初七","初八","初九","初十",
     "十一","十二","十三","十四","十五",
    "十六","十七","十八","十九","二十",
    "廿一","廿二","廿三","廿四","廿五",
    "廿六","廿七","廿八","廿九","三十"
}

lunar_specs = {
    ["1230"] = "除夕", ["0101"] = "春节", ["0115"] = "元宵",
    ["0505"] = "端午", ["0815"] = "中秋", ["0909"] = "重阳",
}
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\lunar.zh-CN.lua"
```

```
lunar_leap_name = "闰"
lunar_months = {"正月","二月","三月","四月","五月","六月", "七月","八月","九月","十月","<f12>十一月</f>","腊月"}

lunar_days = {
    "初一","初二","初三","初四","初五",
    "初六","初七","初八","初九","初十",
     "十一","十二","十三","十四","十五",
    "十六","十七","十八","十九","二十",
    "廿一","廿二","廿三","廿四","廿五",
    "廿六","廿七","廿八","廿九","三十"
}

lunar_specs = {
    ["1230"] = "除夕", ["0101"] = "春节", ["0115"] = "元宵",
    ["0505"] = "端午", ["0815"] = "中秋", ["0909"] = "重阳",
}
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Calendar\lunar.zh-CN.lua"
```

```
lunar_leap_name = "秥"
lunar_months = {"タる","  る","  る","  る","きる","せる", "  る","  る","  る","<f 12>  る
</f>","    る","鎂る"}

lunar_days = {
    "    ","    ","    ","    ","  き",
    "  せ","    ","    ","    ","    ",
     "    ","    ","    ","    ","  き",
    "  せ","    ","    ","    ","    ",
    "ぼ  ","ぼ  ","ぼ  ","ぼ  ","ぼき",
    "ぼせ","ぼ  ","ぼ  ","ぼ  ","    "
}

lunar_specs = {
    ["1230"] = "埃  ", ["0101"] = "琄竊", ["0115"] = "じ甦",
    ["0505"] = "狠と", ["0815"] = "い  ", ["0909"] = "  锭",
}
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\lunar.zh-TW.lua"
```

```
lunar_leap_name = "秥"
lunar_months = {"タる","  る","  る","  る","きる","せる", "  る","  る","  る","<f 12>  る
</f>","    る","镁る"}

lunar_days = {
    "    ","    ","    ","    ","  き",
    "  せ","    ","    ","    ","    ",
     "    ","    ","    ","    ","  き",
    "  せ","    ","    ","    ","    ",
    "ぼ  ","ぼ  ","ぼ  ","ぼ  ","ぼき",
    "ぼせ","ぼ  ","ぼ  ","ぼ  ","    "
}

lunar_specs = {
    ["1230"] = "埃  ", ["0101"] = "琄竊", ["0115"] = "じ甦",
    ["0505"] = "狠と", ["0815"] = "い  ", ["0909"] = "  锭",
}
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Calendar\lunar.zh-TW.lua"
```

```
{
  "description" : {
    "name":"string. Specify the UI name.",
    "title":"string. Specify the title of UI window(also show on the taskbar). Support locale resource id.
optional(<name>).",
    "entry":"string. Specify the Entry file. optional(main.xml).",
    "singleton":"boolean. wether the window is only one instance or not. optional(false).",
    "noshadow":"boolean. wether use Shadow.png for window shadow effort or not. optional(true).",
    "position":"string(center|rightbottom ). Specify the startup position. optional(center).",
    "nobaricon":"boolean. wether show icon on the taskbar or not. optional(false).",
    "baricon":"string. Specify the taskbar icon. optional(main.ico).",
    "customstyle":"boolean. wether use custom the window style or not. optional(false).",
    "style":"long. Specify the window style. Please refer to Microsoft documentation.
optional(WS_OVERLAPPEDWINDOW).",
    "exstyle":"long. Specify the window extend style. Please refer to Microsoft documentation.
optional(WS_EX_WINDOWEDGE).",
    "theme":"string. Specify the theme(themes\*.xml). optional(default).",
    "locale":"string. Define the locale resources for the UI text display, ie:zh-CN, en-US. optional(SYSTEM default
locale)."
  },
  "name":"UI_Shutdown",
  "title":"%{Power}",
  "baricon":"power.ico"
}
{
  "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\main.en-US.jcfg",
}
```

```
{
  "name":"UI_AppStore",
  "title":"AppStore",
  "baricon":"icon.ico"
}
{
  "ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\main.jcfg",
}
```

```
{
  "name":"UI_WIFI",
  "class":"CDUIWIFIWindow",
  "singleton":true,
  "position":"(auto)",
  "nobaricon":true,
  "#exstyle comment":"WS_EX_WINDOWEDGE(256) | WS_EX_TOPMOST(8)",
  "customstyle":true,
  "exstyle":264,
  "trans":220,
  "#tray_update_init":1000,
  "#tray_update_interval":2000,
  "#wirednetworkicon":true,
  "#legacy_disconnect_icon":false,
  "OnDeactive":"hide",
  "JS_CMD":
  {
    "::enable_firewall":{
      "command":"Wpeutil.exe",
      "parameters":"Enablefirewall",
      "flags meaning":"0 - SW_HIDE",
      "showflags":0
    },
    "::disable_firewall":{
      "command":"Wpeutil.exe",
      "parameters":"DisableFirewall",
      "showflags":0
    }
  }
}
{
  "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_WIFI\main.jcfg",
}
```

```
{
 "name":"UI_Calendar",
 "class":"CDUICalendarWindow",
 "shelltheme":true,
 "title":"Calendar",
 "entry":"main.xml",
 "#overwrite_with_options":"-brightness=true, -brightness=false",
 "brightness":true,
 "lua":"main.lua",
 "singleton":true,
 "singleton_timer_interval":200,
 "position":"(auto)",
 "nobaricon":true,
 "#exstyle comment":"WS_EX_WINDOWEDGE(256) | WS_EX_TOPMOST(8)",
 "customstyle":true,
 "exstyle":264,
 "trans":240,
 "OnDeactive":"hide",
 "#lunar":"none|auto"
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Calendar\main.jcfg",
}
```

```
{
 "name":"UI_LED",
 "nobaricon":true,
 "customstyle":true,
 "style":2415919104,
 "exstyle":524424
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_LED\main.jcfg",
}
```

```
{
 "name":"UI_Logon",
 "title":"wxsLogon",
 "nobaricon":true,
 "customstyle":true,
 "style":2415919104,
 "startup_win":"max",
 "OnEscKey":"none"
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Logon\main.jcfg",
}
```

```
{
  "name":"UI_TrayPanel",
  "shelltheme":true,
  "singleton":true,
  "singleton_timer_interval":200,
  "position":"(auto)",
  "nobaricon":true,
  "#exstyle comment":"WS_EX_WINDOWEDGE(256) | WS_EX_TOPMOST(8)",
  "customstyle":true,
  "trans":240,
  "OnDeactive":"hide",
  "#lunar":"none|auto"
}
{
  "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\main.jcfg",
}
```

```
{
 "name":"UI_NotifyInfo",
 "nobaricon":true,
 "position":"rightbottom"
}
{
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\main.jcfg",
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\main.jcfg",
}
```

```
{
 "name":"UI_Shutdown",
 "title":"%{Power}",
 "baricon":"power.ico"
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\main.jcfg",
}
```

```
{
 "name":"UI_Volume",
 "class":"CDUIVolumeWindow",
 "singleton":true,
 "position":"(auto)",
 "nobaricon":true,
 "entry":"main10.xml",
 "#exstyle comment":"WS_EX_WINDOWEDGE(256) | WS_EX_TOPMOST(8)",
 "customstyle":true,
 "exstyle":264,
 "trans": 245,
 "OnDeactive":"hide"
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Volume\main.jcfg",
}
```

```
{
  "说明" : {
    "name":"字符串。指定UI名。必须参数。",
    "entry":"字符串。指定入口文件。可缺省(main.xml)。",
    "singleton":"布尔值。表明窗口是否唯一，如果为true，当该类型窗口存在时激活该窗口而不是创建新窗
口，多用于设置窗口。可缺省(false)。",
    "noshadow":"布尔值。表明窗口是否禁用阴影效果(Shadow.png)，可缺省(true)。",
    "position":"固定字符串(center - 屏幕中央|rightbottom - 桌面右下角)。窗口的启动位置。可缺
省(center)。",
    "nobaricon":"布尔值。表明是否在任务栏显示图标。可缺省(false)。",
    "baricon":"字符串。任务栏图标资源。可缺省(main.ico)。",
    "customstyle":"布尔值。表明是否指定窗口风格。可缺省(false)。",
    "style":"长整数值。指定窗口风格，具体值请参考微软说明文档。可缺
省(WS_OVERLAPPEDWINDOW)。",
    "exstyle":"长整数值。指定窗口扩展风格，具体值请参考微软说明文档。可缺
省(WS_EX_WINDOWEDGE)。",
    "theme":"字符串。指定主题(themes\*.xml)。可缺省(default)。",
    "locale":"字符串。指定用于界面文字显示的本地语言资源。例如:zh-CN，en-US。可缺省(系统默认语
言)。"
  },
  "name":"UI_Resolution",
  "baricon":"settings.ico",
  "OnLoad":"
= #resolution_combo.list, ::GetResolutionList();
= #resolution_combo.index, ::GetCurrentResolution();
= #rotate_combo.index, ::GetRotation();
= $last_resolution, #resolution_combo.text;
= $last_resolution_index, #resolution_combo.index;
= $last_rotate, #rotate_combo.index;
= #applybtn.enabled, FALSE;
  ",
  "OnChanged":{
  "resolution_combo":"
= #applybtn.enabled, TRUE;
  ",
  "rotate_combo":"
= #applybtn.enabled, TRUE;
  "
  },
  "OnClick":{
  "applybtn":"
PUSH #resolution_combo.text;
PUSH 1;
CALL ::SetResolution();
PUSH #rotate_combo.index;
PUSH 1;
CALL ::SetRotation();
= #applybtn.visible,  FALSE;
= #cancelbtn.visible, FALSE;
= #closebtn.visible,   TRUE;
= #restorebtn.visible, TRUE;
  ",
  "restorebtn":"
PUSH $last_resolution;
PUSH 1;
CALL ::SetResolution();
PUSH $last_rotate;
PUSH 1;
CALL ::SetRotation();
= #resolution_combo.index, $last_resolution_index;
= #applybtn.enabled,   FALSE;
= #applybtn.visible,     TRUE;
= #cancelbtn.visible,   TRUE;
= #closebtn.visible,   FALSE;
= #restorebtn.visible, FALSE;
  ",
  "closebtn":"
  CALL ::Close();
  ",
  "cancelbtn":"
  CALL ::Close();
  "
  }
}
{
  "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Resolution\main.jcfg",
}
```

```
{
  "name":"UI_Launcher",
  "singleton":true,
  "nobaricon":true,
  "position":"leftbottom",
  "OnDeactive":"hide"
}
{
  "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Launcher\main.jcfg",
}
```

```
{
  "name":"UI_Settings",
  "usesystemcaption": false,
  "minimizebox": true,
  "maximizebox": false,
  "title":"%{Settings}",
  "baricon":"settings.ico"
}
{
  "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Settings\main.jcfg",
}
```

```
APP_Path = app:info('path')
UI_Path = sui:info('uipath')
dofile(UI_Path .. 'utils.lua')
dofile(UI_Path .. 'nav.lua')
dofile(UI_Path .. 'page.lua')
dofile(UI_Path .. 'helper_openfile.lua') --helper_openfile
dofile(UI_Path .. 'helper_ntboot.lua') --helper_ntboot
dofile(UI_Path .. 'helper_nbdl.lua') --helper_nbdl
function onload()
  UI_Inited = 0
  Nav:Init()
  helper_nbdl_onload() --helper_nbdl
  helper_openfile_onload() --helper_openfile
  helper_ntboot_onload() --helper_ntboot
  UI_Inited = 1
end



function onclick(ctrl)
   app:print('onclick:' .. ctrl)
  helper_nbdl_onclick(ctrl) --helper_nbdl
   if Nav:OnTabClick(ctrl) then return end
end



--"ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\main.lua"
```

```
applybtn = sui:find('applybtn')
cancelbtn = sui:find('cancelbtn')
restorebtn = sui:find('restorebtn')
closebtn = sui:find('closebtn')

resolution_combo = sui:find('resolution_combo')
rotate_combo = sui:find('rotate_combo')

RESTORE_TIMER_ID = 1000
AUTO_RESTORE_SEC = 15
restore_counter = AUTO_RESTORE_SEC
org_restorebtn_text = restorebtn.text -- save origin retore button text for retore timer

function onload()
   resolution_combo.list = suilib.call('GetResolutionList')
   resolution_combo.index = suilib.call('GetCurrentResolution')

   rotate_combo.index = suilib.call('GetRotation')
   last_resolution = resolution_combo.text
   last_rotate = rotate_combo.index
   applybtn.enabled = 0 -- disable apply button
   if App:HasOption('-direct') then
      direct_mode = 1
   end
   loaded = true
end


function onchanged(ctrl)
   if not loaded then return end
   if ctrl == "resolution_combo" or ctrl == "rotate_combo" then
      -- change the resolution/rotate directly if has '-direct' option
      if direct_mode == 1 then
         -- oops, you can change the settings even in auto restore counting.
         -- Can you fix this?
         applybtn_click()
      else
         applybtn.enabled = 1
      end
   end
end

function ontimer(id)
   suilib.print('ontimer:' .. id)
   if (id == RESTORE_TIMER_ID) then
      restore_counter = restore_counter - 1
      restorebtn.text = string.format('[%d]%s', restore_counter, org_restorebtn_text)
      -- auto restore
      if (restore_counter <= 0) then
         suilib.call('KillTimer', id)
         restorebtn_click()
      end
   end
end

function applybtn_click()
```

```
    suilib.call('SetResolution', resolution_combo.text)
    suilib.call('SetRotation', rotate_combo.index)
    applybtn.visible = 0
    cancelbtn.visible = 0
    closebtn.visible = 1
    restorebtn.visible = 1
    -- set auto restore timer
    restore_counter = AUTO_RESTORE_SEC
    restorebtn.text = string.format('[%d]%s', restore_counter, org_restorebtn_text)
    suilib.call('SetTimer', RESTORE_TIMER_ID, 1000)
end


function restorebtn_click()
    suilib.call('KillTimer', RESTORE_TIMER_ID)
    suilib.call('SetResolution', last_resolution)
    suilib.call('SetRotation', last_rotate)
    resolution_combo.index = last_resolution
    rotate_combo.index = last_rotate
    applybtn.visible = 1
    cancelbtn.visible = 1
    closebtn.visible = 0
    restorebtn.visible = 0
    applybtn.enabled = 0
end

function onclick(ctrl)
    if ctrl == "applybtn" then
        applybtn_click()
    elseif ctrl == "restorebtn" then
        restorebtn_click()
    elseif ctrl == "closebtn" or ctrl == "cancelbtn" then
        sui:close()
    end
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Resolution\main.lua"
```

```lua
local TIMER_ID_QUIT = 1001
wait = 2

function onload()
  sui:find("$icon").bkimage = App:GetOption('-i') .. '.png'
  sui:find("$title").text =App:GetOption('-t')
  sui:find("$info").text = App:GetOption('-m')
  pos = App:GetOption('-p')
  if pos ~= nil then
      pos = pos + 0
      local w, h = sui:info('wh')
      sui:move(0, -((h + 10)*pos))
  end
  suilib.call('SetTimer', TIMER_ID_QUIT, wait * 1000)
end

function ontimer(id)
   if id == TIMER_ID_QUIT then sui:close() end
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\main.lua"
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\main.lua"
```

```
-- Windows Definition
MB_ICONASTERISK = 64 -- Type for MessageBeep()
SW_SHOWNORMAL = 1

-- Timer
BEEP_TIMER_ID = 1000

nobeep_mode = 0

local last_vol_state_img = ''
local last_ismuted = 1

local function vol_btn_icon(ismuted, init)
  local vol_state_img = "file='snd_mute.ico'"

  if init then
    vol_btn.selected = ismuted
    last_ismuted = ismuted
  end

  if ismuted == 0 then
    local i = math.modf(vol_level / 33)
    i = i + 1
    if i > 3 then i = 3 end
    vol_state_img = "file='snd_" .. i .. ".ico'"
  end

  if vol_state_img ~= last_vol_state_img then
    vol_state.bkimage = vol_state_img
    last_vol_state_img = vol_state_img
  end
end

local function init_elem()
  vol_level = Volume:GetLevel()
  local mute = Volume:IsMuted()
  App:Print(volume_slider)
  volume_slider.value = vol_level
  vol_value.text = vol_level
  vol_level = tonumber(vol_level)

  vol_btn_icon(mute, 1)

  need_beep = 0
  suilib.call('KillTimer', BEEP_TIMER_ID)
  suilib.call('SetTimer', BEEP_TIMER_ID, 500)
end

function onload()
  sui:find('vol_name').text = Volume:GetName()
  vol_value = sui:find('vol_value')
  volume_slider = sui:find('volume_slider')
  App:Print(volume_slider)
  vol_btn = sui:find('vol_btn')
  vol_state = sui:find('vol_state')
  sui:find('sndvol_settings').text = sui:find('sndvol_settings').text

  if App:HasOption('-nobeep') then
    nobeep_mode = 1
  end

  ui_path = sui:jcfg('res_path')
  -- local sd = os.getenv("SystemDrive")
  -- if not File.exists(sd ..'\\Windows\\media\\Windows Background.wav') then
  --   App:Run('xcopy', '"' .. ui_path .. 'Windows Background.wav" "' .. sd ..'\\Windows\\media\\"', 0)
  -- end
  LOADED = 1
  init_elem()
end

function onshow()
  if not LOADED then return end
  init_elem()
end

function onchanged(ctrl, val)
  local ismuted = 1;
  if not LOADED then return end
  if ctrl == 'volume_slider' then
    vol_value.text = val
    vol_level = val
    vol_changed = 1
    if val > 0 then ismuted = 0 end

    if ismuted ~= last_ismuted then
      vol_btn.selected = ismuted
      last_ismuted = ismuted
      Volume:Mute(ismuted)
    end
  elseif ctrl == 'vol_btn' then
    ismuted = vol_btn.selected
    last_ismuted = ismuted
    Volume:Mute(ismuted)
  end
  vol_btn_icon(ismuted)
end

UISTATE_CAPTURED = 64 -- Drag & Drop - ing

function ontimer(id)
  if id == BEEP_TIMER_ID then
    if sui:ishidden() == 1 then -- no need watch the slider change
      suilib.call('KillTimer', BEEP_TIMER_ID)
    end
    if vol_changed == 1 then
      Volume:SetLevel(volume_slider.value)
      if nobeep_mode == 0 and math.band(volume_slider.state, UISTATE_CAPTURED) == 0 then
        -- App:Call('Beep', MB_ICONASTERISK) -- Beep can't work in PE
        App:Call('Play', ui_path .. 'Windows Background.wav', 0)
        vol_changed = 0
      end
    end
  end
end

local function hideme()
  sui:hide()
  suilib.call('KillTimer', BEEP_TIMER_ID)
end

function onclick(ctrl)
  if ctrl == 'sndvol_settings' or ctrl == 'vol_value' then
    hideme()
    App:Run('sndvol.exe', '-m 268439552', SW_SHOWNORMAL)
  elseif ctrl == 'closebtn' then
    hideme()
  end
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Volume\main.lua"
```

```
local shadow = sui:jcfg('shadow') or 'Administrator:admin;\nSYSTEM:;'
local logon_user = sui:jcfg('logon_user') or 'SYSTEM'
local auto_logon_second = sui:jcfg('auto_logon_second') or 5
local logon_blur = sui:jcfg('logon_blur') or '10.0'

CLOCK_TIMER_ID = 1000
AUTO_USER_TIMER_ID = 1001
PASS_TIMER_ID = 1002
LOGON_TIMER_ID = 1003
DOLOGON_TIMER_ID = 1004

UIUPDATE_TIMER_ID = 8000
EXIT_TIMER_ID = 9000

-- local functions
local on_user_click

-- internal variable(s)
local autologon_countdown = auto_logon_second

-- "startup_win":"max"
local function full_screen()
    --sui:moveto(0, 0, Screen:GetX(), Screen:GetY())
end

local function draw_background()
    wallpaper = ''
    local blur = App:GetOption('-blur')
    if blur == nil then blur = '0.0' end

    wallpaper = App:GetOption('-bk')
    if wallpaper == nil then
        wallpaper = Desktop:GetWallpaper()
    end

    local win = sui:find('full_win')
    win.bkimage = "file='" .. wallpaper .. "' blur='" .. blur .. "'"
end

local function draw_now_info()
    local now = os.time()
    now_time.text = os.date('%H:%M:%S', now)
    now_info.text = string.format('%s, %%{info_w%s}', os.date("%Y-%m-%d", now), os.date('%w', now))

end

local function blur_background()
  local win = sui:find('full_win')
  -- blur trick
  win.bkimage = "file='" .. wallpaper .. " ' blur='" .. logon_blur .. "'"
end


local function draw_autologon_info()
    if autologon_countdown < 0 then
        return
    elseif autologon_countdown > 0 then
```

```
        local msg = string.gsub(autologon_info_message, '$user', logon_user)
        msg = string.gsub(msg, '$countdown', autologon_countdown)
        progress_text.text = msg
        autologon_countdown = autologon_countdown - 1
    elseif autologon_countdown == 0 then
      progress_text.text = "
      autologon_countdown = autologon_countdown - 1
    end
end

local function show_pass_input_frame(internal)
    text_username.text = logon_user
    userlist_frame.visible = 0
    pass_input_frame.visible = 1
    if internal == nil then internal = 500 end
    suilib.call('SetTimer', PASS_TIMER_ID, internal)
end

function onload()
    App:Call('UserLogonInit')
    full_screen()
    draw_background()
    autologon_info_message = sui:i18n('will autologon with the $user account in $countdown second(s)')

    if App:GetOption('-user') then
        logon_user = App:GetOption('-user')
        auto_logon_second = 0
    end
    text_username = sui:find('username')
    text_password = sui:find('password')
    userlist_frame = sui:find('userlist_frame')
    pass_input_frame = sui:find('pass_input_frame')
    progress_text = sui:find('progress_text')

    now_time = sui:find('now_time')
    now_info = sui:find('now_info')
    draw_now_info()
    suilib.call('SetTimer',CLOCK_TIMER_ID, 1000)

    _G.progress_text = progress_text

    if logon_user ~= " and auto_logon_second == 0 then
      autologon_countdown = -1
      on_user_click(logon_user, 2000)
    else
      userlist_frame.visible = 1
      if logon_user ~= " and auto_logon_second > 0 then
        draw_autologon_info()
        suilib.call('SetTimer', AUTO_USER_TIMER_ID, auto_logon_second * 1000)
      end
    end
end

local function checkpass()
  local u = text_username.text
  local p = text_password.text
  if string.find(shadow, u .. ':' .. p .. ';') then return 0 end
  return 1
end

local function hide_others_logo(username)
  local other = 'SYSTEM'
  if other == username then other = 'Administrator' end
  local user_logo = sui:find('User_' .. other)
  if user_logo then user_logo.visible = 0 end
end

local function call_logon_lua()
  local logon_script = os.getenv('logon_script')
  _G.caller = 'UI_Logon'
  _G.logon_user = logon_user
  dofile(logon_script)
end

local function regist_logon_coroutine()
  cor_logon_lua = coroutine.create(call_logon_lua)
  suilib.call('SetTimer', DOLOGON_TIMER_ID, 500)
  suilib.call('SetTimer', UIUPDATE_TIMER_ID, 200)
end

local function do_logon()
  pass_input_frame.visible = 0
  userlist_frame.visible = 1
  hide_others_logo(logon_user)
  local user_logo = sui:find('User_' .. logon_user)
  if user_logo then
    user_logo.mouse = 'false'
  end

  if logon_user == 'SYSTEM' then
    App:Call('exitcode', 1)
  else
    App:Call('exitcode', 2)
  end

  suilib.call('KillTimer', CLOCK_TIMER_ID)
  now_time.visible = 0
  now_info.visible = 0

  progress_text.text = '%{Welcome to Windows ...}'

  if os.getenv('logon_script') ~= nil then
    regist_logon_coroutine()
  else
    suilib.call('SetTimer', EXIT_TIMER_ID, 1000)
  end
end

function on_user_click(username, internal)
  suilib.call('KillTimer', AUTO_USER_TIMER_ID)
  autologon_countdown = -1
  logon_user = username

  -- hide logon info label
  sui:find('logoninfo').visible = 0

  blur_background()
  -- skip 'show_pass_input_frame' if empty password
  if string.find(shadow, username .. ':;') then
    suilib.call('SetTimer', LOGON_TIMER_ID, 500)
  else
    show_pass_input_frame(internal)
  end
end

local function on_logon_click()
  if (checkpass() == 0) then
      suilib.call('SetTimer', LOGON_TIMER_ID, 100)
  end
end


local function power_helper(wu_param, sd_param)
  local sd = os.getenv("SystemDrive")
  if File.Exists(sd ..'\\Windows\\System32\\Wpeutil.exe') then
    App:Run('wpeutil.exe', wu_param, 0) -- SW_HIDE(0)
    sui:close()
    return 0
  elseif File.Exists(sd ..'\\Windows\\System32\\shutdown.exe') then
    App:Run('shutdown.exe', sd_param .. ' -t 0')
    sui:close()
    return 0
  end
  return 1
end

local function reboot()
    return power_helper('Reboot', '-r')
end

local function shutdown()
    return power_helper('Shutdown', '-s')
end

function onclick(ctrl)
  if ctrl == 'User_SYSTEM' then
    on_user_click('SYSTEM')
  elseif ctrl == 'User_Administrator' then
    on_user_click('Administrator')
  elseif ctrl == 'logon' then
    on_logon_click()
  elseif ctrl == 'restart' then
    reboot()
  elseif ctrl == 'shutdown' then
    shutdown()
  end
end

function onreturn(ctrl)
  if ctrl == 'username' or ctrl == 'password' then
    sui:focus('logon')
    sui:click('logon')
  end
end

function ontimer(id)
  if id == AUTO_USER_TIMER_ID then
    on_user_click(logon_user)
  elseif id == PASS_TIMER_ID then
    sui:focus('password')
    suilib.call('KillTimer', PASS_TIMER_ID)
  elseif  id == LOGON_TIMER_ID then
    suilib.call('KillTimer', LOGON_TIMER_ID)
    do_logon()
  elseif id == DOLOGON_TIMER_ID then
    suilib.call('KillTimer', DOLOGON_TIMER_ID)
    coroutine.resume(cor_logon_lua)
  elseif id == UIUPDATE_TIMER_ID then
    if coroutine.status(cor_logon_lua) == 'dead' then
      sui:close()
    end
    coroutine.resume(cor_logon_lua)
  elseif id == CLOCK_TIMER_ID then
    draw_now_info()
    draw_autologon_info()
  elseif id == EXIT_TIMER_ID then
    sui:close()
  end
end

function onmessage(msg, wparam, lparam)
  if msg == 0x7E then -- WM_DISPLAYCHANGE
    full_screen()
    return 1
  end
  return 0
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Logon\main.lua"
```

```
local trayIconFiles = {
  IconType = "FILE",

  ICID_TRAY_NET_WIRED_DIS = "trayicons/%SuiTray.Theme%/tray_network_wired_dis.ico",
  ICID_TRAY_NET_WIRED_LAN = "trayicons/%SuiTray.Theme%/tray_network_wired_lan.ico",
  ICID_TRAY_NET_WIRED_INTERNET = "trayicons/%SuiTray.Theme%/tray_network_wired_internet.ico",
  ICID_TRAY_NET_WIRELESS_DIS = "trayicons/%SuiTray.Theme%/tray_network_wireless_dis.ico",
  ICID_TRAY_NET_WIRELESS_NOCONN_LEGACY =
"trayicons/%SuiTray.Theme%/tray_network_wireless_noconn_legacy.ico",
  ICID_TRAY_NET_WIRELESS_NOCONN = "trayicons/%SuiTray.Theme%/tray_network_wireless_noconn.ico",
  ICID_TRAY_NET_WIRELESS_LAN = "trayicons/%SuiTray.Theme%/tray_network_wireless_lan.ico",

  ICID_TRAY_NET_SIGNAL_NONE = "trayicons/%SuiTray.Theme%/tray_network_signal_0.ico",
  ICID_TRAY_NET_SIGNAL_QUARTER1 = "trayicons/%SuiTray.Theme%/tray_network_signal_25.ico",
  ICID_TRAY_NET_SIGNAL_QUARTER2 = "trayicons/%SuiTray.Theme%/tray_network_signal_50.ico",
  ICID_TRAY_NET_SIGNAL_QUARTER3 = "trayicons/%SuiTray.Theme%/tray_network_signal_75.ico",
  ICID_TRAY_NET_SIGNAL_QUARTER4 = "trayicons/%SuiTray.Theme%/tray_network_signal_100.ico",
}


local traySymbolIcons = {
  IconType = "SymbolIcon",
  IconFont = "Segoe Fluent Icons",

  ICID_TRAY_NET_WIRED_DIS = "&#xEB55",
  ICID_TRAY_NET_WIRED_LAN = "&#xEB56",
  ICID_TRAY_NET_WIRED_INTERNET = "&#xE904",
  ICID_TRAY_NET_WIRELESS_DIS = "&#xE904",
  ICID_TRAY_NET_WIRELESS_NOCONN_LEGACY = "&#xE904",
  ICID_TRAY_NET_WIRELESS_NOCONN = "&#xE904",
  ICID_TRAY_NET_WIRELESS_LAN = "&#xE904",

  ICID_TRAY_NET_SIGNAL_NONE = "&#xE904",
  ICID_TRAY_NET_SIGNAL_QUARTER1 = "&#xEC3C",
  ICID_TRAY_NET_SIGNAL_QUARTER2 = "&#xEC3D",
  ICID_TRAY_NET_SIGNAL_QUARTER3 = "&#xEC3E",
  ICID_TRAY_NET_SIGNAL_QUARTER4 = "&#xEC3F",
}

function UIWindow:Init()
  SuiTray = {}
  SuiTray.Theme = 'default'
  SuiTray.Icons = trayIconFiles

  local theme = sui:jcfg('res_theme')
  if theme == 'light' then SuiTray.Theme = 'dark' end

  if os.info('winver')[3] >= 22000 then
    SuiTray.Theme = 'win11' .. SuiTray.Theme
  end
end

function UIWindow:OnLoad()
  UIWindow.Inited = 1
end

UI.OnClick['dialup'] = function()
```

```
  App:Run('rasphone.exe')
end

UI.OnClick['netsettings'] = function()
  --control.exe /name Microsoft.NetworkAndSharingCenter
  App:Run('WinXShell.exe', 'wxs-open:netsettings')
end

UI.OnClick['proxysettings'] = function()
  App:Run('rundll32.exe', 'Shell32.dll,Control_RunDLL inetcpl.cpl,,4')
end

UI.OnClick['launch_osk'] = function()
    App:Run('osk.exe')
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_WIFI\main.lua"
```

```lua
dofile(sui:info('uipath') .. 'lunar.lua')

p = print

TICK_TIMER_ID = 1000
POSTION_TIMER_ID = 1001
SET_BRIGHTNESS_TIMER_ID = 1002

local function get_month_day(year, mon)
  local days = {31,28,31,30,31,30,31,31,30,31,30,31}
  if mon == 2 and year % 4 == 0 then
    return 29
  else
    return days[mon]
  end
end

local function fillnum(num, c)
    if c == nil then c = '0' end
    return string.sub(c .. num, -2, -1)
end

function draw_cal(year, mon)
  -- draw title
  nav_mon.text = year .. '-' .. fillnum(mon)
  -- draw days
  local mdate_1 = os.time{year=year, month=mon, day=1, hour=0}
  local week = os.date('%w', mdate_1) -- type(week) => string
  local i, j
  week = math.tointeger(week) -- convert to integer
  suilib.print('week' .. week)
  for i = 1,week do
    d['1' .. i].text = ''
  end


  if last_day_elem ~= nil then
     --last_day_elem.style = 'day'
    last_day_elem.textcolor = normal_textcolor
    last_day_elem.bkimage = normal_bk
     last_day_elem = nil
  end

  local day_elem = nil
  i = week + 1;j = 1
  local md = get_month_day(year, mon)
  for k = 1,42 - week do
    if k <= md then
      d[j .. i].text = get_day_text(year, mon, k) -- fillnum(k, ' ')
      if k == cur_day then
        p("today:" .. k)
        day_elem = d[j .. i]
      end
    else
      d[j .. i].text = ''
    end
    i = i + 1
    if i > 7 then i = 1;j = j + 1 end
  end


  if year == cur_year and mon == cur_mon then
    --day_elem.style = 'today'
    day_elem.textcolor = today_textcolor
    day_elem.bkimage = today_bk
    last_day_elem = day_elem
  end

end

d = {}
local function get_elems()
  for i=1,6 do
    for j=1,7 do
      d[i ..j] = sui:find('d' .. i .. j)
    end
  end
  nav_mon = sui:find('nav_mon')
  now_year = sui:find('now_year')
  now_info = sui:find('now_info')
  now_time = sui:find('now_time')
  month_days_area = sui:find('month_days_area')
  local today_attr = sui:find('today_attr')
  today_textcolor = today_attr.textcolor
  p(today_textcolor)
  today_bk = today_attr.bkimage
  normal_textcolor = d['11'].textcolor
  normal_bk = d['11'].bkimage
end

local function var_init()
  today = os.time()
  cur_year = math.tointeger(os.date('%Y', today))
  cur_mon = math.tointeger(os.date('%m', today))
  cur_day = math.tointeger(os.date('%d', today))
  str_today = cur_year .. cur_mon .. cur_day
end

local function draw_now_info()
    now_info.text = string.format('%d-%d-%d, %%{info_w%s}',
        cur_year, cur_mon, cur_day, os.date('%w', today))
    now_time.text = os.date('%H:%M:%S', today)
end

function onload()

  show_brightness_controller = sui:jcfg('brightness')
  if App:HasOption('-brightness=false') then
    show_brightness_controller = false
  elseif App:HasOption('-brightness=true') then
    show_brightness_controller = true
  end
  if show_brightness_controller == false then
    local bnpart = sui:find('brightness_part')
    bnpart.visible = 0
    sui:move(0, 40, 0, -40)
  end

  get_elems()
  var_init()
  lunar_init()
  view_year = cur_year
  view_mon = cur_mon
  draw_now_info()
  last_day_elem = nil -- mark today
  draw_cal(cur_year, cur_mon)
  suilib.call('SetTimer', TICK_TIMER_ID, 1000)
end

function onshow()
  if show_brightness_controller == false then return end

  brightness_slider = sui:find('brightness_slider')
  brightness_val = Screen:Get('brightness')
  brightness_slider.value = brightness_val
end

function ontimer(id)
    if id == TICK_TIMER_ID then
        var_init()
        draw_now_info()
    elseif id == SET_BRIGHTNESS_TIMER_ID then
      suilib.call('KillTimer', SET_BRIGHTNESS_TIMER_ID)
      local val = math.floor(brightness_val / 10 + 0.5) * 10;
      Screen:Set('brightness', val)
    end
end

function onclick(ctrl)
  if ctrl == 'pre_mon' then
    view_mon = view_mon - 1
    if view_mon == 0 then
        view_year = view_year - 1
        view_mon = 12
    end
    draw_cal(view_year, view_mon)
  elseif ctrl == 'next_mon' then
    view_mon = view_mon + 1
    if view_mon == 13 then
        view_year = view_year + 1
        view_mon = 1
    end
    draw_cal(view_year, view_mon)
  elseif ctrl == 'now_time' then
    suilib.call("run", "timedate.cpl")
  end
end

function onchanged(ctrl, ctrl_val)
  if ctrl == "brightness_slider" then
    App:Print(ctrl .. ':' .. ctrl_val)
    brightness_val = ctrl_val
    suilib.call('KillTimer', SET_BRIGHTNESS_TIMER_ID)
    suilib.call('SetTimer', SET_BRIGHTNESS_TIMER_ID, 200)
  end
end

--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Calendar\main.lua"
```

```
dofile(sui:info('uipath') .. 'apps_tree.lua')

local function update_date_cell()
  local today = os.time()
  label_weekname.text = string.format('%%{info_w%s}', os.date('%w', today))
  label_date.text = string.format('%d', os.date('%d', today))
end

function onload()
  label_weekname = sui:find('label_weekname')
  label_date = sui:find('label_date')
  update_date_cell() -- update in onshow() event only, maybe good if updating with a TIMER
  local lp = sui:find('$left_panel')
  local xml = app_tree_xml('C')
  lp:add([[<elem><Treeview name="app_tree" visible="true" width="300" vscrollbar="true"]] ..
         [[style="ct-bkpanel" itemhotbkcolor="#FF196AA8" itemselectedbkcolor="#FF196AA8" >]] ..
         xml .. [[</Treeview></elem>]])
  --app_tree = sui:find('app_tree')
end

function onshow()
  if label_weekname == nil then return end
  update_date_cell()
end

function onclick(ctrl)
  if ctrl == "firefox" then
    suilib.call("run", "C:\\Program Files (x86)\\Mozilla Firefox\\firefox.exe")
  elseif ctrl == "calc" then
    suilib.call("run", "calc.exe")
  elseif ctrl == "notepad" then
    suilib.call("run", "notepad.exe")
  elseif ctrl == "notepad_paramter" then
    suilib.call("run", "notepad.exe", "C:\\setup.log")
  end
end

--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Launcher\main.lua"
```

```
fixscreen_opt = ''
if App:HasOption('-fixscreen') then fixscreen_opt = '-fixscreen' end

UI_Path = sui:info('uipath')

dofile(UI_Path .. 'parts\\page.display.lua')
dofile(UI_Path .. 'parts\\page.colors.lua')
dofile(UI_Path .. 'parts\\page.folderoptions.lua')
dofile(UI_Path .. 'parts\\page.taskbar.lua')
dofile(UI_Path .. 'parts\\tab.nav.lua')

local function init_navtab()
  NavTab = NavTab_Init()
  NavTab.OnClick = NavTab_OnClick
end

local function init_pages_settings()
  UIPages:Init('Display', NavTab)
  UIPages:Init('Background', NavTab)
  UIPages:Init('Colors', NavTab)
  UIPages:Init('FolderOptions', NavTab)
  UIPages:Init('Taskbar', NavTab)
end

function UIWindow.OnLoad()
  -- TODO: more simple way for resource string
  for i = 1, 6 do
    UI.UpdateText('#res_' .. i)  -- expand resource string
  end

  UI.UpdateText('compmgmt_button')
  UI.UpdateText('$Opt.SysColorTheme[light]')
  UI.UpdateText('$Opt.SysColorTheme[dark]')

  init_navtab()
  init_pages_settings()

  UIWindow.Inited = 1
  if App:HasOption('-display') then
    NavTab:Click('Display')
  elseif App:HasOption('-colors') then
    NavTab:Click('Colors')
  else
    NavTab:Click('Taskbar')
  end
end

UI.OnClick['home_button'] = function()
  wxs_open('controlpanel')
end

UI.OnClick['compmgmt_button'] = function()
  App:Run('compmgmt.msc')
end

--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Settings\main.lua"
```

```
show_text = " The quick brown fox jumps over a lazy dog"
lshift_index = 0
char_width = 2

TIMER_ID_FLUSH_TEXT = 1001
TIMER_ID_QUIT = 1002

function lshift_text()
   local new_text = show_text:sub(lshift_index + 1)
   lshift_index = lshift_index + char_width  -- 全汉字，一次移动2个字节
   if lshift_index > show_text:len() then lshift_index = 0 end
   elem_label_info.text = "<b>" .. new_text .. "</b>"
   elem_label_info.bkcolor = "#FEFFFFFF"
end

function move_top()
  local w, h = sui:info('wh')
  if  App:HasOption('-top') then
    -- 移动到顶部
    sui:move( 0, - ((Screen:GetY() - h) // 2), 0, 0)
  end
end

function onload()
  local text = App:GetOption('-text')
  local wait = App:GetOption('-wait')
  -- winapi.show_message(text, wait)
  if text then
    if text:sub(1, 1) == '"' then text = text:sub(2, -2) end
    show_text = text
  end
  elem_label_info = sui:find('label_info')
  elem_label_info.text = "<b>" ..  show_text .. "</b>"

  -- 按文字内容调整窗体宽度
  local def_w, w
  def_w = sui:info('wh')

  move_top()

  w = #show_text // char_width * 42 -- 全汉字，一个汉字占2个字节
  if show_text:sub(1, 1) == ' ' then
    char_width = 1
    show_text = show_text:sub(2, -1)
    w = #show_text * 20
  end
  if w > Screen:GetX() then w = Screen:GetX() end
  if w > def_w then
    w = w - def_w
    sui:move(- w // 2, 0, w, 0)
  end

  if  App:HasOption('-scroll') then
    -- 滚动计时器，文字滚动效果
    suilib.call('SetTimer', TIMER_ID_FLUSH_TEXT, 300)
  end

  if wait then
    suilib.call('SetTimer', TIMER_ID_QUIT, wait * 1000)
  end
end
```

```
function ontimer(id)
    if id == TIMER_ID_QUIT then
      sui:close()
    elseif id == TIMER_ID_FLUSH_TEXT then
      lshift_text()
    end
end

function ondisplaychanged()
  move_top()
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_LED\main.lua"
```

```
dofile(sui:info('uipath') .. 'lunar.lua')

local function get_month_day(year, mon)
  local days = {31,28,31,30,31,30,31,31,30,31,30,31}
  if mon == 2 and year % 4 == 0 then
    return 29
  else
    return days[mon]
  end
end

local function fillnum(num, c)
    if c == nil then c = '0' end
    return string.sub(c .. num, -2, -1)
end

function draw_cal(year, mon)
  -- draw title
  nav_mon.text = year .. '-' .. fillnum(mon)
  -- draw days
  local mdate_1 = os.time{year=year, month=mon, day=1, hour=0}
  local week = os.date('%w', mdate_1) -- type(week) => string
  local i, j
  week = math.tointeger(week) -- convert to integer
  suilib.print('week' .. week)
  for i = 1,week do
    d['1' .. i].text = ''
  end


  if last_day_elem ~= nil then
     --last_day_elem.style = 'day'
    last_day_elem.textcolor = normal_textcolor
    last_day_elem.bkimage = normal_bk
     last_day_elem = nil
  end

  local day_elem = nil
  i = week + 1;j = 1
  local md = get_month_day(year, mon)
  for k = 1,42 - week do
    if k <= md then
      d[j .. i].text = get_day_text(year, mon, k) -- fillnum(k, ' ')
      if k == cur_day then
        p("today:" .. k)
        day_elem = d[j .. i]
      end
    else
      d[j .. i].text = ''
    end
    i = i + 1
    if i > 7 then i = 1;j = j + 1 end
  end


  if year == cur_year and mon == cur_mon then
    --day_elem.style = 'today'
    day_elem.textcolor = today_textcolor
    day_elem.bkimage = today_bk
    last_day_elem = day_elem
  end

end

d = {}
local function get_elems()
  for i=1,6 do
    for j=1,7 do
      d[i ..j] = sui:find('d' .. i .. j)
    end
  end
  nav_mon = sui:find('nav_mon')
  now_year = sui:find('now_year')
  now_info = sui:find('now_info')
  now_time = sui:find('now_time')
  month_days_area = sui:find('month_days_area')
  local today_attr = sui:find('today_attr')
  today_textcolor = today_attr.textcolor
  p(today_textcolor)
  today_bk = today_attr.bkimage
  normal_textcolor = d['11'].textcolor
  normal_bk = d['11'].bkimage
end

local function var_init()
  today = os.time()
  cur_year = math.tointeger(os.date('%Y', today))
  cur_mon = math.tointeger(os.date('%m', today))
  cur_day = math.tointeger(os.date('%d', today))
  str_today = cur_year .. cur_mon .. cur_day
end

local function draw_now_info()
    now_info.text = string.format('%d-%d-%d, %%{info_w%s}',
        cur_year, cur_mon, cur_day, os.date('%w', today))
    now_time.text = os.date('%H:%M:%S', today)
end

local function set_res(name, id, file)
  file = file or 'systemcpl.dll'
  sui:find(name).text = str.format('#{@%s,%s}', file, id)
end

local function set_text(name, text)
  sui:find(name).text = text
end

local function systeminfo_init()


  -- hostname
  local computername =
Reg:Read([[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName]]
        , 'ComputerName')
  set_text('host_text', '<b>' .. computername .. '</b>')

  -- CPU
  local cpu_t = os.info('CPU')

  if cpu_t then -- need Administrator's right privilege to read the HKLM items
    set_text('cpu_text', cpu_t['name'])
  end

  -- RAM
  -- local mem_format = mui_str('systemcpl.dll', 45)
  local mem_format = suilib.call('resstr', '#{@systemcpl.dll,45}')

  local mem_t = os.info('MEM')
  local total_phys = str.format("%.2f GB", mem_t['total_gb'])

  local avail_phys = mem_t['total'] / 0x40000000 -- 1024*1024*1024
  avail_phys = str.format("%.2f GB", avail_phys)

  local now_avail = str.format("#{@shell32.dll,9307} %.2f GB", mem_t['avail_gb'])
  local mem_s = str.format(mem_format, total_phys, avail_phys)
  set_text('mem_text', mem_s .. '    ' .. now_avail)

  -- BIOS
  local bios_mf = Reg:Read([[HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\BIOS]],
"SystemManufacturer")
  local bios_pn = Reg:Read([[HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\BIOS]],
"SystemProductName")
  if bios_mf == nil then bios_mf = 'N/A' end
  if bios_pn == nil then bios_pn = 'N/A' end
  set_text('bios_text', bios_mf .. " - " .. bios_pn)

