Generate `autounattend.xml` files for Windows 10/11

» schneegans.de

This service lets you create [↗ answer files](#) (typically named `unattend.xml` or `autounattend.xml`) to perform **unattended installations** of both Windows 10 and Windows 11, including 24H2. Answer files generated by this service are primarily intended to be used with Windows Setup run from Windows PE to perform clean (rather than upgrade) installations.

[Usage](#) · [Sample scripts](#) · [↗ GitHub](#) · [↗ Donate via PayPal](#) · [↗ Buy Me a Coffee](#)

Import a file generated by this service

| Choose File | No file chosen | Import file |

Presets

| Reset form to default values | Configure for minimal output |

Region and language settings:

◉ Install Windows using these language settings:

Windows display language:

| English ⌄ |

Windows features like Settings and File Explorer will appear in this language. It must match the language of your Windows 10/11 .iso file.

Choose language preferences and keyboard layouts

Select one or more languages in order of preference. The first language will also determine the initial **regional format**, which defines how numbers, dates, times and currency are formatted. You can change the regional format later.

☑ Specify the first language and keyboard layout

Language:

| English (United States) ⌄ |

Keyboard layout / Input Method Editor:

| US ⌄ |

☐ Specify a second language and keyboard layout

Language:

| German (Germany) ⌄ |

Keyboard layout / Input Method Editor:

| German ⌄ |

☐ Specify a third language and keyboard layout

Language:

| Spanish (Spain) ⌄ |

Keyboard layout / Input Method Editor:

Spanish   ⌄

Home location:

United States   ⌄

Windows and apps might use your country or region to give you local content.

○ Select language settings interactively during Windows Setup

Processor architectures:

Intel / AMD 32-bit
Intel / AMD 64-bit
Windows on Arm64

When you select multiple processor architectures, a single `autounattend.xml` file will be created that is applicable to all of these architectures.

Setup settings: ☑ Bypass Windows 11 requirements check (TPM, Secure Boot, etc.)

☐ Allow Windows 11 to be installed without internet connection

This effectively runs the `oobe\BypassNRO.cmd` command, which was discovered by Reddit user ↗ AveYo. You still have to click the *I don't have internet* button during Windows Setup.

⚠ Only check this option if your computer really does not have internet access. If you just want to create local ("offline") user accounts in Windows 11, you can always do so in the *User accounts* section of this form.

☐ Use a distribution share / configuration set

This makes Windows Setup look for a folder named `$OEM$` in the root of the drive where your `autounattend.xml` file is located and ↗ copy its contents to the target partition.

☐ Hide any PowerShell windows during Windows Setup

By default, PowerShell scripts that run during Windows Setup will have a visible window. This makes debugging easier and lets you use interactive prompts in your scripts, but also carries the risk that this window gets inadvertently or deliberately closed, which would kill the script. If you choose to hide PowerShell windows, any PowerShell scripts during Windows Setup will be run with the `-WindowStyle Hidden` switch.

Computer name:

○ Let Windows generate a random computer name like `DESKTOP-ZFAH8Z2`

○ Choose a computer name yourself

Use this name: Portable

● Provide a Windows PowerShell script to set the computer name dynamically

Your script will be evaluated during Windows Setup. The script must return a single string, which must be a ↗ valid computer name. You can also use a script such as

` return Read-Host -Prompt 'Enter computer name'; ` to create an interactive prompt.

```
return Read-Host -Prompt 'Enter computer name';
```

Compact OS: ◉ Let Windows decide whether to use Compact OS

○ Use Compact OS

○ Do not use Compact OS

Time zone: ○ Let Windows determine your time zone based on language and region settings

◉ Set your time zone explicitly

This is useful when your country or region spans multiple time zones, like Australia or the United States.

