RT-Voice PRO

Hearing is understanding

Documentation

crosstales LLC

Date: 19. May 2017
Version: 2.7.2

Table of Contents

Thank you for buying our asset "RT-Voice PRO"!

If you have any questions about this asset, send us an email at rtvoice@crosstales.com.
Please don't forget to rate it or write a little review – it would be very much appreciated.

1. Overview

Have you ever wanted to make software for people with **visual impairment** or who have **difficulties reading**? Do you have lazy **players** who **don't like to read** too much? Or do you even want to **test** your game's **voice dialogues** without having to pay a voice actor yet? With RT-Voice this is very easily done – it's a major time saver!

RT-Voice uses the computer's (already implemented) TTS (text-to-speech) voices to turn the **written lines** into **speech** and dialogue at **run-time**! Therefore, **all text** in your game/app can be **spoken** out loud to the player.

And all of this without any intermediate steps: The transformation is **instantaneous** and **simultaneous** (if needed)!

2. Features

- **Instantaneous transformation** form text-to-speech! No intermediate steps.
- Since the audio is generated during run-time, it **saves** a lot of **space**!
- **No need** to **voice act** for yourself during the test-phase of your game.
- **Multiple** voices at **once** (e.g. scenes at a public square with many people talking **simultaneously**).
- Fine-tune your voices with **rate**, **pitch** and **volume**
- **Current word**, **visemes** and **phomenes** on Windows and iOS (incl. mark functions)
- Generated audio can be **stored** to **files**. Those files can be **reused** inside **Unity**
- 1-n synchronized **loudspeakers** for a **single** AudioSource **origin**.
- Simple **sequence** and **dialog** system
- **No** performance **overhead**!
- Powerful **API** to get **maximum control** as a developer
- Runs on **all Unity platforms**[1]!
- **MaryTTS** support
- **PlayMaker** actions!
- Contains a **Proxy** manager for **Internet** connections
- **Internet availability tester** included!
- **Test-Drive** the voices inside the **Editor**!
- Extensive **demo** scenes, **documentation**, **API** and **support**!
- Full **C# source code**
- We are **committed** to all our assets! This means, we will add new features over time!

---

1   **WebGL** & **WebPlayer** are currently in **BETA**, which means that it won't work out-of-the-box - you might have to work out a setup solution of your own (like CORS or policy server)! We will improve the asset and documentation in the upcoming releases to help you further.

2.1. Supported third-party assets

- [SALSA](#)
- [Localized Dialogs & Cutscenes (LDC)](#)
- [Dialogue System for Unity](#)
- [THE Dialogue Engine](#)
- [PlayMaker](#)
- [Adventure Creator](#)
- [LipSync](#)
- [SLATE](#)
- [Cinema Director](#)
- [uSequencer](#)
- [Quest System Pro](#)
- [NPC Chat](#)

2.2. Platform-specific limitations

2.2.1. Windows

- Native pitch has no effect
- Native rate is internally limited to 20 logarithmic distributed steps
- .NET 4.0 or higher must be installed
- Minimum Windows version: 7

2.2.2. MacOS

- Native pitch has no effect
- Native volume has no effect
- No current words, phonemes and visemes
- Minimum macOS version: 10.6

2.2.3. Android

- Only one native voice at the time (can be solved by generating audio)
- No current words, phonemes and visemes
- Minimum Android version: 4.0.3 (API 15)

2.2.4. iOS

- Only one active native voice at the time
- No audio generation
- Current word but no phonemes and visemes
- Minimum iOS-version: 7.1

2.2.5. WSA (UWP)

- No native audio (only generated audio files)
- Native rate has no effect
- Native pitch has no effect
- Native volume has no effect
- No current words, phonemes and visemes
- Minimum SDK-version: 8.0

2.2.6. MaryTTS

- No native audio (only generated audio files)
- Native rate has no effect
- Native pitch has no effect
- No current words, phonemes and visemes
- Minimum MaryTTS-version: 5.0

3. Demonstration

The asset comes with many demo scenes to show the main usage.

