

# ACUAH editor manual

ver.3.0.0

for ACUAH ver.2.0.0

riemgoshawk (@project\_ACUAH)



音声認識・音声応答

VRM0.0/VRM1.0・VRoidHub 対応

ユーザーカスタマイズ可能な 3D キャラアシスタント



ACUAH

Customizable 3D character assistant application for all creators.



# Terms of Use

1. The copyright of "ACUAH editor" (hereinafter referred to as the "Software") belongs to the provider of the Software "riemgoshawk" (twitter:@project\_ACUAH).
2. In no event shall the provider of this Software be liable for any direct or indirect damages to the user of this Software (hereinafter referred to as the "User") arising from the use of this Software.
3. The User is responsible for understanding and complying with the terms and conditions of use for the character model data and voice data used in this Software.
4. The dialogue scenario data created by the User in this Software and the file list data created to load them in the Android application "ACUAH" (data created with the file name of "acuah\_local\_list") may be used and distributed free of charge for both personal and commercial use under the user's responsibility.
5. In the event that a dispute arises between the User and a third party or the right holder of such data due to infringement of intellectual property rights as a result of the User's use of character model data or voice data in this Software, the User shall resolve such dispute at its own expense and shall not make any claim against the provider of this application. The User shall not make any claim to the application provider.
6. The following acts are prohibited with respect to this software.
  - Redistribution of this Software and all files required to run it (including tutorial data).
  - Reverse engineering or other analysis of this Software or data created by this Software
  - Other annoying acts against the Software provider and third parties using the Software and data created by the Software.
7. The Software is provided to you on an "as is" basis, and the Software provider makes no warranty of any kind, including, but not limited to, that the Software is free of errors, bugs, or other defects, that it will be useful for your purposes, or that it will operate completely.

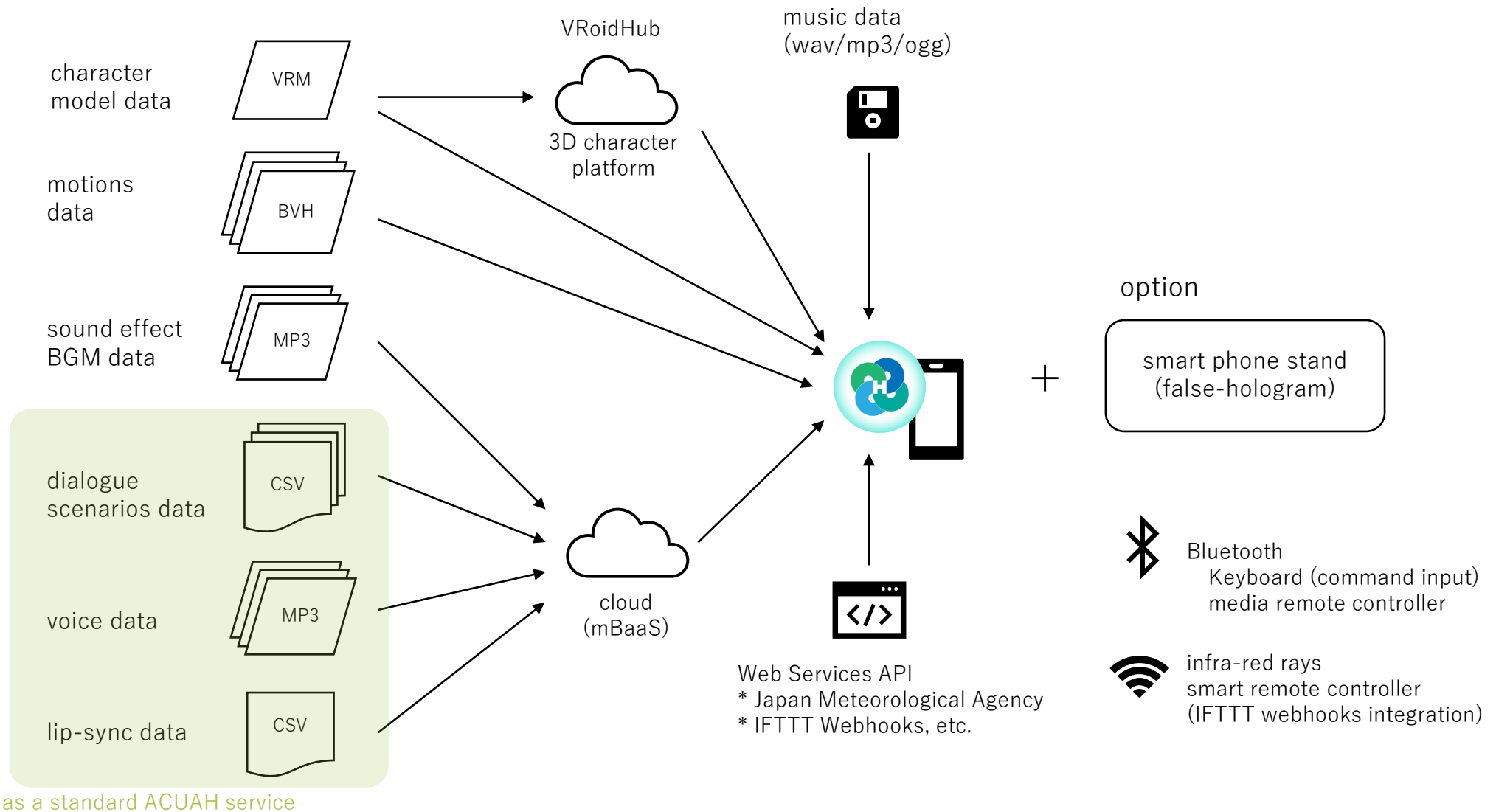


# How to install ACUAH editor

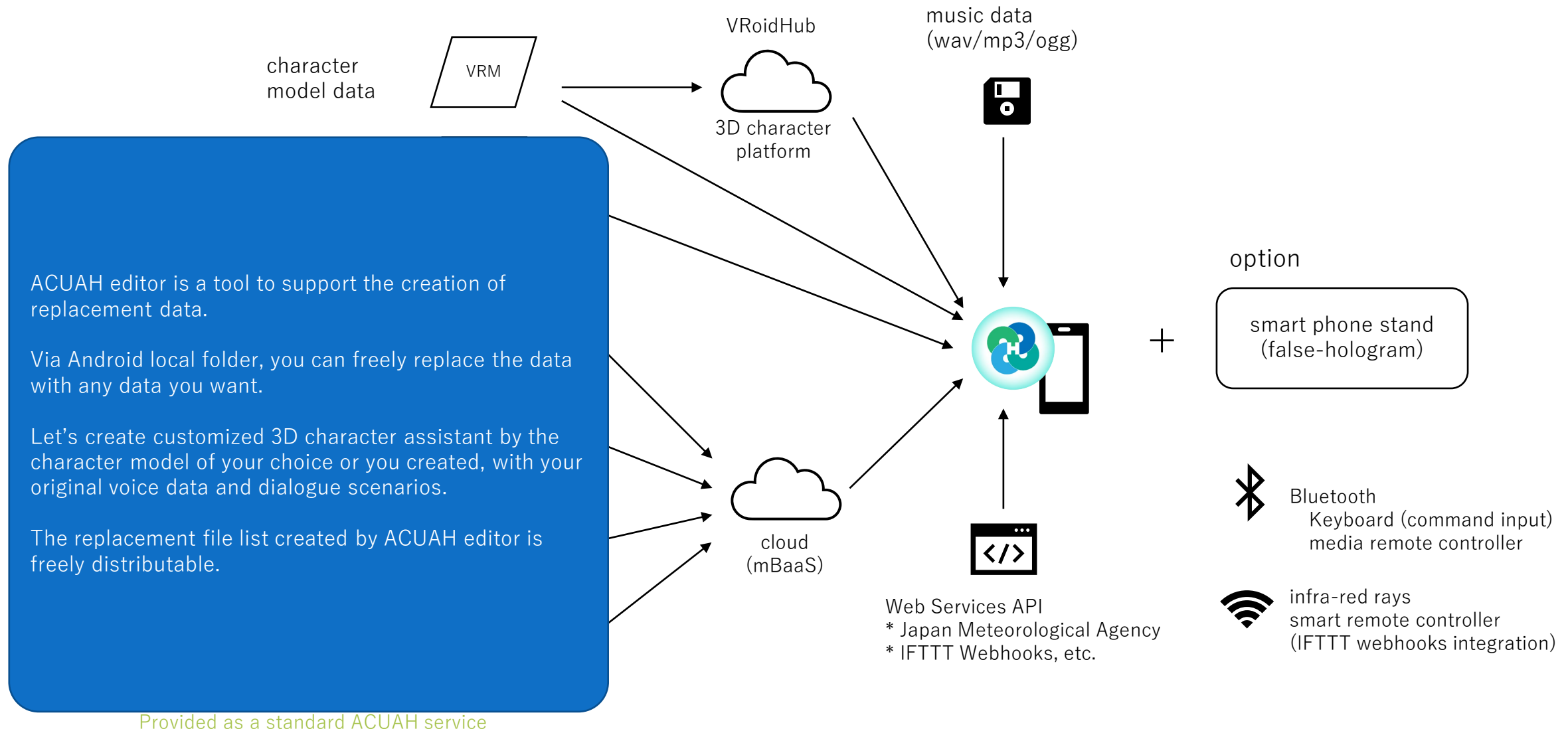
- ACUAH editor is an application for Windows (x64).
- Please agree to the "Terms of Use" on p.3 of this manual and run "ACUAH\_editor\_installer.exe" to install the application.



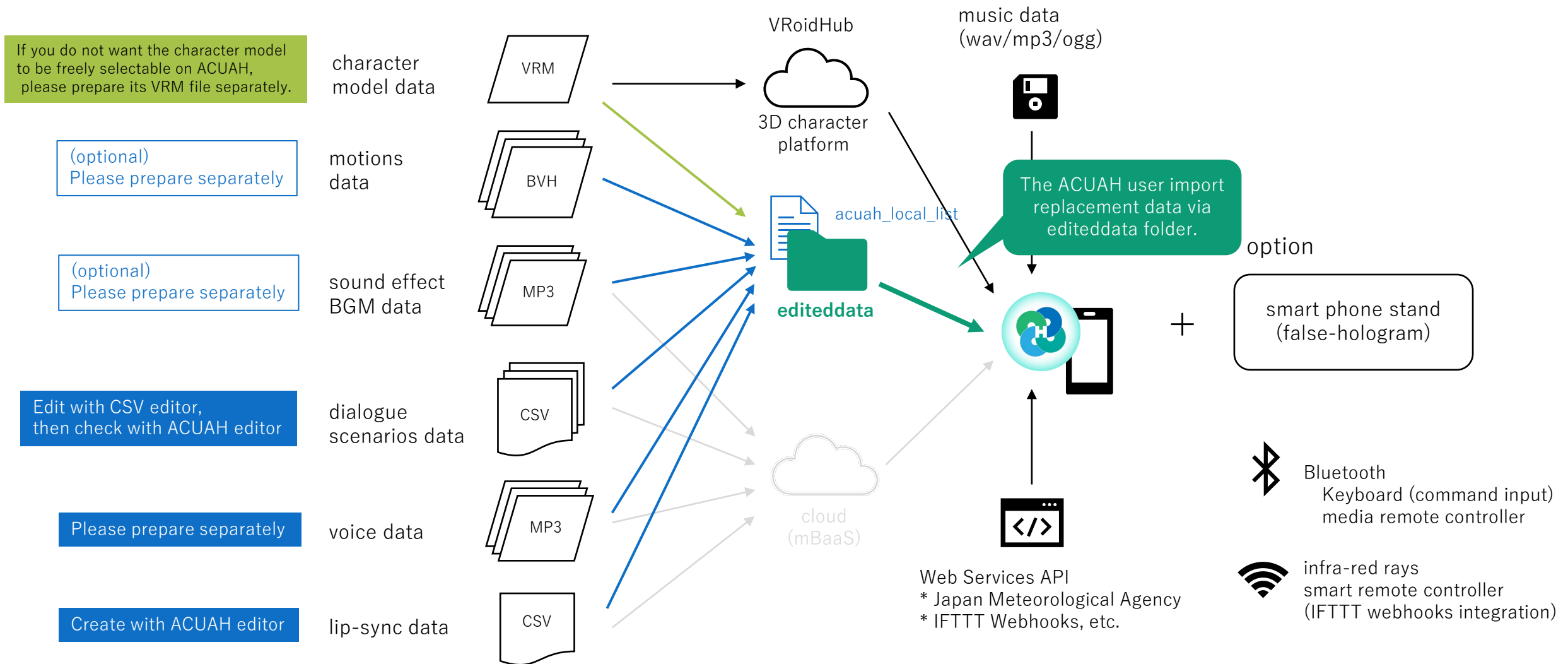
# How ACUAH works



# What is ACUAH editor?



# What is ACUAH editor?



# What is ACUAH editor?

The process of creating dialogue scenario, voice data and lip-sync data is as follows.

If the data is not created by you, the intellectual property rights of the creator of the data exist.

Please be aware of the conditions of use of the data in each process.

- (1) Think about dialogue scenarios.
- (2) Prepare the voice data according to the dialogue scenario.
- (3) (When locking dialogue scenario, voice data and character) Prepare VRM file.
- (4) Create dialogue scenario data (CSV file) from (1) and (2) with CSV editor.
- (5) Create lip-sync data from the data in (2) in ACUAH editor.
- (6) ACUAH editor checks the consistency of files (2) to (5), creates a list of files for replacement and a list of files to be replaced and create the “editeddata” folder.





# Required material data

## Character model data

- Character model data format is VRM <sup>\*1</sup>.
- Since ACUAH and ACUAH editor are linked with VRoidHub<sup>\*2</sup>, you can select your favorite characters from VRoidHub if you do not have character model data by yourself.
- If you want to create your own character models, there are some great applications, VRoid Studio<sup>\*2</sup> and VRoidMobile<sup>\*2</sup>, so please create your favorite ones.
- In BOOTH (<https://booth.pm/>), there are many creators who are commissioned to make VRM character models, so if it is difficult for you to make them yourself, why not ask them to do it for you?
- Character model data in VRM format can be used in ACUAH via the local folder of the Android phone in which ACUAH is installed, via VRoidHub.

\*1 General Incorporated Association VRM Consortium Avatar File Format for VR

<https://vrm.dev/>

\*2 pixiv Inc. VRoidHub, VRoidStudio, VRoidMobile

<https://vroid.com/>



# Required material data

## Dialogue scenario data

- CSV format (Comma Separated Value) file.
- The procedure to create dialogue scenario data for ACUAH is explained in this manual.
- To create the data, you will need a CSV editor (free software such as Cassava Editor is fine) as described below. ) is required.
- Eventually, you will need to convert it to the **character set "UTF-8", "CR+LF" and “without BOM”**.



# Required material data

## Voice data / lip-sync data

- **Both MP3 format and WAV format data are required.**

(Because ACUAH editor (Windows) cannot play MP3 format audio data.)

- Prepare voice data of words you want the character to speak according to the dialogue scenario data.
- Since ACUAH cannot be operated while the character is speaking, we recommend that the voice data be as short as possible, only a few words.
- You can use your own voice, voice of voiceroid, or ask voice actors to create voice data for you. Please feel free to combine your favorite voice data.
- If you would like to request voice data from a voice actor, how about requesting it from a voice request site such as Voices (<https://www.voices.com/>) or iikoe (<https://iikoe.org/>)?
- In ACUAH, lip-sync data should be prepared in advance as a set with voice data. (This application not use real-time waveform analysis of voice data and not be able to create a lip-sync data automatically).
- Lip-sync data can be created using ACUAH editor. It is explained in this manual.



# Required material data

## (Optional) motions data

- **BVH format data can be loaded.**
- Some applications, such as motion capture, support BVH output.
- See also the Carnegie Mellon University Motion Capture Database (<http://mocap.cs.cmu.edu/>).

## (Optional) Sound effect and BGM (background music) data

- **WAV (.wav), MP3 (.mp3), and OGG (.ogg) files can be loaded.**
- Files can be specified in a dialog scenario and played as sound effects or background music.

# Tools required in addition to ACUAH editor

We guide you to the following free software, but you can use any other application.

- **Android / Voice / Text character conversion / Final operation check**

ACUAH

- This is used to create a dialogue scenario.
- You can check the recognized text string for the voice input.  
(Use "edit reaction" on the configuration menu 3).

- **CSV Editor**

Cassava Editor

- It is used to edit dialogue scenarios.
- You can also edit CSV files in Microsoft Excel, but you will need to change the file format to UTF-8N/without BOM after outputting the CSV file.

- **Audio file creation and conversion tool**

mp3DirectCut / bearaudio ( <https://www.bearaudiotool.com/jp/> ) and so on.

- For trimming MP3 files (cutting unnecessary parts such as silence).

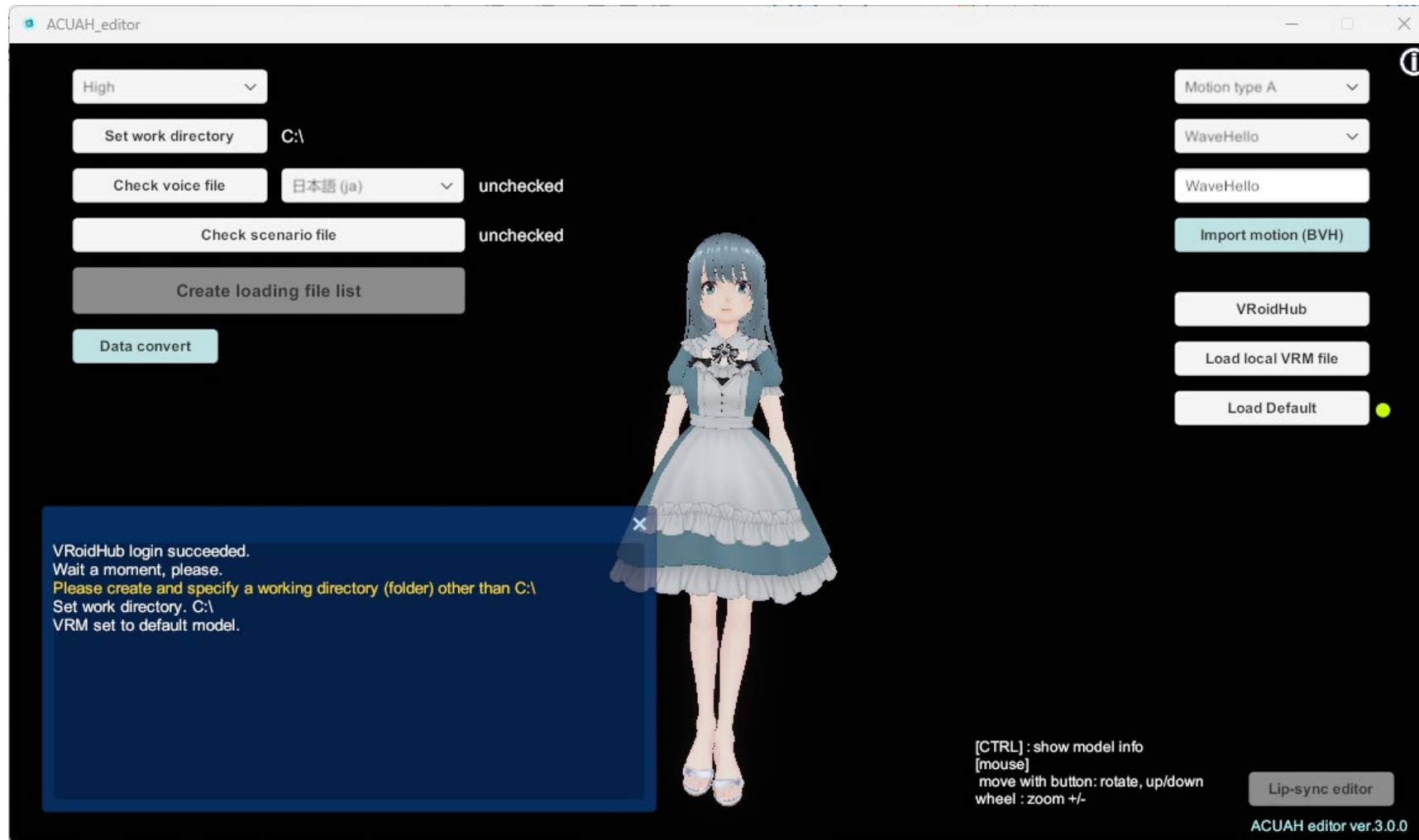
EcoDecoTooL

- ACUAH editor cannot play MP3 files, so you need to create a WAV file with the same name.
- It is possible to convert by specifying the sound pressure (dB), when the volume is low.



