



Audio Processor And Streamer

A Windows desktop application for capturing audio, processing it through VST plugins, and streaming it via HLS/DASH protocols.

Features

- **Audio Capture**
 - WASAPI (Windows Audio Session API) support for capturing system audio
 - ASIO driver support for low-latency professional audio interfaces
 - Multiple simultaneous audio streams
- **VST Plugin Processing**
 - Load and chain multiple VST 2.x plugins
 - Real-time audio processing before encoding
 - Plugin preset management
- **Streaming Formats**
 - HLS (HTTP Live Streaming) with hls.js player
 - DASH (Dynamic Adaptive Streaming over HTTP) with dash.js player
 - Adaptive bitrate streaming with multiple quality profiles
- **Container Formats**
 - fMP4 (Fragmented MP4) for modern browsers
 - MPEG-TS for legacy compatibility
- **Audio Codecs**
 - AAC (Advanced Audio Coding)
 - MP3
 - Opus
- **Built-in Web Server**
 - Serves HLS/DASH manifests and segments
 - Built-in streams overview page with mobile-friendly design
 - Server status indicator with automatic connection monitoring
 - Collapsible stream information sections
 - Configurable port and domain
- **Auto-Update System**
 - Automatic update checking on startup and hourly intervals
 - Download progress indicator
 - Silent update checks (no errors shown when server unavailable)
 - One-click update installation
- **System Tray Support**
 - Minimize to system tray option
 - Background operation while minimized
 - Quick restore from tray icon
- **Real-time Visualization**
 - Input/output spectrum analyzers
 - Audio level monitoring
- **Monitor Output**
 - Route processed audio to a local output device for monitoring

Requirements

- Windows 10/11 (64-bit)
- FFmpeg (included in installer or place in `FFmpeg/bin/` folder)
- VST 2.x plugins (optional, place in `Plugins/` folder)

Installation

Using the Installer

1. Download the latest release from the [Releases](#) page
2. Run `AudioProcessorAndStreamer-Setup-x.x.x.exe`
3. Follow the installation wizard

The installer includes all dependencies - no .NET runtime installation required.

Manual Installation

1. Download and extract the release archive
2. Ensure FFmpeg binaries are in the `FFmpeg/bin/` folder
3. Run `AudioProcessorAndStreamer.exe`

Usage

Quick Start

1. Launch the application
2. Click **Settings** to configure your first stream:
 - Select an audio input device (WASAPI or ASIO)
 - Choose encoding profiles (bitrates)

- o Optionally add VST plugins for processing
- 3. Click **Start** on the web server
- 4. Click **Start** on your stream
- 5. Click the URL displayed to open the streams page in your browser

Configuration

Access settings via the **Settings** button:

Setting	Description
Web Server Port	HTTP port for the streaming server (default: 8080)
Base Domain	Public URL for stream access
Streams Page Path	URL path to the streams overview page
Segment Duration	HLS/DASH segment length in seconds (default: 2)
Playlist Size	Number of segments in playlist (default: 5)
Stream Format	HLS or DASH
Container Format	MP4 or MPEG-TS (HLS only)

Stream Configuration

Each stream can be configured with:

- **Audio Input:** Select WASAPI loopback, WASAPI device, or ASIO device
- **Sample Rate:** 44100, 48000, 96000 Hz
- **Buffer Size:** 256-4096 samples
- **Encoding Profiles:** Multiple bitrate options (64-320 kbps)
- **VST Plugins:** Chain multiple plugins for processing
- **Presets:** Save and load stream configurations (.sts files)
- **Stream Path:** Custom URL path for the stream

New streams automatically load the first available preset from the `Presets/` folder.

Building from Source

Prerequisites

- Visual Studio 2022 or later
- .NET 8.0 SDK
- Inno Setup 6 (for building installer)

Build Commands

```
# Build for development
dotnet build -c Debug -p:Platform=x64

# Build for release
dotnet build -c Release -p:Platform=x64

# Publish self-contained single-file executable
dotnet publish -c Release -p:Platform=x64

# Build installer (requires Inno Setup)
build-installer.bat
```