3.1. Speech

This demo scene shows how to transform written lines into speech. Choose your preferred voice.
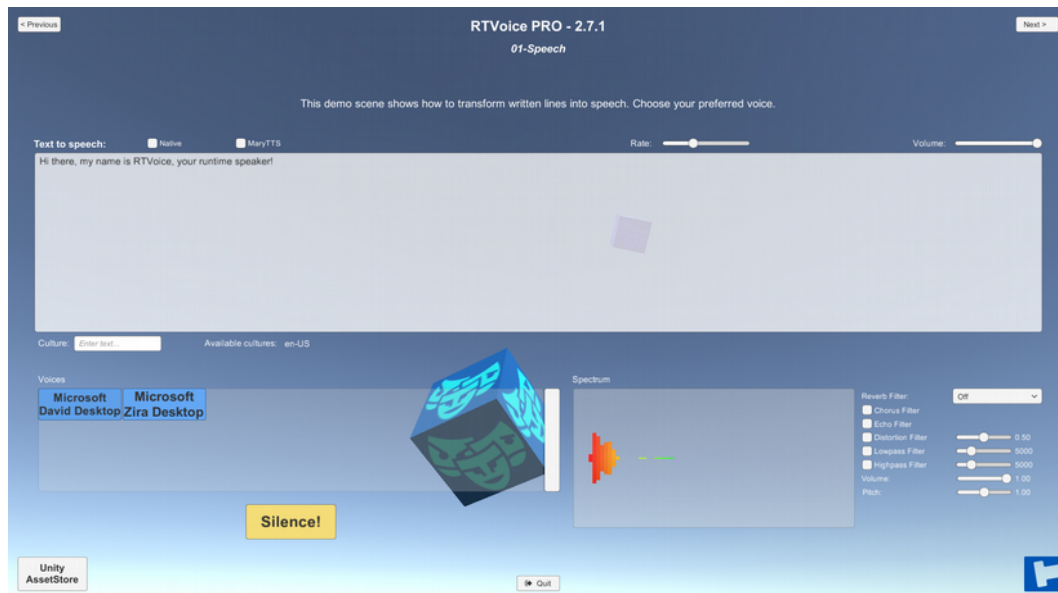
3.2. Dialog

In this demo scene you can act out a dialogue between two "people". You can choose a different voice for both participants.
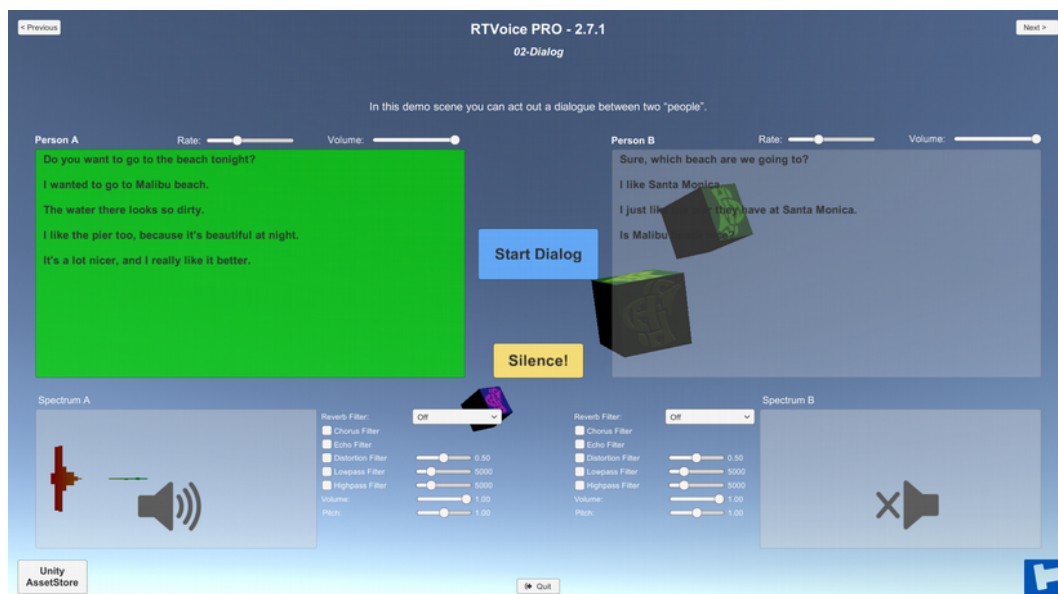
3.3. SimpleNative

The "SimpleNative" scene shows the easiest way for native audio.

