🏠 / **radarr** / **system**

Radarr System

Table of Contents

- ▸ [More Info](#)

- ▸ [Tasks](#)

  - ▸ [Scheduled](#)

  - ▸ [Queue](#)

- ▸ [Backup](#)

- ▸ [Updates](#)

- ▸ [Events](#)

- ▸ [Log Files](#)

Status

Health

- ▸ This page contains a list of health checks errors. These health checks are periodically performed performed by Radarr and on certain events. The resulting warnings and errors are listed here to give advice on how to resolve them.

System Warnings

Branch is not a valid release branch

- ▸ The branch you have set is not a valid release branch. You will not receive updates. Please change to one of the [current release branches](#).

Update to .NET version

- ▸ Newer versions of Radarr are targeted for .NET6 or newer. Mono builds are not provided nor supported starting with v4. v3.2.2 is the last version of Radarr to support legacy mono builds. You are running one of these legacy mono builds, but your platform supports .NET.

See the below entries for how to switch from unsupported, end-of-life mono versions to dotnet.

> Fixing Docker installs

> Fixing FreeBSD installs

> Fixing Standalone installs

Fixing Docker installs

- ▸ Ensure your branch is correct for your docker maintainer and repull your container

Fixing FreeBSD installs

▶ Simply update the Radarr Port with `pkg update && pkg upgrade`

▶ (Optional) Remove the mono package if you wish

Fixing Standalone installs

Errors such as:

```
1 │ Cannot open assembly '/opt/Radarr/Radarr': File does not contain a valid CIL imag
```

▶ Back-Up your existing configuration before the next step.

▶ This should only happen on Linux hosts. Do not install .NET runtime or SDK from Microsoft.

▶ To remedy, download the correct build for your architecture and replace your existing binaries (application)

▶ In short you will need to delete your existing binaries (contents or folder of /opt/Radarr) and replace with the contents of the .tar.gz you just downloaded and then update your service file to not use mono.

> ⚠ DO NOT JUST EXTRACT THE DOWNLOAD OVER THE TOP OF YOUR EXISTING BINARIES.
> YOU MUST DELETE THE OLD ONES FIRST.

▶ The below is a community developed script to remove your mono installation and replace it with the .NET installation. Contributions and corrections are welcome.

▶ This assumes you are on the `master` Radarr branch, so update the variable if needed

▶ This assumes that Radarr runs as the user `radarr`, so update the variables if needed

▶ This assumes Radarr is installed at `/opt/Radarr`, so update the variables if needed

```bash
#!/bin/bash
## User Variables
installdir="/opt/Radarr"
APPUSER="radarr"
branch="master"
## /User Variables
app="radarr"
ARCH=$(dpkg --print-architecture)
# Stop \*arr
sudo systemctl stop $app
# get arch
dlbase="https://$app.servarr.com/v1/update/$branch/updatefile?os=linux&runtime=ne
case "$ARCH" in
"amd64") DLURL="${dlbase}&arch=x64" ;;
15
```

```
16    "armhf") DLURL="${dlbase}&arch=arm" ;;
17    "arm64") DLURL="${dlbase}&arch=arm64" ;;
18    *)
19        echo_error "Arch not supported"
20        exit 1
21        ;;
22    esac
23    echo "Downloading..."
24    wget --content-disposition "$DLURL"
25    tar -xvzf ${app^}.*.tar.gz
26    echo "Installation files downloaded and extracted"
27    echo "Moving existing installation"
28    sudo mv "$installdir/" "$installdir.old/"
29    echo "Installing..."
30    sudo mv "${app^}" "$installdir"
31    sudo chown $APPUSER:$APPUSER -R $installdir
32    sudo sed -i "s|ExecStart=/usr/bin/mono --debug /opt/${app^}/${app^}.exe|ExecStart
33    sudo sed -i "s|ExecStart=/usr/bin/mono /opt/${app^}/${app^}.exe|ExecStart=/opt/${
34    sudo systemctl daemon-reload
35    echo "App Installed"
36    sudo rm -rf "$installdir.old/"
37    rm -rf "${app^}.*.tar.gz"
      sudo systemctl start $app
```

Currently installed mono version is old and unsupported

▶ Radarr is written in .NET and required Mono to run on very old ARM processors. Mono 5.20 is the absolute minimum for Radarr.

▶ The upgrade procedure for Mono varies per platform.

> ⚠ Mono is no longer supported starting in Radarr version 4.0

Currently installed SQLite version is not supported

▶ Radarr stores its data in an SQLite database. The SQLite3 library installed on your system is too old. Radarr requires at least version 3.9.0.

