

# Lidarr System

System information, logs, scheduled tasks, and status monitoring for Lidarr administration and troubleshooting

## Table of Contents

---

- ▶ [Table of Contents](#)
- ▶ [Status](#)
- ▶ [Health](#)
  - ▶ [System Warnings](#)
    - ▶ [Branch is not a valid release branch](#)
    - ▶ [Update to .NET version](#)
      - ▶ [Fixing Docker installs](#)
      - ▶ [Fixing Standalone installs](#)
    - ▶ [Currently installed mono version is old and unsupported](#)
    - ▶ [Currently installed SQLite version is not supported](#)
    - ▶ [New update is available](#)
    - ▶ [Cannot install update because startup folder is not writable by the user](#)
    - ▶ [Updating will not be possible to prevent deleting AppData on Update](#)
    - ▶ [Branch is for a previous version](#)
    - ▶ [Could not connect to signalR](#)
      - ▶ [NGINX](#)
      - ▶ [Apache](#)
      - ▶ [Caddy](#)
    - ▶ [Failed to resolve the IP Address for the Configured Proxy Host](#)
    - ▶ [Proxy Failed Test](#)
    - ▶ [System Time is off by more than 1 day](#)
    - ▶ [Mono Legacy TLS enabled](#)
    - ▶ [Mono and x86 builds are ending](#)
    - ▶ [FPcalc needs updating](#)
  - ▶ [Download Clients](#)
    - ▶ [No download client is available](#)

- ▶ [Unable to communicate with download client](#)
- ▶ [Download clients are unavailable due to failure](#)
- ▶ [Enable Completed Download Handling](#)
- ▶ [Docker bad remote path mapping](#)
- ▶ [Downloading into Root Folder](#)
- ▶ [Bad Download Client Settings](#)
- ▶ [Bad Remote Path Mapping](#)
- ▶ [Permissions Error](#)
- ▶ [Remote File was removed part way through processing](#)
- ▶ [Remote Path is Used and Import Failed](#)
- ▶ [Completed/Failed Download Handling](#)
  - ▶ [Completed Download Handling is disabled](#)
  - ▶ [Download Client Removes Completed Downloads](#)
- ▶ [Indexers](#)
  - ▶ [No indexers available with automatic search enabled, Lidarr will not provide any automatic search results](#)
  - ▶ [No indexers available with RSS sync enabled, Lidarr will not grab new releases automatically](#)
  - ▶ [No indexers are enabled](#)
- ▶ [Enabled indexers do not support searching](#)
  - ▶ [No indexers Available with Interactive Search Enabled](#)
  - ▶ [Indexers are unavailable due to failures](#)
  - ▶ [Jackett All Endpoint Used](#)
    - ▶ [Solutions](#)
- ▶ [Artist Folders](#)
  - ▶ [Missing Root Folder](#)
  - ▶ [Lists are unavailable due to failures](#)
- ▶ [Disk Space](#)
- ▶ [About](#)
- ▶ [More Info](#)
- ▶ [Tasks](#)
  - ▶ [Scheduled](#)
  - ▶ [Queue](#)
- ▶ [Backup](#)
- ▶ [Updates](#)

- ▶ [Events](#)
- ▶ [Log Files](#)



Warning: This page is a work in progress and is currently a rough draft based on Readarr

# Status

---

## Health

---

This page contains a list of health checks errors. These health checks are periodically performed performed by Lidarr 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 Lidarr are targeted for .NET6 or newer. We will no longer be providing legacy mono builds after 1.0 is released. You are running one of these legacy builds but your platform supports .NET.

#### Fixing Docker installs

---

- ▶ Re-pull your container

#### Fixing Standalone installs

---

- ▶ 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/Lidarr) and replace with the contents of the .tar.gz you just downloaded.



