

# MediaTools Video Downloader

## Comprehensive User Guide

**Version:** 2.1.0

**Developer:** Bala

**Organization:** MediaTools

**Email:**[bala.lv.555@gmail.com](mailto:bala.lv.555@gmail.com)

**GitHub:**<https://github.com/MediaTools-tech/mediatools>

**Website:**<https://mediatools.tech>

## Table of Contents

1. Introduction
  - Key Features
  - Technology Stack
2. Installation
  - System Requirements
  - Installation Steps
  - Platform-Specific Notes
3. Getting Started
  - Launching the Application
  - Interface Overview
  - First Download
4. Downloading Videos
  - Basic Download Process
  - Queue Management
5. Audio Download Feature
  - Audio Download Overview
  - Audio Format Selection
  - Audio Quality Considerations
6. Settings and Features
  - Settings Configuration
  - Spotal/Spotify Credentials Settings
  - Performance Tips
  - Updates and Maintenance
  - Advanced Features

- 7. [FAQ and Troubleshooting](#)
    - [General Questions](#)
    - [Installation & Setup](#)
    - [Download Performance](#)
    - [Queue & Session Management](#)
    - [Format & Container Issues](#)
    - [Audio Download Questions](#)
    - [Technical Questions](#)
    - [Advanced Topics](#)
    - [Troubleshooting](#)
  - 8. [Contributing and Feedback](#)
  - 9. [Privacy and Legal Information](#)
    - [Privacy and Security](#)
    - [Legal Information](#)
  - 10. [Contact and Support](#)
  - 11. [Version History](#)
-

## 1. Introduction

MediaTools Video Downloader is a powerful, cross-platform desktop application designed to download videos and audio from popular platforms. Built with user-friendliness in mind, it provides extensive customization options while maintaining simplicity for everyday use.

### Key Features

- **Auto-update option** – Auto update to latest yt-dlp and application
- **Multi-platform support** - Download from yt-dlp supported 1000+ sites
- **Format flexibility** - Choose between MKV (highest quality) or MP4 (best compatibility)
- **Audio download support** - Extract audio in multiple formats (M4A, MP3, Best Audio)
- **Smart Format Fallback**: Automatically retries downloads with alternative formats if the preferred one fails
- **Quality control** - Select optimal quality settings for your needs
- **Speed management** - Control download speeds to manage bandwidth
- **Queue system** - Download multiple videos/audio sequentially
- **Thumbnail support** - Embed thumbnails for easy identification (videos and audio with cover art)
- **Cookie integration** - Access private or membership-only content
- **Cross-platform** - Works on Windows, Linux, and macOS

### Technology Stack

This application leverages industry-standard open-source tools:

- **yt-dlp** (<https://github.com/yt-dlp/yt-dlp>) - Advanced video downloading engine
- **spotdl** (<https://github.com/spotdl/spotdl>) - Spotify downloading engine
- **deno** (<https://deno.com/>) - A modern runtime for JavaScript and TypeScript
- **FFmpeg** (<https://ffmpeg.org>) - Media processing and format conversion

All dependencies are automatically managed by MediaTools Video Downloader, before download prompts user for permission to download.

---

[Back to Table of Contents](#)

## 2. Installation

### 2.1 System Requirements

#### **Minimum Requirements**

##### **Windows:**

- Windows 10 or later (64-bit)
- 4GB RAM
- 500MB free disk space (plus space for downloaded videos/audio)
- Internet connection

##### **Linux:**

- Ubuntu 20.04+ or equivalent distribution
- 4GB RAM
- 500MB free disk space
- Internet connection
- Python 3.8+ (usually pre-installed)

##### **macOS:**

- macOS 10.14 (Mojave) or later
- 4GB RAM
- 500MB free disk space
- Internet connection

#### **Dependencies**

- yt-dlp (automatically managed)
- spotdl (automatically managed)
- deno (automatically managed)
- FFmpeg (automatically managed)

#### **Recommended Requirements**

- 8GB RAM or more
- SSD for faster video processing
- Broadband internet connection (10 MBps or faster)

### 2.2 Installation Steps

#### **Step 1: Choose Your Version & Download**

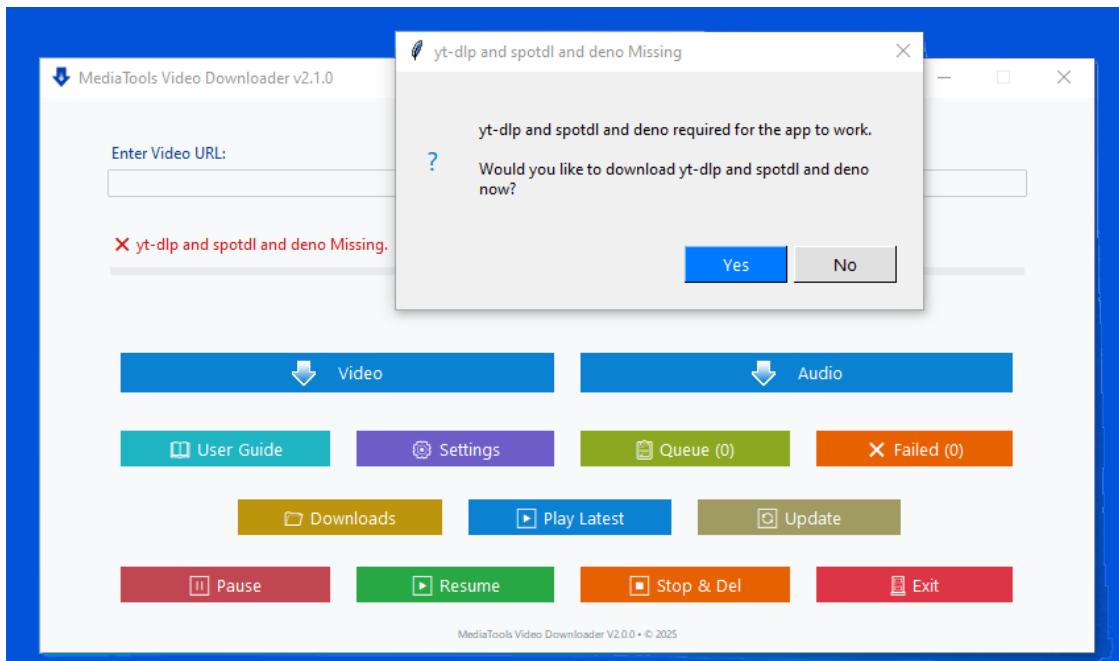
1. Visit the MediaTools GitHub releases page and choose between two versions for each platform:
  - **Single Executable:**
    - Clean appearance
    - Application data stored in User/AppData/ (Windows) or ~/.config/ (Linux)
  - **Portable Version:**
    - One portable dir – Self contained
    - All application files reside in the \_internal/ folder alongside the executable
    - Can be placed anywhere (USB drives, external drive, etc.)
2. Download the appropriate installer for your operating system:
  - **Windows:** mt-vdl.exe
  - **Linux:** mt-vdl (executable)
  - **macOS:** mt-vdl.app or mt-vdl (executable)

**Step 2: Desktop Shortcut (Optional)** The application will offer to create a desktop shortcut for easy access. You can:

- Accept to create the shortcut now
- Decline and create it later

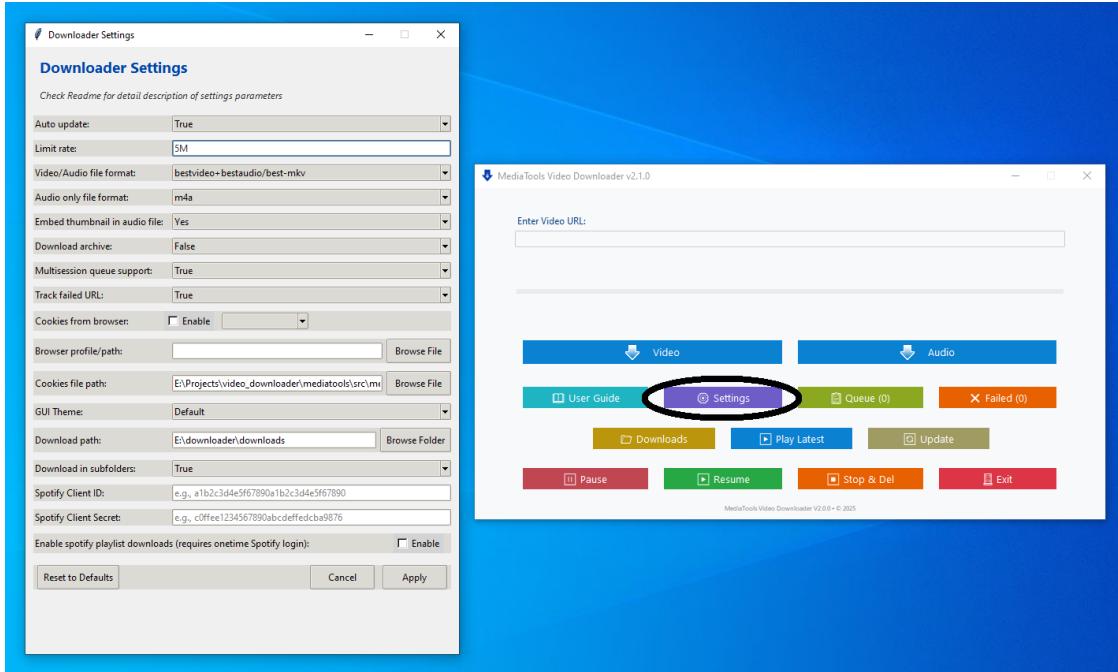
**Step 3: Install Dependencies** On first launch, MediaTools Video Downloader will:

1. Check for yt-dlp and FFmpeg
2. Prompts you for permission to download, if not present
3. Configure them for optimal performance



## Step 4: Configure Initial Settings

- Reset to Defaults as initial settings
- Restart app



## Step 5: Configure Preferred Settings

- After initial default settings and restarting app, if required, you may set your preferred settings like Download location, Default video format, Audio format, Download speed limit etc.

### 2.3 Platform-Specific Notes

#### Windows

- May require administrator privileges for first run
- Windows Defender/Antivirus might flag unknown executables (false positive)
- Add exception if needed: Settings → Security → Windows Security
- Desktop shortcut creation may require permissions

#### Linux

- Ensure execute permissions: chmod +x mt-vdl
- Some distros require additional dependencies

- Desktop file may need to be marked as trusted
- Check file manager thumbnail settings for video previews

#### macOS

- May show “unidentified developer” warning
- Allow in System Preferences → Security & Privacy
- Grant permissions for network access
- Some features require full disk access permission

[Back to Table of Contents](#)

---

### 3. Getting Started

#### 3.1 Launching the Application

##### Windows:

- Double-click the desktop shortcut, or
- Run mt-vdl.exe from the installation folder

##### Linux:

- Click the desktop icon, or
- Run ./mt-vdl from terminal, or
- Launch from applications menu

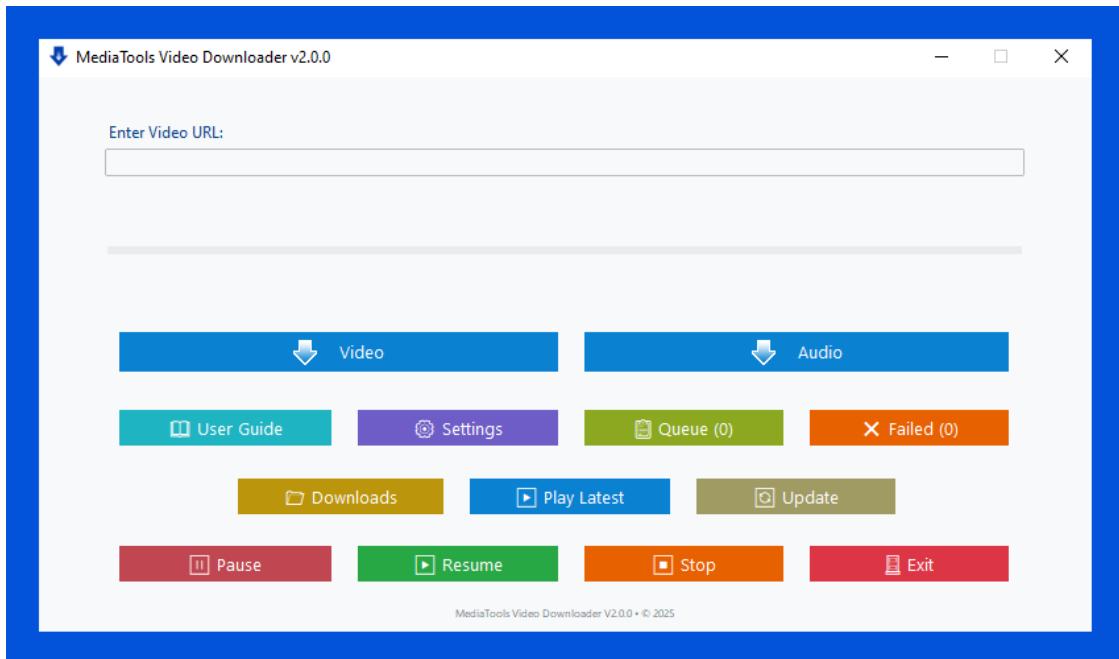
##### macOS:

- Double-click the application icon, or
- Open from Applications folder

#### 3.2 Interface Overview

##### Window Layout

The MediaTools Video Downloader interface is organized into several key areas:



## 1. URL Input Section

- **URL Input Field:** Primary field for pasting video/audio URLs
- **URL Management:**
  - Single URL Processing: Paste and download immediately
  - Queue System: Multiple URLs processed sequentially
  - Clipboard Integration: Right-click paste option

## 2. Progress Visualization

- **Progress Bar:** Real-time status display of overall download progress of active download

## 3. Download Control Buttons

- **Download | Add To Queue:** Process current URL or add to download queue
- **Docs:** Open documentation folder with all help files
- **Settings:** Access application configuration and themes
- **Queue:** View and manage download queue
- **Failed URL:** Review and retry failed downloads
- **Downloads:** Open file explorer to download location
- **Play Latest:** Launch the most recently downloaded video
- **Update:** Check for yt-dlp/FFmpeg and application updates
- **Pause:** Temporarily halt all active downloads (preserves progress)
- **Resume:** Continue paused downloads from interruption point
- **Stop:** Cancel all current operations and clear active downloads
- **Exit:** Safely close the application

## Document Access

- **Docs button:** Opens the local docs/ folder containing:
  - README.md (Quick start guide)
  - LICENSE (Application license)
  - UserGuide.pdf (Complete documentation)
  - FAQ.md (Troubleshooting guide)

## Download Controls

- **Pause/Resume:** Maintains download state for resumable formats
- **Stop:** Complete cancellation with cleanup
- **Real-time Progress:** Continuous feedback during operations

## Quick Access Features

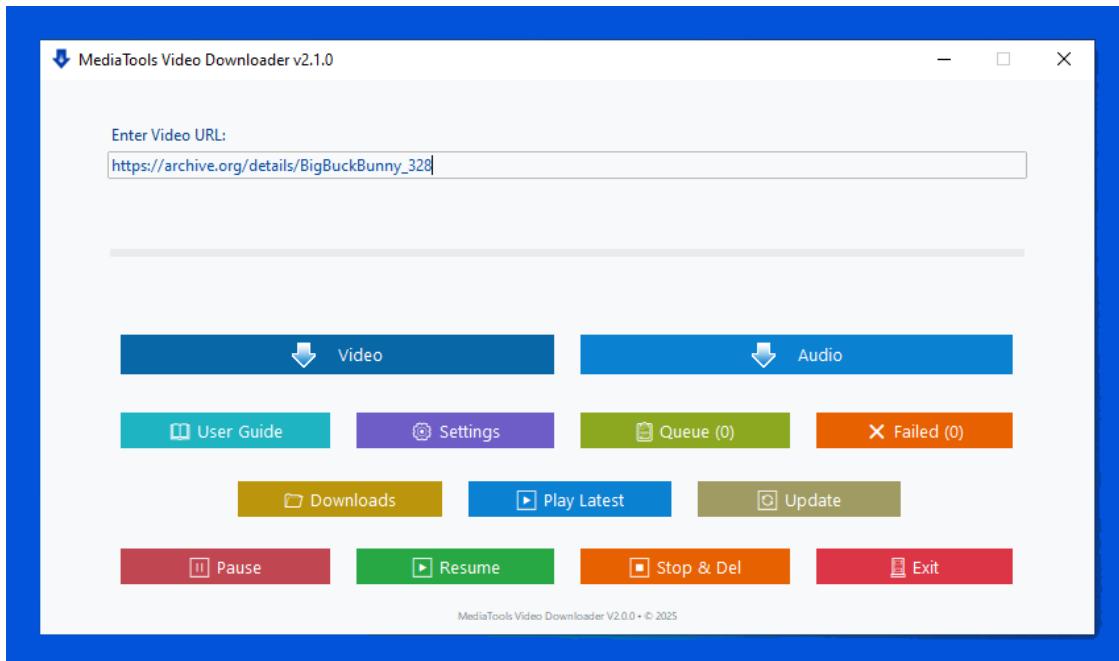
- **Open Download Folder:** Direct access to downloaded files
- **Play Latest Video:** Quick preview of most recent download
- **Update Tools:** Ensure yt-dlp/FFmpeg and application are current