Use this time zone: (UTC-06:00) Central Time (US & Canada) ⌄

Partitioning and formatting: ◉ Partition the disk interactively during Windows Setup

○ Let Windows Setup wipe, partition and format your hard drive (more specifically, disk 0) using these settings:

┌─ Choose partition layout ───┐

◉ GPT

The ↗ GPT partition layout must be used for UEFI systems. Set the size of the EFI System Partition (ESP) to `300`   MB.

○ MBR

MBR The ↗ MBR-based partition layout must be used for legacy BIOS systems.

└───┘

┌─ Choose how to install Windows RE ────────────────────────────────┐

◉ Install on recovery partition

Create a separate partition with a size of `1000`   MB and install Windows RE to it.

○ Install on Windows partition

This will install Windows RE in `C:\Recovery` . No recovery partition will be created.

○ Remove Windows RE

This will delete the `C:\Recovery` folder and thus free about 600 MB of disk space. No recovery partition will be created.

> Windows 24H2 seems to ignore this setting and will always create a recovery partition with a minimum size of 600 MB.

○ Use a custom `diskpart` script to configure your disk(s):

```
SELECT DISK=0
CLEAN
CONVERT GPT
CREATE PARTITION EFI SIZE=300
FORMAT QUICK FS=FAT32 LABEL="System"
CREATE PARTITION MSR SIZE=16
CREATE PARTITION PRIMARY
SHRINK MINIMUM=1000
FORMAT QUICK FS=NTFS LABEL="Windows"
CREATE PARTITION PRIMARY
FORMAT QUICK FS=NTFS LABEL="Recovery"
SET ID="de94bba4-06d1-4d40-a16a-bfd50179d6ac"
GPT ATTRIBUTES=0x8000000000000001
```

If your disk is already partitioned and formatted, enter a `diskpart` command that has no effect, like `REM` or `SELECT DISK=0`. Also note that drive letter assignments (e.g. `ASSIGN LETTER=R`) will not persist.

┌─ Choose partition to install Windows to after script has run ──────────
│ ◉ Install Windows to the first available partition that has enough space and does
│   not already contain an installation of Windows
│
│ ○ Install to another partition:
│
│   Disk (0–based):   0
│
│   Partition (1–based):   3
└──

When you let Windows Setup partition your disks unattendedly, there is a risk they were assigned unexpected index numbers. In rare cases, **disk 0** does not refer to your primary hard drive, but rather your USB thumb drive. You can therefore provide VBScript code to check the assigned disk index numbers before `diskpart` is run. If your script returns with `WScript.Quit 1`, Windows Setup will halt to avoid data loss. BETA

◉ Do not run a script

○ Run this VBScript code to check disk layout:

```
On Error Resume Next
Set wmi = GetObject("winmgmts:\\.\root\cimv2")
Set drive = wmi.Get("Win32_DiskDrive.DeviceID='\\.\PHYSICALDRIVE0'")
If Err.Number <> 0 Then
    Msgbox Err.Description
    WScript.Quit 1
End If
If drive.InterfaceType = "IDE" Or drive.InterfaceType = "SCSI" Then
    WScript.Quit 0
Else
    MsgBox "Assertion failed."
    WScript.Quit 1
End If
```

Windows edition:

◉ Use a **generic** product key

Such a key can be used to install Windows, but will not activate it. You can change the product key later.

Install this edition of Windows: [Pro   ▾]

○ Enter another product key

Use this product key: [VK7JG-NPHTM-C97JM-9MPGT-3V66T]

○ Enter a product key interactively during Windows Setup

You can also enter your key in the `autounattend.xml` file yourself to avoid disclosing it. To do so, find the `<Key>00000-00000-00000-00000-00000</Key>` element and replace the text with your own key.

○ Use product key stored in BIOS/UEFI firmware

Choose this if your computer came pre-installed with Windows and you want to reuse that license.

User accounts:

◉ Let Windows Setup create the following local ("offline") accounts:

Account name	Display name	Password	Group
So0oClose			Administrators ▾
Admin			Administrators ▾
			Users ▾
			Users ▾
			Users ▾

Leave **Display name** empty unless you want it to be different from **Account name**.