3.4. Simple

The "Simple" scene shows the easiest and recommended way for most purposes with generated audio.

3.5. 3DAudio

This scene demonstrates 3D positioned and looped audio.

Needs the Unity Standard Characters (Assets → Import Packages → Characters).

3.6. Loudspeakers

This scene demonstrates 3D positioned loudspeakers with only one audio origin (looped).
Needs the Unity Standard Characters (Assets → Import Packages → Characters).

3.7. SendMessage

This scene shows the usage of Unity's "SendMessage".

3.8. Sequencer

This scene shows the usage of our simple sequencer.

3.9. Native and PreGenerated

These two scenes are showing how you can build applications with exact timing between audio and animations (e.g. lip sync).

3.10.   SpeechText

This scene shows how to speak or store generated audio (see the result inside the folder "_generatedAudio").

4. Setup

RT-Voice has global settings under "Edit\Preferences…" and under "Tools\RTVoice PRO\Configuration…":
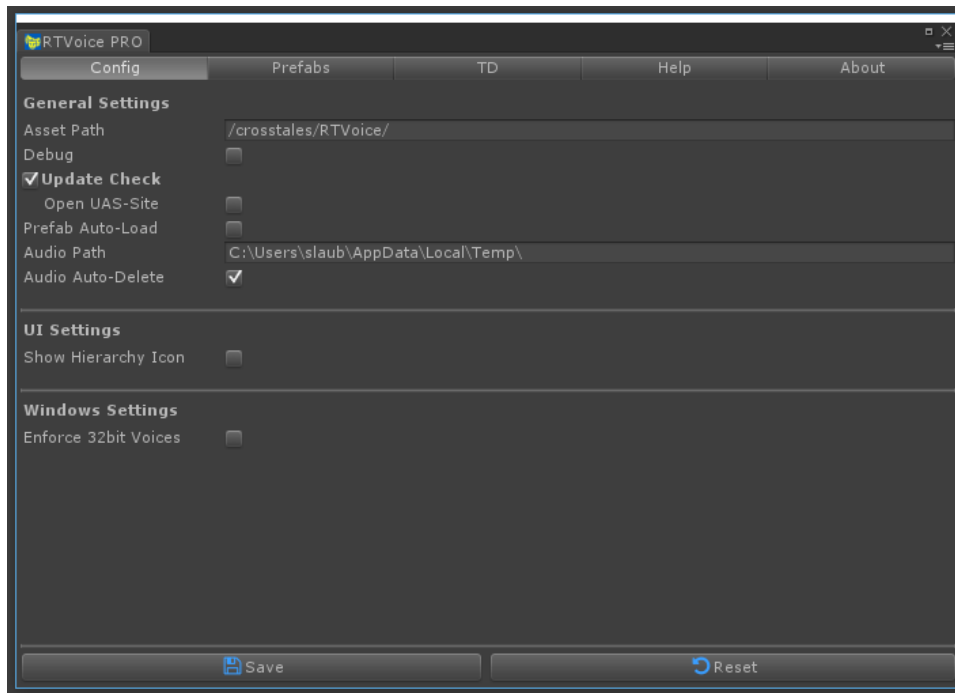
4.1. Add RT-Voice

There are four ways to add RT-Voice to your project:

1. Add the prefab **RTVoice** from Assets/crosstales/RTVoice/Prefabs to the scene
2. Or go to *Tools => RTVoice PRO => Prefabs =>* **RTVoice**
3. Right-click in the *hierarchy-window => RTVoice PRO =>* **RTVoice**
4. Add it from the Prefabs-tab:
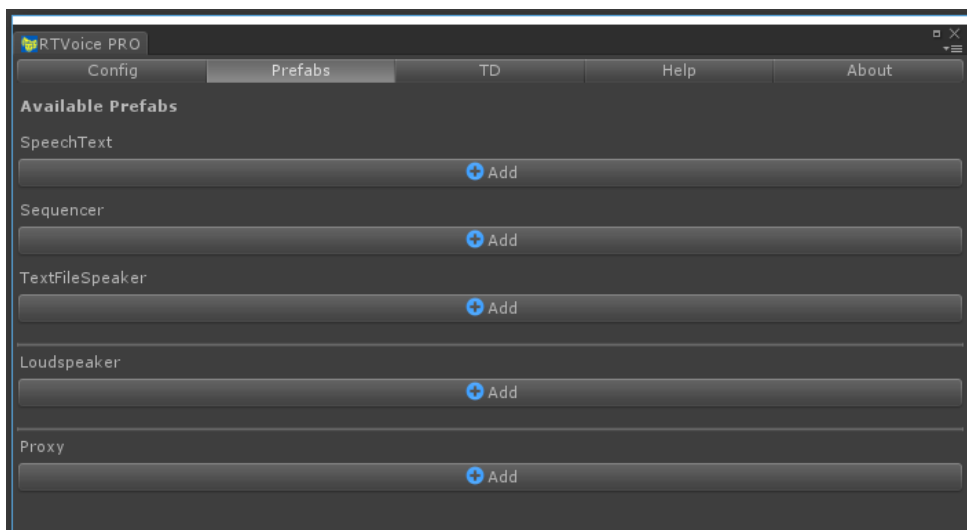
4.2. Other components

The other components can be added in the same way as "RTVoice".

4.2.1. SpeechText

Allows to speak and store generated audio.
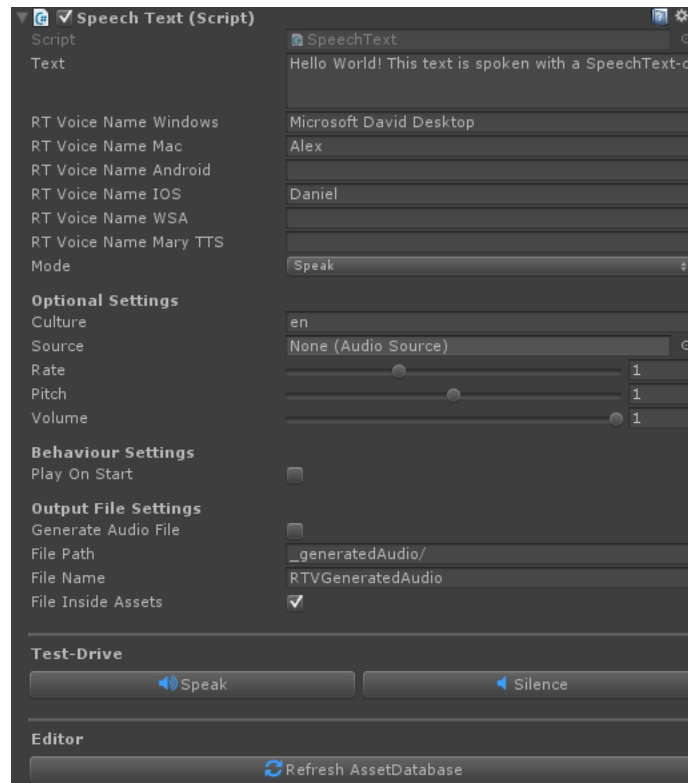
4.2.2. Sequencer

Simple sequencer for dialogues.