> ℹ Note that Radarr uses `libSQLite3.so` which may or may not be contained in a SQLite3 upgrade package.

Database Failed Integrity Check

▶ Your database(s) failed a SQLite Pragma Integrity Check ↗ and have some corruption.

▶ If `Radarr.db` is corrupt please see this FAQ Entry

▸ If `logs.db` is corrupt: Stop Radarr, delete `logs.db` and any `logs.wal` files.

▸ If both are corrupt, review the respective processes above.

New update is available

▸ Rejoice, the developers have released a new update. This generally means awesome new features and squashed piles of bugs (right?). Apparently you don't have Auto-Updating enabled, so you'll have to figure out how to update on your platform. Pressing the Install button on the System => Updates page is probably a good starting point.

> ℹ   This warning will not appear if your current version is less than 14 days old

Cannot install update because startup folder is not writable by the user

▸ This means Radarr will be unable to update itself. You'll have to update Radarr manually or set the permissions on Radarr's Startup directory (the installation directory) to allow Radarr to update itself.

Updating will not be possible to prevent deleting AppData on Update

▸ Radarr detected that AppData folder for your Operating System is located inside the directory that contains the Radarr binaries. Normally it would be C:\ProgramData for Windows and, ~/.config for linux.

▸ Please look at System => Info to see the current AppData & Startup directories.

▸ This means Radarr will be unable to update itself without risking data-loss.

▸ If you're on linux, you'll probably have to change the home directory for the user that is running Radarr and copy the current contents of the ~/.config/Radarr directory to preserve your database.

Branch is for a previous version

▸ The update branch setup in Settings/General is for a previous version of Radarr, therefore the instance will not see correct update information in the System/Updates feed and may not receive new updates when released.

Could not connect to signalR

▸ signalR drives the dynamic UI updates, so if your browser cannot connect to signalR on your server you won't see any real time updates in the UI.

▸ The most common occurrence of this is use of a reverse proxy or cloudflare

▸ Cloudflare needs websockets enabled.

Nginx

▸ Nginx requires the following addition to the location block for the app:

```
1   proxy_http_version 1.1;
2   proxy_set_header Upgrade $http_upgrade;
3   proxy_set_header Connection $http_connection;
```

> ⚠️ Make sure you do not include proxy_set_header Connection "Upgrade"; as suggested by the nginx
>   documentation. THIS WILL NOT WORK
>   See https://github.com/aspnet/AspNetCore/issues/17081 ↗

Apache2

For Apache2 reverse proxy, you need to enable the following modules: proxy, proxy_http, and proxy_wstunnel. Then, add this websocket tunnel directive to your vhost configuration:

```
1   RewriteEngine On
2   RewriteCond %{HTTP:Upgrade} =websocket [NC]
3   RewriteRule /(.*) ws://127.0.0.1:7878/$1 [P,L]
```

If you have a reverse proxy under a subdirectory, the RewriteRule should include your basepath e.g.

```
1   RewriteRule /radarr/(.*) ws://127.0.0.1:7878/radarr/$1 [P,L]
```

If Radarr is not running on the same machine as your reverse proxy. Replace 127.0.0.1 with the appropriate IP address/DNS name of your Radarr app.

Caddy

For Caddy (V1) use this:
Note: you will also need to add the websocket directive to your radarr configuration

```
1   proxy /radarr 127.0.0.1:7878 {
2       websocket
3       transparent
4   }
```

Failed to resolve the IP Address for the Configured Proxy Host

- ▶ Review your proxy settings and ensure they are accurate
- ▶ Ensure your proxy is up, running, and accessible

Proxy Failed Test

- ▶ Your configured proxy failed to test successfully, review the HTTP error provided and/or check logs for more details.

System Time is off by more than 1 day

- ▶ System time is off by more than 1 day. Scheduled tasks may not run correctly until the time is corrected

- ▶ Review your system time and ensure it is synced to an authoritative time server and accurate

PTP Indexer Settings Out of Date

- ▶ The following PassThePopcorn indexers have deprecated settings and should be updated.

Mono and x86 builds are ending

- ▶ Mono and x86 builds are no longer be supported with v4. If you are receiving this error then you are running the mono version of the application or the x86 version. Unfortunately, due to increasing difficulty in development support for these legacy versions we will be discontinuing their support and thus releases for them going forward. Thus it is advised you upgrade to a supported Operating System that does not require neither x86 nor mono. You may also be able to explore using Docker for your needs.