# ACUAH editor

<Main menu screen>



# ACUAH editor

## <Main menu screen> Basic Operation

**Image quality setting**  
(Low) Very Low  
(High) Ultra

**Set work directory**

**Check voice file**  
voice file and lip-sync data consistency check

**Check scenario file**  
consistency check of voice data and dialogue scenario files.

**Check loading file list**  
Create editeddata folder for loading ACUAH file list

**Message window**  
Check execution results and the status of the file loading and other information are shown.

**Lip-sync data language settings**

**About ACUAH editor**

**For motion type select drop-down list**

**For motion confirmation drop-down list**

**motion name text copy field**

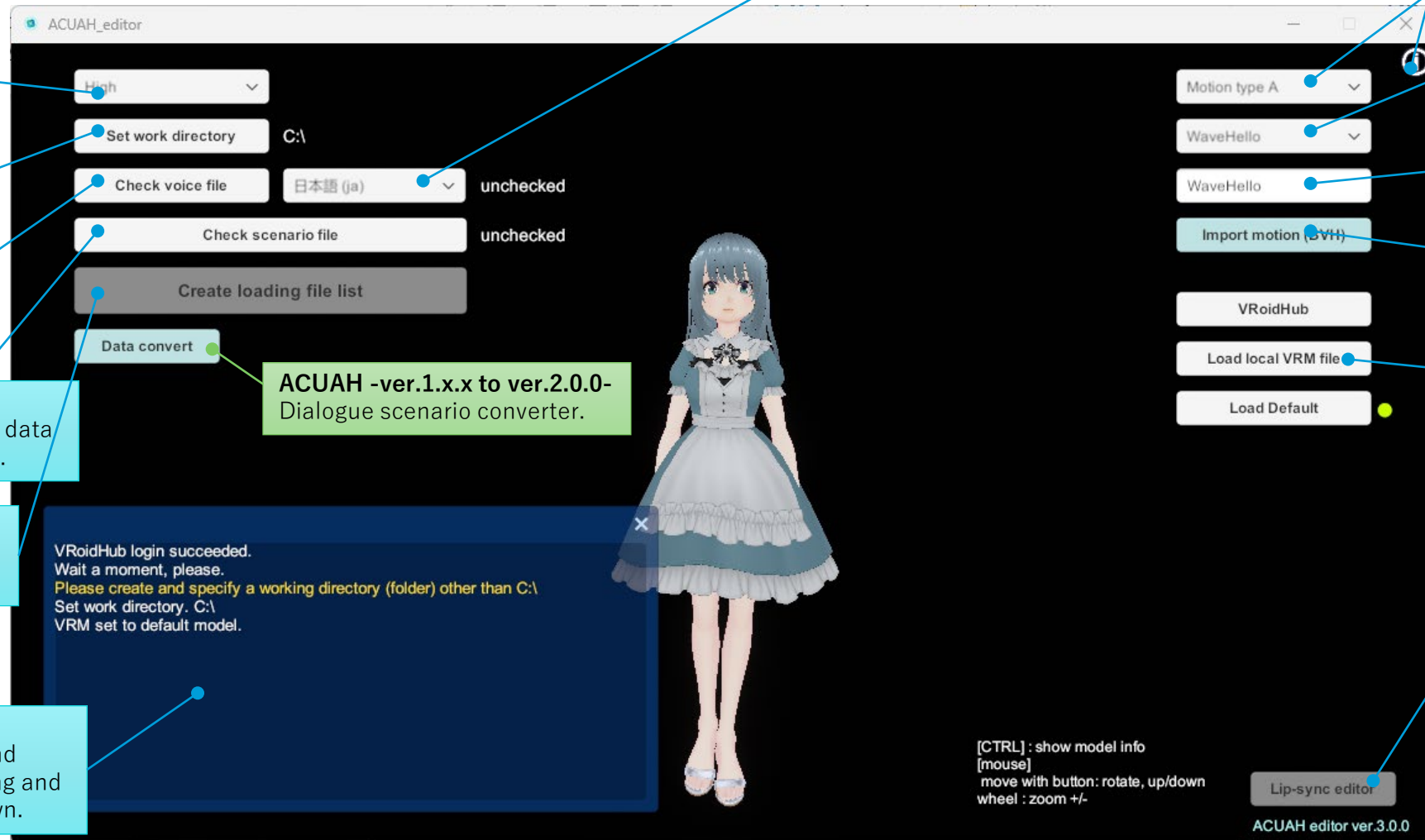
**import motion data (BVH)**

**3D character Platform Specification**

● : you are selecting

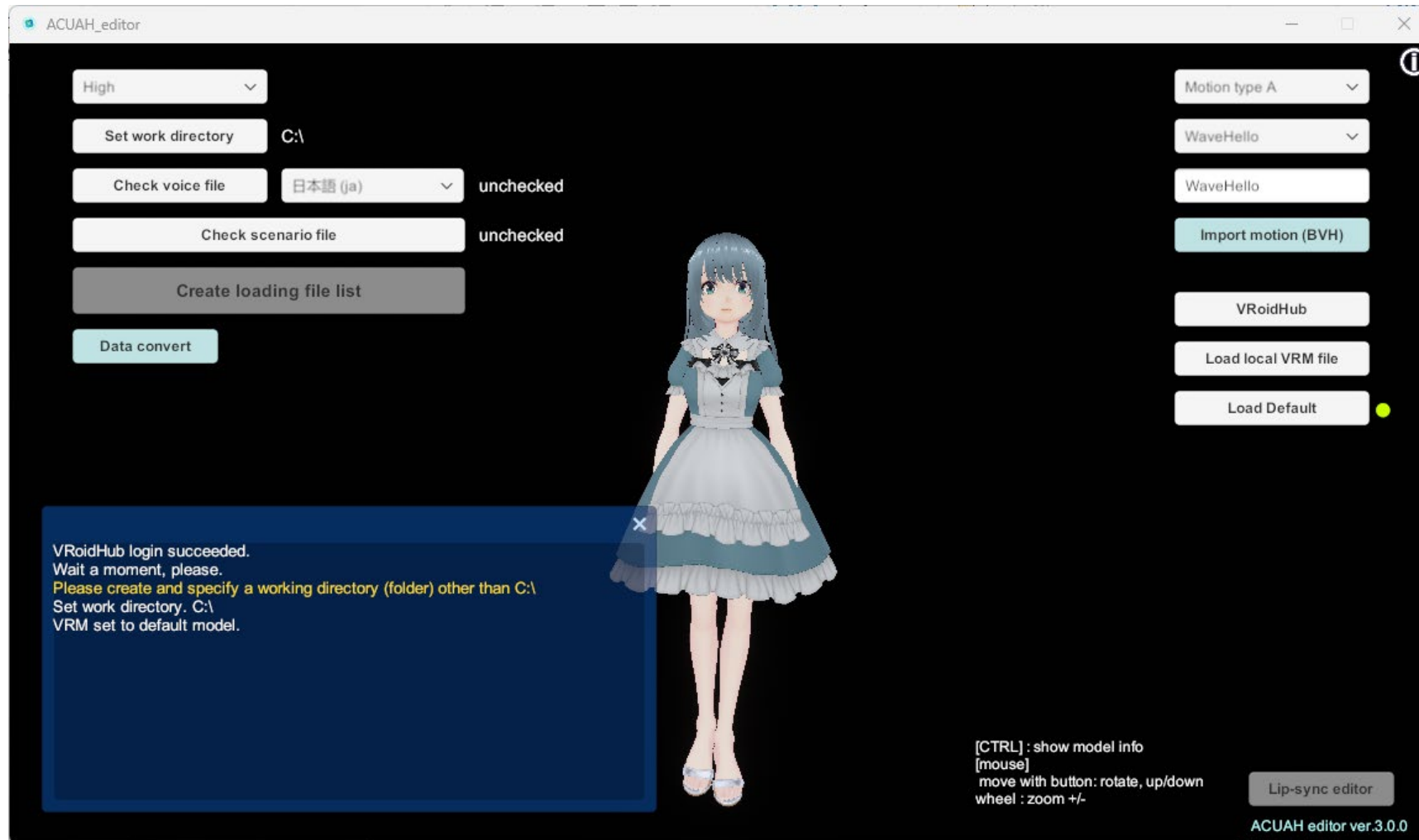
**Go to lip-sync data edit screen**

In order to press this button, sound file and of lip-sync data consistency check must be executed.



# ACUAH editor

## <Main menu screen> Basic Operation



**[CTRL] key**  
Show character  
information

**mouse left/right drag**  
Rotate the character

**mouse scroll wheel drag**  
Move the character  
up/down

**mouse wheel**  
zoom in/out

[CTRL] : show model info  
[mouse]  
move with button: rotate, up/down  
wheel : zoom +/-





# ACUAH editor

<Lip-sync data edit screen>



# ACUAH editor

## <Lip-sync data edit screen> Basic Operation

Image  
quality setting  
(Low) Very Low  
(High) Ultra

voice data selection  
drop-down list

Voice data  
playback button

lip-sync data creation  
operation field

lip-sync data  
save button

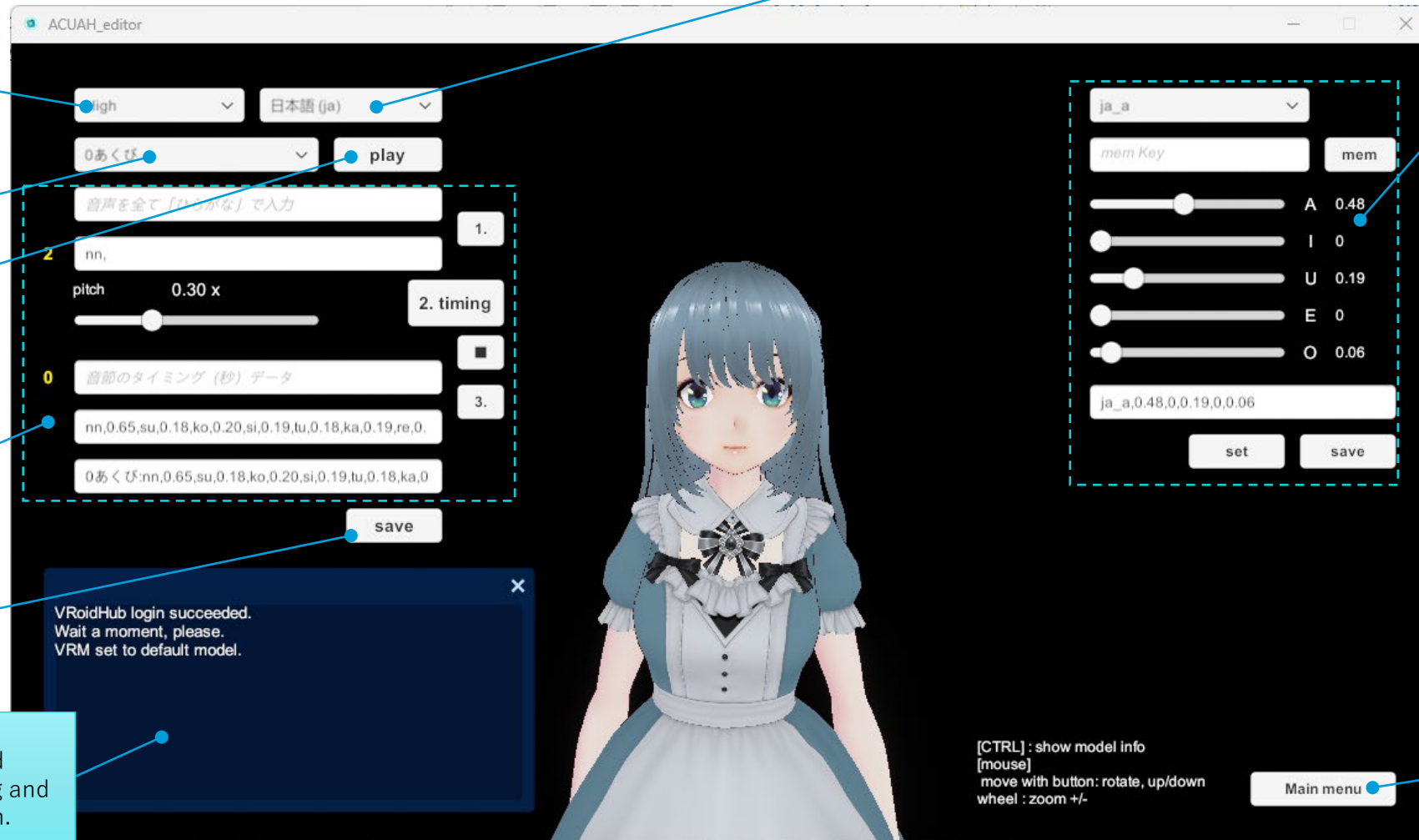
Message window

Check execution results and  
the status of the file loading and  
other information are shown.

Lip-sync data language settings

For lip-sync, editing  
mouth shape data  
field

Sorry, there is  
no explanation  
for this function  
in this manual.  
But you can change values  
on your responsibilities.



# ACUAH editor

## <Lip-sync data edit screen> Basic Operation



**mouse left/right drag**  
Rotate the character

**mouse scroll wheel drag**  
Move the character  
up/down

**mouse wheel**  
zoom in/out



# About scenario data converter

Dialogue scenario file for ACUAH ver.2.0.0-

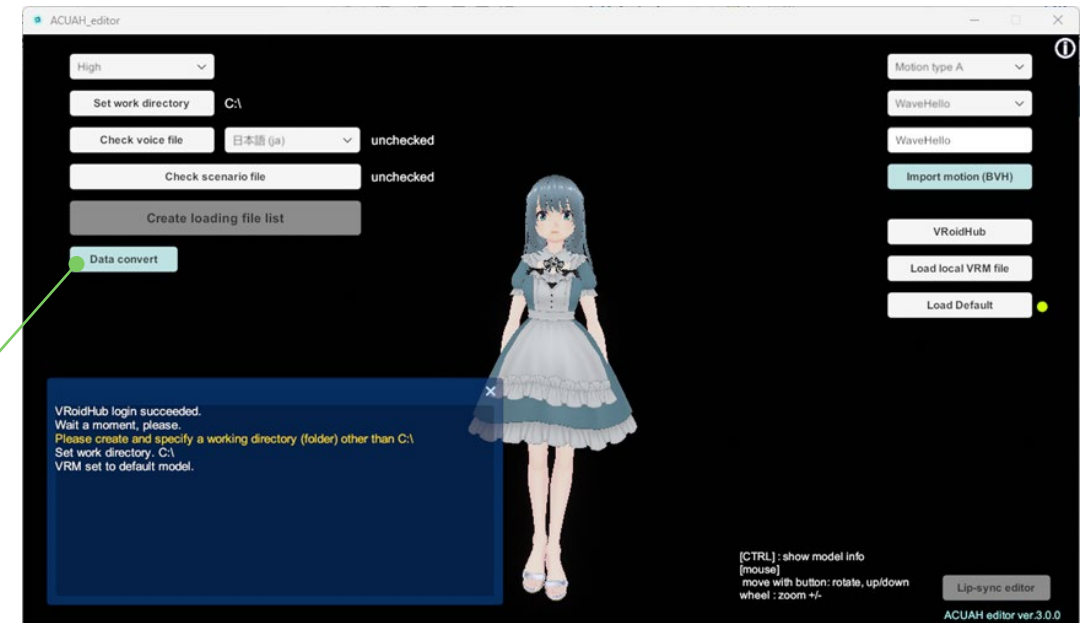


# About scenario data converter

- The names of the preset expressions have been changed as follows in accordance with the VRM 1.0 specification. (<https://vrm.dev/en/vrm1/changed>)
- If you store existing dialogue scenario data files (action\*.csv, scenario-\*.csv) under the working directory and click the “Data convert” button, the strings of the preset expressions will be changed to VRM 1.0 compatible ones.

VRM0.x	VRM1.0
Joy	Happy
Angry	Angry
Sorrow	Sad
Fun	Relaxed
Surprised	Surprised

**ACUAH -ver.1.x.x to ver.2.0.0**  
Dialogue scenario converter.



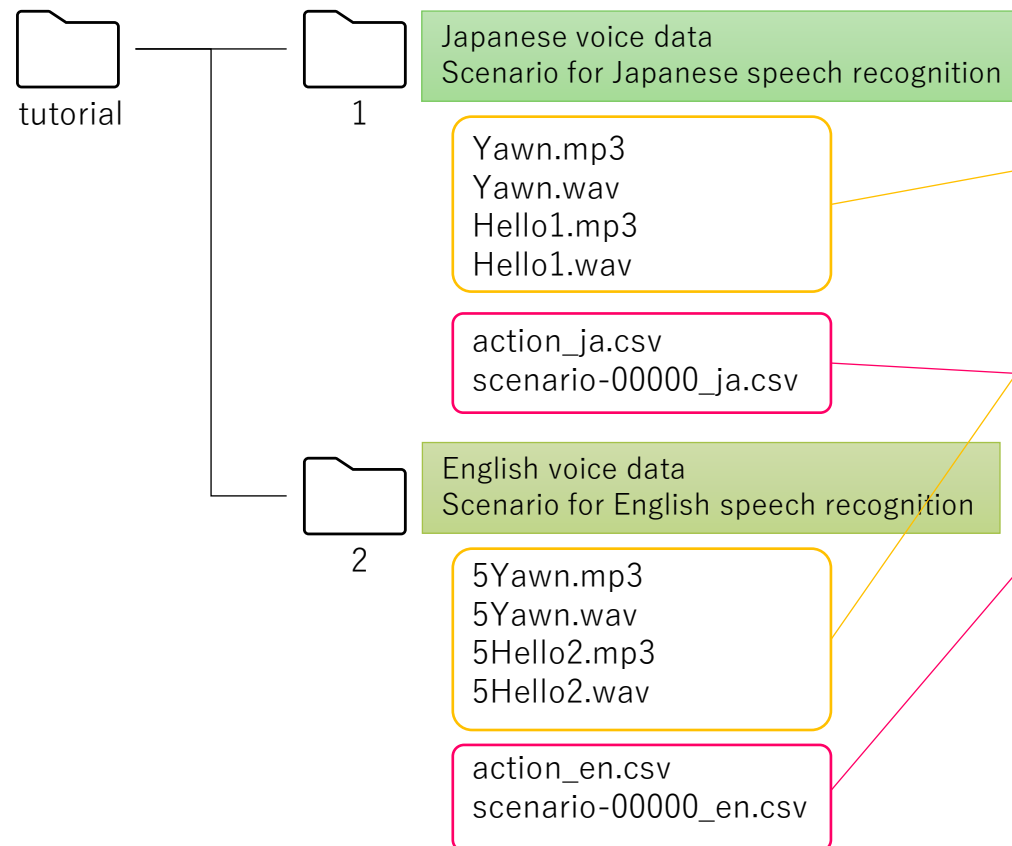
# Tutorial

About tutorial files



# Tutorial: About tutorial files

- To help you understand how to use ACUAH editor, we have included a tutorial.
- This tutorial will show you how to create a dialogue scenario file, how to create lip-sync data, and how to create a loading file list for ACUAH.



## voice data file

With the same file name

**MP3 format and WAV format are** required.

The file name will be automatically converted to "0\*.mp3" or "0\*.wav".

## Action and dialogue scenario file

Before you start working with ACUAH editor, you must create this file by using a CSV editor.

The file name is

action\_ja.csv, scenario-\*\_ja.csv (for Japanese speech recognition)

action\_en.csv, scenario-\*\_en.csv (for English speech recognition)  
(\* is an arbitrary string).

**action\_ja.csv, scenario-00000\_ja.csv**

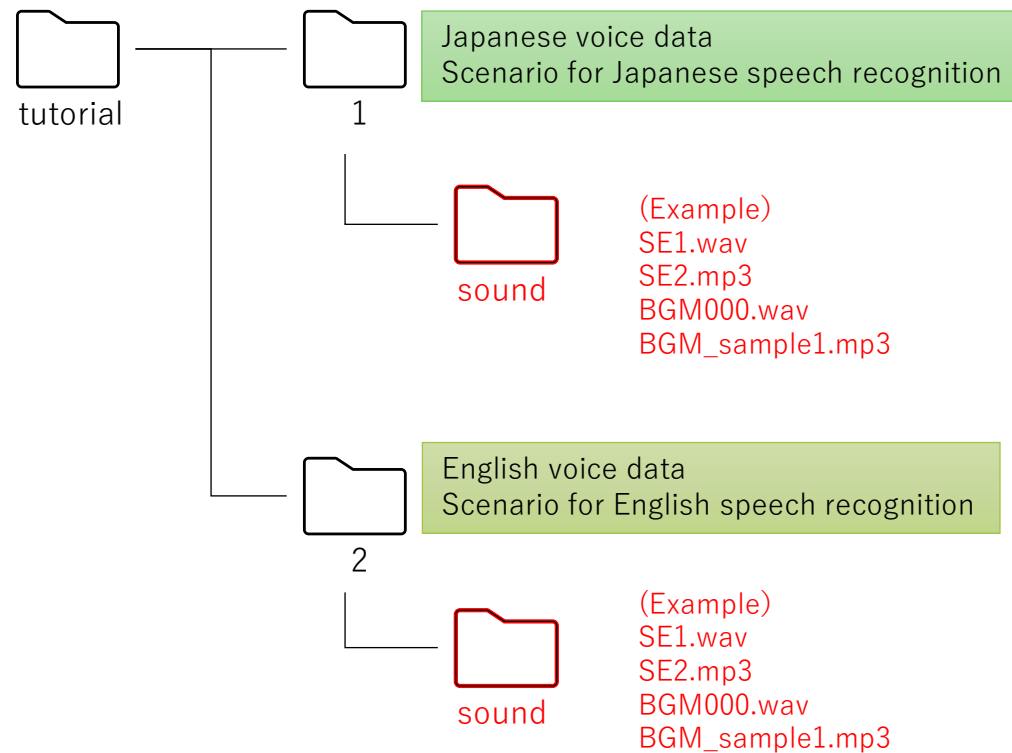
**action\_en.csv, scenario-00000\_en.csv**

**is required because it is the first file to be loaded.**

# Tutorial: About sound effect and BGM files

- To specify and play sound effect and background music (BGM) files in a dialog scenario, create a "sound" folder in the working directory below and place the files there.