## Theme & Customization

- Accessible via the Settings button:
- **Theme Selection:** Choose between Dark, Light, and Minimalist themes
- **Instant Application:** Theme changes apply immediately
- **Persistent Settings:** Preferences saved between sessions

## 3.3 First Download

1. Launch MediaTools Video Downloader
2. Copy a video URL from any yt-dlp supported platform
3. Paste the URL in the input field at the top
4. Click the “Download” button
5. Watch the progress in the status area
6. Click downloads button on App to check your downloads folder



[Back to Table of Contents](#)

---

## 4. Downloading Videos

### 4.1 Basic Download Process

#### Single Video:

1. Copy the video URL from your browser
2. Paste it into the URL input field
3. Click “Download”
4. Monitor progress in the status bar

#### Playlist Download:

1. Paste the playlist URL
2. The application automatically detects it's a playlist
3. Click “Download”
4. All videos in the playlist will be downloaded sequentially
5. Progress bar shows playlist real time download status

**Supported Platforms** yt-dlp supports over 1000 websites. Check [yt-dlp official supported sites list](#) for detailed list.

**Download Progress Information** During download, you'll see:

- **Current File:** Which file is downloading (Url) and **Percentage Complete:** 0% to 100%

**Pause and Resume Note:** Pause and Resume functionality depends on the video platform's support for partial downloads and not supported for Format setting - “bestvideo+bestaudio/best-mp4”.

#### Pausing a Download:

1. Click the “Pause” button during download
2. The download stops and progress is saved, partial files (.part) are retained

#### Resuming a Download:

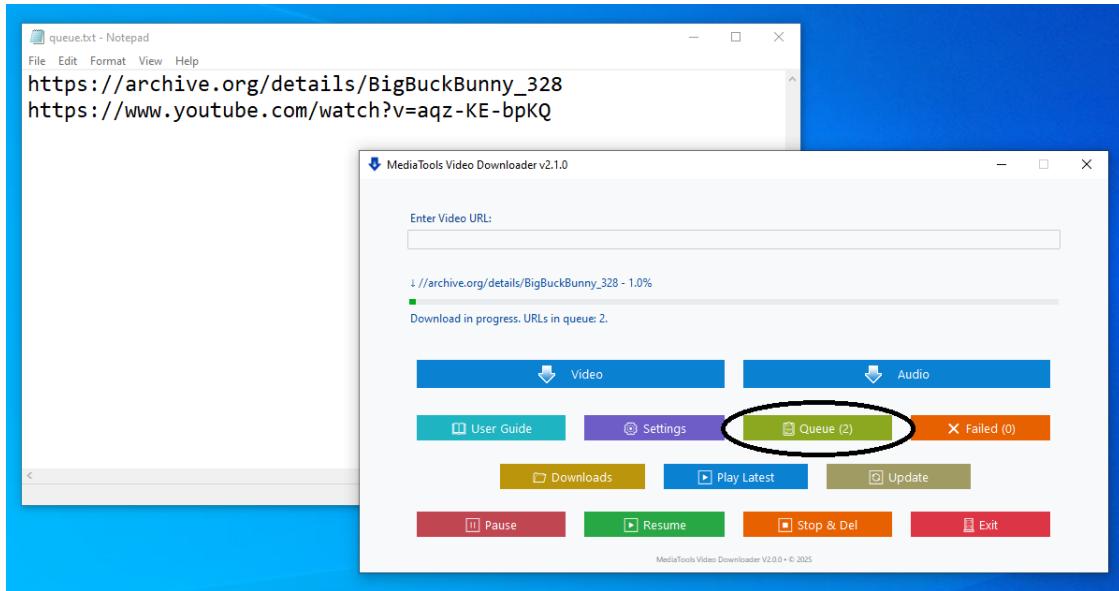
1. Click “Resume” button
2. Download continues from where it stopped

## 4.2 Queue Management

**Understanding the Queue** The queue system allows you to:

- Download multiple videos/audio sequentially
- Manage download order

- Pause/resume entire queues\*
- Track failed downloads\*



**Queue File Format** The Queue.txt file uses a simple format to distinguish between video and audio downloads:

**Video URLs:** (default) <https://www.youtube.com/watch?v=abc123> <https://vimeo.com/example-video>

**Audio URLs:** (prefixed with “audio:”) audio:<https://www.youtube.com/watch?v=abc123>  
audio:<https://vimeo.com/example-video>

#### Adding to QueueManual Addition:

1. There are two ways to create queue:
  - Paste URL in url entry field and click “Download”, every time you add new URL it will be added to queue
  - You can also directly paste multiple URLs in Queue.txt (even works with comma separated URLs) and all these URLs will be added to queue. After adding URLs in Queue.txt, click “Download”. You can directly access Queue.txt by clicking “Queue” button on App
2. If a download is in progress, the new URL is queued
3. After adding URLs in Queue.txt, click “Download” to process the queue
4. You can directly access Queue.txt by clicking “Queue” button on App
5. If a download is in progress, new URLs are added to the end of queue
6. Downloads process in order (FIFO - First In, First Out)

## **Supported formats in Queue.txt:**

- **One URL per line:**

https://youtube.com/video1  
audio:https://youtube.com/video2  
https://vimeo.com/video3

- **Comma-separated URLs** (alternative):

https://youtube.com/video1, audio:https://youtube.com/video2, https://vimeo.com/video3

- **Space-separated URLs** (alternative):

https://youtube.com/video1 audio:https://youtube.com/video2 https://vimeo.com/video3

## **Audio URLs in Queue:**

- When you add a URL for audio download, it's automatically prefixed with "audio:" in the queue
- You can manually edit Queue.txt to add new URLs or change between video and audio for any URL

## **Playlist Handling:**

- Playlists are automatically identified and all videos in playlist downloaded sequentially
- You can see pending video count from playlist in the download status

## **Queue Operations**

### **View Queue:**

- The Queue button on App opens Queue.txt file with list of URLs in queue
- Completed downloads are automatically removed from Queue.txt

### **Manage Queue Items:**

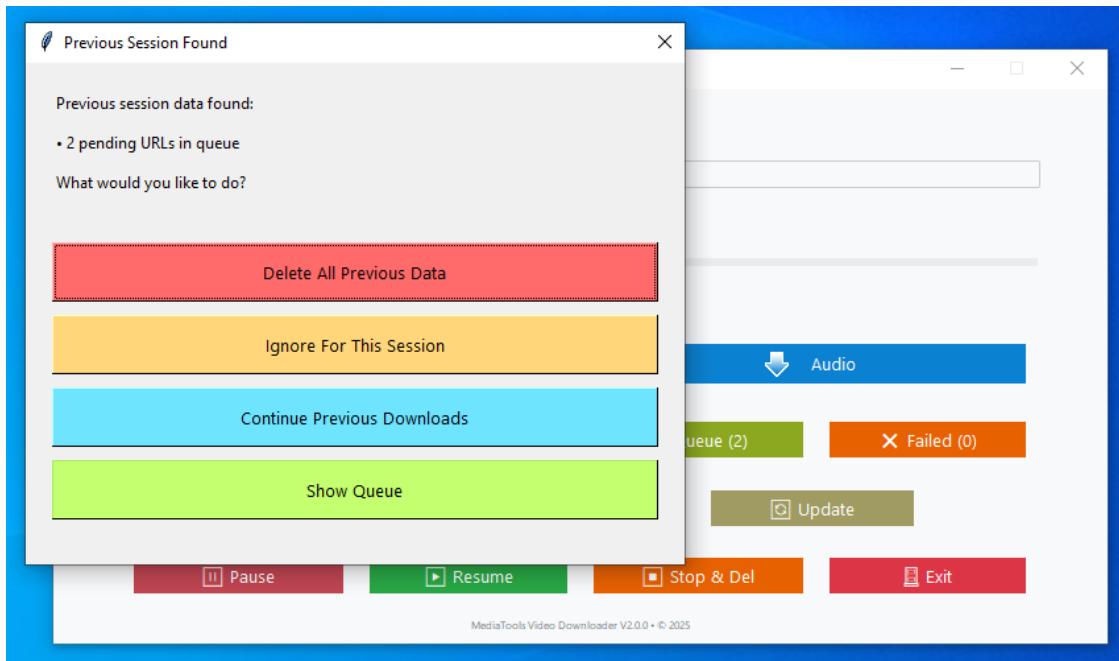
- Pause or Stop (Terminate) download ongoing download
- Stop download will remove URL from queue
- Edit Queue.txt file to modify queue