┌─ First logon ───┐
│ Several settings will only be applied when an administrator logs on for the first │
│ time. You should therefore let Windows log you on to an administrator account │
│ once – this does not affect subsequent logons. Choose which account to use for │
│ this: │
│ │
│ ◉ Logon to the first administrator account created above │
│ │
│ ○ Activate built-in account "Administrator" and logon to this account │
│ │
│ Set its password to: [   ] │
│ │
│ ○ Do not logon │
│ │
│ The installation ends with the sign-in screen being shown. │
└───┘

☐ Obscure all account passwords in your `autounattend.xml` file with Base64

○ Add a Microsoft ("online") user account interactively during Windows Setup

○ Add a local ("offline") user account interactively during Windows Setup

Password expiration:

◉ Passwords do not expire

This is in accordance to NIST guidelines that ↗ no longer recommend password expiration.

○ Use Windows default

Passwords expire after **42** days.

○ Use custom password expiration:

Passwords expire after [42] days.

These settings only apply to local accounts. Also, the password of the built-in account "Administrator" never expires.

Account Lockout policy:

◉ Use default policy

By default, Windows will lock out an account after **10 failed logon attempts** ("threshold") within **10 minutes** ("window"). After **10 minutes** ("duration"), the account is unlocked automatically.

○ Disable policy

⚠ Disabling Account Lockout might leave your computer vulnerable to brute-force attacks.

○ Use custom policy:

Lock out an account after [10] failed logon attempts within [10] minutes. After [10] minutes, unlock the account automatically.

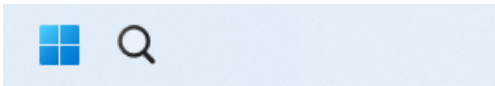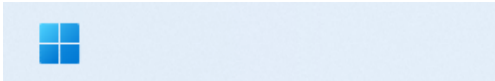**File Explorer tweaks:**
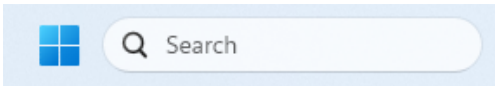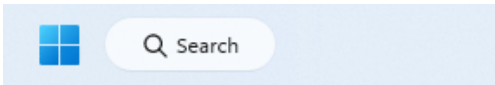
┌─ Choose which files are hidden in File Explorer ─────

◉ Use File Explorer default setting

This will hide any file that has the `Hidden` attribute set.

○ Hide protected operating system files

This will only hide files that have both the `Hidden` and the `System` attribute set.

○ Show all files

This setting is recommended for advanced users.
└────────────────────────────────

☑ Always show file extensions

By default, File Explorer would hide extensions for known file types.

☑ Use classic context (right-click) menu in Windows 11

☑ Open File Explorer to *This PC* instead of *Quick access*

☐ Show *End task* command in the taskbar NEW

Start menu and taskbar:

Choose how to display the search box in the taskbar

○ 🪟 🔍 Search   Full

○ 🪟 🔍 Search   Icon and label

○ 🪟 🔍   Icon

◉ 🪟   Hide

Choose icons to display in the taskbar

○ Use default icons

◉ Remove all icons

○ Configure icons using an XML file:

```
<LayoutModificationTemplate xmlns="http://schemas.microsoft.com/Start/2014/LayoutModi
  <CustomTaskbarLayoutCollection PinListPlacement="Replace">
    <defaultlayout:TaskbarLayout>
      <taskbar:TaskbarPinList>
        <taskbar:DesktopApp DesktopApplicationID="Microsoft.Windows.Explorer" />
        <taskbar:UWA AppUserModelID="windows.immersivecontrolpanel_cw5n1h2txyewy!micr
        <taskbar:DesktopApp DesktopApplicationLinkPath="%APPDATA%\Microsoft\Windows\S
      </taskbar:TaskbarPinList>
    </defaultlayout:TaskbarLayout>
  </CustomTaskbarLayoutCollection>
</LayoutModificationTemplate>
```

See Microsoft's documentation for ↗ **more examples**.