(The actual files will be needed when checking the integrity of the dialog scenario).



[Warning]

Do not include spaces in the filename.



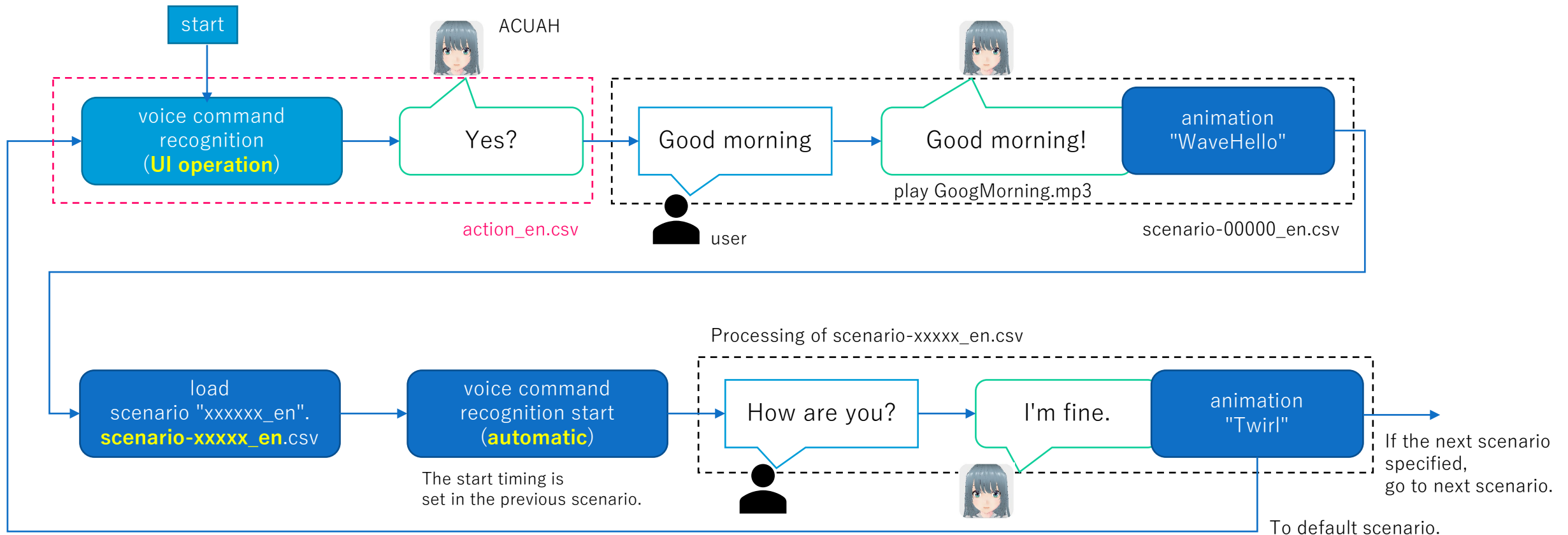
# Tutorial

## 1. Create a dialogue scenario file



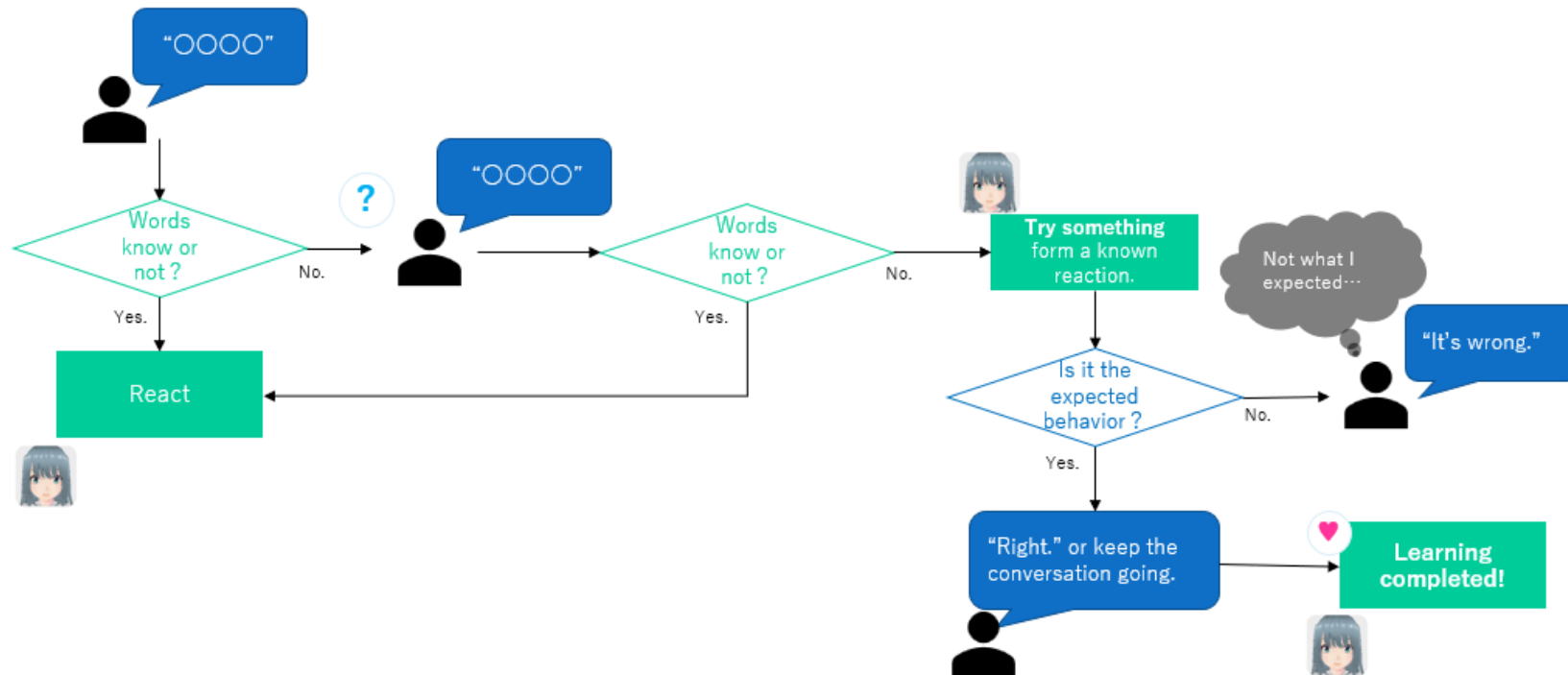
# Tutorial: 1. Create a dialogue scenario file

- ACUAH loads and switches dialogue scenario files each speech recognition with the user.



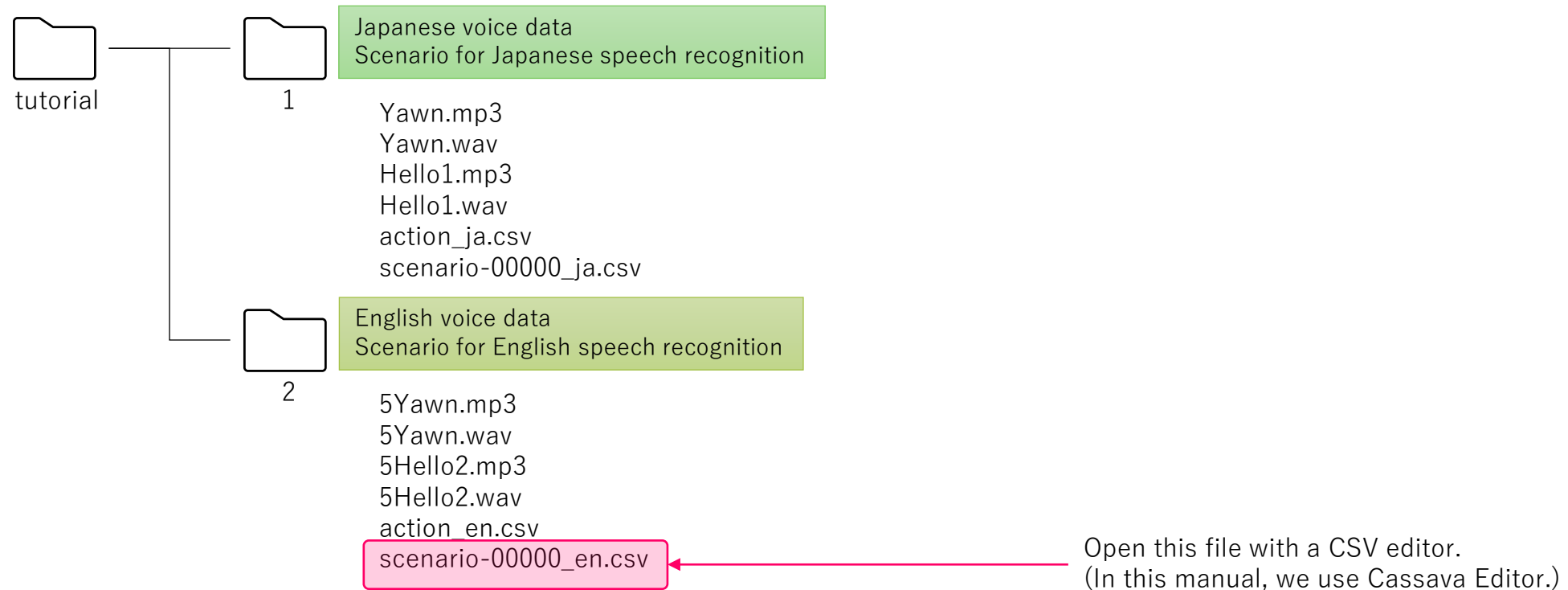
# Tutorial: 1. Create a dialogue scenario file

- ACUAH also has the ability to learn using dialogue scenario files.  
(Conversation type: "Dialogue scenario only")
- When a user speaks to the character that the character does not know (not in the dialogue scenario file), the character randomly choose a response from the dialogue scenario and returns that response.
- If the response is what the user was expecting, the user will continue the conversation, ACUAH learns that the response was correct.

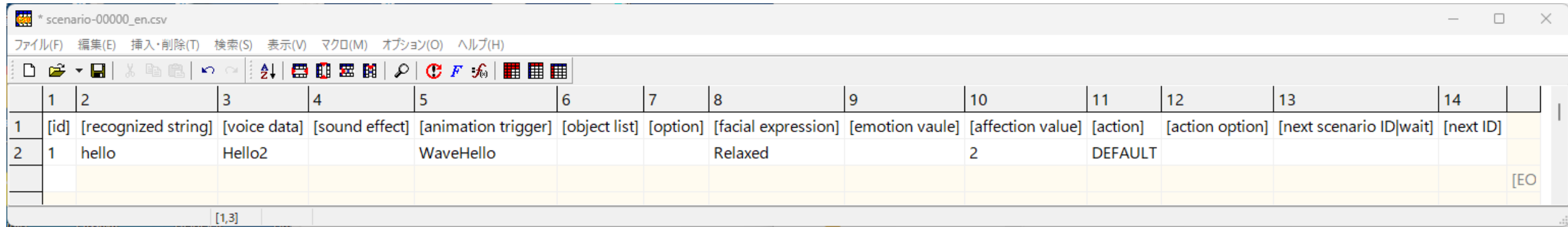


# Tutorial: 1. Create a dialogue scenario file

- How to create a dialogue scenario file (CSV).  
Let's take a look at scenario-00000\_en.csv in the tutorial folder.



# Tutorial: 1. Create a dialogue scenario file



The screenshot shows a CSV editor window titled "scenario-00000\_en.csv". The window has a menu bar with options: ファイル(F), 編集(E), 挿入・削除(T), 検索(S), 表示(V), マクロ(M), オプション(O), ヘルプ(H). Below the menu is a toolbar with various icons for file operations and editing. The main area displays a table with 15 columns and 4 rows. The first row is a header with labels in brackets. The second row contains sample data. The third and fourth rows are empty, with the label "[EO]" in the last cell of the fourth row. The status bar at the bottom shows "[1,3]".

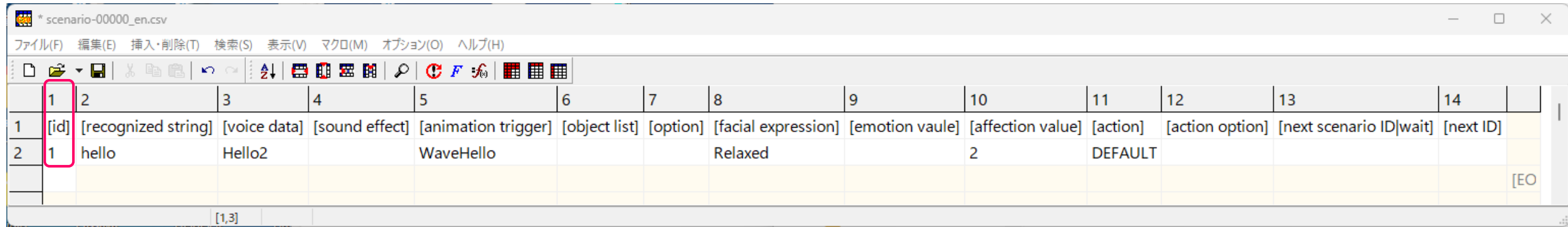
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	
2	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT				
															[EO]

## (1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

The first line is an index what the value in each column (item) represents.  
The values of these items are explained on the next page.

Also, since this first line is inserted for explanatory purposes,  
don't forget to delete them when you finally creating dialogue scenario data.

# Tutorial: 1. Create a dialogue scenario file



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	
1	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT				
2															[EO]

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Column 1: [id]

Enter an arbitrary string of characters (alphanumeric) to identify the response. It does not have to be unique.

This [id] string can be specified in column 14, [next ID].

If multiple lines contain the same [id] string, processing will continue in a dialogue scenario focused only on those lines.

The following string (in uppercase) is also used to control the scenario. **Note that this is case-sensitive when compared to [next ID].**

(\* is an arbitrary string)

P\* :For preset operation. If there is a "P" at the beginning of the line, the line is excluded from learning.

ANY :If there is no corresponding speech recognition string in the scenario, the line is executed without retrying speech recognition.

OTHER :If there is no corresponding speech recognition string in the scenario, the line to be executed after two speech recognition retries.

END :Forced termination of scenario response. It returns to the default scenario.

F\* :The line is excluded from learning.

E\* :Prohibited to use, used inside ACUAH application.

L\* :Prohibited to use. used inside ACUAH application.

DEBUG :Prohibited to use.

TEST :Prohibited to use.

# Tutorial: 1. Create a dialogue scenario file

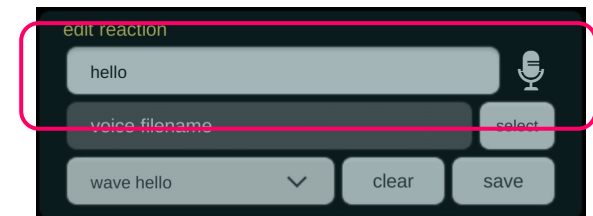
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion value]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	
1	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT				
2															[EO]

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Column 2: [recognized string]

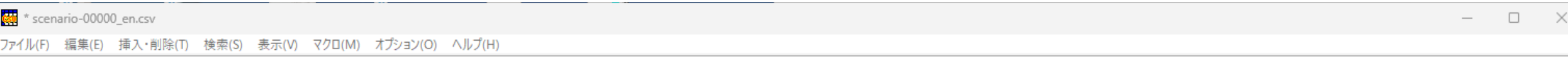
A string converted to text characters by speech recognition when the user speaks into the ACUAH, or by command input. Enter a string of characters that you expect the user to enter.

- (1) Start "ACUAH" on your Android smartphone.
- (2) Use the voice recognition string input field of "edit reaction" on the configuration menu 3 to input the voice recognition string. Please check what kind of string is converted and enter the string correctly.
- (3) In addition, the ACUAH provides a command input using a Bluetooth keyboard. Since the operation is possible, the character string when the character is typed by the keyboard is also possible.



ACUAH configuration menu 3  
edit reaction

## Tutorial: 1. Create a dialogue scenario file



The screenshot shows a CSV editor window titled '\*scenario-00000\_en.csv'. The menu bar includes 'ファイル(F)', '編集(E)', '挿入・削除(I)', '検索(S)', '表示(V)', 'マクロ(M)', 'オプション(O)', and 'ヘルプ(H)'. The toolbar contains various icons for file operations and data manipulation. The table has 15 columns with headers: '[id]', '[recognized string]', '[voice data]', '[sound effect]', '[animation trigger]', '[object list]', '[option]', '[facial expression]', '[emotion value]', '[affection value]', '[action]', '[action option]', '[next scenario ID|wait]', '[next ID]', and an empty column. The data rows are as follows:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion value]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	
2	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT				
															[EO]

The cell containing 'Hello2' in the third row, third column is highlighted with a red rectangle.

(1) Open `¥tutorial¥2¥scenario-00000_en.csv` in a CSV editor.

**Column 3: [voice data]**

Lists the file name of the voice data to be played when the character responds.

The actual file name will be "0~.mp3", but the first "0" (zero) and the extension ".mp3" must be excluded.

You can also include a wait time for audio playback by connecting the file name with a "|" character. The unit is (seconds).

If multiple files are listed by joining them with "-" (half-width hyphen), the voice files will be played back in sequence. (Don't use "-" in file names.)

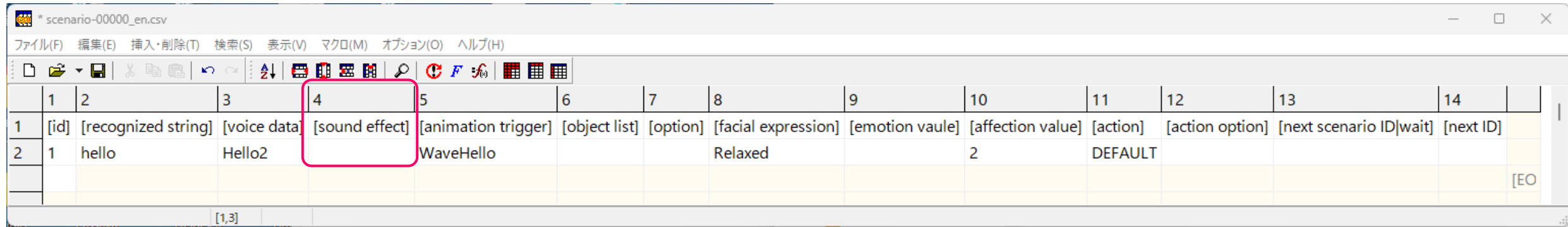
In addition, specifying multiple audio data files using the enclosing character "%&#\$" and a one-byte colon ":" will cause the following behavior.

%value 1:value 2:⋯:value n%	: Select value randomly.
#value 1:value 2:⋯:value n#	: The value is chosen according to the emotion value by dividing the minimum emotion value (-24) to the maximum emotion value (+24) into n. (rounded down to the nearest whole number)
&value 1:value 2:⋯:value n&	: The value is chosen according to affection value by dividing the maximum affection (1000) divide into n. (rounded down to the nearest whole number)
\$value 1:value 2:⋯:value n\$	: The value is selected according to the current time by dividing 0:00 AM. to 11:59 PM. per day into n equal parts of 1440 minutes.

(e.g.) &value1:value2:value3& : affection 1~333 → value1、affection 334~666 → value2、affection 667~1000 → value3



# Tutorial: 1. Create a dialogue scenario file



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	
2	1	hello	Hello2	WaveHello				Relaxed		2	DEFAULT				
															[EO]

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Column 4: [sound effect]

Sound effects can be played over audio data. In addition to the following sound effects, WAV/MP3/OGG format files can be loaded and played.

"alarm", "bird", "button", "cancel", "delete", "learned", "start"

If a file name (including .wav/.mp3/.ogg extension) is listed, the specified file can be played.

It is also possible to specify the waiting time before playback starts by using "|". The unit is (seconds).

In addition, specifying multiple audio data files using the enclosing character "%#&\$" and a one-byte colon ":" will cause the following behavior.

%value 1:value 2:…:value n% : Select value randomly.

#value 1:value 2:…:value n# : The value is chosen according to the emotion value by dividing the minimum emotion value (-24) to the maximum emotion value (+24) into n. (rounded down to the nearest whole number)

&value 1:value 2:…:value n& : The value is chosen according to affection value by dividing the maximum affection (1000) divide into n. (rounded down to the nearest whole number)

\$value 1:value 2:…:value n\$ : The value is selected according to the current time by dividing 0:00 AM. to 11:59 PM. per day into n equal parts of 1440 minutes.