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** Lidarr branch update the variable if needed
- ▶ This assumes that Lidarr runs as the user **lidarr** update the variables if needed
- ▶ This assumes Lidarr is installed at **/opt/Lidarr** update the variables if needed

```

1  #!/bin/bash
2  ## User Variables
3  installdir="/opt/Lidarr"
4  APPUSER="lidarr"
5  branch="master"
6  ## /User Variables
7  app="lidarr"
8  ARCH=$(dpkg --print-architecture)
9  # Stop \*arr
10 sudo systemctl stop $app
11 # get arch
12 dlbase="https://$app.servarr.com/v1/update/$branch/updatefile?os=linux&runtime=ne
13 case "$ARCH" in
14 "amd64") DLURL="${dlbase}&arch=x64" ;;
15 "armhf") DLURL="${dlbase}&arch=arm" ;;
16 "arm64") DLURL="${dlbase}&arch=arm64" ;;
17 *)
18     echo_error "Arch not supported"
19     exit 1
20     ;;
21 esac
22 echo "Downloading..."
23 wget --content-disposition "$DLURL"
24 tar -xvzf "${app^}.*.tar.gz"
25 echo "Installation files downloaded and extracted"
26 echo "Moving existing installation"
27 sudo mv "$installdir/" "$installdir.old/"
28 echo "Installing..."
29 sudo mv "${app^}" "$installdir"
30 sudo chown $APPUSER:$APPUSER -R $installdir
31 sudo sed -i "s|ExecStart=/usr/bin/mono --debug /opt/${app^}/${app^}.exe|ExecStart
32 sudo sed -i "s|ExecStart=/usr/bin/mono /opt/${app^}/${app^}.exe|ExecStart=/opt/${
33 sudo systemctl daemon-reload
34 echo "App Installed"
35 sudo rm -rf "$installdir.old/"
36 rm -rf "${app^}.*.tar.gz"
37 sudo systemctl start $app

```

## Currently installed mono version is old and unsupported

- ▶ Lidarr is written in .NET and requires Mono to run on very old ARM processors. Please note that Mono builds are no longer supported after v1.0
- ▶ Mono 5.20 is the absolute minimum for Lidarr.
- ▶ The upgrade procedure for Mono varies per platform.

### Currently installed SQLite version is not supported

- ▶ Lidarr stores its data in an SQLite database. The SQLite3 library installed on your system is too old. Lidarr requires at least version 3.9.0. Note that Lidarr uses `libSQLite3.so` which may or may not be contained in a SQLite3 upgrade package.

### 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.



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

### Updating will not be possible to prevent deleting AppData on Update

- ▶ Lidarr detected that AppData folder for your Operating System is located inside the directory that contains the Lidarr 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 Lidarr 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 Lidarr and copy the current contents of the `~/.config/Lidarr` directory to preserve your database.

### Branch is for a previous version

- ▶ The update branch setup in `Settings => General` is for a previous version of Lidarr, 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> [↗](#)

## Apache

---

- ▶ 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://127.0.0.1:8686/$1 [P,L]
```

## Caddy

---

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

```
1 | proxy /lidarr 127.0.0.1:8686 {  
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

## Mono Legacy TLS enabled

- ▶ Mono 4.x tls workaround still enabled, consider removing `MONO_TLS_PROVIDER=legacy` environment option

## Mono and x86 builds are ending

- ▶ Mono and x86 builds will no longer be supported in the next build of the application. If you are receiving this error then you are running the mono version of the application or the x86 version. Unfortunately, due to increasing difficulting 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.

## FPcalc needs updating

- ▶ Lidarr uses chromaprint audio fingerprinting to identify tracks. This depends on an external binary `fpcalc`, which is distributed with Lidarr v1 for Windows, Linux and macOS, but must be provided independently on freeBSD.
- ▶ Ensure the fpcalc binary bundled with Lidarr is executable (755 permissions). This would be found in Lidarr's installation directory (e.g. `/opt/Lidarr/fpcalc`) If it is not executable, then correct it's permissions with the below command and then restart Lidarr.
- ▶ Note that for the fix `sudo` may be required or your path to Lidarr's binary folder may be different depending on your environment and setup.

```
1 | chmod +x /opt/Lidarr/fpcalc
```



The below information is for legacy v0.8.0 builds only.

- ▶ To fix this on a native Linux instance, install the appropriate package using your package manager and make sure that fpcalc is on your PATH (this can be checked using `which fpcalc` and verifying that the correct location of fpcalc is returned):

Distribution	Package
Debian/Ubuntu	libchromaprint-tools
Fedora/CentOS	chromaprint-tools

Distribution	Package
Arch	chromaprint
OpenSUSE	chromaprint-fpcalc
Synology	chromaprint

## Download Clients

### No download client is available

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

### Unable to communicate with download client

- ▶ Lidarr was unable to communicate with the configured download client. Please verify 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 client's IP Address - if it's all on the same bare metal machine, this is typically **127.0.0.1**
  - ▶ The Port number 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 Lidarr.
  - ▶ Ensure that SSL encryption is not turned on if you're using both your Lidarr instance and your download client on a local network (i.e., over plain HTTP). See the SSL FAQ entry for more information.