## **Queue Persistence – with Setting option “Multisession queue support” Enabled:**

- Can be enabled with Settings option: “Multisession queue support”
- Failed downloads are tracked separately (if enabled)

**Multi-Session Queue Support** When enabled in settings, this feature allows:

- At startup provides option to continue previous download session
- Continue downloading queue from previous session
- If enabled and supported, partially downloaded last video from previous session will be resumed



[Back to Table of Contents](#)

---

## 5. Audio Download Feature

### 5.1 Audio Download Overview

MediaTools Video Downloader now includes comprehensive audio extraction capabilities, allowing you to download audio tracks from videos in various formats.

#### Key Audio Features:

- Extract audio from any supported video platform
- Multiple audio format options
- Embedded thumbnail/cover art support
- Organized storage in dedicated Audio folder
- Batch download support via queue system

**Audio Storage Location:** All audio downloads are automatically saved to: [Downloads Folder]/Audio/

#### Supported Sources:

- Any video platform supported by yt-dlp
- Music videos, podcasts, interviews, lectures
- Playlists and channel audio extraction

### 5.2 Audio Format Selection

#### Available Audio Formats:

##### M4A (MPEG-4 Audio)

- **Best for:** Modern media players
- **Quality:** Better quality than MP3 at similar bitrates
- **File Size:** Similar to MP3
- **Compatibility:** Good (may not work on older devices)

##### MP3 (MPEG Audio Layer III)

- **Best for:** Universal compatibility, music players, car systems
- **Quality:** Good compression, widely supported
- **File Size:** Moderate
- **Compatibility:** Excellent (works on virtually all devices)

### **Best Audio (Original Quality)**

- **Best for:** Audio quality preservation, archival
- **Quality:** Highest available from source
- **File Size:** Variable (often larger)
- **Compatibility:** Depends on source format

### **Thumbnail Embedding for Audio:**

- **Enabled/Disabled:** Choose whether to embed cover art in audio files
- **MP3:** Reliable thumbnail support
- **M4A:** Excellent thumbnail support
- **Best Audio:** Variable support depending on source format

## **5.3 Audio Quality Considerations**

### **Important Quality Information:**

**Transcoding Behavior:** When downloading audio files, yt-dlp may transcode between formats. Understanding this behavior helps maintain audio quality:

#### **Scenario 1: Downloading same file, lower quality after higher quality**

- If you download M4A first, then download MP3 for the same content
- yt-dlp will transcode the existing M4A file to MP3 format
- **Result:** Quality loss due to transcoding to inferior sound quality format (M4A→MP3).

#### **Scenario 2: Downloading same file, higher quality after lower quality**

- If you download MP3 first, then download M4A for the same content
- Both files are preserved separately
- **Result:** No quality loss, both formats available

### **Recommended Download Strategy if you wish to download same audio file with multiple formats:**

1. **Download lowest quality first (MP3)**
2. **Then download higher quality format (M4A)**
3. **Then highest quality format (bestaudio)**
4. This preserves original quality while allowing format flexibility

### **File Organization:**

- All audio files are stored in Audio/ subfolder
- Separate from video downloads for better organization
- Filename pattern: [Title]-[Video\_ID].[Audio\_Extension]

[Back to Table of Contents](#)

---

## 6. Settings and Features

### 6.1 Settings Configuration

Access settings through: **Settings Button on App**

#### General Settings

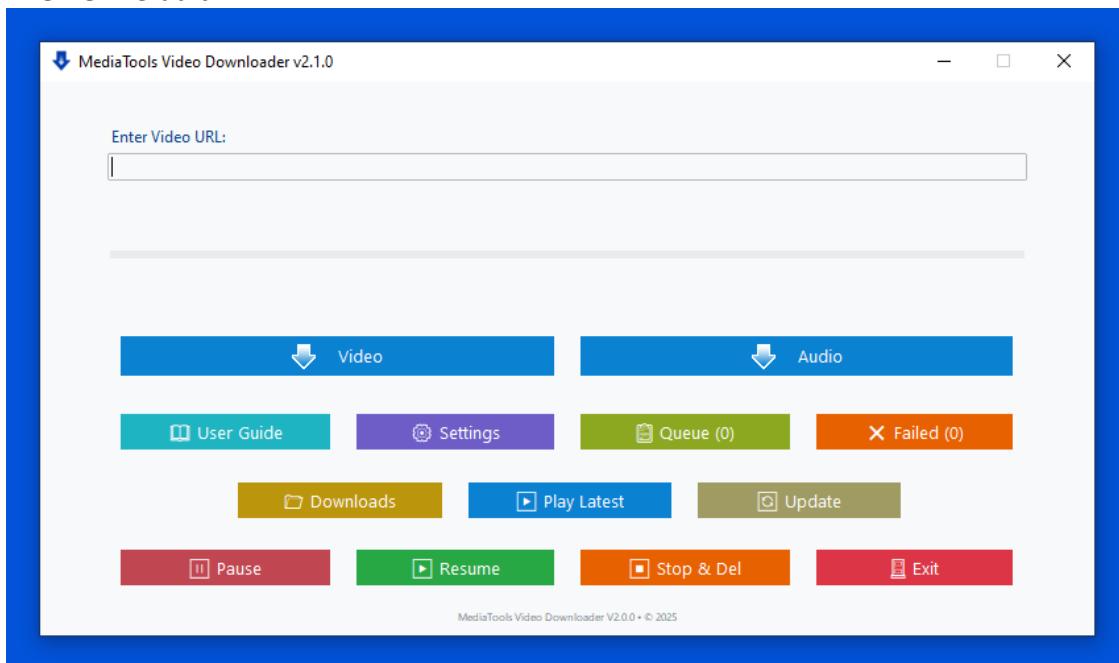
##### Auto Update (yt-dlp & Application)

- **Default:** Enabled (Highly recommended)
- **Description:** Automatically updates yt-dlp and checks for application updates on startup
- **Recommendation:** Keep enabled to ensure compatibility with video platforms and latest features