(e.g.) &value 1:value 2:value 3& : affection 1~333 → value 1、affection 334~666 → value2、affection 667~1000 → value 3

## Tutorial: 1. Create a dialogue scenario file

[illegible]

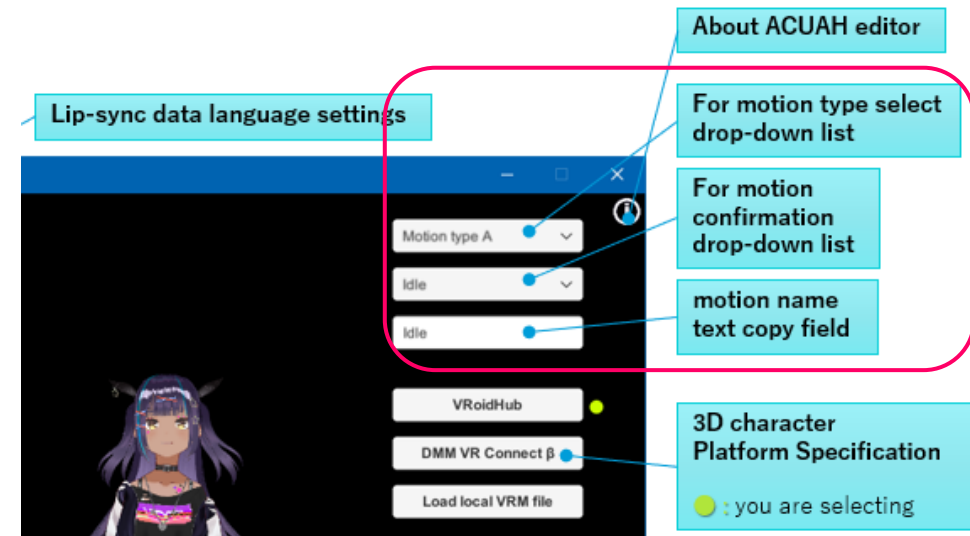
(1) Open `¥tutorial¥2¥scenario-00000_en.csv` in a CSV editor.

**Column 5: [animation trigger]**

Specifies the animation when the character responds.  
 “For motion confirmation drop-down list” is located  
 in the upper right corner in The ACUAH editor main menu.  
 Please select the appropriate animation.  
 You can copy and paste the string,  
 select “Motion name text copy field” and click it.

If row5 is a blank, no motion will be played.

(continue to the next page)



## Tutorial: 1. Create a dialogue scenario file

[illegible]

(1) Open `¥tutorial¥2¥scenario-00000_en.csv` in a CSV editor.

**Column 5: [animation trigger]**

The animation trigger value can be followed by a "|" to indicate the amount of time to wait for the animation to play. The unit is (seconds). Specifying multiple audio data files using the enclosing character "%&\$" and a one-byte colon ":" will cause the following behavior.

%value 1:value 2:⋯:value n%	: Select value randomly.
#value 1:value 2:⋯:value n#	: The value is chosen according to the emotion value by dividing the minimum emotion value (-24) to the maximum emotion value (+24) into n. (rounded down to the nearest whole number)
&value 1:value 2:⋯:value n&	: The value is chosen according to affection value by dividing the maximum affection (1000) divide into n. (rounded down to the nearest whole number)
\$value 1:value 2:⋯:value n\$	: The value is selected according to the current time by dividing 0:00 AM. to 11:59 PM. per day into n equal parts of 1440 minutes.
(e.g.) &value 1:value 2:value 3&	: affection 1~333 → value 1、affection 334~666 → value2、affection 667~1000 → value 3

Also, the "< >" in the scenario file is associated with the action, and the value in the scenario file is ignored.

## Tutorial: 1. Create a dialogue scenario file

[illegible]

(1) Open `¥tutorial¥2¥scenario-00000_en.csv` in a CSV editor.

**Columns 6 and 7: [object list], [option]**

Unused (for future use). Please leave this field blank.

## Tutorial: 1. Create a dialogue scenario file

[illegible]

(1) Open `¥tutorial¥2¥scenario-00000_en.csv` in a CSV editor.

**Columns 8: [facial expression]**

You can specify the character's facial expression with the following string.

The unit of facial expression time is "second". It can be specified using a decimal point.

The size of a character's emotional expression changes depending on the character's level of affection,

So please note that if the character has a low level of affection,

it won't show in their facial expressions as much.

In addition, specifying multiple audio data files using the enclosing character "%&\$\$" and a one-byte colon ":" will cause the following behavior.

```
%value 1:value 2:⋯:value n%           : Select value randomly.
```

#value 1:value 2:⋯:value n# : The value is chosen according to the emotion value by dividing the minimum emotion value (-24) to the maximum emotion value (+24) into n. (rounded down to the nearest whole number)

&value 1:value 2:⋯:value n& : The value is chosen according to affection value by dividing the maximum affection (1000) divide into n.  
(rounded down to the nearest whole number)

(e.g.) &value 1:value 2:value 3& : affection 1~333 → value 1、affection 334~666 → value2、affection 667~1000 → value 3

\$value 1:value 2:…:value n\$ : The value is selected according to the current time by dividing 0:00 AM. to 11:59 PM. per day into n equal parts of 1440 minutes.

Happy  
Angry  
Sad  
Relaxed  
Surprised

# Tutorial: 1. Create a dialogue scenario file

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	
2	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT				
															[EO]

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Columns 9: [emotion value]

Emotion values are like moods that go up and down in real time as you use the application.

The value is an integer ranging from a minimum of -24 to a maximum of +24. 0 is the normal value.

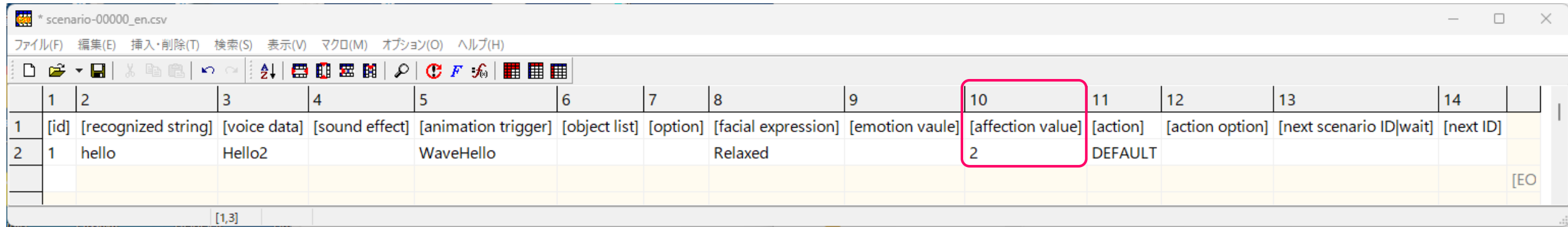
When the application is not running, the value is 1 per hour, approaching the normal value.

When the app is running, the value approaches the normal value of 1 in 12 minutes when the app is in sleep mode.

In addition, specifying multiple audio data files using the enclosing character "%#&\$" and a one-byte colon ":" will cause the following behavior.

%value 1:value 2:…:value n%	: Select value randomly.
#value 1:value 2:…:value n#	: The value is chosen according to the emotion value by dividing the minimum emotion value (-24) to the maximum emotion value (+24) into n. (rounded down to the nearest whole number)
&value 1:value 2:…:value n&	: <u>The value is chosen according to affection value by dividing the maximum affection (1000) divide into n. (rounded down to the nearest whole number)</u>
(e.g.) &value 1:value 2:value 3&	: affection 1~333 → value 1、affection 334~666 → value2、affection 667~1000 → value 3
\$value 1:value 2:…:value n\$	: The value is selected according to the current time by dividing 0:00 AM. to 11:59 PM. per day into n equal parts of 1440 minutes.

# Tutorial: 1. Create a dialogue scenario file



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	
2	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT				[EO]

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Column 10: [affection value]

A parameter value that increases or decreases the character's level of affection.

It can be a positive or negative integer value.

If you set a value here, the affection parameter value will increase or decrease each time the character responds.

The character's affection parameter ranges from 0 to 1000.

Depending on the level of affection, the size emotion of the character's reaction or response voice will change.

Specifying multiple audio data files using the enclosing character "%&\$" and a one-byte colon ":" will cause the following behavior.

%value 1:value 2:…:value n% : Select value randomly.

#value 1:value 2:…:value n# : The value is chosen according to the emotion value by dividing the minimum emotion value (-24) to the maximum emotion value (+24) into n. (rounded down to the nearest whole number)

&value 1:value 2:…:value n& : The value is chosen according to affection value by dividing the maximum affection (1000) devide into n. (rounded down to the nearest whole number)

\$value 1:value 2:…:value n\$ : The value is selected according to the current time by dividing 0:00 AM. to 11:59 PM. per day into n equal parts of 1440 minutes.

(e.g.) &value 1:value 2:value 3& : affection 1~333 → value 1、affection 334~666 → value2、affection 667~1000 → value 3

# Tutorial: 1. Create a dialogue scenario file

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	[EO]
2	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT				

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Columns 11 and 12: [action], [action option]

Specifies the various functions that will be performed as the character responds.  
The [action option] depends on the [action].

In addition, [action] and [action option] can be used to pass values between scenario files and to perform conditional branching.

**Refer to the “List of [action] / [action option] / output value to be filled in the dialogue scenario” and  
“Tutorial: 2. About passing values between scenario files by output variable.”**

(continue to the next page)



# Tutorial: 1. Create a dialogue scenario file

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion value]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	[EO]
2	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT				

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Columns 11 and 12: [action], [action option]

Also, the enclosing character "%#&\$" and a one-byte colon ":" are available in [action] and [action option] and that will cause the following behavior.

%value 1:value 2:…:value n% : Select value randomly.

#value 1:value 2:…:value n# : The value is chosen according to the emotion value by dividing the minimum emotion value (-24) to the maximum emotion value (+24) into n. (rounded down to the nearest whole number)

&value 1:value 2:…:value n& : The value is chosen according to affection value by dividing the maximum affection (1000) divide into n. (rounded down to the nearest whole number)

(e.g.) &value 1:value 2:value 3& : affection 1~333 → value 1、affection 334~666 → value2、affection 667~1000 → value 3

\$value 1:value 2:…:value n\$ : The value is selected according to the current time by dividing 0:00 AM. to 11:59 PM. per day into n equal parts of 1440 minutes.

<Caution>

When setting enclosing characters for [action] and [action option], make sure that the number of values in these items are the same.

If random is specified in [action], the value selected in [action option] is linked to [action].

(e.g.) [action] %DATETIME.SHOW:TIMER.SET% [action option] %:60% → If "TIMER.SET" is selected for [action], "60" is selected for [action option].

# Tutorial: 1. Create a dialogue scenario file

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]
2	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT			
														[EO]

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Columns 13: [next scenario ID|wait]

In [next scenario ID|wait], specify the next dialogue scenario file to load.

It is not specified in the tutorial data, but you can specify it arbitrarily using the file name of the dialogue scenario.

You can also connect them with "|" to set wait time before automatically starting voice recognition and command input after loading the dialogue scenario file. The unit is (seconds).

If left blank, 1.0 second is set.

Here, it is recommended to set the playback time of the audio file specified in the third column + about 0.5 seconds.

Example: **scenario-aabbcc1234\_en.csv after 3.0 seconds of waiting time** → "aabbcc1234\_en|3.0".

("\_ja" and "\_en" are required characters in the language of the dialogue scenario, and it depend on the language setting of Android OS.)

If it is a blank, the default dialogue scenario ("scenario-00000\_\*.csv") will be loaded.

# Tutorial: 1. Create a dialogue scenario file

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]
2	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT			
														[EO]
			[1,3]											

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Columns 14: [next ID]

[next ID] allows you to specify which [id] line should be executed after the next dialogue scenario file is loaded.  
Please enter the string of characters listed under [id] in the appropriate line of the following dialogue scenario file.

- If [next ID] is listed, voice/command input recognition will not be performed at the beginning of processing the next dialogue scenario.
- Please note that when **comparing whether [next ID] and [id] in the next dialogue scenario file match, it is case-sensitive.**
- Also, if **there are multiple lines in the next dialogue scenario file that match [next ID] and [id], the content of the next dialogue scenario is narrowed down to the relevant multiple lines, and processing continues.**

See "2. About passing values between scenario files using output variable." for an example.

(Continued on next page)

# Tutorial: 1. Create a dialogue scenario file

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]
2	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT			
														[EO]
			[1,3]											

(1) Open .¥tutorial¥2¥scenario-00000\_en.csv in a CSV editor.

## Columns 14: [next ID]

Also, the enclosing character "%#&\$" and a one-byte colon ":" are available in [next scenario ID] and [next scenario wait].

%value 1:value 2:…:value n% : Select value randomly.

#value 1:value 2:…:value n# : The value is chosen according to the emotion value by dividing the minimum emotion value (-24) to the maximum emotion value (+24) into n. (rounded down to the nearest whole number)

&value 1:value 2:…:value n& : The value is chosen according to affection value by dividing the maximum affection (1000) devide into n. (rounded down to the nearest whole number)

(e.g.) &value 1:value 2:value 3& : affection 1~333 → value 1、affection 334~666 → value2、affection 667~1000 → value 3

\$value 1:value 2:…:value n\$ : The value is selected according to the current time by dividing 0:00 AM. to 11:59 PM. per day into n equal parts of 1440 minutes.

<Caution>

When setting enclosing characters for [next scenario ID|wait] and [next ID], make sure that the number of values in these items are the same.

If random is specified in [next scenario ID|wait], the value selected in [next ID] is linked to [next scenario ID|wait].

(e.g.) [next scenario ID] %test01\_en|1.0:check01\_en% [next ID] %P1:P99%

→ If "check01\_en" is selected for [next scenario ID], "P99" is selected for [next ID].

# Tutorial: 1. Create a dialogue scenario file

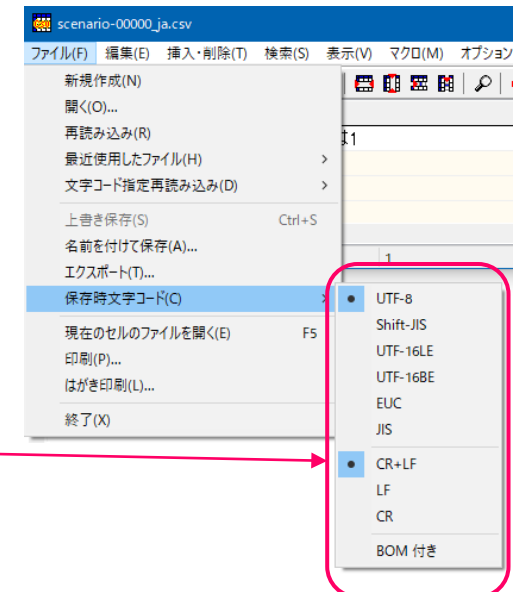
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
	[id]	[recognized string]	[voice data]	[sound effect]	[animation trigger]	[object list]	[option]	[facial expression]	[emotion vaule]	[affection value]	[action]	[action option]	[next scenario ID wait]	[next ID]	
1	1	hello	Hello2		WaveHello			Relaxed		2	DEFAULT				[EO]
2															

## (2) After confirming the data, modify the format and save it.

Once the dialogue scenario is created, format the file and save the CSV file.  
Please check and correct the following points and save the file overwrite.  
It is saved under . . ¥tutorial¥2¥.

- The first line, which contains the item name, should be deleted.
- Ensure that the EOF is at the end of the line following the last line where you entered the data.  
(If there are unnecessary blank lines above the EOF, delete them)
- Save the file with the following character set

UTF-8  
CR+LF  
No BOM (please confirm BOM is unchecked)



# Tutorial: 1. Create a dialogue scenario file

This completes the creation of the dialogue scenario file.

- In the tutorial, it is only edited the "scenario-00000\_en.csv" file, but we have explained what to enter in each column, so please try to create your original dialogue scenario.
- Again, "scenario-00000\_ja.csv" or "scenario-00000\_en.csv" is the default scenario file and is a required file as the default scenario file.
- All other dialogue scenario file names are "scenario-\*\_ja.csv" or "scenario-\*\_en.csv" with " \*" can be any character string you like.

Output values can be used to pass values between dialog scenario files.

In next, "Tutorial: 2. About passing values between scenario files using output variable."



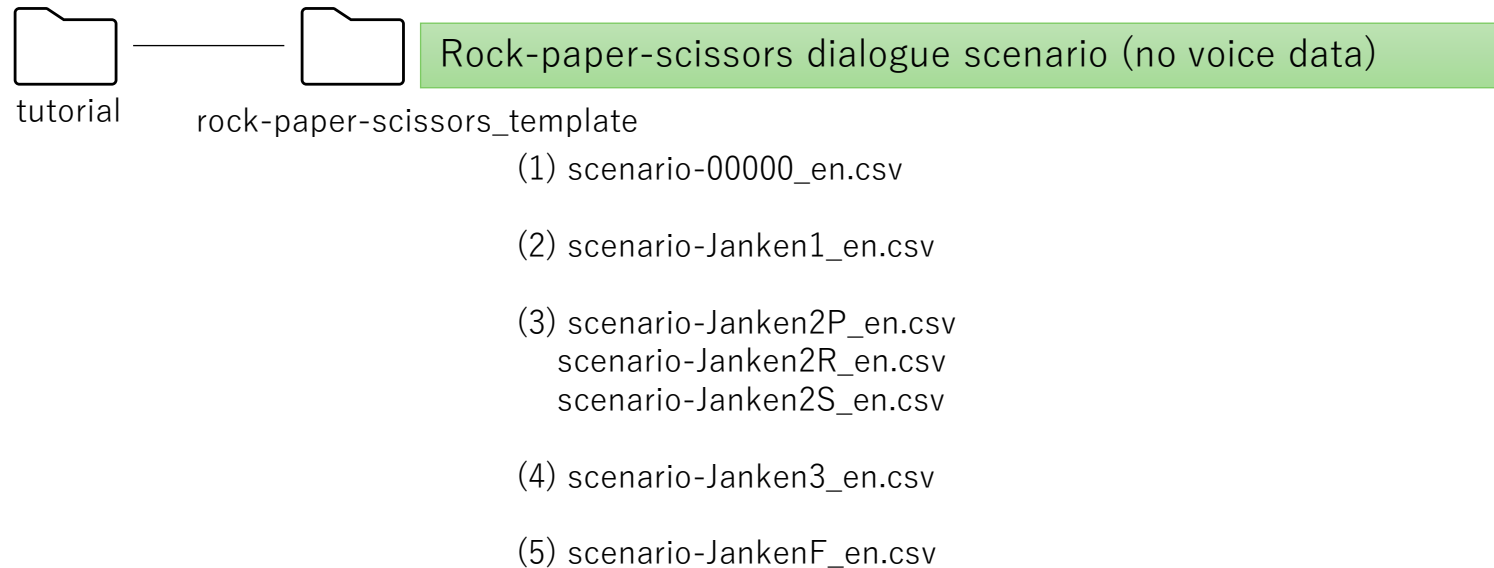
# Tutorial

2. About passing values between scenario files using output variable.



## Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

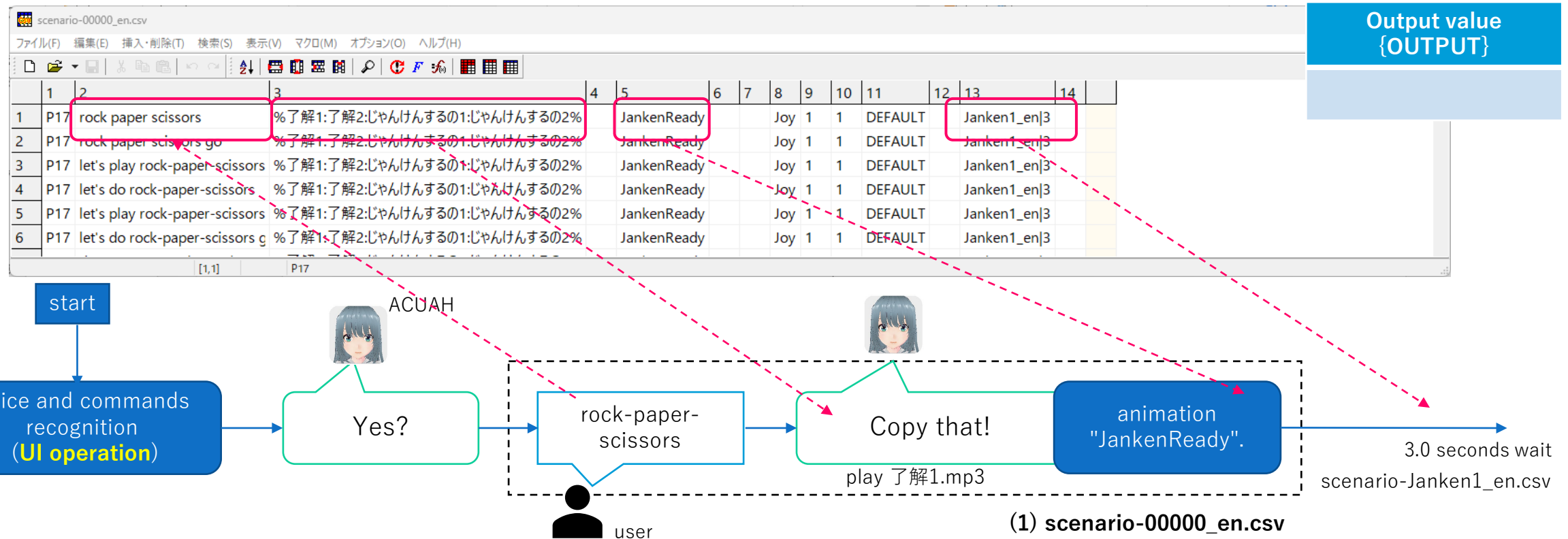




# Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

## (1) scenario-00000\_en.csv

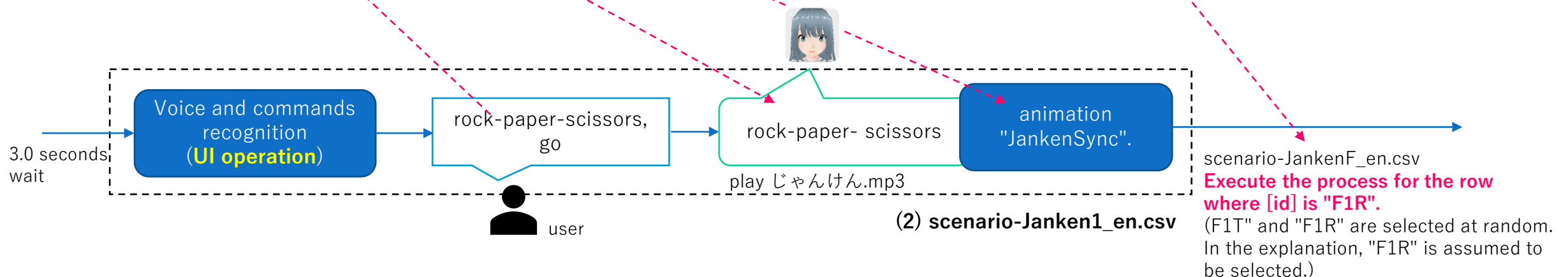


# Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

## (2) scenario-Janken1\_en.csv

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	P1	rock paper scissors go	じゃんけん		JankenSync						GET_ACTION_OPTION	Janken	%JankenF_en:JankenF_en%	%F1T:F1R%
2	P1	rock paper scissors	じゃんけん		JankenSync						GET_ACTION_OPTION	Janken	%JankenF_en:JankenF_en%	%F1T:F1R%
3	P1	rock paper scissor	じゃんけん		JankenSync						GET_ACTION_OPTION	Janken	%JankenF_en:JankenF_en%	%F1T:F1R%
4	FD	{OUTPUT}			JankenSync						GET_ACTION_OPTION	Janken	%JankenF_en:JankenF_en%	%F1T:F1R%
5	END	cancel	%了解1:了解2%		WaveHello						DEFAULT			



