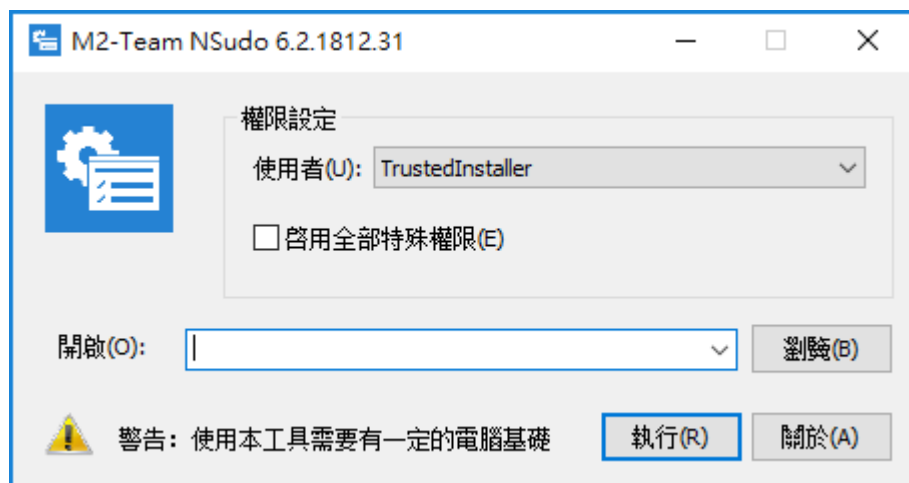


NSudo | 系統管理工具包



目錄

- [關於 NSudo](#)
 - [特性列表](#)
 - [系統要求](#)
 - [項目原型](#)
 - [使用了 NSudo 的第三方項目](#)
 - [第三方介紹](#)
 - [成為 NSudo 的贊助者](#)
 - [獲得支持](#)
- [使用方法](#)
 - [下載 NSudo](#)
 - [NSudo Launcher](#)
 - [NSudo 惡魔模式](#)
 - [NSudo 共享庫](#)
- [許可協議](#) (參閱 License.txt)
- [相關人士](#) (參閱 People.txt)
- [發行日志](#)

關於 NSudo

特性列表

- 以 MIT 許可證發行
- 提供 x86, x86-64, ARM64 二進制
- 支持 Windows Vista 及之後版本
- 使用初雨團隊的 VC-LTL 和 libkcrct 以獲取更小的二進制體積
- 使用 C++17, 但在大部分情況下只使用核心語言特性
- NSudo Launcher
 - 以 TrustedInstaller 訪問令牌運程序
 - 以 System 訪問令牌運程序
 - 以當前用戶的訪問令牌運程序
 - 注: 如果用戶賬戶控制即 UAC 沒有被禁用, 則該模式的權限與標準用戶等價
 - 以提升後的當前用戶的訪問令牌運程序
 - 注: 該模式的權限與提升後的用戶等價
 - 以當前進程的訪問令牌運程序
 - 注: 該模式的權限與提升後的用戶等價
 - 以當前進程 LUA 模式的訪問令牌運程序
 - 注: 該模式的權限與標準用戶等價且該實現和 Internet Explorer 瀏覽器中的 iertutil.dll 中的對應實現一致
 - 支持以指定的特權設置運程序
 - 注: 啟用全部特權, 禁用所有特權
 - 支持以指定的完整性級別運程序
 - 注: 系統、高、中、低
 - 支持以指定的進程優先級運程序
 - 注: 低、低於正常、正常、高於正常、高、實時
 - 支持以指定的窗口模式運程序
 - 注: 顯示窗口、隱藏窗口、最大化、最小化
 - 支持進程創建後並等待其運行結束
 - 支持以指定的當前目錄運程序
 - 支持在當前控制檯窗口下運程序
 - 支持快捷方式列表
 - 注: 你可以通過編輯 NSudo.json 的方式定製
 - 支持多種命令行風格
 - 多語言支持
 - 注: 簡體中文、繁體中文、英語、法語、德語、意大利語、西班牙語
 - 完整的高 DPI 支持
 - 注: 和 Windows 外殼 (conhost.exe) 的實現一樣完美, 在 Windows 10 Build 10240 及之後版本有完整的 Per-Monitor DPI-Aware 支持和在 Windows Vista 到 Windows 8.1 之間的版本有完整的 System DPI-Aware 支持
 - 完整的無障礙訪問支持
 - 注: 你可以在 Windows 講述人下順暢的使用 NSudo Launcher
 - 高性能
 - 注: 因為其實現不需要創建 Windows 服務和 Windows 服務進程

- 為開發者提供 C APIs 和 .Net Core 互操作支持
- NSudo 惡魔模式
 - 對於希望無視管理員權限的進程下的文件和註冊表訪問判斷的開發者而言是最優雅的 解決方案
 - 使用 Microsoft Detours 庫對 API 進行掛鉤以保證最大兼容
 - 其二進制僅依賴了 ntdll.dll 的以函數名導出的 API