Download Clients

No download client is available

- ▶ A properly configured and enabled download client is required for Radarr to be able to download media. Since Radarr supports different download clients, you should determine which best matches your requirements. If you already have a download client installed, you should configure Radarr to use it and create a category. See Settings => Download Client.

Unable to communicate with download client

- ▶ Radarr was unable to communicate with the configured download client. Please verify if the download client is operational and double check the url. This could also indicate an authentication error.

- ▶ This is typically due to improperly configured download client. Things you can typically check:

  - ▶ Your download clients IP Address if its on the same bare metal machine this is typically 127.0.0.1

  - ▶ The Port number of that your download client is using these are filled out with the default port number but if you've changed it you will need to have the same one entered into Radarr.

  - ▶ Ensure SSL encryption is not turned on if you're using both your Radarr instance and your download client on a local network. See the SSL FAQ entry for more information.

  - ▶ Ensure IPv6 is disabled on the system if it is not functional

  - ▶ Ensure a DNS server (e.g. pihole) is not rate limiting queries

Download clients are unavailable due to failure

- ▶ One or more of your download clients is not responding to requests made by Radarr. Therefore Radarr has decided to temporarily stop querying the download client on it's normal 1 minute cycle, which is normally used to track active downloads and import finished ones. However, Radarr will continue to attempt to send downloads to the client, but will in all likeliness fail.

- ▶ You should inspect System=>Logs to see what the reason is for the failures.

- ▶ If you no longer use this download client, disable it in Radarr to prevent the errors.

Enable Completed Download Handling

▸ Radarr requires Completed Download Handling to be able to import files that were downloaded by the download client. It is recommended to enable Completed Download Handling. (Completed Download Handling is enabled by default for new users.)

Docker bad remote path mapping

▸ This error is typically associated with bad docker paths within either your download client or Radarr

▸ An example of this would be:

  ▸ Download client: Download Path: /mnt/user/downloads:/downloads

  ▸ Radarr: Download Path: /mnt/user/downloads:/data

▸ Within this example the download client places its downloads into /downloads and therefore tells Radarr when its complete that the finished movie is in /downloads. Radarr then comes along and says "Okay, cool, let me check in /downloads" Well, inside Radarr you did not allocate a /downloads path you allocated a /data path so it throws this error.

▸ The easiest fix for this is CONSISTENCY if you use one scheme in your download client, use it across the board.

▸ Team Radarr is a big fan of simply using /data.

  ▸ Download client: /mnt/user/data/downloads:/data/downloads

  ▸ Radarr: /mnt/user/data:/data

▸ Now within the download client you can specify where in /data you'd like to place your downloads, now this varies depending on the client but you should be able to tell it "Yeah download client place my files into." /data/torrents (or usenet)/movies and since you used /data in Radarr when the download client tells Radarr it's done Radarr will come along and say "Sweet, I have a /data and I also can see /torrents (or usenet)/movies all is right in the world."

▸ There are many great write ups: our wiki [Docker Guide](#) and TRaSH's [Hard links and Instant Moves (Atomic-Moves)](#) ☒ . Now these guides place heavy emphasis on Hard links and Atomic moves, but the general concept of containers and how path mapping works is the core of these discussions.

▸ See [TRaSH's Remote Path Guide](#) ☒ for more information.

▸ For Linux ensure:

  ▸ If you're using an NFS mount ensure `nolock` is enabled for your mount.

  ▸ If you're using an SMB mount ensure `nobrl` is enabled for your mount.

Downloading into Root Folder

▸ Within the application, a root folder is defined as the configured media library folder. This is not the root folder of a mount. Your download client has an incomplete or complete (or is moving completed downloads) into your root (library) folder.

▸ This frequently causes issues - including data loss - and should not be done. To fix this change your download client so it is not placing downloads within your root folder. Note that 'placing' also includes if your download

client category is set to your root folder or if NZBGet/SABnzbd have sort enabled and are sorting to your root folder.

▶ Please note that this check looks at all defined/configured root folders added not only root folders currently in use. In other words, the folder your download client downloads into or moves completed downloads to, should not be the same folder you have configured as your root/library/final media destination folder in the *arr application.

▶ Configured Root Folders (aka Library folders) can be found in [Settings => Media Management => Root Folders](#)

▶ One example is if your downloads are going into `\data\downloads` then you have a root folder set as `\data\downloads`.

▶ It is suggested to use paths like `\data\media\` for your root folder/library and `\data\downloads\` for your downloads.