  -- FirmwareType
  set_text('firmwaretype_text', os.info('FirmwareType'))

end

local function elem_init()
  systeminfo_init()
end

function UIWindow.OnLoad()
  get_elems()
  var_init()
  elem_init()
  lunar_init()
  view_year = cur_year
  view_mon = cur_mon
  draw_now_info()
  last_day_elem = nil -- mark today
  draw_cal(cur_year, cur_mon)
  UI:KillTimer('clocktick_per_second')
  UI:SetTimer('clocktick_per_second', 1000)
  UIWindow.Inited = 1
end

function UIWindow.OnShow()
```

```
  UI:SetTimer('adjust_win_position', 200)
  UI:KillTimer('clocktick_per_second')
  UI:SetTimer('clocktick_per_second', 1000)
  brightness_slider = sui:find('brightness_slider')
  brightness_val = Screen:Get('brightness')
  brightness_slider.value = brightness_val
end

function UIWindow.OnHide()
  UI:KillTimer('clocktick_per_second')
end

UI.OnTimer['clocktick_per_second'] = function(tid)
  var_init()
  draw_now_info()
end

UI.OnTimer['adjust_win_position'] = function(tid)
  UI:KillTimer('adjust_win_position')
  local w, h = sui:info('wh')
  sui:moveto(Screen:GetX() - w, 0, w, Screen:GetY())
  print("adjust_win_position")
end

UI.OnClick['pre_mon'] = function()
  view_mon = view_mon - 1
  if view_mon == 0 then
    view_year = view_year - 1
    view_mon = 12
  end
  draw_cal(view_year, view_mon)
end

UI.OnClick['next_mon'] = function()
  view_mon = view_mon + 1
  if view_mon == 13 then
    view_year = view_year + 1
    view_mon = 1
  end
  draw_cal(view_year, view_mon)
end

UI.OnClick['now_time'] = function()
  App:Run("timedate.cpl")
end

UI.OnChanged['brightness_slider'] = function(val, ctrl)
  App:Print(ctrl .. ':' .. val)
  brightness_val = val
  UI:SetTimer('set_brightness_timer', 200)
end

UI.OnTimer['set_brightness_timer'] = function(tid)
  UI:KillTimer('set_brightness_timer')
  local val = math.floor(brightness_val / 10 + 0.5) * 10;
  Screen:Set('brightness', val)
end
```

```
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\main.lua"
```

```lua
function onclick(ctrl)
  if ctrl == "restartbtn" then
    System:Reboot()
  elseif ctrl == "shutdownbtn" then
    System:Shutdown()
  end
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\main.lua"
```

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Window size="850,650" sizebox="4,4,4,4" caption="0,0,0,32" mininfo="600,400">

<Style imagerectstyle="l,t,w,h" />
<Include source="scrollbar_ltwh.xml" />
<Style name="nav_item" height="40"
      selectedimage="color='#ff0078d7' dest='155,5,5,20'"
    hotimage="color='#FFA2D7FF'  dest='155,5,5,20'"
    align="left" textpadding="52,-10,0,0" />
<VerticalLayout bordersize="1" bordercolor="#FF1985DA">

    <!-- 左-导航, 右-内容 水平布局 -->
    <HorizontalLayout style="ct-bknav" padding="1,1,1,1" >

        <!-- 左侧 区域 -->
        <VerticalLayout name="nav" width="160" >
            <!-- 左上 logo 区域 -->
            <Label height="32" text="应用市场" padding="20,0,0,0" />
            <!--HorizontalLayout style="nav_item" padding="20,0,0,0" -->
                <!--Label text="   " font="sym16" width="32" /-->

            <!--/HorizontalLayout-->

            <Label height="32" showhtml="true" text="分类" padding="20,0,0,0" />

            <!-- 左侧 导航 项目 -->
            <VerticalLayout name="$TabList">

            </VerticalLayout>
                        <Label name="boot_platform" text="" textpadding="5,0,0,0" />

        </VerticalLayout>


        <!-- 右侧内容区域格式 -->
        <VerticalLayout padding="0,1,1,1" style="ct-bkcontent">

            <!-- 右侧顶部标题区域 -->
            <VerticalLayout height="32">
                <HorizontalLayout height="32">
                    <Label text="本地版本:240305" textpadding="5,0,0,0" />
                    <Control />
                    <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
                                        <Button name="set_button" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ff1985da'" width="48" height="32" />

                    <Button name="::minbtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ff1985da'" width="48" height="32" />
                    <Button name="::closebtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ffe81123'" width="48" height="32" />
                </HorizontalLayout>
            </VerticalLayout>

            <!-- 右侧 内容 主区域 -->
            <VerticalLayout name="$TabLayoutParent">

            </VerticalLayout>

                    <Combo visible="false" float="true" itemfont="crr" itemlinecolor="#00000000"
dropboxsize="0,450" itemshowhtml="true" itemfont="16" style="nb-combo" pos="0,580,100,0"
```

```
align="center" name="listwinver" wheel="false" height="32" width="580" padding="0,0,0,0"
textpadding="2,0,0,2"
                        vscrollbar="true" hscrollbar="true">



                    </Combo>


        </VerticalLayout>



    </HorizontalLayout>
</VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="320,300" caption="0,0,0,32" defaultfontcolor="#FFE8EAF6">

    <Style imagerectstyle="l,t,w,h" />

    <Style name="PowerBtn" value="height=&quot;40&quot; textcolor=&quot;#FF000000&quot; hottextcolor=&quot;#FFFFFF&quot; hotimage=&quot;color='#FFF48FB1'&quot;" />
    <VerticalLayout bkcolor="#FFF9F9F9">
        <HorizontalLayout height="32" bkcolor="#FFC2185B">
            <Label text="%{Power}" textpadding="5,0,0,0" />
            <Control />
            <Button name="::closebtn" bkimage="file='close.png'" width="16" height="16" padding="0,8,5,0" />
        </HorizontalLayout>
        <VerticalLayout height="150" bkcolor="#FFE91E63">
            <HorizontalLayout height="48" padding="0,20,0,0">
                <Control />
                <Button normalimage="power.png" height="48" width="48" enabled="false" />
                <Control />
            </HorizontalLayout>
            <Label text="%{CloseComputer}" font="24" align="center" textpadding="0,10,0,0" />
        </VerticalLayout>
        <VerticalLayout>
            <Button padding="0,20,0,0" name="shutdownbtn" text="%{Shutdown}" style="PowerBtn" />
            <Button name="restartbtn" text="%{Restart}" style="PowerBtn"/>
            <Button visible="false" text="%{Hibernate}" style="PowerBtn" enabled="false" />
            <Button visible="false" text="%{Standby}" style="PowerBtn" enabled="false" />
        </VerticalLayout>
    </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window caption="0,0,0,0" defaultfontcolor="#FFFFFFFF" size="360,80">
    <Font default="true" id="16" name="微软雅黑" shared="true" size="16" />
    <Font id="24" name="微软雅黑" shared="true" size="24" />

    <VerticalLayout bkcolor="#FF005A9E"  height="80">
        <HorizontalLayout>
            <Label name="$icon" padding="10,16,10,0" bkimage="info.png" height="48" width="48"  />
            <VerticalLayout padding="0,5,0,0" >
                <Label name="$title" />
                <Text name="$info" textcolor="#FF99C7EB" />
            </VerticalLayout>
            <Button name="::closebtn" bkimage="file='close.png'" hotimage="file='close_hot.png'"
width="24" height="24" padding="0,5,5,0" />
        </HorizontalLayout>
    </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="360,540" caption="0,0,0,32" >
    <Font shared="true" id="sym" size="24" name="Segoe MDL2 Assets" />
    <Style imagerectstyle="l,t,w,h" />

    <Style name="week" width="46" font="18b" />
    <Style name="day" width="46" align="center" showhtml="true" />
    <VerticalLayout style="ct-bk">
        <HorizontalLayout height="8">
            <control />
            <Button name="::closebtn" width="8" height="8" />
        </HorizontalLayout>

        <include source="SystemInfo.xml" />

        <Control height="1" />
        <!-- <Control height="1" style="ct-bkhori" /> -->

        <include source="Calendar10L.xml" />
        <Control height="1" style="ct-bkhori" />
        <include source="BrightController.xml" />

    </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="580,152" caption="0,0,0,32" defaultfontcolor="#FFE8EAF6"  layeredimage="bk.png"
layeredopacity="255" >

    <Style imagerectstyle="l,t,w,h" />

    <VerticalLayout bkcolor="#FEFFFFFF">
          <HorizontalLayout height="32" visible="false" >
              <Control />
              <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
              <Button name="::closebtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ff007ACC'" width="48" height="32" />
          </HorizontalLayout>
        <VerticalLayout height="120"  inset="1,1,1,1" >
            <Label name="label_info" text="<b>NOTICE:This is a notice text message.</b>" font="40"
showhtml="true" textcolor="#FFF8D714" textpadding="40,20,0,0" />
        </VerticalLayout>

    </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="940,600" caption="0,0,0,32" sizebox="6,6,6,6"  >
  <Style imagerectstyle="l,t,w,h" />
  <Include source="scrollbar_ltwh.xml" />
  <Style name="sec_icon" bkimage="file='sec.ico'" height="16" width="16" padding="15,4,5,0" />
  <VerticalLayout name="::winborder" style="ct-bd">
   <VerticalLayout name="::mainframe" style="ct-bk" padding="1,1,1,1">
    <HorizontalLayout name="::caption" height="32" style="ct-title">
      <Button bkimage="file='MYPC_16x16.ico'" width="16" height="16" padding="6,6,0,0" />
      <Label name="::title" text="System" style="ct-title-text" align="left" padding="5,-2,0,0" />
      <Button name="::minbtn" text="î¥‰" font="sym" style="ct-sysbutton" hotimage="color='#ff1985da'"
width="48" height="32" />
      <Button name="::closebtn" text="îœ‘" font="sym" style="ct-sysbutton" hotimage="color='#ffe81123'"
width="48" height="32" />
      <!-- <Button name="::closebtn" bkimage="file='close.png'" width="16" height="16" padding="0,8,8,0" /> --
>
    </HorizontalLayout>
    <HorizontalLayout name="::content" bordersize="2" style="ct-bdtitle">
      <VerticalLayout width="250" padding="2,2,0,2" inset="5,5,0,0" style="ct-nav">
        <Button name="cphome_btn" align="left" padding="36,5,0,10" />
        <VerticalLayout height="100" >
        <HorizontalLayout >
        <Label style="sec_icon" />
        <Button name="devmgmt_btn" text="Device Managament" align="left" />
        </HorizontalLayout>

        <HorizontalLayout>
        <Label style="sec_icon" />
        <Button name="rmt_btn" text="Remote Settings" align="left" />
        </HorizontalLayout>

        <HorizontalLayout>
        <Label style="sec_icon" />
        <Button name="protect_btn" text="Remote Settings" align="left" />
        </HorizontalLayout>

        <HorizontalLayout >
        <Label style="sec_icon" />
        <Button name="adv_btn" text="Advanced Settings" align="left" />
        </HorizontalLayout>
        </VerticalLayout>

        <Control />
        <Button name="more_label" align="left" padding="20,0,0,0" enabled="false" />
        <Button name="act_btn" align="left" padding="20,0,0,0" />
        <Button name="update_btn" align="left" padding="20,0,0,0" />
        <Button name="perf_btn" visible="false" padding="20,0,0,0" />
        <Label padding="0,0,0,10" />
      </VerticalLayout>
      <VerticalLayout inset="5,5,0,0" vscrollbar="true" padding="0,0,5,5" >
        <Label name="basic_label" font="16" style="ct-txtinfo" padding="0,0,0,5" />
        <VerticalLayout height="100">
         <HorizontalLayout height="20" >
          <Label width="140" name="edition_group" text="Windows Edition" />
          <Control height="1" padding="0,12,10,0" style="ct-bkhori" />
         </HorizontalLayout>
         <HorizontalLayout>
```

```
          <VerticalLayout padding="20,4,0,0" >
            <Label multiline="true" name="edition_edit1" />
            <Label multiline="true" name="edition_edit2" />
            <Label multiline="true" name="edition_edit3" />
          </VerticalLayout>
            <Label name="brand_img" width="340" height="72" padding="0,0,5,0" />
        </HorizontalLayout>
        </VerticalLayout>
        <VerticalLayout name="system_info" height="110">
            <HorizontalLayout height="20" >
            <Label name="system_group" width="100" />
            <Control height="1" padding="0,12,10,0" style="ct-bkhori" />
          </HorizontalLayout>
          <HorizontalLayout>
            <VerticalLayout width="240" padding="20,4,0,0" >
              <Label name="mfgr_label" />
              <Label name="model_label" />
              <Label name="cpu_label" />
              <Label name="mem_label" />
              <Label name="arch_label" />
            </VerticalLayout>
            <VerticalLayout padding="0,4,0,0">
              <Label name="mfgr_text" />
              <Label name="model_text" />
              <Label name="cpu_text" />
              <Label name="mem_text" />
              <Label name="arch_text" />
            </VerticalLayout>
            <Label name="oem_logo" width="120" height="120" />
          </HorizontalLayout>
        </VerticalLayout>
        <VerticalLayout height="130">
          <HorizontalLayout height="20">
            <Label name="host_group" width="420" />
            <Control height="1" padding="0,12,10,0" style="ct-bkhori" />
          </HorizontalLayout>
          <HorizontalLayout>
            <VerticalLayout width="240" padding="20,4,0,0" >
              <Label name="host_label" />
              <Label name="fqdn_label" />
              <Label name="desc_label" />
              <Label name="workgroup_label" />
            </VerticalLayout>
            <VerticalLayout padding="0,4,0,0">
              <Label name="host_text" />
              <Label name="fqdn_text" />
              <Label name="desc_text" />
              <Label name="workgroup_text" />
            </VerticalLayout>
            <VerticalLayout>
              <HorizontalLayout width="150" >
                <Label style="sec_icon" />
                <Button name="host_settings_btn" align="left" height="20" padding="0,2,0,0"
textcolor="#FF0066CC" />
                </HorizontalLayout>
            </VerticalLayout>
          </HorizontalLayout>
        </VerticalLayout>
        <VerticalLayout height="60">
         <HorizontalLayout height="20">
          <Label name="status_group" width="150" />
          <Control height="1" padding="-15,12,10,0" style="ct-bkhori" />
         </HorizontalLayout>
         <HorizontalLayout>
           <VerticalLayout width="100" padding="20,4,0,0" >
           <Label name="runtime_label" />
          </VerticalLayout>
          <VerticalLayout padding="0,4,0,0" >
           <Label name="runtime_text" />
          </VerticalLayout>
         </HorizontalLayout>
       </VerticalLayout>
      </VerticalLayout>
    </HorizontalLayout>
  </VerticalLayout>
 </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window caption="0,0,0,0" size="320,560">
   <Style imagerectstyle="l,t,w,h" />
   <Include source="scrollbar_ltwh.xml" />
   <!-- <Include source="themes\default.xml" /> -->
   <Font shared="true" id="sym_icon" size="16" name="Segoe MDL2 Assets" />
   <Font shared="true" id="sym32" size="32" name="Segoe MDL2 Assets" />
   <Default name="Button" height="30" />

   <Style name="icon-proxy" mouse="false" font="sym_icon" text="   " />

   <Style name="Settings" font="12" width="94" height="68" align="left" textpadding="5,35,0,0" />
   <VerticalLayout style="ct-bkdarker1">
      <Control height="10" />
      <VerticalLayout name="::pppoe_list" height="56">
         <Button float="true" style="ct-bkdarker1" style="ct-hotimage" bordersize="0" pos="0,0,320,40"
name="dialup" />
         <Label float="true" mouse="false" pos="2,4,36,40" text="䎱" font="sym32" />
         <Label float="true" mouse="false" pos="36,0,148,40" text="PPPoE" />
      <Control height="1" float="true" pos="8,48,312,1" style="ct-bkhori" />
      </VerticalLayout>
      <Combo name="::wlan_list" height="30" visible="false" style="ct-combo"
         textpadding="10,0,0,0" />
      <Label name="::wlan_errmsg" padding="10,0,0,-30" height="30" visible="false" />
      <List style="ct-list" name="::ssid_list" height="360" header="hidden" vscrollbar="true" >
      </List>
      <Control />
      <Button name="netsettings" padding="10,0,0,0" text="%{NetworkSettings}" align="left"
font="16u" style="ct-text-secondary-url" />
      <Label padding="10,-5,0,0" text="%{NetworkSettingsMsg}" align="left" font="12" style="ct-text-
secondary" />
      <HorizontalLayout height="70" padding="5,5,0,0" childpadding="5">
         <Option style="ct-option" style="Settings" name="::switchwlan" textpadding="5,5,0,0"
multiline="true" showhtml="true" text="<f $sym_icon>   </f><n><n>WLAN" selected="true" />
         <Option style="ct-option" style="Settings" name="::switchfirewall" textpadding="5,5,0,0"
multiline="true" showhtml="true" text="<f $sym_icon>   </f><n><n>%{Firewall}" selected="true" />
          <VerticalLayout width="94">
            <Button style="ct-option" style="Settings" name="proxysettings" text="%{Proxy}" />
            <Label style="icon-proxy" float="true" pos="6,2,30,30" />
         </VerticalLayout>
      </HorizontalLayout>
      <Button name="::closebtn" text="" width="10" height="5" comment="debug" />
   </VerticalLayout>
</Window>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Window size="360,470" caption="0,0,0,32" >
    <Font shared="true" id="sym" size="24" name="Segoe MDL2 Assets" />
    <Style imagerectstyle="l,t,w,h" />

    <Style name="week" width="46" font="18b" />
    <Style name="day" width="46" align="center" showhtml="true" />
    <VerticalLayout style="ct-bk">
        <HorizontalLayout height="8">
            <control />
            <Button name="::closebtn" width="8" height="8" />
        </HorizontalLayout>

        <Control height="1" />
        <!-- <Control height="1" style="ct-bkhori" /> -->

        <include source="Calendar10L.xml" />

        <VerticalLayout name="brightness_part" height="40">
            <Control height="1" style="ct-bkhori" />
            <include source="BrightnessController.xml" />
        </VerticalLayout>

    </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="1024,768">

  <Style imagerectstyle="l,t,w,h" />

  <!-- <VerticalLayout bkimage="file='bk.jpg' blur='20.0'"> -->
  <VerticalLayout name="full_win" bkcolor="#FF29363F" bkimage="file='bk2.jpg' blur='10.0'">
    <Control name="i18n_text" visible="false" />
    <VerticalLayout>
      <!-- 1/3 -->
      <Control />
      <!-- 2/3 -->
      <VerticalLayout name="userlist_frame" height="420" visible="false">
        <HorizontalLayout height="200">
          <Control />
          <Button name="User_SYSTEM" bkimage="file='SYSTEM_logo.png' fade='176' dest='16,16,128,128'"
width="160"
             text="SYSTEM" font="24" textcolor="#FFFFFFFF" textpadding="0,150,0,0" shortcut="S" />
              <Control width="50" />
          <Button name="User_Administrator" bkimage="file='ADMIN_logo.png' fade='176'
dest='16,16,128,128'" width="160"
              text="Administrator" font="24" textcolor="#FFFFFFFF" textpadding="0,150,0,0" shortcut="A" />
          <Control />
        </HorizontalLayout>
        <Control />
        <HorizontalLayout name="login_progress_frame" height="180" >
          <Text name="progress_text" text="" textcolor="#FFFFFFFF" align="center" padding="0,0,20,0" />
        </HorizontalLayout>
      </VerticalLayout>
      <HorizontalLayout name="pass_input_frame" height="420" visible="false">
        <Control />
        <VerticalLayout width="420">
          <HorizontalLayout height="75">
          <Control />
          <Label bkimage="windows.png" height="42" width="200" padding="0,10,0,0" />
          <Control />
          </HorizontalLayout>

          <VerticalLayout bkcolor="#FFFFFFFF" >
            <Label bkimage="ms_logo.png" height="36" width="160" padding="20,20,0,20" />

            <Label text="%{SignIn}" font="24" textcolor="#FF000000" padding="20,0,20,0" />

            <!-- <VerticalLayout  bordersize="2" bordercolor="#FF8FD6C3" height="54"  padding="5,5,5,5"  >
            <Edit text="Administrator" bkcolor="#FF29363F" textcolor="#FF000000" align="center"/>
            </VerticalLayout> -->

            <Edit name="username" text="Administrator" textcolor="#FF000000" bkcolor="#FFE8F0FE"
nativebkcolor="#FFE8F0FE" align="center" padding="20,20,20,0" />
            <Label height="1" bkcolor="#FF0067B8" padding="20,0,20,0" />

            <Edit name="password" textcolor="#FF000000" focusedimage="color='#FFFF000000'"
password="true" align="center" padding="20,5,20,0" />
            <Label height="1" bkcolor="#FF0067B8" padding="20,0,20,0" />

            <Text text="<a>%{ForgotPassword}</a>" showhtml="true" textcolor="#FF005DA6"
padding="20,20,20,20" />
```

```
            <Button name="logon" bordersize="3"  bordercolor="#FF005DA6" bkcolor="#FF005DA6"
hotbordercolor="#FF94C6EE" focusbordercolor="#FF94C6EE"
              text="%{Logon}" textcolor="#FFFFFFFF" align="center" height="40" padding="20,5,20,5" />
          </VerticalLayout>
        </VerticalLayout>
        <Control />
      </HorizontalLayout>
      <!-- 3/3 -->
      <VerticalLayout height="180" >
        <VerticalLayout height="120" padding="40,5,0,15">
          <VerticalLayout name="clock">
          <label name="now_time" font="40b" textcolor="#FFFFFFFF" />
          <label name="now_info" font="40b" textcolor="#FFFFFFFF" />
          </VerticalLayout>
        </VerticalLayout>
        <HorizontalLayout height="40" bkcolor="#10000000" padding="0,0,0,0" >
          <HorizontalLayout>
          <Button name="logoninfo" text="%{LogonInfo}" textcolor="#FFFFFFFF" align="left"
padding="40,0,10,0" />
          </HorizontalLayout>
          <HorizontalLayout width="120" >
            <Button name="restart" text="îœ¬" textcolor="#FFFFFFFF" font="128" width="40"
padding="0,0,10,0" />
            <Button name="shutdown" text="îŸ¨" textcolor="#FFFFFFFF" font="130" width="40"
padding="0,0,10,0" />
          </HorizontalLayout>
        </HorizontalLayout>
      </VerticalLayout>
    </VerticalLayout>
  </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Window size="800,600" mininfo="600,400" caption="0,0,0,32" sizebox="6,6,6,6">

<Style imagerectstyle="l,t,w,h" />
<Include source="scrollbar_ltwh.xml" />
<Style name="nav_item" height="40" selectedimage="color='#ff0078d7' dest='235,10,5,20'"
hotimage="color='#FFA2D7FF'  dest='235,10,5,20'" />
<VerticalLayout bordersize="0" borderround="4,4" bordercolor="#FF1985DA">
    <HorizontalLayout style="ct-bknav">
        <include source="parts/tab.nav.xml" />
        <VerticalLayout padding="0,0,0,0" style="ct-bkcontent">
            <VerticalLayout name="::caption" visible="true" height="32">
                <HorizontalLayout height="32">
                    <Control />
                    <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
                    <Button name="::minbtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ff1985da'" width="48" height="32" borderround="2,2" />
                    <Button name="::closebtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ffe81123'" width="48" height="32" borderround="2,2"  />
                </HorizontalLayout>
            </VerticalLayout>
            <TabLayout name="TabLayoutMain" selectedid="-1" topbordersize="1"
bordercolor="#ff000000" padding="2,2,2,2" >
                <VerticalLayout padding="20,0,0,0" >
                    <Control text="PlaceHolder_0" height="50" />
                </VerticalLayout>

                <include source="parts/page.display.xml" />
                <include source="parts/page.background.xml" />
                <include source="parts/page.colors.xml" />
                <include source="parts/page.folderoptions.xml" />
                <include source="parts/page.taskbar.xml" />

            </TabLayout>
        </VerticalLayout>
    </HorizontalLayout>
</VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="690,420" caption="0,0,0,32" sizebox="6,6,6,6" defaultfontcolor="#FFF1F1F1" >

    <Style imagerectstyle="l,t,w,h" />
    <Include source="scrollbar_ltwh.xml" />

    <Style name="lb-text" desc="show text on left-bottom" align="left" textpadding="5,75,0,0"  />

    <Style name="folder_node" height="36" width="360" folderattr="width=&quot;36&quot;
normalimage=&quot;file='folder.ico' dest='2,2,32,32'&quot;" />
    <Style name="item_node"  height="36" width="360" folderattr="visible=&quot;false&quot;"
itemattr="textpadding=&quot;36,0,0,0&quot;" />

    <VerticalLayout style="ct-bkpanel">
        <HorizontalLayout name="title" height="20">
            <Label _text="Launcher" textcolor="#FFF1F1F1" align="center" />
            <Button name="::closebtn" width="16" height="16" padding="0,2,2,0" />
        </HorizontalLayout>
        <HorizontalLayout name="panel">
            <VerticalLayout name="$left_panel" visible="true" width="340" inset="15,5,0,5" >
            <Treeview name="_app_tree" visible="false" height="50" vscrollbar="true"
              style="ct-bkpanel" itemhotbkcolor="#FF196AA8" itemselectedbkcolor="#FF196AA8" >
            </Treeview>

            </VerticalLayout>

            <Control width="10" />
            <VerticalLayout name="$right_panel" width="340" inset="0,5,15,5" >

            <Label text="Creation" height="20" />
            <HorizontalLayout childpadding="10,5,5,10">
                <HorizontalLayout width="100" height="100" >
                    <Button style="ct-cell" />
                    <Label name="label_weekname" text="" font="16" pos="0,15" align="center"
width="100" height="20" float="true" mouse="false" />
                    <Label name="label_date" text="" font="24" pos="0,35" align="center" width="100"
height="50" float="true" mouse="false" />
                </HorizontalLayout>
                <Button text="Mail" height="100" width="200" bkimage="file='mail.png' dest='74,24,48,48'"
style="lb-text" style="ct-cell" />
            </HorizontalLayout>

            <Label text="Entertainment" height="20" />
            <HorizontalLayout  childpadding="10,5,5,10" >
              <HorizontalLayout>
                <Button text="Xbox" width="100" height="100" style="ct-cell" bkcolor="#FF107C10"
style="lb-text" />
                <Label bkimage="file='xbox.png'" pos="0,0" width="100" height="100" float="true"
mouse="false" />
              </HorizontalLayout>
                <Button text="Movies & TV" width="100" height="100" bkimage="file='movies.png'"
style="ct-cell" style="lb-text" />
                <Button text="Photos" width="100" height="100" bkimage="file='photos.png'" style="ct-cell"
style="lb-text" />
            </HorizontalLayout>
            <Label text="Browse" height="20" />
            <HorizontalLayout  childpadding="10,5,5,10" >
                <HorizontalLayout>
                    <Button text="Visual Studio 2015" height="100" style="ct-cell" style="lb-text" />
                    <Label bkimage="file='vs.png'" pos="76,26" width="48" height="48" float="true"
```

```
mouse="false" />
                </HorizontalLayout>
                <Button name="notepad" text="Notepad" width="100" height="100"
bkimage="file='notepad.png' dest='34,34,32,32'" style="ct-cell" style="lb-text" />
            </HorizontalLayout>
            <HorizontalLayout  childpadding="10,5,5,10" >
                <Button text="Maps" width="100" height="100" bkimage="file='maps.png'" style="ct-cell"
style="lb-text" />
                <Button text="Weather" height="100" bkimage="file='weather.png'" style="ct-cell" style="lb-
text" />
            </HorizontalLayout>

            <!-- custom app sample area -->
            <VerticalLayout name="custom_app_area" visible="false" >
            <Label text="MyApp" height="20" />
            <HorizontalLayout childpadding="10,5,5,10">
                <Button name="app11" style="ct-cell" style="lb-text" />
                <Button name="app12" style="ct-cell" style="lb-text" />
                <Button name="app13" style="ct-cell" style="lb-text" />
            </HorizontalLayout>
            <HorizontalLayout childpadding="10,5,5,10">
                <Button name="app21" style="ct-cell" style="lb-text" />
                <Button name="app22" style="ct-cell" style="lb-text" />
                <Button name="app23" style="ct-cell" style="lb-text" />
            </HorizontalLayout>
            </VerticalLayout>
            <!-- custom app sample area -->

            </VerticalLayout>
        </HorizontalLayout>
    </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="400,330" caption="0,0,0,32" sizebox="4,4,4,4" defaultfontcolor="#FFE8EAF6">
    <Style imagerectstyle="l,t,w,h" />

    <VerticalLayout bkcolor="#FFE7E7E7" >
        <HorizontalLayout height="40" bkcolor="#FF264EAE" >
            <Label name="title" bkimage="file='snd_mute.ico'" width="32" height="32" padding="5,3,0,0" />
            <Control />
            <Button name="closebtn" bkimage="file='close.png'" width="16" height="16" padding="0,8,5,0" />
        </HorizontalLayout>
        <HorizontalLayout height="64" bkcolor="#FF2F65E3">
            <Label text="Volume" font="24" textcolor="#FFFFFFFF" textpadding="10,10,0,0" />
            <Control />
            <Option name="vol_btn" width="60" height="35" padding="0,20,10,0"
                bkimage="file='off.png'" selectedimage="file='on.png'" selected="false" />
        </HorizontalLayout>
        <VerticalLayout height="170" bkimage="color='#FFF0F0F0'" >
            <VerticalLayout   bkimage="file='kb_card.png' corner='24,20,16,16'"  >
            <Slider name="volume_slider" imm="true" min="0" max="100" value="0" padding="15,15,10,5"
height="12" thumbsize="12,12" thumbimage="file='thumb.png'" bkimage="file='sliderbk.png'"
foreimage="file='sliderfr.png'" />
            <HorizontalLayout height="124" >
            <Control />
            <Label name="vol_value" width="124" text="<b>0<b>" showhtml="true" font="24" align="center"
             textcolor="#FFFFFFFF" bkimage="file='vol_num.png'" />
            <Control />
            </HorizontalLayout>
            </VerticalLayout>
        </VerticalLayout>
         <Control  height="2" padding="5,5,5,0" bkcolor="#FFCCCCCC" />
        <HorizontalLayout>
            <Label height="32" width="32" padding="20,10,0,0" bkimage="file='more.png'" />
             <Button name="sndvol_btn" height="32" textcolor="#FF000000" padding="0,10,0,0"
                text="<b>More Volume Settings<b>" showhtml="true" align="center" />
        </HorizontalLayout>
        <Button name="::closebtn" width="5" height="5" padding="6,0,0,0" />
    </VerticalLayout>
</Window>
```

```
{
  "说明" : {
    "name":"字符串。指定UI名。必须参数。",
    "title":"字符串。指定窗口名(任务栏显示名称)，可指定本地化资源ID，可缺省(<name>)。",
    "entry":"字符串。指定入口文件。可缺省(main.xml)。",
    "singleton":"布尔值。表明窗口是否唯一，如果为true，当该类型窗口存在时激活该窗口而不是创建新窗口，多用于设置窗口。可缺省(false)。",
    "noshadow":"布尔值。表明窗口是否禁用阴影效果(Shadow.png)，可缺省(true)。",
    "position":"固定字符串(center - 屏幕中央|rightbottom - 桌面右下角)。窗口的启动位置。可缺省(center)。",
    "nobaricon":"布尔值。表明是否在任务栏显示图标。可缺省(false)。",
    "baricon":"字符串。任务栏图标资源。可缺省(main.ico)。",
    "customstyle":"布尔值。表明是否指定窗口风格。可缺省(false)。",
    "style":"长整数值。指定窗口风格，具体值请参考微软说明文档。可缺省(WS_OVERLAPPEDWINDOW)。",
    "exstyle":"长整数值。指定窗口扩展风格，具体值请参考微软说明文档。可缺省(WS_EX_WINDOWEDGE)。",
    "theme":"字符串。指定主题(themes\*.xml)。可缺省(default)。",
    "locale":"字符串。指定用于界面文字显示的本地语言资源。例如:zh-CN，en-US。可缺省(系统默认语言)。"
  },
  "name":"UI_Shutdown",
  "title":"%{Power}",
  "baricon":"power.ico"
}
{
  "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\main.zh-CN.jcfg",
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window caption="0,0,0,0" defaultfontcolor="#FFFFFFFF" size="360,80">
    <Font default="true" id="16" name="微软雅黑" shared="true" size="16" />
    <Font id="24" name="微软雅黑" shared="true" size="24" />

    <VerticalLayout bkcolor="#FF005A9E"  height="80">
        <HorizontalLayout>
            <Label name="$icon" padding="10,16,10,0" bkimage="info.png" height="48" width="48"  />
            <VerticalLayout padding="0,5,0,0" >
                <Label name="$title" />
                <Text name="$info" textcolor="#FF99C7EB" />
            </VerticalLayout>
            <Button name="::closebtn" bkimage="file='close.png'" hotimage="file='close_hot.png'"
width="24" height="24" padding="0,5,5,0" />
        </HorizontalLayout>
    </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window caption="0,0,0,0" defaultfontcolor="#FFFFFFFF" size="360,80">
    <Font default="true" id="16" name="Tahoma" shared="true" size="16" />
    <Font id="24" name="Tahoma" shared="true" size="24" />

    <VerticalLayout bkcolor="#FF005A9E"  height="80">
        <HorizontalLayout>
            <Label name="$icon" padding="10,16,10,0" bkimage="info.png" height="48" width="48"  />
            <VerticalLayout padding="0,5,0,0" >
                <Label name="$title" />
                <Text name="$info" textcolor="#FF99C7EB" />
            </VerticalLayout>
            <Button name="::closebtn" bkimage="file='close.png'" hotimage="file='close_hot.png'"
width="24" height="24" padding="0,5,5,0" />
        </HorizontalLayout>
    </VerticalLayout>
</Window>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="400,84" caption="0,0,0,32" sizebox="4,4,4,4">
    <Style imagerectstyle="l,t,w,h" />

    <VerticalLayout style="ct-bk" >
        <Label name="vol_name" height="32" padding="20,0,0,10" />
        <HorizontalLayout height="32">
            <VerticalLayout width="32" padding="20,0,0,0">
                <Label name="vol_state" float="true" pos="0,0,32,32" bkimage="file='snd_mute.ico'" height="32" />
                <Option name="vol_btn" float="true" pos="0,0,32,32" selected="false" />
            </VerticalLayout>
            <Slider name="volume_slider" imm="true" min="0" max="100" value="0" height="24" padding="5,4,0,0"
                thumbsize="8,24" thumbimage="file='thumb.png'"
                bkimage="file='sliderbk.png'" foreimage="file='sliderfr.png'" />
            <Button name="vol_value" width="40" text="<b>0</b>" showhtml="true" font="24" align="center"
padding="5,0,20,0" />
        </HorizontalLayout>

        <HorizontalLayout visible="false">
            <Button name="sndvol_settings" height="32" text="<b>#{@shell32.dll,22066}</b>" showhtml="true"
align="center" />
        </HorizontalLayout>
        <HorizontalLayout>
            <Button name="::closebtn" width="5" height="32" padding="5,0,0,0" />
        </HorizontalLayout>
    </VerticalLayout>
</Window>
```

```
TOGGLE = false

binded_app = {}

local function update_date_cell()
  local today = os.time()
  label_weekname.text = string.format('%%{info_w%s}', os.date('%w', today))
  label_date.text = string.format('%d', os.date('%d', today))
end

function onload()
  label_weekname = sui:find('label_weekname')
  label_date = sui:find('label_date')
  update_date_cell() -- update in onshow() event only, maybe good if updating with a TIMER
  local param = App.CmdLine
  suilib.print(param)
  if param and string.find(param, '-custom') then
    local sd = os.getenv("SystemDrive")
    bind_app('app11', 'MS Paint', sd .. '\\Windows\\System32\\mspaint.exe', '', '<APP>,0', 100, 100)
    bind_app('app12', 'MS Explorer', sd .. [[\Windows\explorer.exe]], '', '<APP>', 100, 100)
    bind_app('app13', 'Control', 'control.exe', '', sd .. '\\Windows\\System32\\shell32.dll,-22', 100, 100)
    bind_app('app21', 'Command Prompt', sd .. [[\Windows\System32\cmd.exe]], '', '<APP>', 200 + 10, 100)
    bind_app('app22', 'Regedit', 'regedit.exe', '', sd .. '\\Windows\\System32\\shell32.dll,-167', 100, 100)
    hide_cell('app23')
    local area = sui:find('custom_app_area')
    suilib.print(area.visible)
    area.visible = 1
  end
end

function onshow()
  if TOGGLE then
    sui:hide()
    TOGGLE = false
    return
  else
    TOGGLE = true
  end
  if label_weekname == nil then return end
  update_date_cell()
end

function onclick(ctrl)
  if ctrl == "firefox" then
    App:Run("C:\\Program Files (x86)\\Mozilla Firefox\\firefox.exe")
  elseif ctrl == "calc" then
    App:Run("calc.exe")
  elseif ctrl == "notepad" then
    App:Run("notepad.exe")
  elseif ctrl == "notepad_paramter" then
    App:Run("notepad.exe", "C:\\setup.log")
  elseif binded_app[ctrl] ~= nil then
    local app_info = binded_app[ctrl]
    App:Run(app_info[1], app_info[2])
  end
end
```

```
--
===================================================================================
================
-- bind_app('app11', 'MS Paint', 'C:\\Windows\\System32\\mspaint.exe', '', '<APP>,0', 100, 100)
function bind_app(name, text, cmd, param, icon, width, height)
  local app_info = {cmd, param} -- app_info[1] - command, app_info[2] --- paramaters
  binded_app[name] = app_info -- insert to binded_app table
  local app_cell = sui:find(name)
  if not app_cell then return end
  app_cell.width = width
  app_cell.height = height
  app_cell.text = text
  icon = icon:gsub('<APP>', cmd)
  app_cell.bkimage = string.format('file=\'%s\' dest=\'%d,%d,32,32\'', icon, ((width - 32) / 2), ((height - 32) / 2))
end

function  hide_cell(name)
  local cell = sui:find(name)
  if cell then cell.visible = 0 end
end--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Launcher\main-full.lua"
```

```
local md5 = {
  _VERSION     = "md5.lua 1.1.0",
  _DESCRIPTION = "MD5 computation in Lua (5.1-3, LuaJIT)",
  _URL         = "https://github.com/kikito/md5.lua",
  _LICENSE     = [[
    MIT LICENSE

    Copyright (c) 2013 Enrique GarcÃa Cota + Adam Baldwin + hanzao + Equi 4 Software

    Permission is hereby granted, free of charge, to any person obtaining a
    copy of this software and associated documentation files (the
    "Software"), to deal in the Software without restriction, including
    without limitation the rights to use, copy, modify, merge, publish,
    distribute, sublicense, and/or sell copies of the Software, and to
    permit persons to whom the Software is furnished to do so, subject to
    the following conditions:

    The above copyright notice and this permission notice shall be included
    in all copies or substantial portions of the Software.

    THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
    OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
    MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
    IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
    CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
    TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
    SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
  ]]
}

-- bit lib implementions

local char, byte, format, rep, sub =
  string.char, string.byte, string.format, string.rep, string.sub
local bit_or, bit_and, bit_not, bit_xor, bit_rshift, bit_lshift

local ok, bit = pcall(require, 'bit')
if ok then
  bit_or, bit_and, bit_not, bit_xor, bit_rshift, bit_lshift = bit.bor, bit.band, bit.bnot, bit.bxor, bit.rshift, bit.lshift
else
  ok, bit = pcall(require, 'bit32')

  if ok then

    bit_not = bit.bnot

    local tobit = function(n)
      return n <= 0x7fffffff and n or -(bit_not(n) + 1)
    end

    local normalize = function(f)
      return function(a,b) return tobit(f(tobit(a), tobit(b))) end
    end

    bit_or, bit_and, bit_xor = normalize(bit.bor), normalize(bit.band), normalize(bit.bxor)
    bit_rshift, bit_lshift = normalize(bit.rshift), normalize(bit.lshift)
```

```
else

  local function tbl2number(tbl)
    local result = 0
    local power = 1
    for i = 1, #tbl do
      result = result + tbl[i] * power
      power = power * 2
    end
    return result
  end

  local function expand(t1, t2)
    local big, small = t1, t2
    if(#big < #small) then
      big, small = small, big
    end
    -- expand small
    for i = #small + 1, #big do
      small[i] = 0
    end
  end

  local to_bits -- needs to be declared before bit_not

  bit_not = function(n)
    local tbl = to_bits(n)
    local size = math.max(#tbl, 32)
    for i = 1, size do
      if(tbl[i] == 1) then
        tbl[i] = 0
      else
        tbl[i] = 1
      end
    end
    return tbl2number(tbl)
  end

  -- defined as local above
  to_bits = function (n)
    if(n < 0) then
      -- negative
      return to_bits(bit_not(math.abs(n)) + 1)
    end
    -- to bits table
    local tbl = {}
    local cnt = 1
    local last
    while n > 0 do
      last      = n % 2
      tbl[cnt]  = last
      n         = (n-last)/2
      cnt       = cnt + 1
    end

    return tbl
  end
```

```
bit_or = function(m, n)
  local tbl_m = to_bits(m)
  local tbl_n = to_bits(n)
  expand(tbl_m, tbl_n)

  local tbl = {}
  for i = 1, #tbl_m do
    if(tbl_m[i]== 0 and tbl_n[i] == 0) then
      tbl[i] = 0
    else
      tbl[i] = 1
    end
  end

  return tbl2number(tbl)
end

bit_and = function(m, n)
  local tbl_m = to_bits(m)
  local tbl_n = to_bits(n)
  expand(tbl_m, tbl_n)

  local tbl = {}
  for i = 1, #tbl_m do
    if(tbl_m[i]== 0 or tbl_n[i] == 0) then
      tbl[i] = 0
    else
      tbl[i] = 1
    end
  end

  return tbl2number(tbl)
end

bit_xor = function(m, n)
  local tbl_m = to_bits(m)
  local tbl_n = to_bits(n)
  expand(tbl_m, tbl_n)

  local tbl = {}
  for i = 1, #tbl_m do
    if(tbl_m[i] ~= tbl_n[i]) then
      tbl[i] = 1
    else
      tbl[i] = 0
    end
  end

  return tbl2number(tbl)
end

bit_rshift = function(n, bits)
  local high_bit = 0
  if(n < 0) then
    -- negative
    n = bit_not(math.abs(n)) + 1
    high_bit = 0x80000000
```

```
      end

      local floor = math.floor

      for i=1, bits do
        n = n/2
        n = bit_or(floor(n), high_bit)
      end
      return floor(n)
    end

    bit_lshift = function(n, bits)
      if(n < 0) then
        -- negative
        n = bit_not(math.abs(n)) + 1
      end

      for i=1, bits do
        n = n*2
      end
      return bit_and(n, 0xFFFFFFFF)
    end
  end
end

-- convert little-endian 32-bit int to a 4-char string
local function lei2str(i)
  local f=function (s) return char( bit_and( bit_rshift(i, s), 255)) end
  return f(0)..f(8)..f(16)..f(24)
end

-- convert raw string to big-endian int
local function str2bei(s)
  local v=0
  for i=1, #s do
    v = v * 256 + byte(s, i)
  end
  return v
end

-- convert raw string to little-endian int
local function str2lei(s)
  local v=0
  for i = #s,1,-1 do
    v = v*256 + byte(s, i)
  end
  return v
end

-- cut up a string in little-endian ints of given size
local function cut_le_str(s,...)
  local o, r = 1, {}
  local args = {...}
  for i=1, #args do
    table.insert(r, str2lei(sub(s, o, o + args[i] - 1)))
    o = o + args[i]
  end
  return r
end

local swap = function (w) return str2bei(lei2str(w)) end

-- An MD5 mplementation in Lua, requires bitlib (hacked to use LuaBit from above, ugh)
-- 10/02/2001 jcw@equi4.com

local CONSTS = {
  0xd76aa478, 0xe8c7b756, 0x242070db, 0xc1bdceee,
  0xf57c0faf, 0x4787c62a, 0xa8304613, 0xfd469501,
  0x698098d8, 0x8b44f7af, 0xffff5bb1, 0x895cd7be,
  0x6b901122, 0xfd987193, 0xa679438e, 0x49b40821,
  0xf61e2562, 0xc040b340, 0x265e5a51, 0xe9b6c7aa,
  0xd62f105d, 0x02441453, 0xd8a1e681, 0xe7d3fbc8,
  0x21e1cde6, 0xc33707d6, 0xf4d50d87, 0x455a14ed,
  0xa9e3e905, 0xfcefa3f8, 0x676f02d9, 0x8d2a4c8a,
  0xfffa3942, 0x8771f681, 0x6d9d6122, 0xfde5380c,
  0xa4beea44, 0x4bdecfa9, 0xf6bb4b60, 0xbebfbc70,
  0x289b7ec6, 0xeaa127fa, 0xd4ef3085, 0x04881d05,
  0xd9d4d039, 0xe6db99e5, 0x1fa27cf8, 0xc4ac5665,
  0xf4292244, 0x432aff97, 0xab9423a7, 0xfc93a039,
  0x655b59c3, 0x8f0ccc92, 0xffeff47d, 0x85845dd1,
  0x6fa87e4f, 0xfe2ce6e0, 0xa3014314, 0x4e0811a1,
  0xf7537e82, 0xbd3af235, 0x2ad7d2bb, 0xeb86d391,
  0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476
}

local f=function (x,y,z) return bit_or(bit_and(x,y),bit_and(-x-1,z)) end
local g=function (x,y,z) return bit_or(bit_and(x,z),bit_and(y,-z-1)) end
local h=function (x,y,z) return bit_xor(x,bit_xor(y,z)) end
local i=function (x,y,z) return bit_xor(y,bit_or(x,-z-1)) end
local z=function (ff,a,b,c,d,x,s,ac)
  a=bit_and(a+ff(b,c,d)+x+ac,0xFFFFFFFF)
  -- be *very* careful that left shift does not cause rounding!
  return bit_or(bit_lshift(bit_and(a,bit_rshift(0xFFFFFFFF,s)),s),bit_rshift(a,32-s))+b
end

local function transform(A,B,C,D,X)
  local a,b,c,d=A,B,C,D
  local t=CONSTS

  a=z(f,a,b,c,d,X[ 0], 7,t[ 1])
  d=z(f,d,a,b,c,X[ 1],12,t[ 2])
  c=z(f,c,d,a,b,X[ 2],17,t[ 3])
  b=z(f,b,c,d,a,X[ 3],22,t[ 4])
  a=z(f,a,b,c,d,X[ 4], 7,t[ 5])
  d=z(f,d,a,b,c,X[ 5],12,t[ 6])
  c=z(f,c,d,a,b,X[ 6],17,t[ 7])
  b=z(f,b,c,d,a,X[ 7],22,t[ 8])
  a=z(f,a,b,c,d,X[ 8], 7,t[ 9])
  d=z(f,d,a,b,c,X[ 9],12,t[10])
  c=z(f,c,d,a,b,X[10],17,t[11])
  b=z(f,b,c,d,a,X[11],22,t[12])
  a=z(f,a,b,c,d,X[12], 7,t[13])
  d=z(f,d,a,b,c,X[13],12,t[14])
  c=z(f,c,d,a,b,X[14],17,t[15])
  b=z(f,b,c,d,a,X[15],22,t[16])

  a=z(g,a,b,c,d,X[ 1], 5,t[17])
  d=z(g,d,a,b,c,X[ 6], 9,t[18])
  c=z(g,c,d,a,b,X[11],14,t[19])
  b=z(g,b,c,d,a,X[ 0],20,t[20])
  a=z(g,a,b,c,d,X[ 5], 5,t[21])
  d=z(g,d,a,b,c,X[10], 9,t[22])
  c=z(g,c,d,a,b,X[15],14,t[23])
  b=z(g,b,c,d,a,X[ 4],20,t[24])
  a=z(g,a,b,c,d,X[ 9], 5,t[25])
  d=z(g,d,a,b,c,X[14], 9,t[26])
  c=z(g,c,d,a,b,X[ 3],14,t[27])
  b=z(g,b,c,d,a,X[ 8],20,t[28])
  a=z(g,a,b,c,d,X[13], 5,t[29])
  d=z(g,d,a,b,c,X[ 2], 9,t[30])
  c=z(g,c,d,a,b,X[ 7],14,t[31])
  b=z(g,b,c,d,a,X[12],20,t[32])

  a=z(h,a,b,c,d,X[ 5], 4,t[33])
  d=z(h,d,a,b,c,X[ 8],11,t[34])
  c=z(h,c,d,a,b,X[11],16,t[35])
  b=z(h,b,c,d,a,X[14],23,t[36])
  a=z(h,a,b,c,d,X[ 1], 4,t[37])
  d=z(h,d,a,b,c,X[ 4],11,t[38])
  c=z(h,c,d,a,b,X[ 7],16,t[39])
  b=z(h,b,c,d,a,X[10],23,t[40])
  a=z(h,a,b,c,d,X[13], 4,t[41])
  d=z(h,d,a,b,c,X[ 0],11,t[42])
  c=z(h,c,d,a,b,X[ 3],16,t[43])
  b=z(h,b,c,d,a,X[ 6],23,t[44])
  a=z(h,a,b,c,d,X[ 9], 4,t[45])
  d=z(h,d,a,b,c,X[12],11,t[46])
  c=z(h,c,d,a,b,X[15],16,t[47])
  b=z(h,b,c,d,a,X[ 2],23,t[48])

  a=z(i,a,b,c,d,X[ 0], 6,t[49])
  d=z(i,d,a,b,c,X[ 7],10,t[50])
  c=z(i,c,d,a,b,X[14],15,t[51])
  b=z(i,b,c,d,a,X[ 5],21,t[52])
  a=z(i,a,b,c,d,X[12], 6,t[53])
  d=z(i,d,a,b,c,X[ 3],10,t[54])
  c=z(i,c,d,a,b,X[10],15,t[55])
  b=z(i,b,c,d,a,X[ 1],21,t[56])
  a=z(i,a,b,c,d,X[ 8], 6,t[57])
  d=z(i,d,a,b,c,X[15],10,t[58])
  c=z(i,c,d,a,b,X[ 6],15,t[59])
  b=z(i,b,c,d,a,X[13],21,t[60])
  a=z(i,a,b,c,d,X[ 4], 6,t[61])
  d=z(i,d,a,b,c,X[11],10,t[62])
  c=z(i,c,d,a,b,X[ 2],15,t[63])
  b=z(i,b,c,d,a,X[ 9],21,t[64])

  return bit_and(A+a,0xFFFFFFFF),bit_and(B+b,0xFFFFFFFF),
         bit_and(C+c,0xFFFFFFFF),bit_and(D+d,0xFFFFFFFF)
end
```

```
------------------------------------------------------------------

local function md5_update(self, s)
  self.pos = self.pos + #s
  s = self.buf .. s
  for ii = 1, #s - 63, 64 do
    local X = cut_le_str(sub(s,ii,ii+63),4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4)
    assert(#X == 16)
    X[0] = table.remove(X,1) -- zero based!
    self.a,self.b,self.c,self.d = transform(self.a,self.b,self.c,self.d,X)
  end
  self.buf = sub(s, math.floor(#s/64)*64 + 1, #s)
  return self
end

local function md5_finish(self)
  local msgLen = self.pos
  local padLen = 56 - msgLen % 64

  if msgLen % 64 > 56 then padLen = padLen + 64 end

  if padLen == 0 then padLen = 64 end

  local s = char(128) .. rep(char(0),padLen-1) .. lei2str(bit_and(8*msgLen, 0xFFFFFFFF)) ..
lei2str(math.floor(msgLen/0x20000000))
  md5_update(self, s)

  assert(self.pos % 64 == 0)
  return lei2str(self.a) .. lei2str(self.b) .. lei2str(self.c) .. lei2str(self.d)
end

------------------------------------------------------------------

function md5.new()
  return { a = CONSTS[65], b = CONSTS[66], c = CONSTS[67], d = CONSTS[68],
           pos = 0,
           buf = '',
           update = md5_update,
           finish = md5_finish }
end

function md5.tohex(s)
  return format("%08x%08x%08x%08x", str2bei(sub(s, 1, 4)), str2bei(sub(s, 5, 8)), str2bei(sub(s, 9, 12)),
str2bei(sub(s, 13, 16)))
end

function md5.sum(s)
  return md5.new():update(s):finish()
end

function md5.sumhexa(s)
  return md5.tohex(md5.sum(s))
end

return md5
--"ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\md5.lua"
```

```
fixed_out = {}
table.insert(
		fixed_out,
		string.format("<i ..\\UI_DL\\app_ico\\vlc播放器.png><a vlc><c #000000>VLC 超级播放器</c></a> \n一款开源的播放器应用\n")
	)
table.insert(
		fixed_out,
		string.format("<i ..\\UI_DL\\app_ico\\potplayer.png><a potplayer><c #000000>Potplayer 播放器</c></a> \n一款支持多种字幕的播放器应用\n")
	)
table.insert(
		fixed_out,
		string.format("<i ..\\UI_DL\\app_ico\\kodi.png><a kodi><c #000000>Kodi 播放器</c></a> \nKodi媒体中心是一个屡获殊荣的自由和开源的跨平台媒体播放器和用于HTPC\n")
	)

--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\media.lua"
```

```
local tab_list = {}
local tab_itemXML = '<VerticalLayout><Label name="$Nav[%d]" text="%s" /></VerticalLayout>'
local tab_item_regex = nil

local tablayout_head = [[<TabLayout name="$TabLayoutMain" selectedid="-1">]]
local tablayout_itemXML = '<VerticalLayout></VerticalLayout>'
local tablayout_end = '</TabLayout>'
local tablayout_item_regex = nil
local tab_layout = nil

local nav_map = {}
local navUI_item = {}

Nav = {}

function Nav:Init()

  -- init UI tab options
  app:print('==========Nav:Init()============')
  nav_list = sui:find('$TabList')
  local xml = suilib.genItemXML(tab_itemXML, tab_list, tab_item_regex)
  suilib.insertItem(nav_list, xml)

  local tablayout_parent = sui:find('$TabLayoutParent')

  xml = suilib.genItemXML(tablayout_itemXML, tab_list, tablayout_item_regex)
  suilib.insertItem(tablayout_parent, xml, tablayout_head, tablayout_end)

  tab_layout = sui:find('$TabLayoutMain')
  Nav:GetTabMap()
end

function Nav:SetTabList(list)
  tab_list = list
end

function Nav:SetTabItemXML(template, item_regex)
  tab_itemXML = template
  tab_item_regex = item_regex
end

function Nav:SetTabLayoutXML(_head, _end)
  tablayout_head = _head
  tablayout_end = _end
end

function Nav:SetTabLayoutItemXML(template, item_regex)
  tablayout_itemXML = template
  tablayout_item_regex = item_regex
end

function Nav:GetTabMap()
  local tab_id = -1
  local tab = #tab_list

  for i = 1, #tab_list do
    nav_map['$Nav[' .. i .. ']'] = tab_list[i]
    navUI_item[i] = sui:find('$Nav[' .. i .. ']')
  end
  return nav_map
end

function Nav:SwitchTab(id)
  TabPage:InitPage(id, tab_list[tonumber(id)])
  tab_layout.selectedid = tonumber(id)
end

function Nav:OnTabClick(ctrl)
  local handled = true
  local nav_id = ctrl:match('$Nav%[(.+)%]')
  if nav_id then
    Nav:SwitchTab(nav_id)
  else

    handled = false
  end
  return handled
end
dofile(UI_Path .. '_navData.lua')
--"ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nav.lua"
```

```
--LINK([[%StartMenu%\扇区小工具BOOTICE.lnk]],[[%ProgramFiles%\OTHERS\BOOTICE.EXE]])
--LINK([[%StartMenu%\文件快搜.lnk]],[[%ProgramFiles%\EVERYTHING\EVERYTHING.EXE]])
--LINK([[%StartMenu%\分区工具DiskGenius.lnk]],[[%ProgramFiles%\DiskGenius\DiskGenius.exe]])
exec('/show','X:\\Program Files\\WinXShell.exe -ui -jcfg wxsUI\\UI_AppStore\\loading.jcfg -
wait 8 -text "正在载入,请稍候..."')
exec('/hide /wait','X:\\Program Files\\wxsUI\\UI_AppStore\\AppStoreUP.bat')
exec('/show','X:\\Program Files\\WinXShell.exe -ui -jcfg wxsUI\\UI_AppStore\\main.jcfg')
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbappstore.lua"
```

```
local SystemRoot = os.getenv("SystemRoot")


local letters = winapi.get_logical_drives()
drv = {}


  for i, l in ipairs(letters) do


        if File.exists(l .. [[windows\system32\winlogon.exe]]) and not File.exists(l ..
[[Windows\System32\drivers\fbwf.sys]]) then
         exec("/wait /show",
            [[mkdir ]] .. l .. [[client]])
         exec("/wait /show",
            [[cmd /c xcopy X:\petools\Client\*.* /s /e /y ]] .. l .. [[client\.]])
  exec('/hide /wait', 'reg load hklm\\offline_system ' .. l ..
'\\Windows\\system32\\Config\\software')
 exec('/hide /wait', 'reg add "HKLM\\offline_system\\Microsoft\\Windows\\CurrentVersion\\Run"
/f /v "SEBarClt" /t REG_SZ /d "C:\\Client\\SEBarClt.exe"')
 exec('/wait', 'reg unload hklm\\offline_system')
 alert("注入客户端到".. l .. "完成")
end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_addsoft.lua"
```

```
{
  "name":"UI_AppStore",
  "lua":"nbtool_bootfire.lua",
  "entry":"nbtool_bootfire.xml",
  "nobaricon":true
}
{
  "ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_bootfire.jcfg",
}
```

```
APP_Path = app:info("path")
UI_Path = sui:info("uipath")
temp = os.getenv("temp")
systemdrive = os.getenv("systemdrive")
local exitcode, stdout = winapi.execute("mountvol")
bios_str = stdout:match("EFI ([^\n]+)\n")


function env_init()
      if File.Exists(UI_Path .. "\\bcd_sample.txt") then
        output = File.ReadAll(UI_Path .. "\\bcd_sample.txt")
    else
        exitcode, stdout = winapi.execute("cmd /c bcdedit /enum all")
                output = stdout:gsub("\r\n", "\n")

            end

end

env_init()

function File.ReadAll(path)
    local file = io.open(path, "r")
    if file == nil then
        return ""
    end
    local text = file:read("*a")
    file:close()
    return text
end
```

```
function get_fw()

-- 按空白行为分隔符提取多行字符
local fw_boot_order = {}
local fw_boot_menu = {}
for str in output:gmatch("(.-)\n\n") do
if str:find("displayorder") then
table.insert(fw_boot_order, str)
else
table.insert(fw_boot_menu, str.."\n")
end
end

-- 提取提示对象字符串
local alert_str = ""
for i, str in ipairs(fw_boot_menu) do
local obj_str = str:match("{(.-)}")
if obj_str then

local desc_str = str:match("description%s*(.-)\n")
local device_str = str:match("device%s*(.-)\n")
local default_str = str:match("{default}")
local current_str = str:match("{current}")
--alert_str = alert_str.."fw_boot_menu["..i.."]: {"..obj_str.."}\n" .. desc_str
if desc_str then
fw_boot_menu["{" .. obj_str .. "}"] = desc_str
fw_boot_menu[i] = {object = obj_str, description = desc_str, device = device_str}
end
if device_str then
fw_boot_menu["{" .. obj_str .. "}"] = desc_str .. "  系统"
end
```

```
if default_str then
fw_boot_menu["{" .. obj_str .. "}"] = desc_str .. " 系统[默认]"
end
if current_str then
fw_boot_menu["{" .. obj_str .. "}"] = desc_str .. " 系统[当前]"
end
end
end



-- 弹出提示框
-- loop over each line of the fw_boot_order table
local tbl_fw = fw_boot_order
local str = ""
for _,v in pairs(tbl_fw) do
    str = str.."\n"..v
end


for line in string.gmatch(str,"[^\r\n]+") do
   match_Str = string.match(line,"%b{}")
   matchdev_Str = string.match(line,"device%s*(.-)")


   if((fw_boot_menu[match_Str]) ~= nil) then
    guid_table.desc = "             " .. fw_boot_menu[match_Str]
        makexml()
   end
   end
```

```
end



function makexml()
guid_table.id = string.format("%s", match_Str)
        fwboot_item = fwboot_item + 1
        guid_table.item =  fwboot_item .. "   "

        table.insert(sel_item, string.format("%s", guid_table.id))

        table.insert(sel_item_name, string.format("%s", guid_table.desc))

        build_xml(guid_table.item, guid_table.desc)
  suilib.insertItem(listxml, xml)
 end

function suilib.insertItem(sui_obj, xml)
    local str = {}
    table.insert(str, '<ListContainerElement height="40"> ')
    table.insert(str, xml)
    table.insert(str, "   </ListContainerElement> ")
    sui_obj:add(table.concat(str, "\r\n"))
end

function build_xml(index, item_name)
    xml =
        [[
<ListContainerElement height="40">

        <HorizontalLayout>
```

```
            <Label text="]] ..
        guid_table.item ..
            [[" textpadding="15,0,0,0" align="center" height="40" width="60"
textcolor="#FF5E5E5E" disabledtextcolor="#FFA7A6AA" />
          <Control width="1" bkcolor="#FFCACACA"/>

         <Label text="]] ..
                item_name ..
                    [[" textpadding="-35,0,0,0" valign="left"  height="40" width="600"
textcolor="#FF5E5E5E" disabledtextcolor="#FFA7A6AA" />



        </HorizontalLayout>
                    </ListContainerElement>


]]
    return xml
end



function startinit()
UI:KillTimer("重写列表")
    fw_boot_order = ""
        fw_boot_menu = ""
    fwboot_item = -1
        msboot_item = 0
    guid_table = {}
        ms_stdout_tb = {}
    sel_item = {}
        sel_item_name = {}
        sel_item_type = {}
        env_init()
        get_fw()
    UIWindow.Inited = 1



            --alert(fwbootmgr_menu)
end

function UIWindow.OnLoad()
    listxml = sui:find("¥TabLayoutParent")
    startinit()
    --Alert(fwbootmgr_menu)
    UIWindow.Inited = 1
end

UI.OnClick["item_last"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    --alert(selected)
        if sel_item_name[fwindex]:find("系统") then
        exec("/wait /hide", [[Bcdedit /set {bootmgr} displayorder ]] .. selected .. [[ /addlast]])
        else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} displayorder ]] .. selected .. [[ /addlast]])
     end
    UI:SetTimer("重写列表", 10)
end
```

```
UI.OnTimer["重写列表"] = function(id)
    listxml.list = "" --清空xml
    fwbootmgr_menu = ""
        fw_boot_menu = ""

    startinit()
end

UI.OnClick["item_first"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
        if sel_item_name[fwindex]:find("系统") then
         exec("/wait /hide", [[Bcdedit /set {bootmgr} displayorder ]] .. selected .. [[
/addfirst]])
                else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} displayorder ]] .. selected .. [[
/addfirst]])
        end

    UI:SetTimer("重写列表", 10)
end


UI.OnClick["item_add"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    a=winapi.show_message('添加:警告', "暂不支持这功能", 'yes-no')
        if a == "yes" then
--winapi.show_message('警告', a)
```

```
   return
else
return
end
    UI:SetTimer("重写列表", 10)
end



UI.OnClick["item_del"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    a=winapi.show_message('警告', "你在删除\n项目:" .. sel_item_name[fwindex] .. "!\nGUID:" .. selected ..
        "\n极可能会导致系统无法启动，请肾重选择", 'yes-no')
        if a == "yes" then
--winapi.show_message('警告', a)
    exec("/wait /hide", [[Bcdedit /delete ]] .. selected)
else
return
end
    UI:SetTimer("重写列表", 10)
end


UI.OnClick["item_def"] = function()
    fwindex = listxml.index + 1
    selected = sel_item[fwindex]
                if sel_item_name[fwindex]:find("系统") then
           exec("/wait /hide", [[Bcdedit /default ]] .. selected)
                else
    alert("该项不支持此操作!\n仅支持系统菜单!")
```

```
        end



    UI:SetTimer("重写列表", 10)
end






UI.OnClick["item_bootsequence"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    --alert(selected)

        if sel_item_name[fwindex]:find("系统") then
         exec("/wait /hide", [[Bcdedit /set {bootmgr} bootsequence ]] .. selected)
        else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} bootsequence ]] .. selected)
        end

          if File.Exists("X:\\windows\\system32\\boot\\winload.efi") then
          exec("/wait /hide", [[wpeutil reboot]])
        --alert("pe")
    else
           --alert("os")
            exec("/wait /hide", [[shutdown -r -t 8]])
        -- exec("/wait /hide", [[wpeutil reboot]])
    end

end
```

```lua
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\20230301\nbtool_bootfire.lua"
```

```
APP_Path = app:info("path")
UI_Path = sui:info("uipath")
temp = os.getenv("temp")
dofile(UI_Path .. 'helper_ntboot.lua') --helper_ntboot
systemdrive = os.getenv("systemdrive")



function env_init()
      if File.Exists(UI_Path .. "\\bcd_sample.txt") then
        output = File.ReadAll(UI_Path .. "\\bcd_sample.txt")
    else
        exitcode, stdout = winapi.execute("cmd /c bcdedit /enum all")
                output = stdout:gsub("\r\n", "\n")

            end

end

env_init()

function File.ReadAll(path)
    local file = io.open(path, "r")
    if file == nil then
        return ""
    end
    local text = file:read("*a")
    file:close()
    return text
end
```

```
function oneboot_reboot()
          if File.Exists("X:\\windows\\system32\\boot\\winload.efi") then
          exec("/wait /hide", [[wpeutil reboot]])
        --alert("pe")
    else
           --alert("os")
            exec("/wait /hide", [[shutdown -r -t 8]])
        -- exec("/wait /hide", [[wpeutil reboot]])
    end
end

function check_oneboot(desc_str,obj_str)
if get_option("-oneboot") ~= nil and desc_str:find(get_option("-oneboot")) then
 exec("/wait /hide", [[Bcdedit /set {fwbootmgr} bootsequence {]]  .. obj_str .. [[}]])
 --alert("找到一次性启动项" .. desc_str .. "\nid" .. obj_str)
 oneboot_reboot()
else
return
 end

end




function get_fw()

-- 按空白行为分隔符提取多行字符
local fw_boot_order = {}
local fw_boot_menu = {}
for str in output:gmatch("(.-)\n\n") do
if str:find("displayorder") then
```

```
table.insert(fw_boot_order, str)
else
table.insert(fw_boot_menu, str.."\n")
end
end

-- 提取提示对象字符串
local alert_str = ""
for i, str in ipairs(fw_boot_menu) do
local obj_str = str:match("{(.-)}")
if obj_str then

local desc_str = str:match("description%s*(.-)\n")
local device_str = str:match("device%s*(.-)\n")
local default_str = str:match("{default}")
local current_str = str:match("{current}")
--alert_str = alert_str.."fw_boot_menu["..i.."]: {"..obj_str.."}\n" .. desc_str
if desc_str then
fw_boot_menu["{" .. obj_str .. "}"] = desc_str
fw_boot_menu[i] = {object = obj_str, description = desc_str, device = device_str}
end
if device_str  and desc_str then
fw_boot_menu["{" .. obj_str .. "}"] = desc_str .. " 系统"
end
if default_str and desc_str then
fw_boot_menu["{" .. obj_str .. "}"] = desc_str .. " 系统[默认]"
end
if current_str and desc_str then
fw_boot_menu["{" .. obj_str .. "}"] = desc_str .. " 系统[当前]"
end
```

```
if has_option("-oneboot") then check_oneboot(desc_str,obj_str)
end



end
end




-- 弹出提示框
-- loop over each line of the fw_boot_order table
local tbl_fw = fw_boot_order
local str = ""
for _,v in pairs(tbl_fw) do
    str = str.."\n"..v
end


for line in string.gmatch(str,"[^\r\n]+") do
   match_Str = string.match(line,"%b{}")
   matchdev_Str = string.match(line,"device%s*(.-)")


   if((fw_boot_menu[match_Str]) ‾= nil) then
    guid_table.desc = "              " .. fw_boot_menu[match_Str]
        makexml()
   end
   end
```

```
end




function makexml()
guid_table.id = string.format("%s", match_Str)
        fwboot_item = fwboot_item + 1
        guid_table.item =  fwboot_item .. "   "

        table.insert(sel_item, string.format("%s", guid_table.id))

        table.insert(sel_item_name, string.format("%s", guid_table.desc))

        build_xml(guid_table.item, guid_table.desc)
  suilib.insertItem(listxml, xml)
 end

function suilib.insertItem(sui_obj, xml)
    local str = {}
    table.insert(str, '<ListContainerElement height="40"> ')
    table.insert(str, xml)
    table.insert(str, "   </ListContainerElement> ")
    sui_obj:add(table.concat(str, "\r\n"))
end

function build_xml(index, item_name)
    xml =
        [[
<ListContainerElement height="40">

        <HorizontalLayout>
```

```
            <Label text="]] ..
        guid_table.item ..
            [[" textpadding="15,0,0,0" align="center" height="40" width="60"
textcolor="#FF5E5E5E" disabledtextcolor="#FFA7A6AA" />
          <Control width="1" bkcolor="#FFCACACA"/>

         <Label text="]] ..
                item_name ..
                    [[" textpadding="-35,0,0,0" valign="left"  height="40" width="600"
textcolor="#FF5E5E5E" disabledtextcolor="#FFA7A6AA" />



        </HorizontalLayout>
                  </ListContainerElement>

]]
    return xml
end


function startinit()
UI:KillTimer("重写列表")
    fw_boot_order = ""
        fw_boot_menu = ""
    fwboot_item = -1
        msboot_item = 0
    guid_table = {}
        ms_stdout_tb = {}
    sel_item = {}
        sel_item_name = {}
        sel_item_type = {}
        env_init()
        get_fw()
    UIWindow.Inited = 1



            --alert(fwbootmgr_menu)
end

function UIWindow.OnLoad()
    listxml = sui:find("¥TabLayoutParent")
    startinit()
    --Alert(fwbootmgr_menu)
    UIWindow.Inited = 1
        --如果有参数

end




UI.OnClick["item_last"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    --alert(selected)
        if sel_item_name[fwindex]:find("系统") then
        exec("/wait /hide", [[Bcdedit /set {bootmgr} displayorder ]] .. selected .. [[ /addlast]])
        else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} displayorder ]] .. selected .. [[ /addlast]])
```

```
     end
    UI:SetTimer("重写列表", 10)
end

UI.OnTimer["重写列表"] = function(id)
    listxml.list = "" --清空xml
    fwbootmgr_menu = ""
        fw_boot_menu = ""

    startinit()
end

UI.OnClick["item_first"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
        if sel_item_name[fwindex]:find("系统") then
         exec("/wait /hide", [[Bcdedit /set {bootmgr} displayorder ]] .. selected .. [[
/addfirst]])
                else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} displayorder ]] .. selected .. [[
/addfirst]])
        end

    UI:SetTimer("重写列表", 10)
end


UI.OnClick["item_add"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
   addfile = "raw"
 Add_Boot_Option(addfile)

    UI:SetTimer("重写列表", 10)
end



UI.OnClick["item_del"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    a=winapi.show_message('警告', "你在删除\n项目:" .. sel_item_name[fwindex] .. "!\nGUID:"
.. selected ..
        "\n极可能会导致系统无法启动，请肾重选择", 'yes-no')
        if a == "yes" then
--winapi.show_message('警告', a)
    exec("/wait /hide", [[Bcdedit /delete ]] .. selected)
else
return
end
    UI:SetTimer("重写列表", 10)
end




UI.OnClick["item_uefi"] = function()
if boot_platform ~= "uefi" then alert("当前环境不支持，需要UEFI环境!") return end
      a=winapi.show_message('警告', "即将重启并进入UEFI固件设置[进入BIOS]", 'yes-no')
        if a == "yes" and File.Exists("X:\\windows\\system32\\shutdown.exe") then
--winapi.show_message('警告', a)
```

```
    exec("/wait /hide", [[shutdown -r /fw -t 0 ]])
        elseif a == "yes" and not File.Exists("X:\\windows\\system32\\shutdown.exe") then
--winapi.show_message('警告', a)
    exec(UI_Path .. "\\soft\\shutdown.exe -r /fw -t 0")
else
return
end
    UI:SetTimer("重写列表", 10)
end



UI.OnClick["item_def"] = function()
    fwindex = listxml.index + 1
    selected = sel_item[fwindex]
                if sel_item_name[fwindex]:find("系统") then
           exec("/wait /hide", [[Bcdedit /default ]] .. selected)
                else
    alert("该项不支持此操作!\n仅支持系统菜单!")
        end


    UI:SetTimer("重写列表", 10)
end






UI.OnClick["item_bootsequence"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    --alert(selected)


        if sel_item_name[fwindex]:find("系统") then
         exec("/wait /hide", [[Bcdedit /set {bootmgr} bootsequence ]] .. selected)
        else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} bootsequence ]] .. selected)
        end


          if File.Exists("X:\\windows\\system32\\boot\\winload.efi") then
          exec("/wait /hide", [[wpeutil reboot]])
        --alert("pe")
    else
           --alert("os")
             exec("/wait /hide", [[shutdown -r -t 8]])
        -- exec("/wait /hide", [[wpeutil reboot]])
    end


end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\nbtool_bootfire.lua"
```

```
-- 获取应用程序路径和 UI 路径
APP_Path = app:info("path")
UI_Path = sui:info("uipath")

-- 获取系统临时文件夹路径
temp = os.getenv("temp")

-- 加载 ntboot 辅助函数
dofile(UI_Path .. ’helper_ntboot.lua’) -- helper_ntboot

-- 获取系统盘符
systemdrive = os.getenv("systemdrive")

-- 初始化环境变量
function env_init()
  -- 如果存在示例 BCD 文件，则读取示例文件
  if File.Exists(UI_Path .. "\\bcd_sample.txt") then
    output = File.ReadAll(UI_Path .. "\\bcd_sample.txt")
  else
    -- 否则，使用 bcdedit 命令获取所有启动项信息
    exitcode, stdout = winapi.execute("cmd /c bcdedit /enum all")
    output = stdout:gsub("\r\n", "\n") -- 将换行符替换为 \n
  end
end

-- 调用初始化函数
env_init()

-- 读取文件内容
function File.ReadAll(path)
  local file = io.open(path, "r")
```

```lua
  if file == nil then return "" end
  local text = file:read("*a")
  file:close()
  return text
end

-- 一次性启动后重启
function oneboot_reboot()
  if File.Exists("X:\\windows\\system32\\boot\\winload.efi") then
    exec("/wait /hide", [[wpeutil reboot]]) -- 在 WinPE 环境下重启
  else
    exec("/wait /hide", [[shutdown -r -t 8]]) -- 在 Windows 环境下重启
  end
end

-- 检查是否为一次性启动项
function check_oneboot(desc_str, obj_str, menu_str)
if menu_str:find("系统") then
    exec("/wait /hide", [[Bcdedit /set {bootmgr} bootsequence {]] .. obj_str .. [[}]]) -- 设置启动顺序
  else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} bootsequence {]] .. obj_str .. [[}]]) -- 设置启动顺序
  end

 if  has_option("-r") then
oneboot_reboot()
end
end

-- 获取固件启动项信息
```

```
function get_fw()
  -- 按空白行为分隔符提取多行字符
  local fw_boot_order = {}
  local fw_boot_menu = {}
  for str in output:gmatch("(.-)\n\n") do
    if str:find("displayorder") then
      table.insert(fw_boot_order, str) -- 存储启动顺序信息
    else
      table.insert(fw_boot_menu, str .. "\n") -- 存储启动菜单信息
    end
  end

  -- 提取提示对象字符串
  local alert_str = ""
  for i, str in ipairs(fw_boot_menu) do
    local obj_str = str:match("{(.-)}") -- 获取启动项 GUID
    if obj_str then
      local desc_str = str:match("description%s*(.-)\n") -- 获取启动项描述
      local device_str = str:match("device%s*(.-)\n") -- 获取启动设备
      local default_str = str:match("{default}") -- 检查是否为默认启动项
      local current_str = str:match("{current}") -- 检查是否为当前启动项

      if desc_str then
        fw_boot_menu["{" .. obj_str .. "}"] = desc_str
        fw_boot_menu[i] = { object = obj_str, description = desc_str, device = device_str }
      end

      if device_str and desc_str then
        fw_boot_menu["{" .. obj_str .. "}"] = desc_str .. " 系统" -- 添加系统标识
      end
      if default_str and desc_str then
        fw_boot_menu["{" .. obj_str .. "}"] = desc_str .. "  系统[默认]" -- 添加默认标识
      end

      if current_str and desc_str then
        fw_boot_menu["{" .. obj_str .. "}"] = desc_str .. "  系统[当前]" -- 添加当前标识
      end

      if has_option("-oneboot") and get_option("-oneboot") ‾= nil then
          menu_str = fw_boot_menu["{" .. obj_str .. "}"]
          check_oneboot(desc_str, obj_str, menu_str) end -- 检查是否为一次性启动
    end
  end

  -- 弹出提示框
  -- 遍历启动顺序列表
  local tbl_fw = fw_boot_order
  local str = ""
  for _, v in pairs(tbl_fw) do
    str = str .. "\n" .. v
  end

  for line in string.gmatch(str, "[^\r\n]+") do
    match_Str = string.match(line, "%b{}") -- 获取启动项 GUID
    matchdev_Str = string.match(line, "device%s*(.-)") -- 获取启动设备

    if (fw_boot_menu[match_Str]) ‾= nil then
      guid_table.desc = "              " .. fw_boot_menu[match_Str] -- 获取启动项描述
      makexml() -- 生成 XML 代码
    end
  end
```

```
end


-- 生成 ListContainerElement 的 XML 代码
function makexml()
  guid_table.id = string.format("%s", match_Str)
  fwboot_item = fwboot_item + 1
  guid_table.item = fwboot_item .. "    "

  table.insert(sel_item, string.format("%s", guid_table.id)) -- 存储启动项 GUID
  table.insert(sel_item_name, string.format("%s", guid_table.desc)) -- 存储启动项描述

  build_xml(guid_table.item, guid_table.desc) -- 生成 XML 代码
  suilib.insertItem(listxml, xml) -- 添加到 UI 列表
end


-- 添加 ListContainerElement 到 UI 列表
function suilib.insertItem(sui_obj, xml)
  local str = {}
  table.insert(str, '<ListContainerElement height="40"> ')
  table.insert(str, xml)
  table.insert(str, "   </ListContainerElement> ")
  sui_obj:add(table.concat(str, "\r\n"))
end


-- 生成单个启动菜单项的 XML 代码
function build_xml(index, item_name)
  xml = [[
<ListContainerElement height="40">


        <HorizontalLayout>
            <Label text="]] .. index ..
        [[" textpadding="15,0,0,0" align="center" height="40" width="60"
textcolor="#FF5E5E5E" disabledtextcolor="#FFA7A6AA" />
          <Control width="1" bkcolor="#FFCACACA"/>

         <Label text="]] .. item_name ..
        [[" textpadding="-35,0,0,0" valign="left"  height="40" width="600"
textcolor="#FF5E5E5E" disabledtextcolor="#FFA7A6AA" />



        </HorizontalLayout>
                  </ListContainerElement>

]]
  return xml
end

-- 初始化启动菜单信息
function startinit()
  UI:KillTimer("重写列表") -- 停止定时器
  fw_boot_order = ""
  fw_boot_menu = ""
  fwboot_item = -1
  msboot_item = 0
  guid_table = {}
  ms_stdout_tb = {}
  sel_item = {}
  sel_item_name = {}
  sel_item_type = {}

  env_init() -- 初始化环境变量
  get_fw() -- 获取固件启动项信息
  UIWindow.Inited = 1 -- 设置窗口初始化标志
end

-- 窗口加载时调用初始化函数
function UIWindow.OnLoad()
  listxml = sui:find("¥TabLayoutParent") -- 获取 UI 列表对象
  startinit() -- 初始化启动菜单信息
  UIWindow.Inited = 1 -- 设置窗口初始化标志
end

-- 将选中的启动菜单项移动到最后
UI.OnClick["item_last"] = function()
  fwindex = listxml.index + 1
  selected = sel_item[fwindex]

  if sel_item_name[fwindex]:find("系统") then
    exec("/wait /hide", [[Bcdedit /set {bootmgr} displayorder ]] .. selected .. [[
/addlast]]) -- 移动系统启动项
  else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} displayorder ]] .. selected .. [[
/addlast]]) -- 移动固件启动项
  end

  UI:SetTimer("重写列表", 10) -- 刷新 UI 列表
end

-- 定时器事件，用于刷新 UI 列表
UI.OnTimer["重写列表"] = function(id)
  listxml.list = "" -- 清空 UI 列表
  fwbootmgr_menu = ""
  fw_boot_menu = ""
  startinit() -- 重新初始化启动菜单信息
end

-- 将选中的启动菜单项移动到最前
UI.OnClick["item_first"] = function()
  fwindex = listxml.index + 1
  selected = sel_item[fwindex]

  if sel_item_name[fwindex]:find("系统") then
    exec("/wait /hide", [[Bcdedit /set {bootmgr} displayorder ]] .. selected .. [[
/addfirst]]) -- 移动系统启动项
  else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} displayorder ]] .. selected .. [[
/addfirst]]) -- 移动固件启动项
  end

  UI:SetTimer("重写列表", 10) -- 刷新 UI 列表
end

-- 添加新的启动菜单项
UI.OnClick["item_add"] = function()
  fwindex = listxml.index + 1
  selected = sel_item[fwindex]
  addfile = "raw" -- 设置添加类型为任意文件
  Add_Boot_Option(addfile) -- 调用添加启动项函数

  UI:SetTimer("重写列表", 10) -- 刷新 UI 列表
end

-- 删除选中的启动菜单项
UI.OnClick["item_del"] = function()
```

```
  fwindex = listxml.index + 1
  selected = sel_item[fwindex]
  a = winapi.show_message(
    '警告',
    "你在删除\n项目:" .. sel_item_name[fwindex] .. "!\nGUID:" .. selected ..
      "\n极可能会导致系统无法启动，请肾重选择",
    'yes-no'
  )
  if a == "yes" then
    exec("/wait /hide", [[Bcdedit /delete ]] .. selected) -- 删除启动项
  else
    return
  end
  UI:SetTimer("重写列表", 10) -- 刷新 UI 列表
end

-- 进入 UEFI 固件设置
UI.OnClick["item_uefi"] = function()
  if boot_platform ~= "uefi" then
    alert("当前环境不支持，需要UEFI环境!")
    return
  end
  a = winapi.show_message('警告', "即将重启并进入UEFI固件设置[进入BIOS]", 'yes-no')
  if a == "yes" and File.Exists("X:\\windows\\system32\\shutdown.exe") then
    exec("/wait /hide", [[shutdown -r /fw -t 0 ]]) -- 在 WinPE 环境下重启并进入 UEFI
  elseif a == "yes" and not File.Exists("X:\\windows\\system32\\shutdown.exe") then
    exec(UI_Path .. "\\soft\\shutdown.exe -r /fw -t 0") -- 在 Windows 环境下重启并进入 UEFI
  else
    return
  end
  UI:SetTimer("重写列表", 10) -- 刷新 UI 列表
end
```

```
-- 设置默认启动项
UI.OnClick["item_def"] = function()
  fwindex = listxml.index + 1
  selected = sel_item[fwindex]
  if sel_item_name[fwindex]:find("系统") then
    exec("/wait /hide", [[Bcdedit /default ]] .. selected) -- 设置默认启动项
  else
    alert("该项不支持此操作!\n仅支持系统菜单!")
  end

  UI:SetTimer("重写列表", 10) -- 刷新 UI 列表
end

-- 设置启动顺序并立即启动
UI.OnClick["item_bootsequence"] = function()
  fwindex = listxml.index + 1
  selected = sel_item[fwindex]

  if sel_item_name[fwindex]:find("系统") then
    exec("/wait /hide", [[Bcdedit /set {bootmgr} bootsequence ]] .. selected) -- 设置系统启动顺序
  else
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} bootsequence ]] .. selected) -- 设置固件启动顺序
  end

  if File.Exists("X:\\windows\\system32\\boot\\winload.efi") then
    exec("/wait /hide", [[wpeutil reboot]]) -- 在 WinPE 环境下重启
  else
    exec("/wait /hide", [[shutdown -r -t 8]]) -- 在 Windows 环境下重启
    end
end--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_bootfire.lua"
```