4.2.3. TextFileSpeaker

Allows to speak text files.
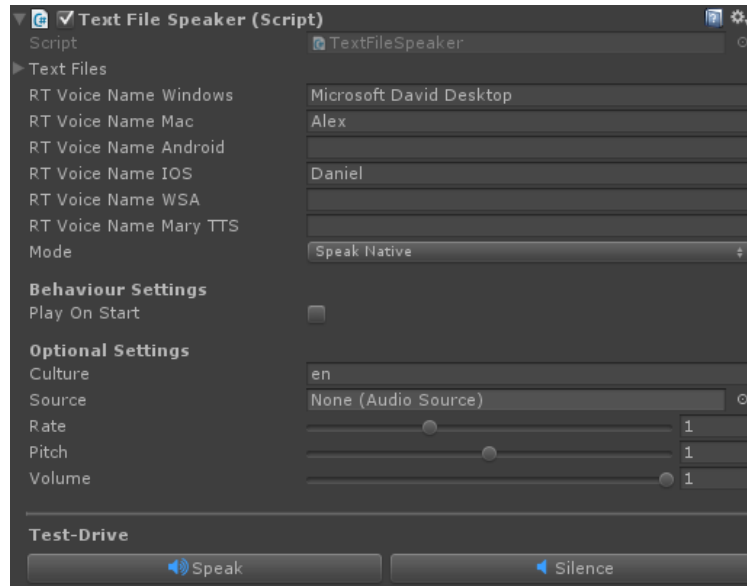
4.2.4. Loudspeaker

Loudspeaker for an AudioSource.

This is useful to use a speech on multiple locations in the game.

4.2.5. InternetCheck

Checks the Internet availabilty.

4.2.6. Proxy

Handles HTTP/HTTPS Internet connections via proxy server.

4.3. Differences between standard and native mode

In the **standard** mode the TTS-system of your OS will **convert** your text to an **audio** file and return it to **Unity** as an "**AudioSource**" for further use (like changing the volume, pitch etc.).

On the other hand, the **native** mode **delegates** the speech-task **entirely** to the underlying TTS-system (outside of Unity). You are **losing** some **control** but it uses slightly **less performance**.

We clearly **recommend** using the **standard** mode.

4.4. Speaker.cs vs. LiveSpeaker.cs

"Speaker.cs" is the main class of "RT-Voice" and presents the API via static methods. "LiveSpeaker.cs" on the other hand is a wrapper for "Speaker.cs" and presents the API as normal C#-instance via public methods. The main usage of "LiveSpeaker.cs" is as a receiver for "SendMessage"-calls.

4.5. MaryTTS

MaryTTS is an open-source TTS with a server, client and many voices.

It enables TTS under all Unity platforms.

You can customize everything by yourself, just follow their guides:

http://mary.dfki.de/

To enable MaryTTS, simply check "MaryTTS" in the RTVoice-component and configure the URL and port.

4.5.1. Important

The default server in RT-Voice is the test server from MaryTTS.

Never release a product with the **default configuration** and install your own server (local/remote)!

5. API

The asset contains various methods and the most important are explained here.

Make sure to **include** the **name space** in your relevant source files:

```
using Crosstales.RTVoice;
```

5.1. Speaker

The "Speaker.cs" is a singleton and contains the following static methods.

5.1.1. Speak

Speaks a text with a given voice and optional AudioSource.

For example:

```
//Immediately speak "hello world" with the first available voice
Speaker.Speak("hello world", audioSource);
```

```
//Immediately speak "hello world" with the first English voice (if available
else it uses the first voice on your OS)
Speaker.Speak("hello world", audioSource, Speaker.VoiceForCulture("en"));
```

```
// Prepare speak "hello world" with the first available voice (without
AudioSource.Play() - this is up to you). With this technique, you can prepare
all audio texts of a scene and you can modify the AudioSource as you like!
Speaker.Speak("hello world", audioSource, null, false);
```

5.1.2. SpeakNative

Speaks a text with a given voice.