▶ Review our [Docker Guide](#) and TRaSH's [Hard links and Instant Moves (Atomic-Moves) Guide](#) ⧉ for more information on the correct and optimal path setup. Note that the concepts apply for docker and non-docker

> ⚠️ **Your download folder where your download client places the downloads and your root/library folder MUST be separate. *Arr will import the file(s) from your download client's folder into your library. The download client should not move anything or download anything to your library.**

Bad Download Client Settings

▶ The location your download client is downloading files to is causing problems. Check the logs for further information. This may be permissions or attempting to go from windows to linux or linux to windows without a remote path map.

Bad Remote Path Mapping

▶ The location your download client is downloading files to is causing problems. Check the logs for further information. This may be permissions or attempting to go from windows to linux or linux to windows without a remote path map. See [TRaSH's Remote Path Guide](#) ⧉ for more information.

Permissions Error

▶ Radarr or the user radarr is running as cannot access the location your download client is downloading files to. This is typically a permission issue.

Remote File was removed part way through processing

▶ A file accessible via a remote path map appears to have been removed prior to processing completing.

Remote Path is Used and Import Failed

▶ Check your logs for more info; Refer to our Troubleshooting Guides

Completed/Failed Download Handling

Completed Download Handling is disabled

▶ (This warning is only generated for existing users before when the Completed Download Handling feature was implemented. This feature is disabled by default to ensure the system continued to operate as expected for current configurations.)

▶ It's recommended to use Completed Download Handling since it provides better compatibility for the unpacking and post-processing logic of various download clients. With it, Radarr will only import a download once the download client reports it as ready.

▶ If you wish to enable Completed Download Handling you should verify the following: * Warning: Completed Download Handling only works properly if the download client and Radarr are on the same machine since it gets the path to be imported directly from the download client otherwise a remote map is needed.

Download Client Removes Completed Downloads

▶ It's required that your download client retain its history of completed downloads until Radarr has imported them. If history retention is disabled then *Arr may not see the completed download before it is removed from the download client. Your download client should be set to keep (usenet) and pause not remove (torrents) downloads after completion: **either indefinitely or for at least 14 days**.

    ▶ Sabnzbd: Switches => Post Processing => Keep Jobs **must** be set to 14 days or greater OR be set to Keep All History

▶ Removing completed downloads from your client can be managed by Radarr and enabled via the download client settings in *Arr. Thus *Arr can ensure that your download client history is cleaned up.

Indexers

No indexers available with automatic search enabled, Radarr will not provide any automatic search results

▶ Simply put you do not have any of your indexers set to allow automatic searches

▶ Go into Settings => Indexers, select an indexer you'd like to allow Automatic Searches and then click save.

No indexers available with RSS sync enabled, Radarr will not grab new releases automatically

▶ Radarr uses the RSS feed to pick up new releases as they come along. More info on that here

▶ To correct this issue go to Settings => Indexers, select an indexer you have and enable RSS Sync

No indexers are enabled

▶ Radarr requires indexers to be able to discover new releases. Please read the wiki on instructions how to add indexers.

Enabled indexers do not support searching

▶ None of the indexers you have enabled support searching. This means Radarr will only be able to find new releases via the RSS feeds. But searching for movies (either Automatic Search or Manual Search) will never return any results. Obviously, the only way to remedy it is to add another indexer.

No indexers available with Interactive Search Enabled

▶ None of the indexers you have enabled support interactive searching. This means the application will only be able to find new releases via the RSS feeds or an automatic search.

Indexers are unavailable due to failures

▶ Errors occurs while Radarr tried to use one of your indexers. To limit retries, Radarr will not use the indexer for an increasing amount of time (up to 24h).

▶ This mechanism is triggered if Radarr was unable to get a response from the indexer (could be caused DNS, proxy/vpn connection, authentication, or an indexer issue), or unable to fetch the nzb/torrent file from the indexer. Please inspect the logs to determine what kind of error caused the problem.

▶ You can prevent the warning by disabling the affected indexer.

▶ Run the Test on the indexer to force Radarr to recheck the indexer, please note that the Health Check warning will not always disappear immediately.

Jackett All Endpoint Used

▶ The Jackett /all endpoint is convenient, but that is its only benefit. Everything else is potential problems, so adding each tracker individually is now required.

▶ [Even Jackett's Devs says it should be avoided and should not be used.](#) ⧉

▶ Using the /all endpoint has no advantages, only disadvantages:

  ▶ you lose control over indexer specific settings (categories, search modes, etc.)