Publish

```
# 0. Full / default build, remote publish, git commit/tag/push
.\build-installer.ps1 -Version 1.0.0 -GitPublish -PublishRemote -ReleaseNotes "Version 1.0.0 release"

# 1. Basic build with current version (no version update)
.\build-installer.ps1

# 2. Build with version update
.\build-installer.ps1 -Version 1.0.0

# 3. Skip build, only create installer (uses existing publish output)
.\build-installer.ps1 -SkipBuild

# 4. Version update + skip build
.\build-installer.ps1 -Version 1.0.0 -SkipBuild

# 5. Custom Inno Setup path
.\build-installer.ps1 -InnoSetupPath "C:\Program Files\Inno Setup 6\ISCC.exe"

# 6. Build + Git publish (commit, tag, push)
.\build-installer.ps1 -Version 1.0.0 -GitPublish -ReleaseNotes "Fixed critical bug in audio processing"

# 7. Build + Remote publish (upload to server)
.\build-installer.ps1 -Version 1.0.0 -PublishRemote -ReleaseNotes "Added new streaming features"

# 8. Build + Git publish + Remote publish (full release workflow)
.\build-installer.ps1 -Version 1.0.0 -GitPublish -PublishRemote -ReleaseNotes "Version 1.0.0 release"

# 9. Remote publish with custom server details
.\build-installer.ps1 -Version 1.0.0 -PublishRemote -ReleaseNotes "Update" -RemoteUser admin -RemoteServer 10.0.0.5

# 10. Remote publish with password authentication
.\build-installer.ps1 -Version 1.0.0 -PublishRemote -ReleaseNotes "Update" -RemotePassword "MySecurePassword123"

# 11. Remote publish with custom path
.\build-installer.ps1 -Version 1.0.0 -PublishRemote -ReleaseNotes "Update" -RemotePath "/var/www/downloads/"

# 12. Full custom remote configuration
.\build-installer.ps1 -Version 1.0.0 -PublishRemote -ReleaseNotes "Update" ^
-RemoteUser deploy ^
-RemoteServer 192.168.1.100 ^
-RemotePassword "pass123" ^
-RemotePath "/home/deploy/files/"

# 13. Skip build + Git publish (if installer already exists)
.\build-installer.ps1 -Version 1.0.0 -SkipBuild -GitPublish -ReleaseNotes "Hotfix"

# 14. Skip build + Remote publish
.\build-installer.ps1 -Version 1.0.0 -SkipBuild -PublishRemote -ReleaseNotes "Hotfix"

# 15. All parameters combined
.\build-installer.ps1 ^
-Version 2.5.3 ^
-GitPublish ^
-PublishRemote ^
-ReleaseNotes "Major update with new features" ^
-RemoteUser root ^
-RemoteServer 192.168.113.2 ^
-RemotePath "/data/server/mahn.it/software/audioprocessorandstreamer/" ^
-InnoSetupPath "C:\Program Files (x86)\Inno Setup 6\ISCC.exe"
```

Parameter Summary:

Parameter	Type	Required	Default	Description
-Version	string	No*	-	Version number (x.y.z format)
-SkipBuild	switch	No	false	Skip dotnet publish step
-InnoSetupPath	string	No	C:\Program Files (x86)\Inno Setup 6\ISCC.exe	Path to ISCC.exe
-GitPublish	switch	No	false	Commit, tag, and push to git
-PublishRemote	switch	No	false	Upload to remote server via SCP
-ReleaseNotes	string	No**	-	Release notes for git commit and autoupdate.json
-RemoteUser	string	No	root	SSH username
-RemoteServer	string	No	192.168.113.2	SSH server address
-RemotePassword	string	No	-	SSH password (uses key auth if omitted)
-RemotePath	string	No	/data/server/mahn.it/software/audioprocessorandstreamer/	Remote destination path

* Required when using `-GitPublish` or `-PublishRemote`
** Required when using `-GitPublish` or `-PublishRemote`

Auto Updates

autoupdates.json

```
{
  "version": "0.9.8",
  "downloadUrl": "https://www.mahn.it/software/audioprocessorandstreamer/AudioProcessorAndStreamer-Setup-0.9.8.exe",
  "releaseNotes": "Added improved debug logging"
}
```

Project Structure

```
AudioProcessorAndStreamer/
├─ Models/          # Data models and configuration
├─ ViewModels/      # MVVM view models
└─ Views/           # WPF views and dialogs
├─ Services/
│ ├─ Audio/          # WASAPI/ASIO capture services
│ ├─ Encoding/       # FFmpeg encoding management
│ ├─ Streaming/      # Stream processing pipeline
│ ├─ Vst/             # VST plugin hosting
│ └─ Web/             # HLS web server
├─ Infrastructure/
└─ Assets/
├─ FFmpeg/
└─ Plugins/          # VST plugins folder
└─ Presets/          # Stream preset files (.sts)
```

Technology Stack

- [.NET 8.0](#) - Application framework
- [WPF](#) - User interface
- [CommunityToolkit.Mvvm](#) - MVVM implementation
- [NAudio](#) - Audio capture (WASAPI/ASIO)
- [VST.NET](#) - VST plugin hosting
- [FFmpeg](#) - Audio encoding
- [ASP.NET Core](#) - Embedded web server
- [hls.js](#) / [dash.js](#) - Browser-based players

License

MIT License - See [LICENSE](#) file for details.

Acknowledgments

- [NAudio](#) - Audio library for .NET
- [VST.NET](#) - VST plugin hosting
- [FFmpeg](#) - Multimedia framework
- [hls.js](#) - HLS client library
- [dash.js](#) - DASH client library

✍️ Built with human ingenuity & a dash of AI wizardry

This project emerged from late-night coding sessions, unexpected inspiration, and the occasional debugging dance. Every line of code has a story behind it.

Found a bug? Have a wild idea? The issues tab is your canvas.

Authored By: 🧑‍💻 [Jurgen Mahn](#) with some help from AI code monkies [Claude code](#)

"Sometimes the code writes itself. Other times, we collaborate with the machines."

👉 Happy hacking, fellow explorer 👈