

## NSIS: Nullsoft Scriptable Install System Howto

; 16.04.2026.01.25.wib [Indonesia]

NSIS (Nullsoft Scriptable Install System) is a professional, open-source tool used to create Windows installers. It is known for being lightweight, fast, and highly customizable through its scripting language.

Here are the key highlights:

**Script-Based:** Everything is controlled via scripts, allowing you to handle complex installation logic, system registry changes, and hardware detection.

**Minimal Overhead:** It adds very little size to your application, making it ideal for web distribution.

**Highly Extensible:** Supports a wide range of plug-ins to add features like custom user interfaces, internet downloads, or specialized system tasks.

**Multi-Language Support:** Easily create installers that support dozens of different languages in a single file.

**Famous Users:** It's trusted by major companies like Google, Mozilla, and Dropbox for their software distribution.

Would you like to see a basic script example to get started or need help with a specific plugin?

[ Download ] <https://sourceforge.net/projects/nsis/>

basic script example

-----  
; --- Basic NSIS Script ---

; The name of the installer

Name "My Simple App"

; The file to generate

OutFile "SimpleSetup.exe"

; Default installation directory

InstallDir "\$PROGRAMFILES\MySimpleApp"

; Request admin privileges for Windows Vista and higher

RequestExecutionLevel admin

; --- Pages ---

Page directory ; Page to select the installation folder

Page instfiles ; Page where the actual installation happens

; --- Installation Section ---

Section "Main Application"

; Set the output path to the installation directory

SetOutPath \$INSTDIR

```
; Add files to be bundled (replace 'myapp.exe' with your file)
; File "myapp.exe"
; File "readme.txt"

; Create an uninstaller executable
WriteUninstaller "$INSTDIR\uninstall.exe"

SectionEnd

; --- Uninstaller Section ---
Section "Uninstall"
    ; Remove installed files
    Delete "$INSTDIR\uninstall.exe"
    ; Delete "$INSTDIR\myapp.exe"

    ; Remove the installation directory
    RMDir "$INSTDIR"

SectionEnd
```

-----

## How to Compile and Run

Install NSIS: Download and install the NSIS compiler.  
[ [https://nsis.sourceforge.io/Main\\_Page](https://nsis.sourceforge.io/Main_Page) ]

Prepare Files: Place any files you want to install (like myapp.exe) in the same folder as your .nsi script.

Compile: Right-click the mysetup.nsi file and select "Compile NSIS Script".

Test: A new file named SimpleSetup.exe will appear. Run it to see your installer in action.

## Common Commands Explained

OutFile: Sets the name of the final .exe installer you will distribute.

\$INSTDIR: A built-in variable representing the directory the user chooses for installation.

SetOutPath: Tells the installer where to extract subsequent files.

File: The command used to include your software files into the installer package.

Would you like to see how to add Start Menu shortcuts or a License Agreement page to this script?

-----

add Start Menu shortcuts or a License Agreement page to this script

-----

To add these features, it is recommended to use the Modern UI 2 (MUI2) library. This is the industry standard for NSIS because it provides a polished, professional look with minimal extra code.

## Enhanced NSIS Script Example

This updated script adds a License Agreement page and creates a Start Menu folder with shortcuts

-----

```
; --- Enhanced NSIS Script with MUI2 ---  
  
!include "MUI2.nsh"  
  
; General Settings  
Name "ApacheInstallerWizard"  
OutFile "ApacheInstallerWizard.exe"  
InstallDir "$PROGRAMFILES\ApacheInstallerWizard"  
RequestExecutionLevel admin  
  
; --- Modern UI Settings ---  
  
; Define the license file (Must be a .txt or .rtf file in the same folder)  
  
!insertmacro MUI_PAGE_WELCOME  
  
!insertmacro MUI_PAGE_LICENSE "license.txt" ; Displays your license  
agreement  
  
!insertmacro MUI_PAGE_DIRECTORY  
  
!insertmacro MUI_PAGE_INSTFILES  
  
!insertmacro MUI_PAGE_FINISH  
  
; Language Settings (Required for MUI)  
  
!insertmacro MUI_LANGUAGE "English"  
  
; --- Installation Section ---  
  
Section "Main Application"  
    SetOutPath $INSTDIR
```

```
; Add your application files here

; File "myapp.exe"

; Create Start Menu Shortcuts

; $SMPROGRAMS points to the Start Menu "Programs" folder
CreateDirectory "$SMPROGRAMS\ApacheInstallerWizard"

CreateShortCut
"$SMPROGRAMS\ApacheInstallerWizard\ApacheInstallerWizard.lnk"
"$INSTDIR\ApacheInstallerWizard.exe"

CreateShortCut
"$SMPROGRAMS\ApacheInstallerWizard\ApacheInstallerWizardUninstall.lnk"
"$INSTDIR\ApacheInstallerWizarduninstall.exe"

; Create a Desktop Shortcut (Optional)

CreateShortCut "$DESKTOP\ApacheInstallerWizard.lnk"
"$INSTDIR\ApacheInstallerWizard.exe"

WriteUninstaller "$INSTDIR\ApacheInstallerWizarduninstall.exe"

SectionEnd

; --- Uninstaller Section ---

Section "Uninstall"

; Delete Files

Delete "$INSTDIR\ApacheInstallerWizard.exe"

Delete "$INSTDIR\ApacheInstallerWizarduninstall.exe"

; Remove Shortcuts and Folders
```

```
Delete
"$SMPROGRAMS\ApacheInstallerWizard\ApacheInstallerWizard.lnk"

Delete
"$SMPROGRAMS\ApacheInstallerWizard\ApacheInstallerWizardUninstall.lnk"

RMDir "$SMPROGRAMS\ApacheInstallerWizard"

Delete "$DESKTOP\ApacheInstallerWizard.lnk"

; Remove Installation Directory

RMDir "$INSTDIR"

SectionEnd
```