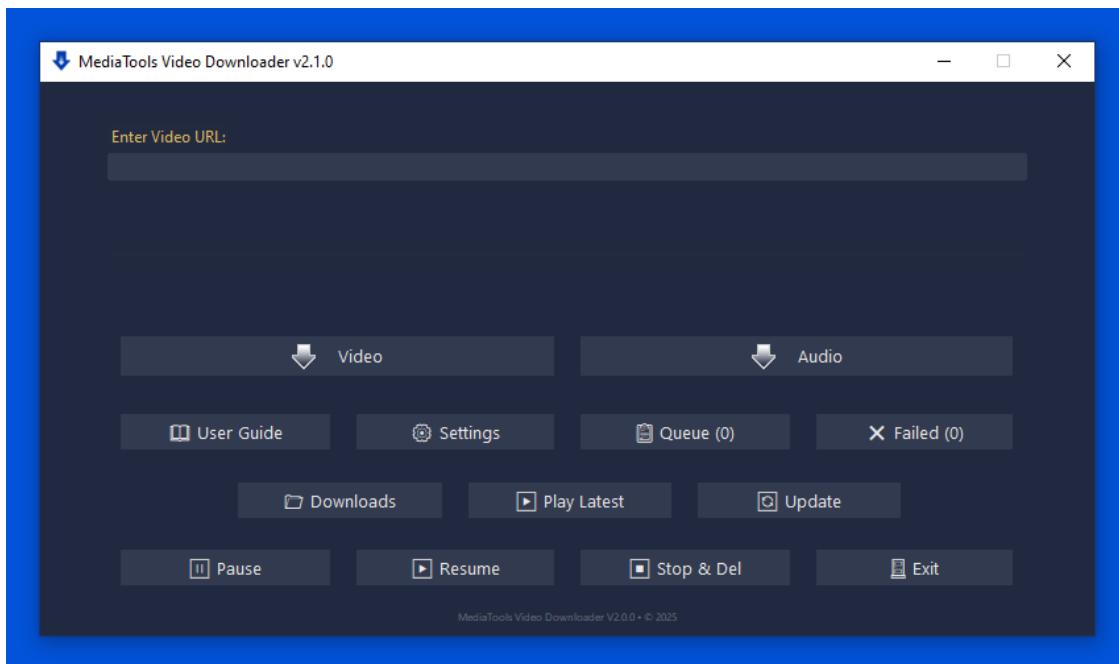
#### GUI Theme

- **Options:** Default, Dark, Unicolor, Minimalist
- **Description:** Changes the application's visual appearance
- **Note:** Theme selection persists across sessions

#### Theme: Default



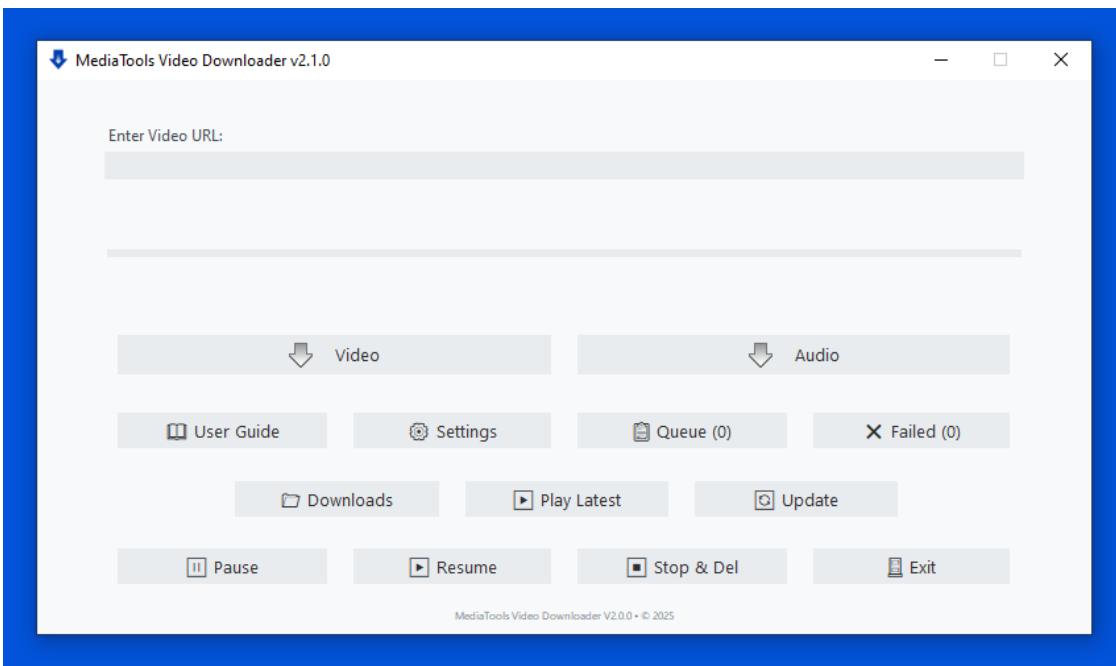
**Theme: Dark**



**Theme: Unicolor**



## Theme: Minimalist



## Download Settings

### Download Speed Limit

- **Default:** 5M (5 megabytes per second)
- **Description:** Limits download speed to prevent network saturation
- **Usage:**
  - Set lower (1M-2M) for shared networks
  - Set higher (10M+) some platforms/sites may block or restrict high speed downloads

### Downloads Directory

- **Default:** [User]/downloads/ or application folder
- **Description:** Where downloaded video and audio files are saved (audio files to "Audio" subfolder and video files to "Video" subfolder)
- **Options:**
  - Click "Browse" to change location
  - Create new folders as needed
  - Ensure sufficient disk space

### Enable Download Archive

- **Default:** Disabled

- **Description:** Tracks downloaded videos to avoid re-downloading
- **When Enabled:**
  - Creates download\_archive.txt in data folder
  - Skips previously downloaded videos automatically

## **Audio Download Settings**

### **Audio Format**

- **Options:**
  - m4a - MPEG-4 Audio (good quality, Apple compatible)
  - mp3 - MP3 Audio (universal compatibility)
  - bestaudio - Best available audio quality (original format)
- **Description:** Select preferred audio format for audio downloads

### **Embed Thumbnail in Audio**

- **Default:** Enabled
- **Description:** Embeds cover art/thumbnail in audio files
- **Benefits:** Visual identification in media players
- **Compatibility:** Works best with MP3 and M4A formats

### **Format Settings:**

#### **Stream and Merge Format**

- **Options:**
  - bestvideo+bestaudio/best-mkv - Best quality in MKV container
  - bestvideo+bestaudio/best-mp4 - Best quality in MP4 container
  - b - Single best quality format

#### **Understanding Format Options:**

##### **MKV Format:**

- **Pros:** Highest quality, supports all codecs, better for archiving
- **Cons:** Larger file sizes, thumbnail support varies
- **Best for:** Quality-focused users, video editing, archival

##### **MP4 Format:**

- **Pros:** Universal compatibility, smaller files, reliable thumbnails
- **Cons:** Slightly lower quality ceiling, limited codec support
- **Best for:** Mobile viewing, sharing, streaming devices

##### **Single Best (b):**

- **Pros:** Fastest download, single file, reliable
- **Cons:** Quality depends on platform's pre-encoded formats
- **Best for:** Quick downloads, low storage space

Format	Description	Best For
<b>MP4</b>	Merges best video and audio streams	Max compatibility (phone, social media)
<b>MKV</b>	Merges best video and audio streams	Highest quality preservation, pause/resume support
<b>b (best)</b>	Single pre-merged file when available	Fastest downloads, no merging required

## Queue Settings

### Multisession Queue Download Support

- **Default:** Enabled
- **Description:** Saves queue state between application sessions
- **When Enabled:**
  - Queue persists after closing app
  - Resumes unfinished downloads
  - Maintains download order

### Track Failed URLs

- **Default:** Enabled
- **Description:** Saves URLs that failed to download
- **Benefits:**
  - Review failed downloads later
  - Retry failed URLs easily
  - Identify problematic sources

## Advanced Settings

### Platform-Specific Download Folders

- **Default:** Enabled
- **Description:** If enabled, creates separate sub-folders for each platform (platform names to be specified in "Subfolder Domains" for platforms/domains to group videos)
- **When Enabled:**
  - YouTube videos → downloads/youtube/
  - Twitter videos → downloads/twitter/
  - Other platforms → downloads/other/

## Subfolder Domains

- **Default:** Empty
- **Description:** Custom domain-to-folder mappings
- **Format:** domain1.com:folder1, domain2.com:folder2
- **Example:** youtube.com:YT, vimeo.com:Vimeo

**Cookie Settings** Cookies allow downloading private or membership-only content.

## Cookie File

- **Description:** Import cookies from a text file
- **Format:** Netscape cookie format
- **How to Get:**
  1. Use browser extension (e.g., "Get cookies.txt")
  2. Export cookies from target website
  3. Save as cookies.txt
  4. Browse and select in settings

## Cookies from Browser

- **Description:** Extract cookies directly from installed browser
- **Supported Browsers:**
  - Brave
  - Chrome
  - Chromium
  - Edge
  - Firefox
  - Opera
  - Safari
  - Vivaldi
  - Whale

## Browser Selection:

- Choose your browser from dropdown
- Optionally specify browser profile
- Cookies are read automatically during download

## Browser Profile:

- **Default:** Default profile used
- **Custom:** Specify profile name for multi-profile browsers
- **Example:** "Profile 1", "Work", "Personal"

### **Important Notes:**

- Cookies may expire; refresh periodically
- Close browser before extracting cookies
- Some platforms detect and block cookie usage

## 6.2 Spotdl/Spotify Credentials Settings

### **Spotify Settings**

#### **Spotify Client ID and Secret**

- **Description:** Your Spotify API credentials, required for downloading metadata from Spotify.
- **How to Get:**
  1. Create a free user account on Spotify.
  2. Go to the [Spotify Developer Dashboard](#).
  3. Create a new application.
  4. You will find your Client ID and Client Secret in the application's settings.
  5. For a detailed guide, you can watch any YouTube video on this topic.

