

ReadSpeaker 

Runtime TTS Unity plugin



Unity[®]

Product sheet

Leverage the power of text to speech (TTS) and deploy it natively in your game projects. Create and edit speech for audio cues and UI narration on the fly with dynamic runtime TTS. Be independent: Prototype speech and dialogue without voice actors. Above all, be inclusive: Enhancing your game with AI-powered voices means you can improve accessibility for gamers with vision and reading disabilities.

The applications of voice technology and TTS for game development are practically endless. TTS is a scalable tool that enables you to keep on building. It makes development in Unity that much easier and quicker, and will help you to stay ahead of the curve.



[Click here to watch our explanation videos](#)

- **Effortlessly add embedded runtime TTS into your existing project with an easy-to-use and flexible API**

- Only 2 lines of code are required to get started
- Advanced features and functionality for precise control of the TTS creation process

- **Customize speech output with Unity components**

- Associate voices with characters in your game
- Adjust voice/volume/pitch/speed and more

- **Integrates seamlessly with Unity's audio system**

- Use Unity's Audio Sources to take full control of shaping the audio output
- Smoothly achieve 3D audio spatialization effects and more

- **Multi-platform support**

- Our plugin offers a seamless experience across supported platforms, it just works!
- **Currently supported platforms:**
 - ✓ Windows 32/64-bit
 - ✓ Linux 32/64-bit
 - ✓ Android
 - ✓ PlayStation®5
 - ✓ PlayStation®4
 - ✓ XBOX Series XIS

- **Fully documented**

- **Demo Scene included**

Other OSs can be supported on request

Currently supported Unity versions:

- 2018.x • 2021.x
- 2019.x • 2022.x
- 2020.x • 2023.x

Voices and Languages

Our portfolio of voices and languages keeps growing. ReadSpeaker currently offers text to speech voices in **40+** languages and **115+** voices. For a complete list of available languages [visit our languages web page](#).

Use SSML to tweak the audio output exactly to your liking

SSML gives you more control over how a voice reads text out loud, specifically in terms of prosody. SSML can also be used to insert pauses/breaks of a certain length, add phonetic transcriptions, and switch voices or languages in the same text. As a game dev you can utilize this specialized TTS output and decide how to effectively use it for the best in-game experience.

Testimonial



“Having the ability to change speaker, volume, pitch, and speed is very powerful, and is something I would like to expose for the use of players.”

“Without a TTS plugin like this we would be left guessing what audio samples we would need to generate, and how they would play back.”

“The plugin allows us to experiment without the need to lock our decisions, which is a very powerful tool to have the privilege to use.”

Conor Bradley

FOUNDER AND DIRECTOR AT SOFT LEAF STUDIOS

Example use cases:

- ❑ Enhance accessibility in your games i.e. by providing narration of on-screen UI elements
- ❑ Prototype voice acting during early stages of development
- ❑ Narrate text of in-game chat messages in multiplayer games
- ❑ Read aloud procedurally generated text and dialog with zero latency

Do you want to take speech, dialog, and accessibility to the next level in your game development projects? Has this information fired up your curiosity even more? Then don't hesitate to [contact us](#), we're more than happy to answer any questions you may have.

[Contact us today!](#)

Technical specifications

The plugin leverages ReadSpeaker SpeechEngine SDK for Windows/Linux and ReadSpeaker SpeechEngine SDK Embedded for Android:

	speechEngine SDK	speechEngine SDK Embedded
Synthesis type	Standard, Neural Standard Not recommended: Neural Lite	Standard, Neural Standard, Neural Lite
OS	<ul style="list-style-type: none"> Windows 10 and higher; <ul style="list-style-type: none"> - last tested Windows 11 Linux <ul style="list-style-type: none"> - CentOS 7.x / 8.x - Ubuntu 18.x - RHEL 7.x / 8.x 	<ul style="list-style-type: none"> Android
Supported architecture	Intel 32 and 64 bit (x86/x86_64)	ARM 32 and 64 bit
CPU properties	Intel x86/64 1 GHz or higher	ARM 32/64 bits 400 MHz or higher
RAM	6 ~ 30 MB	6 ~ 30 MB
Voice footprint	7 ~ 600 MB per voice	4 ~ 600 MB per voice
Runtime Memory	<ul style="list-style-type: none"> Standard: 35 MB ~ 90 MB Neural Standard, Neural Lite: 5 MB ~ 10MB 	<ul style="list-style-type: none"> Standard: 35 MB ~ 90 MB Neural Standard, Neural Lite: 5 MB ~ 10MB
Supported input formats	<ul style="list-style-type: none"> Plain text SSML 	<ul style="list-style-type: none"> Plain text SSML
Supported character encoding for text input	All voices: UTF8 Multibyte code set for Japanese: 932, Korean: 949, Chinese: 936, THAI: 874, Russian: 1251, RON/SLK/ CES/POL : 1250, HIN : 57002, ARB : 1256, HEB : 1255, other Western: 1252	All voices: UTF8 Multibyte code set for Japanese: 932, Korean: 949, Chinese: 936, THAI: 874, Russian: 1251, RON/SLK/ CES/POL : 1250, HIN : 57002, ARB : 1256, HEB : 1255, other Western: 1252
Supported audio output formats	<ul style="list-style-type: none"> PCM (Wave) 16-bit linear 8-bit linear 8-bit A-law 8-bit μ-law 4-bit Dialogic ADPCM 	<ul style="list-style-type: none"> PCM (Wave) 16-bit linear 8-bit linear 8-bit A-law 8-bit μ-law 4-bit Dialogic ADPCM
Features:	<ul style="list-style-type: none"> DSP* User dictionary IPA support Voice switch 	<ul style="list-style-type: none"> DSP* User dictionary IPA support Voice switch