系統要求

- 支持的系統版本: Windows NT 6.0 及之後版本
- 支持的處理器架構: x86, x86-64(AMD64), ARM64

項目原型

注意: NSudo 基於 raymai97 的超級命令提示符, 請參閱 [這裏](#) 以獲取更多關於 超級命令提示符的信息。

使用了 NSudo 的第三方項目

- MSMG ToolKit
- Sledgehammer (WUMT Wrapper Script)
- Dism++

第三方介紹

- HowToDoNinja: <https://howtodoninja.com/how-to/nsudo-run-programs-with-full-admin-privileges-windows/>
- MajorGeeks: <https://www.majorgeeks.com/files/details/nsudo.html>
- softpedia.com: <https://www.softpedia.com/get/Tweak/System-Tweak/NSudo.shtml>
- TrishTech.com: <https://www.trishtech.com/2018/11/nsudo-run-programs-with-full-privileges-in-windows/>
- Wilders Security Forums: <https://www.wilderssecurity.com/threads/396818>

成為 NSudo 的贊助者

Patreon: <https://www.patreon.com/MouriNaruto>

愛發電: <https://afdian.net/@MouriNaruto>

- 如果我達到了每個月 1000 美元的目標, 我將會每年更新兩個大版本。
- 如果我達到了每個月 2000 美元的目標, 我會為 NSudo 二進制添加 EV 代碼簽名證書。

感謝支持。

毛利

獲得支持

聯繫方式

- 郵箱: Mouri_Naruto@Outlook.com

社區

- [GitHub Issues](#)
- [My Digital Life](#)
- [QQ 羣](#)

使用方法

下載 NSudo

二進制

- [當前版本](#)
- [所有版本](#)
- [AppVeyor CI](#)

源代碼

- [GitHub](#)
- [碼雲](#)

NSudo Installer (非官方)

- [源代碼](#)
- [當前版本](#)

Chocolatey (非官方)

```
choco install nsudo
```

scoop (非官方)

```
scoop bucket add extras  
scoop install nsudo
```

第三方下載站

- [MajorGeeks](#)
- [softpedia.com](#)

NSudo Launcher

快速上手

請打開【CPU 架構】目錄, 然後雙擊 NSudo.exe。根據提示操作即可。例如, 如果你想在 你的 Intel 或 AMD 設備上使用 64 位 NSudo, 你首先需要打開的是 x64 目錄, 然後雙擊 NSudoG.exe。

命令行選項

格式: NSudoL [選項與參數] 命令列執行或常用任務名

選項:

-U:[選項] 以指定使用者選項建立處理程序。

可用選項：

T TrustedInstaller

S System

C 當前使用者

E 當前使用者（提權）

P 當前處理程序

D 當前處理程序（降權）

PS：這是一個必須被包含的參數。

-P:[選項] 以指定特殊權限選項建立處理程序。

可用選項：

E 啟用全部特殊權限

D 禁用所有特殊權限

PS：如果想以默認特殊權限選項建立處理程序，請不要包含「-P」參數。

-M:[選項] 以指定完整性選項建立處理程序。

可用選項：

S 系統

H 高

M 中

L 低

PS：如果想以默認完整性選項建立處理程序的話，請不要包含「-M」參數。

-Priority:[選項] 以指定處理程序優先級選項建立處理程序。

可用選項：

Idle 低

BelowNormal 低於正常

Normal 正常

AboveNormal 高於正常

High 高

RealTime 實時

PS：如果想以默認處理序優先權選項建立處理程序，請不要包含「-Priority」參數。

-ShowWindowMode:[選項] 以指定視窗模式選項建立處理程序。

可用選項：

Show 顯示視窗

Hide 隱藏視窗

Maximize 最大化

Minimize 最小化

PS：如果想以默認視窗模式選項建立處理程序的話，請不要包含「-ShowWindowMode」參數。

-Wait 令 NSudo Launcher 等待建立的處理程序結束後再退出。

PS：如果不想等待，請不要包含「-Wait」參數。

-CurrentDirectory:[目錄路徑] 設置處理程序的的當前目錄。

PS：如果你想用 NSudo Launcher 的當前目錄，請不要包含「-CurrentDirectory」參數。

-UseCurrentConsole 使用當前控制台視窗建立處理程序。

PS：如果你想在控制台視窗建立處理程序，請不要包含「-UseCurrentConsole」參數。

-Version 顯示 NSudo Launcher 版本資訊。

-? 顯示該內容。
-H 顯示該內容。
-Help 顯示該內容。

上下文清單管理請使用 https://github.com/Thdub/NSudo_Installer。

PS:

1. 所有的 NSudo Launcher 命令列參數不區分大小寫。
1. 可以在命令行參數中使用 "/" 或 "--" 代替 "-" 和使用 "=" 代替 ":"。例如
"/U:T" 和 "-U=T" 是等價的。
1. 為了保證最佳體驗，NSudoLC 不支持上下文清單。

例子:

以TrustedInstaller權限，啓用所有特殊權限，完整性默認執行命令提示字元
NSudoL -U:T -P:E cmd

例子: 以 TrustedInstaller 權限, 啓用所有特權, 完整性默認運行命令提示符

```
NSudo -U:T -P:E cmd
```

從 NSudo 5.0.1708.16 開始命令行支持嵌套引號, 例如:

```
NSudo -U:T cmd /c "dir "C:\Program Files" & pause"
```

常用列表

關於常用列表的自定義,可以使用記事本等工具編輯 NSudo.json。你可以照着示例的做法添加你的自定義項目:

```
{
  "ShortCutList_V2": {
    "命令提示符": "cmd",
    "PowerShell": "powershell",
    "PowerShell ISE": "powershell_ise",
    "Hosts編輯": "notepad %windir%\\System32\\Drivers\\etc\\hosts"
  }
}
```