# Tutorial: 2. About passing values between scenario files using output variable.

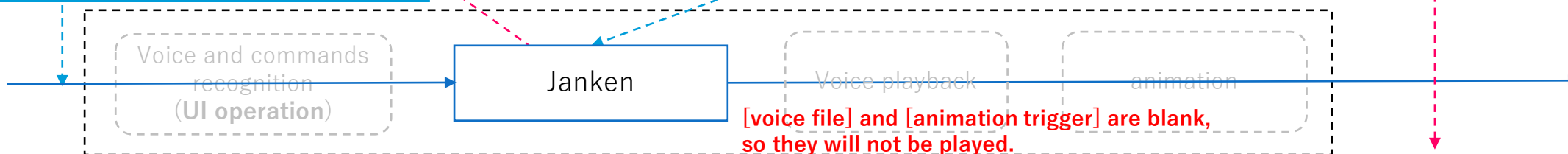
- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

## (5) scenario-JankenF\_en.csv

scenario-JankenF_en.csv													Output value {OUTPUT}	
	1	2	3	4	5	6	7	8	9	10	11	12		
1	F1T	{OUTPUT}										JankenRock JankenScissors JankenPaper	JankenF_en	F2
2	F1R	{OUTPUT}										%GET_ACTION_OPTION:GET_ACTION_OPTION:GET_ACTION_OPTION% %JankenRock:JankenScissors:JankenPaper%	JankenF_en	F2
3	F2	JankenRock											Janken2R_en 2	
4	F2	JankenScissor											Janken2S_en 2	
5	F2	JankenPaper											Janken2P_en 2	

[action] GET\_ACTION\_OPTION  
→ Get the value of [action option] and store it in the output value.

The "{OUTPUT}" in [recognized string] is replaced by the value of the output value.



No voice or command recognition performed, compares the output value string with the string in [recognized string].  
"{OUTPUT}" is replaced by the value of the output value, so it matches all output values.

(5) scenario-JankenF\_en.csv

In the same scenario-JankenF\_en.csv, go to the line where [id] is "F2".

# Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

## (5) scenario-JankenF\_en.csv

scenario-JankenF_en.csv													
ファイル(F) 編集(E) 挿入・削除(T) 検索(S) 表示(V) マクロ(M) オプション(O) ヘルプ(H)													
[Icons]													
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	F1T												
2	F1R												
3	F2	JankenRock									DEFAULT		Janken2R_en 2
4	F2	JankenScissor									DEFAULT		Janken2S_en 2
5	F2	JankenPaper									DEFAULT		Janken2P_en 2
6	F3C												
7	F3L												
8	F3V												
9	F4												
10	F5												
11	F5												
12	F5												
13	F6												

**Output value {OUTPUT}**  
JankenPaper

Because there are multiple lines in scenario-JankenF.csv where [id] is "F2", before starting to process scenario-JankenF\_en.csv the processing range※ is narrowed down to the range of rows where [id] is "F2".

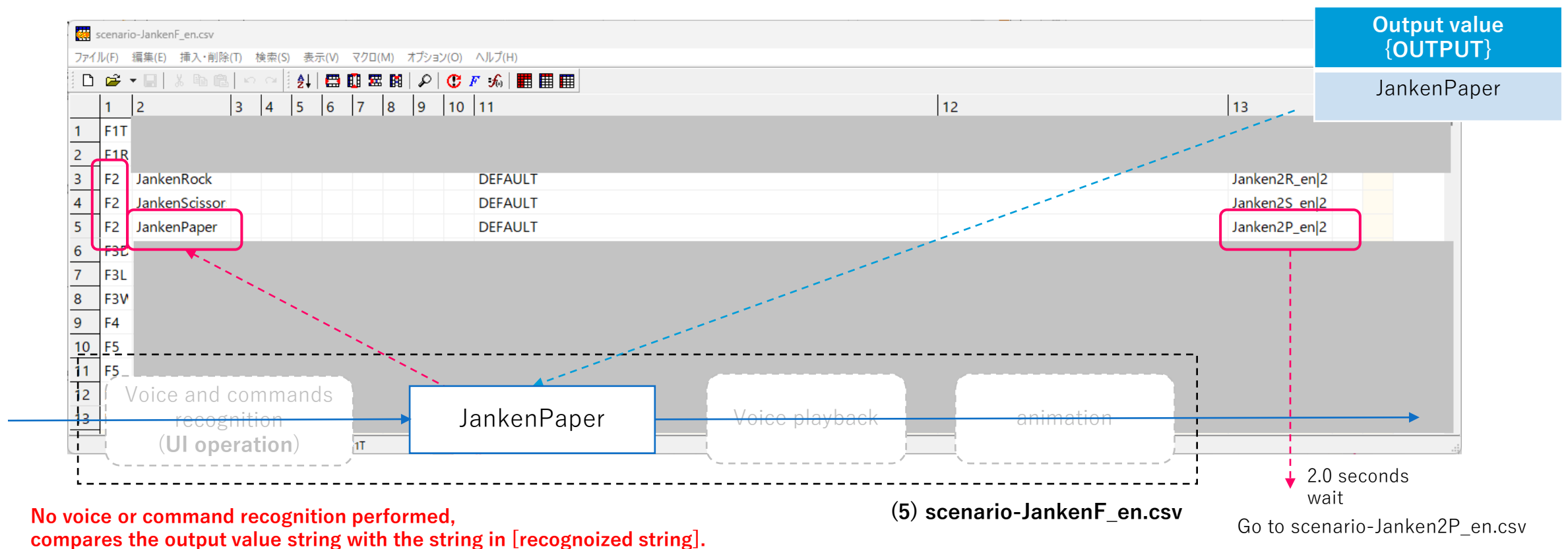
\*[note] Lines with the same [id] do not have to be listed together.

In the same scenario-JankenF\_en.csv, go to the line where [id] is "F2".

# Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

## (5) scenario-JankenF\_en.csv



# Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

## (3) scenario-Janken2P\_en.csv

[action] SAVE\_VAR\_COUNT  
→ Stored in count file and increase or decrease the value of the variable name (string) in [action option].  
→ "JankenPaper" value +3

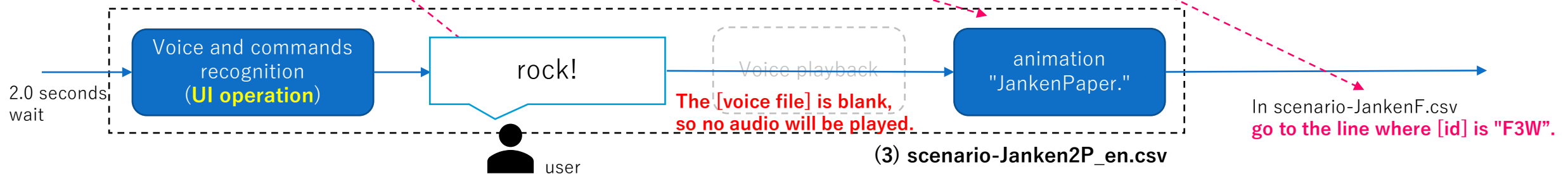
count file

JankenPaper, +3

Output value  
{OUTPUT}

JankenPaper

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	P1	rock			JankenPaper					SAVE_VAR_COUNT	JankenPaper 3	JankenF_en	F3W	
2	P2	scissors			JankenPaper					SAVE_VAR_COUNT	JankenPaper -2	JankenF_en	F3L	
3	P2	scissor			JankenPaper					SAVE_VAR_COUNT	JankenPaper -2	JankenF_en	F3L	
4	P2	Caesars			JankenPaper					SAVE_VAR_COUNT	JankenPaper -2	JankenF_en	F3L	
5	P2	Caesar			JankenPaper					SAVE_VAR_COUNT	JankenPaper -2	JankenF_en	F3L	
6	P3	paper			JankenPaper					SAVE_VAR_COUNT	JankenPaper	JankenF_en	F3D	



# Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

## (5) scenario-JankenF\_en.csv

[action] GET\_ACTION\_OPTION  
→ Get the value of [action option] and store it in the output value.  
→ Get "WIN" and store it as an output value.

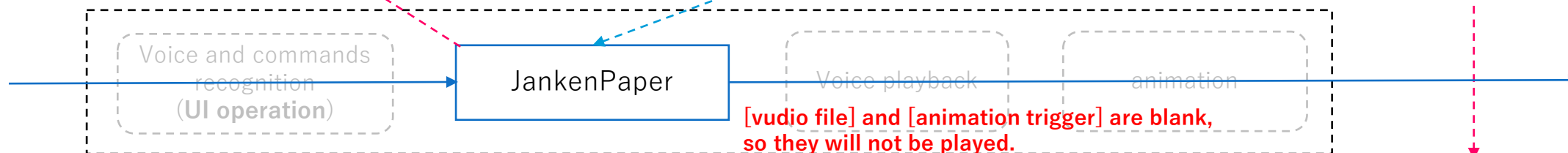
count file

JankenPaper, +3

Output value  
{OUTPUT}

WIN

	1	2	3	4	5	6	7	8	9	10	11	12	13
5	F2	JankenPaper									DEFAULT		Janken2P_en 2
6	F3D	{OUTPUT}									GET_ACTION_OPTION	DRAW	JankenF_en F4
7	F3L	{OUTPUT}									GET_ACTION_OPTION	LOSE	JankenF_en F4
8	F3W	{OUTPUT}									GET_ACTION_OPTION	WIN	JankenF_en F4
9	F4	{OUTPUT}									WAIT	2000	JankenF_en F5
10	F5	WIN	%勝	lear	Tw		Joy	2	10		GET_ACTION_OPTION	I win!	JankenF_en F6
11	F5	LOSE	%負	Qui			See	2	%		GET_ACTION_OPTION	I lost	JankenF_en F6



No voice or command recognition performed, compares the output value string with the string in [recognized string].  
"{OUTPUT}" is replaced by the value of the output value, so it matches all output values.

(5) scenario-JankenF\_en.csv

In the same scenario-JankenF\_en.csv, go to the line where [id] is "F4".



# Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

## (5) scenario-JankenF\_en.csv

count file

JankenPaper, +3

Output value  
{OUTPUT}

WIN

scenario-JankenF\_en.csv

ファイル(F) 編集(E) 挿入・削除(T) 検索(S) 表示(V) マクロ(M) オプション(O) ヘルプ(H)

📁

📄

✂

📄

📄

🔄

📄

📄

🔍

🔄

📄

📄

	1	2	3	4	5	6	7	8	9	10	11
5	F2	JankenPaper									DEFA
6	F3D	{OUTPUT}									GET_A
7	F3L	{OUTPUT}									GET_A
8	F3W	{OUTPUT}									GET_ACTION_OPTION
9	F4	{OUTPUT}									WAIT
10	F5	WIN	%勝	lear	Tw	Joy	2	10			GET_ACTION_OPTION
11	F5	LOSE	%負	Qui	Ser	2	%				GET_ACTION_OPTION

[action] WAIT  
→ [action option] (milliseconds) wait.  
→ Before transitioning to the next scenario  
set the wait time.

WIN  
2000

	13		
	Janken2P_en 2		
	JankenF_en	F4	
	JankenF_en	F4	
	JankenF_en	F4	
	JankenF_en	F4	
	JankenF_en	F5	
	JankenF_en	F6	
	JankenF_en	F6	

[action] WAIT  
→ [action option] (milliseconds) wait.  
→ Before transitioning to the next scenario set the wait time.

WIN

[vudio file] and [animation trigger] are blank, so they will not be played.

Wait 2000 milliseconds (= 2 seconds).

No voice or command recognition performed, compares the output value string with the string in [recognized string].  
"{OUTPUT}" is replaced by the value of the output value, so it matches all output values.

(5) scenario-JankenF\_en.csv

In same scenario-JankenF\_en.csv go to the line where [id] is "F5".





# Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

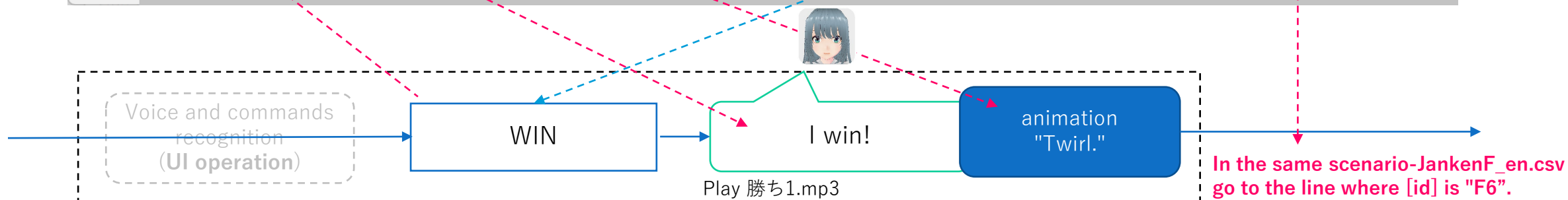
## (5) scenario-JankenF\_en.csv

[action] GET\_ACTION\_OPTION  
→ Get the value of [action option] and store it in the output value.  
→ Get "I win!" and store it as an output value.

Output value  
{OUTPUT}

I win!

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	F3W													
9	F4													
10	F5	WIN	%勝ち1:勝ち2%	earned	Twirl		Joy	2	10	GET_ACTION_OPTION I win!		JankenF_en	F6	
11	F5	LOSE	%負け1:負け2%		QuickFormalBow		Sorrow	-2	%-2:-1%	GET_ACTION_OPTION I lost.		JankenF_en	F6	
12	F5	DRAW	あいこで							GET_ACTION_OPTION It's a draw.		Janken1_en	FD	
13	F6													



No voice or command recognition performed,  
compares the output value string with the string in [recognized string].

(5) scenario-JankenF\_en.csv

## Tutorial: 2. About passing values between scenario files using output variable.

- Using a Rock-paper-scissors dialogue scenario file, this section describes how to pass values between scenario files.

(5) scenario-JankenF\_en.csv

scenario-JankenF\_en.csv

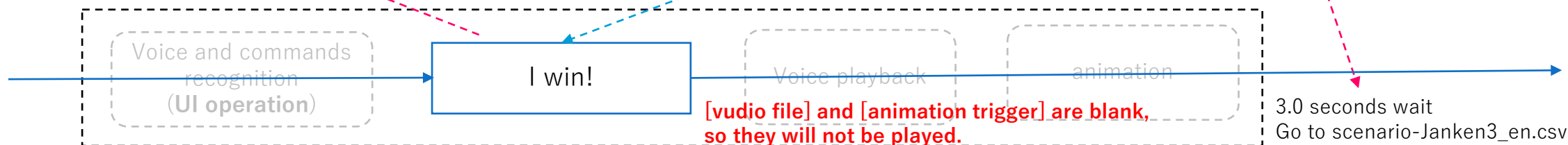
ファイル(F) 編集(E) 挿入・削除(T) 検索(S) 表示(V) マクロ(M) オプション(O) ヘルプ(H)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	F3W	{OUTPUT}									GET_ACTION_OPTION	WIN	JankenF_en	F4
9	F4	{OUTPUT}									WAIT	2000	JankenF_en	F5
10	F5	WIN	%勝ち1:勝ち2%	learned Twirl				Joy	2	10	GET_ACTION_OPTION	I win!	JankenF_en	F6
11	F5	LOSE	%負け1:負け2%	QuickFormalBow				Sorrow	-2	%-2:-1%	GET_ACTION_OPTION	I lost.	JankenF_en	F6
12	F5	DRAW	あいこで								GET_ACTION_OPTION	It's a draw.	Janken1_en	FD
13	F6	{OUTPUT}											Janken3_en 3	

[1.8] F3W

Output value {OUTPUT}

I win!



No voice or command recognition performed,  
compares the output value string with the string in [recognized string].  
"OUTPUT" is replaced by the value of the output value, so it matches all output values.

**(5) scenario-JankenF\_en.csv**

In scenario-Janken3\_en.csv  
"I win!" appears at the bottom of the  
screen for 5 seconds.

## Tutorial: 2. About passing values between scenario files using output variable.

These are the methods for passing values between scenario files using output values.

The main points are as follows

- Confirm which values will be stored in the output value by checking the “[action] / [action option] / output values to be entered in the dialogue scenario” appendix.
- If you want to use the value of the output in the next scenario file, enter the [id] of the corresponding line in the next scenario file in the [next ID] field.

Note that strings are case-sensitive.

- If [recognized string] in the following scenario file is "{OUTPUT}", saved output values can be loaded into it.
- The string "{OUTPUT}", which is replaced by the output value, is also available in [action option].
- If you want to change the scenario according to the output values stored up to the previous scenario, specify the string for each output value in the [recognized string] field. (e.g., "TRUE", "FALSE", "ERROR", etc.).



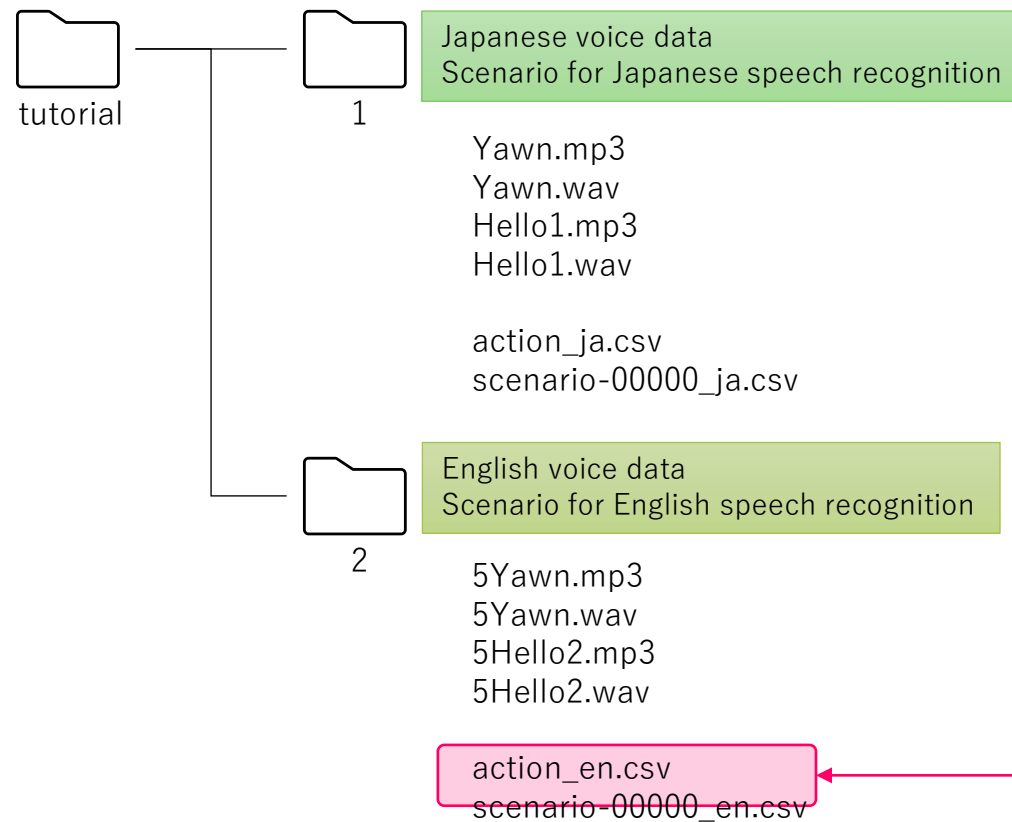
# Tutorial

## 3. About the action file



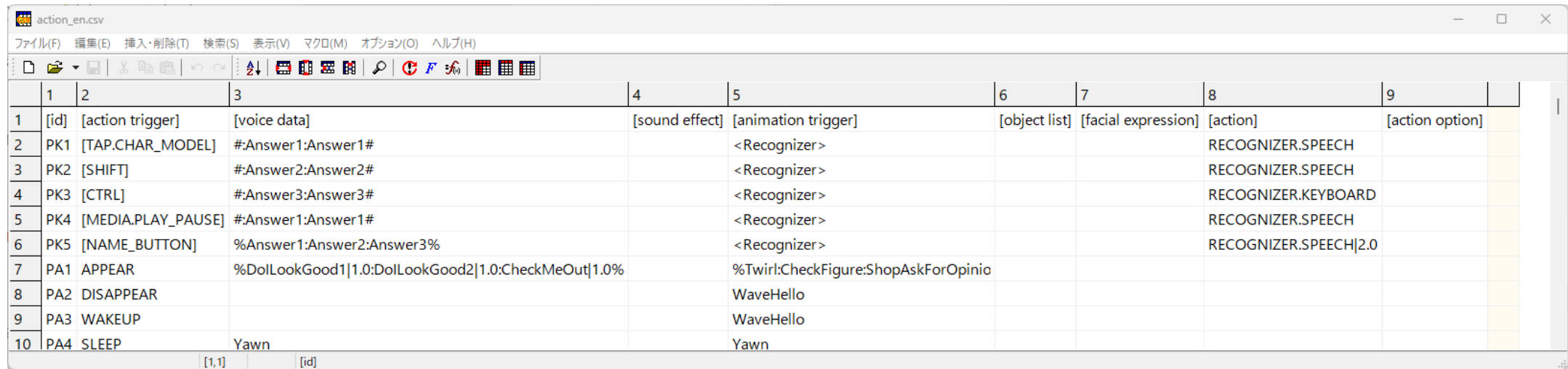
# Tutorial: 3. About the action file

- The voice and motion of the character when the user performs operations such as tapping the character or pressing a button, the functions to be performed are described in the “action\_\*.csv file”. Also some random actions in the character(s) idle state are described in the same “action\_\*.csv” file.