-----  
How it works:

`!insertmacro MUI_PAGE_LICENSE`: Automatically creates a page that displays your `license.txt` file and requires the user to click "I Agree" to proceed.

`CreateDirectory`: Creates a new folder in the Start Menu to group your shortcuts.

`CreateShortcut`:

Takes the destination path (the `.lnk` file) as the first argument.

Takes the source target (the `.exe` file) as the second argument.

`$SMPROGRAMS` and `$DESKTOP` are standard NSIS constants for these system folders.

**Uninstallation**: It is critical to delete the shortcuts during uninstallation so they don't remain as "broken" links on the user's computer.

Would you like to know how to add a custom icon to these shortcuts or include optional components for the user to select?

-----

## 2. Consistency in Uninstaller Naming

In your Section "Uninstall", the Delete command must match the uninstaller name exactly. You used:

```
WriteUninstaller "$INSTDIR\ApacheInstallerWizarduninstall.exe"
```

If you want to be safe and clean up everything in the folder, you can use Delete "\$INSTDIR\\*.\*" (use with caution) or stick to naming it simply uninstall.exe to avoid long filenames.

## 3. Add an Icon

To give your installer a professional look, add this line in the General Settings area (before the pages):

-----

```
!define MUI_ICON "your_icon.ico"
```

```
!define MUI_UNICON "your_icon.ico" ; Icon for the uninstaller
```

-----

Ready to compile?

Make sure ApacheInstallerWizard.exe and license.txt are in the same folder as your .nsi script, then right-click and compile.

Would you like to know how to automatically launch the app after the installation finishes on the "Finish" page?

-----

To add the auto-launch feature in English, use the MUI\_FINISHPAGE\_RUN define.

Add these lines specifically before the !insertmacro MUI\_PAGE\_FINISH line in your script:

-----

; --- Modern UI Settings ---

```
!insertmacro MUI_PAGE_WELCOME
```

```
!insertmacro MUI_PAGE_LICENSE "license.txt"
```

```
!insertmacro MUI_PAGE_DIRECTORY
```

```
!insertmacro MUI_PAGE_INSTFILES
```

; --- Add these lines here ---

```
!define MUI_FINISHPAGE_RUN  
"$INSTDIR\ApacheInstallerWizard.exe"
```

```
!define MUI_FINISHPAGE_RUN_TEXT "Launch ApacheInstallerWizard  
now"
```

```
!insertmacro MUI_PAGE_FINISH
```

-----

Key Details:

**Placement:** The !define must come before the !insertmacro for the Finish page, or it won't take effect.

**Checkbox:** This will automatically create a checkbox on the final screen that says "Launch ApacheInstallerWizard now."

**Execution:** When the user clicks "Finish" with the box checked, the installer will launch your application.

One Small Correction for your script:

In your previous script, you used ApacheInstallerWizarduninstall.exe as the uninstaller name. Usually, Windows expects a simple uninstall.exe. If you keep your custom name, ensure your shortcuts and delete commands match it exactly to avoid "file not found" errors.

Would you like to add an "Open Readme" checkbox to the finish page as well?

-----  
Open Readme.html as html5  
-----

To open a local Readme.html file using the default web browser (as HTML5) instead of a simple text editor, you should use the ExecShell command.

The standard MUI\_FINISHPAGE\_SHOWREADME command often defaults to opening files with their associated system application, which might be a code editor for .html files. By using a custom function with ExecShell, you ensure the Windows shell handles the file, which usually triggers the default browser.

Add these lines to your script:

1. Update the Modern UI Settings

Define a custom function for the "Show Readme" checkbox and place it before the finish page macro

-----  
; --- Modern UI Settings ---

```
!insertmacro MUI_PAGE_WELCOME
```

```
!insertmacro MUI_PAGE_LICENSE "license.txt"
```

```
!insertmacro MUI_PAGE_DIRECTORY
```

```
!insertmacro MUI_PAGE_INSTFILES
```

```
; Setup Readme Checkbox to call a function
```

```
!define MUI_FINISHPAGE_SHOWREADME ""
```

```
!define MUI_FINISHPAGE_SHOWREADME_TEXT "Open Readme  
Guide"
```

```
!define MUI_FINISHPAGE_SHOWREADME_FUNCTION
"OpenReadme"
```

```
!insertmacro MUI_PAGE_FINISH
```

---

## 2. Define the Function

Add this function at the bottom of your script (where other functions are located).