NSudo 惡魔模式

NSudo 惡魔模式 (NSudo Devil Mode) 是爲想無視文件和註冊表操作權限檢查的開發者 量身定做的一個用起來還算優雅的解決方案。

其原理是使用開源的 Microsoft Detours 庫對 Windows NT 內核的文件和註冊表相關的 系統調用進行 Inline Hook 以傳入選項讓開發者基本不用修改自己的實現也能充分的利用 管理員權限所提供的特權, 這也使得開發者只需要把 NSudo 惡魔模式的動態鏈接庫加載入 自己的以管理員身份運行的應用進程的地址空間內即可啓用 NSudo 惡魔模式。

由於 NSudo 惡魔模式可以在大部分情況下替代類似 NSudo 的工具, 於是 NSudo 未來的 功能會變得更加專業向。畢竟作爲 NSudo 的作者的我可不希望 NSudo 就這麼輕易地被 替代掉。當然, 由於 NSudo 惡魔模式屬於

Dism++ 春哥附體的後續版本 (畢竟我也是 Dism++ 的其中一位開發者, 這麼說還是有依據的), 於是未來 Dism++ 的春哥附體的實現 會被替換成 NSudo 惡魔模式以幫助我更好的重構 Dism++ 的實現。當然, NSudo 也會支持 以惡魔模式運行應用。

NSudo 惡魔模式的起源、命名和意義

正如上文所說, NSudo 惡魔模式屬於 Dism++ 春哥附體的後續版本, 而且你也能在本文中 瞭解 NSudo 惡魔模式和 Dism++ 春哥附體的區別。

命名為 NSudo 惡魔模式的靈感來源是《入間同學入魔了》的被蛋爺改造過的擁有四個檔位 的入間手中的 "惡食戒指" 。

最開始聽到 MSMG Toolkit 的作者希望我能提供 NSudo 的 SDK 方便他進行二次開發時, 我試着做了基於 COM 接口的 NSudo Shared Library 即 NSudoAPI, 但是由於 NSudoAPI 暴露的細節太多, 如果不是對 Windows 安全特性足夠了解的開發者是很難駕馭的。於是 我覺得得換個方向, 於是就做了 NSudo 惡魔模式。

NSudo 惡魔模式掛鉤的 Windows NT 內核系統調用列表

名稱	起源
NtCreateKey	Dism++ 春哥附體
NtCreateKeyTransacted	NSudo 惡魔模式
NtOpenKey	Dism++ 春哥附體, 並增強了效果
NtOpenKeyTransacted	NSudo 惡魔模式
NtOpenKeyEx	Dism++ 春哥附體
NtOpenKeyTransactedEx	NSudo 惡魔模式
NtCreateFile	Dism++ 春哥附體
NtOpenFile	Dism++ 春哥附體

如何使用 NSudo 惡魔模式

調用 LoadLibrary 加載 NSudo 惡魔模式的動態鏈接庫以啓用 NSudo 惡魔模式, 調用 FreeLibrary 釋放 NSudo 惡魔模式的動態鏈接庫的 HMODULE 句柄即可禁用 NSudo 惡魔模式。

當然, 你的應用需要在管理員權限下運行, 相對於原本要求 SYSTEM 和 TrustedInstaller 權限的情況下其實好了不少。

注: 如果你有本事把 NSudo 惡魔模式遠程注入到以管理員或者更高權限的進程內 (譬如 7-Zip), 也能為該進程賦能 (無視文件和註冊表的權限)。

下面提供一個使用 C# 編寫的測試用例。(遍歷 C:\System Volume Information 目錄 的內容, 當啓用 NSudo 惡魔模式的情況下可以正常顯示, 禁用後會拋出文件夾拒絕訪問 的異常。)

```
using System;
using System.IO;
using System.Runtime.InteropServices;
```



```
namespace Demo
{
    class Program
    {
        [DllImport("kernel32.dll", CharSet = CharSet.Unicode)]
        static extern IntPtr LoadLibrary(string lpLibFileName);

        [DllImport("kernel32.dll", SetLastError = true)]
        [return: MarshalAs(UnmanagedType.Bool)]
        static extern bool FreeLibrary(IntPtr hLibModule);

        static void Main(string[] args)
        {
            IntPtr NSudoDevilModeModuleHandle = LoadLibrary(
                @"E:\Github\M2Team\NSudo\Output\Release\x64\NSudoDevilMode.dll");

            {
                DirectoryInfo Folder = new DirectoryInfo(
                    @"C:\System Volume Information");

                foreach (FileInfo File in Folder.GetFiles())
                {
                    Console.WriteLine(File.FullName);
                }
            }

            FreeLibrary(NSudoDevilModeModuleHandle);

            {
                DirectoryInfo Folder = new DirectoryInfo(
                    @"C:\System Volume Information");

                foreach (FileInfo File in Folder.GetFiles())
                {
                    Console.WriteLine(File.FullName);
                }
            }

            Console.ReadKey();
        }
    }
}
```

NSudo 惡魔模式的技術內幕

啓用 SeBackupPrivilege 和 SeRestorePrivilege 是前提條件, 但是你也需要在創建文件 或註冊表句柄的時候傳入對應的選項, 否則是不生效的。

