



# VAICCS 1.0 beta 3

Official Manual



## Contents

Introduction.....	2
Minimum Specifications .....	2
Personal vs Commercial features.....	2
Interface.....	3
File .....	3
Models.....	4
View.....	4
Help.....	4
Main .....	5
Voice Profiles.....	6
Custom Words.....	6
Recommended Voice Profile Collection .....	7
Noise Cancelation .....	7
Automations.....	8
Quick Start.....	8
Automation features.....	8
Create Shows .....	8
Save Your Schedule .....	8
Automatic Recording .....	8
Auto-Start on Startup .....	9
Adding a Show.....	9
Editing a Show.....	9
Removing a Show .....	9
Model Download Manager.....	9
Autostart & Startup Modifiers .....	10
Release fixes .....	11
Attribution and Third-Party Components .....	12

# Introduction

Voice Artificial Intelligence Closed Captioning Software (VAICCS) is a graphical user interface optimized to send text from a serial output to your Closed Captioning Encoder.

## Minimum Specifications

Windows 11, 10 (Home and Pro)

Windows Server 2025, 2022, 2019, 2016

16GB of DDR4 or faster RAM

20 GB of Hard Drive space

Serial COM connection (RS232 preferred)

This software will not work on 32-bit Operating Systems

This software is currently not available on MacOS or Linux (yet 😊)

## Personal vs Commercial features

Features	Personal	Commercial
Voice to text	✓	✓
Voice Profiles	✓	✓
Custom Words	✓	✓
Serial Output	✓	✓
Noise Cancelation	✗	✓
Auto Scheduler	✗	✓
Shortcut Target Flags	✗	✓
Priority Support	✗	✓

# Interface

Like any Windows program, VAICCS has a set of menu bar drop menus and a set of tabs that will provide you with essential functions within this application.



## File

- **Save Settings** (Ctrl+S): write current settings to the last opened settings file, or prompt Save As if none exists.

- **Save Settings As...** (Ctrl+Shift+S): open a Save As dialog to export current GUI settings to a JSON file.

- **Options...**: open the Options dialog where you can set SRT caption duration, configure restricted-word (bleep) replacement modes and preview, and adjust auto-save transcript settings.

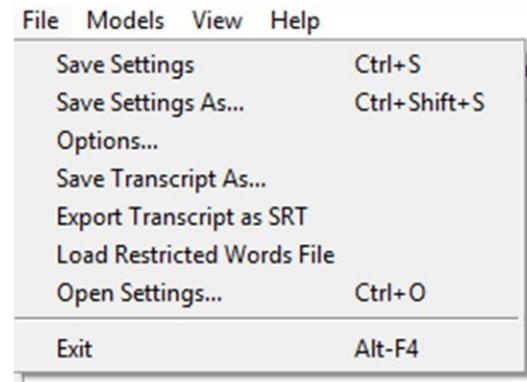
- **Save Transcript As...**: save the current transcript pane contents to a plain text ` .txt` file via Save As.

- **Export Transcript as SRT**: export the transcript to a simple SRT file (each non-empty line becomes a caption with a fixed duration set in Options).

- **Load Restricted Words File** (checkable): toggle and choose a restricted/bad-words file for the current session. If a file is already loaded the menu lets you Unload, Replace, or Cancel. The file is applied for the session only (see ` main.load\_bad\_words` ).

- **Open Settings...** (Ctrl+O): load a previously saved settings JSON into the session (applies values to the UI; does not automatically persist unless you Save Settings).

- **Exit** (Alt+F4): quit the app (also ensures the capture engine and automation scheduler are stopped cleanly).



## Models

- **Vosk Models...**: open the Vosk Model Manager dialog (lists languages and available models fetched from the Vosk model index, shows installed models in `models/` , and supports download/install operations). Use this to point the GUI at an unpacked Vosk model or to install a model into the local `models/` folder.
- **Hance Models...**: placeholder dialog for future Hance model management (opens the Hance models manager).

## View

- **Auto-scroll** (checkable): when enabled the transcript automatically scrolls to show the latest incoming captions. When the user scrolls up this is disabled automatically.
- **Jump to latest**: immediately jump the transcript to the end and re-enable auto-scroll.