---

Why this works:

**MUI\_FINISHPAGE\_SHOWREADME\_FUNCTION:** This tells NSIS not to try and open a file directly, but to run your custom code when the box is checked.

**ExecShell "open":** This mimics double-clicking the file in Windows. If the user's default browser is set to handle .html files, it will open there.

**File Path:** Ensure Readme.html is actually copied to the \$INSTDIR during the installation section using the File command.

---

This is the complete, production-ready script for ApacheInstallerWizard. It incorporates the Modern UI 2 (MUI2) interface,

a license agreement, Start Menu and Desktop shortcuts, an auto-launch checkbox, and an HTML5 Readme function.

---

; --- ApacheInstallerWizard Full Script ---

; 1. General Settings

```
!define PRODUCT_NAME "ApacheInstallerWizard"
```

```
!define PRODUCT_VERSION "1.0"
```

```
!define PRODUCT_PUBLISHER "piasDraco[ DANA ]"
```

```
!define PRODUCT_WEB_SITE
```

```
"http://piasdraco.iceiy.com/apm_installer_wizard"
```

```
Name "${PRODUCT_NAME}"
```

```
OutFile "ApacheInstallerWizardSetup.exe"
```

```
InstallDir "$PROGRAMFILES\ApacheInstallerWizard"
```

```
RequestExecutionLevel admin ; Required for writing to Program Files
```

; 2. Include Modern UI 2

```
!include "MUI2.nsh"
```

; 3. Interface Settings

```
!define MUI_ABORTWARNING
```

```
!define MUI_ICON "${NSISDIR}\Contrib\Graphics\Icons\modern-  
install.ico"
```

```
!define MUI_UNICON "${NSISDIR}\Contrib\Graphics\Icons\modern-  
uninstall.ico"
```

; 4. Finish Page Settings (Must be defined BEFORE the Finish Page  
macro)

```
!define MUI_FINISHPAGE_RUN
```

```
"$INSTDIR\ApacheInstallerWizard.exe"
```

```
!define MUI_FINISHPAGE_RUN_TEXT "Launch ApacheInstallerWizard  
now"
```

```
; Setup HTML5 Readme function
```

```
!define MUI_FINISHPAGE_SHOWREADME ""
```

```
!define MUI_FINISHPAGE_SHOWREADME_TEXT "Open Readme  
Guide"
```

```
!define MUI_FINISHPAGE_SHOWREADME_FUNCTION  
"OpenReadmeHTML"
```

```
; 5. Installer Pages
```

```
!insertmacro MUI_PAGE_WELCOME
```

```
!insertmacro MUI_PAGE_LICENSE "license.txt" ; Ensure this file exists  
in the script folder
```

```
!insertmacro MUI_PAGE_DIRECTORY
```

```
!insertmacro MUI_PAGE_INSTFILES
```

```
!insertmacro MUI_PAGE_FINISH
```

```
; 6. Uninstaller Pages
```

```
!insertmacro MUI_UNPAGE_CONFIRM
```

```
!insertmacro MUI_UNPAGE_INSTFILES
```

```
; 7. Languages
```

```
!insertmacro MUI_LANGUAGE "English"
```

```
; 8. Installation Section
```

```
Section "Main Application" SEC01
```

```
SetOutPath "$INSTDIR"

SetOverwrite on

; Add your application files here
File "ApacheInstallerWizard.exe"
File "Readme.html"

; Create Start Menu Folder and Shortcuts
CreateDirectory "$SMPROGRAMS\ApacheInstallerWizard"

CreateShortCut
"$SMPROGRAMS\ApacheInstallerWizard\ApacheInstallerWizard.lnk"
"$INSTDIR\ApacheInstallerWizard.exe"

CreateShortCut
"$SMPROGRAMS\ApacheInstallerWizard\Uninstall.lnk"
"$INSTDIR\uninstall.exe"

; Create Desktop Shortcut

CreateShortCut "$DESKTOP\ApacheInstallerWizard.lnk"
"$INSTDIR\ApacheInstallerWizard.exe"

; Store installation folder in registry for uninstaller

WriteRegStr HKLM "Software\ApacheInstallerWizard" "Install_Dir"
"$INSTDIR"

; Create Uninstaller

WriteUninstaller "$INSTDIR\uninstall.exe"

SectionEnd
```

; 9. HTML5 Readme Function

Function OpenReadmeHTML

; Uses ShellExecute to open the HTML file in the default browser

ExecShell "open" "\$INSTDIR\Readme.html"

FunctionEnd

; 10. Uninstaller Section

Section "Uninstall"