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Window size="800,600" sizebox="4,4,4,4" caption="0,0,0,32" mininfo="600,400">

<Style imagerectstyle="l,t,w,h" />
<Include source="scrollbar_ltwh.xml" />

<VerticalLayout bordersize="1" bordercolor="#FF1985DA">

    <!-- 左-导航, 右-内容 水平布局 -->
    <HorizontalLayout style="ct-bknav" padding="1,1,1,1" >

        <!-- 左侧 区域 -->
        <VerticalLayout name="nav" width="1" >
            <!-- 左上 logo 区域 -->
            <Label height="32" text="应用基地" padding="20,0,0,0" />

            <Label height="32" showhtml="true" text="分类" padding="20,0,0,0" />

            <!-- 左侧 导航 项目 -->
            <VerticalLayout name="$TabList">

            </VerticalLayout>
        </VerticalLayout>


        <!-- 右侧内容区域格式 -->
        <VerticalLayout padding="0,1,1,1" style="ct-bkcontent">

            <!-- 右侧顶部标题区域 -->
            <VerticalLayout height="32"  bkcolor="#FFFFFFFF">
                <HorizontalLayout height="32">
                                 <text name="boot_platform" align="left" text="" showhtml="true"
textpadding="15,0,0,0" />
                    <Control />

                    <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
                    <Button name="::minbtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ff1985da'" width="48" height="32"/>
                    <Button name="::closebtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ffe81123'" width="48" height="32"/>
                </HorizontalLayout>
            </VerticalLayout>

            <!-- 右侧 内容 主区域 -->
            <VerticalLayout >



<List name="$TabLayoutParent" bkcolor="#FFFFFFFF" inset="0,0,0,0" bordersize="1"
bordercolor="#FFD7D7D7" itemshowhtml="true" vscrollbar="true" hscrollbar="true" itemalign="left"
itemaltbk="true" itemshowrowline="true" itemshowcolumnline="true" itemlinecolor="#FFD7D7D7">
            <ListHeader height="40" bkcolor="#FFE6ECF7" bordersize="1" bordercolor="#FFD7D7D7">
                <ListHeaderItem text="序号" width="50" align="left" textpadding="15,0,0,0" />
                <ListHeaderItem text="启动项"  width="300" align="center" textpadding="15,0,0,0" />
                                <button name= "item_def" text="默认"  width="50" align="left"
textpadding="15,0,0,0" />
                                 <button name= "item_add" text="添加"  width="50" align="left"
textpadding="15,0,0,0" />
                                 <button name= "item_del" text="删除"  width="50" align="left"
textpadding="15,0,0,0" />
                                 <button name= "item_uefi" text="固件设置"  width="80" align="left"
```

```
textpadding="15,0,0,0" />
                                <button name= "item_bootsequence" text="立即启动"  width="80" align="left"
textpadding="15,0,0,0" />

                               <button name= "item_first" text="置顶"  width="50" align="left"
textpadding="15,0,0,0" />
                               <button name= "item_last" text="垫底"  width="50" align="left"
textpadding="15,0,0,0" />

           </ListHeader>


</List>

                              <text name="bt_text" align="left" text="" showhtml="true"
bkcolor="#FFFFFFFF"  textpadding="0,0,0,0" />