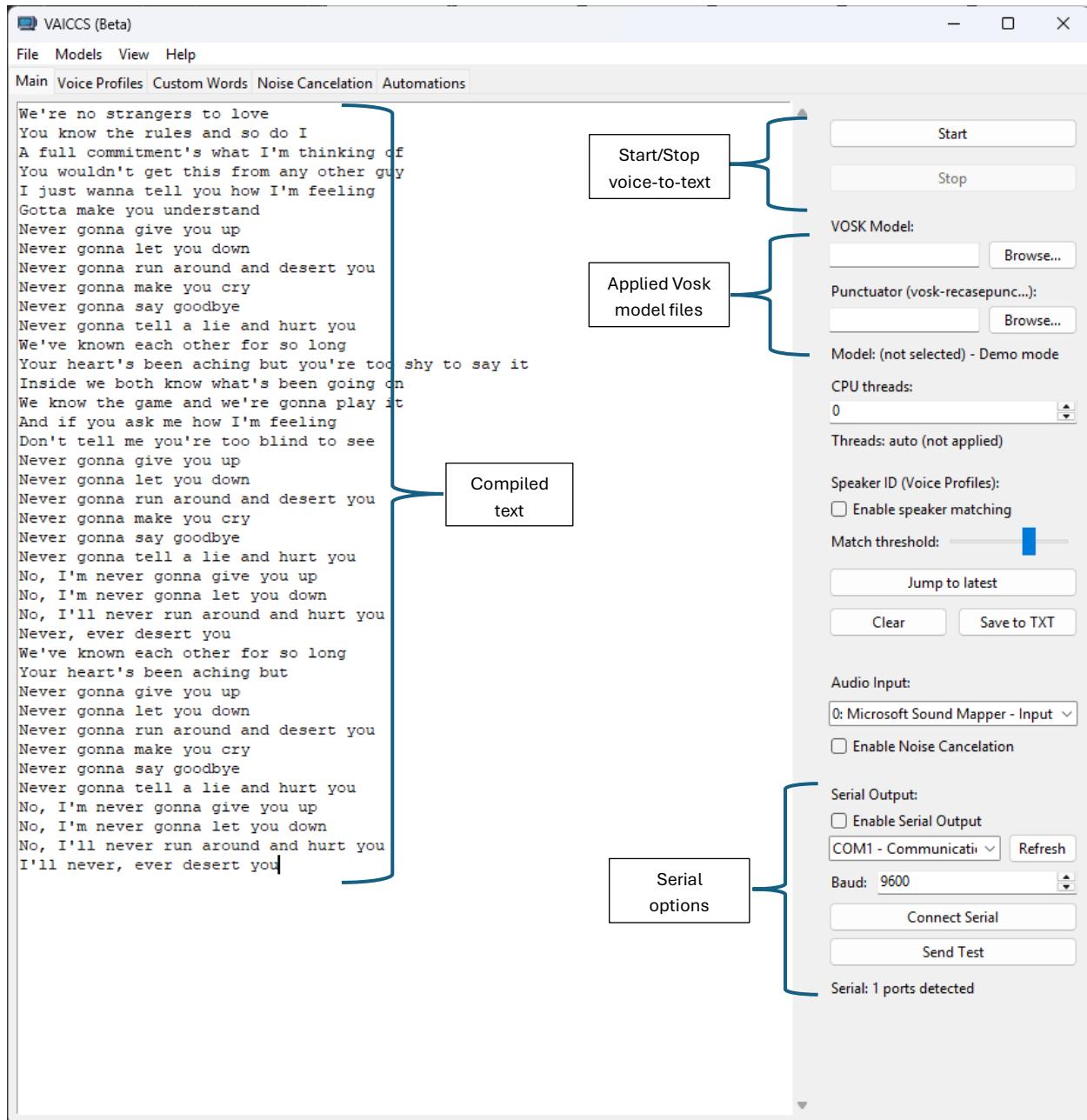
## Help

- **Activate**: open the activation dialog (activation/licensing UI).
- Check for Updates: Checks to see if there is an update available and gives the option to download and install a new version.
- **About**: show application information and build details in a small About dialog.

## Notes:

- Keyboard shortcuts implemented in the GUI include `Ctrl+S` , `Ctrl+Shift+S` (Save As), `Ctrl+O` (Open Settings), and a binding to ensure Exit runs on `Alt+F4` .
- Many menu actions open modal dialogs or file choosers; settings loaded from JSON are applied to the UI for the session and include optional embedded custom vocabulary and sample audio payloads when present.

# Main



- To use real recognition, download a compatible Vosk model and point the GUI to the unpacked model directory.
- If `vosk` or native dependencies are not available, the GUI can run in demo mode so you can still test audio capture and UI features.

## Voice Profiles

- Create profiles from WAV files in the Voice Profiles tab.
- Profiles are stored under `voice\_profiles/` and include embeddings and metadata for quick matching.
- **Create a profile:** open the **Voice Profiles** tab, enter a profile name, add one or more WAV files, then click **Create Profile**. Profiles are saved under `voice\_profiles/` as `\*.npy` embeddings with metadata in `voice\_profiles/index.json`.
- **Match audio:** use the GUI's matching tools or run `voice\_profiles.py match` from the CLI to find the top-K closest profiles.
- **Storage:** embeddings and metadata allow quick tests in CI using the included `voice\_profiles/` fixtures.