首先說明一點, 那就是 Windows 內核當發現調用者上下文爲 SYSTEM 令牌的時候, 據 Microsoft 文檔描述是爲了提升 Windows 的性能會自動忽略掉大部分訪問檢查, 畢竟很多 Windows 系統關鍵組件運行在 SYSTEM 令牌上下文下面, 對於 Windows 用戶模式而言, SYSTEM 令牌是至高無上的, 所以訪問檢查沒必要做, 做了也提升不了安

全性反而降低了效率。所以這也是為什麼除了 SYSTEM 令牌上下文外的其他令牌都需要啟用相關特權 + 創建文件和註冊表句柄的 API 傳入對應選項才能忽略掉相關訪問檢查。

我用一個最簡單的例子來說明減少不需要的內核級訪問檢查的好處, 那就是在 Windows AppContainer 下運行的代碼, 由於會多出一個額外的內核級訪問檢查 (用 IDA 分析 ntoskrnl.exe, 然後用 F5 查看相關函數可以發現, 其實就是多出了一個分支和寥寥數行實現), 大概會比在 AppContainer 外運行會損失 15% 的性能 (這也可以說明越底層的實現越需要重視性能問題)。Windows AppContainer 是 Windows 8 開始提供的用戶模式沙盒, 主要用在商店應用和瀏覽器的沙盒上面。

Windows 的大部分內部使用了創建文件和註冊表句柄的 API 並沒有傳入對應的選項, 於是就出現了普通管理員下即使開啓了這兩個特權有些目錄照樣還是無法進行增刪查改。而 NSudo 惡魔模式通過 Inline Hook 對 Windows 用戶模式的系統調用層進行掛鉤以智能傳入相關選項, 這也是 NSudo 惡魔模式能在非 SYSTEM 的但擁有這兩個特權的令牌上下文下繞過文件和註冊表訪問判斷的緣由。

Windows 用戶模式系統調用層指的是 ntdll.dll 導出的前綴為 Nt 或 Zw 的 API, Windows 用戶模式下的 API 最終全會調用這部分以通過軟中斷陷阱門或者系統調用指令進入內核模式完成最終操作。

智能, 指的是隻有當前進程令牌上下文能夠啓用 SeBackupPrivilege 和 SeRestorePrivilege 的時候, 纔會傳入對應選項。畢竟如果這兩個特權沒有開啓的話, 傳入了相關選項是會返回錯誤的, 這也是為什麼 Windows 相關實現並沒有傳入的原因。

當然 NSudo 惡魔模式爲了對調用者更加透明和符合最小權限原則, 在初始化的時候首先會創建一份當前進程令牌的模擬令牌副本, 然後對該副本開啓這兩個特權。在 Hook 中, 會先備份當前線程上下文的令牌, 接着替換成模擬令牌副本 (或者用 Microsoft 文檔的稱法是模擬令牌上下文), 傳入相關選項調用原 API 後再恢復爲原來線程上下文的令牌。(實現細節請參考在 NSudo 代碼倉庫的 NSudo 惡魔模式的源代碼)

我說的有些囉嗦, 請見諒, 希望對你有幫助。

NSudo 共享庫

NSudoCreateProcess 函數

創建一個新進程和對應的主線程。

C/C++ 函數原型

```
EXTERN_C HRESULT WINAPI NSudoCreateProcess(  
    _In_ NSUDO_USER_MODE_TYPE UserModeType,  
    _In_ NSUDO_PRIVILEGES_MODE_TYPE PrivilegesModeType,  
    _In_ NSUDO_MANDATORY_LABEL_TYPE MandatoryLabelType,  
    _In_ NSUDO_PROCESS_PRIORITY_CLASS_TYPE ProcessPriorityClassType,  
    _In_ NSUDO_SHOW_WINDOW_MODE_TYPE ShowWindowModeType,  
    _In_ DWORD WaitInterval,  
    _In_ BOOL CreateNewConsole,  
    _In_ LPCWSTR CommandLine,  
    _In_opt_ LPCWSTR CurrentDirectory);
```

UserModeType 參數

一個類型爲 NSUDO_USER_MODE_TYPE 枚舉類型的值, 用以表示用戶類別。

```
typedef enum class _NSUDO_USER_MODE_TYPE
{
    DEFAULT,
    TRUSTED_INSTALLER,
    SYSTEM,
    CURRENT_USER,
    CURRENT_PROCESS,
    CURRENT_PROCESS_DROP_RIGHT
} NSUDO_USER_MODE_TYPE, *PNSUDO_USER_MODE_TYPE;
```

PrivilegesModeType 參數

一個類型爲 NSUDO_PRIVILEGES_MODE_TYPE 枚舉類型的值, 用以表示特權類型。

```
typedef enum class _NSUDO_PRIVILEGES_MODE_TYPE
{
    DEFAULT,
    ENABLE_ALL_PRIVILEGES,
    DISABLE_ALL_PRIVILEGES
} NSUDO_PRIVILEGES_MODE_TYPE, *PNSUDO_PRIVILEGES_MODE_TYPE;
```

MandatoryLabelType 參數

一個類型爲 NSUDO_MANDATORY_LABEL_TYPE 枚舉類型的值, 用以表示完整性標籤。

```
typedef enum class _NSUDO_MANDATORY_LABEL_TYPE
{
    UNTRUSTED,
    LOW,
    MEDIUM,
    MEDIUM_PLUS,
    HIGH,
    SYSTEM,
    PROTECTED_PROCESS,
} NSUDO_MANDATORY_LABEL_TYPE, *PNSUDO_MANDATORY_LABEL_TYPE;
```