; Remove Files

Delete "\$INSTDIR\ApacheInstallerWizard.exe"

Delete "\$INSTDIR\Readme.html"

Delete "\$INSTDIR\uninstall.exe"

; Remove Shortcuts

Delete

"\$SMPROGRAMS\ApacheInstallerWizard\ApacheInstallerWizard.lnk"

Delete "\$SMPROGRAMS\ApacheInstallerWizard\Uninstall.lnk"

RMDir "\$SMPROGRAMS\ApacheInstallerWizard"

Delete "\$DESKTOP\ApacheInstallerWizard.lnk"

; Remove Directory

RMDir "\$INSTDIR"

DeleteRegKey HKLM "Software\ApacheInstallerWizard"

SectionEnd

-----

Required Files in your Script Folder:

Before compiling this script with the NSIS compiler, ensure the following files are in the same folder as the .nsi file:

ApacheInstallerWizard.exe: Your main application file.

license.txt: Your license agreement text.

Readme.html: Your help file.

Quick Tip: If you want the "Launch" and "Readme" boxes unchecked by default, add `!define MUI_FINISHPAGE_RUN_NOTCHECKED` and `!define MUI_FINISHPAGE_SHOWREADME_NOTCHECKED` above the Finish page macro.

Would you like to know how to add custom branding text to the bottom of the installer window?

-----

To add custom branding text, use the `BrandingText` command. In the Modern UI (MUI2),

this text appears at the bottom-left corner of the installer window, replacing the default "Nullsoft Install System v3.11 [piasDraco Soft[DANA] ]" text.

Add this line in the General Settings section (near the top of your script):

-----

; --- General Settings ---

Name "\${PRODUCT\_NAME}"

OutFile "ApacheInstallerWizardSetup.exe"

; Add your custom branding text here

```
BrandingText "Apache Installer Wizard - Copyleft © 2017 [piasDraco  
Soft [DANA] ]"
```

```
InstallDir "$PROGRAMFILES\ApacheInstallerWizard"
```

---

### Advanced Branding Options

If you want to make the branding text more dynamic or stylish, you can use these additional settings:

#### Dynamic Versioning:

Use your defined variables so you only have to update the version in one place.

---

```
BrandingText "${PRODUCT_NAME} v${PRODUCT_VERSION}"
```

---

#### MUI Sidebar Branding (Optional):

If you want text to appear on the left sidebar (only works on some MUI skins), you can use:

---

```
!define MUI_BRANDINGTEXT "Powered by Apache PHP MariaDB"
```

---

#### Changing the Font/Size (Optional):

If you want the branding text to look different, add this:

---

```
SetFont "Comic Sans MS" 8
```

---

Important Details:

**System Wide:** This command changes the font for the entire installer interface (buttons, labels, and directory paths).

**Availability:** Since "Comic Sans MS" is a standard Windows font, it will work on almost all target machines. If the font is missing, Windows will automatically use the default system font as a fallback.

**Modern UI (MUI2) Note:** If you want to change the font of the header text (the big title at the top) specifically, you would need to use `!define MUI_HEADERFONT`.

Would you like to also change the color of the text to match the new font style?

---

To change the Header Text font (the large title at the top of the installer pages), you need to define the header font before including the Modern UI macros.

Add these lines in the Modern UI Settings section of your script:

---

```
; --- Modern UI Settings ---
```

```
!define MUI_ABORTWARNING
```

```
; 1. Define the Font Name
```

```
!define MUI_HEADERFONT "Comic Sans MS"
```

```
; 2. Define the Font Size (usually 12-14 is good for headers)
```

```
!define MUI_HEADERFONT_SIZE 14
```

```
; (Optional) Make the header bold
!define MUI_HEADERFONT_BOLD
```

```
; Pages...
```

```
!insertmacro MUI_PAGE_WELCOME
```

-----

Summary of Font Controls:

SetFont "Comic Sans MS" 8: This changes the font for the entire installer (buttons, descriptions, path names).

MUI\_HEADERFONT: This specifically overrides the font for the top title bar on each page.

Pro Tip: Custom Header Color

If you want to change the color of that header text to make it stand out even more, you can add this line:

-----

To set a Dark Red background with White text for the header, Welcome, and Finish pages in NSIS (MUI2), you need to define specific color macros before including the MUI2.nsh header.

Color Settings

Add these definitions in your Modern UI Settings section. Note that NSIS colors use hexadecimal format:

-----

```
; --- Modern UI Settings ---
```

```
!define MUI_ABORTWARNING
```

```
; Set Header and Wizard Background to Dark Red (Hex: 8B0000)
```

```
!define MUI_BGCOLOR 8B0000
```

```
; Set Header and Wizard Text to White (Hex: FFFFFFFF)
```

```
!define MUI_TEXTCOLOR FFFFFFFF
```

```
; (Optional) Specifically for the Header Text Color
```

```
!define MUI_HEADER_TEXT_COLOR FFFFFFFF
```

```
; Ensure labels on the header have a transparent background to see  
the red
```

```
!define MUI_HEADER_TRANSPARENT_TEXT
```

```
!include "MUI2.nsh"
```

-----

## Important Details

**MUI\_BGCOLOR:** This controls the background for the header area of all standard pages and the entire background for the Welcome and Finish pages.