#### **Enable Spotify Playlist Downloads**

- **Description:** Enables OAuth to download private and collaborative playlists.
- **How to Set Up:**
  1. Go to your application on the [Spotify Developer Dashboard](#).
  2. Click on “Edit Settings”.
  3. Under “Redirect URLs”, enter `http://127.0.0.1:8000/callback`.
  4. Save the settings.

## 6.3 Performance Tips

### **Optimizing Download Speed**

1. **Adjust Speed Limit:** Know what download speed will work best for your requirements

### **Managing Disk Space**

1. **Choose Appropriate Format:** MP4 is smaller than MKV, audio files are smaller than video
2. **Monitor Available Space:** Ensure sufficient room before large downloads
3. **Regular Cleanup:** Delete paused and skipped video files (.part, webp etc.)

4. **Platform-Specific Folders:** Organize by platform for easier management

### Queue Management Best Practices

1. **Prioritize Important Downloads:** Place urgent items at queue front
2. **Use Download Archive:** Avoid re-downloading same content

## 6.4 Updates and Maintenance

**Keeping yt-dlp and Application Updated:** Setting Auto-Update in settings is highly recommended

### Auto-Update:

- Enable in Settings → Auto Update
- Checks for updates on startup
- Downloads automatically if available

### Why Update yt-dlp:

- yt-dlp frequently releases updated versions as video platforms frequently change their APIs
- New platform support added
- Bug fixes and performance improvements
- If Auto-Update not enabled, yt-dlp can be updated manually through Update button on App

**Application Updates** The update system now checks for both:

- **yt-dlp updates** - for video platform compatibility
- **Application updates** - for new features and bug fixes

### Checking for Updates Manually:

1. Click “Update” button in the application
2. Visit GitHub releases page: <https://github.com/MediaTools-tech/mediatools>
3. Download latest version if available
4. Install over existing version (settings preserved)

## 6.5 Advanced Features

### Thumbnail Embedding

#### What It Does:

- Embeds video thumbnail inside the video file
- Embeds cover art in audio files (when enabled)
- Allows preview in file managers and media players
- Useful for organizing large media libraries

#### Format Compatibility:

- **MP4:** Reliable thumbnail support
- **MKV:** Variable support depending on player
- **MP3/M4A:** Good thumbnail/cover art support
- Formats with limited or no thumbnail embedding support: WEBM, WAV, FLAC, OGG, AAC, WMA, AIFF

#### Settings:

- Enabled by default for MP4 format and audio files
- Automatically converted to compatible format
- Thumbnails stored with media metadata

**Download Naming** Media files are automatically named using this pattern: [Title]-[Video\_ID].[Extension]

**Example:**How\_to\_Use\_MediaTools-dQw4w9WgXcQ.mp4Song\_Name-dQw4w9WgXcQ.mp3

#### Features:

- Special characters removed for compatibility
- Title truncated to 100 characters
- Video ID ensures uniqueness
- Prevents filename conflicts

#### Restricted FilenamesAutomatically Enabled:

- Removes special characters: <> : " / \ | ? \*
- Replaces spaces with underscores
- Ensures cross-platform compatibility
- Prevents filesystem errors

**Download Archive** It's a text file containing IDs of downloaded videos, not names.

**Purpose:** Prevents re-downloading videos you already have.

### **How It Works:**

1. After successful download, video ID is saved
2. On next download attempt, ID is checked
3. If found, download is skipped
4. Saves time and bandwidth

**Archive Location:**[App Data]/data/download\_archive.txt

### **Use Cases:**

- Downloading updated playlists
- Channel subscriptions
- Avoiding duplicates

### **Failed URL Tracking When Enabled:**

- Failed downloads are logged with reason
- URLs saved to failed\_url.txt
- Allows bulk retry later
- Helps identify problematic sources

### **Accessing Failed URLs:**

- Current session Failed\_url.txt can be accessed by clicking “Failed” button on App or navigate to application data folder for Failed\_url.txt and Failed\_url\_old.txt (for failed URLs from previous session)

## 7. FAQ and Troubleshooting

### 7.1 FAQ

#### General Questions

##### **Q. Is MediaTools Video Downloader free?**

A. Yes, MediaTools Video Downloader is open-source and free to use under the MIT License.

##### **Q. Which websites are supported?**

A. yt-dlp supports 1000+ sites. For the complete current list, check: [yt-dlp supported sites list](#)

##### **Q. Do I need to install Python?**

A. No, the application is self-contained and includes all necessary components.

##### **Q. Is my data collected or shared?**

A. No. The application does not collect, store, or transmit any personal data. All processing happens locally on your computer.

##### **Q. My antivirus flags the application - is it safe?**

A. Yes, this is a common false positive with applications that download executables (yt-dlp/FFmpeg). The application is open-source and completely safe. You may need to add an exception in your antivirus software.

##### **Q. Why can't I download from some websites?**

A.

- The website may not be supported by yt-dlp, check yt-dlp supported sites list
- Regional restrictions may block access
- Some sites require cookies for authentication

#### Installation & Setup

##### **Q. What if I skip downloading yt-dlp and ffmpeg?**

A.

**yt-dlp:** The application cannot function without yt-dlp. You have two options:

- Either allow the application to automatically download yt-dlp when prompted
- Or manually download yt-dlp from [yt-dlp official repository](#) and place the executable in the bin/ folder

**FFmpeg:** The application can work without FFmpeg, but with limitations:

- Format 'b' (pre-merged streams) will work without FFmpeg, though these may not offer the best available quality
- Video merging (bestvideo+bestaudio) requires FFmpeg
- Audio format conversion requires FFmpeg

**Q. How to fix “yt-dlp not found” error?**

- App → Settings → Reset to defaults, Restart App
- Allow app to download yt-dlp automatically
- **Manual solution:**
  - Download yt-dlp from the [official releases page](#)
  - Place the executable (yt-dlp or yt-dlp.exe) in the bin/ folder
  - Restart the application

## Download Performance

**Q. What download speed should I set?**

A.

**② 5M (5 MBps) is recommended for most users because:**

- Conservative: Unlikely to be blocked by servers
- Stable: Provides consistent download performance
- Network-friendly: Doesn't monopolize your internet connection
- Reliable: Works well with most hosting platforms

**② Higher speeds (10M-20M) may work but risk:**

- Server blocking: Some platforms detect and block high-speed downloads
- IP throttling: Your IP might be temporarily limited
- Adjust based on your: Internet plan, network stability, and target platform policies

**Q. Can I change the download folder?**

A. Yes. Go to Settings → Download Path and select a new folder. Make sure you have write permission to that location.

**Q. Why does progress bar runs multiple times for a single video download?**

A. This is normal behavior indicating separate component downloads:

1. Video stream - Main video content
2. Audio stream - Separate audio track
3. Thumbnail - Video preview image
4. Metadata - Video information

The application downloads these components then merges them into the final file.