ProcessPriorityClassType 參數

一個類型爲 NSUDO_PROCESS_PRIORITY_CLASS_TYPE 枚舉類型的值, 用以表示進程優先級。

```
typedef enum class _NSUDO_PROCESS_PRIORITY_CLASS_TYPE
{
    IDLE,
```

```
    BELOW_NORMAL,  
    NORMAL,  
    ABOVE_NORMAL,  
    HIGH,  
    REALTIME,  
} NSUDO_PROCESS_PRIORITY_CLASS_TYPE, *PNSUDO_PROCESS_PRIORITY_CLASS_TYPE;
```

ShowWindowModeType 參數

一個類型為 NSUDO_SHOW_WINDOW_MODE_TYPE 枚舉類型的值, 用以表示顯示窗口模式。

```
typedef enum class _NSUDO_SHOW_WINDOW_MODE_TYPE  
{  
    DEFAULT,  
    SHOW,  
    HIDE,  
    MAXIMIZE,  
    MINIMIZE,  
} NSUDO_SHOW_WINDOW_MODE_TYPE, *PNSUDO_SHOW_WINDOW_MODE_TYPE;
```

WaitInterval 參數

等待創建的進程的時間 (毫秒)。

CreateNewConsole 參數

如果這個參數設為 TRUE, 新進程則在新的控制檯窗口運行, 否則則直接在父進程對應的控制檯窗口運行 (默認設定)。

CommandLine 參數

要執行的命令行, 對應的字符串最大長度為 32k 即 32768 字符, 命令行所執行的應用的路徑被限制為 MAX_PATH 即 260 字符。

CurrentDirectory 參數

用來指定進程的當前目錄, 需要使用完整路徑。可以使用 UNC 路徑。如果該參數為 nullptr, 新進程將使用調用該函數的進程所使用的當前路徑。

返回值

HRESULT 類型。如果函數調用成功, 則返回值為 S_OK。

C# 接口

在你的項目中加載 M2.NSudo 程序集, 你就會知曉其用法。

代碼示例

```
using System;

namespace M2.NSudo.Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            NSudoInstance instance = new NSudoInstance();

            instance.CreateProcess(
                NSUDO_USER_MODE_TYPE.TRUSTED_INSTALLER,
                NSUDO_PRIVILEGES_MODE_TYPE.ENABLE_ALL_PRIVILEGES,
                NSUDO_MANDATORY_LABEL_TYPE.SYSTEM,
                NSUDO_PROCESS_PRIORITY_CLASS_TYPE.NORMAL,
                NSUDO_SHOW_WINDOW_MODE_TYPE.DEFAULT,
                0,
                true,
                "cmd",
                null);

            Console.ReadKey();
        }
    }
}
```

發行日志

NSudo 8.2

- 添加當前用戶 (提權) 模式支持 (由 xspeed1989 建議)
- 修復在 Windows 服務上下文使用 NSudo 遇到的阻礙性問題 (感謝 xspeed1989)
- 改善數處實現
- 修復在 NSudo 不支持的語言設置下界面顯示中文的問題 (感謝 rlesch)(#56)
- 升級 Mile.Cpp 到最新版本
 - 從 Mile.Project 升級到 Mile.Project.VisualStudio 最新版本
 - 將 Mile.Windows.TrustedLibraryLoader 與 Mile.Platform.Windows 升級到 Mile.Library 最新版本
 - 升級 VC-LTL 到最新版本
- 升級 Windows Template Library (WTL) 到 10.0.10320
- 移除 ARM32 支持
 - 理由: <https://mourinaruto.github.io/2021/05/17/Drop-ARM32-support-on-the-Windows-platform/>
- 改善數處文檔相關的實現
 - 改善項目網站體驗 (由 青春永不落幕 貢獻)
 - 改善碼雲體驗
 - 使用 GitHub Actions 部署項目網站
- Add German Language. (由 Steve 貢獻)
- 移除一些試驗性實現, 包括 NSudo Sweeper
- 添加日誌支持

NSudo 8.0 Update 1 (8.0.1)

- 更新意大利語翻譯 (由 garf02 貢獻)
- 使用 NuGet 包版本 VC-LTL 替代獨立版 VC-LTL (感謝 mingkuang)
- 創建 NSudo Sweeper 項目 (實驗性)
- 使用 Windows Template Library (WTL) 構建界面
- 改善項目網站實現 (由 青春永不落幕 貢獻)
- 整理整個項目
- 添加 Mile、Mile.Project、MINT 子項目使 NSudo 實現更加模塊化
- 改善 AppVeyor 和 GitHub Action 連續集成的支持 (感謝 mingkuang)
- 添加一鍵編譯腳本
- 升級 VC-LTL 到 4.1.1-Beta7
- 修復在 Windows 10 Build 21277 下崩潰的問題 (感謝 jgtoy)
- NSudo .NET 互操作庫採用 .NET 5.0 編譯
- 添加用戶手冊