**MUI\_TEXTCOLOR:** This sets the main text color for those same areas to white.

**MUI\_HEADER\_TRANSPARENT\_TEXT:** Without this, the small labels in the header might keep a white rectangular background. This forces them to be transparent so your dark red background shows through.

**Hex Format:** Do not use the 0x prefix or # in these specific MUI defines; just use the 6-character hex code.

## Specific Page Text Colors

If you find that text in specific boxes (like the License text box) is still black on white, you can customize those individually as well:

```
!define MUI_LICENSEPAGE_BGCOLOR 8B0000
```

```
!define MUI_DIRECTORYPAGE_BGCOLOR 8B0000
```

Would you like to also change the Progress Bar color to white to match your text theme?

-----

To change the Progress Bar color to green, you need to use a specific instruction within the `.onInit` function or directly in the section. In NSIS, the progress bar is a standard Windows control, and its color is typically managed via the `SendMessage` command.

Add this code to your script:

### 1. Add the Green Progress Bar Logic

Add this function at the bottom of your script. It targets the progress bar (control ID 1004) and changes its color to green.

-----

```
; --- Progress Bar Color Logic ---
```

```
Function .onGUIInit
```

```
    ; Find the progress bar and set color to Green (Hex: 00FF00)
```

```
    ; The format for SendMessage here is 0x0409  
(PBM_SETBARCOLOR)
```

```
    GetDlgItem $0 $HWNDPARENT 1004
```

```
    SendMessage $0 0x0409 0 0x00FF00 ; 0x00FF00 is Green in Blue-  
Green-Red format
```

```
FunctionEnd
```

---

## 2. Update the Modern UI Settings (Dark Red Theme)

Since you requested a Dark Red background with White text earlier, here is how the top of your script should look to keep everything consistent:

---

```
; --- Modern UI Settings ---
```

```
!define MUI_ABORTWARNING
```

```
; Colors for Header/Welcome/Finish
```

```
!define MUI_BGCOLOR 8B0000 ; Dark Red Background
```

```
!define MUI_TEXTCOLOR FFFFFFFF ; White Text
```

```
!define MUI_HEADER_TEXT_COLOR FFFFFFFF
```

```
!define MUI_HEADER_TRANSPARENT_TEXT
```

```
; Progress Bar Style (Optional: makes it look more modern)
```

```
!define MUI_INSTFILES_PAGE_COLORS "FFFFFF 8B0000" ; White text  
on Dark Red background for the log
```

---

A Quick Note on Hex Colors:

NSIS often uses BGR (Blue-Green-Red) instead of the standard RGB for some internal commands. However, for 0x00FF00, it remains Green regardless.

Would you like to add a custom background image to the Welcome/Finish pages to make the Dark Red theme look even more professional?

-----  
o make the ApacheInstallerWizard look truly professional with your Dark Red, White, and Green theme, we can add branding metadata, custom images, and consistent UI styling.

Here is the complete, high-end script including all the features we discussed:

Professional ApacheInstallerWizard.nsi

-----

; --- 1. General Settings & Metadata ---

```
!define PRODUCT_NAME "ApacheInstallerWizard"
```

```
!define PRODUCT_VERSION "1.0.0"
```

```
!define PRODUCT_PUBLISHER "piasDraco[ DANA ]"
```

```
!define PRODUCT_WEB_SITE
```

```
"http://piasdraco.iceiy.com/apm_installer_wizard"
```

```
Name "${PRODUCT_NAME}"
```

```
OutFile "ApacheInstallerWizard_Setup.exe"
```

```
InstallDir "$PROGRAMFILES\ApacheInstallerWizard"
```

```
RequestExecutionLevel admin
```

```
; File Details for Windows Properties
```

```
VIProductVersion "1.0.0.0"
```

```
VIAddVersionKey "ProductName" "${PRODUCT_NAME}"
```

```
VIAddVersionKey "FileDescription" "Professional Apache PHP  
MariaDB Setup Wizard"
```

```
VIAddVersionKey "LegalCopyright" "Copyleft © 2017  
${PRODUCT_PUBLISHER}"
```

; --- 2. Font & Branding Settings ---

SetFont "Comic Sans MS" 8

BrandingText "Apache Installer Wizard - Apache PHP MariaDB One  
Bundle Windows Edition"

; --- 3. Modern UI 2 Settings ---

!include "MUI2.nsh"

!define MUI\_ABORTWARNING

; Color Theme: Dark Red Background, White Text

!define MUI\_BGCOLOR 8B0000

!define MUI\_TEXTCOLOR FFFFFFFF

!define MUI\_HEADER\_TEXT\_COLOR FFFFFFFF

!define MUI\_HEADER\_TRANSPARENT\_TEXT

; Header Font

!define MUI\_HEADERFONT "Comic Sans MS"