  ▶ mixing search modes (IMDB, query, etc.) might cause low-quality results

  ▶ indexer specific categories (>= 100000) cannot be used.

  ▶ slow indexers will slow down the overall result

  ▶ total results are limited to 1000

  ▶ if one of the trackers in /all returns an error, *Arr will disable it and now you do not get any results.

Solutions

▶ Add each tracker in Jackett manually as an indexer in *Arr

▶ Check out [Prowlarr](#) which can sync indexers to *Arr and from the Lidarr/Radarr/Readarr development team.

▶ Check out [NZBHydra2](#) ⧉ which can sync indexers to *Arr. But do not use their single aggregate endpoint and use `multi` if sync will be used.

Movie Folders

Missing Root Folder

▶ This error is typically identified if a Movie or Collection is assigned a root folder but that root folder is no longer available.

▶ This error may also be if a list is still pointed at a root folder and that root folder is no longer available.

▶ If you would like to remove this warning simply find the Movie(s) or Collection(s) that is(are) still using the old root folder and edit it to the correct root folder.

Movies Table View

1. Go to the Movies (Library) Tab

2. Table View & Enable the Path column then sort by path

Movies Custom Filter

1. Create a custom filter with the old root folder path

2. Select mass edit on the top bar and from the Root Paths drop down select the new root path that you want these movie to be moved to.

3. Next you will receive a pop-up that states "Would you like to move the Movie folders to 'root path'?" This will also state This will also rename the Movie folder per the Movie folder format in settings. Simply select No if the you do not want Radarr to move your files

4. Run the Check Health Task in System => Tasks

Collections Custom Filter

1. Create a custom filter in Collections with the old root folder path

2. Select the collections and from the Root Paths drop down select the new root path that you want these collections' future movies to be assigned to.

3. Run the Check Health Task in System => Tasks

Movie Path Mount is Read Only

A mount containing a movie path is read only and is not writable by the user Radarr is running as.

Movies

Movie was removed from TMDb

▶ The movie is linked to a TMDb Id that was deleted from TMDb, usually because it was a duplicate, wasn't a movie or changed ID for some other reason. Deleted movies will not receive any updates and should be corrected by the user to ensure continued functionality. Remove the movie from Radarr without deleting the files, then try to re-add it. If it doesn't show up in a search, check Sonarr because it might be a TV miniseries like Stephen King's It.

▶ You can find and edit deleted movies by creating a custom filter using the following steps:

   1. Click Movies from the left menu

   2. Click the dropdown on Filter and select "Custom Filter"

   3. Enter a label, for example "Deleted Movies"

   4. Make the filter as follows: `Release Status` is `Deleted`

   5. Click save and select the newly created filter from the filter dropdown menu

Lists are unavailable due to failures

▶ Typically this simply means that Radarr is no longer able to communicate via API or via logging in to your chosen list provider. Your best bet if the problem persists is to contact them in order to rule them out, as their systems maybe overloaded from time to time.

▶ Review System => Events filtered for Warning (Warning & Errors) to see the historical failures or check logs for details.

Notifications

Discord as Slack Notification

▶ You have Discord configured to use a Slack webhook. This is not advised and the native Discord functionality should be used instead. If you're receiving this that likely means you just configured this by following an outdated guide. Please advise the guide author to update their steps.

Disk Space

▶ This section will show you available disk space

▶ In docker this can be tricky as it will typically show you the available space within your Docker image

About

▶ This will tell you about your current install of Radarr

More Info

▶ Home Page: Radarr's home page

▶ Wiki: You're here already

▶ Reddit: r/radarr

▶ Discord: Join our discord

▶ Donations: If you're feeling generous and would like to donate click here

▶ Donations to Sonarr: If you're feeling generous and would like to donate to the project that started it all click here

▶ Source: GitHub

▶ Feature Requests: Got a great idea drop it here

Tasks

Scheduled

▶ This section lists all scheduled tasks that Radarr runs

▶ Application Check Update - This will run every on the displayed schedule in the UI, checking to see if Radarr is on the most current version then triggering the update script to update Radarr. Settings=> Update

⚠   Note: If on Docker this will not update your container as a new image will need to be downloaded.

▶ Backup - This will run a backup of your Radarr's database on a set schedule more details on this can be found here. More information about backups can be found System => Backups.

▶ Check Health - Check Health will run on the displayed schedule in the UI checking the overall health of your Radarr. To see a list of possible health related issues see the Wiki Entry on Health Checks.