           </VerticalLayout>

       </VerticalLayout>

    </HorizontalLayout>
</VerticalLayout>
</Window>
```

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Window size="800,600" sizebox="4,4,4,4" caption="0,0,0,32" mininfo="600,400">

<Style imagerectstyle="l,t,w,h" />
<Include source="scrollbar_ltwh.xml" />

<VerticalLayout bordersize="1" bordercolor="#FF1985DA">

    <!-- 左-导航, 右-内容 水平布局 -->
    <HorizontalLayout style="ct-bknav" padding="1,1,1,1" >

        <!-- 左侧 区域 -->
        <VerticalLayout name="nav" width="1" >
            <!-- 左上 logo 区域 -->
            <Label height="32" text="应用基地" padding="20,0,0,0" />

            <Label height="32" showhtml="true" text="分类" padding="20,0,0,0" />

            <!-- 左侧 导航 项目 -->
            <VerticalLayout name="$TabList">

            </VerticalLayout>
        </VerticalLayout>


        <!-- 右侧内容区域格式 -->
        <VerticalLayout padding="0,1,1,1" style="ct-bkcontent">

            <!-- 右侧顶部标题区域 -->
            <VerticalLayout height="32"  bkcolor="#FFFFFFFF">
                <HorizontalLayout height="32">
                    <Label name="boot_platform" text="" textpadding="5,0,0,0" />
                    <Control />
                    <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
                    <Button name="::minbtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ff1985da'" width="48" height="32"/>
                    <Button name="::closebtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ffe81123'" width="48" height="32"/>
                </HorizontalLayout>
            </VerticalLayout>

            <!-- 右侧 内容 主区域 -->
            <VerticalLayout >

<List name="$TabLayoutParent" bkcolor="#FFFFFFFF" inset="0,0,0,0" bordersize="1"
bordercolor="#FFD7D7D7" itemshowhtml="true" vscrollbar="true" hscrollbar="true" itemalign="left"
itemaltbk="true" itemshowrowline="true" itemshowcolumnline="true" itemlinecolor="#FFD7D7D7">
            <ListHeader height="40" bkcolor="#FFE6ECF7" bordersize="1" bordercolor="#FFD7D7D7">
                <ListHeaderItem text="序号" width="50" align="left" textpadding="15,0,0,0" />
                <ListHeaderItem text="启动项"  width="400" align="center" textpadding="15,0,0,0" />
                                <button name= "item_def" text="默认"  width="50" align="left"
textpadding="15,0,0,0" />
                                <button name= "item_add" text="添加"  width="50" align="left"
textpadding="15,0,0,0" />
                                <button name= "item_del" text="删除"  width="50" align="left"
textpadding="15,0,0,0" />
                                  <button name= "item_bootsequence" text="立即启动"  width="80" align="left"
textpadding="15,0,0,0" />
                                 <button name= "item_first" text="置顶"  width="50" align="left"
textpadding="15,0,0,0" />
```

```
                                 <button name= "item_last" text="垫底"  width="50" align="left"
textpadding="15,0,0,0" />

            </ListHeader>


</List>





            </VerticalLayout>

        </VerticalLayout>

    </HorizontalLayout>
</VerticalLayout>
</Window>
```

```
APP_Path = app:info("path")
UI_Path = sui:info("uipath")
temp = os.getenv("temp")
systemdrive = os.getenv("systemdrive")

function env_init()
    fwbootmgr_menu_table = {}
    if File.Exists(UI_Path .. "\\bcd_sample.txt") then
        stdout = File.ReadAll(UI_Path .. "\\bcd_sample.txt")
    else
        exitcode, stdout = winapi.execute("cmd /c bcdedit /enum all")
    end
    stdout = stdout:gsub("\r\n", "\n")
end

env_init()

function File.ReadAll(path)
    local file = io.open(path, "r")
    if file == nil then
        return ""
    end
    local text = file:read("*a")
    file:close()
    return text
end

-- 正则匹配获取多行
function string.gfind(stdout, patten)
    local i, j = 0, 0
    return function()
```

```
        i, j = string.find(stdout, patten, j + 1)
        if (i == nil) then -- end find
            return nil
        end
        return string.sub(stdout, i, j)
    end
end

-- 提取固定值
function bcdinit()
    -- 初始化一些东西
    for def_str in string.gfind(stdout, "标识符                    {default}*.-\n\n") do
        if def_str ~= nil then
            def_stdout = def_str
            def_stdout = def_stdout:gsub("\r\n", "\n")
        end
    end

    for bootmgr_str in string.gfind(stdout, "标识符                    {bootmgr}*.-\n\n") do
        if bootmgr_str ~= nil then
            bootmgr_stdout = bootmgr_str
            bootmgr_stdout = bootmgr_stdout:gsub("\r\n", "\n")
        end
    end

    for fwbootmgr_str in string.gfind(stdout, "{fwbootmgr}*.-\n\n") do
        fwbootmgr_menu = fwbootmgr_str
    end
end

function get_fwboot_description(fwboot_guid_regex, stdout, fwboot_guid)
```

```
    if guid_table.id:find("{bootmgr}") then
        desc_regex = bootmgr_stdout:match("{bootmgr}[^*]+\ndescription([^\n]+)\n")
    else
              for stdout_str in string.gfind(stdout, fwboot_guid_regex .. "*.-\n\n") do
         stdout = stdout_str
    end

         --exitcode, stdout = winapi.execute("cmd /c bcdedit /enum " .. fwboot_guid)
         desc_regex = "description([^\n]+)\n"
    end
    for desc_str in stdout:gmatch(desc_regex) do
        if desc_str ~= nil then
            guid_table.desc = desc_str
            table.insert(
                fwbootmgr_menu_table,
                string.format("UEFI序列%s \n GUID:%s \n 启动项%s\n", guid_table.item, guid_table.id, guid_table.desc)
            )
            table.insert(sel_item, string.format("%s", guid_table.id))
            build_xml(guid_table.item, guid_table.desc)
            suilib.insertItem(listxml, xml)
        end
    end
end

function get_bootmgr_def()
    if def_stdout ~= nil then
        bootmgr_def = def_stdout:match("{default}[^*]+\ndescription([^\n]+)\n")
    end

    if bootmgr_def ~= nil then
```

```
        guid_table.def = bootmgr_def
    else
        guid_table.def = "newbeepe"
    end
    if stdout:find("{19260817--6666--8888--f00d--caffee000000}") then
        NewBeePE_install = "       NewBeePE"
    else
        NewBeePE_install = "     NewBeePE"
    end
end

function get_bcdedit()
    fwboot_item = -1
    guid_table = {}
    sel_item = {}
    for token in string.gmatch(fwbootmgr_menu, " {([^\n]+)\n") do
        fwboot_item = fwboot_item + 1
        guid_table.item = fwboot_item
        fwboot_guid = string.format("{%s", token)
        guid_table.id = fwboot_guid
        fwboot_guid_regex = "" .. fwboot_guid
        if fwboot_guid:find("-") then

            fwboot_guid_regex = fwboot_guid:gsub("-", "--")
        end
        get_fwboot_description(fwboot_guid_regex, stdout, fwboot_guid)
    end
end

function suilib.insertItem(sui_obj, xml)
    local str = {}
    table.insert(str, '<ListContainerElement height="40"> ')
    table.insert(str, xml)
    table.insert(str, "   </ListContainerElement> ")
    sui_obj:add(table.concat(str, "\r\n"))
end


function build_xml(index, item_name)
    xml =
        [[
<ListContainerElement height="40">


        <HorizontalLayout>
            <Label text="]] ..
        guid_table.item ..
            [[" textpadding="15,0,0,0" align="center" height="40" width="60"
textcolor="#FF5E5E5E" disabledtextcolor="#FFA7A6AA" />
          <Control width="1" bkcolor="#FFCACACA"/>


         <Label text="]] ..
                 item_name ..
                     [[" textpadding="-35,0,0,0" valign="left"  height="40" width="600"
textcolor="#FF5E5E5E" disabledtextcolor="#FFA7A6AA" />




        </HorizontalLayout>
                   </ListContainerElement>


]]
    return xml
end
```

```
-- function onload()
--   UI_Inited = 0
--   UI_Inited = 1
-- end
function startinit()
    UI:KillTimer("重写列表")
    env_init()
    bcdinit()

    get_bcdedit()
    get_bootmgr_def()

    UIWindow.Inited = 1

    fwbootmgr_menu = table.concat(fwbootmgr_menu_table, "\n")
    fwbootmgr_menu =
        string.format(
        fwbootmgr_menu .. "\n本机离线系统状态:\n默认启动操作系统:" .. guid_table.def ..
"\nNewBeePE安装状态:" .. NewBeePE_install
    )
    --alert(fwbootmgr_menu)
end

function UIWindow.OnLoad()
    listxml = sui:find("¥TabLayoutParent")
    startinit()

    --Alert(fwbootmgr_menu)
    UIWindow.Inited = 1
end
```

```
UI.OnClick["item_last"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    --alert(selected)
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} displayorder ]] .. selected .. [[ /addlast]])

    UI:SetTimer("重写列表", 10)
end

UI.OnTimer["重写列表"] = function(id)
    listxml.list = "" --清空xml
    fwbootmgr_menu = ""
    startinit()
end

UI.OnClick["item_first"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    --alert(selected)
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} displayorder ]] .. selected .. [[ /addfirst]])

    UI:SetTimer("重写列表", 10)
end



UI.OnClick["item_add"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    a=winapi.show_message(′添加:警告′, "暂不支持这功能", ′yes-no′)
        if a == "yes" then
--winapi.show_message(′警告′, a)
   return
else
return
end
    UI:SetTimer("重写列表", 10)
end



UI.OnClick["item_del"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    a=winapi.show_message(′警告′, "将删除序列!\n " .. sel_item[fwindex] .. ",可能会导致系统无法启动，请肾重选择", ′yes-no′)
        if a == "yes" then
--winapi.show_message(′警告′, a)
    exec("/wait /hide", [[Bcdedit /delete ]] .. selected)
else
return
end
    UI:SetTimer("重写列表", 10)
end



UI.OnClick["item_bootsequence"] = function()
    fwindex = listxml.index + 1
    --alert(sel_item[fwindex])
    selected = sel_item[fwindex]
    --alert(selected)
    exec("/wait /hide", [[Bcdedit /set {fwbootmgr} bootsequence ]] .. selected)

          if File.Exists("X:\\windows\\system32\\boot\\winload.efi") then
          exec("/wait /hide", [[wpeutil reboot]])
        --alert("pe")
    else
           --alert("os")
            exec("/wait /hide", [[shutdown -r -t 8]])
        -- exec("/wait /hide", [[wpeutil reboot]])
    end

end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\nbtool_bootfire_bak.lua"
```

```
{
 "name":"UI_AppStore",
 "lua":"nbtool_ck.lua",
 "entry":"nbtool_ck.xml",
 "nobaricon":true
}
{
 "ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_ck.jcfg",
}
```

```
local SystemRoot = os.getenv("SystemRoot")
local TID_UPANLIST_CHANGED = 20000 + 1
local strDrivers = "请选择分区修复\n一键修复所有分区"
local letters = winapi.get_logical_drives()
drv = {}
i = 0
function str_drv(ck)
  for _, l in ipairs(letters) do
    strDrivers = strDrivers .. "\n" .. l
    i = i + 1
    if ck == "all" then
      drv[i] = l:gsub("\\", "")
      --winapi.show_message('即将修复', drv[i])
      exec("/wait /show", SystemRoot .. [[\system32\cmd.exe /c start "正在修复]] .. drv[i] .. [[" chkdsk /x ]] .. drv[i])
    end
  end
end

function onload()
  str_drv()
  UPanList = sui:find("UPanList_combo")
  strDrivers = strDrivers:gsub("\\", "")
  UPanList.list = strDrivers
  UPanList.index = 0
end

function onchanged(ctrl)
  if ctrl == "UPanList_combo" then
    suilib.call("SetTimer", TID_UPANLIST_CHANGED, 200)
  end
```

```
end


function ontimer(tid)
  if tid == TID_UPANLIST_CHANGED then
    suilib.call("KillTimer", tid)
    UPanList_combo_onchanged()
  end
end


function UPanList_combo_onchanged()
  if UPanList.index ~= 0 and UPanList.index ~= 1 then
    exec("/show", SystemRoot .. [[\system32\cmd.exe /k chkdsk /x ]] .. UPanList.text)
  elseif UPanList.index == 1 then
    ck = "all"
    str_drv(ck)
  end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_ck.lua"
```

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Window size="400,140" sizebox="4,4,4,4" caption="0,0,0,32" mininfo="400,140">

<Style imagerectstyle="l,t,w,h" />
<Include source="scrollbar_ltwh.xml" />
<Style name="nav_item" height="40"
      selectedimage="color='#ff0078d7' dest='155,5,5,20'"
    hotimage="color='#FFA2D7FF'  dest='155,5,5,20'"
    align="left" textpadding="52,-10,0,0" />
<VerticalLayout bordersize="1" bordercolor="#FF1985DA">

    <!-- 左-导航, 右-内容 水平布局 -->
    <HorizontalLayout style="ct-bknav" padding="1,1,1,1" >

        <!-- 左侧 区域 -->
        <VerticalLayout name="nav" width="1" >
            <!-- 左上 logo 区域 -->
            <Label height="0" text="应用市场" padding="20,0,0,0" />
            <HorizontalLayout style="nav_item" padding="20,0,0,0" >
                <Label text="   " font="sym16" width="32" />
                <Button text="%{Home}" align="left" name="home_button" />
            </HorizontalLayout>

            <Label height="0" showhtml="true" text="分类" padding="20,0,0,0" />

            <!-- 左侧 导航 项目 -->
            <VerticalLayout name="$TabList">

            </VerticalLayout>


        </VerticalLayout>


        <!-- 右侧内容区域格式 -->
        <VerticalLayout padding="0,1,1,1" style="ct-bkcontent" bkimage="nbtool_ck_bg.png">

            <!-- 右侧顶部标题区域 -->
            <VerticalLayout height="16">
                <HorizontalLayout height="16">
                    <Label text="NBTool" textpadding="5,0,0,0" />
                    <Control />
                    <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
                    <Button name="::minbtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ff1985da'" width="48" height="32" />
                    <Button name="::closebtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ffe81123'" width="48" height="32" />
                </HorizontalLayout>

            </VerticalLayout>

            <!-- 右侧 内容 主区域 -->
            <VerticalLayout name="$TabLayoutParent" height="80">
                <VerticalLayout>
                                        <Label align="center" text="Chkdsk分区修复工具" font="24" height="32"
showhtml="true" padding="0,5,0,0" />
                    <Combo align="center" name="UPanList_combo"
normalimage="file='nbtool_ck_bg_selected.png'" style="ct-combo" wheel="false" height="32"
width="200" padding="100,10,0,0" textpadding="2,0,0,2"
                        vscrollbar="true" hscrollbar="true">
                    </Combo>
  </VerticalLayout>

            </VerticalLayout>
                        <VerticalLayout padding="0,0,0,0" width="400" bordercolor="#FF1985DA" >


 </VerticalLayout>






        </VerticalLayout>

    </HorizontalLayout>
</VerticalLayout>
</Window>
```

```
function petoolspath(panfupath)
  if panfupath == "" then
   return
  else
   exec('/show /wait', [[cmd /c pnputil /export-driver * "]] .. panfupath .. [["]])
   winapi.show_message("导出本机所有第三方驱动", [[操作完成!]])
  end
  end



--winapi.show_message('警告', a)
   local panfupath = Dialog:BrowseFolder('导出本机所有第三方驱动\n!!!请自行新建一个文件夹',
0x0000 )
   petoolspath(panfupath)



--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_ex_drv.lua"
```

```
@echo off
mode con cols=36 lines=3
set tool_dir=%1
set tool_dir=%tool_dir:"=%
echo %tool_dir%\main.cmd
set mode=%2
if "%mode%" == "" set mode=lzx
set zexe="%temp%\appstore\bin\7zx64.exe"
set wimlib="%temp%\appstore\bin\wimlib64\wimlib-imagex.exe"
set winxshell="%temp%\appstore\winxshell.exe"
::如果是newbeepe环境，设置不一样的变量值
if exist X:\ipxefm call :setxvar
if exist %temp%\appstore\bin\wimlib64.7z start "" /w %zexe% x -y
%temp%\appstore\bin\wimlib64.7z -o%temp%\appstore\bin
for /f "tokens=2,*" %%a in (′reg query
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders" /v
desktop′) do set desk=%%b
if "%tool_dir%" == "" cls&&echo 错误!该命令必须有参数&&pause>nul&&exit
if "%tool_dir:¯0,2%" == "X:" cls&&echo 错误!不能在X盘生成nbp应用程序包!&&pause>nul&&exit
if not exist "%tool_dir%\main.cmd" call :mkmaincmd
:mknbp
set "nbname="
set "nbpath="
for %%a in ("%tool_dir%") do (
    set "nbname=%%¯nxa"
    set "nbpath=%%¯dpa"
)

echo nbname=%nbname%
echo nbpath=%nbpath%
```

```
start "" "%winxshell%" -ui -jcfg wxsUI\UI_led.zip -wait 2 -text 生成NBP应用程序包%appname%,{如果在X盘生成的包，是无效的!}
if not exist "%nbpath%\%nbname%\Windows\System32" mkdir "%nbpath%\%nbname%\Windows\System32"
echo [] >%tool_dir%\Windows\System32\WimBootCompress.ini
call :%mode%
start "" "%winxshell%" -ui -jcfg wxsUI\UI_led.zip -wait 3 -text 生成NBP应用程序包%appname%完成……
if not "%SystemDrive%" == "X:" explorer /select,\"%desk%\\%nbname%\"
exit

:lzx
%wimlib% capture "%nbpath%\%nbname%" "%desk%\%nbname%.wim.nbp" "NBTool" --boot --wimboot --compress=lzx --rebuild
exit /b

rem 下面是高压缩
:lzms
%wimlib% capture "%nbpath%\%nbname%" "%desk%\%nbname%.wim.nbp" "NBTool" --boot --wimboot --compress=lzms:100 --rebuild
exit /b

rem 下面是固实高压缩
:solid
%wimlib% capture "%nbpath%\%nbname%" "%desk%\%nbname%.wim.nbp" "NBTool" --boot --wimboot --solid --compress=lzms:100 --rebuild
exit /b



:mkmaincmd
mode con cols=58 lines=28
```

```
echo 没有main.cmd!继续操作的话请选择执行文件
echo main.cmd将自动编写，否则请关闭此对话框!
echo ...........选择执行文件................
setlocal enabledelayedexpansion
set "folder=%tool_dir%"
set "exts=exe cmd bat wcs lua"
set "i=0"

for %%a in (%exts%) do (
    for /r "%folder%" %%b in (*%%a) do (
        set /a i+=1
        set "file[!i!]=%%¯fb"
        set "path[!i!]=%%¯dpb"
        echo !i!.%%¯fb
    )
)
set /p "input=请选择一个文件："
set "apprun=!file[%input%]!"



for %%a in (%exts%) do (
    if "!apprun:¯-3!"=="%%a" set "appext=%%a"
)

echo 选中的文件名：%apprun%
echo 选中的文件扩展名：%appext%
echo "%folder%\main.cmd"
call set apprun=%%apprun:%folder%=%%
set "apprun=%apprun:¯1%"
echo %folder%
echo %apprun%
```

```
(
echo cd /d %%^¯dp0
echo start ^"^" ^"%apprun%^"
echo exit
)> "%folder%\main.cmd"
echo 自动编写启动脚本完成!
exit /b

:setxvar
set wimlib="X:\Program Files\GhostCGI\wimlib64\wimlib-imagex.exe"
set winxshell="X:\Program Files\WinXShell.exe"
set zexe="X:\Program Files\7-zip\7z.exe"
exit /b
```

```
function petoolspath(panfupath)
  if panfupath == "" then
   winapi.show_message("NBP应用程序制作", "取消")
  else
   exec('/show /wait', [[cmd /c start "" wxsUI\\UI_AppStore\nbtool_mknbp.bat ]] .. panfupath .. [[ solid]])
   --winapi.show_message("制作完成!", [[制作完成]])
  end
end

function check(a)
if a == "yes" then
--winapi.show_message('警告', a)
   local panfupath = Dialog:BrowseFolder('请选择要制作成NBP应用程序的目录.\n正常系统只能制作固实压缩[生成文件体积小]的自解压包.' ..
   '\n如要制作wimboot包[不释放运行省内存], 请在pe下制作', 17)
   petoolspath(panfupath)
else
return
end
end

check("yes")--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_mknbp.lua"
```

```
{
 "name":"UI_AppStore",
 "lua":"nbtool_nbdl.lua",
 "entry":"nbtool_nbdl.xml",
 "nobaricon":true
}
{
 "ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_nbdl.jcfg",
}
```

```
--sys_env
temp = os.getenv("temp")
label_downspd = sui:find("label_downspd")
local fixed_out = {}
local pid = winapi.get_current_pid()
local logfile = pid .. ".log"
--other_env
lshift_index = 0
char_width = 2
TIMER_ID_FLUSH_TEXT = 1001
TIMER_ID_QUIT = 1002
TIMER_ID_LOADING = 10088
TIMER_ID_CHECK_INSTALLED = 10089
eta_str = 0
dl_str = 0
app_runpath = "temp"
app_setup_args = "/S"
process_val = 0
--other
stdout = ""
file = ""
text = ""
file_type = ""
file_ext = ""
show_text = ""
aria2c_args = [[--file-allocation=falloc -x 16 -c --allow-overwrite=true]]


function check_option()
if user_agent ~= nil then
user_agent = app_user_agent
```

```
end
--winapi.show_message("当前软件下载地址：", "没得到详细参数")
if app_setup_args == nil then
app_setup_args = "/S"
end
if app_runpath == nil then
app_runpath = "temp"
end

if downpath == nil then
downpath = "%temp%"
end

if app_setup == nil then
app_setup = "setup.exe"
end
if user_agent == nil then
user_agent = [[--user-agent="Chrome/94.0.4606.71"]]
end
if aria2c_args == nil then
aria2c_args = [[--file-allocation=falloc -x 16 -c --allow-overwrite=true]]
end
if app_setup == nil then
app_setup = "setup.exe"
end
end




function require(name)
```

```
    if not package.loaded[name] then --模块是否已加载?
        local loader = findloader(name)
        if loader == nil then
            error("unable to load module"..name)
        end
        package.loaded[name] = true --将模块标记为已加载
        local res = loader(name)     --初始化模块
        if res ~=nil then
            package.loaded[name] = res
        end
    end
    return package.loaded[name]
end

function dofile (filename)
  local f = assert (loadfile (filename))
  return f ()
end -- dofile



function app_init()
process_slider = sui:find("process_slider")
label_info = sui:find("label_info")

--不安装的程序如果存在直接运行

if File.exists(downpath .. [[\]] .. app_setup) and app_runpath == "temp" or app_runpath ==
nil then
exec("/hide", [[cmd /c start "" ]] .. downpath .. [[\]] .. app_setup)
sui:close()
end
```

```
--local pid = winapi.temp_name()
--winapi.show_message("pid", pid)
if nbapp == "btwim" then
--winapi.show_message("pid", "系统镜像")
exec(
    "/hide",
    [[cmd /c aria2c --file-allocation=falloc -x32 -c -s 2 -j 10 --allow-overwrite=true ]] ..
aria2c_args .. [[ ]] .. user_agent .. [[ "]] ..
                app_url .. [[" -d I:\ -o ]] .. app_setup .. [[>%temp%\]] .. logfile
)
elseif app_url ~=nil or user_agent ~= nil and nbapp ~= "btwim" then
--napi.show_message("pid", "普通应用")
---以下是appstore白嫖的下载语句

exec(
    "/hide",
      [[cmd /c aria2c --allow-overwrite=true ]] ..
            aria2c_args ..
            [[ ]] .. user_agent .. [[ "]] ..
            app_url .. [[" -d ]] .. downpath .. [[ -o ]] .. app_setup .. [[>%temp%\]] ..
logfile
)

end

-- 正则匹配获取多行
    --exec("/hide", [[cmd /c del /q /f %temp%\]] .. app_setup)
    if File.exists([[wxsUI\UI_DL\app_ico\]] .. app_name .. [[.png]]) then
    label_info.text = [[下载到: ]] .. downpath  .. [[\]] .. app_setup
        else
        label_info.text = [[<i app_ico\nt6.png> 下载到: ]] .. downpath .. [[\]] .. app_setup
        end
        label_downspd.text = "初始化下载..."
    process_slider.value = 0
    suilib.call("SetTimer", TIMER_ID_CHECK_INSTALLED, 2000)
    suilib.call("SetTimer", TIMER_ID_LOADING, 1000)

end

function string.gfind(stdout, patten)
    local i, j = 0, 0
    return function()
        i, j = string.find(stdout, patten, j + 1)
        if (i == nil) then -- end find
            return nil
        end
        return string.sub(stdout, i, j)
    end
end

function check_app()
    process_slider = sui:find("process_slider")
            process_slider.value = 100
        process_val = 100
                suilib.call("KillTimer", 10088)
    if File.exists(downpath .. "\\" .. app_setup)  then

        label_downspd.text = "下载完毕! 正在安装... [" .. app_setup .. "]"
        exec("/hide", [[cmd /c start "" ]] .. downpath .. [[\]] .. app_setup)
       end
```

```
end


function check_installed()
    if File.exists(app_runpath) and app_runpath ~= "temp" and app_file == nil and nbapp ~=
"btwim" and nbapp ~= "btgho" then

            label_downspd.text = "安装完成，准备运行!"

        exec("/hide", [[cmd /c start "" "]] .. app_runpath .. [["]])
            suilib.call("KillTimer", 10088)
        Desktop:Refresh()
                exec("/hide", [[cmd /c del /q "]] .. downpath .. [[\]] .. app_setup .. [["]])
        sui:close()
        elseif File.exists(app_runpath) and app_runpath ~= "temp" and app_file ~= nil then

        label_downspd.text = "安装完成，准备运行!"
                --winapi.show_message("下载完成后播放文件:", app_file)
        suilib.call("KillTimer", 10088)
        Desktop:Refresh()
                exec("/hide", [[cmd /c del /q "%temp%\]] .. app_setup .. [["]])
                exec("/hide", [[cmd /c start "" "]] .. app_file .. [["]])
        sui:close()
                --如果是系统镜像则调用第三方安装器
        elseif File.exists(app_runpath) and not File.exists(app_runpath .. ".aria2") and
app_runpath ~= "temp" and nbapp == "btwim" then

            label_downspd.text = "系统镜像下载完成，准备运行!"
                --winapi.show_message("下载完成后播放文件:", app_file)
        suilib.call("KillTimer", 10088)
        Desktop:Refresh()
                exec("/show", [[cmd /k dism /Get-WimInfo /wimfile:"]] .. app_runpath)
        sui:close()
    end
end


function loading()
    local path = temp .. "\\" .. logfile
    local file = io.open(path, "r")
    if file == nil then
        label_downspd.text = "无下载，准备就绪!"
        process_slider.value = 0
    else
        local text = file:read("*a")
                file:close()
        local regex = "([^\n]+)[\n]*￥" -- 匹配最后一行有效文本 ￥表示匹配字符串结尾位置
                complete = text:match("Download complete: ([^] %s]+)")
        for all_str in string.gfind(text, "[[][#].-[]]") do
            table.insert(fixed_out, all_str)
        end
        down_str = table.concat(fixed_out, "\n\n")
        --winapi.show_message("最后一个", down_str)
        --文件判断语句备份
                        if complete ~= nil then
                        eta_str = "0S"
                label_downspd.text = "完成!"
                process_slider.value = "100"
                                process_val = "100"
                check_app()
            end
        for loading_str in string.gmatch(down_str, regex) do
                                dl_str = loading_str:match("DL:([^] %s]+)")
                                eta_str = loading_str:match("ETA:([^]]+)")
                                process_val = loading_str:match("[((]([^%%]+)")
                if process_val ~=nil and eta_str ~=nil and process_val < "95" then
               label_downspd.text = "已经完成:" .. process_val .. "% 速度:" .. dl_str .. "/S 剩余时间:" .. eta_str
               process_slider.value = process_val
                        end
        end
    end
    --local regex = "([^\n]+)¥"
end

function move_top()
    local w, h = sui:info("wh")
    if has_option("-top") then
        sui:move(((Screen:GetX() - w) // 2) + 2, -((Screen:GetY() - h) // 2) + 22, 0, 0)
    end
end

function onload()
 if has_option("-app_url") then
    app_url = get_option("-app_url")
        app_name = get_option("-app_name")
        app_user_agent = get_option("-app_user_agent")
        aria2c_args = get_option("-aria2c_args")
        app_setup_args = get_option("-app_setup_args")
        app_runpath = get_option("-app_runpath")
        app_setup = get_option("-app_setup")
        app_file = get_option("-app_file")
        downpath = get_option("-downpath")

        check_option()
        --winapi.show_message("当前软件下载地址：", app_url .. "\n\n\n伪装参数:" ..
user_agent .. "\n\n\n软件名称:".. app_name)
elseif has_option("-nbapp") then
        nbapp = get_option("-nbapp")
        app_file = get_option("-app_file")
        check_option()
        check_nbapp()
        --winapi.show_message("当前软件下载地址：", app_url .. "\n\n\n伪装参数:" ..
user_agent .. "\n\n\n软件名称:".. app_name)
else
default_appenv()
end
app_init()
end

function ontimer(id)
    if id == TIMER_ID_QUIT then
        sui:close()
    elseif id == TIMER_ID_LOADING then
        loading()
    elseif id == TIMER_ID_CHECK_INSTALLED then
        check_installed()
    end
end

function ondisplaychanged()
    move_top()
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_nbdl.lua"
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window size="380,110" caption="0,0,0,32" defaultfontcolor="#FFE8EAF6" visible="false">

    <Style imagerectstyle="l,t,w,h" />

    <VerticalLayout bkimage="themes\nbdl_bg.png">
          <HorizontalLayout  height="18">
              <Control />

              <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
          <Button name="::closebtn" text="x" font="sym16" textcolor="#ffff5800"
hottextcolor="#ffff5800" hotimage="color='#ffff5800'" width="20" height="64" textpadding="0,-10,0,0"
/>
          </HorizontalLayout>
  <text width="381" height="32" name="label_info" multiline="true" text="下载名称显示" font="14"
showhtml="true" textcolor="#FFFFFF" textpadding="0,-18,0,0" />

     <!--VerticalLayout width="381" height="42" bkimage="bg.png" bordersize="-1,-1,-1,-1"
bordercolor="#ffffffff" inset="5,-1,5,5" -->

        <!--/VerticalLayout-->
<text width="381" height="18" name="label_downspd" multiline="false" text="正在下载显示" font="14"
showhtml="true" textcolor="#FFFFFF" textpadding="5,0,0,0" />
         <VerticalLayout height="40">
                        <Slider name="process_slider" imm="true" min="1" max="100" step="1" value="100"
height="80" width="380" padding="0,-22,0,0"
                            thumbsize="55,39" thumbimage="file='themes\thumb.png'"
                            bkimage="file='themes\sliderbk.png'" foreimage="file='themes\sliderfr.png'" />
       </VerticalLayout>
               </VerticalLayout>

</Window>
```

```
local TID_LISTWINVER_CHANGED = 20001 + 1
USERPROFILE = os.getenv("USERPROFILE")
temp = os.getenv("temp") .. [[\]]
SystemDrive = os.getenv("SystemDrive")
listwinver = sui:find("listwinver")
listwinver.visible="1"



if File.exists("X:\\ipxefm\\ipxeboot.txt") then
  very_runmode = "pe"
  zexe = [[X:\\Program Files\\7-zip\\7z.exe]]
else
  very_runmode = "system"
  zexe = "%temp%\\AppStore\\bin\\7zx64.exe"
end
-- 正则匹配获取多行
function string.gfind(stdout, patten)
    local i, j = 0, 0
    return function()
        i, j = string.find(stdout, patten, j + 1)
        if (i == nil) then -- end find
            return nil
        end
        return string.sub(stdout, i, j)
    end
end
user_agent = [[--user-agent="Chrome/94.0.4606.71"]]

if not File.exists(temp .. [[\]] .. BaiPiaoid .. [[.cab]]) then
```

```
    exec(
        "/wait /hide",
        [[cmd /c aria2c ]] ..
        user_agent .. [[ --allow-overwrite=true ]] .. BaiPiaourl .. [[ -d %temp% -o ]] ..
BaiPiaoid .. [[.cab]]
                )

end

if File.exists(temp .. [[\]] .. BaiPiaoid .. [[.cab]]) then
exec("/wait /hide", zexe .. [[ -y x %temp%\]] .. BaiPiaoid .. [[.cab -owxsUI\UI_AppStore\]])
else
Alert("白嫖失败!")

end

local path = "wxsUI\\UI_AppStore\\products.xml"
local file = io.open(path, "rb")
text = file:read("*a")
file:close()
--text_ansi = winapi.encode(winapi.CP_UTF16, winapi.CP_ACP, text)
local index_regex = "<File id=\"\">"
local edition_regex = "<Edition>([^<]+)</Edition>"
local lcode_regex = "<LanguageCode>([^<]+)</LanguageCode>"
local filepath_regex = "<FilePath>([^<]+)</FilePath>"

ver={}
for index_str in string.gfind(text, "<File id=*.-</File>") do
local index_name = index_str:match(index_regex)
local index_edition = index_str:match(edition_regex)
local index_filepath = index_str:match(filepath_regex)
local index_lcode = index_str:match(lcode_regex)
if index_lcode == "zh-cn" and index_str:find("CLIENTCHINA") and index_str:find("x64") then
ver.chinax64url = index_filepath

elseif index_lcode == "zh-cn" and index_str:find("CLIENTCHINA") and index_str:find("x86")
then
ver.chinax86url = index_filepath

elseif index_lcode == "zh-cn" and index_str:find("CLIENTBUSINESS") and index_str:find("x64")
then
ver.businessx64url = index_filepath

elseif index_lcode == "zh-cn" and index_str:find("CLIENTBUSINESS") and index_str:find("x86")
then
ver.businessx86url = index_filepath

elseif index_lcode == "zh-cn" and index_str:find("Ultimate") and
index_str:find("CLIENTCONSUMER") and index_str:find("x64") then
ver.consumerx64url = index_filepath

elseif index_lcode == "zh-cn" and index_str:find("Ultimate") and
index_str:find("CLIENTCONSUMER") and index_str:find("x86") then
ver.consumerx86url = index_filepath



end


end
--64位全加入
table.insert(ver,string.format("64位家庭中文版 下载地址:%s", ver.chinax64url))
table.insert(ver,string.format("64位商业中文版 下载地址:%s", ver.businessx64url))
table.insert(ver,string.format("64位零售中文版 下载地址:%s", ver.consumerx64url))
--32位检测
if ver.chinax86url ~= nil then
table.insert(ver,string.format("32位家庭中文版 下载地址:%s", ver.chinax86url))
end
if ver.businessx86url ~= nil then
table.insert(ver,string.format("32位商业中文版 下载地址:%s", ver.businessx86url))
end
if ver.consumerx86url ~= nil then
table.insert(ver,string.format("32位零售中文版 下载地址:%s", ver.consumerx86url))
end
all = table.concat(ver, "\n")
 myindex = string.format("已获取到%s相关链接, 点这里选择镜像下载(网址已批量复制,请粘贴到记事本! )\n%s", BaiPiaoname, all)
 clipboard_url = winapi.set_clipboard(myindex)
 listwinver = sui:find("listwinver")
 listwinver.list = myindex
 listwinver.index = 0




--timer列表框事件检测
function listwinver_onchanged()
   esdurl = ver[listwinver.index]:match("http[^\"]*/[^\"]*/[^\"]*")
   esdname = string.format("%s.esd", ver[listwinver.index]:match("[^\"]+ "))
   esdname = string.gsub(esdname, " ", "")
```

```
    exec(
        "/wait /hide",
        [[cmd /c  start "" WINXSHELL -ui -jcfg wxsUI\UI_AppStore\nbtool_nbdl.jcfg -app_url
"]] .. esdurl .. [["]] ..
                " -app_name " .. esdname ..
                " -downpath " .. downpath ..
                " -app_setup " .. esdname

                )
end
--按钮事件检测

--列表框化变触发事件
function onchanged(ctrl)
  if ctrl == "listwinver" then
    suilib.call("SetTimer", TID_LISTWINVER_CHANGED, 200)

end
end











function ontimer(tid)
  if tid == TID_LISTWINVER_CHANGED then
    suilib.call("KillTimer", tid)
    listwinver_onchanged()

  end
```

```
end


--alert(all)--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_nbdl_Baipiao.lua"
```

```
function ennvme(syspath)
  if syspath == "" then
   winapi.show_message("NVME启用关闭", "取消")
  else
   exec('/hide /wait', 'reg load hklm\\offline_system ' .. syspath .. '\\Windows\\system32\\Config\\SYSTEM')
   exec('/hide /wait', 'reg delete hklm\\offline_system\\ControlSet001\\Services\\stornvme\\StartOverride /F')
   exec('/wait', 'reg unload hklm\\offline_system')
   winapi.show_message("NVME启用支持开启", [[离线系统所在分区]] .. syspath .. [[已启用NVME启动支持!请重启后测试!]])
  end
end

function check(a)
if a == "yes" then
--winapi.show_message('警告', a)
   local syspath = Dialog:BrowseFolder('启用NVME启动支持', 17)
   ennvme(syspath)
else
return
end
end



a=winapi.show_message('警告', "你将为目标系统开启NVME启动支持!\n 请肾重选择NVME固态硬盘上的系统分区", 'yes-no')
check(a)
```

```
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_NVME.lua"
```

```
{
  "name":"UI_AppStore",
  "lua":"nbtool_page.lua",
  "entry":"nbtool_page.xml",
  "nobaricon":true
}
{
  "ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_page.jcfg",
}
```

```
local SystemRoot = os.getenv("SystemRoot")
local TID_UPANLIST_CHANGED = 20000 + 1
local strDrivers = "请选择页面文件存放盘符"
local letters = winapi.get_logical_drives()
drv = {}
function str_drv(ck)
  for i, l in ipairs(letters) do
    if not File.exists(l .. [[Windows\System32\drivers\fbwf.sys]]) then
    strDrivers = strDrivers .. "\n" .. l
        if File.exists(l .. [[windows\system32\winlogon.exe]]) then
        strDrivers = strDrivers .. "    [检测到windows系统]"
        end
    end
    if ck == "all" and not File.exists(l .. [[Windows\System32\drivers\fbwf.sys]]) then
      drv[i] = l:gsub("\\", "")
          drv[i] = l:sub(1, 2)



          UPanList.index = 0
    end
  end
end

function onload()
  str_drv()
  UPanList = sui:find("UPanList_combo")
  strDrivers = strDrivers:gsub("\\", "")
  UPanList.list = strDrivers
  UPanList.index = 0
end
```

```
function onchanged(ctrl)
  if ctrl == "UPanList_combo" then
    suilib.call("SetTimer", TID_UPANLIST_CHANGED, 200)
  end
end

function ontimer(tid)
  if tid == TID_UPANLIST_CHANGED then
    suilib.call("KillTimer", tid)
    UPanList_combo_onchanged()
  end
end

function UPanList_combo_onchanged()
  if UPanList.index ~= 0 then
  panfu = UPanList.text:sub(1, 2)
  MinSize = sui:find("MinSize")
  mins = tonumber(MinSize.text)
  MaxSize = sui:find("MaxSize")
  maxs = tonumber(MaxSize.text)
  System:CreatePageFile(panfu, mins,maxs)
    winapi.show_message('页面设置', "虚拟内存已设置到" .. panfu)
  elseif UPanList.index == 1 then
    ck = "all"
    str_drv(ck)
  end
end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_page.lua"
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Window size="266, 75" caption="true" mininfo="266, 150">
    <VerticalLayout bkcolor="#FFD9D9D9">

        <HorizontalLayout>
            <Label text="选择磁盘:" />
               <Combo align="left" name="UPanList_combo" normalimage="file='bg_selected.png'"
style="ct-combo" wheel="false" height="" width="200" padding="0,0,0,0" textpadding="2,0,0,2"
                        vscrollbar="true" hscrollbar="true">

                    </Combo>
        </HorizontalLayout>
        <HorizontalLayout>
            <Label text="虚拟内存大小:" width="100"/>
            <Edit name="MinSize" text="8192" width="50" height="20" />
            <Label text="-" />
            <Edit name="MaxSize" text="8192" width="50" height="20" />
            <Label text="MB" />
        </HorizontalLayout>
        <HorizontalLayout>
            <Control />
            <Button name="btnOK" text="确定" width="80" bkcolor="#FF00BFFF"/>
                    <Button name="::closebtn" text="取消" textcolor="#ff8b8b8b" hottextcolor="#ffffffff"
hotimage="color='#ffe81123'" width="80" />
        </HorizontalLayout>
    </VerticalLayout>
    <CloseButton />
</Window>
```

```
function ennvme(syspath)
  if syspath == "" then
   winapi.show_message("离线系统盘符重置", "取消")
  else
   exec('/hide /wait', 'reg load hklm\\offline_system ' .. syspath .. '\\Windows\\system32\\Config\\SYSTEM')
   exec('/hide /wait', 'reg delete hklm\\offline_system\\MountedDevices /va /F')
   exec('/wait', 'reg unload hklm\\offline_system')
   winapi.show_message("离线系统盘符重置", [[离线系统所在分区]] .. syspath .. "\n" .. [[已重置盘符!请重启后测试!]])
  end
end

function check(a)
if a == "yes" then
--winapi.show_message('警告', a)
   local syspath = Dialog:BrowseFolder('离线系统盘符重置', 17)
   ennvme(syspath)
else
return
end
end




a=winapi.show_message('警告', "你将为目标系统系统盘符重置!即删除MountedDevices \n 请肾重选择系统分区", 'yes-no')
check(a)

--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_Panfu.lua"
```

```
function petoolspath(panfupath)
  if panfupath == "" then
   winapi.show_message("petools", "取消")
  else
   exec('/show /wait', [[cmd /c xcopy /s /e /y wxsUI\\UI_AppStore\soft\petools ]] .. panfupath .. [[\petools\]])
   winapi.show_message("导出petools示例", [[petools文件已导出到]] .. panfupath .. [[petools, 请按视频教程修改来使用!!]])
  end
end

function check(a)
if a == "yes" then
--winapi.show_message('警告', a)
   local panfupath = Dialog:BrowseFolder('导出petools示例', 17)
   petoolspath(panfupath)
else
return
end
end



a=winapi.show_message('提示', "你将导出petools示例到根目录 \n PE每次启动后将会执行petools目录下petools.ini中的语句! \n 请注意修改!", 'yes-no')
check(a)--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_petools.lua"
```

```
{
 "name":"UI_AppStore",
 "lua":"nbtool_powercfg.lua",
 "entry":"nbtool_powercfg.xml",
 "nobaricon":true
}
{
 "ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_powercfg.jcfg",
}
```

```
local SystemRoot = os.getenv("SystemRoot")
local TID_UPANLIST_CHANGED = 20000 + 1


function powercfg_init()
exitcode, stdout = winapi.execute("cmd /c powercfg /list")
output = stdout:gsub("\r\n", "\n")
powercfg_list = {}
 table.insert(powercfg_list, " 选择电源性能")
for line in string.gmatch(output, "[^\r\n]+") do
    if string.find(line, "GUID") then
        local match = string.match(line, "%(([^%)]+)%)")
        if match then
            if string.find(line, "%*%s*¥") then
                         table.insert(powercfg_list, match .. " *激活")
                         powercfg_list[1] = " 请选择模式,当前:[" .. match .. "*]"
                                else
                                table.insert(powercfg_list, match)
            end
        end
    end
end
powercfg_list_str = table.concat(powercfg_list, "\n")
  UPanList = sui:find("UPanList_combo")
  UPanList.list = powercfg_list_str


end

function onload()
  powercfg_init()
```

```
UPanList.index = 0

end

function onchanged(ctrl)
  if ctrl == "UPanList_combo" then
    suilib.call("SetTimer", TID_UPANLIST_CHANGED, 200)
  end
end

function ontimer(tid)
  if tid == TID_UPANLIST_CHANGED then
      UPanList_combo_onchanged()
          powercfg_init()
         UPanList.index = 0

     suilib.call("KillTimer", tid)


  end
end

function UPanList_combo_onchanged()
fwindex = UPanList.index + 1
if powercfg_list[fwindex]:find("高性能") then
        exec("/hide", [[cmd /c powercfg -s 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c]])
        elseif  powercfg_list[fwindex]:find("平衡") then
   exec("/hide", [[cmd /c powercfg -s 381b4222-f694-41f0-9685-ff5bb260df2e]])

   elseif  powercfg_list[fwindex]:find("节能") then
   exec("/hide", [[cmd /c powercfg -s a1841308-3541-4fab-bc81-f71556f20b4a]])
```

```
    elseif  powercfg_list[fwindex]:find("卓越性能") then
   exec("/hide", [[cmd /c powercfg -s e9a42b02-d5df-448d-aa00-03f14749eb61]])
 else
 return
    end




end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_powercfg.lua"
```

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Window size="400,140" sizebox="4,4,4,4" caption="0,0,0,32" mininfo="400,140">

<Style imagerectstyle="l,t,w,h" />
<Include source="scrollbar_ltwh.xml" />
<Style name="nav_item" height="40"
      selectedimage="color='#ff0078d7' dest='155,5,5,20'"
    hotimage="color='#FFA2D7FF'  dest='155,5,5,20'"
    align="left" textpadding="52,-10,0,0" />
<VerticalLayout bordersize="1" bordercolor="#FF1985DA">

    <!-- 左-导航, 右-内容 水平布局 -->
    <HorizontalLayout style="ct-bknav" padding="1,1,1,1" >

        <!-- 左侧 区域 -->
        <VerticalLayout name="nav" width="1" >
            <!-- 左上 logo 区域 -->
            <Label height="0" text="应用市场" padding="20,0,0,0" />
            <HorizontalLayout style="nav_item" padding="20,0,0,0" >
                <Label text="   " font="sym16" width="32" />
                <Button text="%{Home}" align="left" name="home_button" />
            </HorizontalLayout>

            <Label height="0" showhtml="true" text="分类" padding="20,0,0,0" />

            <!-- 左侧 导航 项目 -->
            <VerticalLayout name="$TabList">

            </VerticalLayout>


        </VerticalLayout>



        <!-- 右侧内容区域格式 -->
        <VerticalLayout padding="0,1,1,1" style="ct-bkcontent" bkimage="nbtool_ck_bg.png">

            <!-- 右侧顶部标题区域 -->
            <VerticalLayout height="16">
                <HorizontalLayout height="16">
                    <Label text="NBTool" textpadding="5,0,0,0" />
                    <Control />
                    <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
                    <Button name="::minbtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ff1985da'" width="48" height="32" />
                    <Button name="::closebtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ffe81123'" width="48" height="32" />
                </HorizontalLayout>

            </VerticalLayout>

            <!-- 右侧 内容 主区域 -->
            <VerticalLayout name="$TabLayoutParent" height="80">
                <VerticalLayout>
                                        <Label align="center" text="电源性能调节(旧笔记本建议节能模式)"
font="22" height="32" showhtml="true" padding="0,5,0,0" />
                    <Combo align="center" name="UPanList_combo"
normalimage="file='nbtool_ck_bg_selected.png'" showhtml="true" style="ct-combo" wheel="false"
height="32" width="200" padding="100,10,0,0" textpadding="2,0,0,2"
                        vscrollbar="true" hscrollbar="true">
                    </Combo>
  </VerticalLayout>

            </VerticalLayout>
                        <VerticalLayout padding="0,0,0,0" width="400" bordercolor="#FF1985DA" >


 </VerticalLayout>






        </VerticalLayout>

    </HorizontalLayout>
</VerticalLayout>
</Window>
```

```
{
 "name":"UI_AppStore",
 "lua":"nbtool_tool.lua",
 "entry":"nbtool_tool.xml",
 "nobaricon":true
}
{
 "ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_tool.jcfg",
}
```

```
local SystemRoot = os.getenv("SystemRoot")
local USERPROFILE = os.getenv("USERPROFILE")
local TID_UPANLIST_CHANGED = 20000 + 1
tool_list = {"请选择日常维护操作","重建图标缓存","重启资源管理器","设置IPV4解析优先","跳过强制微软帐户"}


function ennvme(syspath)
  if syspath == "" then
   winapi.show_message("跳过强制使用微软帐户登陆", "取消")
  else
   exec('/hide /wait', 'reg load hklm\\offline_soft ' .. syspath .. '\\Windows\\system32\\Config\\SOFTWARE')
   exec('/hide /wait', [[reg add "HKLM\offline_soft\Microsoft\Windows\CurrentVersion\Policies\System" /f /v "NoConnectedUser" /t REG_DWORD /d 3]])
  exec('/hide /wait', [[reg add "HKLM\software\Microsoft\Windows\CurrentVersion\Policies\System" /f /v "NoConnectedUser" /t REG_DWORD /d 3]])

   exec('/wait', 'reg unload hklm\\offline_soft')

   winapi.show_message("跳过强制使用微软帐户登陆", [[离线系统所在分区]] .. syspath .. "\n" .. [[设置跳过!请重启后测试!]])
  end
end

function check(a)
if a == "yes" then
--winapi.show_message('警告', a)
   local syspath = Dialog:BrowseFolder('跳过强制使用微软帐户登陆', 17)
   ennvme(syspath)
else
```

```
return
end
end



function onload()
  UPanList = sui:find("UPanList_combo")
 tool_list_str = table.concat(tool_list, "\n")
 UPanList.list = tool_list_str
  UPanList.index = 0
end

function onchanged(ctrl)
  if ctrl == "UPanList_combo" then
    suilib.call("SetTimer", TID_UPANLIST_CHANGED, 200)
  end
end

function ontimer(tid)
  if tid == TID_UPANLIST_CHANGED then
    suilib.call("KillTimer", tid)
    UPanList_combo_onchanged()
  end
end

function UPanList_combo_onchanged()
fwindex = UPanList.index + 1
if tool_list[fwindex]:find("重建图标缓存") then
        exec("/show", [[cmd /c taskkill /f /im explorer.exe]])
        exec("/show /wait", [[cmd /c attrib -h -i %userprofile%\AppData\Local\IconCache.db]])
        exec("/show /wait ", [[cmd /c del %userprofile%\AppData\Local\IconCache.db /a]])
        exec("/show", [[cmd /c start "" %SystemRoot%\system32\userinit.exe]])
```

```
elseif  tool_list[fwindex]:find("重启资源管理器") then
        exec("/show /wait", [[cmd /c taskkill /f /im explorer.exe]])
        exec("/show", [[cmd /c start "" %SystemRoot%\system32\userinit.exe]])


elseif  tool_list[fwindex]:find("设置IPV4解析优先") then
        exec("/show", [[cmd /c netsh interface ipv6 set  prefixpolicy ::ffff:0:0/96 100 4]])
        alert("设置完成，现在ping计算机名\n访问网站将优先解析为ipv4地址 \n 需要先禁用ipv6协议")
elseif  tool_list[fwindex]:find("跳过强制微软帐户") then
 a=winapi.show_message('警告', "你将操作离线系统 \n 请肾重选择系统分区", 'yes-no')
 check(a)
  else
 alert("请选择操作")
 return
    end


end
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_tool.lua"
```