NSudo 8.0

- 精簡二進制體積
 - 使用 FILE 而不是 std::ifstream
 - 使用新的編譯器選項
 - NSudo 合併入 NSudoG

- 優化圖標資源
- 使用 jsmn 而不是 JSON for Modern C++.
- NSudo 更名為 NSudo Launcher (NSudoLG.exe 和 NSudoLC.exe)
- 移除上下文支持, 因為 https://github.com/Thdub/NSudo_Installer 體驗更好
- 添加意大利語翻譯 (由 garf02 貢獻)
- 添加西班牙語翻譯 (由 Miguel Obando 貢獻)
- 遵循語義化版本規範
- 改善數個代碼和文檔方面的實現
- 添加為開發者設計的相關設施
 - 添加 NSudo Shared Library 且帶有 C/C++ 和 .Net 互操作支持
 - 添加 NSudo Devil Mode (NSudoDM).
 - 添加 Mouri Internal Library Essentials (Mile).
- 編譯本項目需要注意的變更事項
 - Visual Studio 已經升級到 2019
 - Windows 10 SDK 已經升級到最新
 - 改善對 AppVeyor CI 和 GitHub Actions CI 的支持 (感謝 Margen67)

NSudo 6.2.1812.31

- 添加法語翻譯 (由 Thomas Dubreuil 貢獻)
- 使用 JSON for Modern C++ 替代 RapidJSON 以符合 C++17 規範
- 改善圖形界面用戶體驗 (感謝 Lenny)
- 修復上下文菜單 Bug (感謝 Thomas Dubreuil 和 龍魂)
- 修復命令行解析器 Bug (感謝 wzzw)
- 添加繁體中文翻譯 (由 羅宇凡 貢獻)

NSudo 6.1.1811.18

- 把 NSudoC 與 NSudoG 合併入 NSudo
- 為 ARM 和 ARM64 的 Release 二進制編譯配置添加 VC-LTL 支持, 並移除 VC-LTL 4.0 之前版本的支持 (大力感謝 mingkuang)
- 添加以下新的命令行選項
 - CurrentDirectory (由 testtest322 建議)
 - Help
 - H
 - Priority (由 testtest322 建議)
 - ShowWindowMode (由 testtest322 建議)
 - UseCurrentConsole
 - Version
 - Wait (由 testtest322、wzzw 和 Domagoj Smolčić 建議)
- 移除一些未文檔化的命令行使用方式
- 改進數個實現
 - 重構命令行解析器
 - 引入新式創建進程前端
 - 使用 ATL 實現主窗口
 - 修復上下文菜單 Bug (感謝 Thomas Dubreuil)
- 更新許可的版權所有者
- 在文檔移除捐贈鏈接

NSudo 6.0.1804.5

- 修復在 Windows Vista 和 Server 2008 下崩潰的問題 (感謝 hydra79545)
- 與 M2-Team UWP 項目共享實現 (詳情請閱讀 "<https://github.com/Project-Nagisa/Nagisa/blob/master/Changelog.md>")
- 移除無用實現
- 改進 NSudoStartService 函數的實現
- 使用 RapidJSON 替代 JSON for Modern C++ 以減小二進制大小

NSudo 6.0.1802.2 v2

- 修復點擊運行按鈕只彈出命令提示符的問題。(感謝 AeonX)

NSudo 6.0.1802.2

- 修復多個 Bug 和改善多個實現
- 增加兩個獨立的可執行文件用於不同情況
 - NSudoC.exe
 - 純命令行版本, 子系統設置為 "控制檯"
 - 在控制檯下運行良好, 但是在非控制檯進程調用會出現黑色控制檯窗口
 - 爲了保證最佳體驗, NSudoC 不支持上下文菜單
 - NSudoG.exe
 - 純命令行版本, 子系統設置為 "Windows"
 - 可以靜默運行, 沒有黑色控制檯窗口
- NSudo 將通過 M2MessageDialog 而不是 TaskDialog 顯示信息
 - 理由
 - 因爲可以使用縱向滾動條, NSudo 可以在出錯時提供更加詳細的內容
 - 你可以複製對話框裏的內容
 - 支持 Windows 講述人, 於是可以使用 CapsLock+H 讓講述人讀取對話框內容
 - 比 TaskDialog 的字體更大
 - M2MessageDialog 特性
 - 在 Windows 10 Build 10240 或之後版本完全支持 Per-Monitor DPI Aware
 - 完全支持 Windows 講述人
 - 你可以使用縱向滾動條並且可以複製裏面的內容
 - 比 TaskDialog 的字體更大
 - 你可以按 Enter 鍵關閉對話框
 - 如果你想在你的項目使用 M2MessageDialog, 請從此處下載以下文件:
<https://github.com/M2Team/NSudo/tree/master/NSudoSDK>
 - M2DPIScaling.cpp
 - M2DPIScaling.h
 - M2MessageDialog.cpp
 - M2MessageDialog.h
 - M2MessageDialogResource.h
 - M2MessageDialogResource.rc
- 移除繁體中文和日語的翻譯, 因爲翻譯內容已經過時而且我不懂怎麼用
- 升級 JSON for Modern C++ 到 3.0.1
- 右鍵菜單
 - 增加多語言描述

- 為所有項目添加 "開啓全部特權" 選項
- 更新命令行幫助和文檔

NSudo 6.0.1801.19

- 修復 NSudoDuplicateSessionToken 函數定義 Bug (感謝 mingkuang)
- 修復在圖形界面下無法啓用全部特權的 Bug (感謝 abbodi1406)
- 修復沒有 VC-LTL 時 x86 和 x86-64(AMD64)的 Release 編譯配置未採用靜態編譯的 Bug