!define MUI\_HEADERFONT\_SIZE 14

!define MUI\_HEADERFONT\_BOLD

; Icons (Replace with your .ico paths)

; !define MUI\_ICON "app\_icon.ico"

; !define MUI\_UNICON "app\_icon.ico"

; Finish Page Options

```
!define MUI_FINISHPAGE_RUN  
"$INSTDIR\ApacheInstallerWizard.exe"
```

```
!define MUI_FINISHPAGE_RUN_TEXT "Launch Wizard Now"
```

```
!define MUI_FINISHPAGE_SHOWREADME ""
```

```
!define MUI_FINISHPAGE_SHOWREADME_TEXT "Open HTML5  
Guide"
```

```
!define MUI_FINISHPAGE_SHOWREADME_FUNCTION  
"OpenReadmeHTML"
```

; --- 4. Pages ---

```
!insertmacro MUI_PAGE_WELCOME
```

```
!insertmacro MUI_PAGE_LICENSE "license.txt"
```

```
!insertmacro MUI_PAGE_DIRECTORY
```

```
!insertmacro MUI_PAGE_INSTFILES
```

```
!insertmacro MUI_PAGE_FINISH
```

```
!insertmacro MUI_UNPAGE_CONFIRM
```

```
!insertmacro MUI_UNPAGE_INSTFILES
```

```
!insertmacro MUI_LANGUAGE "English"
```

; --- 5. Installation Logic ---

```
Section "Main Application" SEC01
```

```
    SetOutPath "$INSTDIR"
```

```
; Add your files
File "ApacheInstallerWizard.exe"
File "Readme.html"
File "license.txt"

; Shortcuts

CreateDirectory "$SMPROGRAMS\ApacheInstallerWizard"

CreateShortCut
"$SMPROGRAMS\ApacheInstallerWizard\ApacheInstallerWizard.lnk"
"$INSTDIR\ApacheInstallerWizard.exe"

CreateShortCut "$DESKTOP\ApacheInstallerWizard.lnk"
"$INSTDIR\ApacheInstallerWizard.exe"

WriteUninstaller "$INSTDIR\uninstall.exe"

SectionEnd

; --- 6. Progress Bar Color (Green) ---

Function .onGUIInit
    ; Set Progress Bar to Green (BGR format: 00FF00)
    GetDlgItem $0 $HWNDPARENT 1004
    SendMessage $0 0x0409 0 0x00FF00
FunctionEnd

; --- 7. Functions ---

Function OpenReadmeHTML
    ExecShell "open" "$INSTDIR\Readme.html"
```

FunctionEnd

; --- 8. Uninstaller ---

Section "Uninstall"

Delete "\$INSTDIR\ApacheInstallerWizard.exe"

Delete "\$INSTDIR\Readme.html"

Delete "\$INSTDIR\license.txt"

Delete "\$INSTDIR\uninstall.exe"

RMDir /r "\$SMPROGRAMS\ApacheInstallerWizard"

Delete "\$DESKTOP\ApacheInstallerWizard.lnk"

RMDir "\$INSTDIR"

SectionEnd

-----  
To finalize your ApacheInstallerWizard as a high-end professional installer,

I have integrated the logo.ico and sidebar.bmp files into the script. I also ensured the Visual Continuity and

UI Feedback (Green Progress Bar) are perfectly aligned.

-----  
Professional ApacheInstallerWizard Script

-----  
; --- 1. General Settings & Metadata ---

!define PRODUCT\_NAME "ApacheInstallerWizard"

!define PRODUCT\_VERSION "1.0.0"

```
!define PRODUCT_PUBLISHER "piasDraco[ DANA ]"

!define PRODUCT_WEB_SITE
"http://piasdraco.iceiy.com/apm_installer_wizard"

Name "${PRODUCT_NAME}"

OutFile "ApacheInstallerWizard_Setup.exe"

InstallDir "$PROGRAMFILES\ApacheInstallerWizard"

RequestExecutionLevel admin

; Metadata for Windows File Properties

VIProductVersion "1.0.0.0"

VIAddVersionKey "ProductName" "${PRODUCT_NAME}"

VIAddVersionKey "FileDescription" "Professional Apache PHP
MariaDB Setup Wizard"

VIAddVersionKey "CompanyName" "${PRODUCT_PUBLISHER}"

VIAddVersionKey "LegalCopyright" "Copyleft © 2017
${PRODUCT_PUBLISHER}"

; --- 2. Branding & Font ---

SetFont "Comic Sans MS" 8

BrandingText "Apache Installer Wizard - Professional Edition"

; --- 3. Modern UI 2 Settings ---

!include "MUI2.nsh"

!define MUI_ABORTWARNING
```