Open this file with an open CSV editor.  
(This document uses the Cassava Editor.)

# Tutorial: 3. About the action file



	1	2	3	4	5	6	7	8	9	
1	[id]	[action trigger]	[voice data]	[sound effect]	[animation trigger]	[object list]	[facial expression]	[action]	[action option]	
2	PK1	[TAP.CHAR_MODEL]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH		
3	PK2	[SHIFT]	#:Answer2:Answer2#		<Recognizer>			RECOGNIZER.SPEECH		
4	PK3	[CTRL]	#:Answer3:Answer3#		<Recognizer>			RECOGNIZER.KEYBOARD		
5	PK4	[MEDIA.PLAY_PAUSE]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH		
6	PK5	[NAME_BUTTON]	%Answer1:Answer2:Answer3%		<Recognizer>			RECOGNIZER.SPEECH 2.0		
7	PA1	APPEAR	%DolLookGood1 1.0:DolLookGood2 1.0:CheckMeOut 1.0%		%Twirl:CheckFigure:ShopAskForOpinio					
8	PA2	DISAPPEAR			WaveHello					
9	PA3	WAKEUP			WaveHello					
10	PA4	SLEEP	Yawn		Yawn					

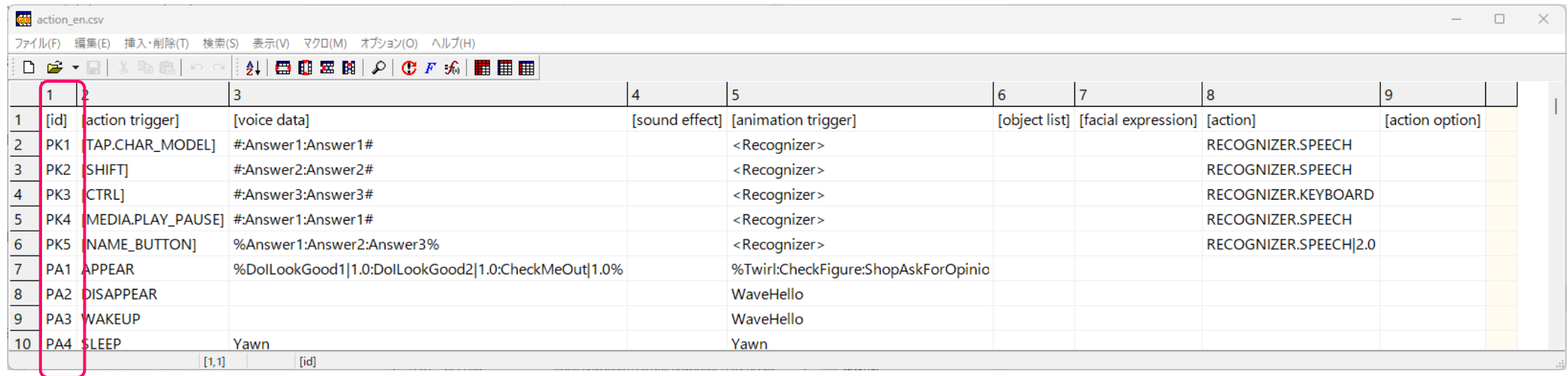
## (1) Open .¥tutorial¥2¥action\_en.csv with CSV editor.

The first line is an index what the value in each column (item) represents.  
The values of these items are explained on the next page.

Also, since this first line is inserted for explanatory purposes, don't forget to delete them when you finally creating dialogue scenario data.

Basically, each item is the same as in the dialogue scenario file.

# Tutorial: 3. About the action file



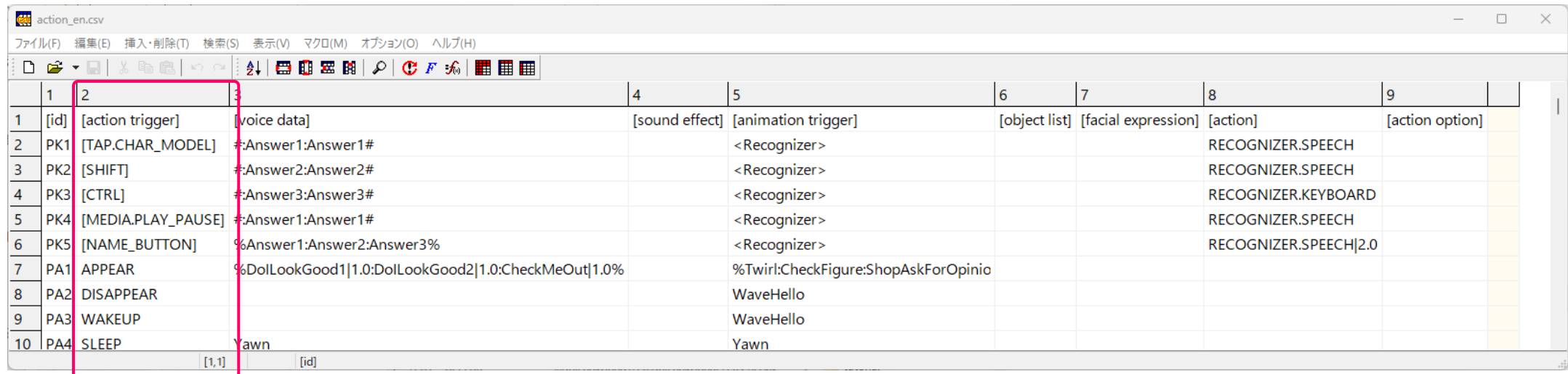
1	2	3	4	5	6	7	8	9
1	[id] action trigger]	[voice data]	[sound effect]	[animation trigger]	[object list]	[facial expression]	[action]	[action option]
2	PK1 TAP.CHAR_MODEL]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH	
3	PK2 SHIFT]	#:Answer2:Answer2#		<Recognizer>			RECOGNIZER.SPEECH	
4	PK3 CTRL]	#:Answer3:Answer3#		<Recognizer>			RECOGNIZER.KEYBOARD	
5	PK4 MEDIA.PLAY_PAUSE]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH	
6	PK5 NAME_BUTTON]	%Answer1:Answer2:Answer3%		<Recognizer>			RECOGNIZER.SPEECH 2.0	
7	PA1 APPEAR	%DolLookGood1 1.0:DolLookGood2 1.0:CheckMeOut 1.0%		%Twirl:CheckFigure:ShopAskForOpinio				
8	PA2 DISAPPEAR			WaveHello				
9	PA3 WAKEUP			WaveHello				
10	PA4 SLEEP	Yawn		Yawn				

(1) Open .¥tutorial¥2¥action\_en.csv with CSV editor.

## Column 1: [id]

Enter any string of alphanumeric characters to identify the response.  
In ver. 1.7.0, this value is not used, so there are no restrictions.

# Tutorial: 3. About the action file



	1	2	3	4	5	6	7	8	9
1	[id]	[action trigger]	[voice data]	[sound effect]	[animation trigger]	[object list]	[facial expression]	[action]	[action option]
2	PK1	[TAP.CHAR_MODEL]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH	
3	PK2	[SHIFT]	#:Answer2:Answer2#		<Recognizer>			RECOGNIZER.SPEECH	
4	PK3	[CTRL]	#:Answer3:Answer3#		<Recognizer>			RECOGNIZER.KEYBOARD	
5	PK4	[MEDIA.PLAY_PAUSE]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH	
6	PK5	[NAME_BUTTON]	%Answer1:Answer2:Answer3%		<Recognizer>			RECOGNIZER.SPEECH 2.0	
7	PA1	APPEAR	%DolLookGood1 1.0:DolLookGood2 1.0:CheckMeOut 1.0%		%Twirl:CheckFigure:ShopAskForOpinio				
8	PA2	DISAPPEAR			WaveHello				
9	PA3	WAKEUP			WaveHello				
10	PA4	SLEEP	Yawn		Yawn				

(1) Open .¥tutorial¥2¥action\_en.csv with CSV editor.

## Column 2: [action trigger]

**You can only enter a fixed string here.**

See the list on the next page for details.

**If there are multiple rows with the same [action trigger], the action is randomly selected from among the applicable rows.**

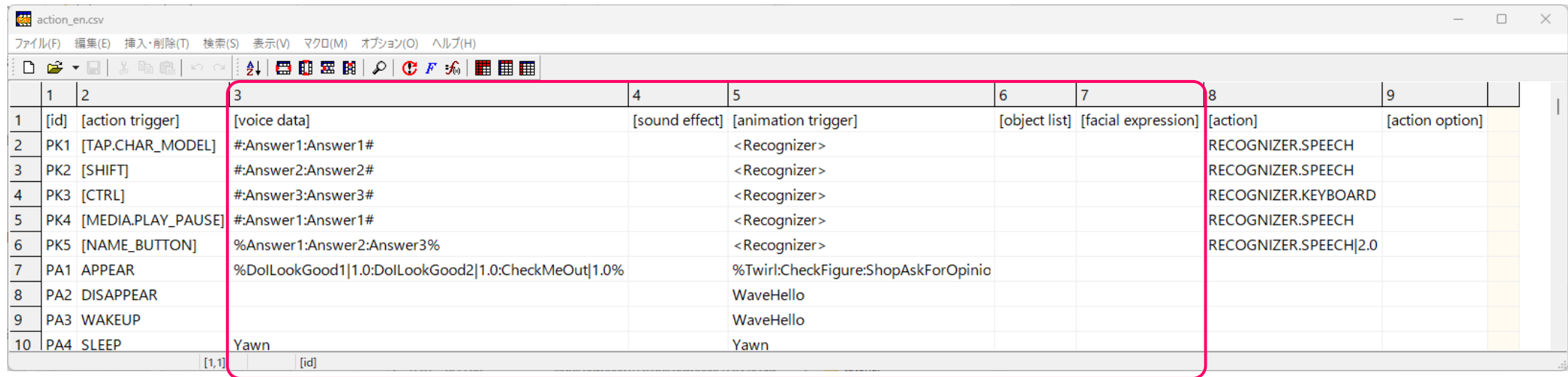


# Tutorial: 3. About the action file

action trigger	screen	contents
[TAP.CHAR_MODEL]	Character display screen	Executed when the character is tapped.
[SHIFT]	Character display screen	Executed when the [SHIFT] key is pressed. (Bluetooth Keyboard)
[CTRL]	Character display screen	Executed when the [CTRL] key is pressed. (Bluetooth keyboard)
[MEDIA.PLAY_PAUSE]	Character display screen	Executed when the media playback button is pressed on a Bluetooth media remote control or other device.
[NAME_BUTTON]	Character Selection and Settings Screen	Executed when the “name” button is pressed. But the feature cannot be changed.
APPEAR	Character display screen	Executed when the character is displayed.
DISAPPEAR	Character display screen	Executed when the character hides by saying "good night."
WAKEUP	Character display screen	Executed when the character wakes up.
SLEEP	Character display screen	Executed when the character sleeps (below the set value for remaining battery power or above the set value for inactivity time).
RECOVERD	Character display screen	Executed when the character wakes up (battery power restored).
HELLO	Character display screen	(Not used)
GOT_IT	Character display screen	(Not used)
NOT_UNDERSTOOD	Character display screen	Executed when the character something was not understood.
NOT_FOUND	Character display screen	Executed when the character could not find something.
SUCCEDED	Character display screen	(Not used)
FAILED	Character display screen	(Not used)
TOO_CLOSE	Character display screen Character Selection and Settings Screen	What the character says something when the camera is too close to them.
WAKEUP_CALL	Character display screen	Executed in the wakeup call feature.
MUSIC_IMPRESSION	Character display screen	Executed after music playback.
IDLE	Character display screen	Randomly executed when a character is waiting (in idle state).



# Tutorial: 3. About the action file



	1	2	3	4	5	6	7	8	9
1	[id]	[action trigger]	[voice data]	[sound effect]	[animation trigger]	[object list]	[facial expression]	[action]	[action option]
2	PK1	[TAP.CHAR_MODEL]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH	
3	PK2	[SHIFT]	#:Answer2:Answer2#		<Recognizer>			RECOGNIZER.SPEECH	
4	PK3	[CTRL]	#:Answer3:Answer3#		<Recognizer>			RECOGNIZER.KEYBOARD	
5	PK4	[MEDIA.PLAY_PAUSE]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH	
6	PK5	[NAME_BUTTON]	%Answer1:Answer2:Answer3%		<Recognizer>			RECOGNIZER.SPEECH 2.0	
7	PA1	APPEAR	%DolLookGood1 1.0:DolLookGood2 1.0:CheckMeOut 1.0%		%Twirl:CheckFigure:ShopAskForOpinio				
8	PA2	DISAPPEAR			WaveHello				
9	PA3	WAKEUP			WaveHello				
10	PA4	SLEEP	Yawn		Yawn				

(1) Open .¥tutorial¥2¥action\_en.csv with CSV editor.

## Columns 3-7: [voice data], [sound effect], [animation trigger], [object list], [facial expression]

The values and description method will be the same as in the dialogue scenario file.

Specifying multiple audio data files using the enclosing character "%#&\$" and a one-byte colon ":" will cause the following behavior.

%value 1:value 2:…:value n% : Select value randomly.

#value 1:value 2:…:value n# : The value is chosen according to the emotion value by dividing the minimum emotion value (-24) to the maximum emotion value (+24) into n. (rounded down to the nearest whole number)

&value 1:value 2:…:value n& : The value is chosen according to affection value by dividing the maximum affection (1000) divide into n. (rounded down to the nearest whole number)

\$value 1:value 2:…:value n\$ : The value is selected according to the current time by dividing 0:00 AM. to 11:59 PM. per day into n equal parts of 1440 minutes.

# Tutorial: 3. About the action files

	1	2	3	4	5	6	7	8	9
1	[id]	[action trigger]	[voice data]	[sound effect]	[animation trigger]	[object list]	[facial expression]	[action]	[action option]
2	PK1	[TAP.CHAR_MODEL]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH	
3	PK2	[SHIFT]	#:Answer2:Answer2#		<Recognizer>			RECOGNIZER.SPEECH	
4	PK3	[CTRL]	#:Answer3:Answer3#		<Recognizer>			RECOGNIZER.KEYBOARD	
5	PK4	[MEDIA.PLAY_PAUSE]	#:Answer1:Answer1#		<Recognizer>			RECOGNIZER.SPEECH	
6	PK5	[NAME_BUTTON]	%Answer1:Answer2:Answer3%		<Recognizer>			RECOGNIZER.SPEECH 2.0	
7	PA1	APPEAR	%DolLookGood1 1.0:DolLookGood2 1.0:CheckMeOut 1.0%		%Twirl:CheckFigure:ShopAskForOpinio				
8	PA2	DISAPPEAR			WaveHello				
9	PA3	WAKEUP			WaveHello				
10	PA4	SLEEP	Yawn		Yawn				

(1) Open .¥tutorial¥2¥action\_en.csv with CSV editor.

## Columns 8 and 9: [action], [action option].

The following items are available in ver.1.6.0.

action	action option	Contents
RECOGNIZER.SPEECH	-	Starts speech recognition. You can connect with " " to describe the wait time to start. The unit is (second). [animation trigger] should be "<Recognizer>".
RECOGNIZER.KEYBOARD	-	Starts command input recognition. Other are the same as above.
POST, TWEET.REPLY	-	The character tweet or reply to a tweet if user is using Twitter integration.

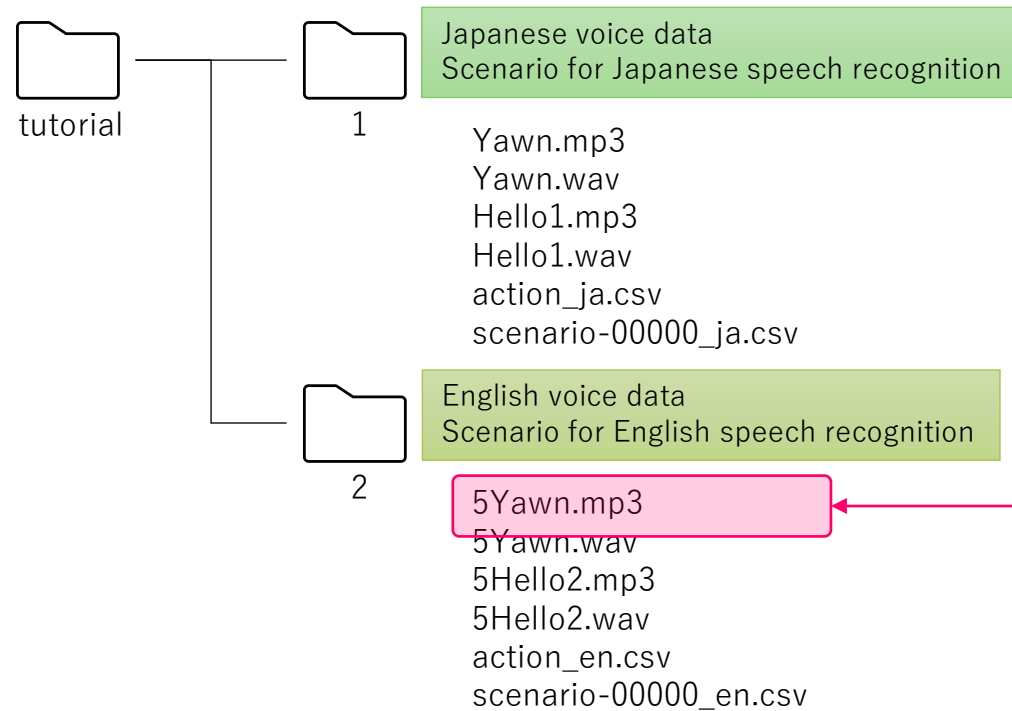
# Tutorial

## 4. Prepare the voice data



## Tutorial: 4. Prepare the voice data

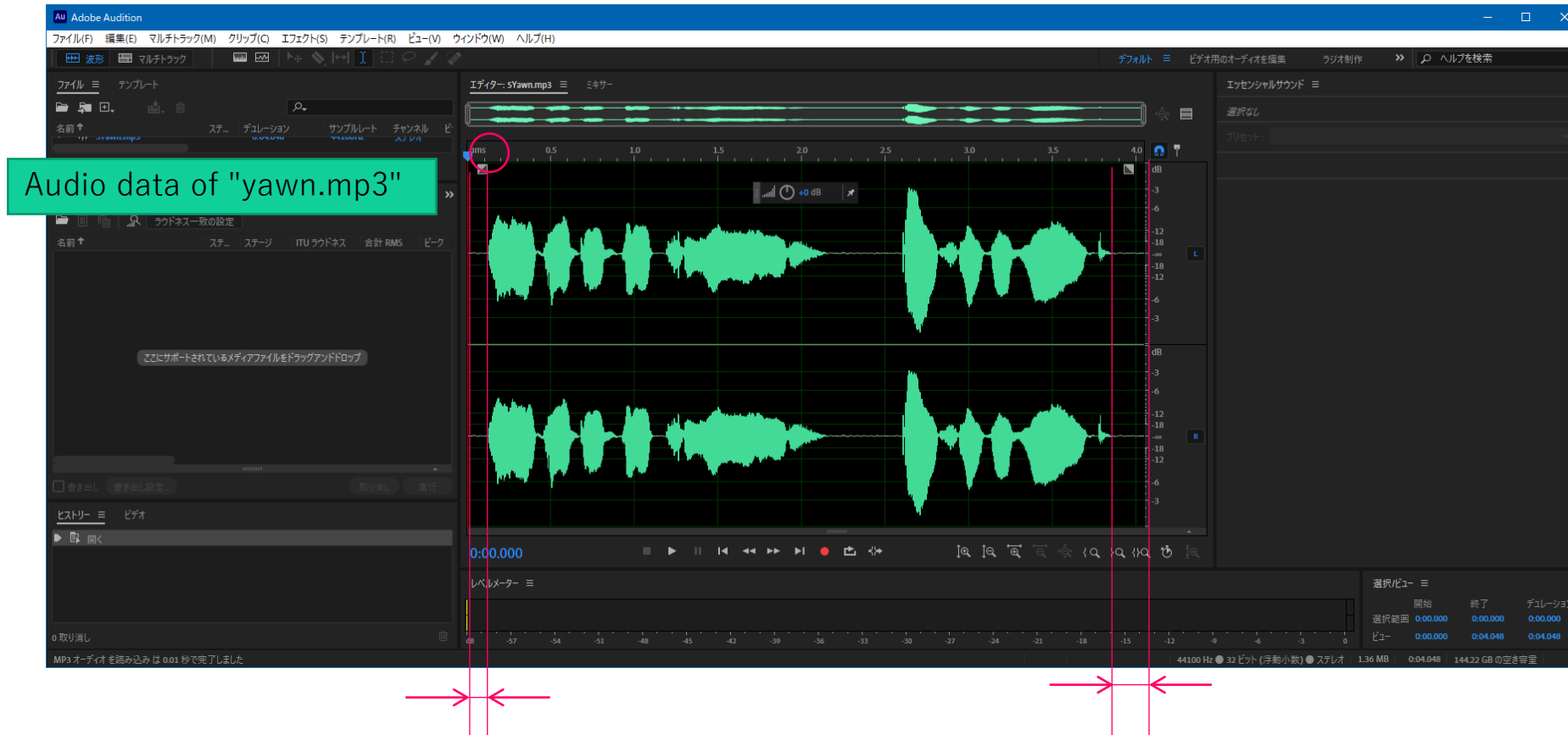
- Prepare the voice data suitable for the dialogue scenario.
- We have included voice data for tutorial purposes, and will use it to explain the points of creating voice data.