**② Why does download fail when the platform is supported by yt-dlp and I can see the video in my browser?**

This is a common issue that can happen for several reasons:

## **Platform Anti-Bot Measures**

**Many platforms actively block automated downloads:**

- Uses sophisticated bot detection
- May throttle downloads from certain IP ranges
- Require cookies from a logged-in session
- May need authentication or has rate limits

**Solution:**

- **Update yt-dlp** (if Auto-Update is disabled in settings)
- **Provide browser profile path** in Settings (recommended - use your actual browser session profile path)
- **Use cookies.txt file** (advanced users only - be aware of security risks when sharing cookies)

## **Queue & Session Management**

**Q. Why is my queue from previous session not showing up?**

**A.** Ensure “Multisession queue support” is enabled:

1. Go to Settings
2. Enable “Multisession queue support”
3. Restart the application

**Q. Why does Windows Explorer slow down with download folder?**

**A.** This can occur when partially downloaded files accumulate:

**Cause:** With multisession support enabled, paused and ignored/skipped downloads create temporary files:

- Manually clean the download folder periodically (Remove accumulated, partially downloaded .part files, if any)
- The application includes robust cleanup logic, but manual intervention may be needed for skipped downloads

## **Format & Container Issues**

**Q. Where are video files stored?**

**A.** All video downloads are automatically saved to:  
[Your Downloads Folder]/Video/

**Q. Which video format should I choose?**

**A.**

- **MKV:** Highest quality, supports pause/resume, better for archival
- **MP4:** Best compatibility (phones, tablets, social media), smaller file sizes
- **Single best pre-merged A/V (b):** Fastest download, no merging required, but may compromise on quality

**Q. What video formats are used in yt-dlp cmd with various download options?**

**A.**

- **bestvideo+bestaudio/best-mkv**  
--format "bestvideo+bestaudio/best" --merge-output-format mkv  
*Downloads best video and audio separately, then merges into MKV container*
- **bestvideo+bestaudio/best-mp4**  
--format "bestvideo+bestaudio/best" --merge-output-format mp4  
*Downloads best video and audio separately, then merges into MP4 container*
- **b**  
--format "b"  
*Downloads single best pre-merged format (no merging required)*

**Q. What happens if my chosen format fails to download?**

**A.** The app automatically uses a smart fallback system to ensure your download succeeds:

### Format Fallback Chain

- **MKV → MP4 → Single best** (if MKV fails, tries MP4, then single file)
- **MP4 → Single best** (if MP4 fails, tries single file)
- **Single best** (no fallback needed - most reliable)

### How It Works

1. Starts with your preferred format selection
2. If any issues occur (merging failures, codec problems, etc.)
3. Automatically retries with the next format in the chain

### Example Scenarios

- MKV download fails → Automatically tries MP4 → Download succeeds
- MP4 download fails → Automatically tries single file → Download succeeds
- No manual intervention needed - the app handles it automatically

*This system ensures maximum download success rates while respecting your format preferences.*

**Q. Why are MKV files bigger?**

**A.**

- MKV files are often larger because they are commonly used to store the highest quality video and audio available, including lossless audio tracks
- MP4 files are more focused on broad compatibility and are typically compressed more for smaller file sizes

**Q. Why is pause/resume not supported for MP4 format like MKV?**

**A. Technical limitation of MP4 container:**

- **MKV:** Supports true pause/resume at any point
- **MP4:** The MP4 container format doesn't support efficient resuming of partial downloads like MKV does. Resuming MP4 downloads often requires re-downloading significant portions
- **MP4:** Can only resume from certain checkpoints

**Q. How can I pause playlist download with MP4 format as pause isn't supported (with MP4)?**

**A. You can pause(functionally), with caveats (With - Multisession queue support):**

- With MP4 merge format, Playlist progress is saved between sessions (exit and restart)
- Already downloaded videos will not be downloaded again when you restart downloader
- Only last partially downloaded video (at exit time) will download again to continue playlist download

Ensure: "Multisession queue support" is enabled. The application will remember your position in the playlist and only overwrite the currently downloading video if interrupted.

**Q. Why isn't thumbnail showing for MKV files?**

**A. MKV thumbnails require external system support:**

- **Windows:** Install [Icaros](#) to enable MKV thumbnail previews in File Explorer
- **Linux:** Most modern file managers (Nautilus, Dolphin) support MKV thumbnails natively
- **Note:** This is a Windows Explorer limitation, not an application issue. The thumbnails are embedded correctly in the files

## **Audio Download Questions**

**Q. Where are audio files stored?**

**A. All audio downloads are automatically saved to:**

[Your Downloads Folder]/Audio/

**Q. Which audio format should I choose?**

**A.**

- **M4A:** Best balance of quality and compatibility (recommended)
- **MP3:** Universal compatibility, works on all devices
- **Best Audio:** Highest quality, but format may vary

**Q. What audio formats are used in yt-dlp cmd with various download options?**

**A.**

- **mp4**  
--format "bestaudio[ext=m4a]/bestaudio[ext=mp4]/bestaudio[acodec=aac]/bestaudio"  
*Prioritizes m4a/mp4 containers with AAC codec, falls back to any best audio*
- **mp3**  
--format "bestaudio" --audio-format "mp3"  
*Downloads best available audio, then converts to MP3 format*
- **bestaudio**  
--format "bestaudio"  
*Downloads the highest quality audio in its original format*

**Q. Why does downloading MP3 after M4A replace the file instead of creating both?**

**A.** This is yt-dlp behavior, not a bug:

- When you download a lower quality format after a higher quality one
- yt-dlp transcodes the existing higher quality M4A file to the new lower quality format MP3
- This prevents duplicate content but may reduce quality
- **Solution:** Download lower quality formats first(MP3), then higher quality(M4A) if both formats needed

**Q. Can I download both video and audio from the same URL?**

**A.** Yes, you can download both video and audio from the same URL, you need to add URL twice, first as “Download Video” and secondly as “Download Audio”. Use the queue system to download both separately if needed.

**Q. Why are some audio files much larger than others?**

**A.** Audio file size depends on:

- Source quality and bitrate
- Audio format and compression
- Duration of the audio
- Embedded metadata and thumbnails

**Q. How do I enable/disable thumbnails for audio files?**

A. Go to Settings → Audio Settings → “Embed Thumbnail in Audio” toggle

## Technical Questions

**Q. Where are my downloads stored?**

A. Default paths(for installable and portable versions):

- **Windows:**%LOCALAPPDATA%\Video Downloader\downloads\ or app folder
- **Linux:**~/downloads/ or ~/.local/share/Video Downloader/downloads/
- **macOS:**~/downloads/ or application directory
- **Audio Files:**[Downloads Folder]/Audio/
- **Video Files:**[Downloads Folder]/Video/

**Q. Where are application settings stored?**

- **Windows:**%LOCALAPPDATA%\Video Downloader\data
- **Linux:**~/.local/share/Video Downloader/data
- **macOS:**~/Library/Application Support/Video Downloader/data

**Q. Can I run multiple instances?**

A. No, only one instance can run at a time to prevent conflicts.

**Q. Does it support proxy servers?**

A. Not currently through the GUI. Advanced users can configure yt-dlp's proxy settings manually.

## Advanced Topics

**Q. Why are some videos downloading slowly despite fast internet?**

A. Common causes:

1. Server limitations: Source platform may throttle downloads
2. Geographic restrictions: Content delivery network delays
3. Check your network settings

**Q. Why do some downloads fail with “Unsupported URL”?**

A. Possible reasons:

1. Platform not supported by yt-dlp
2. Regional restrictions on the content
3. Private/age-restricted content requiring cookies
4. Temporary server issues on the source platform

Check: The [yt-dlp supported sites list](#) for platform compatibility.

**Q. What if the app shows “Permission denied”?**

**A.**

- On Linux/macOS, make sure the download path and temp folder are writable:
  - chmod -R 755 ~/Videos
- On Windows, run the app as administrator

**Q. How do I clean up stuck temp files?**

**A.** If the app closes unexpectedly, leftover .part files may remain. The app tries to remove them on exit, but you can safely delete them manually from the temp/ folder.

**Q. Where are logs stored?**

**A.** You can find the latest session logs in the data/ folder.

**Q. Can I use my own yt-dlp or FFmpeg binaries?**

**A.** Yes. Place your custom builds inside the bin/ folder with the same names (yt-dlp, ffmpeg). The app automatically detects and uses them.

**Q. How do I use cookies for private videos?**

**A.** In Settings, specify the path to your cookies.txt file or enable “Cookies from Browser.” This enables downloading of age-restricted or private content.