☑ Disable widgets

This hides the news and weather widget in the lower-left corner in Windows 11.

☑ Left-align the taskbar in Windows 11

☑ Hide the *Task view* button from the taskbar

☐ Always show all tray icons

On Windows 10, this will simply set the `EnableAutoTray` registry value. On Windows 11, this will create a scheduled task that runs periodically in the background and sets all tray icons visible.

☑ Do not show Bing results when searching in the Start menu or the search box

Windows 10

Configure the tiles that are initially shown in the Windows 10 Start menu. You can add or remove tiles later. On Windows 11, this setting is simply ignored.

○ Use default tiles

If you select apps in the *Remove bloatware* section below, Windows 10 will automatically remove their tiles from the Start menu.

○ Remove all tiles

◉ Configure tiles using an XML file created by PowerShell's `Export-StartLayout` cmdlet on another Windows 10 computer:

```
        <start:Group Name="">
          <start:Tile Size="2x2" Column="0" Row="2" AppUserModelID="44576milosp.One
          <start:DesktopApplicationTile Size="2x2" Column="0" Row="4" DesktopApplic
          <start:DesktopApplicationTile Size="2x2" Column="4" Row="2" DesktopApplic
          <start:DesktopApplicationTile Size="2x2" Column="2" Row="2" DesktopApplic
          <start:DesktopApplicationTile Size="2x2" Column="0" Row="0" DesktopApplic
          <start:DesktopApplicationTile Size="2x2" Column="4" Row="0" DesktopApplic
          <start:DesktopApplicationTile Size="2x2" Column="4" Row="4" DesktopApplic
          <start:DesktopApplicationTile Size="2x2" Column="2" Row="0" DesktopApplic
          <start:DesktopApplicationTile Size="2x2" Column="2" Row="4" DesktopApplic
        </start:Group>
      </defaultlayout:StartLayout>
    </StartLayoutCollection>
  </DefaultLayoutOverride>
</LayoutModificationTemplate>
```

Windows 11

Configure the pins that are initially shown in the Windows 11 Start menu. You can add or remove pins later. On Windows 10, this setting is simply ignored.

○ Use default pins

⚠ Avoid this setting if you select apps in the *Remove bloatware* section below, as Windows 11 will deceptively retain their pins in the Start menu.

○ Remove all pins

◉ Configure pins using a JSON file created by PowerShell's `Export-StartLayout` cmdlet on another Windows 11 computer:

```
{
  "pinnedList": [
    {
      "desktopAppLink": "%ALLUSERSPROFILE%\\Microsoft\\Windows\\Start Menu\\Programs\
    },
    {
      "desktopAppLink": "%APPDATA%\\Microsoft\\Windows\\Start Menu\\Programs\\File E>
    },
    {
      "packagedAppId": "windows.immersivecontrolpanel_cw5n1h2txyewy!microsoft.window$
    }
  ]
}
```

System tweaks: ☐ Disable Windows Defender

This disables certain services (`Sense` , `WdBoot` , `WdFilter` , `WdNisDrv` , `WdNisSvc` , `WinDefend`) during the Windows PE stage of Windows Setup and thus prevents the `MsMpEng.exe` process from running. This method was adapted from an article by ↗ Rudy Mens.

☐ Disable Windows Update

This will create a scheduled task (named `PauseWindowsUpdate`) that pauses updates for one week again and again. If you want to run Windows Update for once, click *Resume updates* in Settings. If you want to enable Windows Update permanently, disable or delete that task.

☐ Disable User Account Control (UAC)

☐ Disable Smart App Control

This turns Smart App Control off in Windows 11. Note that you will not be able to turn it back on.

☐ Disable SmartScreen in Windows and Edge

☐ Disable Fast Startup

☐ Disable System Protection / System Restore

Windows will not create restore points for drive `C:` and thus use less disk space.

