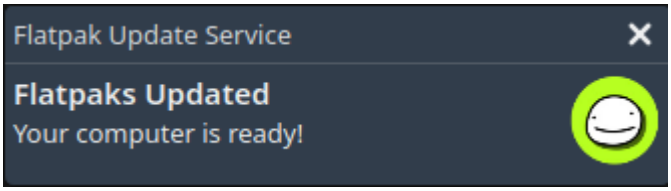
Tolga's Systemd Timer and Service for Flatpak Updates example:

This setup is a hands-free way to keep your Flatpak applications updated automatically. With the timer and service file, you don't have to worry about manually checking for updates, as everything is done for you at regular intervals. The following explains how the timer and service are structured and work together.



Timer Unit: `tolga.timer`

Create file: `sudo nano ~/.config/systemd/user/tolga.timer`

```
[Unit]
Description=Run Tolga’s Service Every 5 Minutes VER:2.0A

[Timer]
OnBootSec=1min
OnUnitActiveSec=5min
Unit=tolga.service

[Install]
WantedBy=timers.target
```

This timer is responsible for triggering the service that handles Flatpak updates.

- *`OnBootSec=30`*: The timer will wait 1 minute after the system boots up before it starts the service.
- *`OnUnitActiveSec=3`*: After the service is activated, the timer will trigger it every 5 minutes. This is a very short interval, but it ensures that the service is checked and run repeatedly.
- *`Unit=tolga.service`*: This tells the timer which service to start — in this case, the service named `tolga.service`.

Why is this useful?

My timer ensures that the Flatpak update process runs automatically in the background, without needing manual intervention. This is a hands-free way of making sure that your Flatpak apps are always up to date.

Service Unit: `tolga.service`

Create file: ✔ `sudo nano ~/.config/systemd/user/tolga.service`

```
[Unit]
Description=Tolga's Flatpak Automatic Update and Notification VER:2.0A
Documentation=man:flatpak(1)
Wants=network-online.target
After=network-online.target

[Service]
Type=oneshot
ExecCondition=/bin/bash -c '[[ "$(busctl get-property
org.freedesktop.NetworkManager /org/freedesktop/NetworkManager
org.freedesktop.NetworkManager Metered | cut -c 3-)" == @(2|4) ]]'
ExecStart=/usr/bin/flatpak --system uninstall --unused -y --noninteractive ;
/usr/bin/flatpak --system update -y --noninteractive ; /usr/bin/flatpak --system
repair ; /usr/bin/notify-send "Flatpaks Updated" "Your computer is ready!" --app-
name="Flatpak Update Service" -i /usr/local/bin/LinuxTweaks/images/LinuxTweak.png
-u NORMAL

TimeoutStopFailureMode=abort

Environment=SYSTEMD_SLEEP_FREEZE_USER_SESSIONS=0
```

This service contains the commands to update and notify you about Flatpak updates.

- *`Type=oneshot`*: The service runs once, completes its task, and then stops.
- *`ExecCondition`*: This checks whether the network is metered (like on mobile networks) and prevents updates from running if the connection is metered.
- *`ExecStart`*: This is where the updates actually happen:
    - It uninstalls unused Flatpak apps.
    - It updates all Flatpak apps.
    - It repairs any issues with Flatpak installations.
    - It sends a notification to let you know that everything has been updated.
- *`TimeoutStopFailureMode=abort`*: If the service doesn't stop correctly, it will abort and help you troubleshoot.
- *`Environment=SYSTEMD_SLEEP_FREEZE_USER_SESSIONS=0`*: This ensures the service won't be interrupted during sleep mode.

Why is this useful?

My service automates the process of cleaning up and updating Flatpaks while also sending a notification so you know it's done. This helps ensure that your system is always running the latest versions of Flatpak apps without you having to do anything.

To activate and enable the timer nd service Instructions

Here's how you can set this up and get everything running:

1. *Reload systemd*:

```
systemctl --user daemon-reexec
```

2. *Reload the systemd configuration:*

```
systemctl --user daemon-reload
```

3. *Enable and start the timer: This will enable the timer and start it immediately:*

```
systemctl --user enable --now tolga.timer
```

4. *Manually start the service (optional): If you want to trigger the service right away, you can start it manually:*

```
systemctl --user start tolga.service
```

Why would you use this?

I believe this is a great setup for automatically keeping your Flatpaks up to date without any manual intervention. It runs in the background, ensuring your system is always up to date with minimal effort from you.

Bonus tip: the following is a AIO way to get it up and running

```
systemctl --user daemon-reexec
systemctl --user daemon-reload
systemctl --user enable --now tolga.timer
systemctl --user start tolga.service
```