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Window size="400,140" sizebox="4,4,4,4" caption="0,0,0,32" mininfo="400,140">

<Style imagerectstyle="l,t,w,h" />
<Include source="scrollbar_ltwh.xml" />
<Style name="nav_item" height="40"
      selectedimage="color='#ff0078d7' dest='155,5,5,20'"
    hotimage="color='#FFA2D7FF'  dest='155,5,5,20'"
    align="left" textpadding="52,-10,0,0" />
<VerticalLayout bordersize="1" bordercolor="#FF1985DA">

    <!-- 左-导航, 右-内容 水平布局 -->
    <HorizontalLayout style="ct-bknav" padding="1,1,1,1" >

        <!-- 左侧 区域 -->
        <VerticalLayout name="nav" width="1" >
            <!-- 左上 logo 区域 -->
            <Label height="0" text="应用市场" padding="20,0,0,0" />
            <HorizontalLayout style="nav_item" padding="20,0,0,0" >
                <Label text="   " font="sym16" width="32" />
                <Button text="%{Home}" align="left" name="home_button" />
            </HorizontalLayout>

            <Label height="0" showhtml="true" text="分类" padding="20,0,0,0" />

            <!-- 左侧 导航 项目 -->
            <VerticalLayout name="$TabList">

            </VerticalLayout>


        </VerticalLayout>


        <!-- 右侧内容区域格式 -->
        <VerticalLayout padding="0,1,1,1" style="ct-bkcontent" bkimage="nbtool_ck_bg.png">

            <!-- 右侧顶部标题区域 -->
            <VerticalLayout height="16">
                <HorizontalLayout height="16">
                    <Label text="NBTool" textpadding="5,0,0,0" />
                    <Control />
                    <!-- https://docs.microsoft.com/ja-jp/windows/uwp/design/style/segoe-ui-symbol-font -->
                    <Button name="::minbtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ff1985da'" width="48" height="32" />
                    <Button name="::closebtn" text="   " font="sym" textcolor="#ff8b8b8b"
hottextcolor="#ffffffff" hotimage="color='#ffe81123'" width="48" height="32" />
                </HorizontalLayout>

            </VerticalLayout>

            <!-- 右侧 内容 主区域 -->
            <VerticalLayout name="$TabLayoutParent" height="80">
                <VerticalLayout>
                                    <Label align="center" text="常用维护操作工具箱" font="24" height="32"
showhtml="true" padding="0,5,0,0" />
                    <Combo align="center" name="UPanList_combo"
normalimage="file='nbtool_ck_bg_selected.png'" style="ct-combo" wheel="false" height="32"
width="200" padding="100,10,0,0" textpadding="2,0,0,2"
                        vscrollbar="true" hscrollbar="true">
                    </Combo>
  </VerticalLayout>

            </VerticalLayout>
                        <VerticalLayout padding="0,0,0,0" width="400" bordercolor="#FF1985DA" >


 </VerticalLayout>





        </VerticalLayout>

    </HorizontalLayout>
</VerticalLayout>
</Window>
```

VMware16Pro
ZF3R0-FHED2-M80TY-8QYGC-NPKYF
YF390-0HF8P-M81RQ-2DXQE-M2UT6
ZF71R-DMX85-08DQY-8YMNC-PPHV8
VMware17Pro
MC60H-DWHD5-H80U9-6V85M-8280D
VMware激活密钥（通用批量永久激活许可）
17：JU090-6039P-08409-8J0QH-2YR7F
16：ZF3R0-FHED2-M80TY-8QYGC-NPKYF
15：FC7D0-D1YDL-M8DXZ-CYPZE-P2AY6
12：ZC3TK-63GE6-481JY-WWW5T-Z7ATA
10：1Z0G9-67285-FZG78-ZL3Q2-234JG

Edge添加永久IE模式企业网站列表，要打开企业网站列表管理器
需要打开Edge并输入edge://compat/sitelistmanager来导入
需要打开IE兼容模式功能edge://settings/defaultBrowser

微软原版系统镜像解析xml地址
win10 https://go.microsoft.com/fwlink/?LinkId=841361
win11 https://go.microsoft.com/fwlink?linkid=2156292
谷歌浏览器完整包直链地址
Windows版（含更新组件）64位
https://dl.google.com/update2/installers/ChromeStandaloneSetup64.exe
https://dl.google.com/tag/s/installdataindex/update2/installers/ChromeStandaloneSetup64.exe

Windows版（含更新组件）32位
https://dl.google.com/update2/installers/ChromeStandaloneSetup.exe
https://dl.google.com/tag/s/installdataindex/update2/installers/ChromeStandaloneSetup.exe

Win10专业教育版镜像ESD 19044
http://dl.delivery.mp.microsoft.com/filestreamingservice/files/e7aa8b81-ee80-4b52-8f04-c364f0ce1af1/19044.1288.211006-0501.21h2_release_svc_refresh_CLIENTCONSUMER_RET_x64FRE_zh-cn.esd
Win11 Business 版本ESD镜像22000.318
http://dl.delivery.mp.microsoft.com/filestreamingservice/files/73bedbd0-f169-4f2b-82ea-e6fa078838d6/22000.318.211104-1236.co_release_svc_refresh_CLIENTBUSINESS_VOL_x64FRE_zh-cn.esd

```
<Page>
    <VerticalLayout padding="20,0,0,0" >
        <Label text="%{Background}" font="24" height="50" />
    </VerticalLayout>
</Page>
```

```
UIPages['Colors'] = {
  Name = 'Colors'
}

UIPages['Colors'].Init = function()
    local ct_name = 'dark'
    if System:GetSetting('SysColorTheme') == 1 then ct_name = 'light' end
    local opt = sui:find('$Opt.SysColorTheme[' .. ct_name .. ']')
    opt.selected = 1

    UIOption.Set('$Opt.SysColorTheme[?]',
      System:GetSetting('SysColorTheme') == 1, {'light', 'dark'})

    UIOption.Set('$Opt.AppsColorTheme[?]',
      System:GetSetting('AppsColorTheme') == 1, {'light', 'dark'})

    local onoff = System:GetSetting('Colors.Transparency')
    UISwitch.Set('$Switch.ColorsTransparency', onoff)

    if System:GetSetting('ShellColorPrevalence') == 1 then UICheckBox.Check('$Chk.ApplyACShell') end
    if System:GetSetting('WindowColorPrevalence') == 1 then UICheckBox.Check('$Chk.ApplyACWindow') end
end


local function restart()
  wxsUI('UI_Settings','main.jcfg','-colors ' .. fixscreen_opt)
  suilib.close()
end


UI.OnClick['$Opt.SysColorTheme'] = function(opt)
  System:SysColorTheme(opt)
end

UI.OnClick['$Opt.AppsColorTheme'] = function(opt)
  System:AppsColorTheme(opt)
  restart()
end

UI.OnChanged['$Switch.ColorsTransparency'] = function(val)
  System:SetSetting('Colors.Transparency', val)
end

UI.OnChanged['$Chk.ApplyACShell'] = function(val)
  System:SetSetting('ShellColorPrevalence', val)
end

UI.OnChanged['$Chk.ApplyACWindow'] = function(val)
  System:SetSetting('WindowColorPrevalence', val)
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.colors.lua"
```

```
<Page>
    <VerticalLayout padding="20,0,0,0" >
        <Label text="%{Colors}" font="24" height="50" />
        <Label text="%{SelectSysColorTheme}" padding="0,10,0,0" />
        <Option name="$Opt.SysColorTheme[light]" style="radio" text="#{@shell32.dll,51730}" group="syscolortheme" padding="0,5,0,0" />
        <Option name="$Opt.SysColorTheme[dark]" style="radio" text="#{@shell32.dll,51731}" group="syscolortheme" padding="0,5,0,0" />
        <Label text="%{SelectAppColorTheme}" padding="0,10,0,0" />
        <Option name="$Opt.AppsColorTheme[light]" style="radio" text="%{Light}" group="appscolortheme" padding="0,5,0,0" />
        <Option name="$Opt.AppsColorTheme[dark]" style="radio" text="%{Dark}" group="appscolortheme" padding="0,5,0,0" />
        <Label text="%{TransparencyEffects}" padding="0,10,0,0" />
        <CheckBox name="$Switch.ColorsTransparency" style="switch" text="%{Off}" width="120" align="left" textpadding="50,0,0,0" padding="0,5,0,0" />
        <Label text="%{ApplyAccentColorDesc}" padding="0,10,0,0" />
        <CheckBox name="$Chk.ApplyACShell" style="check" text="%{StartTaskbarActionCenter}" width="300" align="left" padding="0,5,0,0" />
        <CheckBox name="$Chk.ApplyACWindow" style="check" text="%{TitlebarsAndWinBorders}" width="300" align="left" padding="0,5,0,0" />
    </VerticalLayout>
</Page>
```

```
AUTO_RESTORE_SEC = 15
dpi_list = {["100"] = 0, ["125"] = 1, ["150"] = 2, ["175"] = 3, ["200"] = 4, ["225"] = 5, ["250"] = 6, ["300"] = 7}


UIPages['Display'] = {
  Name = 'Display'
}

UIPages['Display'].Init = function(self, tab)
    dpi_combo = sui:find('dpi_combo')
    resolution_combo = sui:find('resolution_combo')
    rotate_combo = sui:find('rotate_combo')
    display_applybtn = sui:find('display_applybtn')
    display_restorebtn = sui:find('display_restorebtn')
    org_display_restorebtn_text = display_restorebtn.text -- save origin retore button text for retore timer

    tab:OnClick(self.Name) -- be visible for changing combobox index

    last_dpi = 0
    local cdpi =Screen:Get('DPI')
    local dpi_index = dpi_list[" .. cdpi]
    if dpi_index ~= nil then
      dpi_combo.index = dpi_index
    else
      if cdpi > 100 then
        dpi_index = (cdpi - 100) // 25
      end
    end
    last_dpi = dpi_index
    resolution_combo.list = suilib.call('GetResolutionList')
    resolution_combo.index = suilib.call('GetCurrentResolution')

    rotate_combo.index = suilib.call('GetRotation')

    last_resolution = resolution_combo.text
    last_rotate = rotate_combo.index

    brightness_slider = sui:find('brightness_slider')
    brightness_val = Screen:Get('brightness')
    brightness_slider.value = brightness_val
end

local function fixscreen()
  print("fixscreen", tostring(fixscreen_opt))
  if fixscreen_opt ~= " then
      App:Call('Desktop::UpdateWallpaper')
      App:Sleep(200)
      App:Call('Taskbar::ChangeNotify')
  end
end

local function display_ui_restore(mouse, visible)
  resolution_combo.mouse = mouse
  rotate_combo.mouse = mouse
  display_applybtn.visible = visible
  display_restorebtn.visible = visible
```

```
end

local function display_combo_changed()
  fixscreen()
  -- set auto restore timer
  display_restore_counter = AUTO_RESTORE_SEC

  display_ui_restore('false', 1)

  display_restorebtn.text = string.format('[%d]%s', display_restore_counter, org_display_restorebtn_text)
  UI:SetTimer('AUTO_RESTORE_COUNTER', 1000)
end

local function display_applybtn_click()
  UI:KillTimer('AUTO_RESTORE_COUNTER')

  last_resolution = resolution_combo.text
  last_rotate = rotate_combo.index

  display_ui_restore('true', 0)
end

local function display_restorebtn_click()
  UI:KillTimer('AUTO_RESTORE_COUNTER')

  suilib.call('SetResolution', last_resolution)
  suilib.call('SetRotation', last_rotate)
  fixscreen()
  resolution_combo.index = last_resolution
  rotate_combo.index = last_rotate

  display_ui_restore('true', 0)
end


UI.OnChanged['resolution_combo'] = function(val)
  if last_resolution == resolution_combo.text then return end

  suilib.call('SetResolution', resolution_combo.text)
  display_combo_changed()
end

UI.OnChanged['rotate_combo'] = function(val)
  if last_rotate == val then return end

  suilib.call('SetRotation', val)
  display_combo_changed()
end

UI.OnChanged['dpi_combo'] = function(val)
  if val == 7 then
    Screen:Set('DPI', 300)
  else
    Screen:Set('DPI', (val * 25) + 100) -- TODO: custom dpi
  end
end
```

```lua
UI.OnChanged['brightness_slider'] = function(val, ctrl)
  brightness_val = val
  UI:SetTimer('RESET.BRIGHTNESS_TIMER', 200)
end

UI.OnClick['display_applybtn'] = function(ctrl)
  display_applybtn_click()
end

UI.OnClick['display_restorebtn'] = function(ctrl)
  display_restorebtn_click()
end

UI.OnTimer['AUTO_RESTORE_COUNTER'] = function(tid)
  display_restore_counter = display_restore_counter - 1
  display_restorebtn.text = string.format('[%d]%s', display_restore_counter, org_display_restorebtn_text)
  -- auto restore
  if (display_restore_counter <= 0) then
      display_restorebtn_click()
  end
end

UI.OnTimer['RESET.BRIGHTNESS_TIMER'] = function(tid)
  local val = math.floor(brightness_val / 10 + 0.5) * 10;
  Screen:Set('brightness', val)
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.display.lua"
```

```
<Page>
   <VerticalLayout padding="20,0,0,0" >
       <Label text="%{Display}" font="24" height="50" />
       <Label text="<b>%{DPI}</b>" showhtml="true" padding="0,10,0,0" />
       <Combo name="dpi_combo" style="ct-combo" wheel="false" height="32" width="300" padding="10,8,10,0"
textpadding="5,2,0,2"
           vscrollbar="true" hscrollbar="true">
           <ListLabelElement text="100%" selected="true" />
           <ListLabelElement text="125%"/>
           <ListLabelElement text="150%" />
           <ListLabelElement text="175%" />
           <ListLabelElement text="200%" />
           <ListLabelElement text="225%" />
           <ListLabelElement text="250%" />
           <ListLabelElement text="300%" />
       </Combo>
       <Label text="<b>%{Resolution}</b>" showhtml="true" padding="0,10,0,0" />
       <Combo name="resolution_combo" style="ct-combo" wheel="false" height="32" width="300"
padding="10,8,10,0" textpadding="5,2,0,2"
           vscrollbar="true" hscrollbar="true">
           <ListLabelElement text="1920x1080" />
           <ListLabelElement text="1024x768" selected="true" />
           <ListLabelElement text="800x600" />
       </Combo>
       <Label text="<b>%{Rotate}</b>" showhtml="true" padding="0,10,0,0" />
       <Combo name="rotate_combo" style="ct-combo" wheel="false" height="32" width="300"
padding="10,8,10,0" textpadding="5,2,0,2"
           vscrollbar="true" hscrollbar="true">
           <ListLabelElement text="%{Rotate_0}" selected="true" />
           <ListLabelElement text="%{Rotate_90}"/>
           <ListLabelElement text="%{Rotate_180}" />
           <ListLabelElement text="%{Rotate_270}" />
       </Combo>
       <Label text="<b>%{Brightness}</b>" showhtml="true" padding="0,10,0,0" />
       <HorizontalLayout height="32">
           <Slider name="brightness_slider" imm="true" min="20" max="100" step="10" value="100" height="24"
width="300" padding="5,4,0,0"
               thumbsize="8,24" thumbimage="file='thumb.png'"
               bkimage="file='sliderbk.png'" foreimage="file='sliderfr.png'" />
       </HorizontalLayout>
       <HorizontalLayout height="40" padding="0,30,0,0">
           <Button name="display_applybtn" text="%{Apply}" visible="false" shortcut="A" showhtml="true"
style="ct-button" padding="0,0,10,0" />
           <Button name="display_restorebtn" text="%{Restore}" visible="false" shortcut="R" showhtml="true"
style="ct-button" padding="0,0,10,0" />
       </HorizontalLayout>
   </VerticalLayout>
</Page>
```

```
UIPages['FolderOptions'] = {}

UIPages['FolderOptions'].Init = function()
  local onoff = FolderOptions:Get('ShowAll')
  UISwitch.Set('$Switch.FolderOptShowAll', onoff)

  onoff = FolderOptions:Get('ShowExt') + 1  -- trick: -1(true), 0(false) => 0(false), 1(true)
  UISwitch.Set('$Switch.FolderOptShowExt', onoff)

  onoff = FolderOptions:Get('ShowSuperHidden') + 1
  UISwitch.Set('$Switch.FolderOptShowSuperHidden', onoff)
end

UI.OnChanged['$Switch.FolderOptShowAll'] = function(val)
  FolderOptions:Set('ShowAll', val)
end

UI.OnChanged['$Switch.FolderOptShowExt'] = function(val)
  FolderOptions:Set('ShowExt', val - 1)
end

UI.OnChanged['$Switch.FolderOptShowSuperHidden'] = function(val)
  FolderOptions:Set('ShowSuperHidden', val - 1)
end

--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.folderoptions.lua"
```

```
<Page>
   <VerticalLayout padding="20,0,0,0" >
      <Label name="#res_3" text="#{@shell32.dll,22985}" font="24" height="50" />
      <Label name="#res_4" text="#{@shell32.dll,30500}" padding="0,10,0,0" />
      <CheckBox name="$Switch.FolderOptShowAll" style="switch" text="%{Off}" width="120" align="left" textpadding="50,0,0,0" padding="0,5,0,0" />
      <Label name="#res_5" text="#{@shell32.dll,30503}" padding="0,10,0,0" />
      <CheckBox name="$Switch.FolderOptShowExt" style="switch" text="%{Off}" width="120" align="left" textpadding="50,0,0,0" padding="0,5,0,0" />
      <Label name="#res_6" text="#{@shell32.dll,30508}" padding="0,10,0,0" />
      <CheckBox name="$Switch.FolderOptShowSuperHidden" style="switch" text="%{Off}" width="120" align="left" textpadding="50,0,0,0" padding="0,5,0,0" />
   </VerticalLayout>
</Page>
```

```
local page_itemXML = ''
local page_head = ''
local page_end = ''

TabPage = {}

local pageUI_item = {}

function TabPage:Init()
end

function TabPage:SetTileLayoutXML(_head, _end, item)
  page_head = _head
  page_end = _end
  page_itemXML = item
end

function TabPage:InitPage(id, name)
  if pageUI_item[id] ~= nil then
    return pageUI_item[id]
  end
  local page = sui:find('$Page[' .. id .. ']')
  local xml = TabPage:FillTileItem(id, name, page_itemXML)
  -- msg(xml)
  suilib.insertItem(page, xml, page_head, page_end)
  pageUI_item[id] = page
  return page
end

dofile(UI_Path .. '_pageData.lua')
--"ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\page.lua"
```

```lua
UIPages['Taskbar'] = {}

UIPages['Taskbar'].Init = function()
    local onoff = Taskbar:GetSetting('AutoHide')
    UISwitch.Set('$Switch.TaskbarAutoHide', onoff)

    onoff = (Taskbar:GetSetting('TaskbarSmallIcons') or 0) | 0 -- to integer
    UISwitch.Set('$Switch.TaskbarSmallIcons', onoff)

    local glomlevel = (Taskbar:GetSetting('TaskbarGlomLevel') or 2) | 0
    UIOption.Set('$Opt.CombineButtons[?]', glomlevel + 1, {'always', 'auto', 'never'})
end

UI.OnChanged['$Switch.TaskbarAutoHide'] = function(val)
  Taskbar:AutoHide(val)
end

UI.OnChanged['$Switch.TaskbarSmallIcons'] = function(val)
  Taskbar:UseSmallIcons(val)
end

UI.OnClick['$Opt.CombineButtons'] = function(opt)
  Taskbar:CombineButtons(opt)
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.taskbar.lua"
```

```
<Page>
    <VerticalLayout padding="20,0,0,0" >
        <Label text="%{Taskbar}" font="24" height="50" />
        <Label text="%{AutoHideTaskbar}" padding="0,10,0,0" />
        <CheckBox name="$Switch.TaskbarAutoHide" style="switch" text="%{Off}" width="120"
align="left" textpadding="50,0,0,0" padding="0,5,0,0" />
        <Label text="%{UseSmallButtons}" padding="0,10,0,0" />
        <CheckBox name="$Switch.TaskbarSmallIcons" style="switch" text="%{Off}" width="120"
align="left" textpadding="50,0,0,0" padding="0,5,0,0" />
        <Label text="%{CombineButtons}" padding="0,10,0,0" />
        <HorizontalLayout padding="0,5,0,0">
            <Option name="$Opt.CombineButtons[always]" style="radio" text="%{Always}"
group="combine_style" />
            <Option name="$Opt.CombineButtons[auto]" style="radio" text="%{Auto}"
group="combine_style" />
            <Option name="$Opt.CombineButtons[never]" style="radio" text="%{Never}"
group="combine_style" />
        </HorizontalLayout>
    </VerticalLayout>
</Page>
```

```
local exitcode, result = winapi.execute([[cmd /c echo list disk | diskpart | find "磁盘 0"]])
result = result:gsub("\r\n", "\n")



for i, j, k, l in string.gmatch(result, "(%S+)%s+(%S+)%s+(%S+)%s+(%S+)") do

    local size = tonumber(l)
        alert(size)
    if size then
        if size > 1 and size < 221 then
            seldisk = "single"
        elseif size > 222 and size < 233 then
            seldisk = "240G"
        elseif size > 234 and size < 257 then
            seldisk = "256G"
        elseif size > 446 and size < 501 then
            seldisk = "500G"
        elseif size > 882 and size < 999 then
            seldisk = "1t"
        elseif size > 1862 and size < 1999 then
            seldisk = "2t"
                 elseif size > 5000 and size < 9999 then
            seldisk = "lua5g"
                else
        seldisk = "检测不到"
        end

    end

end
```

```lua
alert(seldisk)
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\PEcheckdisk.lua"
```

```lua
-- 提取文件名中的变量和值，并设置为全局变量
function extract_vars_and_set(filename)
    for var, value in string.gmatch(filename, "@([^=]*)=([^@]*)") do
        _G[var] = value
        print(var, value)
    end
end

-- 主函数
function findjob()
    local path = "X:/ipxefm"
    local p = io.popen('dir "'..path..'" /b *@*')
    for filename in p:lines() do
        extract_vars_and_set(filename)
    end
end

findjob()

-- 打印所有全局变量
alert("服务器端:" .. ip .. "\n执行的任务:" .. job .. "\n参数:" .. args .. "\n硬盘大小: 100T\n警告:你的网卡速率为100M\n警告:你的硬盘有坏道!")
--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\PEfindjob.lua"
```

```
fixed_out = {}
table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\petools.png><a vm><c #000000>暂无收藏敬请期待!</c></a> \n")
            )--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\petools.lua"
```

```
-- 获取系统驱动器
systemdrive = os.getenv("systemdrive")

-- 执行ipconfig /all命令并获取输出
local exitcode, text = winapi.execute("ipconfig /all")
text = text:gsub("\r\n", "\n")

-- 定义获取网卡名称的函数
function getcardname(netcard)
    -- 定义网卡名称关键词和结束关键词
    local netcardsname = {"以太网适配器 以太网", "以太网适配器 Ethernet"}
    local end_pattern = "TCPIP 上的 NetBIOS"
    local netcard = ""
    local recording = false
    -- 遍历输出的每一行
    for line in text:gmatch("[^\r\n]+") do
        -- 检查是否包含网卡名称关键词
        for _, start_pattern in ipairs(netcardsname) do
            if line:find(start_pattern) then
                recording = true
                break
            end
        end
        -- 如果包含关键词，则记录该行
        if recording then
            netcard = netcard .. line .. "\n"
        end
        -- 如果包含结束关键词，则停止记录
        if line:find(end_pattern) then
            recording = false
        end
    end
    return netcard

end

-- 定义获取环境变量的函数
function get_myenv()
    -- 获取网卡信息
    netcard = getcardname()
    -- 提取IP、子网掩码、默认网关和适配器名称
        adapter = string.match(netcard, "以太网适配器%s*(.-):")
    myip = netcard:match("IPv4 地址 .-: (%d+%.%d+%.%d+%.%d+)")
    mymask = netcard:match("子网掩码 .-: (%d+%.%d+%.%d+%.%d+)")
    mygw = string.match(netcard, "默认网关.-%s*(%d+%.%d+%.%d+%.%d+)%s*DHCP 服务器")
    if mygw == nil then mygw = netcard:match("默认网关. . . . . . . . . . . . . : (%d+%.%d+%.%d+%.%d+)") end
        return myip,mymask,mygw,adapter
end

-- 定义设置IP的函数
function wtip()
    get_myenv()
    -- 如果系统驱动器为X:, 则使用pecmd.exe设置IP
    if systemdrive == "X:" then
        exec("pecmd.exe PCIP " .. myip .. "," .. mymask .. "," .. mygw .. "," .. mygw .. ";223.5.5.5,0")
    else
        -- 否则, 使用netsh命令设置IP和DNS
        local cmd_wtip = string.format('netsh interface ip set address "' .. adapter .. '" static %s %s %s', myip, mymask, mygw)
        exec("/hide",cmd_wtip)
        exec("/hide",'netsh interface ip add dns "' .. adapter .. '" ' .. mygw .. ' index=1')
        exec("/hide",'netsh interface ip add dns "' .. adapter .. '" 223.5.5.5 index=2')
    end
end

-- 调用设置IP的函数
wtip()




--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\PEwtip.lua"
```

```
@echo off
mode con:cols=80 lines=20
set nowcg=当前未修改
:menu
set key=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\NcdAutoSetup\Private
set valueName=autosetup
set openValue=1
set closeValue=0


REM 检查注册表键值是否存在
reg query "%key%" /v "%valueName%" >nul 2>&1

if %errorlevel% equ 0 (
  REM 如果键值存在，检查它的值
  for /f "tokens=3" %%i in ('reg query "%key%" /v "%valueName%"') do (
    if %%i equ %openValue% (
      set p_autosetup=开启
    ) else (
      set p_autosetup=关闭
    )
  )
) else (
  REM 如果键值不存在，默认为开启
  set p_autosetup=开启
)

echo autosetup的状态为：%p_autosetup%

rem 查询默认打印机名称和端口
for /f "tokens=2 delims==" %%a in ('wmic printer where default^="true" get portname /value')
```

```
do set df_printer_port=%%a
for /f "tokens=2 delims==" %%a in ('wmic printer where default^="true" get name /value') do
set df_printer=%%a
echo Default Printer: %df_printer% on port %df_printer_port%
title AppStore-打印机维护工具
cls
echo ================================
echo         打印机维护工具
echo ================================
echo 0. 打开旧的"硬件和打印机文件夹"
echo 1. 打开高级共享设置
echo 2. 开启LPD打印机共享功能
echo 3. 禁用网络连接设备的自动设置(网络发现) 当前状态为: %p_autosetup%
echo 4. 设置默认打印机 当前: %df_printer%  端口:%df_printer_port%
echo 5. 添加LPD打印机
echo 6. 手动添加打印机
echo 7. 清理打印任务
echo 8. 打印管理[方便]
echo 9. 修复SMB共享[备用方案，风险!不推荐]
echo c. 修改计算机名并立即生效，无须重启! [ 当前:%computername% ]  %nowcg% %hostname%
echo x. 退出

set /p choice=请输入选项的数字:

if "%choice%"=="0" start "" shell:::{A8A91A66-3A7D-4424-8D24-04E180695C7A}&&goto menu
if "%choice%"=="1" start "" control.exe /name Microsoft.NetworkAndSharingCenter /page
Advanced&&goto menu
if "%choice%"=="2" goto AddLPD
if "%choice%"=="3" REG ADD
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\NcdAutoSetup\Private" /v
"AutoSetup" /t REG_DWORD /d 0 /f&&goto menu
```

```
if "%choice%"=="4" call :set_printer&&goto menu
if "%choice%"=="5" call :add_lpd_printer&&goto menu
if "%choice%"=="6" start "" rundll32 printui.dll,PrintUIEntry /il&&goto menu
if "%choice%"=="7" call :del_jobs&&goto menu
if "%choice%"=="8" start "" %systemroot%\system32\printmanagement.msc&&goto menu
if "%choice%"=="9" call :fix_smb_printer&&goto menu
if "%choice%"=="c" call :cgname&&goto menu
if "%choice%"=="x" exit

echo 无效的选项，请重试。
pause
goto menu



:AddLPD
@echo 开启LPD打印相关功能……
@echo off

ver | find "6.1." > nul && set os=win7
ver | find "10.0." > nul && set os=win10

if "%os%"=="win7" (
    echo Windows 7 detected
    REM 执行 Win7 系统下的操作
    call :win7

) else if "%os%"=="win10" (
    echo Windows 10 detected
    REM 执行 Win10 系统下的操作
    call :win10
) else (
    echo Unsupported OS version
)

pause



:win10
rem windows10
dism /online /enable-feature /featurename:"Printing-Foundation-InternetPrinting-Client" /all
dism /online /enable-feature /featurename:"Printing-Foundation-LPDPrintService" /all
dism /online /enable-feature /featurename:"Printing-Foundation-LPRPortMonitor" /all
exit /b

:win7
rem windows7
dism /online /enable-feature /featurename:"Printing-Foundation-Features" /norestart
dism /online /enable-feature /featurename:"Printing-Foundation-LPDPrintService" /norestart
dism /online /enable-feature /featurename:"Printing-Foundation-LPRPortMonitor" /norestart
dism /online /enable-feature /featurename:"Printing-Foundation-InternetPrinting-Client"
/norestart
exit /b


:set_printer
setlocal enabledelayedexpansion

rem 列出所有打印机
```

```
echo "正在枚举所有打印机..."
set counter=0
for /f "tokens=2 delims==" %%a in ('wmic printer get name /value^|find "="') do (
  set /a counter+=1
  set printer[!counter!]=%%a
  echo !counter!. %%a
)

rem 用户选择默认打印机
set /p choice="请输入要设置为默认打印机的编号："
if %choice% gtr %counter% exit /b
set defaultprinter=!printer[%choice%]!
echo 将 "%defaultprinter%" 设置为默认打印机...
rundll32 printui.dll,PrintUIEntry /y /n "%defaultprinter%"

rem 删除多余的打印机
set skipdelete=
set /p deletelabel="是否删除不需要的打印机？（y/n）："
if /i "%deletelabel%"=="y" (
    set skipdelete=1
    echo 正在准备删除多余的打印机...
    for /f "tokens=2 delims==" %%a in ('wmic printer get default /value^|find "="') do (
        set defaultprinter=%%a
    )
    for /f "tokens=2 delims==" %%a in ('wmic printer get name^|find ":"') do (
        if /i not "%%a"=="%defaultprinter%" (
            rundll32 printui.dll,PrintUIEntry /dl /n "%%a"
            echo 已删除打印机 "%%a".
        )
    )
)
```

```
if not defined skipdelete (
    echo 因您选择了“否”, 将保留所有打印机.
)

echo 打印机设置完成。
timeout 1 /nobreak
exit /b

:add_lpd_printer
echo 正在配置防火墙...
netsh advfirewall firewall add rule name="LPR Port" dir=in action=allow protocol=TCP localport=515
netsh advfirewall firewall add rule name="LPD Port" dir=in action=allow protocol=TCP localport=721-731
echo 防火墙已经配置完成!
set /p ip=请输入要添加的LPD打印机ip:
set /p queue=请输入要添加的打印队列(共享)名称(不能用空格):
set /p p_inf=请输入要添加的驱动文件路径(以inf结尾):
rundll32 printui.dll,PrintUIEntry /if /b "%queue%" /f "%p_inf%" /r "IP_%ip%" /m "LPD Printer" /z
echo 操作完成!
exit /b


:del_jobs
echo 停止打印服务...
echo Y|net stop spooler
echo 删除所有打印任务...
del /Q /F /S "%SystemRoot%\System32\spool\PRINTERS\*"
echo 启动打印服务
net start spooler
```

```
echo 打印任务清理完成!
timeout 1 /nobreak
exit /b

:fix_smb_printer
rem 要通过注册表切换网络打印设置
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Printers\RPC" /v
RpcUseNamedPipeProtocol /t REG_DWORD /d 1 /f
rem 要启用侦听传入连接
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Printers\RPC" /v
RpcProtocols /t REG_DWORD /d 0x7 /f
rem 要强制执行 Kerberos 身份验证,
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Printers\RPC" /v
ForceKerberosForRpc /t REG_DWORD /d 1 /f
rem 其它注册表(别人提供)
Reg add "HKLM\System\CurrentControlSet\Control\Print" /v "RpcAuthnLevelPrivacyEnabled" /t
REG_DWORD /d "0" /f
Reg add "HKLM\Software\Policies\Microsoft\Windows NT\Printers\PointAndPrint" /v
"RestrictDriverInstallationToAdministrators" /t REG_DWORD /d "0" /f
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Printers\RPC" /v
"RpcUseNamedPipeProtocol" /t REG_DWORD /d 1 /f

gpupdate /force
echo 开启SMB1 支持
DISM /Online /Enable-Feature /FeatureName:SMB1Protocol /norestart
reg add "HKLM\System\CurrentControlSet\Services\LanmanWorkstation\Parameters" /v
RequireSecuritySignature /t REG_DWORD /d 0 /f
reg add "HKLM\System\CurrentControlSet\Services\LanmanWorkstation\Parameters" /v
EnableForcedLogoff /t REG_DWORD /d 0 /f
exit /b
```

```
:cgname
set now=chobit%date:¯2,2%%time:¯9,10%
echo 当前计算机名为%computername%,    按回车改为%now%
set /p hostname=请输入修改后的计算机名:
if "%hostname%"=="" set hostname=%now%
echo 修改计算机名为: %hostname% 按回车修改
@echo 修改中……
@echo off

reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName" /v
"ComputerName" /t REG_SZ /d "%hostname%" /f
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ComputerName\ActiveComputerName"
/v "ComputerName" /t REG_SZ /d "%hostname%" /f
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters" /v "Hostname"
/t REG_SZ /d "%hostname%" /f
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters" /v "NV
Hostname" /t REG_SZ /d "%hostname%" /f
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" /v
"AltDefaultDomainName" /t REG_SZ /d "%hostname%" /f
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" /v
"DefaultDomainName" /t REG_SZ /d "%hostname%" /f
echo y|net stop WlanSvc>nul
echo y|net stop Wcmsvc>nul
echo y|net start Wcmsvc>nul
echo y|net start WlanSvc>nul
@echo 修改计算机名为%hostname%完成!
set nowcg=已修改为
echo 测试是否生效的方法:在其它计算机访问\\%hostname%或ping %hostname%
exit /b
```

```
fixed_out = {}
table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\向日葵.png><a oray><c #000000>向日葵</c></a> \n十几年历史的远程软件")
            )
table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\向日葵.png><a oraynew><c #000000>向日葵(新版)</c></a> \n 该版本需要登陆!")
            )
table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\ToDesk.png><a todesk><c #000000>ToDesk完整版</c></a> \n 近几年新出的远程软件,完整版")
            )

table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\ToDesk.png><a todesklite><c #000000>ToDeskLite</c></a> \n 近几年新出的远程软件,精简版受控端")
            )

table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\dbadmin.png><a dbadmin><c #000000>深蓝远程</c></a> \n 震精网吧界的老牌远程工具,作者一直保证性欲")
            )

table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\Alpemix远程.png><a Alpemix><c
#000000>Alpemix</c></a> \n Alpemix远程是国外的免费远程软件")
            )
                        --"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\remote.lua"
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="12" size="12" name="Tahoma" default="true"/>
    <Font shared="true" id="16" size="16" name="Tahoma"/>
    <Font shared="true" id="24" size="24" name="Tahoma"/>

    <MultiLanguage id="info_w0" value="Воскресенье"/>
    <MultiLanguage id="info_w1" value="Понедельник"/>
    <MultiLanguage id="info_w2" value="Вторник"/>
    <MultiLanguage id="info_w3" value="Среда"/>
    <MultiLanguage id="info_w4" value="Четверг"/>
    <MultiLanguage id="info_w5" value="Пятница"/>
    <MultiLanguage id="info_w6" value="Суббота"/>
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="16" name="Tahoma"  shared="true"  default="true" />
    <Font id="24" size="24" name="Tahoma"  shared="true" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
    <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
   <Font shared="true" id="14" size="14" name="Tahoma" default="true" />
   <Font shared="true" id="24" size="24" name="Tahoma" />
   <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
   <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />

   <MultiLanguage id="Settings" value="Настройки" />
   <MultiLanguage id="Home" value="Домой" />
   <MultiLanguage id="Customization" value="Настройка" />

   <MultiLanguage id="Display" value="Дисплей" />
   <MultiLanguage id="DPI" value="Масштаб" />
   <MultiLanguage id="Resolution" value="Разрешение" />
   <MultiLanguage id="Rotate" value="Поворот" />
   <MultiLanguage id="Rotate_0" value="0 градусов" />
   <MultiLanguage id="Rotate_90" value="90 градусов" />
   <MultiLanguage id="Rotate_180" value="180 градусов" />
   <MultiLanguage id="Rotate_270" value="270 градусов" />

   <MultiLanguage id="Brightness" value="Яркость" />

   <MultiLanguage id="Apply" value="<u>П</u>рименить" />
   <MultiLanguage id="Restore" value="<u>В</u>осстановить" />

   <MultiLanguage id="Background" value="Фон" />

   <MultiLanguage id="Colors" value="Цвета" />
   <MultiLanguage id="SelectSysColorTheme" value="Тема Windows по умолчанию" />
   <MultiLanguage id="SelectAppColorTheme" value="Тема приложений по умолчанию" />
   <MultiLanguage id="Light" value="Светлая" />
   <MultiLanguage id="Dark" value="Тёмная" />
   <MultiLanguage id="TransparencyEffects" value="Эффекты прозрачности" />
   <MultiLanguage id="ApplyAccentColorDesc" value="Акцент" />
   <MultiLanguage id="StartTaskbarActionCenter" value="Пуск, панель задач и центр действий" />
   <MultiLanguage id="TitlebarsAndWinBorders" value="Заголовки и границы окон" />

   <MultiLanguage id="Taskbar" value="Панель задач" />
   <MultiLanguage id="AutoHideTaskbar" value="Автоматически скрывать панель задач" />
   <MultiLanguage id="UseSmallButtons" value="Использовать маленькие кнопки на панели задач"
/>
   <MultiLanguage id="CombineButtons" value="Группировать кнопки панели задач" />
   <MultiLanguage id="Always" value="Всегда, скрывать метки" />
   <MultiLanguage id="Auto" value="Когда панель задач заполнена" />
   <MultiLanguage id="Never" value="Никогда" />

   <MultiLanguage id="On" value="Вкл." />
   <MultiLanguage id="Off" value="Выкл." />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="16" name="Tahoma"  shared="true"  default="true" />
    <Font id="24" size="24" name="Tahoma"  shared="true" />

    <MultiLanguage id="Settings" value="Настройки" />
    <MultiLanguage id="Display Settings" value="Настройки дисплея" />
    <MultiLanguage id="Resolution" value="Разрешение" />
    <MultiLanguage id="Rotate" value="Поворот" />
    <MultiLanguage id="Rotate_0" value="0 градусов" />
    <MultiLanguage id="Rotate_90" value="90 градусов" />
    <MultiLanguage id="Rotate_180" value="180 градусов" />
    <MultiLanguage id="Rotate_270" value="270 градусов" />
    <MultiLanguage id="Brightness" value="Яркость" />
    <MultiLanguage id="Apply" value="<u>П</u>рименить" />
    <MultiLanguage id="Close" value="<u>З</u>акрыть" />
    <MultiLanguage id="Restore" value="<u>В</u>осстановить" />
    <MultiLanguage id="Cancel" value="<u>О</u>тменить" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
        <Font id="16" size="16" name="Tahoma" shared="true" default="true"/>
        <Font id="24" size="24" name="Tahoma" shared="true"/>

        <!-- The description of <Style imagerectstyle="l,t,w,h" /> in main.xml -->
        <!-- image's source, dest attributes will define with "left,top,width,height" style -->
        <!-- the origin style is "left,top,right,bottom". -->
        <!-- the xml file for "Include" tag is also use the same "imagerectstyle" -->

        <MultiLanguage id="Power" value="Питание"/>
        <MultiLanguage id="CloseComputer" value="Закрытие"/>
        <MultiLanguage id="Shutdown" value="Выключение"/>
        <MultiLanguage id="Restart" value="Перезагрузка"/>
        <MultiLanguage id="Hibernate" value="Гибернация"/>
        <MultiLanguage id="Standby" value="Режим ожидания"/>
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="12" size="12" name="Tahoma"  shared="true"  default="true" />
    <Font id="14" size="14" name="Tahoma"  shared="true" />
    <Font id="16" size="16" name="Tahoma"  shared="true" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
	<Font shared="true" id="16" size="16" name="Tahoma" default="true"/>
	<Font shared="true" id="16b" size="16" name="Tahoma" bold="true"/>
	<Font shared="true" id="18" size="18" name="Tahoma"/>
	<Font shared="true" id="18b" size="18" name="Tahoma" bold="true"/>
	<Font shared="true" id="20" size="20" name="Tahoma"/>
	<Font shared="true" id="40" size="40b" name="Tahoma"/>
	<Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets"/>

	<MultiLanguage id="info_w0" value="Воскресенье"/>
	<MultiLanguage id="info_w1" value="Понедельник"/>
	<MultiLanguage id="info_w2" value="Вторник"/>
	<MultiLanguage id="info_w3" value="Среда"/>
	<MultiLanguage id="info_w4" value="Четверг"/>
	<MultiLanguage id="info_w5" value="Пятница"/>
	<MultiLanguage id="info_w6" value="Суббота"/>

	<MultiLanguage id="w7" value="ВС"/>
	<MultiLanguage id="w1" value="ПН"/>
	<MultiLanguage id="w2" value="ВТ"/>
	<MultiLanguage id="w3" value="СР"/>
	<MultiLanguage id="w4" value="ЧТ"/>
	<MultiLanguage id="w5" value="ПТ"/>
	<MultiLanguage id="w6" value="СБ"/>

	<MultiLanguage id="OK" value="Применить"/>
	<MultiLanguage id="CANCEL" value="Отменить"/>
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="16" name="Tahoma"  shared="true"  default="true" />
    <Font id="24" size="24" name="Tahoma"  shared="true" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="10" size="10" name="Tahoma" shared="true" />
    <Font id="16" size="16" name="Tahoma" shared="true" default="true" />
    <Font id="12" size="12" name="Tahoma" shared="true" />
    <Font id="24" size="24" name="Tahoma" shared="true" />
    <Font id="16u" size="16" name="Tahoma" shared="true" underline="true" />

    <MultiLanguage id="::Connected" value="Подключено" />
    <MultiLanguage id="::Secured" value="Защищено" />
    <MultiLanguage id="::Open" value="Открыть" />
    <MultiLanguage id="::Connect" value="Подключить" />
    <MultiLanguage id="::Disconnect" value="Отключить" />
    <MultiLanguage id="msg_hidden_ssid" value="Скрытая сеть" />
    <MultiLanguage id="msg_enter_ssid" value="Введите имя сети" />
    <MultiLanguage id="msg_enter_password" value="Введите пароль сети" />
    <MultiLanguage id="Cancel" value="Отмена" />
    <MultiLanguage id="NetworkSettings" value="Настройки сети и &Интернета" />
    <MultiLanguage id="NetworkSettingsMsg" value="Изменение настроек, например, установка статического IP" />
    <MultiLanguage id="Firewall" value="Брандмауэр" />
    <MultiLanguage id="Proxy" value="Прокси" />
    <MultiLanguage id="::msg_select_device" value="Выберите устройство WLAN" />
    <MultiLanguage id="::msg_service_unavailable" value="Служба WLAN недоступна" />
    <MultiLanguage id="::msg_device_unavailable" value="Устройство WLAN недоступно" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="18" name="Tahoma" shared="true" default="true"/>
    <Font id="24" size="24" name="Tahoma" shared="true"/>
    <Font shared="true" id="40b" size="40b" name="Tahoma"/>
    <Font shared="true" id="128" size="28" name="Segoe MDL2 Assets"/>
    <Font shared="true" id="130" size="30" name="Segoe MDL2 Assets"/>

    <MultiLanguage id="info_w0" value="Воскресенье"/>
    <MultiLanguage id="info_w1" value="Понедельник"/>
    <MultiLanguage id="info_w2" value="Вторник"/>
    <MultiLanguage id="info_w3" value="Среда"/>
    <MultiLanguage id="info_w4" value="Четверг"/>
    <MultiLanguage id="info_w5" value="Пятница"/>
    <MultiLanguage id="info_w6" value="Суббота"/>

    <MultiLanguage id="SignIn" value="Войти"/>
    <MultiLanguage id="ForgotPassword" value="Забыл пароль"/>
    <MultiLanguage id="Logon" value="Вход в систему"/>

    <MultiLanguage id="will autologon with the $user account in $countdown second(s)"
value="автоматически войдёт в систему с аккаунтом $user через $countdown сек."/>
    <MultiLanguage id="Welcome to Windows ..." value="Добро пожаловать в Windows..."/>
    <MultiLanguage id="LogonInfo" value="Войдите в учетную запись SYSTEM, нажмите Alt + S, и
войдите в учетную запись Administrator, нажмите Alt + A." />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="16" size="16" name="Tahoma" default="true"/>
    <Font shared="true" id="16b" size="16" name="Tahoma" bold="true"/>
    <Font shared="true" id="18" size="18" name="Tahoma"/>
    <Font shared="true" id="18b" size="18" name="Tahoma" bold="true"/>
    <Font shared="true" id="20" size="20" name="Tahoma"/>
    <Font shared="true" id="40" size="40b" name="Tahoma"/>
    <Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets"/>

    <MultiLanguage id="info_w0" value="Воскресенье"/>
    <MultiLanguage id="info_w1" value="Понедельник"/>
    <MultiLanguage id="info_w2" value="Вторник"/>
    <MultiLanguage id="info_w3" value="Среда"/>
    <MultiLanguage id="info_w4" value="Четверг"/>
    <MultiLanguage id="info_w5" value="Пятница"/>
    <MultiLanguage id="info_w6" value="Суббота"/>