☐ Enable long paths

This sets the ↗ LongPathsEnabled registry value, which enables several programs (including PowerShell, 7-Zip and TreeSize) to use long paths with up to 32,767 characters without resorting to the `\\?\` prefix.

☑ Enable Remote Desktop services (RDP)

☐ Harden ACLs

This removes write permissions on `C:\` for the *Authenticated Users* group. In particular, this prevents unprivileged users from creating bogus folders such as `C:\Windows`.

☑ Allow execution of PowerShell script files

This runs the command `Set-ExecutionPolicy -ExecutionPolicy 'RemoteSigned'`, which allows the execution of unsigned `.ps1` files.

☐ Do not update Last Access Time stamp

This runs the command `fsutil.exe behavior set disableLastAccess 1`, which can improve file system performance.

☑ Prevent Windows Update from rebooting your computer

This creates a scheduled task that periodically moves your ↗ active hours, tricking Windows into thinking your device is in use all the time.

☐ Turn off system sounds

This changes the sound scheme from *Windows Default* to *No sounds* for all users.

☑ Disable app suggestions / Content Delivery Manager

This sets ↗ several registry values that prevent the silent download and installation of ↗ suggested apps. Note that as a side effect this will also disable the *Manage devices* button in the *Mobile devices* section in Settings. If this bothers you, ↗ change the value of `DisableWindowsConsumerFeatures` to `0`.

☑ Prevent device encryption

Windows 11 would otherwise enable ↗ BitLocker encryption automatically.

☐ Hide Edge *First Run Experience*

☐ Make Edge uninstallable

This will modify the `C:\Windows\System32\IntegratedServicesRegionPolicy Set.json` file such that Edge can be uninstalled even outside the European Economic Area. Note that Windows Update will eventually reset the file to its original state. Therefore, if you want to uninstall Edge, do so right after Windows has been installed. Also, you might see an error `0x80070306` when installing a cumulative update (such as KB5048667), but this should get resolved when you repeat the installation of that update.

☐ Disable the *Enhance Pointer Precision* mouse setting

This may be useful for players of first-person shooters.

☑ Delete empty `C:\windows.old` folder

☐ Audit process creation events

Each time a new process is created, Windows writes an event to the *Security* log. This is a powerful tool for troubleshooting.

    ☐ Include command line in log events

Visual effects: ◉ Use default settings chosen by Windows

          ○ Adjust for best appearance

          ○ Adjust for best performance

          ○ Use custom settings:

             ☐ Animate controls and elements inside windows

             ☐ Animate windows when minimizing and maximizing

             ☐ Animations in the taskbar

             ☐ Enable Peek

             ☐ Fade or slide menus into view

             ☐ Fade or slide ToolTips into view

             ☐ Fade out menu items after clicking

             ☐ Save taskbar thumbnail previews

             ☐ Show shadows under mouse pointer

             ☐ Use drop shadows for icon labels on the desktop

             ☐ Show thumbnails instead of icons

             ☐ Show translucent selection rectangle

             ☐ Show window contents while dragging

             ☐ Slide open combo boxes

             ☐ Smooth edges of screen fonts

☐ Smooth-scroll list boxes

☐ Show shadows under windows

Desktop icons: ○ Show default desktop icons

◉ Show these desktop icons:

☐ Control Panel	☐ Desktop	☐ Documents
☐ Downloads	☐ Gallery	☐ Home
☐ Music	☐ Network	☐ Pictures
☑ Recycle bin	☐ This PC	☐ User's Files
☐ Videos		

Virtual machine support:

☐ Install **Oracle VirtualBox Guest Additions**

☐ Install **VMware Tools**

☐ Install **VirtIO Guest Tools** and **QEMU Guest Agent** (e.g. for Proxmox VE)

Make sure to check the [usage notes](https://schneegans.de/windows/unattend-generator/) for how to properly configure your VM.

WLAN / Wi-Fi setup:

○ Configure Wi-Fi interactively during Windows Setup

◉ Skip Wi-Fi configuration

Choose this if you have a wired connection to the internet.

