# OneDrive Control - Setup and Usage Guide

This guide walks you through setting up the OneDrive Control system, which includes a background service to manage OneDrive sync based on user activity, and a tray controller to manually override sync behavior.

## 1. Requirements

- .NET 8 SDK installed  
- Visual Studio 2022 Community or later  
- Administrator privileges for some setup steps

## 2. File Structure

Organize your files like so:  
C:\Users\<YourName>\Documents\OneDriveControlService\  
 ├── BackgroundService\  
 ├── TrayController\  
 ├── Dev\ (source code)  
 └── TrayController.exe (release binary)

## 3. Building the Background Service

- Open `BackgroundService` in VS Code or Visual Studio  
- Build with `dotnet publish -c Release -o publish`  
- Use Task Scheduler to run `OneDriveControlService.exe` on login, hidden

## 4. Tray Controller Setup

- Open `TrayController\Dev` in Visual Studio  
- Build in Release mode  
- Copy `TrayController.exe` to `TrayController\`  
- When first run, the app will place a shortcut in your startup folder

## 5. Using the Tray Controller

Right-click the tray icon to:  
- Enable sync override for 15 min, 1h, 4h, 24h, until reboot, or indefinitely  
- Cancel an active override  
- Exit the tray app

## 6. Notes

- Override status is saved in `%APPDATA%\OneDriveControl\override.txt`  
- Logs are written to the service's folder with 48-hour retention  
- The background service starts OneDrive when idle >1hr or when override is active

If you need to update the startup shortcut, delete it from the Startup folder and re-run TrayController.exe.

Tray Icon Installation and Usage

The tray icon controller allows users to easily override the automatic OneDrive control behavior directly from the system tray. This is useful for temporarily re-enabling OneDrive while working actively.

## Setup Steps

1. Navigate to the TrayController project folder. Build it using Visual Studio or run:

dotnet publish -c Release -o publish

2. Copy the following files into a folder where you want the tray app to reside (e.g., C:\Users\YourName\Documents\OneDriveControlService\TrayController):

- TrayController.exe  
- tray\_icon.ico  
- All DLL dependencies from /publish if not bundled

3. Double-click TrayController.exe to test it. You should see the cloud icon in the lower-right system tray.

## Auto Start on Login

The tray app can automatically add itself to Windows Startup:

- On first launch, it checks the Windows Startup folder for a shortcut.  
- If one does not exist, it creates a shortcut pointing to its own executable.  
- This ensures the tray app starts with Windows every time.  
  
You can confirm this by checking:  
 %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup

## Using the Tray Icon

- Right-click the tray icon to open a menu of override options.  
- Select a time period to enable OneDrive even during activity.  
- The override is saved in your AppData and automatically expires.  
- Select “Cancel override” to return to automatic mode.  
- “Exit Tray” closes the tray app until it is launched again.

# OneDrive Idle Control - Setup Guide (2025 Edition)

## What It Does

This tool:  
- Stops OneDrive when you're actively using your computer  
- Restarts OneDrive after 60 minutes of idle time  
- Launches OneDrive with administrator rights (requires UAC or elevated task)  
- Cleans up logs older than 48 hours  
- Runs invisibly in the background

## Requirements

- Windows 10 or 11  
- .NET 8.0 SDK or newer: https://dotnet.microsoft.com/en-us/download/dotnet/8.0  
- C# support in Visual Studio Code (install "C#" by Microsoft)  
- Run this command to add the missing dependency:  
  
dotnet add package Microsoft.Extensions.Hosting

## Installation Steps

1. Clone or unzip the project.  
2. Open PowerShell in the project folder.  
3. Run:  
  
dotnet restore  
dotnet publish -c Release -o publish  
  
4. The compiled executable will be inside the "publish" folder.

## Task Scheduler Setup (Optional for Startup)

1. Open Task Scheduler (Win + R → taskschd.msc).  
2. Create a new Task:  
- Run only when user is logged on  
- Run with highest privileges (important for admin launch)  
3. Add a trigger: "At log on"  
4. Add an action: "Start a program" → point to your OneDriveControlService.exe

## Log Files

Logs are saved next to the .exe as:  
  
onedrive\_log\_YYYYMMDD.log  
  
Only events (start, kill, errors) are logged. Old logs are deleted automatically after 2 days.

## To Customize Idle Time

Edit the `Worker.cs` file:  
  
const int IdleThresholdMinutes = 60;  
  
Then rebuild the project.

## To Update

Rebuild the project with:  
  
dotnet publish -c Release -o publish

## To Uninstall

1. Delete the scheduled task (if created).  
2. Delete the published files and source folder.

## Support

If OneDrive does not launch, make sure your scheduled task runs with highest privileges or run the app manually once to approve UAC.