For example:

```
//Speak "hello world" with the first available voice
Speaker.SpeakNative("hello world");
```

```
//Speak "hello world" with the first English voice (if available else it uses
the first voice on your OS)
Speaker.SpeakNative("hello world", Speaker.VoiceForCulture("en"));
```

5.1.3. Silence
Silence all active TTS-voices.
Example:

```
//Silence all voices
Speaker.Silence();
```

5.1.4. Voices

```
List<Voice> Voices
```
Returns all available voices (alphabetically ordered by 'Name').

5.1.5. VoicesForCulture

```
List<Voice> VoicesForCulture(string culture)
```
Returns all available voices for a given culture (alphabetically ordered by 'Name').

5.1.6. VoiceForCulture

```
Voice VoiceForCulture(string culture, int index)
```
Returns the voice for the given culture and index.

5.1.7. VoiceForName

```
Voice VoiceForName(string name)
```
Returns the voice for the given name or null if not found.

5.1.8. Cultures

```
List<string> Cultures
```
Returns all available cultures (alphabetically ordered by 'Culture').

5.2. Callbacks

There are various callbacks available. Subscribe them in the "Start"-method and unsubscribe in "OnDestroy".

5.2.1. Speak start and complete

`SpeakStart(SpeakEventArgs e);`

`SpeakStart` **OnSpeakStart;**

Triggered whenever a speak is started.

`SpeakComplete(SpeakEventArgs e);`

`SpeakComplete` **OnSpeakComplete;**

Triggered whenever a native speak is completed.

5.2.1. Current word (native, Windows and iOS only)

`SpeakCurrentWord(CurrentWordEventArgs e);`

`SpeakCurrentWord` **OnSpeakCurrentWord;**

Triggered whenever a new word is spoken (native, Windows and iOS only).

5.2.2. Current phoneme (native, Windows only)

`SpeakCurrentPhoneme(CurrentPhonemeEventArgs e);`

`SpeakCurrentPhoneme` **OnSpeakCurrentPhoneme;**

Triggered whenever a new phoneme is spoken (native mode, Windows only).

5.2.3. Current viseme (native, Windows only)

`SpeakCurrentViseme(CurrentVisemeEventArgs e);`

`SpeakCurrentViseme` **OnSpeakCurrentViseme;**

Triggered whenever a new viseme is spoken (native mode, Windows only).

5.2.4. Speak audio generation start and complete

`SpeakAudioGenerationStart(SpeakEventArgs e);`

`SpeakAudioGenerationStart` **OnSpeakAudioGenerationStart;**

Triggered whenever a speak audio generation is started.

`SpeakAudioGenerationComplete(SpeakEventArgs e);`

`SpeakAudioGenerationComplete` **OnSpeakAudioGenerationComplete;**

Triggered whenever a speak audio generation is completed.

5.2.5. Provider change

```
ProviderChange(string provider);
```

```
ProviderChange OnProviderChange;
```

Triggered whenever a provider chamges (e.g. Windows to MaryTTS).

5.2.6. Errors

```
ErrorInfo(string info);
```

```
ErrorInfo OnErorInfo;
```

Triggered whenever an error occurs.

5.2.7. Example

Get informed when a speak starts and completes:

```
void Start() {
 // Subscribe event listeners
 Speaker.OnSpeakStart += speakStartMethod;
 Speaker.OnSpeakComplete += speakCompleteMethod;

 Speaker.SpeakNative("Hello world!");
}

void OnDestroy() {
 // Unsubscribe event listeners
 Speaker.OnSpeakStart -= speakStartMethod;
 Speaker.OnSpeakComplete -= speakCompleteMethod;
}

private void speakStartMethod(SpeakEventArgs e) {
 Debug.Log("speakStartMethod: " + e);
}

private void speakCompleteMethod(SpeakEventArgs e) {
 Debug.LogWarning("speakCompleteMethod: " + e);
}
```


5.3. Complete API

For more details, please see the RTVoice-api.pdf

6. Additional voices

RT-Voice works great with third-party voices (e.g. IVONA, Cereproc etc.).

