Mobile Artificial Intelligence Speech Engine Manual

v3.0.0

March 9, 2026

Contents

1 Introduction

Maise (Mobile Artificial Intelligence Speech Engine) is a free and open-source Android application that provides high-quality, fully on-device text-to-speech synthesis (TTS) and automatic speech recognition (ASR). All processing runs locally using ONNX Runtime — no internet connection is required and no audio or text data ever leaves your device.

Maise integrates with Android at the system level:

- The **TTS component** is registered as an Android `TextToSpeechService`, meaning any app that uses the standard `TextToSpeech` API will automatically use Maise once it is set as the system default — no per-app integration is needed.
- The **ASR component** is registered as an Android `RecognitionService`, making it compatible with any app that uses the standard `SpeechRecognizer` API.

1.1 Companion App

Maise is the voice companion to **Maid** (Mobile Artificial Intelligence Distribution), a free and open-source AI chat application for Android. Maid can be found at `https://github.com/Mobile-Artificial-Intelligence/maid`.

2 How It Works

2.1 Text-to-Speech

The TTS pipeline converts raw text into spoken audio entirely on-device through four stages:

1. **Text normalisation** — Raw input text is cleaned and normalised. Numbers are expanded to words, abbreviations are resolved, and punctuation is handled so that the phonemizer receives well-formed prose.

2. **Phonemization** — Open Phonemizer converts the normalised text into a sequence of IPA phonemes.

3. **Neural synthesis** — The phoneme sequence and a per-voice style embedding are fed into Kokoro, a high-quality multi-lingual neural TTS model running under ONNX Runtime, which produces a raw PCM audio waveform.

4. **Streaming playback** — Synthesis and playback run concurrently in a producer-consumer pipeline so that audio begins playing before the entire input has been processed.

Audio output is 24 kHz mono 16-bit PCM, played directly through the device speaker or any connected audio output.

2.2 Automatic Speech Recognition

The ASR pipeline transcribes spoken audio to text through three stages:

1. **Recording** — Audio is captured from the microphone at 16 kHz mono 16-bit PCM. Voice Activity Detection (VAD) using the WebRTC VAD algorithm automatically determines when speech ends, so you do not need to manually stop recording. The maximum recording duration is 30 seconds.

2. **Log-mel spectrogram** — A Whisper-compatible 80-band log-mel spectrogram is computed on-device from the captured audio.

3. **Transcription** — The spectrogram is processed by distil-whisper/distil-small.en, an encoder-decoder Transformer model, using greedy decoding to produce the transcribed text.

3 The App

The Maise app has a two-tab interface: **TTS** for text-to-speech and **ASR** for automatic speech recognition. Tap the tab labels at the top of the screen to switch between them.

3.1 TTS Tab

The TTS tab lets you select a voice and preview speech synthesis directly in the app.

- **Voice** — A dropdown listing all available voices. Select a voice to use it for both in-app preview and the system-wide TTS service. Your selection is saved automatically and persists across restarts.
- **Text field** — Enter any text you want to hear spoken. The field supports multiple lines.
- **Speak** — Tap to synthesise and play the entered text. The status line below the button shows the current state: *Loading models...*, *Ready*, *Synthesizing...*, *Playing...*, or *Done*.
- **Open TTS Settings** — Opens the Android system TTS settings page directly, where you can set Maise as the preferred engine (see Section 5.1).
- **Report mispronunciation** — Opens the Maise GitHub issues page so you can report words that are synthesised incorrectly.

3.2 ASR Tab

The ASR tab lets you test on-device speech recognition.

- **Start / Stop Recording** — Tap once to begin recording; tap again to stop early. Recording also stops automatically when silence is detected after speech. The status line shows the current state: *Loading model...*, *Ready — tap to record*, *Recording...*, *Transcribing...*, or *Done*.
- **Transcription output** — The read-only text field displays the recognised text after each recording. If no speech is detected it shows *(no speech detected)*.
- **Open Keyboard Settings** — Opens the input method settings where you can enable the Maise keyboard (see Section 5.3).
- **Open Speech Settings** — Opens the language and input settings where you can set Maise as the preferred speech recognition service (see Section 5.2).

4 Voices

Maise ships with 68 Kokoro voices across multiple languages. All voices are bundled with the app — no downloads are required after installation.

Language	Voices
English (US)	alloy, aoede, bella, heart, jessica, kore, nicole, nova, river, sarah, sky, adam, echo, eric, fenrir, liam, michael, onyx, puck, santa
English (GB)	alice, emma, isabella, lily, daniel, fable, george, lewis
German	dora, alex, santa
French	siwis
Greek	alpha-f, beta-f, omega-m, psi-m
Italian	sara, nicola
Japanese	alpha-f, gongitsune, nezumi, tebukuro, kumo
Portuguese (BR)	dora, alex, santa
Chinese (Simplified)	xiaobei, xiaoni, xiaoxiao, xiaoyi, yunjian, yunxi, yunxia, yunyang

The default voice is `en-US-heart-kokoro`. To change it, use the Voice dropdown on the TTS tab.

5 Setup

5.1 Text-to-Speech

To use Maise as the system TTS engine for all apps on your device:

1. Open the Android **Settings** app.
2. Navigate to **Accessibility** → **Text-to-Speech Output**.
3. Under *Preferred engine*, select **Maise**.

Alternatively, tap **Open TTS Settings** on the TTS tab to jump directly to this screen.

Once Maise is set as the preferred engine, any app that calls the Android `TextToSpeech` API — including Maid — will use Maise automatically without any further configuration.

5.2 Automatic Speech Recognition

To use Maise as the system speech recognition service:

1. Open the Android **Settings** app.
2. Navigate to **Apps** → **Default Apps** → **Assist & voice input**.
3. Under *Speech recognizer*, select **Maise**.

Alternatively, tap **Open Speech Settings** on the ASR tab. You must also grant the **Microphone** permission to Maise the first time it is used for recognition.

5.3 Voice Input Method (Keyboard)

Maise also registers as an input method that provides a voice dictation button inside any text field. To enable it:

1. Open the Android **Settings** app.
2. Navigate to **System** → **Language & input** → **On-screen keyboard** (or **Virtual keyboard**).
3. Tap **Manage keyboards** and enable **Maise**.

Alternatively, tap **Open Keyboard Settings** on the ASR tab. Once enabled, a globe or keyboard icon in any text field allows switching to Maise for voice input.

6 Building from Source

6.1 Cloning the Repository

```
git clone https://github.com/Mobile-Artificial-Intelligence/maise.git
```

6.2 Building

Build a release APK using the Gradle wrapper:

```
./gradlew :app:assembleRelease
```

The output APK will be at:

- Release: `app/build/outputs/apk/release/app-release.apk`
- Debug: `app/build/outputs/apk/debug/app-debug.apk`

The app targets **ARM64-v8a** devices running Android 8.0 (API 26) or later.