### Download clients are unavailable due to failure

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

### Enable Completed Download Handling

- ▶ Lidarr 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 Lidarr
- ▶ An example of bad (inconsistent) paths would be:
  - ▶ Download client: `/mnt/user/downloads:/downloads`
  - ▶ Lidarr: `/mnt/user/downloads:/data`
- ▶ In this example the download client places its downloads into `/downloads` and tells Lidarr when its complete that the finished book is in `/downloads`. Lidarr then comes along and says "Okay, cool, let me check in `/downloads`." Well, inside Lidarr 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 Lidarr is a big fan of simply using `/data`.
  - ▶ Download client: `/mnt/user/data/downloads:/data/downloads`
  - ▶ Lidarr: `/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/downloads/movies`" and since you used `/data` in Lidarr when the download client tells Lidarr it's done Lidarr will come along and say "Sweet, I have a `/data` and I also can see `/data/downloads/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.
- ▶ If you're crossing operating systems or native and docker then you need a remote path map. See [TRaSH's Remote Path Guide for Radarr](#) [↗](#) and [Sonarr](#) [↗](#) for more information.

## 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

---

- ▶ Lidarr (or the user lidarr 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

---

- ▶ Lidarr 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`. (It is enabled by default for new users.)

## Download Client Removes Completed Downloads

---

- ▶ It's required that your download client retain its history of completed downloads until Lidarr 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**.

- ▶ Sabnzb: 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 Lidarr 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, Lidarr 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, Lidarr will not grab new releases automatically

---

- ▶ Lidarr 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

---

- ▶ Lidarr 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 Lidarr 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 Lidarr tried to use one of your indexers. To limit retries, Lidarr will not use the indexer for an increasing amount of time (up to 24h).
- ▶ This mechanism is triggered if Lidarr 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 causes the problem.
- ▶ You can prevent the warning by disabling the affected indexer.

- ▶ Run the Test on the indexer to force Lidarr 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 ( $\geq 100000$ ) cannot be used.
  - ▶ slow indexers will slow down the overall result
  - ▶ total results are limited to 1000
  - ▶ if one of the trackers returns an error, \*Arr will disable it and now you will 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.

## Artist Folders

---

### Missing Root Folder

---

- ▶ This error is typically identified if a Artist is looking for 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 but that root folder is no longer available.
- ▶ If you would like to remove this warning simply find the album that is still using the old root folder and edit it to the correct root folder.
- ▶ Easiest way to find the problem artist is to:
  - ▶ Go to the Artist (Library) Tab
  - ▶ Create a custom filter with the old root folder path
  - ▶ Select mass edit on the top bar and from the Root Paths drop down select the new root path that you want these artist to be moved to.
  - ▶ Next you will receive a pop-up that states Would you like to move the Artist folders to 'root path' ? This will also state This will also rename the Artist folder per the Artist folder format in settings. Simply select No if the you do not want Lidarr to move your files
- ▶ Run the Check Health Task in System => Tasks

## Lists are unavailable due to failures

---

- ▶ Typically this simply means that Lidarr 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.
- ▶ Review System => Events filtered for Warning (Warning & Errors) to see the historical failures or check logs for details.

## 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 Lidarr

## More Info

---

- ▶ Home Page: Lidarr's home page
- ▶ Wiki: You're here already
- ▶ Reddit: r/lidarr
- ▶ 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 page lists all scheduled tasks that Lidarr runs
  - ▶ Application Check Update - This will run every on the displayed schedule in the UI, checking to see if Lidarr is on the most current version then triggering the update script to update Lidarr. 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 Lidarr'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 Lidarr. 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 artists. 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 Lidarr
- ▶ 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 Artist - This goes through and refreshes all the metadata for all monitored and unmonitored artists
- ▶ 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



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 upcoming tasks as well as a history of recently ran tasks as well as how long those tasks took.

## Backup



This section will be more tailored to the buttons and overall point of the page. However, if you're looking for how to back/restore your Lidarr instance click [the Lidarr backup FAQ](#).

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

Here you will have two options at the top of the screen

- ▶ Backup Now - This option will trigger a manual backup of your Lidarr'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 Lidarr Zip backup

Finally if you have any previous backups and would like to download them from Lidarr 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.

- ▶ One - To restore from a previous backup - This will open a new window to confirm you want to restore from this backup
- ▶ Two - 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 Lidarr (Rejoice!)



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 Lidarr. This can be used to diagnose some light issues. However, this does not replace Trace Logs discussed in Logging. Events are the equivalent of INFO Logs.

- ▶ Components - This column will tell you what component within Lidarr 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 Lidarr

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 Lidarr'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.
  - ▶ Lidarr uses rolling log files limited to 1MB each. The current log file is always lidarr.txt, for the the other files lidarr.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 lidarr.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 lidarr.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.