; Visual Continuity: Dark Red & White Theme

!define MUI\_BGCOLOR 8B0000

!define MUI\_TEXTCOLOR FFFFFFFF

!define MUI\_HEADER\_TEXT\_COLOR FFFFFFFF

!define MUI\_HEADER\_TRANSPARENT\_TEXT

; Header Font

!define MUI\_HEADERFONT "Comic Sans MS"

!define MUI\_HEADERFONT\_SIZE 14

!define MUI\_HEADERFONT\_BOLD

; Visual Assets

!define MUI\_ICON "logo.ico" ; Installer Icon

!define MUI\_UNICON "logo.ico" ; Uninstaller Icon

!define MUI\_WELCOMEFINISHPAGE\_BITMAP "sidebar.bmp" ;  
Sidebar image (164x314)

!define MUI\_HEADERIMAGE

!define MUI\_HEADERIMAGE\_BITMAP "logo.ico" ; Shows logo in top  
right header

!define MUI\_HEADERIMAGE\_RIGHT

; UI Feedback: Auto-Launch & HTML5 Readme

!define MUI\_FINISHPAGE\_RUN  
"\$INSTDIR\ApacheInstallerWizard.exe"

!define MUI\_FINISHPAGE\_RUN\_TEXT "Launch ApacheInstallerWizard  
now"

```
!define MUI_FINISHPAGE_SHOWREADME ""

!define MUI_FINISHPAGE_SHOWREADME_TEXT "Open HTML5 User
Guide"

!define MUI_FINISHPAGE_SHOWREADME_FUNCTION
"OpenReadmeHTML"

; --- 4. Pages ---

!insertmacro MUI_PAGE_WELCOME

!insertmacro MUI_PAGE_LICENSE "license.txt"

!insertmacro MUI_PAGE_DIRECTORY

!insertmacro MUI_PAGE_INSTFILES

!insertmacro MUI_PAGE_FINISH

!insertmacro MUI_UNPAGE_CONFIRM

!insertmacro MUI_UNPAGE_INSTFILES

!insertmacro MUI_LANGUAGE "English"

; --- 5. Installation Section ---

Section "Main Application" SEC01
    SetOutPath "$INSTDIR"

; Bundled Files

File "ApacheInstallerWizard.exe"

File "Readme.html"

File "license.txt"
```

```
File "logo.ico"

; Clean Uninstallation: Shortcuts

CreateDirectory "$SMPROGRAMS\ApacheInstallerWizard"

CreateShortCut
"$SMPROGRAMS\ApacheInstallerWizard\ApacheInstallerWizard.lnk"
"$INSTDIR\ApacheInstallerWizard.exe" "" "$INSTDIR\logo.ico"

CreateShortCut "$DESKTOP\ApacheInstallerWizard.lnk"
"$INSTDIR\ApacheInstallerWizard.exe" "" "$INSTDIR\logo.ico"

CreateShortCut
"$SMPROGRAMS\ApacheInstallerWizard\Uninstall.lnk"
"$INSTDIR\uninstall.exe"

WriteUninstaller "$INSTDIR\uninstall.exe"

SectionEnd

; --- 6. UI Feedback: Green Progress Bar ---

Function .onGUIInit

    ; Set Progress Bar to Green (BGR: 00FF00)

    GetDlgItem $0 $HWNDPARENT 1004

    SendMessage $0 0x0409 0 0x00FF00

FunctionEnd

; --- 7. Functions ---

Function OpenReadmeHTML

    ExecShell "open" "$INSTDIR\Readme.html"

FunctionEnd
```

; --- 8. Uninstaller Section (Clean Uninstallation) ---

Section "Uninstall"

; Remove Files

Delete "\$INSTDIR\ApacheInstallerWizard.exe"

Delete "\$INSTDIR\Readme.html"

Delete "\$INSTDIR\license.txt"

Delete "\$INSTDIR\logo.ico"

Delete "\$INSTDIR\uninstall.exe"

; Remove Shortcuts & Folders

RMDir /r "\$SMPROGRAMS\ApacheInstallerWizard"

Delete "\$DESKTOP\ApacheInstallerWizard.lnk"

; Remove Directory

RMDir "\$INSTDIR"

SectionEnd

-----

To restrict the installer so it only installs on Windows 10, you need to include the WinVer.nsh library. This allows the installer to check the operating system version before the installation starts.

Add the following logic to your script:

1. Include the Windows Version Library

Add this at the very top with your other includes:

-----

```
!include "WinVer.nsh"
```

---

## 2. Add the Version Check Function

Add the `.onInit` function (or update it if you already have one). This function runs

the moment the user double-clicks the setup file.

---

Function `.onInit`

```
    ; Check if the OS is exactly Windows 10
    ; ${IsWin10} checks for Windows 10 specifically
    ${IfNot} ${IsWin10}
```

```
        MessageBox MB_OK|MB_ICONSTOP "Error: This software can
only be installed on Windows 10."
```

```
        Quit ; Closes the installer immediately
```

```
    ${EndIf}
```

FunctionEnd

---

[25.12.2025.21.07.wib]

---

To allow installation only on Windows 10 and 11 and verify Administrator privileges, you need to update your `.onInit` function.