○ Configure Wi-Fi using these settings:

Network name (SSID):   [   ]

Connect even if not broadcasting: ☐

Authentication: Open   ⌄

Password: 00000000

If both your Wi-Fi router and your computer's Wi-Fi adapter support it, make sure to select WPA3. Otherwise, Windows Setup will try to switch from WPA2 to WPA3 and require manual interaction.

You should not enter your actual Wi-Fi password here. Once you have downloaded the `autounattend.xml` file, find the password enclosed in `<keyMaterial>…</keyMaterial>` and adjust it.

○ Configure Wi-Fi using an XML file created by `netsh.exe wlan export profile key=clear` on another computer:

```
<WLANProfile xmlns="http://www.microsoft.com/networking/WLAN/profile/v1">
  <name>WLAN-123456</name>
  <SSIDConfig>
    <SSID>
      <hex>574C414E2D313233343536</hex>
      <name>WLAN-123456</name>
    </SSID>
    <nonBroadcast>true</nonBroadcast>
  </SSIDConfig>
  <connectionType>ESS</connectionType>
  <connectionMode>auto</connectionMode>
  <MSM>
    <security>
      <authEncryption>
        <authentication>WPA3SAE</authentication>
        <encryption>AES</encryption>
        <useOneX>false</useOneX>
        <transitionMode xmlns="http://www.microsoft.com/networking/WLAN/profile/v4">true<
      </authEncryption>
      <sharedKey>
        <keyType>passPhrase</keyType>
        <protected>false</protected>
        <keyMaterial>password</keyMaterial>
      </sharedKey>
    </security>
  </MSM>
</WLANProfile>
```

Express settings:	◉ Disable all
	Windows will not send diagnostic data, personalized input or your location history to Microsoft. Choose this if you value privacy.
	○ Enable all
	Windows will send data to Microsoft to provide location-based services, improve language recognition, and show personalized ads.
	○ Choose settings interactively during Windows Setup
	This lets you enable some settings while disabling others.

Lock key settings:

○ Use default lock key states and behaviors

◉ Configure the initial state and behavior of lock keys as follows:

Key	Initial state	When pressed
Caps Lock	Off ⌄	Toggle state ⌄
Num Lock	On ⌄	Toggle state ⌄
Scroll Lock	Off ⌄	Toggle state ⌄

This will affect all users and also the login screen.

Personalization settings: These settings are particularly useful if you want to use Windows without activation when the *Personalization* settings page is not available.

┌─ Colors BETA ───┐

○ Use default color theme

◉ Use custom color theme

   Choose color theme for taskbar and Start menu: [Dark   ▾]

   Choose color theme for apps: [Dark   ▾]

   Accent color: [   ]

   Show accent color on Start menu and taskbar: ☐

   Show accent color on title bars and window borders: ☐

   Windows and surfaces appear translucent: ☐

└───┘

┌─ Desktop wallpaper ───┐

◉ Use default wallpaper settings

○ Use a solid color background:

   [   ]

○ Provide a Windows PowerShell script to load a desktop wallpaper image

   Your script will be evaluated during Windows Setup, after your computer has
   connected to a network. The script must return a `byte[]` value, which must
   contain the image data. For example, you can download an image from the
   internet, locate an image file on a removable drive or a network share and load it
   via `[IO.File]::ReadAllBytes`, or provide the image data directly via
   `[convert]::FromBase64String`.

```
# Photo by Jess Loiterton: https://www.pexels.com/photo/mountains-and-beach-under-dra
$url = 'https://images.pexels.com/photos/5007737/pexels-photo-5007737.jpeg';
return ( Invoke-WebRequest -Uri $url -UseBasicParsing -TimeoutSec 30 ).Content;
```

└───┘

Remove bloatware: Windows comes with several apps that many users do not want or do not need. Check all the apps you want removed during Windows Setup:

[Select all] [Deselect all]