**Q. I deleted desktop shortcut, how can I get it back?**

**A.** Close the Video Downloader application completely

**Windows:**

- Right-click on mt-vdl.exe
- Select “Send to” → “Desktop (create shortcut)”
- Or
- Navigate to the application folder → data/
- Delete the file called first\_run.flag
- Restart the application - it will prompt create the desktop shortcut

**If not Windows OS:**

- Navigate to the application folder → data/
- Delete the file ./mediatools (rm ~/.mediatools)
- Restart the application - it will prompt create the desktop shortcut

## 7.2 Troubleshooting

### Common Issues and Solutions

#### Q. “yt-dlp not found” Error?

A. See “Installation & Setup” section above for complete solutions

#### Q. “FFmpeg not found” Error?

A. Cause: FFmpeg not installed or not accessible

1. Go to App → Settings → Reset to defaults, Restart App
2. Allow app to download FFmpeg, if fails try step 3
3. Manually download FFmpeg if needed:
  - **Windows:**<https://www.gyan.dev/ffmpeg/builds/>
  - **Linux:**sudo apt install ffmpeg
  - **macOS:**brew install ffmpeg
  - Place in application’s bin folder

#### Q. Download Fails Immediately?

A.

#### Possible Causes:

1. Video no longer available
2. Private video requiring authentication
3. Unsupported platform

#### Solution:

1. Verify URL is correct and accessible in browser
2. Check if video requires login (use cookies)
3. Try updating yt-dlp
4. Check platform support (check yt-dlp supported domain list)

#### Q. Slow Download Speeds?

A.

#### Possible Causes:

1. Internet connection speed
2. Server-side throttling
3. Download speed limit setting

**Solution:**

1. Check your internet speed
2. Check if download speed set in Settings too low
3. Some platforms throttle, particularly during peak hours

**Q. Videos/Audio Download but Won't Play?**

A.

**Possible Causes:**

1. Incomplete download
2. Corrupted file
3. Missing codecs in media player

**Solution:**

1. Check file size matches expected size
2. Re-download the media
3. Try different media player (VLC/MPV/MPC-BE recommended for videos)
4. Update your media player codecs

**Q. Application Won't Start?**

A.

**Solution:**

- Check system requirements
- Run as administrator (Windows)
- Check antivirus isn't blocking
- Check if bin/ folder has necessary executables
- Reinstall the application
- Delete data folder to reset
- Try running app executable from command window

**Q. Queue Not Persisting?**

A. Cause: Multisession queue support disabled

**Solution:**

1. Go to Settings → Queue Settings
2. Enable “Multisession Queue Download Support”, Save
3. Restart App
4. Queue will now save between sessions

**Q. Downloads stuck at 0% or very slow?**

A.

**Solutions:**

- Check your internet connection
- Try a different video URL to test
- Restart the application

**Q. Audio files missing thumbnails?**

A.

- Check if “Embed Thumbnail in Audio” is enabled in Settings
- Formats with limited or no thumbnail embedding support: WEBM, WAV, FLAC, OGG, AAC, WMA, AIFF
- Some sources may not provide cover art
- Thumbnail embedding requires FFmpeg

[Back to Table of Contents](#)

---

## 8. Contributing and Feedback

**Reporting Bugs** If you encounter issues:

1. **Check This Guide:** Review troubleshooting section
2. **Verify Latest Version:** Update to latest release
3. **Gather Information:**
  - Operating system and version
  - Application version
  - Steps to reproduce
  - Error messages or logs
4. **Submit Issue:**<https://github.com/MediaTools-tech/mediatools/issues>

**Feature Requests** Have an idea for improvement?

1. **Check Existing Issues:** Search for duplicates
2. **Create New Issue:**
  - Clear description of the feature
  - Use case explanation
  - Expected behavior
3. **Use Proper Label:** Apply the “enhancement” label

## 9. Privacy and Legal Information

### 9.1 Privacy and Security

#### Data Privacy

##### MediaTools Video Downloader:

- Does **NOT** collect personal information
- Does **NOT** send usage statistics
- Does **NOT** require account creation
- Does **NOT** connect to external servers (except to the video platforms for downloading content)

#### Cookie Security

When using the cookie import feature:

- Cookies are stored locally only
- They are never transmitted to third parties
- They are encrypted using your browser's own secure storage
- **Treat exported cookie files as sensitive data**

#### Safe Usage Practices

1. **Download Legal Content:** Always respect copyright and the terms of service of the platforms you use
2. **Verify URLs:** Ensure download links are from trusted and legitimate sources
3. **Scan Downloads:** Use antivirus software on downloaded files if you have any concerns
4. **Update Regularly:** Keep the application and its dependencies (like yt-dlp) updated for the latest security and feature improvements
5. **Review Permissions:** Ensure the application has only the necessary file system permissions to function

### 9.2 Legal Information

#### License

MediaTools Video Downloader is licensed under the **MIT License**.

Copyright (c) 2025 MediaTools (Bala)

Permission is granted to use, modify, and distribute this software under the terms specified in the LICENSE file.

## **Third-Party Software**

This application uses the following open-source components:

- **yt-dlp**
  - **License:** Unlicense
  - **Source:**<https://github.com/yt-dlp/yt-dlp>
  - **Purpose:** Core video downloading engine
- **FFmpeg**
  - **License:** LGPL 2.1 / GPL 2.0
  - **Source:**<https://ffmpeg.org>
  - **Purpose:** Media processing and conversion

These components retain their original licenses.

## **Disclaimer**

MediaTools Video Downloader is a graphical user interface (GUI) tool that facilitates the use of **yt-dlp**, an independent third-party video downloading application. MediaTools does not directly download videos; it provides a user-friendly interface to execute yt-dlp commands.

## **Dependency Notice**

**yt-dlp** is downloaded automatically only with explicit user permission during the first-time setup. Users may alternatively obtain yt-dlp independently and place the executable in the application's /bin folder.

## **User Responsibilities**

By using MediaTools Video Downloader, you agree that you are solely responsible for:

- Compliance with all applicable laws and regulations in your jurisdiction
- Respecting copyright, intellectual property rights, and digital rights management (DRM)
- Adherence to the terms of service and usage policies of the platforms you download from
- Obtaining proper authorization or licensing for any content you download
- Any legal consequences resulting from the use of downloaded content

## **No Warranty**

MediaTools Video Downloader is provided “**AS IS**” without warranty of any kind, either express or implied, including but not limited to warranties of merchantability, fitness for a particular purpose, or non-infringement.

## **Limitation of Liability**

The developer and the MediaTools organization shall not be held liable for:

- Any misuse of this software
- Legal violations or copyright infringement by users
- Any damages arising from the use or inability to use this software

## **Intended Use**

This software is designed for downloading publicly accessible content that you have the legal right to access and download. Any use for other purposes is strictly at your own risk.

[Back to Table of Contents](#)

---

## 10. Contact and Support

### Email Support

- Email: [bala.lv.555@gmail.com](mailto:bala.lv.555@gmail.com)

### Website

- <https://mediatools.tech>

### GitHub Issues

- **Bug Reports & Feature Requests:** <https://github.com/MediaTools-tech/mediatools/issues>
  - For feature requests, please use the “enhancement” label
- Public discussion and community help are welcome here

### Documentation

- This User Guide
- README.md in the application folder
- GitHub wiki (when available)

### Community

Stay updated and connect with us:

- **GitHub Repository:** <https://github.com/MediaTools-tech/mediatools>
- **Watch** the repository for notifications on updates
- **Star** the project if you find it useful!
- Share your feedback and suggestions

### Credits

- **Developed by:** Bala
- **Organization:** MediaTools

### Special Thanks To:

- The **yt-dlp** development team
- The **FFmpeg** project contributors
- The wider open-source community
- All early testers and bug reporters

[Back to Table of Contents](#)

---

## 11. Version History

### Version 2.1.0 (Spotify Integration)

- Added Spotify support for downloading tracks, albums, and playlists.
- Integrated spotdl and deno for improved functionality.
- Refactored code for better modularity.
- Added Spotify credential settings to the GUI.

### Version 2.0.0 (Audio Download Release)

- Added audio download feature with format selection (MP3, M4A, Best Audio)
- Audio files organized in dedicated Audio folder
- Thumbnail embedding option for audio files
- Enhanced update system for both yt-dlp and application
- Improved format fallback handling
- Additional audio-related settings and configurations

### Version 1.0.0 (Initial Release)

- First stable release
- Video download functionality
- Queue management system
- Settings configuration panel
- Cross-platform support (Windows, macOS, Linux)
- Browser cookie integration
- Auto-update feature

**Document Version:** 2.1.0

**Last Updated:** 03 Jan 2026

**Document Author:** Bala (MediaTools)

### End of User Guide

For the latest version of this guide, please visit: <https://github.com/MediaTools-tech/mediatools>

[Back to Table of Contents](#)

---