Add these updates to your script:

### 1. Required Includes

Ensure these are at the very top of your `.nsi` file:

---

---

User Experience: Instead of the installer failing halfway through, the user gets a clear explanation immediately.

Windows 11 Note: If you want to allow both Windows 10 and Windows 11, you should use `#{AtLeastWin10}` instead of `#{IsWin10}`.

Updated "Professional" Placement:

In your full script, place the `.onInit` function right before or after the `.onGUIInit` function (the one we used for the green progress bar).

Would you like to also check if the user has Administrator privileges before allowing the install to proceed?

---

```
!include "WinVer.nsh"
```

```
!include "LogicLib.nsh"
```

---

The Updated `.onInit` Function

This function will now perform two critical checks before the installer even opens:

---

Function `.onInit`

```
; --- 1. Check for Administrator Privileges ---
```

```
UserInfo::GetAccountType
```

```
Pop $0
```

```
#{If} $0 != "Admin"
```

```
    MessageBox MB_OK|MB_ICONSTOP "Error: Administrator  
privileges are required. Please right-click the installer and select 'Run  
as administrator'."
```

```
Quit
${EndIf}

; --- 2. Check for Windows 10 or 11 ---

; ${AtLeastWin10} includes Windows 10, Windows 11, and modern
Server editions.

${IfNot} ${AtLeastWin10}

    MessageBox MB_OK|MB_ICONSTOP "Error:
${PRODUCT_NAME} can only be installed on Windows 10 or Windows
11."

    Quit
${EndIf}

FunctionEnd
```

-----

Why this works:

`${AtLeastWin10}`: Since Windows 11 identifies itself internally as version 10.0 (build 22000+), "AtLeastWin10" is the standard way to cover both operating systems.

`UserInfo::GetAccountType`: This queries the Windows user database. If the returned value isn't "Admin," it stops the installation to prevent "Access Denied" errors when writing to Program Files.

**Visual Consistency**: Both checks use `MB_ICONSTOP` (a red 'X'), which looks professional and matches your dark red theme.

Quick Consistency Check:

Make sure you still have this line in your General Settings to trigger the Windows UAC (User Account Control) prompt automatically:

-----  
RequestExecutionLevel admin  
-----

To check for 4GB of RAM and ensure enough Disk Space, you need to add specific logic to your .onInit function and use the AddSize command.

### 1. Check for 4GB RAM

NSIS uses the System plugin to query the hardware. Since 4GB is actually

MB, we check for that value.

Add this into your Function .onInit:

-----

```
; --- 3. Check for Minimum RAM (4GB) ---
```

```
; Get total physical memory in KB
```

```
System::Alloc 64
```

```
Pop $0
```

```
System::Call "kernel32::GlobalMemoryStatusEx(i r0)"
```

```
System::Call "*$0(i, i, l .r1, l .r2, l .r3, l .r4, l .r5, l .r6, l .r7, l .r8)"
```

```
System::Free $0
```

```
; Convert KB to MB (approximately)
```

```
System::Int64Op $2 / 1048576
```

```
Pop $R0 ; $R0 now contains RAM in MB
```

```
#{If} $R0 < 4000 ; Using 4000 to be safe with hardware reporting
```

```
MessageBox MB_OK|MB_ICONEXCLAMATION "Warning: This application requires at least 4GB of RAM. You only have $R0 MB."
```

```
    ; Quit ; Uncomment this if you want to force-stop the install  
    ${EndIf}
```

---

## 2. Check for Disk Space

NSIS handles disk space checks automatically if you tell it how big your files are.

In your Installation Section, use AddSize:

---

```
Section "Main Application" SEC01
```

```
    ; Tells the installer this section needs 500MB (in Kilobytes)
```

```
    AddSize 500000
```

```
    SetOutPath "$INSTDIR"
```

```
    File "ApacheInstallerWizard.exe"
```

```
    ; ... rest of your files
```

```
SectionEnd
```

---

### How Disk Space works in NSIS:

**Automatic Calculation:** When you use the File command, NSIS calculates the space needed.

Manual Override: If your app downloads extra data later or extracts large archives, AddSize manually tells the installer to check for that extra space before allowing the user to click "Install."

User Feedback: If the user doesn't have enough space on their chosen drive, the "Install" button will be greyed out, and a "Disk space required" warning will appear in red.

Combined Final .onInit Logic:

Your .onInit should now look like this order:

Admin Check (Highest priority)

Windows Version Check (OS compatibility)

RAM Check (Hardware compatibility)

Would you like to add a System Requirement Summary on the Welcome page so users see these specs immediately?

-----

<https://httpd.apache.org/>

<https://www.php.net/>

<https://mariadb.org/>

----- bundle ma'isi

~m7r=0<3<v0 =z v v0z7777 z=7777

by: piasDraco [/pias : pias25]