▶ Clean Up Recycle Bin - The recycling bin will be cleared out on the displayed schedule in the UI. This will only be used if the recycling bin is set in File Management

▶ Housekeeping - On the displayed schedule in the UI this will dust out all the cobwebs, sweeps and vacuums the floors, mops, shines, and even makes nice neat little folded notes just for you. But does not take out the trash. That it just was not paid enough for.

▶ Import List Sync - On the displayed schedule in the UI this will run your Lists and import any possible new movies. More info about lists can be found Settings => Lists.

▶ Messaging Cleanup - On the displayed schedule in the UI this cleans up those messages that appear in the bottom left corner of Radarr

▶ Refresh Monitored Downloads - This goes through and refreshes the downloads queue located under Activity. Essentially pinging your download client to check for finished downloads.

▶ Refresh Movie - This goes through and refreshes all the metadata for all monitored and unmonitored movies

▶ RSS Sync - This will run the RSS Sync. This can be changed in settings => options. More information on the RSS function can be found on our FAQ

ℹ   All these tasks can be ran manually outside their scheduled times by hitting the icon to the far right of each of the tasks.

Queue

▶ The queue will show you running and upcoming tasks as well as a history of recently ran tasks as well as how long those tasks took.

Backup

ℹ   If you're looking for how to back/restore your Radarr instance click here.

- ▶ Within the Backup section you will be presented with previous backups (unless you have a fresh install that hasn't made any backups).

- ▶ Backup Now - This option will trigger a manual backup of your Radarr's database

- ▶ Restore Backup - This will open a new screen to restore from a previous backup

  - ▶ By selecting Choose File this will prompt your browser to open a dialog box to restore from a Radarr Zip backup

- ▶ If you have any previous backups and would like to download them from Radarr to be placed in a non standard location you simply can select one of these files to download them

- ▶ Off to the right of each of the previous download you have two options.

  - ▶ Restore (Clock Icon) - To restore from a previous backup - This will open a new window to confirm you want to restore from this backup

  - ▶ Delete (Trashcan) - To delete a previous backup

Updates

- ▶ The update screen will show the past 5 updates that have been made as well as the current version you are on.

- ▶ This page will also display the update notes from the Developers telling you what has been fixed or added to Radarr (Rejoice!)

> ℹ️ A Maintenance Release contains bug fixes and other various improvements. Take a look at the commit history for specifics.

Events

- ▶ The events tab will show you what has been happening within your Radarr. This can be used to diagnose some light issues. However, this does not replace Trace Logs discussed in Logging.

> ℹ️ Events are the equivalent of INFO Logs.

- ▶ Components - This column will tell you what component within Radarr has been triggered

- ▶ Message - This column will tell you what message as been sent from the component from the previous column.

- ▶ Gear Icon - This option will allow you to change how many Components/Messages are displayed per page (Default is 50)

- ▶ Options at the top of the page

▸ Refresh - This option will refresh the current page, pulling a new events log

▸ Clear - This will clear the current events log allowing you to start from fresh

Log Files

▸ This page will allow you to download and see what current log files are available for Radarr

▸ On the top row there are several options to allow you to control your log files.

▸ The top row on the far left there is a dropdown that will allow you to switch from Log files and Updater Log Files

  ▸ Log Files - The bread and butter of any support issue more on log files can be found here.

  ▸ Updater Log Files - This will show the log files associated with Radarr's updater script

> ⓘ   If you're on docker this will be empty as you should be updating by downloading a new docker image

▸ Refresh - This will refresh the current page and display any newly created logs

▸ Delete - This will clear all logs allowing you to start from fresh

▸ File Name - This will display the file name associated with the log

▸ Last Written - This is the local time that this particular log file was written to.

  ▸ Radarr uses rolling log files limited to 1MB each. The current log file is always radarr.txt, for the the other files radarr.0.txt is the next newest (the higher the number the older it is) up to 51 log files total. This log file contains `fatal`, `error`, `warn`, and `info` entries.

  ▸ When Debug log level is enabled, additional radarr.debug.txt rolling log files will be present, up to 51 files. This log files contains `fatal`, `error`, `warn`, `info`, and `debug` entries. It usually covers a ~40h period.

  ▸ When Trace log level is enabled, additional radarr.trace.txt rolling log files will be present, up to 51 files. This log files contains `fatal`, `error`, `warn`, `info`, `debug`, and `trace` entries. Due to trace verbosity it only covers a couple of hours at most.

Powered by Wiki.js