6.1. Windows

All SAPI5-compatible voices are supported. Microsoft also provides a wide range of voices for different languages:

https://www.microsoft.com/en-us/download/details.aspx?id=27224

To install and use those voices follow this manual:

http://superuser.com/a/872573

6.1.1. Important

Don't install those Microsoft voices or RTVoice won't work:

- hui hui

- hun yee

- han han

6.2. MacOS

Apple delivers many voices for different languages. To add or customize them, follow the tutorial below:

http://osxdaily.com/2011/07/25/how-to-add-new-voices-to-mac-os-x/

6.3. Android

You can add various voices on your Android phone:

http://www.geoffsimons.com/2012/06/7-best-android-text-to-speech-engines.html

There is also a possibility to download high quality voices:

http://www.androidauthority.com/google-text-to-speech-engine-659528/

6.4. iOS

You can only change the quality of the installed voices:

https://support.apple.com/en-us/HT202362

6.5. WSA (UWP)

No information so far. If you know a working guide, please let us know.

6.6. MaryTTS

Follow those guides:

http://mary.dfki.de/

7. Third-party support (PlayMaker etc.)

„RT-Voice" supports various assets from other publishers. Please import the desired packages from the „3rd party"-folder.

8. Upgrade to new version

Follow this steps to upgrade your version of "RT-Voice PRO":

1. Update "RT-Voice PRO" to the latest version from the "Unity AssetStore"
2. Inside your project in Unity, go to menu "File" => "New Scene"
3. Delete the "Assets/crosstales/RTVoice" folder from the Project-view
4. Import the latest version from the "Unity AssetStore"

9. Use the source files

1. Create an empty scene inside your project
2. Delete the "Assets/crosstales/RTVoice/Plugins/RTVoice.dll"
3. Import the "Assets/crosstales/RTVoice/Sources.unitypackage"

10.   Important notes

After this setup, the "RT-Voice" is ready to use. It is important to know that it uses the **singleton**-pattern, which means that **once instantiated**, the "RT-Voice" will **live until** the application is **terminated**.

Remember: it must be instantiated before you try to access it! Otherwise it's not possible to use it.

11. Problems, improvements etc.

If you encounter any problems with this asset, just <u>send us an email</u> with a problem description and the invoice number and we will try to solve it.

12.   Release notes

See "VERSIONS.txt" under "Assets/crosstales/RTVoice/Documentation".

13.   Credits

The icons are based on <u>Font Awesome</u>.

14.   Contact and further information

crosstales LLC
Weberstrasse 21
CH-8004 Zürich

Homepage:	https://www.crosstales.com/en/portfolio/rtvoice/
Email:	rtvoice@crosstales.com
AssetStore:	https://goo.gl/qwtXyb
Forum:	http://goo.gl/Z6MZMl
Documentation:	https://www.crosstales.com/media/data/assets/rtvoice/RTVoice-doc.pdf
API:	http://goo.gl/6w4Fy0
Windows-Demo:	https://www.crosstales.com/media/data/assets/rtvoice/downloads/RTVoice_demo_win.zip
Mac-Demo:	https://www.crosstales.com/media/data/assets/rtvoice/downloads/RTVoice_demo_mac.zip
Android-Demo:	https://www.crosstales.com/media/rtvoice/RTVoice.apk

15.   Our other products

Bad Word Filter	The "Bad Word Filter" (aka profanity or obscenity filter) is exactly what the title suggests: a tool to filter swearwords and other "bad sentences".
DJ	DJ is a player for external music-files. It allows a user to play his own sound inside any Unity-app. It can also read ID3-tags.
Radio	Have you ever wanted to implement radio stations but don't want (or can't) pay an horrendous amount of money? Whenever you like to provide good sound from famous artists for your games or apps, tune in on one of the uncountable Internet MP3/OGG radio stations available for free.
TPS	Turbo Platform Switch is a Unity editor extension to reduce the time for assets to import during platform switches. We measured speed improvements up to 50x faster than the built-in switch in Unity.
TrueRandom	True Random can generate "true random" numbers for you and your application. The randomness comes from atmospheric noise, which for many purposes is better than the pseudo-random number algorithms typically used in computer programs.