☑ 3D Viewer   ☑ Bing Search
☐ Calculator   ☑ Camera
☑ Clipchamp   ☐ Clock
☐ Copilot   ☑ Cortana
☑ Dev Home   ☑ Family
☑ Feedback Hub   ☑ Get Help
☑ Handwriting (all languages) BETA   ☑ Internet Explorer
☑ Mail and Calendar   ☑ Maps

☑ Math Input Panel	☐ Media Features
☑ Mixed Reality	☑ Movies & TV
☑ News	☐ Notepad (modern)
☑ Office 365	☑ OneDrive
☐ OneNote	☑ OneSync
☐ OpenSSH Client	☑ Outlook for Windows
☑ Paint	☑ Paint 3D
☑ People	☑ Photos
☑ Power Automate	☑ PowerShell 2.0
☑ PowerShell ISE	☑ Quick Assist
☐ Recall	☐ Remote Desktop Client
☑ Skype	☐ Snipping Tool
☑ Solitaire Collection	☑ Speech (all languages) BETA
☐ Steps Recorder	☑ Sticky Notes
☑ Teams	☑ Tips
☑ To Do	☑ Voice Recorder
☑ Wallet	☑ Weather
☑ Windows Fax and Scan	☐ Windows Hello
☑ Windows Media Player (classic)	☑ Windows Media Player (modern)
☐ Windows Terminal	☑ WordPad
☑ Xbox Apps	☑ Your Phone / Phone Link

⚠ Bloatware removal works best with the original Windows ↗ 10 and ↗ 11 .iso images downloaded from Microsoft. I did not perform any tests with custom images.

Run custom scripts:

You may want to take a look at some sample scripts first.

┌─ Scripts to run in the system context, before user accounts are created ─────

1. Run

```

```

as a [.cmd ∨] file.

2. Run

```
New-NetFirewallRule -DisplayName 'ICMPv4' -Profile 'Any' -Protocol 'ICMPv4';
```

as a [.ps1 ∨] file.

3. Run

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Remote Assistance]
"fAllowToGetHelp"=dword:00000000
```

as a `.reg` ∨ file.

4. Run

```
WScript.Echo "Hello!"
```

as a `.vbs` ∨ file.

Scripts to modify the default user's registry hive

This will automatically mount the `C:\Users\Default\NTUSER.DAT` hive, run your
scripts and unmount the hive again. Your scripts will be run *before* user accounts are
created – hence, they will affect all user accounts, including the built-in account
"Administrator". You must access the keys in this hive as follows:

.reg   `[HKEY_USERS\DefaultUser\…`

.cmd `HKU\DefaultUser\…`

.ps1 `Registry::HKU\DefaultUser\…`

1. Run

```
Windows Registry Editor Version 5.00

[HKEY_USERS\DefaultUser\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced]
"ShowTaskViewButton"=dword:00000000
```

as a `.reg` ∨ file.

2. Run

```
reg.exe add "HKU\DefaultUser\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Adva
```

as a `.cmd` ∨ file.

3. Run

```
Set-ItemProperty -LiteralPath "Registry::HKU\DefaultUser\SOFTWARE\Microsoft\Windows\
```

as a `.ps1` ∨ file.

Scripts to run when the first user logs on after Windows has been installed

The first user to log on is typically an administrator. In this case, these scripts will run
with elevated privileges.

1. Run

```
setx.exe DIRCMD "/A /O:GN /C /N"
```

as a .cmd ⌄ file.

2. Run

```
Get-NetConnectionProfile -InterfaceAlias 'Ethernet*' | Set-NetConnectionProfile -Net
```

◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▶

as a .ps1 ⌄ file.

3. Run

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Control Panel\Desktop]
"ScreenSaverIsSecure"="1"
```

as a .reg ⌄ file.

4. Run

```
WScript.Echo "Hello!"
```

as a .vbs ⌄ file.

Scripts to run whenever a user logs on for the first time

1. Run

```
setx.exe DIRCMD "/A /O:GN /C /N"
```

as a .cmd ⌄ file.

