

# Windows install

- [Win10-11 Install](#)

# Win10-11 Install

## Get the Windows iso

for [Windows 10](#)

for [Windows 11](#)

download the installation media and create a iso which you will save to a USB

[video guide](#)

when running the installation media. Select `Create installation media` and when asked "What media to use" select `ISO file`

## install win10

use win10 iso and select pro edition, don't use a product key and use local/offline account. Turn off all advertising options and optional features such as cortana

once setup run this in admin powershell `irm https://get.activated.win/ | iex` and select HWID , this option will make a product key onto your bios for future reinstalls and activate your windows

## go through win optimization scripts

[video](#)

on ventoy use the `Ultimate-Windows-Optimization-Guide-main` scripts or download from <https://github.com/FR33THYFR33THY/Ultimate-Windows-Optimization-Guide> > select code > download zip

run `Allow Scripts` and turn on (recommended)

## go to 3 setup

run 4 Data Language Region Time and change timezone to Indiana East then sync clock

run 8 Startup Apps and deselect all except the ones you want to run on startup

run 9 Background Apps and select option 1 to turn off

run 11 Store Updates and turn off everything

run 12 Windows Updates Pause spam the pause updates for 7 days until it pauses, this needs to run every 5 weeks

## go to 5 graphics

run 4 Nvidia Driver use option 1 (recommended)

run 5 Nvidia settings use option 1 (recommended)

run 6 Resolution Refresh Rate change refresh rate to max

run 10 Direct X and 11 C++ and install all

## go to 6 Windows

run 1 Start Menu Taskbar use option 1 (recommended)

run 3 Copilot use option 1 to turn off (recommended)

run 4 Widgets use option 1 to turn off (recommended)

run 5 Gamemode turn on

run 6 Gamebar use option 1 to turn off (recommended)

run 7 Pointer Precision put pointer speed on 6 and turn off enhance pointer precision

run 8 Scaling turn off let windows try to fix apps so they're not blurry

run 9 Power Plan use option 1 to turn on (recommended)

run 10 Timer Resolution use option 1 to turn on (recommended)

run 11 Core Isolation make sure Memory Isolation is turned off

run `12 Registry` use option 1 to optimize (recommended)

run `13 Signout Lockscreen` use option 1 for black (recommended)

run `14 Edge` use option 1 to turn edge off (recommended)

run `15 Bloatware` use option 2 to remove all bloatware (recommended) if microsoft store is needed install with option 3

run `22 Cleanup` to cleanup drive and temp files

run `23 Restore Point` to create a restore point now

## go to `7 Hardware`

run `1 Mouse Optimization`

# install winget

get the `App Installer` <https://apps.microsoft.com/detail/9nblggh4nns1?hl=en-US&gl=US>

now check that winget is installed with `winget` in powershell

install software with winget script

## edit the winget script

copy and paste the script into notepad, then `save as` and select `all file types` then name the script and add filetype as `.ps1` like `winget_install.ps1`

in powershell to get the winget program id, use command `winget search <program name>` then replace the `Name = "example"` line and the `--id example.example` line in the script for all the programs you wish to install.

find the script in `file explorer` and `right click` then `run with powershell`

```
# Install Programs using Winget
# This script uses Winget to download and install programs.

# Function to check if Winget is installed
function Check-Winget {
    if (-not (Get-Command winget -ErrorAction SilentlyContinue)) {
        Write-Output "Winget is not installed. Please ensure Windows is updated."
    }
}
```

```

    exit 1
}
}

# Programs to Install - Add or Remove programs here
# @{ Name = "Program Name"; Command = "winget install --id <program-id> --accept-package-agreements --
accept-source-agreements" }
$programs = @(
    @{ Name = "qBittorrent"; Command = "winget install --id qBittorrent.qBittorrent --accept-package-
agreements --accept-source-agreements" }
    @{ Name = "ShareX"; Command = "winget install --id ShareX.ShareX --accept-package-agreements --accept-
source-agreements" }
    @{ Name = "Discord"; Command = "winget install --id Discord.Discord --accept-package-agreements --
accept-source-agreements" }
    @{ Name = "Steam"; Command = "winget install --id Valve.Steam --accept-package-agreements --accept-
source-agreements" }
    @{ Name = "OBS Studio"; Command = "winget install --id OBSProject.OBSStudio --accept-package-
agreements --accept-source-agreements" }
    @{ Name = "notepad++"; Command = "winget install --id Notepad++.Notepad++ --accept-package-
agreements --accept-source-agreements" }
    @{ Name = "Firefox"; Command = "winget install --id Mozilla.Firefox --accept-package-agreements --accept-
source-agreements" }
    @{ Name = "7-Zip"; Command = "winget install --id 7zip.7zip --accept-package-agreements --accept-source-
agreements" }
    @{ Name = "Proton VPN"; Command = "winget install --id Proton.ProtonVPN --accept-package-agreements --
accept-source-agreements" }
    @{ Name = "Geforce Experience"; Command = "winget install --id Nvidia.GeForceExperience --accept-
package-agreements --accept-source-agreements" }

)

# Check for Winget
Check-Winget

# Install Programs
foreach ($program in $programs) {
    Write-Output "Installing $($program.Name)..."
    Invoke-Expression $program.Command
    if ($LASTEXITCODE -eq 0) {

```

```
    Write-Output "$($program.Name) installed successfully."
}
else {
    Write-Output "Failed to install $($program.Name)."
}
}

Write-Output "All installations complete!"
```