

# AudioScribe

## Windows Setup Product Spec

Repo: <https://github.com/christreadaway/audioscribe>

Branch: `claude/setup-audioscribe-windows-DreIB`

Primary file: `audioscribe_windows.py` (283 lines)

Launcher: `AudioScribe_Windows.bat`

## What AudioScribe Does

A local-only audio transcription app with a browser-based UI. The user uploads an audio file, it transcribes using WhisperX (an OpenAI Whisper variant), optionally identifies speakers via pyannote.audio, and saves a .txt transcript to `~/Downloads`. No data leaves the machine. Runs a Gradio web app at `http://127.0.0.1:7860`.

## Architecture (single file: `audioscribe_windows.py`)

Section	Lines	What it does
<code>torch.load</code> patch	29-36	Patches PyTorch 2.6+ to allow <code>weights_only=False</code> for pyannote
Config/constants	38-64	Paths, 21 supported languages, 6 Whisper model sizes
<code>load/save_token</code>	70-81	Read/write HuggingFace token from <code>~/.audioscribe_token.txt</code>
<code>get_device</code>	87-97	Detects CUDA vs CPU, sets float16 vs int8
<code>transcribe()</code>	103-196	Core: load model, transcribe, align, diarize, save .txt
<code>build_ui()</code>	202-261	Gradio Blocks UI with all inputs and transcript output
<code>main()</code>	267-283	Prints banner, launches Gradio on <code>127.0.0.1:7860</code>

## Dependencies

Package	Purpose
<code>torch</code> , <code>torchaudio</code>	ML framework, audio processing
<code>whisperx</code>	Transcription engine (wraps faster-whisper + ctranslate2)
<code>gradio==3.50.2</code>	Web UI (must be this version - see Known Issues)
<code>ffmpeg</code>	Audio decoding (system install, not pip)
<code>pyannote.audio</code>	Speaker diarization (pulled in by whisperx)

## Known Issues (Current State of the Branch)

### Issue 1: Gradio API mismatch on the branch

The branch currently has code written for NEW Gradio (5.x/6.x) API, but the project should use `gradio==3.50.2` (as documented in README and .bat file). WhisperX was built against old Gradio. Two lines need to be reverted on the branch:

- Line 213: change `sources=["upload"]` back to `source="upload"`

- Line 245: change buttons=["copy"] back to show\_copy\_button=True

## Issue 2: Python version requirement

WhisperX depends on ctranslate2==4.4.0 which requires Python <3.14. Python 3.11 is the recommended version. Python 3.12 also works. Python 3.13 and 3.14 do NOT work. If pip install whisperx fails with "No matching distribution found for ctranslate2", this is the cause.

## Issue 3: Virtual environment required

The .bat launcher expects a .venv folder. System-wide pip installs on a wrong Python version will not work. A venv created with py -3.11 is required.

## Correct Setup Steps (Windows)

---

### Step 1: Install Python 3.11

Download from <https://www.python.org/downloads/release/python-3119/>

Check "Add Python to PATH" during installation.

Verify: py -3.11 --version

### Step 2: Install FFmpeg

```
winget install FFmpeg
```

### Step 3: Clone the repo

```
git clone https://github.com/christreadaway/audioscribe.git
cd audioscribe
```

### Step 4: Create virtual environment

```
py -3.11 -m venv .venv
.venv\Scripts\Activate.ps1
```

If you get an execution policy error, run this first:

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned

### Step 5: Install dependencies

For NVIDIA GPU (faster):

```
pip install torch torchaudio --index-url https://download.pytorch.org/whl/cu118
pip install whisperx gradio==3.50.2
```

For CPU only:

```
pip install torch torchaudio whisperx gradio==3.50.2
```

### Step 6: Run

```
python audioscribe_windows.py
```

Or double-click AudioScribe\_Windows.bat.

## Files in Repo

---

File	Purpose
audioscribe_windows.py	Main application (283 lines)
AudioScribe_Windows.bat	Double-click launcher (expects .venv to exist)
README.md	Full user-facing docs (219 lines)
LICENSE	MIT license

## What to Do Next

---

- Revert the two Gradio API changes on the branch so the code matches gradio==3.50.2: line 213 (sources -> source) and line 245 (buttons -> show\_copy\_button).
- Install Python 3.11, create the .venv, install deps with gradio==3.50.2, and test end-to-end.
- Verify transcription works with a short audio file using the "tiny" model.
- Optionally test speaker diarization with a Hugging Face token.