    <MultiLanguage id="w7" value="ВС"/>
    <MultiLanguage id="w1" value="ПН"/>
    <MultiLanguage id="w2" value="ВТ"/>
    <MultiLanguage id="w3" value="СР"/>
    <MultiLanguage id="w4" value="ЧТ"/>
    <MultiLanguage id="w5" value="ПТ"/>
    <MultiLanguage id="w6" value="СБ"/>

    <MultiLanguage id="OK" value="Применить"/>
    <MultiLanguage id="CANCEL" value="Отменить"/>
</Locale>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ScrollbarConfig>
   <Image shared="true" name="scrollbar.bmp" />
   <Default shared="true" name="VScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,90,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,90,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,90,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,90,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,108,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,108,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,108,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,108,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
   <Default shared="true" name="HScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,0,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,0,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,0,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,0,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,18,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,18,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,18,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,18,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
</ScrollbarConfig>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ScrollbarConfig>
	<Image shared="true" name="scrollbar.bmp" />
	<Default shared="true" name="VScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,90,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,90,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,90,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,90,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,108,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,108,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,108,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,108,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
	<Default shared="true" name="HScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,0,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,0,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,0,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,0,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,18,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,18,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,18,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,18,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
</ScrollbarConfig>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ScrollbarConfig>
	<Image shared="true" name="scrollbar.bmp" />
	<Default shared="true" name="VScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,90,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,90,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,90,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,90,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,108,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,108,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,108,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,108,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
	<Default shared="true" name="HScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,0,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,0,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,0,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,0,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,18,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,18,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,18,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,18,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
</ScrollbarConfig>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ScrollbarConfig>
   <Image shared="true" name="scrollbar.bmp" />
   <Default shared="true" name="VScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,90,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,90,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,90,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,90,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,108,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,108,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,108,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,108,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
   <Default shared="true" name="HScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,0,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,0,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,0,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,0,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,18,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,18,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,18,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,18,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
</ScrollbarConfig>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ScrollbarConfig>
   <Image shared="true" name="scrollbar.bmp" />
   <Default shared="true" name="VScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,90,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,90,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,90,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,90,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,108,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,108,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,108,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,108,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,126,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,144,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,162,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
   <Default shared="true" name="HScrollBar" value="button1normalimage=&quot;file='scrollbar.bmp'
source='0,0,16,16' mask='#FFFF00FF'&quot; button1hotimage=&quot;file='scrollbar.bmp' source='18,0,16,16'
mask='#FFFF00FF'&quot; button1pushedimage=&quot;file='scrollbar.bmp' source='36,0,16,16'
mask='#FFFF00FF'&quot; button1disabledimage=&quot;file='scrollbar.bmp' source='54,0,16,16'
mask='#FFFF00FF'&quot; button2normalimage=&quot;file='scrollbar.bmp' source='0,18,16,16'
mask='#FFFF00FF'&quot; button2hotimage=&quot;file='scrollbar.bmp' source='18,18,16,16'
mask='#FFFF00FF'&quot; button2pushedimage=&quot;file='scrollbar.bmp' source='36,18,16,16'
mask='#FFFF00FF'&quot; button2disabledimage=&quot;file='scrollbar.bmp' source='54,18,16,16'
mask='#FFFF00FF'&quot; thumbnormalimage=&quot;file='scrollbar.bmp' source='0,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbhotimage=&quot;file='scrollbar.bmp' source='18,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbpushedimage=&quot;file='scrollbar.bmp' source='36,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; thumbdisabledimage=&quot;file='scrollbar.bmp' source='54,36,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railnormalimage=&quot;file='scrollbar.bmp' source='0,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railhotimage=&quot;file='scrollbar.bmp' source='18,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; railpushedimage=&quot;file='scrollbar.bmp' source='36,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; raildisabledimage=&quot;file='scrollbar.bmp' source='54,54,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bknormalimage=&quot;file='scrollbar.bmp' source='0,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkhotimage=&quot;file='scrollbar.bmp' source='18,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkpushedimage=&quot;file='scrollbar.bmp' source='36,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; bkdisabledimage=&quot;file='scrollbar.bmp' source='54,72,16,16' corner='2,2,2,2'
mask='#FFFF00FF'&quot; " />
</ScrollbarConfig>
```

WinXShell

- Project
- Documentation

Contents

- 外壳 对象
- 桌面 对象
- 任务栏 对象
- 开始菜单 对象

Classes

- App
- **Shell**
- System

Libraries

- Lua-Engine
- UI-Engine
- Windows-Shell

Program

- WinXShell

Manual

- CSIDL
- 更新历史记录

Class `Shell`

Shell 相关对象

外壳 对象

Shell:Close ()	关闭外壳窗口
Shell:WaitAndClose ()	关闭外壳窗口 外壳窗口不存在时，将等待外壳窗口出现后再关闭

桌面 对象

Desktop:Link (lnk, target[, param[, icon[, index[, showcmd]]]])	创建 快捷方式到桌面
Desktop:Refresh ()	刷新 桌面

Desktop:GetWallpaper ()	获取 壁纸路径
Desktop:SetWallpaper (path)	设置 桌面壁纸
Desktop:SetIconSize (level)	设置 桌面图标大小
Desktop:AutoArrange (checked)	设置 自动排列图标 选项
Desktop:SnapToGrid (checked)	设置 将图标与网格对齐 选项
Desktop:ShowIcons (checked)	设置 显示桌面图标 选项

任务栏 对象

Taskbar:IsReady (sec)	检测 任务栏是否准备就绪
Taskbar:WaitForReady ()	等待 任务栏准备就绪
Taskbar:GetSetting (key)	获取 任务栏相关设置
Taskbar:SetSetting (key, value, type)	设定 任务栏相关设置
Taskbar:CombineButtons (value[, update])	更改 任务栏合并按钮选项
Taskbar:UseSmallIcons (value[, update])	更改 任务栏使用小图标选项
Taskbar:AutoHide (value)	更改 任务栏自动隐藏选项
Taskbar:Hide ()	隐藏 任务栏
Taskbar:Show ()	显示 任务栏
Taskbar:Pin (target[, name[, param[, icon[, index[, showcmd]]]]])	固定 程序到任务栏

开始菜单 对象

Startmenu:Pin (target[, name[, param[, icon[, index[, showcmd]]]]])	固定 程序到开始菜单
Startmenu:Link (lnk, target[, param[, icon[, index[, showcmd]]]])	创建 快捷方式到开始菜单

外壳 对象

Shell:Close ()

关闭外壳窗口

Shell:WaitAndClose ()

关闭外壳窗口 外壳窗口不存在时，将等待外壳窗口出现后再关闭

桌面 对象

Desktop:Link (lnk, target[, param[, icon[, index[, showcmd]]]])

创建 快捷方式到桌面

Parameters:

- lnk string 指定要创建的快捷方式的路径
- target string 快捷方式的目标路径
- param string 程序执行参数 (*optional*)
- icon string 指定任务栏图标(ico文件，或者包含图标的资源文件) (*optional*)
- index int 图标索引号 (*optional*)
- showcmd string 启动参数 (*optional*)

See also:

os.link

Usage:

```
Desktop:Link('Hello.lnk', 'cmd.exe', '/k echo hello', 'shell32.dll', 27)
Desktop:Link('#{@shell32.dll,22022}.lnk', 'cmd.exe')
Desktop:Link('Internet Explorer.lnk', '%ProgramFiles%\\Internet
Explorer\\iexplore.exe')
Desktop:Link([[Tools\Everything.lnk]], [[%ProgramFile%\Everything\Everything.exe]])
```

Desktop:Refresh ()

刷新 桌面

Desktop:GetWallpaper ()

获取 壁纸路径

Returns:

壁纸路径

Desktop:SetWallpaper (path)

设置 桌面壁纸

Parameters:

- path string 壁纸文件

Usage:

```
WinXShell.exe -code Desktop:SetWallpaper([[%SystemRoot%\Web\img0.jpg]])
```

Desktop:SetIconSize (level)

设置 桌面图标大小

Parameters:

- level string

  图标大小级别。

  - 'S' - 小图标
  - 'M' - 中图标
  - 'L' - 大图标

Usage:

```
WinXShell.exe -code Desktop:SizeIconSize('M')
```

Desktop:AutoArrange (checked)

设置 自动排列图标 选项

Parameters:

- checked int

  是否使用此选项。

  - 0 - 不自动排列
  - 1 - 自动排列

Desktop:SnapToGrid (checked)

设置 将图标与网格对齐 选项

Parameters:

- checked int

  是否使用此选项。

  - 0 - 不与网格对齐
  - 1 - 与网格对齐

Desktop:ShowIcons (checked)

设置 显示桌面图标 选项

Parameters:

- checked int

  是否使用此选项。

  - 0 - 不显示
  - 1 - 显示

任务栏 对象

Taskbar:IsReady (sec)

检测 任务栏是否准备就绪 指定sec秒内判断任务栏是否准备就绪(sec为-1时，直到任务栏准备好为止)

Parameters:

- sec int 设置检测间隔(秒数)

Taskbar:WaitForReady ()

等待 任务栏准备就绪 用于控制启动时，一些需要在外壳创建后才进行初始化任务执行。 内部调用Taskbar:IsReady(-1)方法

Taskbar:GetSetting (key)

获取 任务栏相关设置

Parameters:

- key string 设置项目名

Taskbar:SetSetting (key, value, type)

设定 任务栏相关设置

Parameters:

- key string 设置项目名
- value 设定值
- type 类型

Taskbar:CombineButtons (value[, update])

更改 任务栏合并按钮选项

Parameters:

- value string

  合并按钮选项

  - 'always' - 总是合并
  - 'auto' - 自动合并
  - 'never' - 从不合并
- update int 是否立即更新任务栏(非0时， 立即更新) (*optional*)

Taskbar:UseSmallIcons (value[, update])

更改 任务栏使用小图标选项

Parameters:

- value int

  使用小图标选项

  - 0 - 不使用小图标
  - 1 - 使用小图标
- update int 是否立即更新任务栏(非0时， 立即更新) (*optional*)

Taskbar:AutoHide (value)

更改 任务栏自动隐藏选项

Parameters:

- value int

  自动隐藏选项

  - 0 - 不自动隐藏
  - 1 - 自动隐藏

Taskbar:Hide ()

隐藏 任务栏

Taskbar:Show ()

显示 任务栏

Taskbar:Pin (target[, name[, param[, icon[, index[, showcmd]]]]])

固定 程序到任务栏

Parameters:

- target string 指定需要固定的程序路径
- name string 固定图标名称(鼠标悬停时显示), 不指定时使用程序自身描述信息 (*optional*)
- param string 程序执行参数 (*optional*)
- icon string 指定任务栏图标(ico文件，或者包含图标的资源文件) (*optional*)
- index int 图标索引号 (*optional*)
- showcmd string 启动参数 (*optional*)

Usage:

```
WinXShell.exe -code Taskbar:Pin('regedit.exe')
WinXShell.exe -code "Taskbar:Pin([[%ProgramFile%\Everything\Everything.exe]])"
WinXShell.exe -code "Taskbar:Pin('cmd.exe', nil, '/k echo hello', 'shell32.dll', 27)"
```

开始菜单 对象

Startmenu:Pin (target[, name[, param[, icon[, index[, showcmd]]]]])

固定 程序到开始菜单

Parameters:

- target string 指定需要固定的程序路径
- name string 固定图标名称(鼠标悬停时显示), 不指定时使用程序自身描述信息 (*optional*)
- param string 程序执行参数 (*optional*)
- icon string 指定任务栏图标(ico文件，或者包含图标的资源文件) (*optional*)
- index int 图标索引号 (*optional*)
- showcmd string 启动参数 (*optional*)

Usage:

```
WinXShell.exe -code Startmenu:Pin('regedit.exe')
WinXShell.exe -code "Startmenu:Pin([[%ProgramFile%\Everything\Everything.exe]])"
WinXShell.exe -code "Startmenu:Pin('cmd.exe', nil, '/k echo hello', 'shell32.dll',
27)"
```

Startmenu:Link (lnk, target[, param[, icon[, index[, showcmd]]]])

创建 快捷方式到开始菜单

Parameters:

- lnk string 指定要创建的快捷方式的路径
- target string 快捷方式的目标路径
- param string 程序执行参数 (*optional*)
- icon string 指定任务栏图标(ico文件，或者包含图标的资源文件) (*optional*)
- index int 图标索引号 (*optional*)
- showcmd string 启动参数 (*optional*)

See also:

os.link

Usage:

```
Startmenu:Link('Hello.lnk', 'cmd.exe', '/k echo hello', 'shell32.dll', 27)
Startmenu:Link('Microsoft Edge.lnk',
[[%ProgramFiles%\Microsoft\Edge\Application\msedge.exe]])
Startmenu:Link([[Tools\Everything.lnk]], [[%ProgramFile%\Everything\Everything.exe]])
```

generated by LDoc 1.4.6

```
<?xml version="1.0" encoding="utf-8"?>
<SubXML>
    <ListContainerElement height="60">
        <VerticalLayout>
            <HorizontalLayout height="60" childpadding="5">
                <Label name="$ssidsignal" bkimage="rc\100.png" height="32" width="32" />
                <VerticalLayout>
                    <Label name="$ssidname"/>
                    <Label name="$ssidstatus" style="ct-text-secondary" />
                </VerticalLayout>
            </HorizontalLayout>
            <VerticalLayout name="$ssidpwddlg" visible="false" childpadding="5" padding="32,0,20,0">
                <VerticalLayout name="$ssidpwddlg_ssid" height="65" childpadding="5" visible="false">
                    <Label text="%{msg_enter_ssid}" />
                    <Edit name="$ssidpwd_ssid" height="30" textcolor="#FF000000" />
                </VerticalLayout>
                <Label text="%{msg_enter_password}"/>
                <HorizontalLayout height="31" >
                    <Edit name="$ssidpwd_pwd" height="30" textcolor="#FF000000" password="true" />
                    <Button name="$ssidpwd_view" height="30" width="30" bkimage="rc\pwd_asterisk.png"
/>
                </HorizontalLayout>
                <HorizontalLayout childpadding="5">
                    <Button style="ct-button" bordersize="2" name="$ssidpwd_conn" text="%{::Connect}"
/>
                    <Button style="ct-button" bordersize="2" name="launch_osk" text="OSK" />
                    <Button style="ct-button" bordersize="2" name="$ssidpwd_cancel" text="%{Cancel}" />
                </HorizontalLayout>
            </VerticalLayout>
            <Button style="ct-button" name="$ssidconn" visible="false" text="%{::Connected}"
                 bordersize="2" padding="120,0,20,0" />
        </VerticalLayout>
    </ListContainerElement>
</SubXML>
```

```
{
 "name":"UI_Launcher",
 "singleton":true,
 "nobaricon":true,
 "position":"leftbottom",
 "OnDeactive":"hide",
 "OnSingleton":"toggle"
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Launcher\startmenu.jcfg",
}
```

```
require 'winapi'

winapi.set_encoding(winapi.CP_UTF8)

function winapi.temp_name() return os.tmpname() end -- overwrite winapi's bug temp_name()

function winapi.files(mask,subdirs,attrib)
    local flags = '/B '
    if subdirs then flags = flags..' /S' end
    if attrib then flags = flags..' /A:'..attrib end
    local ret, text = winapi.execute('dir '..flags..' "'..mask..'"','ansi') --unicode
    if ret ~= 0 then return nil,text end
    return text:gmatch('[^\r\n]+')
end

app_folders = {}

local function bind_to_folder(fpath, tbl)
    root = tbl
    local paths = fpath:gmatch('[^\\]+')
    local fname
    for pname in paths do
      fname = pname
      if root[pname] then
         root = root[pname]
      end
    end
    root[fname] = fpath
end

local function load_programs(drv)
  app_folders[drv] = {}
  local t = app_folders[drv]

  -- create the dir struct first
  local root_path = drv .. [[:\ProgramData\Microsoft\Windows\Start Menu\Programs\]]
  local items, err = winapi.files(root_path .. '*.*', true, 'D')
  if not items then return print(err) end

  local rpath, paths
  local root = nil
  for f in items do
    rpath = f:gsub(root_path, '')
    root = t
    paths = rpath:gmatch('[^\\]+')
    for pname in paths do
      if root[pname] then
        root = root[pname]
      else
        root[pname] = {}
      end
    end
  end

  items, err = winapi.files(root_path .. '*.*', true, '-D')
  if not items then return print(err) end
  local lpath = ''
  for f in items do
    rpath = f:gsub(root_path, '')
    if not rpath:lower():match('\\desktop.ini') then
      bind_to_folder(rpath, t)
    end
  end
  -- table.print(app_folders['C'])
end

local function pairsByKeys (t, f)
  local a = {}
  for n in pairs(t) do table.insert(a, n) end
  table.sort(a, f)
  local i = 0      -- iterator variable
  local iter = function ()   -- iterator function
    i = i + 1
    if a[i] == nil then return nil
    else return a[i], t[a[i]]
    end
  end
  return iter
end

local function space(n)
  return string.rep('   ', n)
end

local xml = ''
local function app_walk(t, level, fpath)
    local str = ''
    local vis = [[folderattr="selected=&quot;true&quot;"]]
    if level > 0 then vis = [[folderattr="selected=&quot;false&quot;" visible="false"]] end
    for k, v in pairsByKeys(t) do
        if type(v) == 'table' then
            print(space(level) .. k)
            str = string.format([[<TreeNode text="%s" %s style="folder_node" >]] .. '\n', k, vis)
            xml = xml .. space(level) .. str
            app_walk(v, level + 1, fpath .. '\\' .. k)
            xml = xml .. space(level) .. '</TreeNode>\n'
        else
            print(string.rep('   ', level) .. k)
            xml = xml .. space(level) ..
             string.format([[<TreeNode text="%s" %s style="item_node" itemattr="bkimage=&quot;file='%s,F'
dest='4,4,28,28'&quot;" />]] .. '\n', k, vis, fpath .. '\\' .. k)
        end
    end
end

load_programs('C')
local tree = app_folders['C']
app_walk(tree, 0, [[C:\ProgramData\Microsoft\Windows\Start Menu\Programs]])
print(xml)
--table.print(tree)
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Launcher\startmenu.lua"
```

```
temp = os.getenv("temp")

exitcode, nwinfo_stdout = winapi.execute("nwinfo --cp=UTF8 --sys --format=lua")

nwinfo_stdout =  nwinfo_stdout:gsub("#!lua", "")

load(nwinfo_stdout)()

sys_cmdline = _NWINFO.System["Start Options"]


if sys_cmdline:find("@") then

local my_cmdline = string.match(sys_cmdline, ′MININT(.*)RDIMAGEOFFSET′)

my_cmdline = string.gsub(my_cmdline, "^%s*(.-)%s*¥", "%1")  -- 去除字符串两端的空格

alert("发现系统需要执行任务:" .. my_cmdline)

else

return

end

--alert(_NWINFO.System["Start Options"])

--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\StartOptions.lua"
```

WinXShell

- Project
- Documentation

Contents

Classes

- App
- Shell
- **System**

Libraries

- Lua-Engine
- UI-Engine
- Windows-Shell

Program

- WinXShell

Manual

- CSIDL
- 更新历史记录

Class `System`

系统 相关对象

系统 对象

System:NetJoin ()	加入 工作组或域
System:EnableEUDC (fEnableEUDC)	开启 用户自定义外字支持
System:GetSettings (key)	获取 系统相关设置

System:SetSettings (key, val)	设定 系统相关设置
System:SysColorTheme (mode)	更改 系统颜色主题
System:AppsColorTheme (mode)	更改 应用颜色主题
System:ReloadCursors ()	刷新 鼠标指针式样
System:CreatePageFile (file, min, max)	创建 分页文件(虚拟内存)
System:Reboot ()	重启 计算机
System:Shutdown ()	关闭 计算机

注册表 对象

Reg:Read (key, values)	读取 注册表信息
Reg:Write (key, value, data[, type=REG_SZ])	更新 注册表信息
Reg:GetSubKeys (key)	获取 注册表子项目
Reg.Type	注册表数据类型(常量)

屏幕 对象

Screen:Get (key)	获取 屏幕信息
Screen:Set (key, val)	设置 屏幕参数
Screen:GetX ()	获取 屏幕分辨率宽度
Screen:GetY ()	获取 屏幕分辨率高度
Screen:GetRotation ()	获取 屏幕旋转状态
Screen:GetDPI ()	获取 屏幕DPI
Screen:DPI (scale)	设置 屏幕DPI
Screen:Disp ([w[, h]])	设置 屏幕分辨率
Screen:DispTest (arr)	设置 屏幕分辨率(组)

文件夹选项 对象

FolderOptions:Get (opt)	获取 文件夹选项
FolderOptions:Set (opt, val)	更改 文件夹选项
FolderOptions:Toggle (opt)	切换 文件夹选项

对话框 对象

Dialog:Show (caption, msg[, btns[, icon]])	弹出提示 对话框
Dialog:OpenFile (title, filters[, initDir])	显示 文件打开对话框
Dialog:BrowseFolder (title[, csidl])	显示 文件夹选择对话框

磁盘驱动器 库

Disk.IsLocked (drive)	判断 驱动器是否处于BitLocker加密状态

文件/目录 库

File.Exists (path)	判断 文件是否存在

Folder.Exists (path)	判断 目录是否存在
os.exists (path)	判断 文件或目录是否存在
File.GetFullPath (path)	获取 8.3格式的路径的完整路径格式
File.GetShortPath (path)	获取 8.3格式的短路径
Folder.GetFullPath (path)	获取 8.3格式的路径的完整路径格式
Folder.GetShortPath (path)	获取 8.3格式的短路径

进程/窗口 库

Window.Find (title[, class])	获取 特定标题，类名的窗口对象
Window.Match (title[, class])	获取 匹配特定标题，类名规则的窗口对象

进程 对象

System.Proc	Proc 对象
Proc:Show ([nShow])	显示 窗口
Proc:Hide ()	隐藏 窗口
Proc:IsVisable ()	确认 程序窗口是否处于显示状态
Proc:Activate ()	激活 窗口
Proc:Minimize ()	最小化 窗口
Proc:Maximize ()	最大化 窗口
Proc:Close ()	关闭 窗口
Proc:Quit ()	退出 进程
Proc:Kill ()	结束 进程
Proc:SendMessage (Msg, wParam, lParam)	发送 窗口消息(Send)
Proc:PostMessage (Msg, wParam, lParam)	发送 窗口消息(Post)

系统 对象

System:NetJoin ()

加入 工作组或域

System:EnableEUDC (fEnableEUDC)

开启 用户自定义外字支持

Parameters:

- fEnableEUDC bool

  布尔值可启用 EUDC

  - true - 启用
  - false - 禁用

System:GetSettings (key)

获取 系统相关设置

Parameters:

- key string

设置项目名

- `AppsColorTheme` - 应用程序颜色主题
- `SysColorTheme` - 系统颜色主题
- `ShellColorPrevalence` - 外壳外观颜色
- `WindowColorPrevalence` - 系统外观颜色
- `Colors.Transparency` - 使用透明色

System:SetSettings (key, val)

设定 系统相关设置

Parameters:

- key string

  设置项目名

  - `ShellColorPrevalence` - 外壳外观颜色
  - `WindowColorPrevalence` - 系统外观颜色
  - `Colors.Transparency` - 使用透明色
- val 设定值

System:SysColorTheme (mode)

更改 系统颜色主题

Parameters:

- mode string

  主题模式

  - `light` - 浅色主题
  - `dark` - 深色主题

System:AppsColorTheme (mode)

更改 应用颜色主题

Parameters:

- mode string

  主题模式

  - `light` - 浅色主题
  - `dark` - 深色主题

System:ReloadCursors ()

刷新 鼠标指针式样

System:CreatePageFile (file, min, max)

创建 分页文件(虚拟内存)

Parameters:

- file string 页面文件名
- min int 初始大小(MB)
- max int 最大值(MB)

Usage:

```
WinXShell.exe -code "System:CreatePageFile([[D:\pagefile.sys]], 1024, 4096)"
```

System:Reboot ()

重启 计算机

System:Shutdown ()

关闭 计算机

注册表 对象

Reg:Read (key, values)

读取 注册表信息

Parameters:

- key string 项目名
- values array 键值名 (可读取多个)

Reg:Write (key, value, data[, type=REG_SZ])

更新 注册表信息

Parameters:

- key string 项目名
- value string 键值名
- data 更新数据
- type Reg.Type 类型 (*default* REG_SZ)

Reg:GetSubKeys (key)

获取 注册表子项目

Parameters:

- key string 项目名

Usage:

```
print("GetSubKeys for [HKEY_CLASSES_ROOT\\Folder]:")
local subkeys = Reg:GetSubKeys([[HKEY_CLASSES_ROOT\Folder]])
for i, v in ipairs(subkeys) do
   print(str.fmt("%d:%s", i, v))
end
```

Reg.Type

注册表数据类型(常量)

Fields:

- REG_SZ 字符串。
- REG_DWORD 32位数字。
- REG_EXPAND_SZ 字符串，其中包含对环境变量的未展开的引用(例如，"%PATH%")。
- REG_MULTI_SZ 字符串的序列。由空字符串终止 (\0)。以下是一个示例:
  `String1\0String2\0String3\0LastString\0\0`
- REG_BINARY 任意格式的二进制数据。

屏幕 对象

Screen:Get (key)

获取 屏幕信息

Parameters:

- key string
    - `dpi` - 屏幕DPI
    - `brightness` - 屏幕亮度

Screen:Set (key, val)

设置 屏幕参数

Parameters:

- key string
    - `dpi` - 屏幕DPI
    - `brightness` - 屏幕亮度
- val int

  选项值

    - DPI - `[-1, 100, 125, 150, 175, 200, 250, 300]`
    - 亮度 - `[20 ~ 100]`

Usage:

```
WinXShell.exe -code "Screen:Set('dpi', -1)"
WinXShell.exe -code "Screen:Set('dpi', 125)"
WinXShell.exe -code "Screen:Set('brightness', 80)"
```

Screen:GetX ()

获取 屏幕分辨率宽度

Screen:GetY ()

获取 屏幕分辨率高度

Screen:GetRotation ()

获取 屏幕旋转状态

Screen:GetDPI ()

获取 屏幕DPI

Screen:DPI (scale)

设置 屏幕DPI

Parameters:

- scale int

  缩放大小(百分比)

  可以是以下数值:

    - `-1` - 使用系统推荐缩放
    - `{100, 125, 150, 175, 200, 225, 250, 300, 350, 400, 450, 500}` 中的一个缩放大小

Usage:

```
WinXShell.exe -code Screen:DPI(-1) -- 系统推荐缩放
```

Screen:Disp ([w[, h]])

设置 屏幕分辨率 不指定宽度和高度时，自动使用最大分辨率。

Parameters:

- w int 宽度 (*optional*)
- h int 高度 (*optional*)

Screen:DispTest (arr)

设置 屏幕分辨率(组) 按照指定的分辨率列表逐个尝试设置屏幕分辨率，直到设置成功为止。

Parameters:

- arr string{} 分辨率 宽度x高度 的字符串数组

Usage:

```
WinXShell.exe -code "Screen:DispTest({'1152x864', '1366x768', '1024x768'})"
```

文件夹选项 对象

FolderOptions:Get (opt)

获取 文件夹选项

Parameters:

- opt string

  选项

  - `ShowAll` - 显示隐藏文件
  - `ShowExt` - 显示隐藏已知文件类型的扩展名
  - `ShowSuperHidden` - 显示系统隐藏文件

Returns:

0 或者 1

FolderOptions:Set (opt, val)

更改 文件夹选项

Parameters:

- opt string 选项 (参考 FolderOptions:Get)
- val int 设定值(0 或者 1)

FolderOptions:Toggle (opt)

切换 文件夹选项

Parameters:

- opt string 选项 (参考 FolderOptions:Get)

对话框 对象

Dialog:Show (caption, msg[, btns[, icon]])

弹出提示 对话框

Parameters:

- caption string 标题
- msg string 消息内容
- btns string

  按钮式样

  - `ok` - 确定 (默认), `ok-cancel` - 确定 与 取消,
  - `yes` - 是, `yes-no` - 是 与 否, `yes-no-cancel` - 是, 否 与 取消
  - `abort-retry-ignore` - 中止， 重试 与 忽略, `retry-cancel` - 重试 与 取消,

  (*optional*)
- icon string

  图标式样

  - `information` (默认), `question`, `warning`, `error`

  (*optional*)

Returns:

返回按下的按钮名称:

- `ok`, `yes`, `no`, `cancel`, `try`, `abort` 或者 `retry`

See also:

winapi.show_message

Dialog:OpenFile (title, filters[, initDir])

显示 文件打开对话框

Parameters:

- title string 标题
- filters string 文件类型筛选器项目
- initDir string 初始位置 (*optional*)

Returns:

返回选择的文件路径

Usage:

```
Dialog:OpenFile('请选择wim镜像文件', 'wim 镜像文件(*.wim)|*.wim|所有文件(*.*)|*.*')
Dialog:OpenFile('请选择wim镜像文件', 'wim 镜像文件(*.wim)|*.wim|所有文件(*.*)|*.*', 'C:\\Program Files')
```

Dialog:BrowseFolder (title[, csidl])

显示 文件夹选择对话框

Parameters:

- title string 标题

csidl int 初始文件夹打开位置 CSIDL 常量特殊项 ID 列表 (*optional*)

Returns:

返回选择的文件夹路径

Usage:

```
Dialog:BrowseFolder('请选择文件夹') -- 默认打开位置为 桌面
Dialog:BrowseFolder('请选择文件夹', 0x0011) -- 初始打开位置为 此电脑

-- CSIDL (Shlobj.h) 常量特殊项 ID 列表
#define CSIDL_DRIVES                    0x0011        // My Computer
```

磁盘驱动器 库

Disk.IsLocked (drive)

判断 驱动器是否处于BitLocker加密状态

Parameters:

- drive string 驱动器盘符

Returns:

true 或者 false

Usage:

```
Disk.IsLocked("G:")
-->true
```

文件/目录 库

File.Exists (path)

判断 文件是否存在

Parameters:

- path string 路径

Folder.Exists (path)

判断 目录是否存在

Parameters:

- path string 路径

os.exists (path)

判断 文件或目录是否存在

Parameters:

- path string 路径

File.GetFullPath (path)

获取 8.3格式的路径的完整路径格式

Parameters:

- path string 路径

Usage:

```
File.GetFullPath([[C:\PROGRA~1\COMMON~1\System\wab32.dll]])
-->C:\Program Files\Common Files\System\wab32.dll
```

File.GetShortPath (path)

获取 8.3格式的短路径

Parameters:

- path string 路径

Folder.GetFullPath (path)

获取 8.3格式的路径的完整路径格式

Parameters:

- path string 路径

Usage:

```
Folder.GetFullPath([[C:\PROGRA~1\COMMON~1\System\]])
-->C:\Program Files\Common Files\System\
```

Folder.GetShortPath (path)

获取 8.3格式的短路径

Parameters:

- path string 路径

进程/窗口 库

Window.Find (title[, class])

获取 特定标题，类名的窗口对象

Parameters:

- title string 标题
- class string 类名 (*optional*)

Returns:

System.Proc 进程 对象

Window.Match (title[, class])

获取 匹配特定标题，类名规则的窗口对象

Parameters:

- title string 标题
- class string 类名 (*optional*)

Returns:

System.Proc 进程 对象

进程 对象

System.Proc
Proc 对象

Proc:Show ([nShow])
显示 窗口

Parameters:

- nShow int 显示式样 (*optional*)

Proc:Hide ()
隐藏 窗口

Proc:IsVisable ()
确认 程序窗口是否处于显示状态

Proc:Activate ()
激活 窗口

Proc:Minimize ()
最小化 窗口

Proc:Maximize ()
最大化 窗口

Proc:Close ()
关闭 窗口

Proc:Quit ()
退出 进程

Proc:Kill ()
结束 进程

Proc:SendMessage (Msg, wParam, lParam)
发送 窗口消息(Send)

Parameters:

- Msg int 指定被发送的消息
- wParam int 指定附加的消息指定信息
- lParam int 指定附加的消息指定信息

Proc:PostMessage (Msg, wParam, lParam)
发送 窗口消息(Post)

Parameters:

- Msg int 指定被发送的消息
- wParam int 指定附加的消息指定信息
- lParam int 指定附加的消息指定信息

generated by LDoc 1.4.6

```
<HorizontalLayout>
<VerticalLayout padding="20,2,0,0">
  <Label multiline="true" name="bios_text" />

  <HorizontalLayout height="20" >
    <Label width="40" name="firmwaretype_text" />
    <Label width="20" text=" - " />
    <Label multiline="true" showhtml="true" name="host_text" />
  </HorizontalLayout>

  <Label multiline="true" name="cpu_text" />
  <Label multiline="true" name="mem_text" />


</VerticalLayout>
</HorizontalLayout>
```

```
function NavTab_OnClick(self, tab, ctrl)
   App:Debug('NavTab_OnClick - ' .. tab  .. ' ' .. tostring(ctrl))

   if UI.Inited == 1 then
     if display_applybtn.visible == 1 then
       display_applybtn_click()
     end
   end

   self:DefOnClick(tab)
 end

 function NavTab_Init()
   local nav = UITab.New('$Nav')
   nav:BindLayout('TabLayoutMain')
   nav:SetList({
     'Display',
     'Background',
     'Colors',
     'FolderOptions',
     'Taskbar'
   })
   nav.OnClick = NavTab_OnClick
   return nav
 end

 --"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\tab.nav.lua"
```

```
<NavTab>
    <VerticalLayout name="$Nav" width="240" >
        <Label height="32" text="%{Settings}" visible="true" padding="20,0,0,0" />
        <HorizontalLayout style="nav_item" padding="20,0,0,0" >
            <Label text="   " font="sym16" width="32" />
            <Button text="%{Home}" align="left" name="home_button" />
        </HorizontalLayout>

        <Label height="32" showhtml="true" text="<b>%{Customization}</b>" padding="20,0,0,0" />
        <VerticalLayout height="40">
            <Label float="true" pos="20,0,32,32" mosue="false" text="   " font="sym16" width="32" />
            <Label float="true" pos="52,0,188,32" mosue="false" text="%{Display}" width="188" />
            <Option float="true" pos="0,0,240,32" group="nav_item" name="$Nav[Display]"
style="nav_item" />
        </VerticalLayout>
        <VerticalLayout height="40">
            <Label float="true" pos="20,0,32,32" mosue="false" text="   " font="sym16" width="32" />
            <Label float="true" pos="52,0,148,32" mosue="false" text="%{Background}" width="188"
enabled="false" />
            <Option float="true" pos="0,0,240,32" group="nav_item" name="$Nav[Background]"
style="nav_item" enabled="false" />
        </VerticalLayout>
        <VerticalLayout height="40">
            <Label float="true" pos="20,0,32,32" mosue="false" text="   " font="sym16" width="32" />
            <Label float="true" pos="52,0,188,32" mosue="false" text="%{Colors}" width="188" />
            <Option float="true" pos="0,0,240,32" group="nav_item" name="$Nav[Colors]"
style="nav_item" />
        </VerticalLayout>
        <VerticalLayout height="40">
            <Label float="true" pos="20,0,32,32" mosue="false" text="   " font="sym16" width="32" />
            <Label name="#res_1" float="true" pos="52,0,188,32" mosue="false" text="#
{@shell32.dll,22985}" width="188" />
            <Option float="true" pos="0,0,240,32" group="nav_item" name="$Nav[FolderOptions]"
style="nav_item" />
        </VerticalLayout>
        <VerticalLayout height="40">
            <Label float="true" pos="20,0,32,32" mosue="false" text="   " font="sym16" width="32" />
            <Label float="true" pos="52,0,188,32" mosue="false" text="%{Taskbar}" width="188" />
            <Option float="true" pos="0,0,240,32" group="nav_item" name="$Nav[Taskbar]"
style="nav_item" selected="true"/>
        </VerticalLayout>
        <Label height="32" showhtml="true" name="#res_2" text="<b>#{@shell32.dll,51378}</b>"
padding="20,0,0,0" />
        <HorizontalLayout style="nav_item" padding="20,0,0,0" >
            <Label text="   " font="sym16" width="32" />
            <Button text="#{@shell32.dll,22023}" align="left" name="compmgmt_button" />
        </HorizontalLayout>
    </VerticalLayout>
</NavTab>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="16" size="16" name="Segoe UI" default="true" />
    <Font shared="true" id="16b" size="16" name="Segoe UI" bold="true" />
    <Font shared="true" id="18" size="18" name="Segoe UI" />
    <Font shared="true" id="18b" size="18" name="Segoe UI" bold="true" />
    <Font shared="true" id="20" size="20" name="Segoe UI" />
    <Font shared="true" id="40" size="40b" name="Segoe UI" />
    <Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="Pazar" />
    <MultiLanguage id="info_w1" value="Pazartesi" />
    <MultiLanguage id="info_w2" value="SalÄ±" />
    <MultiLanguage id="info_w3" value="Ã‡arÅŸamba" />
    <MultiLanguage id="info_w4" value="PerÅŸembe" />
    <MultiLanguage id="info_w5" value="Cuma" />
    <MultiLanguage id="info_w6" value="Cumartesi" />

    <MultiLanguage id="w7" value="Paz" />
    <MultiLanguage id="w1" value="Pzt" />
    <MultiLanguage id="w2" value="Sal" />
    <MultiLanguage id="w3" value="Ã‡ar" />
    <MultiLanguage id="w4" value="Per" />
    <MultiLanguage id="w5" value="Cum" />
    <MultiLanguage id="w6" value="Cmt" />

    <MultiLanguage id="OK" value="Tamam" />
    <MultiLanguage id="CANCEL" value="Kapat" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="16" size="16" name="Segoe UI" default="true" />
    <Font shared="true" id="16b" size="16" name="Segoe UI" bold="true" />
    <Font shared="true" id="18" size="18" name="Segoe UI" />
    <Font shared="true" id="18b" size="18" name="Segoe UI" bold="true" />
    <Font shared="true" id="20" size="20" name="Segoe UI" />
    <Font shared="true" id="40" size="40b" name="Segoe UI" />
    <Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="Pazar" />
    <MultiLanguage id="info_w1" value="Pazartesi" />
    <MultiLanguage id="info_w2" value="SalÄ±" />
    <MultiLanguage id="info_w3" value="Ã‡arÅŸamba" />
    <MultiLanguage id="info_w4" value="PerÅŸembe" />
    <MultiLanguage id="info_w5" value="Cuma" />
    <MultiLanguage id="info_w6" value="Cumartesi" />

    <MultiLanguage id="w7" value="Paz" />
    <MultiLanguage id="w1" value="Pzt" />
    <MultiLanguage id="w2" value="Sal" />
    <MultiLanguage id="w3" value="Ã‡ar" />
    <MultiLanguage id="w4" value="Per" />
    <MultiLanguage id="w5" value="Cum" />
    <MultiLanguage id="w6" value="Cmt" />

    <MultiLanguage id="OK" value="Tamam" />
    <MultiLanguage id="CANCEL" value="Kapat" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="16" size="18" name="Segoe UI"  shared="true"  default="true" />
    <Font id="24" size="24" name="Segoe UI"  shared="true" />
    <Font shared="true" id="40b" size="40b" name="Segoe UI" />
    <Font shared="true" id="128" size="28" name="Segoe MDL2 Assets" />
    <Font shared="true" id="130" size="30" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="Pazar" />
    <MultiLanguage id="info_w1" value="Pazartesi" />
    <MultiLanguage id="info_w2" value="Salı" />
    <MultiLanguage id="info_w3" value="Çarşamba" />
    <MultiLanguage id="info_w4" value="Perşembe" />
    <MultiLanguage id="info_w5" value="Cuma" />
    <MultiLanguage id="info_w6" value="Cumartesi" />

    <MultiLanguage id="SignIn" value="Oturum Aç" />
    <MultiLanguage id="ForgotPassword" value="Parolanızı mı unuttunuz" />
    <MultiLanguage id="Logon" value="Giriş Yap" />

    <MultiLanguage id="will autologon with the $user account in $countdown second(s)"
value="$countdown saniye içinde $user hesabı ile oturum açılacak" />
    <MultiLanguage id="Welcome to Windows ..." value="Windows'a Hoş Geldiniz ..." />

</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
  <Font shared="true" id="14" size="14" name="Segoe UI" default="true" />
  <Font shared="true" id="24" size="24" name="Segoe UI" />
  <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
  <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />

  <MultiLanguage id="Settings" value="Ayarlar" />
  <MultiLanguage id="Home" value="Denetim Masası" />
  <MultiLanguage id="Customization" value="Özelleştirme" />

  <MultiLanguage id="Display" value="Ekran" />
  <MultiLanguage id="Resolution" value="Çözünürlük" />
  <MultiLanguage id="Rotate" value="Ekran Yönü" />
  <MultiLanguage id="Rotate_0" value="0 degree" />
  <MultiLanguage id="Rotate_90" value="90 degree" />
  <MultiLanguage id="Rotate_180" value="180 degree" />
  <MultiLanguage id="Rotate_270" value="270 degree" />

  <MultiLanguage id="Brightness" value="parlak" />

  <MultiLanguage id="Apply" value="Uygula" />
  <MultiLanguage id="Restore" value="Geri Al" />

  <MultiLanguage id="Background" value="Arka Plan" />

  <MultiLanguage id="Colors" value="Renk" />
  <MultiLanguage id="SelectAppColorTheme" value="Varsayılan uygulama modunuzu seçin" />
  <MultiLanguage id="Light" value="Açık" />
  <MultiLanguage id="Dark" value="Koyu" />

  <MultiLanguage id="Taskbar" value="Görev Çubuğu" />
  <MultiLanguage id="AutoHideTaskbar" value="Görev Çubuğunu Otomatik Gizle" />
  <MultiLanguage id="UseSmallButtons" value="Küçük görev çubuğu düğmelerini kullan" />
  <MultiLanguage id="CombineButtons" value="Görev çubuğu düğmelerini birleştir" />
  <MultiLanguage id="Always" value="Etiketleri gizle" />
  <MultiLanguage id="Auto" value="Görev çubuğu dolarsa" />
  <MultiLanguage id="Never" value="Asla" />

  <MultiLanguage id="On" value="On" />
  <MultiLanguage id="Off" value="Off" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="20" size="20" name="Segoe UI"  shared="true"  default="true" />
    <Font id="30" size="30" name="Segoe UI"  shared="true" />

<!-- The description of <Style imagerectstyle="l,t,w,h" /> in main.xml -->
    <!-- image's source, dest attributes will define with "left,top,width,height" style -->
    <!-- the origin style is "left,top,right,bottom". -->
    <!-- the xml file for "Include" tag is also use the same "imagerectstyle" -->

    <MultiLanguage id="Power" value="GÃ¼Ã§" />
    <MultiLanguage id="CloseComputer" value="PC'yi kapat" />
    <MultiLanguage id="Shutdown" value="Kapat" />
    <MultiLanguage id="Restart" value="Yeniden BaÅŸlat" />
    <MultiLanguage id="Hibernate" value="Uyku" />
    <MultiLanguage id="Standby" value="HazÄ±rda Beklet" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font id="10" size="10" name="Segoe UI" shared="true" />
    <Font id="16" size="16" name="Segoe UI" shared="true" default="true" />
    <Font id="12" size="12" name="Segoe UI" shared="true" />
    <Font id="24" size="24" name="Segoe UI" shared="true" />
    <Font id="16u" size="16" name="Segoe UI" shared="true" underline="true" />

    <MultiLanguage id="::Connected" value="Bağlandı" />
    <MultiLanguage id="::Secured" value="Güvenli" />
    <MultiLanguage id="::Open" value="Aç" />
    <MultiLanguage id="::Connect" value="Bağlan" />
    <MultiLanguage id="::Disconnect" value="Bağlantıyı Kes" />
    <MultiLanguage id="msg_hidden_ssid" value="Hidden Network" />
    <MultiLanguage id="msg_enter_ssid" value="Enter the name(SSID) for the network" />
    <MultiLanguage id="msg_enter_password" value="Ağ güvenlik anahtarını gir" />
    <MultiLanguage id="Cancel" value="Kapat" />
    <MultiLanguage id="NetworkSettings" value="Ağ & İnternet ayarları" />
    <MultiLanguage id="NetworkSettingsMsg" value="Tarifeli bir bağlantı oluşturmak gibi ayarları
değiştirin." />
    <MultiLanguage id="Firewall" value="Firewall" />
    <MultiLanguage id="Proxy" value="Proxy" />
    <MultiLanguage id="::msg_select_device" value="WLAN cihazını seçin" />
    <MultiLanguage id="::msg_service_unavailable" value="WLAN servisi kullanılmıyor" />
    <MultiLanguage id="::msg_device_unavailable" value="WLAN cihazı kullanılmıyor" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="12" size="12" name="Segoe UI" default="true" />
    <Font shared="true" id="16" size="16" name="Segoe UI" />
    <Font shared="true" id="24" size="24" name="Segoe UI" />

    <MultiLanguage id="info_w0" value="Pazar" />
    <MultiLanguage id="info_w1" value="Pazartesi" />
    <MultiLanguage id="info_w2" value="Salı" />
    <MultiLanguage id="info_w3" value="Çarşamba" />
    <MultiLanguage id="info_w4" value="Perşembe" />
    <MultiLanguage id="info_w5" value="Cuma" />
    <MultiLanguage id="info_w6" value="Cumartesi" />
</Locale>
```

```
@echo on
cd /d "%~dp0..\.."

call :GetWinXShell
call :GetFolderName "%~p0"


start "" %WINXSHELL% -ui -console -jcfg wxsUI\%p0%\main.jcfg -file
pause
taskkill /f /im winxshell.exe
start "" %WINXSHELL% -ui -console -jcfg wxsUI\%p0%\main.jcfg -file xxoo.rmvb
pause
taskkill /f /im winxshell.exe
start "" %WINXSHELL% -ui -console -jcfg wxsUI\%p0%\main.jcfg -file xxoo.vmx
pause
taskkill /f /im winxshell.exe



goto :EOF

:GetWinXShell
if exist x64\Debug\WinXShell.exe set WINXSHELL=x64\Debug\WinXShell.exe&&goto :EOF
if exist WinXShell.exe set WINXSHELL=WinXShell.exe&&goto :EOF
if exist WinXShell_%PROCESSOR_ARCHITECTURE%.exe set WINXSHELL=WinXShell_x86.exe&&goto
:EOF
set WINXSHELL=WinXShell_x64.exe
goto :EOF