NSudo 5.3.1801.11

- 修復獲取 System 令牌函數的一個潛在 Bug (感謝 mingkuang)
- 對 x86 和 x86-64(AMD64)的 Release 編譯配置提供 VC-LTL 庫支持 (感謝 mingkuang)
 - PS: 把 NSudo 和 VC-LTL 一起使用可以減小 NSudo 二進制的體積
- 整理文檔

NSudo 5.2 (5.2.1709.8 - 5.2.1710.26)

- 整理代碼, 修復若干 Bugs
- 更新文檔, 增加英文自述
- 添加對 ARM 和 ARM64 平臺的支持 (感謝 fcharlie)
- 優化命令行解析
- 添加右鍵菜單支持
 - 使用 /Install 或 -Install 參數添加右鍵菜單 (命令行參數大小寫不敏感)
 - 使用 /Uninstall 或 -Uninstall 參數移除右鍵菜單 (命令行參數大小寫不敏感)

NSudo 5.1 (5.0.1708.9 - 5.1.1708.19)

- 修復批處理調用 NSudo 後批處理變量不生效的問題 (感謝 弎員外)
- 令 NSudo 在帶有命令行的狀態下也能自動請求管理員權限 (感謝 鳶一雨音)
- 更換新圖標, 順便解決在 Windows Vista 之前版本系統上不顯示 NSudo 圖標的問題 (PS: NSudo 最低要求依舊是 Windows Vista)
- 改進命令行解析 (感謝 鳶一雨音)
- 更新源代碼許可的版權 (對說辭進行了優化) 和更新感謝名單 (新增人士)

NSudo 5.0 (4.4.1705.28 - 5.0.1707.31)

- 使用新的獲取會話 ID 方法解決在 Server 系統的遠程桌面會話上使用 NSudo 運行應用可能無法顯示界面的問題 (感謝 sebus)
- 更新文檔和許可協議以符合實際情況
- 移除 VC-LTL (由 fcharlie 建議), 理由如下:
 - 雖然二進制大小增加 80KB, 但源代碼大小縮小 57.6MB
 - 源代碼大小縮小後, NSudo 的雲編譯速度大幅提升
 - 可以少屏蔽大量編譯警告
- 使用 NSudoSDK 項目代替 M2-SDK 項目
- 改進版本定義頭文件
- 編譯器啓用 SDL 檢查、調整編譯輸出目錄和更新 CI 編譯配置文件
- 調整並優化代碼(感謝 fcharlie 的建議)
- .gitignore 文件更新(由 fcharlie 實現)
- 完全使用 MSDN 文檔化 API 實現 NSudoAPI.h 以方便人們調用

- 與 Nagisa 項目共用 m2base.h
- 整理屏蔽的警告, 該版本 NSudo 屏蔽了以下警告實現 /W4 /WX 編譯
 - C4505 未引用的本地函數已移除(等級 4)
- NSudo 快捷列表文件格式從 ini 遷移到 json 並更新列表內容
- 進程創建時添加環境塊以改善兼容性
- 把 Windows XP 控件支持聲明和 Per-Monitor DPI Aware V1 支持移入清單文件
- 在清單文件添加兼容性 GUID 定義和 Per-Monitor DPI Aware V2 支持
- 修復當未在瀏覽窗口選擇文件的情況下命令行文本框出現""的問題

NSudo 4.4.1705.19

- 適配最新版 M2-SDK
- 適配最新版 VC-LTL
- 修改編譯選項
- 使用 git 子模塊機制 (由 myfreeer 實現)
- 配置 AppVeyor (由 myfreeer 提供靈感)
- 開始使用 AppVeyor 自動編譯
- 更新 M2-SDK 和 VC-LTL 子模塊
- 命令行解析從 main 函數拆分
- 修復升級 VC-LTL 後出現的編譯警告 (有空會 pull fix 到 VC-LTL)
- 版本號重新由自己而不是 CI 編譯服務控制
- 整理解決方案佈局

NSudo 4.3.1703.25

- 32 位版本取消對 SSE 和 SSE2 指令集的依賴 (為了保證完美的兼容性)
- 移除 NTIShell, NSudo.AppContainer, MiniFM 子項目
- NSudoSDK 完全被 M2-SDK 和 M2.NSudo.h 替代
- 關於界面佈局調整
- 子系統設置調整為 Windows 子系統 (為了不再彈出黑框)
- 優化代碼, 減少全局變量
- System 令牌副本創建函數移除會話 ID 參數 (因為現實情況只能使用當前會話 ID)
- 使用舊版應用調用方式 (即使用 cmd, 解決無法調用帶參數應用的問題)
- 優化在 UI 自動化工具 (例如講述人等讀屏軟件) 上的使用體驗
- "運行" 按鈕被設為默認按鈕以提升使用體驗
- 優化多語言資源以減小體積
- 修復 UI 標題欄沒有圖標的問題
- 為 UI 增加最小化按鈕
- 修復數個庫函數返回值 Bug
- 修復數個命令行解析 Bug
- 修復 UI 圖標的 DPI 縮放問題
- 開始使用 Visual Studio 2017 編譯
- 移除 NSudo-GUI 項目
- 代碼不再包含 M2-SDK 和 VC-LTL 的內容, 需要單獨從 github 克隆