Open this file in your MP3 editing software. In this manual, we use Adobe Audition, but you can use any free software like mp3DirectCut , bearaudio, and so on.

# Tutorial: 4. Prepare the voice data

## (1) Cut (trim) the silence portion of the voice data.

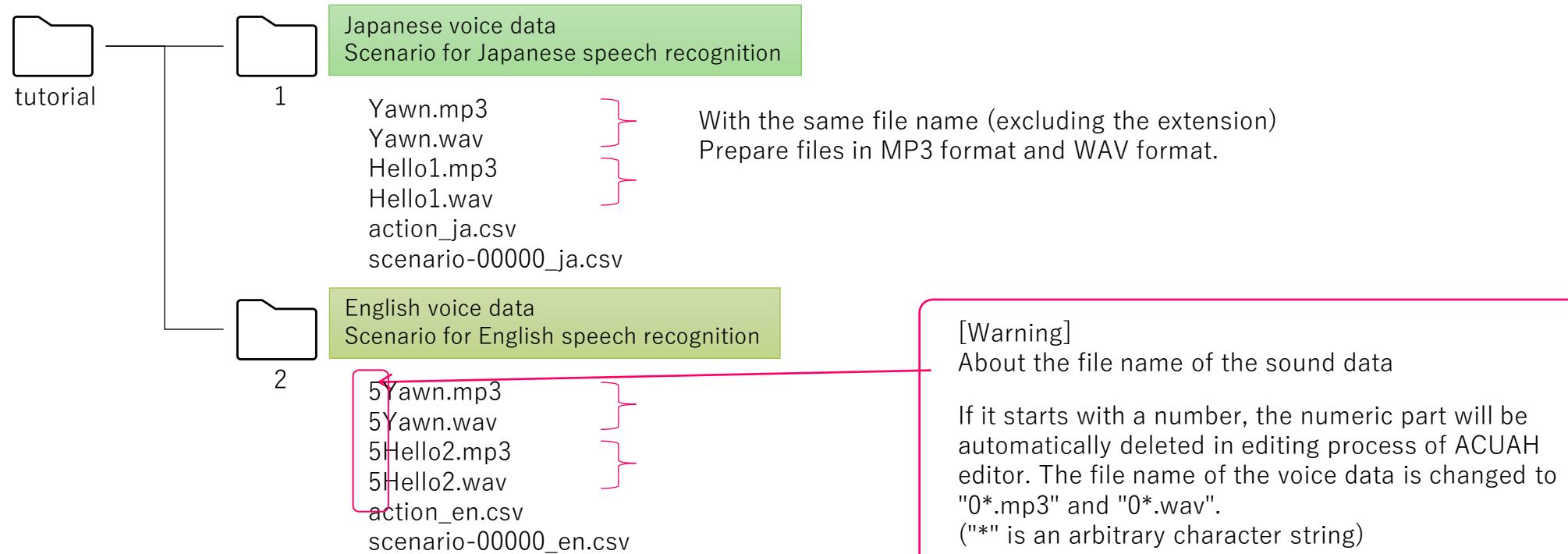


# Tutorial: 4. Prepare the voice data

## (2) Prepare voice data in MP3 or WAV format.

Because ACUAH editor cannot play MP3 format audio data, please prepare the data created by (1) in WAV format with the same file name.

Also, adjust the volume (sound pressure) so that there is not too much difference between multiple voice data.



# Tutorial: 4. Prepare the voice data

This completes the preparation of the voice data.

- If you load the voice data into ACUAH and play it, the volume may be too low.  
If this happens, try adjusting the sound pressure (dB) again using audio editor like EcoDecoTool.
- When adjusting the volume of multiple audio data, it is helpful to have software with a “Match Loudness Settings” feature for multiple audio files, such as Adobe Audition.  
-14LKFS(LUFS) is a standard value.





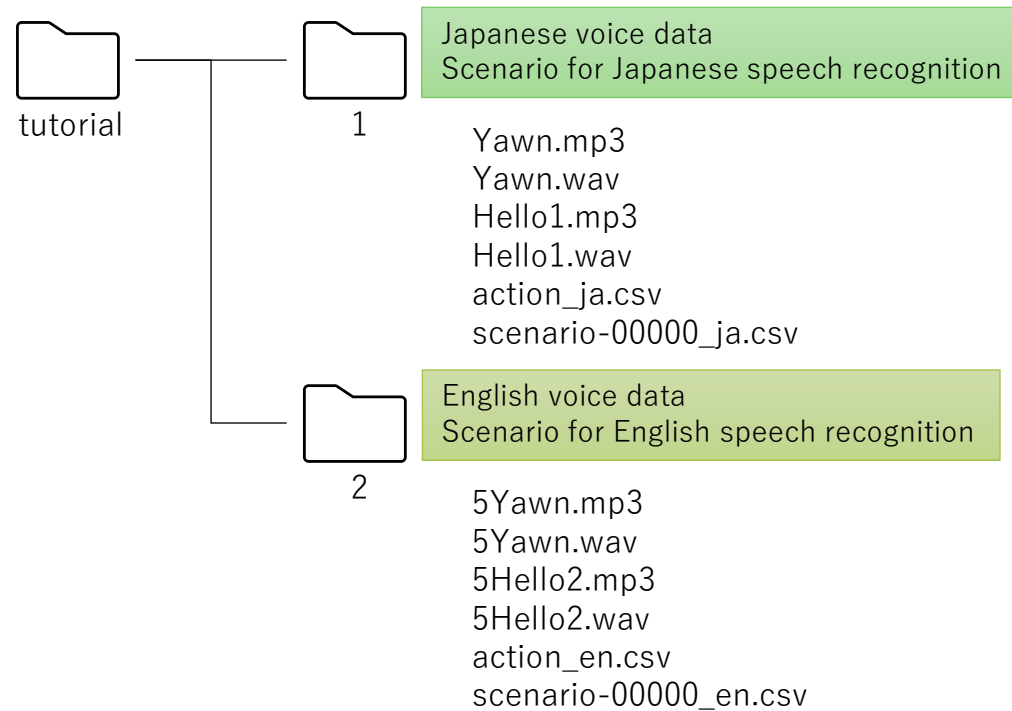
# Tutorial

## 5. Start ACUAH editor



# Tutorial: 5. Start ACUAH editor

- Save the data you prepared in “1. Create a dialogue scenario file” and “2. Prepare the voice data” in one folder.
- The following are saved in folders "1" and "2" for this tutorial.  
(MP3 format, WAV format voice data, "scenario-\*\_\*\*.csv" dialogue scenario file)



# Tutorial: 5. Start ACUAH editor

## (1) Set the image quality and working directory.

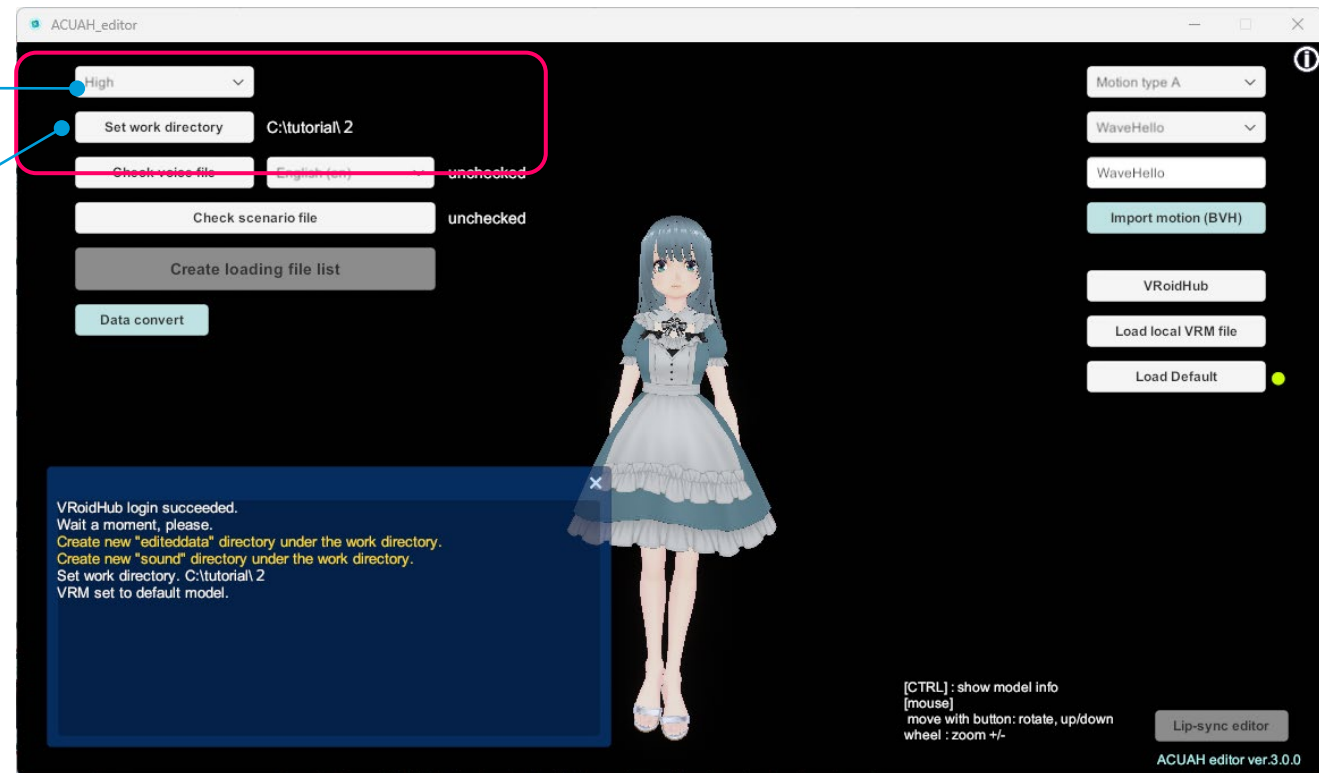
**Image quality setting**  
(Low) Very Low  
(High) Ultra

Please select the image quality.  
By default, is set to "High".

### Set work directory

It is set to the working directory.  
Click "Set work directory."  
An Explorer window will launch.  
Please select the folder where the  
created voice data and dialogue scenario files  
are saved in.

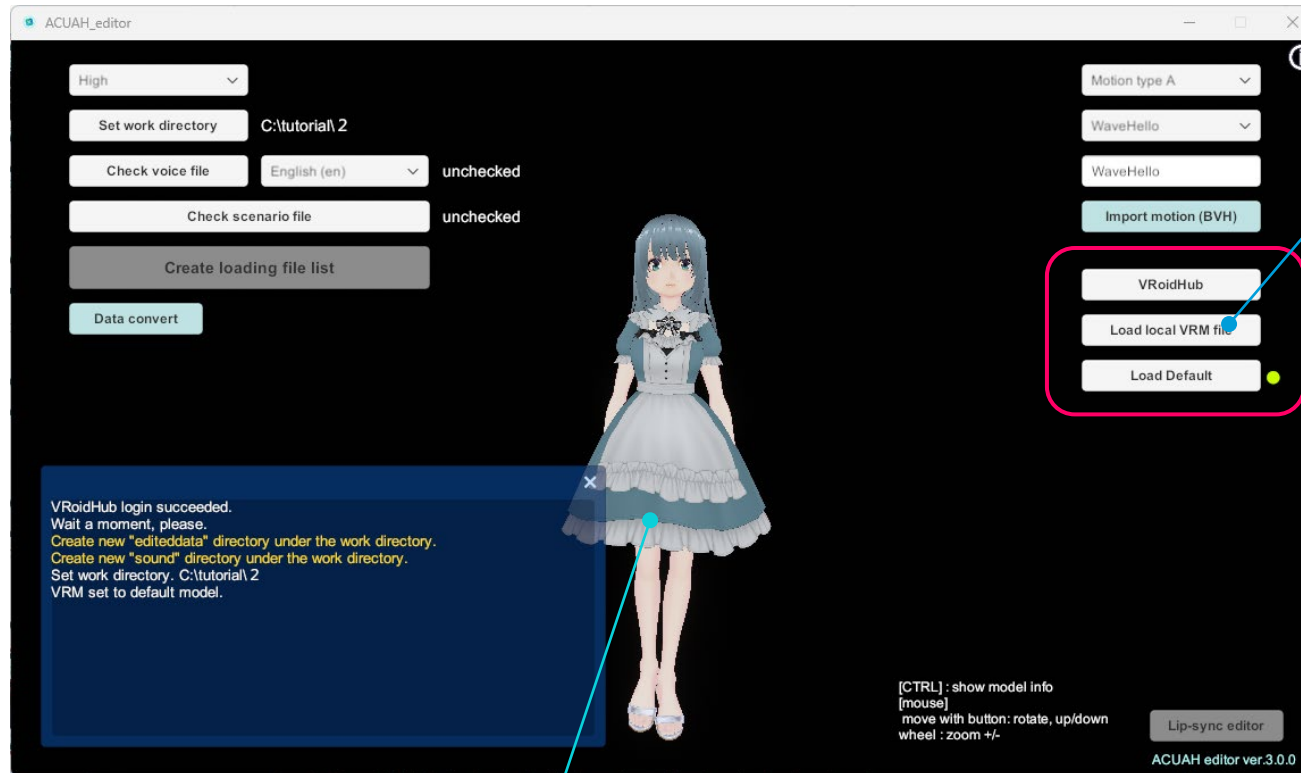
In this tutorial, please select and set the  
.`C:\tutorial\2`



And in this process, it will create a directory called "editeddata" under your working directory.

# Tutorial: 5. Start ACUAH editor

## (2) Select a character.



### 3D character Platform Specification

To the right of the one you are selecting ● will be attached.

### VRoidHub

Download the character model from VRoidHub.

### Load local VRM File

Load VRM file from PC local directory

### Load Default

Load Default (standard) VRM model.

Select a character In any of the above ways.

You can still use the application's standard character model.

The first time you start  
Application standard VRM character model  
(ACUAH (short sleeve)) will be displayed.



# Tutorial: 5. Start ACUAH editor

## (3) Check the voice data file (proceed to the lip-sync data edit screen).

### Lip-sync data language settings

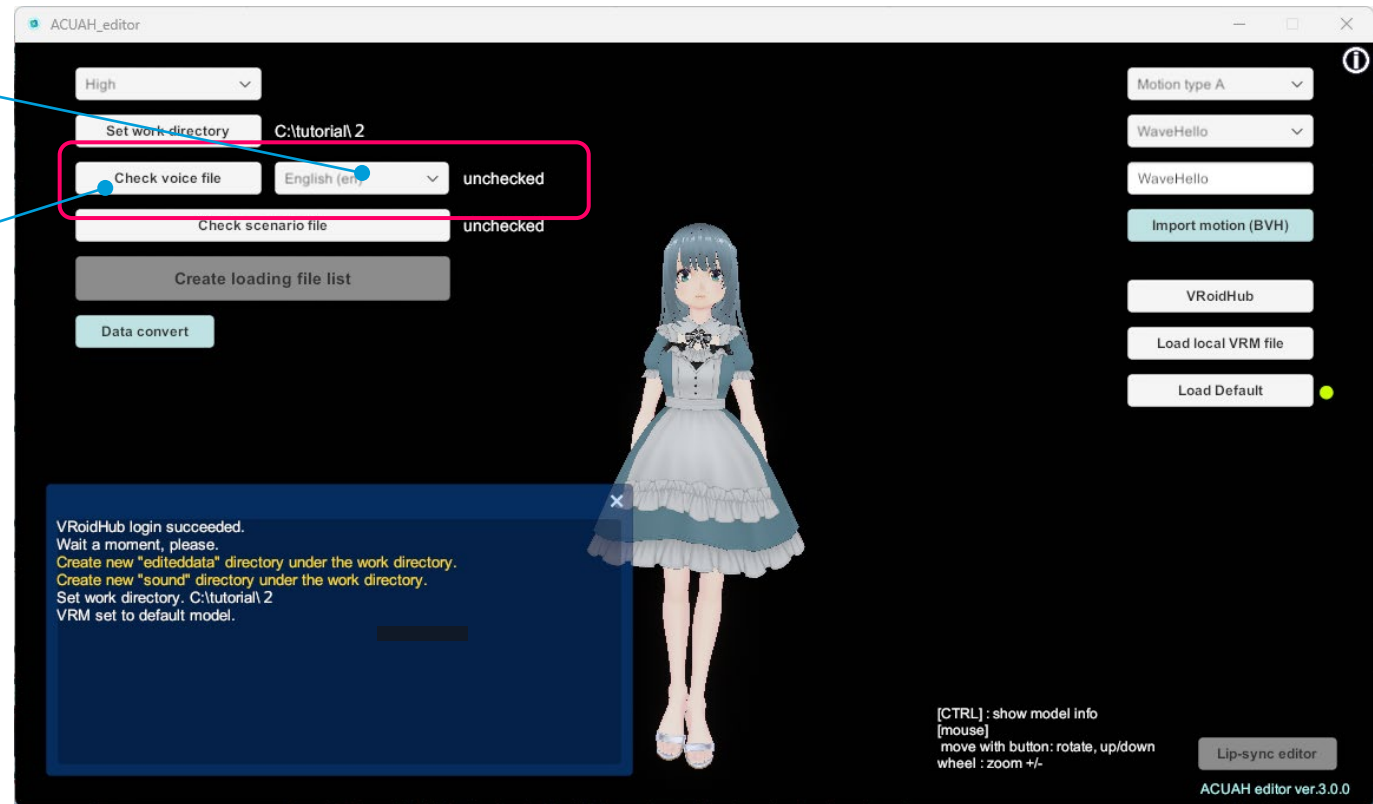
The data under ¥tutorial¥2 directory is English voice data.  
So in this tutorial please set language setting to "English(en)".

### Check voice file

voice file and lip-sync data consistency check

First, to create lip sync data, click on "Check voice file."

Check for consistency with the voice date and lip-sync data under work directory.  
But, at first it will be fail and the error (**Failed**) shown.



# Tutorial: 5. Start ACUAH editor

## (3) Check the voice data file (proceed to the lip-sync data edit screen).

In the message window, you will see the following  
The message is displayed.