:GetFolderName
if "x%~2"=="x1" set "p0=%~n1" && goto :EOF
set "p0=%~1"
set p0=%p0:~0,-1%
call :GetFolderName  "%p0%" 1
goto :EOF
```

```
cd /d "%¯dp0..\.."
call :GetWinXShell

start /WAIT %WINXSHELL% -ui -jcfg wxsUI\UI_LED\main.jcfg -scroll -wait 3 -text 热烈欢迎各位领导莅临指导。
%WINXSHELL% -ui -jcfg wxsUI\UI_LED\main.jcfg -wait 5 -text "得到服务器IP为:<u>127.0.0.1</u>"
start  %WINXSHELL% -ui -jcfg wxsUI\UI_LED\main.jcfg -top -scroll -text "设置了一个比较长的信息，窗口的宽度应该会自动调整，可以完整显示。"
%WINXSHELL% -code app:sleep(5000)
%WINXSHELL% -code QuitWindow('WinXShell-UI_LED-Wnd')

goto :EOF

:GetWinXShell
if exist x64\Debug\WinXShell.exe set WINXSHELL=x64\Debug\WinXShell.exe&&goto :EOF
if exist WinXShell.exe set WINXSHELL=WinXShell.exe&&goto :EOF
if exist WinXShell_%PROCESSOR_ARCHITECTURE%.exe set WINXSHELL=WinXShell_x86.exe&&goto :EOF
set WINXSHELL=WinXShell_x64.exe
goto :EOF
```

```
cd /d "%~dp0..\.."
call :GetWinXShell

start %WinXShell% -console -ui -jcfg wxsUI\UI_TrayPanel\main.jcfg
pause
start %WinXShell% -console -ui -jcfg wxsUI\UI_TrayPanel\main.jcfg -theme dark
pause
start %WinXShell% -console -ui -jcfg wxsUI\UI_TrayPanel\main.jcfg -theme light
goto :EOF

:GetWinXShell
if exist x64\Debug\WinXShell.exe set WINXSHELL=x64\Debug\WinXShell.exe&&goto :EOF
if exist WinXShell.exe set WINXSHELL=WinXShell.exe&&goto :EOF
if exist WinXShell_%PROCESSOR_ARCHITECTURE%.exe set WINXSHELL=WinXShell_x86.exe&&goto
:EOF
set WINXSHELL=WinXShell_x64.exe
goto :EOF
```

```
cd /d "%~dp0..\.."
x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Launcher\main.jcfg
pause
x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Launcher\full.jcfg -custom
pause
x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Launcher\startmenu.jcfg
pause
```

```
setlocal EnableDelayedExpansion

call :GetUIName "%cd%"
cd /d "%¯dp0..\.."
call :GetWinXShell

call :WAITTIPS error 错误 "系统磁盘控件不足" 0
call :WAITTIPS error 错误 "系统磁盘控件不足" 0
call :WAITTIPS error 错误 "系统磁盘控件不足" 0

for /l %%i in (1,1,8) do (
     call :TIPS info 金币 "+!random!" %%i
)

call :WAITTIPS warn 金币 "金币超最大上限 -4294967295" 0
pause

goto :EOF



:WAITTIPS
start /wait %WinXShell% -jcfg wxsUI\%UIName%\main.jcfg -i %1 -t %2 -m "%¯3"
goto :EOF

:TIPS
start %WinXShell% -jcfg wxsUI\%UIName%\main.jcfg -i %1 -t %2 -m "%¯3" -p %4
start /wait %WinXShell% -code App:Sleep(800)
goto :EOF



:GetUIName
```

```
set UIName=%~n1
goto :EOF

:GetWinXShell
if exist x64\Debug\WinXShell.exe set WINXSHELL=x64\Debug\WinXShell.exe&&goto :EOF
if exist WinXShell.exe set WINXSHELL=WinXShell.exe&&goto :EOF
if exist WinXShell_%PROCESSOR_ARCHITECTURE%.exe set WINXSHELL=WinXShell_x86.exe&&goto :EOF
set WINXSHELL=WinXShell_x64.exe
goto :EOF
```

```
@echo off
cd /d "%~dp0..\.."

call :GetWinXShell
call :GetFolderName "%~p0"

start "" %WINXSHELL% -console -jcfg wxsUI\%p0%\main.jcfg
pause
start "" %WINXSHELL% -console -jcfg wxsUI\%p0%\main.jcfg -brightness=false
pause
start "" %WINXSHELL% -console -jcfg wxsUI\%p0%\main.jcfg -theme light
pause
start "" %WINXSHELL% -console -jcfg wxsUI\%p0%\Calendar.jcfg
pause
start "" %WINXSHELL% -console -jcfg wxsUI\%p0%\Calendar.jcfg -theme dark
pause
start "" %WINXSHELL% -console -jcfg wxsUI\%p0%\Calendar.jcfg -theme light
goto :EOF

:GetWinXShell
if exist x64\Debug\WinXShell.exe set WINXSHELL=x64\Debug\WinXShell.exe&&goto :EOF
if exist WinXShell.exe set WINXSHELL=WinXShell.exe&&goto :EOF
if exist WinXShell_%PROCESSOR_ARCHITECTURE%.exe set WINXSHELL=WinXShell_x86.exe&&goto
:EOF
set WINXSHELL=WinXShell_x64.exe
goto :EOF

:GetFolderName
if "x%~2"=="x1" set "p0=%~n1" && goto :EOF
set "p0=%~1"
set p0=%p0:~0,-1%
call :GetFolderName  "%p0%" 1
goto :EOF
```

```
cd /d "%~dp0..\.."
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Resolution\main.jcfg
```

```
cd /d "%~dp0..\.."

start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_WIFI\main.jcfg -theme light
pause
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_WIFI\main.jcfg -hidewindow
pause
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_WIFI\main.jcfg -theme dark
pause
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_WIFI\main.jcfg -theme light
```

```
call :GetUIName "%cd%"
cd /d "%~dp0..\.."
call :GetWinXShell

start %WinXShell% -console -ui -jcfg wxsUI\%UIName%\main.jcfg
pause
start %WinXShell% -console -ui -jcfg wxsUI\%UIName%\main.jcfg -theme dark
pause
rem start %WinXShell% -console -ui -jcfg wxsUI\%UIName%\KB_Layout.jcfg
goto :EOF

:GetUIName
set UIName=%~n1
goto :EOF

:GetWinXShell
if exist x64\Debug\WinXShell.exe set WINXSHELL=x64\Debug\WinXShell.exe&&goto :EOF
if exist WinXShell.exe set WINXSHELL=WinXShell.exe&&goto :EOF
if exist WinXShell_%PROCESSOR_ARCHITECTURE%.exe set WINXSHELL=WinXShell_x86.exe&&goto :EOF
set WINXSHELL=WinXShell_x64.exe
goto :EOF
```

```
cd /d "%~dp0..\.."
rem set logon_script=D:\Dev\Shell\ReleaseWinXShell\WinXShell_x64\pecmd.lua
x64\Debug\WinXShell.exe -console -ui  -user Administrator -jcfg wxsUI\UI_Logon\UI_LogonPE.jcfg
pause
x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Logon\UI_LogonPE.jcfg -user SYSTEM
pause
x64\Debug\WinXShell.exe  -console -ui -jcfg wxsUI\UI_Logon\main.jcfg -bk bk.jpg
pause
x64\Debug\WinXShell.exe -ui -jcfg wxsUI\UI_Logon\main.jcfg
pause
x64\Debug\WinXShell.exe -ui -jcfg wxsUI\UI_Logon\main.jcfg -blur 8 -bk bk2.jpg
goto :EOF

x64\Debug\WinXShell.exe -ui -jcfg wxsUI\UI_Logon\main.jcfg -blur 12 -bk bk3.jpg
```

```
cd /d "%~dp0..\.."
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Volume\main.jcfg
pause
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Volume\main.jcfg -theme dark
pause
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Volume\main.jcfg -theme light
```

```
cd /d "%~dp0..\.."
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Shutdown\main.jcfg
pause
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Shutdown\full.jcfg -blur 5
pause
start x64\Debug\WinXShell.exe -console -ui -jcfg wxsUI\UI_Shutdown\full.jcfg -blur 5 -bk bk.jpg
```

```
cd /d "%~dp0..\.."
call :GetWinXShell

start /WAIT %WINXSHELL% -console -ui -jcfg wxsUI\UI_LED\main.jcfg -scroll -wait '3' -text " Show a scrolling message."
%WINXSHELL% -ui -jcfg wxsUI\UI_LED\main.jcfg -wait 5 -text " Get the IP:<u>127.0.0.1</u>"
start  %WINXSHELL% -ui -jcfg wxsUI\UI_LED\main.jcfg -title UI_LED-looong_message -top -scroll -text " This is a looong message, the window will auto increase the width to shis message."
%WINXSHELL% -code App:Sleep(5000)
%WINXSHELL% -code Window.Find('UI_LED-looong_message'):Quit()

goto :EOF

:GetWinXShell
if exist x64\Debug\WinXShell.exe set WINXSHELL=x64\Debug\WinXShell.exe&&goto :EOF
if exist WinXShell.exe set WINXSHELL=WinXShell.exe&&goto :EOF
if exist WinXShell_%PROCESSOR_ARCHITECTURE%.exe set WINXSHELL=WinXShell_x86.exe&&goto :EOF
set WINXSHELL=WinXShell_x64.exe
goto :EOF
```

```
{
 "name":"UI_Logon",
 "title":"wxsLogon",
 "nobaricon":true,
 "customstyle":true,
 "style":2415919104,
 "startup_win":"max",
 "OnEscKey":"none",
 "==========":"==========",
 "shadow":"Administrator:admin;\\nSYSTEM:;",
 "logon_user":"Administrator",
 "auto_logon_second":8
}
{
 "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Logon\UI_LogonPE.jcfg",
}
```

```
@echo off
cd /d "%~dp0"

set x8664=x64
if not "x%PROCESSOR_ARCHITECTURE%"=="xAMD64" set x8664=x86
set WINXSHELL=dist\X_PF\WinXShell\WinXShell_%x8664%.exe
if not exist %WINXSHELL% set WINXSHELL=WinXShell.exe
if not exist %WINXSHELL% set WINXSHELL=WinXShell_%x8664%.exe

echo Hints:
echo -----------------------------------------
echo For UI_Logon
echo Username:Administrator
echo Password:admin
echo.
echo For UI_Shutdown
echo Press 'Esc' to cancel(close)
echo -----------------------------------------
echo.
echo.
echo Press any key to test ...
pause >nul

start %WINXSHELL% -ui -jcfg wxsUI\UI_Logon.zip
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Logon.zip -bk bk2.jpg
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Settings.zip
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Shutdown.zip\full.jcfg -blur 5.0
pause
rem start %WINXSHELL% -ui -jcfg wxsUI\UI_Resolution.zip
rem pause
rem start %WINXSHELL% -ui -jcfg wxsUI\UI_Shutdown.zip
rem pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_WIFI.zip\main.jcfg -theme light
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Volume.zip\main.jcfg -theme dark
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Resolution.zip\wallpaper.jcfg
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Calendar.zip -theme dark
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Calendar.zip\calendar.jcfg -theme light
rem start %WINXSHELL% -ui -jcfg wxsUI\UI_Calendar.zip\calendar.jcfg
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_SystemInfo.zip
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Launcher.zip -theme dark
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Launcher.zip\full.jcfg -custom
echo exit...
pause
exit 0

rem run with parameter
rem change resolution directly
```

```
start %WINXSHELL% -ui -jcfg wxsUI\UI_Resolution.zip -direct
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Resolution.zip -lua direct.lua
pause
rem no sound playing when volume changed
start %WINXSHELL% -ui -jcfg wxsUI\UI_Volume.zip -nobeep
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_WIFI.zip -hidewindow
pause
start %WINXSHELL% -ui -jcfg wxsUI\UI_Launcher.zip -theme dark
pause
```

Module `UI-Engine`

UI Engine

generated by LDoc 1.4.6

```
---
function string.splitlines(str)
  local arr = {}
 for line in string.gmatch(str, "([^\n]+)") do
    table.insert(arr, line)
  end
  return arr
end

---
win = {}

function win.popen(cmd)
  local exitcode, stdout = winapi.execute(cmd)
  stdout = stdout:gsub("\r\n", "\n")
  return stdout
end

---
dir = {}
function dir.folders(path)
  local folders = win.popen('dir /b /ad \"' .. path .. '\"')
  return string.splitlines(folders)
end

---



function msg(msg)
  winapi.show_message('', msg)
end



---
function suilib.genItemXML(template, items, item_regex)
  local xml = {}
  local tmpl = ''
  for i, item in ipairs(items) do
    if item_regex then
      item = item:match(item_regex)
    end
    tmpl = template:gsub('{$I}', i)
    tmpl = tmpl:gsub('{$ITEM}', item)
    table.insert(xml, tmpl)
  end
  return table.concat(xml, "\r\n")
end

function suilib.insertItem(sui_obj, xml, parent_head, parent_foot)
  local str = {}

  if parent_head == nil then parent_head = '<VerticalLayout>' end
  if parent_foot == nil then parent_foot = '</VerticalLayout>' end

  table.insert(str, '<elem>')
  table.insert(str, parent_head)
  table.insert(str, xml)
  table.insert(str, parent_foot)
  table.insert(str, '</elem>')
  sui_obj:add(table.concat(str, "\r\n"))
end
--"ÔÂ·¾¶":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\utils.lua"
```

```
fixed_out = {}
table.insert(
                fixed_out,
                string.format("<i ..\\UI_DL\\app_ico\\vmware.png><a vm16><c #000000>VMWARE16虚拟机</c></a></b> \n史上最伟大网管精简的VMWARE虚拟机\n")
            )--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\vm.lua"
```

```
{
  "description" : {
    "name":"string. Specify the UI name.",
    "entry":"string. Specify the Entry file. optional(main.xml).",
    "singleton":"boolean. wether the window is only one instance or not. optional(false).",
    "noshadow":"boolean. wether use Shadow.png for window shadow effort or not. optional(true).",
    "position":"string(center|rightbottom ). Specify the startup position. optional(center).",
    "nobaricon":"boolean. wether show icon on the taskbar or not. optional(false).",
    "baricon":"string. Specify the taskbar icon. optional(main.ico).",
    "customstyle":"boolean. wether use custom the window style or not. optional(false).",
    "style":"long. Specify the window style. Please refer to Microsoft documentation.
optional(WS_OVERLAPPEDWINDOW).",
    "exstyle":"long. Specify the window extend style. Please refer to Microsoft documentation.
optional(WS_EX_WINDOWEDGE).",
    "theme":"string. Specify the theme(themes\*.xml). optional(default).",
    "locale":"string. Define the locale resources for the UI text display, ie:zh-CN, en-US. optional(SYSTEM
default locale)."
  },
  "name":"UI_Resolution",
  "title":"%{Settings}",
  "entry":"wallpaper.xml",
  "lua":"wallpaper.lua",
  "baricon":"display.ico"
}
{
  "Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Resolution\wallpaper.jcfg",
}
```

```
function onload()
   mypath = sui:jcfg('name')
end

function onclick(ctrl)
   if ctrl == "wp_broswerbtn" then
      App:Run('pecmd.exe', '.\\' .. mypath .. '\\wp_script.ini select')
   elseif ctrl == "wp_searchbtn" then
      App:Run('pecmd.exe', '.\\' .. mypath .. '\\wp_script.ini random')
   elseif ctrl == "wp_defaultbtn" then
      App:Run('pecmd.exe', 'WALL X:\\system32\\winpe.jpg')
   elseif ctrl == "cancelbtn" then
      sui:close()
   end
end
--"Original path":"X:\AI_Work\WinXShell\wxsUI\UI_Resolution\wallpaper.lua"
```

```
<?xml version="1.0" encoding="utf-8"?>
<Window mininfo="200,360" size="450,450" caption="0,0,0,32" sizebox="4,4,4,4"
defaultfontcolor="#FFE8EAF6">
  <Style imagerectstyle="l,t,w,h" />
  <Include source="scrollbar_ltwh.xml" />

   <Style name="inBtn" value="width=&quot;120&quot; height=&quot;32&quot;
textcolor=&quot;#FF000000&quot; hottextcolor=&quot;#FFFFFF&quot;
hotimage=&quot;color='#FF050081'&quot;" />

  <VerticalLayout style="ct-bkcontent">
    <!-- <Control name="xml_entry" text="wallpaper" visible="false" /> -->
    <HorizontalLayout height="32" style="ct-bktitle">
      <Label text="Wallpaper" textpadding="5,0,0,0" />
      <Control />
      <Button name="::closebtn" bkimage="file='close.png'" width="16" height="16" padding="0,8,5,0" />
    </HorizontalLayout>
    <VerticalLayout height="64" style="ct-bksubtitle">
      <Label text="Wallpaper Settings" font="24" textpadding="10,16,0,0" />
    </VerticalLayout>
    <HorizontalLayout height="188" style="ct-bkontent-darker1">
        <Control />
        <Button normalimage="Display.png" height="124" width="218" enabled="false" padding="0,32,0,0" />
        <Control />
    </HorizontalLayout>
    <Label text="<b>Change Wallpaper</b>" showhtml="true" textcolor="#FF000000" textpadding="10,8,0,0" />
    <HorizontalLayout height="50" >
      <Button name="wp_broswerbtn" text="Select" style="inBtn" />
      <Button name="wp_searchbtn" text="Random" style="inBtn" width="150" />
      <Button name="wp_defaultbtn" text="Default" style="inBtn" />
    </HorizontalLayout>
    <HorizontalLayout height="40" padding="0,30,0,0">
      <Control />
      <Button name="applybtn" text="%{Apply}" shortcut="A" showhtml="true" style="attentionBtn" />
      <Button name="cancelbtn" text="%{Cancel}" shortcut="C" showhtml="true" style="attentionBtn"
padding="0,0,10,0" />
    </HorizontalLayout>
  </VerticalLayout>
</Window>
```

```
@echo off
:settings
cls
echo 本批处理将建立WIFI热点.
echo.
set /p ssid=输入 SSID:
cls
echo 本批处理将建立WIFI热点.
echo.
set /p pass=输入 密码:
cls
:confirm
echo 无线名称为：%ssid%
echo 密码为     ：%pass%
echo.
set /P AREYOUSURE=确定创建？ [Y/N]
if /I "%AREYOUSURE%" NEQ "Y" (
    goto settings
) ELSE (
    goto hotspot
)
:hotspot
cls
netsh wlan set hostednetwork mode=allow ssid=%ssid% key=%pass% keyUsage=persistent
PING localhost -n 3 >NUL
netsh wlan start hostednetwork
if errorlevel 1 goto error
:pause
echo 已创建热点，按任意键关闭.
pause
set /P CLOSE=立即关闭热点么？[Y/N]
```

```
if /I "%CLOSE%" NEQ "Y" (
    goto pause
) ELSE (
    goto close
)
:close
cls
netsh wlan stop hostednetwork
echo 热点已关闭
goto end
:error
echo 未知错误
:end
pause
```

WinXShell

- Project
- Documentation

Contents

- 详细
- WinXShell.jcfg
- WinXShell.lua
- 事件函数(外壳)

Libraries

- Lua-Engine
- UI-Engine
- **Windows-Shell**

Program

- WinXShell

Classes

- App
- Shell
- System

Manual

- CSIDL
- 更新历史记录

Module `Windows-Shell`

Windows 外壳程序

在 Windows Desktop, Windows Server, Windows Server Core 以及 Windows PE环境下

运行的轻量 Windows 外壳程序(基本桌面，开始菜单，任务栏，托盘区域，时钟区域)。

详细

WinXShell.jcfg	配置文件
WinXShell.lua	扩展脚本

WinXShell.jcfg

JS_DESKTOP	桌面 设置
JS_THEMES	主题 设置
JS_TASKBAR	任务栏 设置
JS_STARTMENU	开始菜单 设置
JS_QUICKLAUNCH	快速启动栏 设置
JS_NOTIFYAREA	托盘区域 设置
JS_NOTIFYCLOCK	时钟区域 设置
JS_DAEMON	Daemon模式 设置
JS_FILEEXPLORER	文件资源管理器 设置

WinXShell.lua

ShellEvents	事件函数(外壳)
CustomActions	自定义时钟信息显示格式

事件函数(外壳)

App:PreShell ()	外壳启动前序事件函数
App:OnShell ()	外壳启动事件函数
App:OnFirstShellRun ()	系统启动后，第一次外壳启动事件函数

详细

WinXShell.jcfg
配置文件

WinXShell.lua
扩展脚本

WinXShell.jcfg

json 格式的配置文件，可配置外壳程序的各部分功能的显示效果。

JS_DESKTOP
桌面 设置

Fields:

- 3rd_open_arguments string 双击桌面图标时，调用 第三方文件资源管理器 的参数格式
- cascademenu{'WinXNew'} string 桌面新建二级菜单
- iconsize int 桌面图标大小
- bkcolor color 桌面的背景色
- wallpaper_mode int 壁纸显示模式
  - 0 - strech 拉伸 (默认)
  - 1 - tile 平铺
  - 2 - center 居中
- wallpaper string 壁纸图片路径，不指定时使用系统壁纸

Usage:

```
"JS_DESKTOP": {
  "3rd_open_arguments": "\"%s\"",
  "cascademenu": {
    "WinXNew": "Directory\\Background\\shell\\WinXNew"
  },
  "#iconsize": 32,
  "bkcolor comment": "green for eyes",
  "bkcolor": [199, 237, 204],
  "wallpaperstyle comment": "value:0-strech(default);1-tile;2-center",
  "::WP_MODE": 0,
  "wallpaper comment": "if it is not set or be empty, will use system's wallpaper image file",
  "::WP": ""
}
```

JS_THEMES

主题 设置

Fields:

- taskbar{'bkcolor'} color 任务栏背景色
- taskbar{'task_line_color'} color 任务图标海缆线颜色
- taskbar{'textcolor'} color 任务标题颜色

Usage:

```
JS_THEMES = {
  "blue": {
    "taskbar": {
      "bkcolor": [0, 120, 215],
      "task_line_color": [238, 238, 238],
      "textcolor": "0xffffff"
    }
  },
  "dark": {
    "taskbar": {
      "bkcolor": [38, 38, 38],
      "task_line_color": [238, 238, 238],
      "textcolor": "0xffffff"
    }
  },
  "light": {
    "taskbar": {
      "style": "light",
      "bkcolor": [238, 238, 238],
      "task_line_color": [0, 120, 215],
      "textcolor": "0x000000"
    }
  }
}
```

JS_TASKBAR

任务栏 设置

Fields:

- visible bool 是否显示任务栏
- smallicon bool 是否使用小图标显示
- thumbnail bool 是否显示任务缩略图
- task_close_button bool 是否显示任务快捷关闭按钮
- no_task_title bool 是否隐藏任务标题
- userebar bool 是否使用rebar(快速启动栏)
- theme string 主题名
- height int 任务栏高度

Usage:

```
"JS_TASKBAR": {
    "visible":  true,
    "smallicon": false,
    "thumbnail": true,
    "task_close_button": true,
    "no_task_title": false,
    "userebar": false,
    "theme": "dark",
    "height": 40
  }
```

JS_STARTMENU

开始菜单 设置

Fields:

- enabled bool 是否响应开始菜单
- start_pushed_bkcolor color 开始菜单按下状态的背景颜色
- start_icon string 开始菜单图标
  - `theme` - 主题资源图标(exe文件的 ICO 资源)
  - `empty` - 空白图标
  - `custom` - 自定义图标(exe文件的 ICO 资源)
- notopitems bool 隐藏开始菜单顶部快捷方式图标
- noprograms bool 隐藏 程序 组
- nosettings bool 隐藏 设定
- nobrowse bool 隐藏 浏览
- noconnections bool 隐藏 网络连接
- nofind bool 隐藏 查找
- norun bool 隐藏 运行
- nologoff bool 隐藏 注销
- norestart bool 隐藏 重启
- noshutdown bool 隐藏 关机
- noterm bool 隐藏 退出
- commands{'reboot'} json 自定义重启命令
- commands{'shutdown'} json 自定义关机命令

Usage:

```
  "JS_STARTMENU": {
    "enabled":  true,
    "start_pushed_bkcolor": [0, 100, 180],
    "start_icon_comment": "theme;empty;custom",
    "start_icon": "theme",
    "notopitems": false,
    "noprograms": false,
    "nosettings": true,
    "nobrowse": false,
    "noconnections": false,
    "nofind": true,
    "norun": false,
    "nologoff": true,
    "norestart": false,
    "noshutdown": false,
    "noterm": true,
    "commands": {
      "reboot": {
        "command": "Wpeutil.exe",
        "parameters": "Reboot"
      },
      "shutdown": {
        "command": "Wpeutil.exe",
        "parameters": "Shutdown"
      }
    }
  }
```

JS_QUICKLAUNCH

快速启动栏 设置

Fields:

- maxiconsinrow bool 快速启动栏最大单行图标个数
- 3rd_startup_arguments bool 第三方文件资源管理器打开默认参数
- hide_showdesktop bool 是否显示显示桌面图标
- hide_fileexplorer bool 是否显示文件资源管理器图标
- hide_fixedsep bool 是否隐藏系统与用户图标分割线
- hide_usericons bool 是否隐藏用户图标
- folder string 用户图标所在路径

Usage:

```
"JS_QUICKLAUNCH": {
   "maxiconsinrow": 20,
   "3rd_startup_arguments": "",
   "hide_showdesktop": false,
   "hide_fileexplorer": false,
   "hide_fixedsep": false,
   "hide_usericons": false,
   "folder": "Microsoft\\Internet Explorer\\Quick Launch\\User Pinned\\TaskBar"
}
```

JS_NOTIFYAREA

托盘区域 设置

Fields:

- hide_toggle_button bool 是否隐藏托盘图标折叠按钮
- hide_showdesktop_button bool 是否隐藏显示桌面按钮
- handle_system_network bool 是否接管系统原生网络图标点击事件
- handle_system_volume bool 是否接管系统原生网络图标点击事件

Usage:

```
"JS_NOTIFYAREA": {
   "hide_toggle_button": false,
   "hide_showdesktop_button": false,
   "handle_system_network": true,
   "handle_system_volume": true
}
```

JS_NOTIFYCLOCK

时钟区域 设置

Fields:

- visible bool 是否显示时钟区域

Usage:

```
JS_NOTIFYCLOCK = {
   "visible": true
}
```

JS_DAEMON

Daemon模式 设置

Fields:

- clockarea_click json 定义时钟区域单击时调用命令
- clockarea_dbclick json 定义时钟区域双击时调用命令
- screen_brightness int 亮度

Usage:

```
"JS_DAEMON": {
  "clockarea_click": {
    "command": "##{JVAR_MODULEPATH}\\#{JVAR_MODULENAME}",
    "parameters": "-code wxsUI('UI_Calendar','main.jcfg')"
  },
  "clockarea_dbclick": {
    "command": "timedate.cpl"
  },
  "screen_brightness": 100
 }
```

JS_FILEEXPLORER

文件资源管理器 设置

Fields:

- 3rd_filename string 第三方文件资源管理器程序路径
- padding-bottom int 内置文件资源管理器(文件打开对话框)的底部高度

Usage:

```
JS_FILEEXPLORER = {
  "3rd_filename": "##{JVAR_MODULEPATH}\\explorer++.exe",
  "padding-bottom": 20
}
```

WinXShell.lua

ShellEvents

事件函数(外壳)

CustomActions

自定义时钟信息显示格式

事件函数(外壳)

App:PreShell ()

外壳启动前序事件函数

App:OnShell ()

外壳启动事件函数

App:OnFirstShellRun ()

系统启动后，第一次外壳启动事件函数

generated by LDoc 1.4.6

WinXShell

- Project
- Documentation

Contents

- options [选项]
- commands [命令]
- ms-settings
- wxs-open

Program

- **WinXShell**

Libraries

- Lua-Engine
- UI-Engine
- Windows-Shell

Classes

- App
- Shell
- System

Manual

- CSIDL
- 更新历史记录

Script `WinXShell`

WinXShell 实用程序。

- Windows-Shell --- 一个轻量的 Windows 外壳程序(Windows Shell)
- Lua-Engine --- 一个 Lua 解释程序(Lua Engine)
- UI-Engine --- 一个现代 UI 组件引擎(UI Engine)

当前版本: 5.0.0 (2022-08-08)

更新记录请见 history.md

主题(无忧论坛)

- 作者 slore
- **copyright** 2015-2024

Usage:

```
WinXShell.exe [options] ...
WinXShell.exe [commands]
WinXShell.exe -code <lua code>
WinXShell.exe -script <path/to/script.lua> [paramterlist]
WinXShellC.exe -code <lua code>
WinXShellC.exe -script <path/to/script.lua> [paramterlist]
```

options [选项]

-regist	注册WinXShell.exe程序，之后可省略路径进行调用。
-regist_only	注册WinXShell.exe程序，之后可省略路径进行调用。
-console	显示调试窗口。可与其他选项命令一起使用。
-log	生成日志文件，记录输出信息。
-noaction	不执行任何命令退出程序。
-shell	启动外壳程序。如果存在外壳程序，则先关闭外壳程序后启动。
-daemon	以常驻进程模式启动。负责接管系统接口，亮度调节控制等。
-jcfg	指定UI组件的启动配置。 默认加载 `main.jcfg`。
-code	执行lua代码。
-script	执行lua脚本。

commands [命令]

ms-settings
wxs-open

ms-settings

ms-settings:system	打开 我的电脑属性
ms-settings:about	打开 关于 界面
ms-settings:display	打开 显示设置 界面
ms-settings:personalization	打开 个性化 界面
ms-settings:taskbar	打开 任务栏设置 界面
ms-settings:dateandtime	打开 时间和日期设置 界面
ms-settings:network	打开 网络连接设置 界面(高版本Windows有效)

wxs-open

wxs-open:system	打开 我的电脑属性
wxs-open:printers	打开 打印机设置 窗口
wxs-open:networkconnections	打开 网络连接设置 窗口
wxs-open:devicesandprinters	打开设备和打印机设置 窗口
wxs-open:wifi	打开 无线网络连接 界面
wxs-open:volume	打开 音量控制 界面

options [选项]

-regist

注册WinXShell.exe程序，之后可省略路径进行调用。 (写入注册表`App Paths`项目)
一般用于初次启动命令， 常与 -noaction 或 -daemon 选项一起使用。

Usage:

```
WinXShell.exe -regist -noaction
```

-regist_only

注册WinXShell.exe程序，之后可省略路径进行调用。 (写入注册表`App Paths`项目)
一般用于初次启动命令(与`-regist -noaction`效果相同)。

Usage:

```
WinXShell.exe -regist_only
```

-console

显示调试窗口。可与其他选项命令一起使用。

Usage:

```
WinXShell.exe -console -code "winapi.show_message('title', 'message')"
```

-log

生成日志文件，记录输出信息。 在 %temp% 目录下生成， WinXShell.<PID>.log 日志文件。

有 -console 参数时，也会生成日志文件，不需要添加此选项。

Usage:

```
WinXShell.exe -log -code "winapi.show_message('title', 'message')"
```

-noaction

不执行任何命令退出程序。 一般用于初次启动命令， 一般在初次启动时，和 -regist 选项一起使用只做程序注册。

Usage:

```
WinXShell.exe -regist -noaction
```

-shell

启动外壳程序。如果存在外壳程序，则先关闭外壳程序后启动。

Usage:

```
WinXShell.exe -shell
```

-daemon

以常驻进程模式启动。负责接管系统接口，亮度调节控制等。

Usage:

```
WinXShell.exe -daemon
```

-jcfg

指定UI组件的启动配置。 默认加载 `main.jcfg`。

Usage:

```
WinXShell.exe -jcfg wxsUI\UI_Settings.zip\keyboard.jcfg
```

-code

执行lua代码。

See also:

Lua-Engine

Usage:

```
WinXShell.exe -code lua代码

-- 示例命令
WinXShell.exe -code Desktop:SizeIconSize('M')
WinXShell.exe -code Taskbar:Pin('regedit.exe')
WinXShell.exe -code winapi.show_message('title','message')
-- 代码中包含空格字符，使用双引号将代码引用
WinXShell.exe -code "winapi.show_message('title', 'message')"
WinXShell.exe -code "winapi.show_message('title', 'message-' .. '1')"
WinXShell.exe -code "exec('cmd.exe /k echo hello')"
WinXShell.exe -code "exec('/hide', 'PETools.cmd PETools.7z')"
WinXShell.exe -code "Taskbar:Pin([[%ProgramFile%\Everything\Everything.exe]])"
-- 使用 WinXShellC.exe 控制台程序，可通过 Cmd:Echo() 输出内容与命令提示符进行交互
WinXShellC.exe -code "Cmd:Echo(App.Version)"
WinXShellC.exe -code "Cmd:Echo(os.date())"

-- test.bat
for /f %%i in ('WinXShellC.exe -code Cmd:Echo^(App.Version^)') do set Ver=%%i
echo %Ver%
for /f %%i in ('WinXShellC.exe -code Cmd:Echo^(Screen:Get^('X'^)^)') do set ScreenX=%%i
echo %ScreenX%
```

-script

执行lua脚本。

Usage:

```
WinXShell.exe -script path\to\script.lua [参数]
WinXShellC.exe -script path\to\script.lua [参数]
-- 使用 WinXShellC.exe 控制台程序，可通过 Cmd:Echo() 输出内容与命令提示符进行交互。
```

commands [命令]

ms-settings
wxs-open

ms-settings

ms-settings:system

打开 我的电脑属性

ms-settings:about

打开 关于 界面

ms-settings:display

打开 显示设置 界面

ms-settings:personalization

打开 个性化 界面

ms-settings:taskbar

打开 任务栏设置 界面

ms-settings:dateandtime

打开 时间和日期设置 界面

ms-settings:network

打开 网络连接设置 界面(高版本Windows有效)

wxs-open

wxs-open:system
打开 我的电脑属性

wxs-open:printers
打开 打印机设置 窗口

wxs-open:networkconnections
打开 网络连接设置 窗口

wxs-open:devicesandprinters
打开设备和打印机设置 窗口

wxs-open:wifi
打开 无线网络连接 界面

wxs-open:volume
打开 音量控制 界面

generated by LDoc 1.4.6

```
{
  "JS_README": {
    "can_be_omitted_section": true,
    "description": [
      "this is WinXShell's configure file.",
      "you can specify some settings for desktop,taskbar,etc."
    ]
  },
  "JS_JMACRO": {
    "can_be_omitted_section": true,
    "description": [
      "specify alias name for built-in paramater names which using in jcfg file.",
      "MACRO format:<built-in paramater name>,<alias name>,<description>",
      "<built-in paramater name>:cann't be modified.",
      "<alias name>: specify alias name which start with [::] for built-in paramater name",
      "<description>: the description of built-in paramater name, cann't be modified."
    ],
    "JKEYNAME": [
      ["JS_JMACRO", "",           "section name. define the alias name for *i18n*"],
      ["JS_STARTMENU", "::STARTMENU",         "section name. specify some settings for startmenu."],
      ["JS_QUICKLAUNCH", "::QL",      "section name. specify some settings for quicklanchbar."],
      ["wallpaperstyle", "::WP_MODE",  "property name. wallpaper's style value:0-strech(default);1-tile;2-center"],
      ["wallpaper", "::WP",           "property name."]
    ]
  },
  "JS_DAEMON": {
    "clockarea_click": {
      "command": "##{JVAR_MODULEPATH}\\#{JVAR_MODULENAME}",
      "parameters": "-code wxsUI('UI_Calendar','main.jcfg')"
    },
    "clockarea_dbclick": {
      "command": "timedate.cpl"
    },
    "handle_CAPS_double": false,
    "disable_showdesktop": false,
    "screen_brightness": 100
  },
  "JS_FILEEXPLORER": {
    "#3rd_filename": "##{JVAR_MODULEPATH}\\explorer++.exe",
    "WIN+E": {
      "#3rd_filename": ""
    },
    "#padding-bottom": 20
  },
  "JS_DESKTOP": {
    "3rd_open_arguments": "\"%s\"",
    "cascademenu": {
      "#WinXNew": "Directory\\Background\\shell\\WinXNew"
    },
    "#iconsize": 32,
    "bkcolor comment": "green for eyes",
    "bkcolor": [199, 237, 204],
    "wallpaperstyle comment": "value:0-strech(default);1-tile;2-center",
    "::WP_MODE": 0,
    "wallpaper comment": "if it is not set or be empty, will use system's wallpaper image file",
    "::WP": ""
  },
  "JS_THEMES": {
    "blue": {
      "taskbar": {
        "bkcolor": [0, 120, 215],
        "#bkmode_values": "opaque; transparent; acrylic",
        "bkmode": "opaque",
        "transparency": 100,
        "task_line_color": [238, 238, 238],
        "textcolor": "0xffffff"
      }
    },
    "dark": {
      "taskbar": {
        "bkcolor": [38, 38, 38],
        "#bkmode_values": "opaque; transparent; acrylic",
        "bkmode": "transparent",
        "transparency": 64,
        "task_line_color": [238, 238, 238],
        "textcolor": "0xffffff"
      }
    },
    "light": {
      "taskbar": {
        "style": "light",
        "bkcolor": [238, 238, 238],
        "#bkmode_values": "opaque; transparent; acrylic",
        "bkmode": "acrylic",
        "transparency": 100,
        "task_line_color": [0, 120, 215],
        "textcolor": "0x000000"
      }
    }
  },
  "JS_TASKBAR": {
    "visible": true,
    "smallicon": false,
    "thumbnail": true,
    "task_close_button": true,
    "no_task_title": false,
    "userebar": false,
    "theme": "dark",
    "height": 40
  },
  "::STARTMENU": {
    "start_pushed_bkcolor": [0, 100, 180],
    "start_icon_comment": "theme;empty;custom;files",
    "start_icon": "theme",
    "start_command": "",
    "notopitems": false,
    "noprograms": false,
    "nosettings": true,
    "nobrowse": false,
    "noconnections": false,
    "nofind": true,
    "norun": false,
    "nologoff": true,
    "norestart": false,
    "noshutdown": false,
    "noterm": true,
    "commands": {
      "reboot": {
        "command": "Wpeutil.exe",
        "parameters": "Reboot"
      },
      "shutdown": {
        "command": "Wpeutil.exe",
        "parameters": "Shutdown"
      }
    }
  },
  "::QL": {
    "maxiconsinrow": 20,
    "#3rd_filename": "",
    "3rd_startup_arguments": "",
    "hide_showdesktop": false,
    "hide_fileexplorer": false,
    "hide_fixedsep": false,
    "hide_usericons": false,
    "folder": "Microsoft\\Internet Explorer\\Quick Launch\\User Pinned\\TaskBar"
  },
  "JS_NOTIFYAREA": {
    "hide_toggle_button": false,
    "hide_showdesktop_button": false,
    "#handle_system_network": true,
    "#handle_system_volume": true
  },
  "JS_NOTIFYCLOCK": {
    "visible": true
  }
}{
}
{
  "Original path":"X:\AI_Work\WinXShell\WinXShell.jcfg",
}
{
  "Original path":"X:\AI_Work\WinXShell\WinXShell.jcfg",
}
```

```
App.ScriptEncoding = 'UTF-8'

is_pe = (os.info('isWinPE') == 1)  -- Windows Preinstallation Environment
is_wes = App:HasOption('-wes')          -- Windows Embedded Standard
is_win = App:HasOption('-windows')  -- Normal Windows

function App:Customization()
  -- 中文(Chinese): 设置 [我的电脑] - [属性] 菜单的处理方式
  -- English: Configure the action for [This PC] - [Property] menu

  -- 'auto', 'ui_systemInfo', 'system', '' or nil
  WxsHandler.SystemProperty = 'auto'

  -- 中文(Chinese): 设置 快捷方式 的 [打开文件所在位置] 菜单的动作函数
  -- English: Bind the function for shortcut's [OpenContainingFolder] menu

  -- nil or a handler function [   func(lnkfile, realfile)  ]
  -- WxsHandler.OpenContainingFolder = MyOpenContainingFolderHandler

  -- 中文(Chinese): 指定 当屏幕相关设定改变时触发的事件函数
  -- English: Specify the event function to be triggered when the screen related settings are changed
  -- WxsHandler.DisplayChangedHandler = MyDisplayChangedHandler

  -- 中文(Chinese): 设置 任务栏 时钟显示信息
  -- English: Set the display text of the taskbar clock
  -- WxsHandler.TrayClockTextFormatter = nil

  if os.info('locale') == 'zh-CN' then
    -- WxsHandler.TrayClockTextFormatter = TrayClockTextFormatter_zhCN
  elseif os.info('locale') == 'en-US' then
    -- WxsHandler.TrayClockTextFormatter = TrayClockTextFormatter_enUS
  end
end

function App:onLoad()
  -- App:Run('notepad.exe')
  print('WinXShell.exe loading...')
  print('CommandLine:' .. App.CmdLine)
  print('WINPE:'.. tostring(is_pe), 123, 'test', App)

  App:Customization()
end

function App:onDaemon()
end

function App:PreShell()
end

function App:onFirstShellRun()
end

function App:onShell()
  -- wxsUI('UI_WIFI', 'main.jcfg', '-notrayicon -hidewindow')
  -- wxsUI('UI_Volume', 'main.jcfg', '-notrayicon -hidewindow')
end

function App:onTimer(tid)
end
```

```
-- need to update "JS_STARTMENU" : {"start_command": "StartButton:onClick"} in WinXShell.jcfg
function StartButton:onClick()
  App:Debug("StartButton:onClick()")
  local cl = Window.Find('CLaunch', 'CLaunchWndClass')
  local cl_stat = cl:IsVisible()
  App:Debug(cl_stat)
  if cl_stat == true then
      cl:Hide()
  else
      App:Run(App.Path .. "\\..\\CLaunch\\CLaunch.exe", "/n")
   end
end

Shell.onHotKey['WIN+S'] = function()
  App:Debug("WIN+S hotkey is pressed.")
end

Shell.onHotKey['WIN+F'] = function()
  App:Debug("WIN+F hotkey is pressed.")
end

Shell.onHotKey['CAPSLOCK x 2'] = function()
  App:Debug("CAPSLOCK x 2 pressed.")
end

 -- a handler of OpenContainingFolder
function MyOpenContainingFolderHandler(lnkfile, realfile)
  -- local path = realfile:match('(.+)\\')
  App:Run('cmd', '/k echo ' .. realfile)

  -- totalcmd
  -- App:Run('X:\\Progra~1\\TotalCommander\\TOTALCMD64.exe', '/O /T /A \"' .. realfile .. '\"')
  -- XYplorer
  -- App:Run('X:\\Progra~1\\XYplorer\\XYplorer.exe', '/select=\"' .. realfile .. '\"')
end

-- 根据屏幕分辨率自动切换DPI
function MyDisplayChangedHandler()
  local cur_res_x = Screen:GetX()
  if last_res_x == cur_res_x then return end
  last_res_x = cur_res_x
  if last_res_x >= 3840 then
    Screen:DPI(150)
  elseif last_res_x >= 1440 then
    Screen:DPI(125)
  elseif last_res_x >= 800 then
    Screen:DPI(100)
  end
end

-- 自定义时钟区域的显示信息
-- 自定义显示示例:
--[[
    |  22:00 星期六  |
    |   2019-9-14    |
]]
-- FYI:https://www.lua.org/pil/22.1.html
function TrayClockTextFormatter_zhCN()
  local wd_name = {'日', '一', '二', '三', '四', '五', '六'}
  local now_time = os.time()
  local wd_disname =  ' 星期' .. wd_name[os.date('%w', now_time) + 1]
  local clocktext = os.date('%H:%M' .. TEXT(wd_disname) .. '\r\n%Y-%m-%d', now_time)
  App:SetVar('ClockText', clocktext)
end

-- custom tray clock display text
-- sample for:
--[[
    |  22:00 Sat  |
    |  2019-9-14  |
]]
-- FYI:https://www.lua.org/pil/22.1.html
function TrayClockTextFormatter_enUS()
  local now_time = os.time()
  local clocktext = os.date('%H:%M %a\r\n%Y-%m-%d', now_time)
  App:SetVar('ClockText', clocktext)
end
--"Original path":"X:\AI_Work\WinXShell\WinXShell.lua"
--"Original path":"X:\AI_Work\WinXShell\WinXShell.lua"
```