NSudo 4.2

- 引入新 NSudoSDK API 並且對已有 NSudoSDK API 進行改善

- 優化代碼, 以減少 Windows API 調用次數
- 修復不帶任何參數情況下可能的奔潰問題
- 修復控制檯部分不能在非管理員權限顯示命令行幫助的問題
- 基於 ShellExecute 自建調用宿主, 以去除對 cmd.exe 的依賴
- 引入 NTShell (相當於 NSudo 1.0) 重製版, 作為 NSudoSDK 的一個示例
- 更改 MiniFM 圖標

NSudo 4.1

- 修復命令行使用 -U:D 導致程序奔潰的問題
- 更正命令行的 NSudoC 殘餘描述 (感謝 NotePad)
- 支持文件拖拽 (感謝 NotePad)

NSudo 4.0

- 重寫代碼, 提供 NSudoSDK, 使代碼容易使用在其他項目上
- 命令行下新增 "/" 前綴參數支持, 例如: NSudo /U:T /P:E cmd (感謝 th1r5bvn23)
- 支持默認參數, 即以 TrustedInstaller 令牌且開啓全部特權運行 (感謝 老九)
- 在默認快捷命令列表加入 host 編輯
- 增加 NSudo 和 MiniFM 的 Per-Monitor DPI Aware 支持
- 採用 VC-LTL 大幅度減小程序體積 (感謝 mingkuang)
- 更改圖標 (感謝 20011010wo)
- 精簡併優化主界面 (感謝 kCaRhC 卡殼, さくら)
- 使用 TaskDialog 替代 MessageBox
- 對關於界面進行調整, 並在關於界面加入命令行幫助
- 修復彈出文件不存在的問題
- 修復命令行解析的一個潛在 Bug
- 緩解 NSudo 圖形界面的空格問題 (瀏覽功能自動給命令行加引號)
- 消除在編譯時的警告 (/Wall 和 /WX 兩個參數同時使用)

NSudo 2016.1

- 修復 TrustedInstaller 下運程序界面不顯示問題 (感謝 abbodi1406)
- 修復命令行解析的漏洞和 UI 錯誤 (感謝 imadlatch)
- 整理代碼, 提升可讀性
- 當前目錄設為 NSudo 所在目錄 (未來會更加靈活)
- ShortCut 實現無限項目
- 新增簡易文件管理器小工具 (感謝 20011010wo)

NSudo 2016

- 支持多語言 (程序內含簡中, 繁中, 英文, 日文)
- 命令行處理重寫
- 實現代碼全部重構, 效率更高

NSudo 3.2 Fix1

- 優化程序邏輯; 減少無用代碼
- 命令行版和圖形版二合一

NSudo 3.2

- 修復無法使用帶有空格的路徑的問題
- NSudo 和 NSudoC 單文件化
- 增加 NSudo.bat 方便新手準確調用與電腦架構相符的 NSudo 版本
- NSudoSDK 增加靜態庫 (用 NSudo SDK 開發的工具可以實現單文件)
- 編譯平臺採用 Visual Studio 2015 + Windows 10 SDK

NSudo 3.1 Debug

- 修復 UI 的 ComboBox 不能輸入太長文字的問題
- 修復某些情況下不能使用的問題 (由於開發機 Windows10 的 Bug 而導致誤認為那種方式可行)
- 增加真正的令牌降權 (除了 cmd 會誤顯示管理員外; 其他的會將其看作普通用戶)
- 增加命令行版本
- 增加常用列表自定義功能

NSudo 3.0 R2

- 修復不能打開其他被系統關聯文件的 Bug
- SDK 的頭文件改進: 增加#pragma comment(lib,"NSudoAPI.lib")

NSudo 3.0

- 支持外部應用調用 (很抱歉讓一些人等太久)
- 增加了常用調用列表 (暫時不支持自定義; 未來 3.1 會加入)
- 加入了降權功能 (當然, 是完美降權到 UAC 未提權前。當然原理不是用獲取 explorer 令牌 和創建計劃任務)
- 支持對權限令牌的自定義
- 界面的完全重構 (相對於 2.x 來說)
- 代碼優化 (相對於 NSudo 3.0 M1 來說)
- 加入 NSudo SDK
- 原生 64 位版本
- 實現了調用外部程序無視 WOW64 重定向的方法 (NSudoCreateProcess)
- WinPE 支持 (雖然沒起多大作用)

NSudo 2.1

- 實現自動開啓所有權限 Token
- 對 cmd 的調用使用絕對路徑, 估計可以避免一些不必要的 Bug
- 優化程序代碼

NSudo 2.0

- 代碼全部使用 C++ Win32 SDK 重寫 (程序從 692KB 縮小到 92KB)
- 提供獲取權限的選項
- 提供命令行參數模式
- 更換了圖標

NSudo 1.2

- 未公開發布 (估計還是在修復 SessionID 問題)

NSudo 1.1

- 修復 SessionID 問題
- 32 位和 64 位版本合體 (根據架構確定運行那個架構的命令提示符, 採用 SysNative 目錄 (64 位 Vista 開始有的重定向) 調用 64 位 cmd)

NTIShell 1.0

- 根據 raymai97 的超級命令提示符製作的第一個版本