## Custom Words

from the GUI open the Custom Vocabulary dialog (File → Custom Vocabulary or the Tools/Options area where the app exposes the runtime vocabulary manager).

- **Add a word:** click “Add” (or “New Word”), enter the word text and (optionally) a pronunciation / lexicon string, then click “Save” or “Apply”.
- **Attach sample audio:** select a word and use “Add Sample” or “Attach WAV” to browse to a WAV file. The GUI copies the file into `custom\_vocab\_data/<safe\_word>/` (safe folder names use alphanumeric characters plus ` - ` and ` \_ `).
- **Apply to running recognizer:** click “Apply” or “OK” in the dialog to immediately send the updated word list to the running recognizer; changes are used as a runtime grammar to bias recognition.
- **Persist changes:** use “File → Save Settings As...” to persist GUI settings (the custom vocab path is saved via `CustomVocabManager.serializable\_path()`). The word/pronunciation map itself is stored in `custom\_vocab.json` next to `custom\_vocab.py` .

- **Where files live:** words and pronunciations are stored in `custom\_vocab.json` ; WAV samples are stored in `custom\_vocab\_data/<safe\_word>/` under the project root.

This feature biases on-device recognition at runtime — it does not rebuild the Vosk model graph automatically. For permanent model changes, export the lexicon lines and follow Vosk/Kaldi model customization procedures.

## Recommended Voice Profile Collection

### **Clip length:**

Aim for 5–15 seconds per clip.

Shorter clips (<3s) often don't provide enough acoustic detail.

Longer clips (>30s) are fine, but you can break them into smaller segments for more embeddings.

## Noise Cancelation

The Noise Cancelation tab exposes controls to install and manage a real-time noise-filter that runs before audio reaches the recognizer.

- **Quick toggle (Main tab):** there is a checkbox on the Main tab labeled "Enable Noise Cancelation" to quickly enable or disable the filter while the app is running. This toggle invokes the same install/uninstall code used by the Noise tab.
- **Hance model file:** use the "Browse..." button on the Noise Cancelation tab to select an optional Hance model file. If provided, the UI will attempt to initialize the Hance SDK with that model when installing the filter.
- **Install Noise Filter:** installs the noise-processing wrapper into the app. On success the status label changes to show the installed state and the quick-toggle is set to enabled.
- **Uninstall:** restores the original audio callback, sets the status to "Not installed", and disables the quick-toggle.

- **Status & Notes:** the tab shows a status label (e.g., "Not installed", "Installed (model:xyz)", or "Disabled in Personal/Eval mode"). A small note explains that Hance SDK integration is attempted if available; otherwise the app uses a built-in RMS-based noise-gate fallback.

- **License gating:** Hance/noise controls are disabled in Personal/Eval mode. To re-enable controls without restarting the app (see activation/Help → Activate).

User tip: prefer the tab's Install/Uninstall controls when selecting a model file; use the Main tab toggle for quick on/off during a session.

## Automations

The **Show Automations** feature allows you to schedule shows to automatically start and stop recording at specified times on selected days of the week.

### Quick Start

1. **Open the Automations Tab** - Click the "Automations" tab at the top of the window
2. **Add a Show** - Click "+ Add Show" and fill in the details
3. **Click Apply** - The show will now run automatically on its scheduled days/times

### Automation features

#### Create Shows

- Add multiple shows with different schedules
- Name each show
- Pick which days it should run (Monday through Sunday)
- Set start and end times (5-minute intervals)

#### Save Your Schedule

- Save all your shows to a JSON settings file
- Load them anytime (same file or different file)
- Share your schedule with others

#### Automatic Recording

- Shows start recording automatically at the scheduled time
- Shows stop recording automatically at the end time

- Works 24/7 while the app is running

## Auto-Start on Startup

- Use command-line flag to load schedule automatically
- Perfect for server/headless deployments
- Can auto-start the first matching show

## Adding a Show

1. Click “+ Add Show”
2. Enter the show name
3. Check the days you want (Mon-Sun)
4. Select start time from dropdown
5. Select end time from dropdown
6. Click “Apply”

Show Automations:

+ Add Show Apply

Show 1

Show Name: New Show

Days:  Mon  Tue  Wed  Thu  Fri  Sat  Sun

Start: 12 : 00 PM

End: 1 : 00 PM Remove

## Editing a Show

1. Modify any field
2. Click “Apply”

## Removing a Show

1. Click the “Remove” button on the show
2. Click \*\*Apply\*\*

## Model Download Manager

VAICCS provides a download manager of the official Vosk and Hance AI models. First press the refresh button and for Vosk select a language then select a model. It downloads directly from the official Vosk models website and automatically put the model in the models folder that is located next to the VAICCS executable.