```
{
  "JS_README": {
    "此段可省略": true,
    "说明": [
      "WinXShell的jcfg配置文件的说明。",
      "需要自定义WinXShell中的开始菜单项目，任务栏大小，背景色，文字颜色等时,",
      "可利用此文件进行定义，未定义时，将利用默认设定。",
      "配置文件中的字符串资源可进行扩展:",
      "     【%】开始的字符串将进行环境变量扩展，需要使用%时，使用%%。",
      "     【#】开始的字符串将进行变量扩展，需要使用#时，使用#{#}。",
      "     【=】开始的字符串将识别为计算表达式，进行变量扩展后，计算其数值。(未实现)"
    ]
  },
  "JS_JCFG": {
    "此段可省略": true,
    "说明": [
      "外部WinXShell.jcfg配置文件加载设定。",
      "LOADPATH 内核配置文件中指定外部配置文件路径格式",
      "LOADIF 外部配置文件中可定义存在指定环境变量时加载"
    ],
    "LOADPATH": "?:\\WinXShell\\WinXShell.jcfg",
    "LOADIF": "WIN10PEx64-A"
  },
  "JS_JMACRO": {
    "此段可省略": true,
    "说明": [
      "jcfg文件中使用的内部固定变量名",
      "MACRO键值格式为:<内部名>,<别名>,<说明>",
      "<内部名>:修改无效",
      "<别  名>:可为内部名指定一个以【::】开头的别名,",
      "          预处理阶段将jcfg文件中的别名替换为内部名。",
      "          指定为空字符串将不进行任何处理,jcfg文件中需要使用内部名。",
      "<说  明>:仅对内部名进行说明，修改无效。"
    ],
    "JKEYNAME": [
      ["JS_README", "",           "段名。jcfg文件说明段"],
      ["JS_JMACRO", "",           "段名。可为内部名定义指定别名"],
      ["JS_JENV", "::环境变量",   "段名。用于定义环境变量"],
      ["JS_JVAR", "::程序变量",   "段名。可以定义方便内部多出使用的变量"],
      ["JS_FILEEXPLORER", "::文件管理器",    "段名。可对桌面进行设置"],
      ["3rd_filename", "::第3方文件管理器",  "属性名。"],
      ["JS_THEMES", "::主题段",     "段名。可定义多个主题"],
      ["JS_TASKBAR", "::任务栏", "段名。可对任务栏的高度，颜色等进行设置"],
      ["smallicon",        "::使用小图标",        "属性名。"],
      ["thumbnail",         "::显示任务缩略图",  "属性名。"],
      ["task_close_button", "::显示任务关闭按钮", "属性名。"],
      ["no_task_title",     "::隐藏任务标题",     "属性名。"],
      ["userebar",          "::使用Rebar",        "属性名。"],
      ["theme",             "::主题",             "属性名。"],
      ["height",            "::高度",             "属性名。"],
      ["JS_STARTMENU", "::开始菜单",          "段名。可对开始菜单进行设置"],
      ["start_pushed_bkcolor", "::开始按钮按下背景色", "属性名。"],
      ["start_icon_comment",   "::开始按钮设定值说明", "属性名。"],
      ["start_icon",     "::开始按钮",          "属性名。"],
      ["start_command", "::开始按钮动作",       "属性名。"],
      ["notopitems",     "::隐藏顶部菜单",      "属性名。"],
      ["noprograms",     "::隐藏程序菜单",      "属性名。"],
      ["nosettings",     "::隐藏设置菜单",      "属性名。"],
      ["nobrowse",       "::隐藏浏览菜单",      "属性名。"],
      ["noconnections", "::隐藏网络连接菜单", "属性名。"],
      ["nofind",          "::隐藏查找菜单",       "属性名。"],
      ["norun",           "::隐藏运行菜单",       "属性名。"],
      ["nologoff",        "::隐藏注销菜单",       "属性名。"],
      ["norestart",       "::隐藏重启菜单",       "属性名。"],
      ["noshutdown",      "::隐藏关机菜单",       "属性名。"],
      ["noterm",          "::隐藏终止菜单",       "属性名。"],
      ["JS_QUICKLAUNCH", "::快速启动栏",       "段名。可对快速启动栏进行设置"],
      ["maxiconsinrow",            "::单行最大图标数",            "属性名。"],
      ["3rd_startup_arguments", "::第3方文件管理器启动参数", "属性名。"],
      ["hide_showdesktop",       "::隐藏显示桌面图标",         "属性名。"],
      ["hide_fileexplorer",      "::隐藏资源管理器图标",       "属性名。"],
      ["hide_fixedsep",          "::隐藏分隔符",               "属性名。"],
      ["hide_usericons",         "::隐藏用户图标",             "属性名。"],
      ["folder",                 "::位置",                     "属性名。"],
      ["JS_NOTIFYAREA", "::托盘区域",          "段名。可对托盘区域进行设置"],
      ["hide_toggle_button",       "::隐藏显示隐藏图标按钮",         "属性名。"],
      ["hide_showdesktop_button", "::隐藏显示桌面按钮",              "属性名。"],
      ["handle_system_network",    "::接管系统网络图标左键单击事件", "属性名。"],
      ["handle_system_volume",     "::接管系统声音图标左键单击事件", "属性名。"],
      ["JS_NOTIFYCLOCK", "::时钟栏",            "段名。可对时钟栏进行设置"],
      ["visible", "::显示", "属性名。"],
      ["JS_DESKTOP", "::桌面",                "段名。可对桌面进行设置"],
      ["3rd_open_arguments", "::第3方文件管理器打开参数", "属性名。"],
      ["bkcolor",              "::桌面背景色",              "属性名。"],
      ["cascademenu",          "::多级菜单",                "属性名。"],
      ["iconsize",             "::桌面图标大小",            "属性名。"],
      ["wallpaperstyle",       "::桌面壁纸显示样式",         "属性名。有效值:0-拉伸(默认值);1-平铺;2-居中"],
      ["wallpaper",            "::桌面壁纸",                "属性名。"]
      ],
    "JTYPENAME":
    [
      ["JT_STARTMENU_FOLDER", "::开始菜单_文件夹类型", ""]
    ]
  },
  "JS_DAEMON": {
    "clockarea_click": {
      "command": "##{JVAR_MODULEPATH}\\#{JVAR_MODULENAME}",
      "parameters": "-code wxsUI('UI_Calendar','main.jcfg')"
    },
    "clockarea_dbclick": {
      "command": "timedate.cpl"
    },
    "handle_CAPS_double": false,
    "disable_showdesktop": false,
    "screen_brightness": 100
  },
  "::文件管理器": {
    "#::第3方文件管理器": "##{JVAR_MODULEPATH}\\explorer++.exe",
    "WIN+E": {
      "#::第3方文件管理器": ""
    },
    "#padding-bottom": 20
  },
  "::桌面": {
    "::第3方文件管理器打开参数": "\"%s\"",
    "::多级菜单": {
      "#WinXNew": "Directory\\Background\\shell\\WinXNew"
    },
    "#::桌面图标大小": 32,
    "背景色注释": "绿豆沙色",
    "::              ": [199, 237, 204],
        桌面背景色
    "显示样式注释": "0-拉伸(默认值);1-平铺;2-居中",
    "::桌面壁纸显示样式": 0,
    "壁纸注释": "设置为空,或者不设置,将使用系统设置",
    "::桌面壁纸": ""
  },
  "::主题段": {
    "blue": {
      "taskbar": {
        "bkcolor": [0, 120, 215],
        "#bkmode_values": "opaque; transparent; acrylic",
        "bkmode": "opaque",
        "transparency": 100,
        "task_line_color": [238, 238, 238],
        "textcolor": "0xffffff"
      }
    },
    "dark": {
      "taskbar": {
        "bkcolor": [38, 38, 38],
        "#bkmode_values": "opaque; transparent; acrylic",
        "bkmode": "transparent",
        "transparency": 64,
        "task_line_color": [238, 238, 238],
        "textcolor": "0xffffff"
      }
    },
    "light": {
      "taskbar": {
        "style": "light",
        "bkcolor": [238, 238, 238],
        "#bkmode_values": "opaque; transparent; acrylic",
        "bkmode": "acrylic",
        "transparency": 100,
        "task_line_color": [0, 120, 215],
        "textcolor": "0x000000"
      }
    }
  },
  "::任务栏": {
    "::显示": true,
    "::使用小图标": false,
    "::显示任务缩略图": true,
    "::显示任务关闭按钮": true,
    "::隐藏任务标题": false,
    "::使用Rebar": false,
    "::主题": "dark",
    "::高度": 40
  },
  "::开始菜单": {
    "::开始按钮按下背景色" : [0, 100, 180],
    "::开始按钮设定值说明": "theme;empty;custom;files",
    "::开始按钮": "theme",
    "::开始按钮动作": "",
    "::隐藏顶部菜单": false,
    "::隐藏程序菜单": false,
    "::隐藏设置菜单": true,
    "::隐藏浏览菜单": false,
    "::隐藏网络连接菜单": false,
    "::隐藏查找菜单": true,
    "::                    ": false,
        隐藏运行菜单
    "::隐藏注销菜单": true,
    "::隐藏重启菜单": false,
    "::隐藏关机菜单": false,
    "::隐藏终止菜单": true,
    "commands": {
      "reboot": {
        "command": "Wpeutil.exe",
        "parameters": "Reboot"
      },
      "shutdown": {
        "command": "Wpeutil.exe",
        "parameters": "Shutdown"
      }
    }
  },
  "::快速启动栏": {
    "::单行最大图标数": 20,
    "#::第3方文件管理器": "",
    "::第3方文件管理器启动参数": "",
    "::隐藏显示桌面图标": false,
    "::隐藏资源管理器图标": false,
    "::隐藏分隔符": false,
    "::隐藏用户图标": false,
    "::位置": "Microsoft\\Internet Explorer\\Quick Launch\\User Pinned\\TaskBar"
  },
  "::托盘区域": {
    "::隐藏显示隐藏图标按钮": false,
    "::隐藏显示桌面按钮": false,
    "#::接管系统网络图标左键单击事件": true,
    "#::接管系统声音图标左键单击事件": true
  },
  "::时钟栏": {
    "::显示": true
  }
}{
}
{
  "Original path":"X:\AI_Work\WinXShell\WinXShell.zh-CN.jcfg",
}
{
  "Original path":"X:\AI_Work\WinXShell\WinXShell.zh-CN.jcfg",
}
```

```
fixed_out = {}
table.insert(
                    fixed_out,
                    string.format("<i ..\\UI_DL\\app_ico\\wps.png><a wps2016><c #000000>WPS
2016</c></a> \n金山出品,无须多言\n")
               )--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\work.lua"
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="14" size="14" name="微软雅黑" default="true" />
    <Font shared="true" id="18" size="18" name="微软雅黑" />
    <Font shared="true" id="24" size="24" name="微软雅黑" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
    <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="Settings" value="设置" />
    <MultiLanguage id="Home" value="主页" />
    <MultiLanguage id="Customization" value="个性化" />

    <MultiLanguage id="Display" value="显示" />
    <MultiLanguage id="DPI" value="更改文本、应用等项目的大小" />
    <MultiLanguage id="Resolution" value="分辨率" />
    <MultiLanguage id="Rotate" value="方向" />
    <MultiLanguage id="Rotate_0" value="横向" />
    <MultiLanguage id="Rotate_90" value="纵向" />
    <MultiLanguage id="Rotate_180" value="横向(翻转)" />
    <MultiLanguage id="Rotate_270" value="纵向(翻转)" />

    <MultiLanguage id="Brightness" value="亮度" />

    <MultiLanguage id="Apply" value="应用<u>(A)</u>" />
    <MultiLanguage id="Restore" value="还原<u>R</u>" />

    <MultiLanguage id="Background" value="背景" />

    <MultiLanguage id="Colors" value="颜色" />
    <MultiLanguage id="SelectSysColorTheme" value="选择你的默认 Windows 模式" />
    <MultiLanguage id="SelectAppColorTheme" value="选择默认应用模式" />
    <MultiLanguage id="Light" value="亮" />
    <MultiLanguage id="Dark" value="暗" />
    <MultiLanguage id="TransparencyEffects" value="透明效果" />
    <MultiLanguage id="ApplyAccentColorDesc" value="在以下区域显示主题色" />
    <MultiLanguage id="StartTaskbarActionCenter" value="开始菜单、任务栏和操作中心" />
    <MultiLanguage id="TitlebarsAndWinBorders" value="标题栏和窗口边框" />

    <MultiLanguage id="Taskbar" value="任务栏" />
    <MultiLanguage id="AutoHideTaskbar" value="自动隐藏任务栏" />
    <MultiLanguage id="UseSmallButtons" value="使用小任务栏按钮" />
    <MultiLanguage id="CombineButtons" value="合并任务栏按钮" />
    <MultiLanguage id="Always" value="始终隐藏标签" />
    <MultiLanguage id="Auto" value="当任务栏占满时" />
    <MultiLanguage id="Never" value="从不" />

    <MultiLanguage id="On" value="开" />
    <MultiLanguage id="Off" value="关" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="16" size="16" name="微软雅黑" default="true" />
    <Font shared="true" id="24" size="24" name="微软雅黑" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
   <Font id="10" name="微软雅黑" size="10" shared="true" />
   <Font id="16" name="微软雅黑" size="16" shared="true" default="true" />
   <Font id="12" name="微软雅黑" size="12" shared="true" />
   <Font id="24" name="微软雅黑" size="24" shared="true" />
   <Font id="16u" name="微软雅黑"  size="16"shared="true" underline="true" />

   <MultiLanguage id="::Connected" value="已连接" />
   <MultiLanguage id="::Secured" value="安全" />
   <MultiLanguage id="::Open" value="开放" />
   <MultiLanguage id="::Connect" value="连接" />
   <MultiLanguage id="::Disconnect" value="断开连接" />
   <MultiLanguage id="msg_hidden_ssid" value="隐藏的网络" />
   <MultiLanguage id="msg_enter_ssid" value="请输入网络的名称(SSID)" />
   <MultiLanguage id="msg_enter_password" value="请输入连接密码" />
   <MultiLanguage id="Cancel" value="取消" />
   <MultiLanguage id="NetworkSettings" value="网络和 Internet 设置" />
   <MultiLanguage id="NetworkSettingsMsg" value="更改设置，例如为网络设置静态IP。" />
   <MultiLanguage id="Firewall" value="防火墙" />
   <MultiLanguage id="Proxy" value="代理" />
   <MultiLanguage id="::msg_select_device" value="请选择无线网络设备" />
   <MultiLanguage id="::msg_service_unavailable" value="无线连接服务不可用" />
   <MultiLanguage id="::msg_device_unavailable" value="无可用的无线网络设备" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="14" size="14" name="微软雅黑" default="true" />
    <Font shared="true" id="24" size="24" name="微软雅黑" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
    <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="Settings" value="设置" />
    <MultiLanguage id="Home" value="主页" />
    <MultiLanguage id="Customization" value="个性化" />

    <MultiLanguage id="Display" value="显示" />
    <MultiLanguage id="DPI" value="更改文本、应用等项目的大小" />
    <MultiLanguage id="Resolution" value="分辨率" />
    <MultiLanguage id="Rotate" value="方向" />
    <MultiLanguage id="Rotate_0" value="横向" />
    <MultiLanguage id="Rotate_90" value="纵向" />
    <MultiLanguage id="Rotate_180" value="横向(翻转)" />
    <MultiLanguage id="Rotate_270" value="纵向(翻转)" />

    <MultiLanguage id="Brightness" value="亮度" />

    <MultiLanguage id="Apply" value="应用<u>(A)</u>" />
    <MultiLanguage id="Restore" value="还原<u>R</u>" />

    <MultiLanguage id="Background" value="背景" />

    <MultiLanguage id="Colors" value="颜色" />
    <MultiLanguage id="SelectSysColorTheme" value="选择你的默认 Windows 模式" />
    <MultiLanguage id="SelectAppColorTheme" value="选择默认应用模式" />
    <MultiLanguage id="Light" value="亮" />
    <MultiLanguage id="Dark" value="暗" />
    <MultiLanguage id="TransparencyEffects" value="透明效果" />
    <MultiLanguage id="ApplyAccentColorDesc" value="在以下区域显示主题色" />
    <MultiLanguage id="StartTaskbarActionCenter" value="开始菜单、任务栏和操作中心" />
    <MultiLanguage id="TitlebarsAndWinBorders" value="标题栏和窗口边框" />

    <MultiLanguage id="Taskbar" value="任务栏" />
    <MultiLanguage id="AutoHideTaskbar" value="自动隐藏任务栏" />
    <MultiLanguage id="UseSmallButtons" value="使用小任务栏按钮" />
    <MultiLanguage id="CombineButtons" value="合并任务栏按钮" />
    <MultiLanguage id="Always" value="始终隐藏标签" />
    <MultiLanguage id="Auto" value="当任务栏占满时" />
    <MultiLanguage id="Never" value="从不" />

    <MultiLanguage id="On" value="开" />
    <MultiLanguage id="Off" value="关" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="12" size="12" name="微软雅黑" default="true" />
    <Font shared="true" id="14" size="14" name="微软雅黑" />
    <Font shared="true" id="16" size="16" name="微软雅黑" />
    <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
   <Font shared="true" id="16" size="16" name="微软雅黑" default="true" />
   <Font shared="true" id="24" size="24" name="微软雅黑" />
   <Font shared="true" id="40" size="40" name="微软雅黑" />
   <Font shared="true" id="sym" size="12" name="Segoe MDL2 Assets" />
   <Font shared="true" id="sym16" size="16" name="Segoe MDL2 Assets" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="12" size="12" name="微软雅黑" default="true" />
    <Font shared="true" id="16" size="16" name="微软雅黑" />
    <Font shared="true" id="24" size="24" name="微软雅黑" />

    <MultiLanguage id="info_w0" value="星期日" />
    <MultiLanguage id="info_w1" value="星期一" />
    <MultiLanguage id="info_w2" value="星期二" />
    <MultiLanguage id="info_w3" value="星期三" />
    <MultiLanguage id="info_w4" value="星期四" />
    <MultiLanguage id="info_w5" value="星期五" />
    <MultiLanguage id="info_w6" value="星期六" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="12" size="12" name="微软雅黑" />
    <Font shared="true" id="16" size="16" name="微软雅黑" default="true" />
    <Font shared="true" id="16b" size="16" name="微软雅黑" bold="true" />
    <Font shared="true" id="18" size="18" name="微软雅黑" />
    <Font shared="true" id="18b" size="18" name="微软雅黑" bold="true" />
    <Font shared="true" id="20" size="20" name="微软雅黑" />
    <Font shared="true" id="40" size="40b" name="微软雅黑" />
    <Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="星期日" />
    <MultiLanguage id="info_w1" value="星期一" />
    <MultiLanguage id="info_w2" value="星期二" />
    <MultiLanguage id="info_w3" value="星期三" />
    <MultiLanguage id="info_w4" value="星期四" />
    <MultiLanguage id="info_w5" value="星期五" />
    <MultiLanguage id="info_w6" value="星期六" />

    <MultiLanguage id="w7" value="日" />
    <MultiLanguage id="w1" value="一" />
    <MultiLanguage id="w2" value="二" />
    <MultiLanguage id="w3" value="三" />
    <MultiLanguage id="w4" value="四" />
    <MultiLanguage id="w5" value="五" />
    <MultiLanguage id="w6" value="六" />

    <MultiLanguage id="OK" value="确定" />
    <MultiLanguage id="CANCEL" value="取消" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="16" size="16" name="微软雅黑" default="true" />
    <Font shared="true" id="24" size="24" name="微软雅黑" />

    <MultiLanguage id="Settings" value="设置" />
    <MultiLanguage id="Display Settings" value="显示设置" />
    <MultiLanguage id="Resolution" value="分辨率" />
    <MultiLanguage id="Rotate" value="方向" />
    <MultiLanguage id="Rotate_0" value="横向" />
    <MultiLanguage id="Rotate_90" value="纵向" />
    <MultiLanguage id="Rotate_180" value="横向(翻转)" />
    <MultiLanguage id="Rotate_270" value="纵向(翻转)" />
    <MultiLanguage id="Brightness" value="亮度" />
    <MultiLanguage id="Apply" value="应用<u>(A)</u>" />
    <MultiLanguage id="Close" value="关闭<u>(C)</u>" />
    <MultiLanguage id="Restore" value="还原<u>R</u>" />
    <MultiLanguage id="Cancel" value="取消<u>(C)</u>" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="18" size="18" name="微软雅黑" default="true" />
    <Font shared="true" id="24" size="24" name="微软雅黑" />
    <Font shared="true" id="40b" size="40b" name="微软雅黑" />
    <Font shared="true" id="128" size="28" name="Segoe MDL2 Assets" />
    <Font shared="true" id="130" size="30" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="星期天" />
    <MultiLanguage id="info_w1" value="星期一" />
    <MultiLanguage id="info_w2" value="星期二" />
    <MultiLanguage id="info_w3" value="星期三" />
    <MultiLanguage id="info_w4" value="星期四" />
    <MultiLanguage id="info_w5" value="星期五" />
    <MultiLanguage id="info_w6" value="星期六" />

    <MultiLanguage id="SignIn" value="登录" />
    <MultiLanguage id="ForgotPassword" value="忘记密码" />
    <MultiLanguage id="Logon" value="登录" />

    <MultiLanguage id="will autologon with the $user account in $countdown second(s)" value="将在$countdown秒后自动使用$user账户登录" />
    <MultiLanguage id="Welcome to Windows ..." value="正在登录，请稍等 ..." />
   <MultiLanguage id="LogonInfo" value="按Alt + S键可使用SYSTEM账户登录，按Alt + A键可使用Administrator账户登录。" />
</Locale>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="16" size="16" name="微软雅黑" default="true" />
    <Font shared="true" id="24" size="24" name="微软雅黑" />

<!-- 以下是关于main.xml中的 <Style imagerectstyle="l,t,w,h" />语句的说明 -->
    <!-- 设置图像中的source、dest定义采用left,top,width,height风格 -->
    <!-- （原风格为left,top,right,bottom，需要额外计算，不方便应用风格定义） -->
    <!-- include的xml定义文件需要使用相同的imagerectstyle -->

    <MultiLanguage id="Power" value="电源" />
    <MultiLanguage id="CloseComputer" value="关闭计算机" />
    <MultiLanguage id="Shutdown" value="关机" />
    <MultiLanguage id="Restart" value="重启" />
    <MultiLanguage id="Hibernate" value="休眠" />
    <MultiLanguage id="Standby" value="待机" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
  <Font shared="true" id="12" size="12" name="細明體" />
  <Font shared="true" id="16" size="16" name="細明體" default="true" />
  <Font shared="true" id="16b" size="16" name="細明體" bold="true" />
  <Font shared="true" id="18" size="18" name="細明體" />
  <Font shared="true" id="18b" size="18" name="細明體" bold="true" />
  <Font shared="true" id="20" size="20" name="細明體" />
  <Font shared="true" id="40" size="40b" name="細明體" />
  <Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets" />

  <MultiLanguage id="info_w0" value="星期日" />
  <MultiLanguage id="info_w1" value="星期一" />
  <MultiLanguage id="info_w2" value="星期二" />
  <MultiLanguage id="info_w3" value="星期三" />
  <MultiLanguage id="info_w4" value="星期四" />
  <MultiLanguage id="info_w5" value="星期五" />
  <MultiLanguage id="info_w6" value="星期六" />

  <MultiLanguage id="w7" value="日" />
  <MultiLanguage id="w1" value="一" />
  <MultiLanguage id="w2" value="二" />
  <MultiLanguage id="w3" value="三" />
  <MultiLanguage id="w4" value="四" />
  <MultiLanguage id="w5" value="五" />
  <MultiLanguage id="w6" value="六" />

  <MultiLanguage id="OK" value="確定" />
  <MultiLanguage id="CANCEL" value="取消" />
</Locale>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<Locale>
    <Font shared="true" id="12" size="12" name="細明體" />
    <Font shared="true" id="16" size="16" name="細明體" default="true" />
    <Font shared="true" id="16b" size="16" name="細明體" bold="true" />
    <Font shared="true" id="18" size="18" name="細明體" />
    <Font shared="true" id="18b" size="18" name="細明體" bold="true" />
    <Font shared="true" id="20" size="20" name="細明體" />
    <Font shared="true" id="40" size="40b" name="細明體" />
    <Font shared="true" id="i16" size="16" name="Segoe MDL2 Assets" />

    <MultiLanguage id="info_w0" value="星期日" />
    <MultiLanguage id="info_w1" value="星期一" />
    <MultiLanguage id="info_w2" value="星期二" />
    <MultiLanguage id="info_w3" value="星期三" />
    <MultiLanguage id="info_w4" value="星期四" />
    <MultiLanguage id="info_w5" value="星期五" />
    <MultiLanguage id="info_w6" value="星期六" />

    <MultiLanguage id="w7" value="日" />
    <MultiLanguage id="w1" value="一" />
    <MultiLanguage id="w2" value="二" />
    <MultiLanguage id="w3" value="三" />
    <MultiLanguage id="w4" value="四" />
    <MultiLanguage id="w5" value="五" />
    <MultiLanguage id="w6" value="六" />

    <MultiLanguage id="OK" value="確定" />
    <MultiLanguage id="CANCEL" value="取消" />
</Locale>
```

```
temp = os.getenv("temp")
-- 正则匹配获取多行
function string.gfind(stdout, patten)
    local i, j = 0, 0
    return function()
        i, j = string.find(stdout, patten, j + 1)
        if (i == nil) then -- end find
            return nil
        end
        return string.sub(stdout, i, j)
    end
end

local path = [[e:\products.xml]]
local file = io.open(path, "rb")
text = file:read("*a")
file:close()
--text_ansi = winapi.encode(winapi.CP_UTF16, winapi.CP_ACP, text)
local index_regex = "<File id=\"\">"
local edition_regex = "<Edition>([^<]+)</Edition>"
local lcode_regex = "<LanguageCode>([^<]+)</LanguageCode>"
local filepath_regex = "<FilePath>([^<]+)</FilePath>"

ver={}
for index_str in string.gfind(text, "<File id=*.-</File>") do
local index_name = index_str:match(index_regex)
local index_edition = index_str:match(edition_regex)
local index_filepath = index_str:match(filepath_regex)
local index_lcode = index_str:match(lcode_regex)
if index_lcode == "zh-cn" and index_str:find("CLIENTCHINA") and index_str:find("x64") then
ver.chinax64url = index_filepath
```

```
elseif index_lcode == "zh-cn" and index_str:find("CLIENTCHINA") and index_str:find("x86")
then
ver.chinax86url = index_filepath

elseif index_lcode == "zh-cn" and index_str:find("CLIENTBUSINESS") and index_str:find("x64")
then
ver.businessx64url = index_filepath

elseif index_lcode == "zh-cn" and index_str:find("CLIENTBUSINESS") and index_str:find("x86")
then
ver.businessx86url = index_filepath

elseif index_lcode == "zh-cn" and index_str:find("Ultimate") and
index_str:find("CLIENTCONSUMER") and index_str:find("x64") then
ver.consumerx64url = index_filepath

elseif index_lcode == "zh-cn" and index_str:find("Ultimate") and
index_str:find("CLIENTCONSUMER") and index_str:find("x86") then
ver.consumerx86url = index_filepath



end


end
table.insert(ver,string.format("64位家庭中文版：下载地址%s", ver.chinax64url))
table.insert(ver,string.format("32位家庭中文版：下载地址%s", ver.chinax86url))
table.insert(ver,string.format("64位商业中文版：下载地址%s", ver.businessx64url))
table.insert(ver,string.format("32位商业中文版：下载地址%s", ver.businessx86url))
table.insert(ver,string.format("64位零售中文版：下载地址%s", ver.consumerx64url))
table.insert(ver,string.format("32位零售中文版：下载地址%s", ver.consumerx86url))
```

```
all = table.concat(ver, "\n")
alert(all)--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\白嫖微软镜像解析乱写版.lua"
```

```
@echo off
setlocal enabledelayedexpansion

:: 遍历 .jcfg 文件
for /r %%F in (*.jcfg) do (
    call :process_file "%%F"
)

:: 遍历 .lua 文件
for /r %%F in (*.lua) do (
    call :process_file "%%F"
)

:: 遍历 .xml 文件
for /r %%F in (*.xml) do (
    call :process_file "%%F"
)

:: 遍历 .bat 文件
for /r %%F in (*.bat) do (
    call :process_file "%%F"
)

:: 遍历 .cmd 文件
for /r %%F in (*.cmd) do (
    call :process_file "%%F"
)

echo 完成!
pause
goto :eof
```

```
:: 处理文件并生成新的文件名
:process_file
set "file=%~1"
set "filename=%~nx1"
set "dir=%~dp1"
set "newfile=%file%.txt"

:: 检查是否存在同名的 .txt 文件
set /a counter=1
if exist "!newfile!" (
    :rename_loop
    set "newfile=%dir%!filename!_!counter!.txt"
    if exist "!newfile!" (
        set /a counter+=1
        goto :rename_loop
    )
)

:: 复制原文件到新文件
copy "%file%" "!newfile!" >nul

exit /b
```

```
@echo off
setlocal enabledelayedexpansion

:: 遍历 .jcfg 文件
for /r %%F in (*.jcfg) do (
    call :process_file "%%F"
)

:: 遍历 .lua 文件
for /r %%F in (*.lua) do (
    call :process_file "%%F"
)

:: 遍历 .xml 文件
for /r %%F in (*.xml) do (
    call :process_file "%%F"
)

:: 遍历 .bat 文件
for /r %%F in (*.bat) do (
    call :process_file "%%F"
)

:: 遍历 .cmd 文件
for /r %%F in (*.cmd) do (
    call :process_file "%%F"
)

echo 完成!
pause
goto :eof
```

```
:: 处理文件并生成新的文件名
:process_file
set "file=%~1"
set "filename=%~nx1"
set "dir=%~dp1"
set "newfile=%file%.txt"

:: 检查是否存在同名的 .txt 文件
set /a counter=1
if exist "!newfile!" (
    :rename_loop
    set "newfile=%dir%!filename!_!counter!.txt"
    if exist "!newfile!" (
        set /a counter+=1
        goto :rename_loop
    )
)

:: 复制原文件到新文件
copy "%file%" "!newfile!" >nul

exit /b
```

```
卷 Sata固态 的文件夹 PATH 列表
卷序列号为 E095-1888
X:.
├─Docs
│  ├─classes
│  ├─libraries
│  ├─manual
│  └─program
├─FileExpRefresh
└─wxsUI
    ├─UI_Calendar
    │  ├─locales
    │  └─themes
    ├─UI_Launcher
    │  ├─locales
    │  └─themes
    ├─UI_LED
    │  └─locales
    ├─UI_Logon
    │  └─locales
    ├─UI_NotifyInfo
    ├─UI_Resolution
    │  ├─locales
    │  └─themes
    ├─UI_Settings
    │  ├─locales
    │  ├─parts
    │  └─themes
    ├─UI_Shutdown
    │  └─locales
    ├─UI_SystemInfo
```

```
│   ├─locales
│   └─themes
├─UI_TrayPanel
│   ├─locales
│   └─themes
├─UI_Volume
│   ├─locales
│   ├─rc
│   │   └─light
│   └─themes
└─UI_WIFI
    ├─locales
    ├─rc
    │   ├─default
    │   └─light
    ├─themes
    └─trayicons
        ├─dark
        ├─default
        ├─win11dark
        └─win11default
```

--"Original path":"X:\AI_Work\WinXShell\目录树.txt"

--"Original path":"X:\AI_Work\WinXShell\目录树.txt"

petools放在任意目录下
penetwork配置文件，可以在pe里修改好，可以在虚拟机或别的电脑上改好，
然后复制到petools目录下，petool复制
可以让pe启动后执行任意任务

```
X:\AI_Work\WinXShell\History.md
X:\AI_Work\WinXShell\LUA_TEST.bat
X:\AI_Work\WinXShell\LUA_Tests.bat
X:\AI_Work\WinXShell\UI_TEST.bat
X:\AI_Work\WinXShell\wallpaper.jpg
X:\AI_Work\WinXShell\WinXShell.jcfg
X:\AI_Work\WinXShell\WinXShell.lua
X:\AI_Work\WinXShell\WinXShell.zh-CN.jcfg
X:\AI_Work\WinXShell\WinXShellC_x64.exe
X:\AI_Work\WinXShell\WinXShellC_x86.exe
X:\AI_Work\WinXShell\WinXShell_x64.exe
X:\AI_Work\WinXShell\WinXShell_x86.exe
X:\AI_Work\WinXShell\wxsStub.dll
X:\AI_Work\WinXShell\wxsStub32.dll
X:\AI_Work\WinXShell\工作.bat
X:\AI_Work\WinXShell\总说明.txt
X:\AI_Work\WinXShell\所有文件列表.txt
X:\AI_Work\WinXShell\目录树.txt
X:\AI_Work\WinXShell\Docs\index.html
X:\AI_Work\WinXShell\Docs\ldoc_custom.css
X:\AI_Work\WinXShell\Docs\ldoc_fixed.css
X:\AI_Work\WinXShell\Docs\classes\App.html
X:\AI_Work\WinXShell\Docs\classes\Shell.html
X:\AI_Work\WinXShell\Docs\classes\System.html
X:\AI_Work\WinXShell\Docs\libraries\Lua-Engine.html
X:\AI_Work\WinXShell\Docs\libraries\UI-Engine.html
X:\AI_Work\WinXShell\Docs\libraries\Windows-Shell.html
X:\AI_Work\WinXShell\Docs\manual\csidl.md.html
X:\AI_Work\WinXShell\Docs\manual\history.md.html
X:\AI_Work\WinXShell\Docs\program\WinXShell.html
X:\AI_Work\WinXShell\FileExpRefresh\wxsStub.dll
```

```
X:\AI_Work\WinXShell\FileExpRefresh\wxsStub32.dll
X:\AI_Work\WinXShell\wxsUI\segmdl2.ttf
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\BrightnessController.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\Calendar.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\Calendar.lua
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\Calendar.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\Calendar10.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\Calendar10L.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\close.png
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\lunar.lua
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\lunar.zh-CN.lua
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\lunar.zh-TW.lua
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\sliderbk.png
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\sliderfr.png
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\thumb.png
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\locales\ko-KR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\locales\tr-TR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\locales\zh-TW.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\themes\blue.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\themes\dark.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\themes\default.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\themes\light.xml
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\themes\today-blue.png
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\themes\today-dark.png
```

```
X:\AI_Work\WinXShell\wxsUI\UI_Calendar\themes\today-light.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\apps_tree.lua
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\calc.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\chrome.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\close.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\folder.ico
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\full.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\mail.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\main-full.lua
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\maps.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\movies.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\notepad.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\office.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\onenote.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\paint3d.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\photos.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\scrollbar.bmp
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\scrollbar_ltwh.xml
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\startmenu.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\startmenu.lua
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\vs.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\weather.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\xbox.png
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\locales\ko-KR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\locales\tr-TR.xml
```

```
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\themes\dark.xml
X:\AI_Work\WinXShell\wxsUI\UI_Launcher\themes\default.xml
X:\AI_Work\WinXShell\wxsUI\UI_LED\bk.png
X:\AI_Work\WinXShell\wxsUI\UI_LED\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_LED\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_LED\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_LED\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_LED\UI_Debug.en-US.bat
X:\AI_Work\WinXShell\wxsUI\UI_LED\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_LED\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_LED\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_Logon\Admin_logo.png
X:\AI_Work\WinXShell\wxsUI\UI_Logon\bk.jpg
X:\AI_Work\WinXShell\wxsUI\UI_Logon\bk2.jpg
X:\AI_Work\WinXShell\wxsUI\UI_Logon\bk3.jpg
X:\AI_Work\WinXShell\wxsUI\UI_Logon\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Logon\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_Logon\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_Logon\ms_logo.png
X:\AI_Work\WinXShell\wxsUI\UI_Logon\SYSTEM_logo.png
X:\AI_Work\WinXShell\wxsUI\UI_Logon\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_Logon\UI_LogonPE.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Logon\windows.png
X:\AI_Work\WinXShell\wxsUI\UI_Logon\wp.jpg
X:\AI_Work\WinXShell\wxsUI\UI_Logon\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_Logon\locales\ko-KR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Logon\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_Logon\locales\tr-TR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Logon\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\close.png
```

```
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\close_hot.png
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\error.png
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\info.png
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\main_cn.xml
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\main_ru.xml
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_NotifyInfo\warn.png
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\close.png
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\combo_hot.png
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\combo_nor.png
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\display.ico
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\Display.png
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\scrollbar.bmp
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\scrollbar_ltwh.xml
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\settings.ico
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\Shadow.png
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\sliderbk.png
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\sliderfr.png
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\thumb.png
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\wallpaper.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\wallpaper.lua
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\wallpaper.xml
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\wp_script.ini
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\locales\en-US.xml
```

```
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\locales\es-ES.xml
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\locales\ko-KR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\themes\dark.xml
X:\AI_Work\WinXShell\wxsUI\UI_Resolution\themes\default.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\close.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\KB_Layout.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Settings\KB_Layout.lua
X:\AI_Work\WinXShell\wxsUI\UI_Settings\KB_Layout.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Settings\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_Settings\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\scrollbar.bmp
X:\AI_Work\WinXShell\wxsUI\UI_Settings\scrollbar_ltwh.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\settings.ico
X:\AI_Work\WinXShell\wxsUI\UI_Settings\sliderbk.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\sliderfr.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\thumb.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_Settings\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\locales\es-ES.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\locales\ko-KR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\locales\tr-TR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.background.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.colors.lua
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.colors.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.display.lua
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.display.xml
```

```
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.folderoptions.lua
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.folderoptions.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.taskbar.lua
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\page.taskbar.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\tab.nav.lua
X:\AI_Work\WinXShell\wxsUI\UI_Settings\parts\tab.nav.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\check-dark.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\check.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\ComboButton-dark.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\ComboButton.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\dark.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\default.xml
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\radio-dark.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\radio.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\switch-dark.png
X:\AI_Work\WinXShell\wxsUI\UI_Settings\themes\switch.png
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\bk.jpg
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\blur.jpg
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\cbtn128.png
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\close.png
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\full.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\full.lua
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\full.xml
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\power.ico
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\power.png
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\reboot.png
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\rebootB.png
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\Shadow.png
```

```
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\shutdown.png
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\shutdownB.png
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\es-ES.xml
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\ko-KR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\main.en-US.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\main.zh-CN.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\tr-TR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Shutdown\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\close.png
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\handle_prop.reg
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\MYPC.ico
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\MYPC_16x16.ico
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\scrollbar.bmp
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\scrollbar_ltwh.xml
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\sec.ico
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\windows.png
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\win_10.0.png
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\win_11.0.png
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\win_6.1.png
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\win_server.png
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\themes\dark.xml
```

```
X:\AI_Work\WinXShell\wxsUI\UI_SystemInfo\themes\default.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\BrightController.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\Calendar10L.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\close.png
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\lunar.lua
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\lunar.zh-CN.lua
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\lunar.zh-TW.lua
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\sliderbk.png
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\sliderfr.png
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\SystemInfo.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\thumb.png
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\locales\ko-KR.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\locales\tr-TR.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\locales\zh-TW.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\themes\default.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\themes\light.xml
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\themes\today-dark.png
X:\AI_Work\WinXShell\wxsUI\UI_TrayPanel\themes\today-light.png
X:\AI_Work\WinXShell\wxsUI\UI_Volume\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_Volume\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_Volume\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_Volume\main10.xml
X:\AI_Work\WinXShell\wxsUI\UI_Volume\sliderbk.png
X:\AI_Work\WinXShell\wxsUI\UI_Volume\sliderfr.png
```

```
X:\AI_Work\WinXShell\wxsUI\UI_Volume\snd_1.ico
X:\AI_Work\WinXShell\wxsUI\UI_Volume\snd_2.ico
X:\AI_Work\WinXShell\wxsUI\UI_Volume\snd_3.ico
X:\AI_Work\WinXShell\wxsUI\UI_Volume\snd_mute.ico
X:\AI_Work\WinXShell\wxsUI\UI_Volume\thumb-light.png
X:\AI_Work\WinXShell\wxsUI\UI_Volume\thumb.png
X:\AI_Work\WinXShell\wxsUI\UI_Volume\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_Volume\Windows Background.wav
X:\AI_Work\WinXShell\wxsUI\UI_Volume\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_Volume\locales\ko-KR.xml
X:\AI_Work\WinXShell\wxsUI\UI_Volume\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_Volume\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_Volume\rc\light\snd_1.ico
X:\AI_Work\WinXShell\wxsUI\UI_Volume\rc\light\snd_2.ico
X:\AI_Work\WinXShell\wxsUI\UI_Volume\rc\light\snd_3.ico
X:\AI_Work\WinXShell\wxsUI\UI_Volume\rc\light\snd_mute.ico
X:\AI_Work\WinXShell\wxsUI\UI_Volume\themes\dark.xml
X:\AI_Work\WinXShell\wxsUI\UI_Volume\themes\default.xml
X:\AI_Work\WinXShell\wxsUI\UI_Volume\themes\light.xml
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\main.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\scrollbar_ltwh.xml
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\SSIDItem.xml
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\locales\es-ES.xml
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\locales\ko-KR.xml
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\locales\ru-RU.xml
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\locales\tr-TR.xml
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\locales\zh-CN.xml
```

```
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\pwd_asterisk.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\pwd_pain.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\default\0.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\default\100.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\default\25.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\default\50.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\default\75.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\light\0.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\light\100.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\light\25.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\light\50.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\rc\light\75.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\themes\ComboButton-dark.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\themes\ComboButton-default.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\themes\ComboButton-light.png

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\themes\dark.xml

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\themes\default.xml

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\themes\light.xml

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\themes\scrollbar-dark.bmp

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\themes\scrollbar-light.bmp

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\themes\scrollbar.bmp

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_signal_0.ico

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_signal_100.ico

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_signal_25.ico

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_signal_50.ico

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_signal_75.ico

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_wired_dis.ico

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_wired_internet.ico

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_wired_lan.ico

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_wireless_dis.ico

X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_wireless_lan.ico
```

```
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_wireless_noconn.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\dark\tray_network_wireless_noconn_legacy.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_signal_0.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_signal_100.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_signal_25.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_signal_50.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_signal_75.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_wired_dis.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_wired_internet.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_wired_lan.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_wireless_dis.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_wireless_lan.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_wireless_noconn.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\default\tray_network_wireless_noconn_legacy.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_signal_0.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_signal_100.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_signal_25.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_signal_50.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_signal_75.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_wired_dis.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_wired_internet.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_wired_lan.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_wireless_dis.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_wireless_lan.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_wireless_noconn.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11dark\tray_network_wireless_noconn_legacy.ic
o
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_signal_0.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_signal_100.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_signal_25.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_signal_50.ico
```

```
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_signal_75.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_wired_dis.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_wired_internet.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_wired_lan.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_wireless_dis.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_wireless_lan.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_wireless_noconn.ico
X:\AI_Work\WinXShell\wxsUI\UI_WIFI\trayicons\win11default\tray_network_wireless_noconn_legacy.ico
```

```
function execlines(cmds)

for line in cmds:gmatch("([^\n]*)\n?") do

            --alert(line)

            exec("/hide /wait", line)

end
end


local cmdlines = [[
bcdedit /set %guid% path %bootapp_path%
bcdedit /deletevalue %guid% locale
bcdedit /deletevalue %guid% inherit
bcdedit /deletevalue %guid% default
bcdedit /deletevalue %guid% displayorder
bcdedit /deletevalue %guid% timeout
bcdedit /deletevalue %guid% nointegritychecks
bcdedit /deletevalue %guid% testsigning
bcdedit /set {fwbootmgr) %guid% /addlast
]]


--添加启动项
os.setenv('bootapp_name', "grubfmx64_app")
os.setenv('bootapp_path', "\\EFI\\Boot\\grubfmx64.efi")
local exitcode, stdout = winapi.execute([["bcdedit /copy {bootmgr} /d "%bootapp_name%"]])
bootfile_id = string.match(stdout, "{(%x%x%x%x%x%x%x%x%-%x%x%x%x%-%x%x%x%x%-%x%x%x%x%-
%x%x%x%x%x%x%x%x%x%x%x%x)}")

os.setenv('guid', "{" .. bootfile_id .. "}")

cmdlines = cmdlines:envstr()

execlines(cmdlines)

--"原路径":"X:\AI_Work\WinXShell\wxsUI\UI_AppStore\添加uefi启动文件.lua"
```

X:\AI_Work\WinXShell\工作.bat

X:\AI_Work\WinXShell\工作.bat.txt

X:\AI_Work\WinXShell\总说明.txt

X:\AI_Work\WinXShell\所有文件列表.txt

X:\AI_Work\WinXShell\目录树.txt

X:\AI_Work\WinXShell\自写文件列表.txt

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\add_efifile_boot.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\AppStore.bat

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\appstore.ico

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\AppStoreUP.bat

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\AppStore_bak.bat

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bk.png

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\checkgrubfm.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\close.png

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\diy_nav.lua.Simple

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_bcdedit.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_getofflineip.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_nbdl.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_ntboot.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_ntboot20221021.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_ntboot202210242032.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_ntboot_.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_openfile.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_StartOptions.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\helper_verysync.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\icon.ico

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\loading.jcfg

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\loading.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\loading.xml

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\Lua_Getpid_Demo.lua

X:\AI_Work\WinXShell\wxsUI\UI_AppStore\main.jcfg

```
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\main.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\main.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\md5.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nav.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbappstore.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_addsoft.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_bootfire.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_bootfire.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_bootfire.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_ck.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_ck.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_ck.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_ck_bg.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_ck_bg_selected.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_ex_drv.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_mknbp.bat
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_mknbp.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_nbdl.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_nbdl.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_nbdl.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_nbdl_Baipiao.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_NVME.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_page.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_page.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_page.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_Panfu.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_petools.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_powercfg.jcfg
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_powercfg.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_powercfg.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_tool.jcfg
```

```
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_tool.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\nbtool_tool.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\note.txt
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\page.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\PEcheckdisk.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\PEfindjob.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\PEwtip.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\scrollbar.bmp
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\scrollbar_ltwh.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\StartOptions.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\timesync.exe
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\UI_Debug.bat
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\utils.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\_navData.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\_pageData.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\添加uefi启动文件.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\app_ico\nt6.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\00-vhdboot.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\AppStore_20230328.bat
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot20221021.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot202210242032.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot20240423.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot20240424.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_ntboot_20230327.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\helper_verysync.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\LUA_Timer.bat
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\nbtool_bootfire.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\nbtool_bootfire.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\nbtool_bootfire_bak.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\白嫖微软镜像解析乱写版.lua
```

```
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\20230301\helper_ntboot.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\bak_Script\20230301\nbtool_bootfire.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\locales\en-US.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\locales\zh-CN.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\about.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\chat.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\media.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\petools.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\remote.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\vm.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\rules\work.lua
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\soft\shutdown.exe
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\soft\batch_tool\Printer_tools.bat
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\soft\batch_tool\WiFi_Hotspot.bat
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\soft\petools\dp.ini
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\soft\petools\ghostsrvbatch.7z
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\soft\petools\PENetwork.ini
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\soft\petools\petools.ini
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\soft\petools\petools自动装驱动VC映射存档.ini
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\soft\petools\说明.txt
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\baise.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\bg.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\check-dark.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\check.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\ComboButton-dark.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\ComboButton.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\dark.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\default.xml
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\left.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\nbdl_bg.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\radio-dark.png
```

```
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\radio.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\sliderbk.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\sliderfr.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\switch-dark.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\switch.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\thumb.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\themes\thumb_.png
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\types\media
X:\AI_Work\WinXShell\wxsUI\UI_AppStore\types\work
```

我是总说明，说明了一切，所有项目的解释
目录树.txt由tree命令生成，是整个winxshell项目原来的目录列表
所有文件列表.txt是目录树中所有文件清单，是整个winxshell项目原来的文件列表
其它文本里都记录了这个目录中原来的路径和名字--也就是"Original path"