2. Run

```
Set-WinHomeLocation -GeoId 94;
```

as a .ps1 ⌄ file.

3. Run

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Control Panel\Desktop]
"ScreenSaverIsSecure"="1"
```

as a .reg ⌄ file.

4. Run

```
WScript.Echo "Hello!"
```

as a `.vbs` ▾ file.

☐ Restart File Explorer after scripts have run

This will restart the `explorer.exe` process that is responsible for displaying Start menu, taskbar and Desktop. If your scripts in this section make changes to any of these elements, check this box so that they take effect immediately.

Your scripts will be run as follows:

.cmd `C:\Windows\Setup\Scripts\unattend-01.cmd`

.ps1 `powershell.exe -WindowStyle Normal -NoProfile -Command "Get-Content -LiteralPath 'C:\Windows\Setup\Scripts\unattend-02.ps1' -Raw | Invoke-Expression;"`

.reg `reg.exe import "C:\Windows\Setup\Scripts\unattend-03.reg"`

.vbs `cscript.exe //E:vbscript "C:\Windows\Setup\Scripts\unattend-04.vbs"`

.js `cscript.exe //E:jscript "C:\Windows\Setup\Scripts\unattend-05.js"`

Windows Defender Application Control:

◉ Do not configure WDAC policy

○ Configure a basic WDAC policy using these settings:

Applications in `C:\Windows` , `C:\Program Files` and `C:\Program Files (x86)` are allowed to run. Applications stored elsewhere and those in known user-writable folders such as `C:\Windows\Temp` or `C:\Windows\Debug\WIA` are not allowed to run. To disable this WDAC policy later, simply delete the file `C:\Windows\System32\CodeIntegrity\CiPolicies\Active\{d26bff32-33a2-48a3-b037-10357ee48427}.cip` and reboot. To create a more customized policy, see my [online WDAC generator](#).

┌─ Choose how to enforce the policy ──────────────────────────────
│
│ ○ Auditing mode
│
│   Logs drivers and applications that would have been blocked.
│
│ ◉ Auditing mode on boot failure
│
│   When the policy blocks a system driver and thus would prevent Windows from booting, use audit mode. Otherwise, use enforcement mode.
│
│ ○ Enforcement mode
│
│   Drivers and applications will be blocked unless allowed by the policy.
│
└──

┌─ Choose script enforcement ─────────────────────────────────────
│
│ ◉ Restricted
```

PowerShell will run in **Constrained Language Mode**. See ↗ Script Enforcement
for details.

○ Unrestricted

PowerShell will run in **Full Language Mode**.

**Placeholders for more components:**

You can optionally generate templates for ↗ all available components, with respect to their valid configuration passes. Look for `<!--Placeholder-->` comments in the generated `autounattend.xml` file and fill in the desired settings yourself.

☐ auditSystem
☐ auditUser
☐ oobeSystem

↗ Microsoft–Windows–DeviceAccess

☐ specialize

↗ Microsoft–Windows–DeviceGuard–Unattend

☐ offlineServicing

↗ Microsoft–Windows–DiagCpl

☐ offlineServicing
☐ generalize
☐ specialize

**Download settings:**

☑ Use filename `notautounattend.xml` rather than `autounattend.xml`

Windows Setup will not process the `notautounattend.xml` file automatically. Instead, you need to run a command such as `setup.exe /Unattend: D:\notautounattend.xml`. This is useful to prevent Windows Setup from inadvertently wiping your hard drive, and lets you specify ↗ additional parameters such as `/NoReboot`. In particular, calling `setup.exe /NoReboot /Unattend: D:\notautounattend.xml` offers a great opportunity to remove 8.3 file names during Windows Setup while still using an answer file.

**Submit form:**

[ Bookmark selection ]  [ View .xml file ]  [ Download .xml file ]  [ Download .iso file ]

---

Contact: Christoph Schneegans (✉ christoph@schneegans.de)