# Autostart & Startup Modifiers

VAICCS supports passing simple modifiers to the executable via the shortcut Target so users can autostart with a saved configuration.

Supported modifiers (append to the shortcut Target after the exe path):

- ` -save:<file>` — load a saved settings JSON file before the GUI starts. If the file sits next to the exe, pass just the filename. If located elsewhere, pass the full quoted path.
- ` -autostart:true` or ` -autostart` — after loading ` -save` (if present), automatically start the caption engine. The GUI still validates the model path and may prompt or fall back to demo mode.
- ` -show\_error` — if a native import (e.g., Vosk) fails, the launcher will display the import log to help diagnose issues.

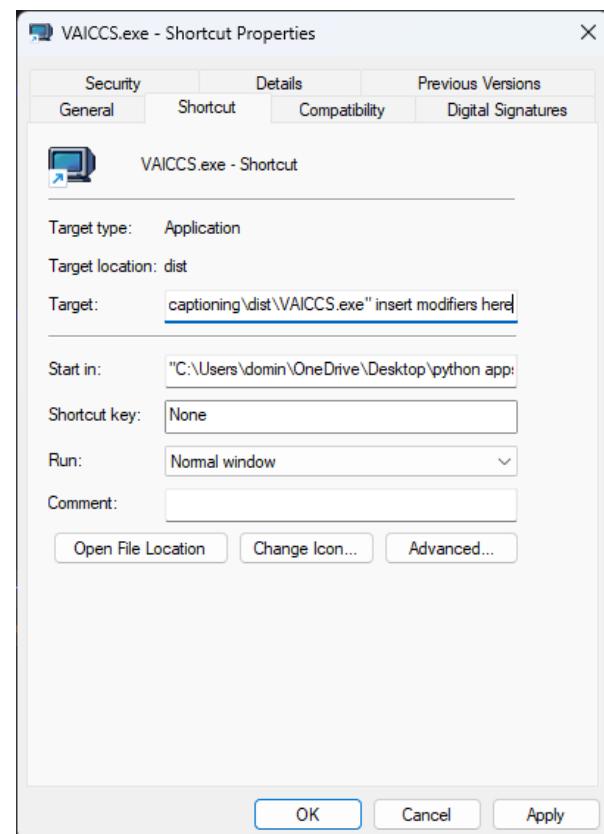
Examples (Windows Shortcut Target):

```
"C:\Program Files\VAICCS\VAICCS.exe" -save:"settings.json" -autostart:true -show_error
```

```
"C:\Program Files\VAICCS\VAICCS.exe" -  
save:"C:\Users\User\Desktop\settings.json" -  
autostart -show_error
```

Notes about ` -save` path resolution:

- If a relative filename is provided (no drive or leading slash), the launcher searches in this order and uses the first match:
  1. EXE directory (so a `settings.json` next to the exe is found)
  2. Current working directory
  3. Application module directory (project root)



## Release fixes

### VAICCS 1.0 Beta 1

- Initial beta release

### VAICCS 1.0 Beta 2

- Added "Check for Updates" feature under Help dropdown in Top bar
- Added showing the version of the program on the splash screen and Help: About window
- Fixed the serial output feature. There would be no output seen when the "Send test" button or when spoken word is said.

### VAICCS 1.0 Beta 3

- Fixed "Check for Updates" dialogue so that it would show the Release notes properly.
- Fixed if a user wanted to cancel a download of a update. Previously the cancel button didn't work.
- Put the proper VAICCS icon on the top left hand corner in the "Check for Updates" and "download" windows.
- Made improvements on the Punctuator. NOTE, using the “gigaspeech” model file will not work properly alongside the Punctuator if that has a model loaded in as well.
- Added the feature where one word at a time goes though the serial output when Vosk outputs words sentences/paragraphs at a time (previously whole paragraphs would be pushed to a closed caption encoder all at once)
- When a word is being sent through the serial output, it gets highlighted in the main tab
- Added multiple highlighter color options
- Adjusted default main window size
- Compiled using Python 3.13.11

## Attribution and Third-Party Components

This software uses VOSK, licensed under the Apache License, Version 2.0. You may not remove or alter the following notice:

VOSK Speech Recognition Toolkit

Copyright © [Alpha Cephei]

Licensed under the Apache License, Version 2.0

<http://www.apache.org/licenses/LICENSE-2.0>

The Apache License grants you rights to use, modify, and distribute VOSK, subject to its terms.

This software also uses Hance, licensed under the MIT License.

Copyright © 2025 Hance AS

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

<https://mit-license.org/>

<https://pypi.org/project/hance/>