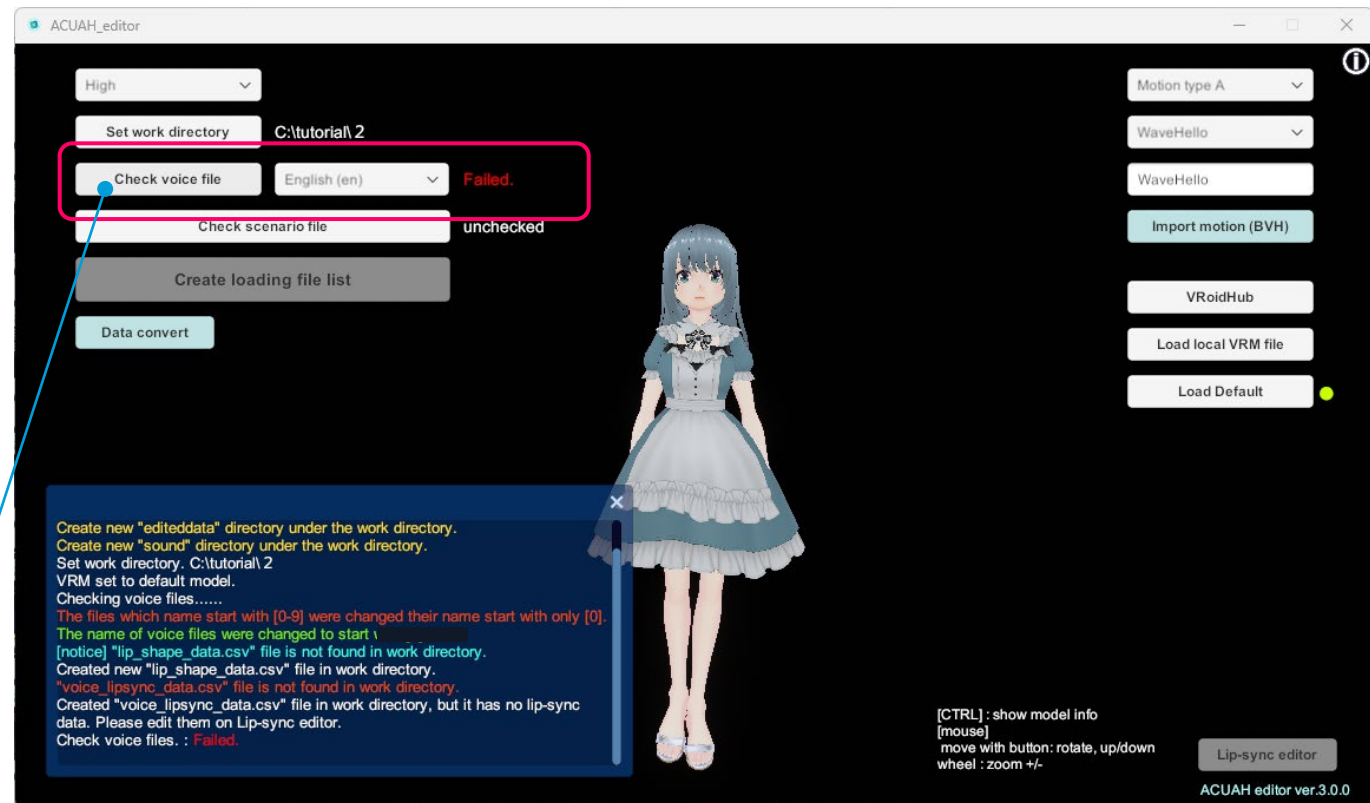
The name of voice files were changed  
to start with [0].

[notice] "lip\_shape\_data.csv" file is not found  
in work directory.

Created new "lip\_shape\_data.csv"  
file in work directory.

"voice\_lipsync\_data.csv" file is not found  
in work directory.

Created "voice\_lipsync\_data.csv" file in work  
directory, but it has no lip-sync data.  
Please edit then on Lip-sync editor.



The initial check resulted in an error "Failed.", but this check made the necessary files created in work directory.

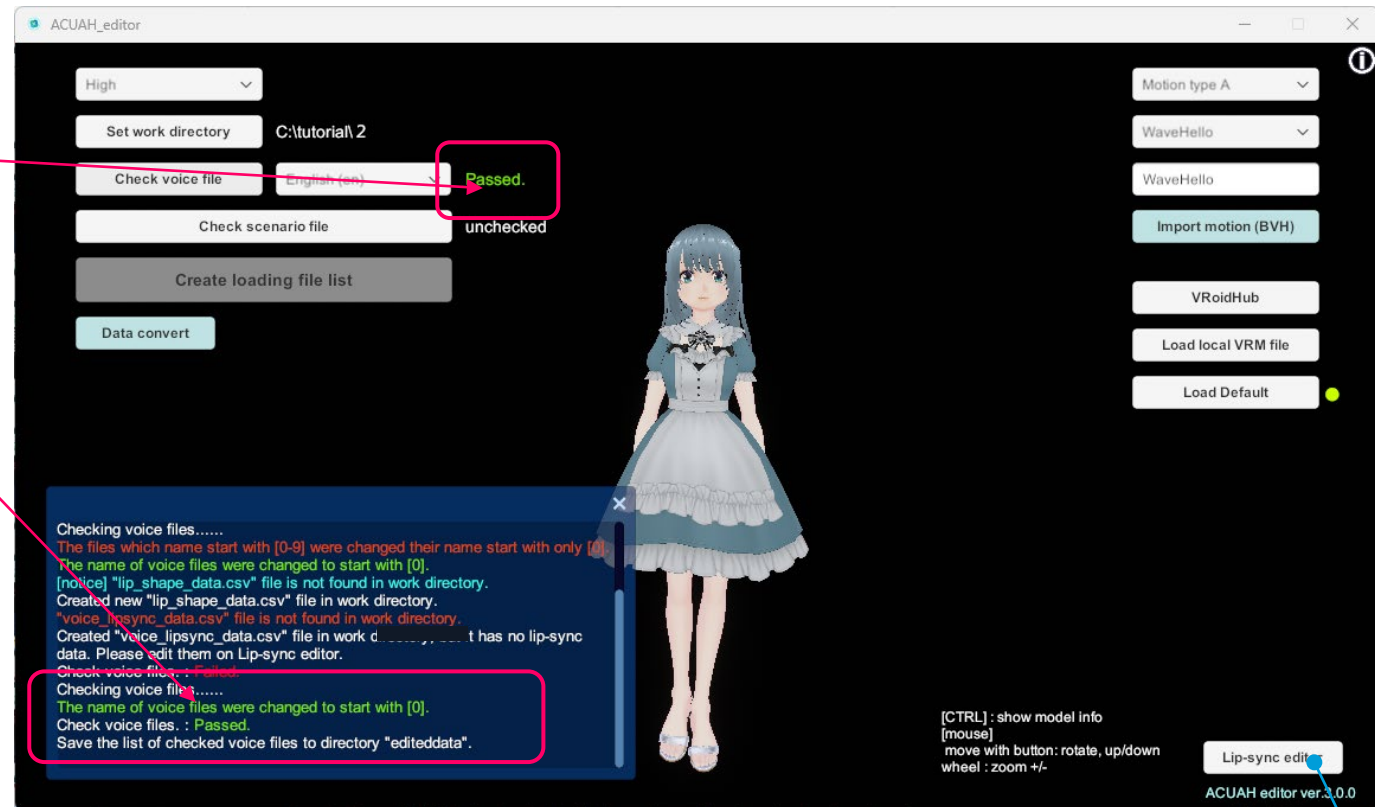
Then, click "Check voice file" again.

# Tutorial: 5. Start ACUAH editor

(3) Check the voice data file (proceed to the lip-sync data edit screen).

Now it shows "Passed."

If you pass, the "Lip-sync editor" button at the bottom right of the screen will be clickable.



Click on the "Lip-sync editor" button.

Go to  
lip-sync data  
edit screen

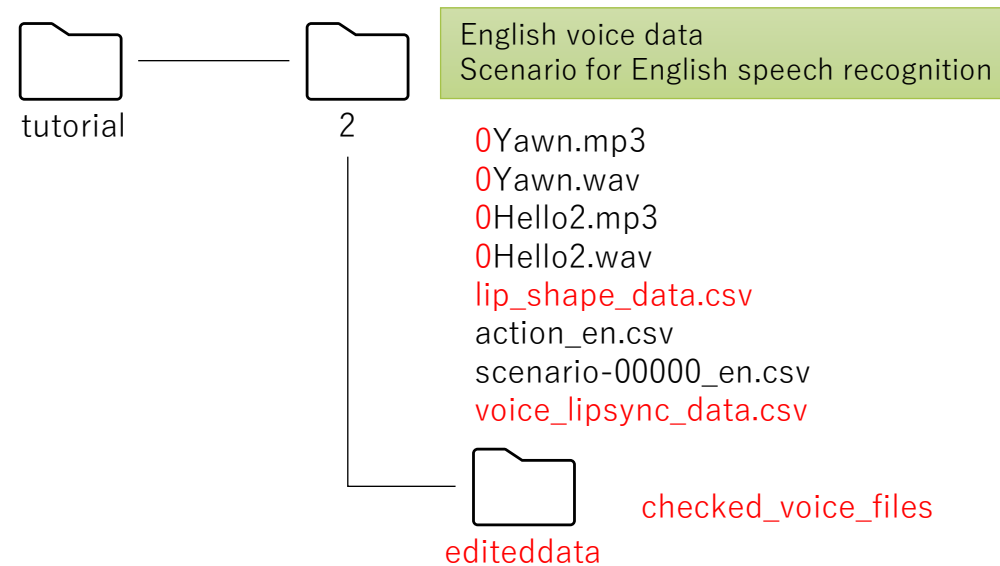
# Tutorial: 5. Start ACUAH editor

This completes the startup of ACUAH editor.

- You are now ready to proceed to the lip sync data editing screen.

. ¥tutorial¥2

The folder status (file storage status) of





# Tutorial

## 6. Edit lip-sync data



# Tutorial: 6. Edit lip-sync data

## (1) Screen description

### Lip-sync data language settings

Please select language according to lip-sync data that you create.  
The creation process varies depends on the language.

### Image quality setting (Low) Very Low

(High) Ultra

Please select the image quality.

In ACUAH, lip-sync mouth movement varies depending on the image quality setting.

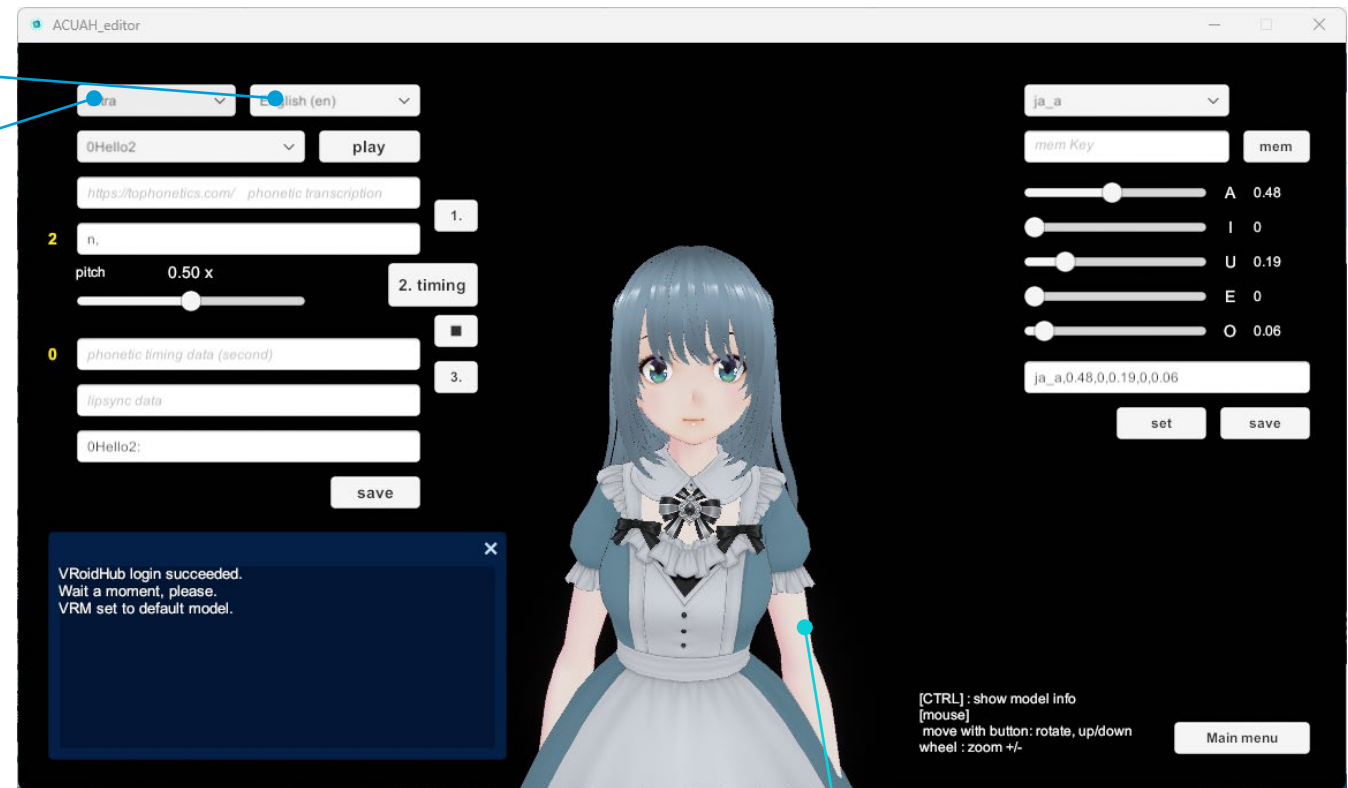
Very Low, Low, Medium:

It has a limited animation-like expression.

High, Very High, Ultra

Make the shape of the mouth change smoothly  
It has a full animation-like expression.

Since “ACUAH editor” lip-sync expression does the same thing as “ACUAH”, please change the image quality setting to confirm.



The character you selected will be displayed.



# Tutorial: 6. Edit lip-sync data

## (2) Play voice data.

### Voice data selection drop-down list

The list includes

0Hello2  
0Yawn

You will see two options, select  
Click the "play" button on the right.

### Voice data playback button

Since the lip-sync data has not been created yet,  
the voice data will only be played.



First, we create lip-sync data for the "Hello" voice data.  
**Select "0Hello2"** from the voice data selection drop-down list.

# Tutorial: 6. Edit lip-sync data

## (3) Get the pronunciation symbol of voice data.

### Text (pronunciation symbol) input field for voice data

In English, use the diacritics.

If you play the voice data "0Hello2", you can hear "Hi there."

Change this phrase to the pronunciation symbol by the following pages and others.

<https://tophoneics.com/>

"hai ðeə."

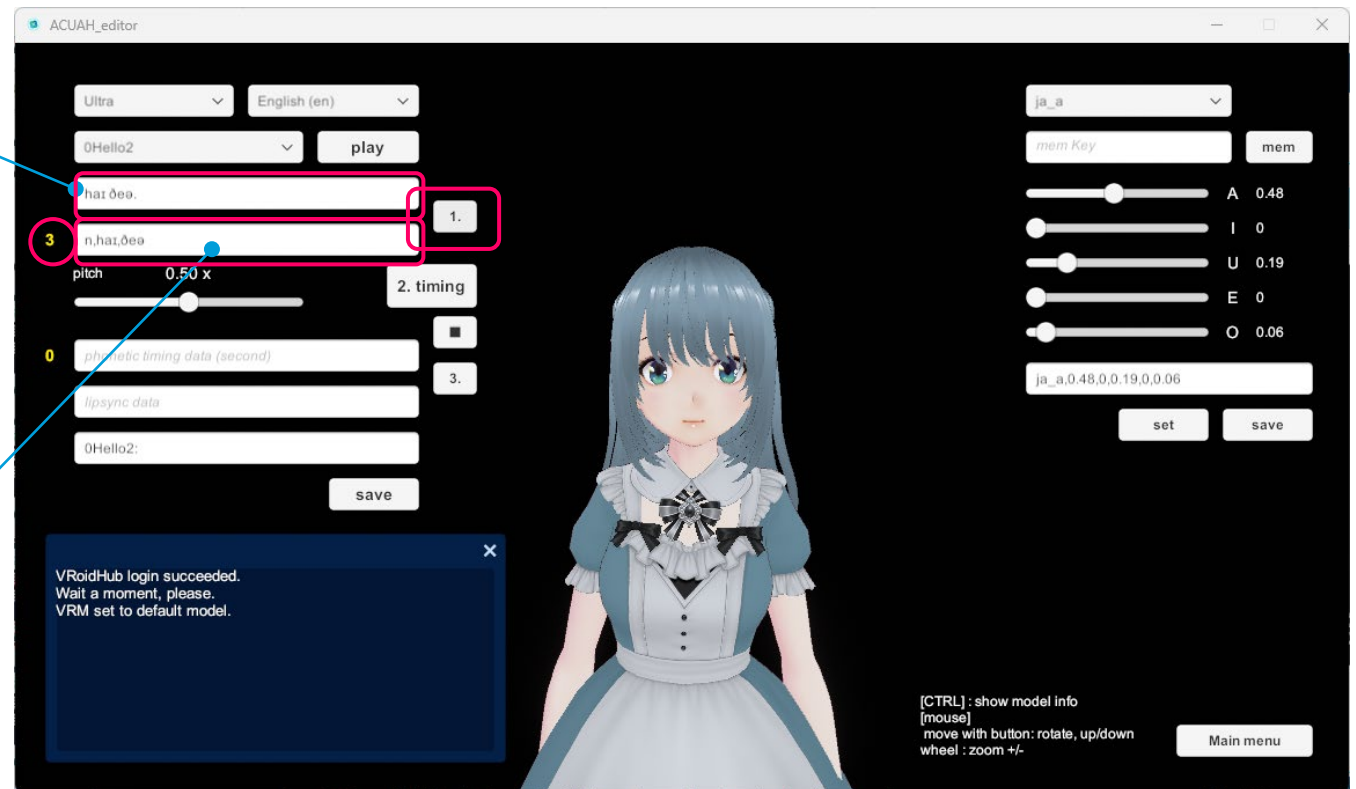
Then copy it to the text (pronunciation symbol) input field of the voice data and click on the button "1."

### Syllable-by-syllable diacritical marks data

In the syllable-by-syllable diacritic data field, "n,hai,ðeə" will be displayed.

### [Warning]

The first string "n" in syllable-by-syllable diacritical marks data field is for a silence time padding, so please not to delete.



# Tutorial: 6. Edit lip-sync data

## (4) Acquisition of pronunciation timing "0Hello2"

Next, acquisition of the pronunciation timing of each syllable.

### Timing acquisition button

Click once on "2. timing."

At a very slow rate, the voice of "0Hello2" will be played.

### Pitch slider

(You can change the playback speed by the pitch slider.  
The initial value is 0.50x.)

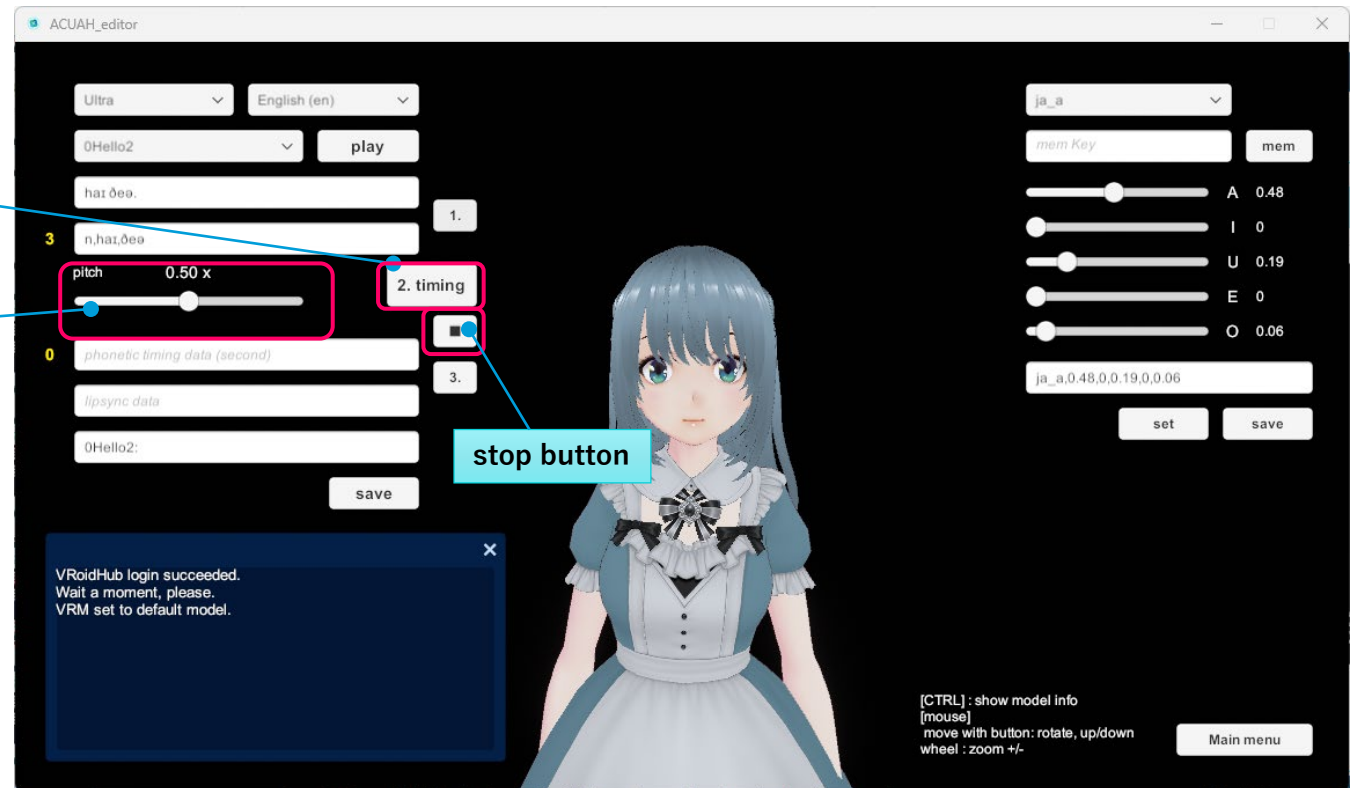
The **timing acquisition button** is

**1st click: Start playing voice data**

**Second and subsequent clicks: getting the timing**

(Obtaining time from voice data playback)

You can interrupt the timing acquisition  
by clicking the "■" stop button.



# Tutorial: 6. Edit lip-sync data

## (4) Acquisition of pronunciation timing "0Hello2"

Let's actually get the timing.

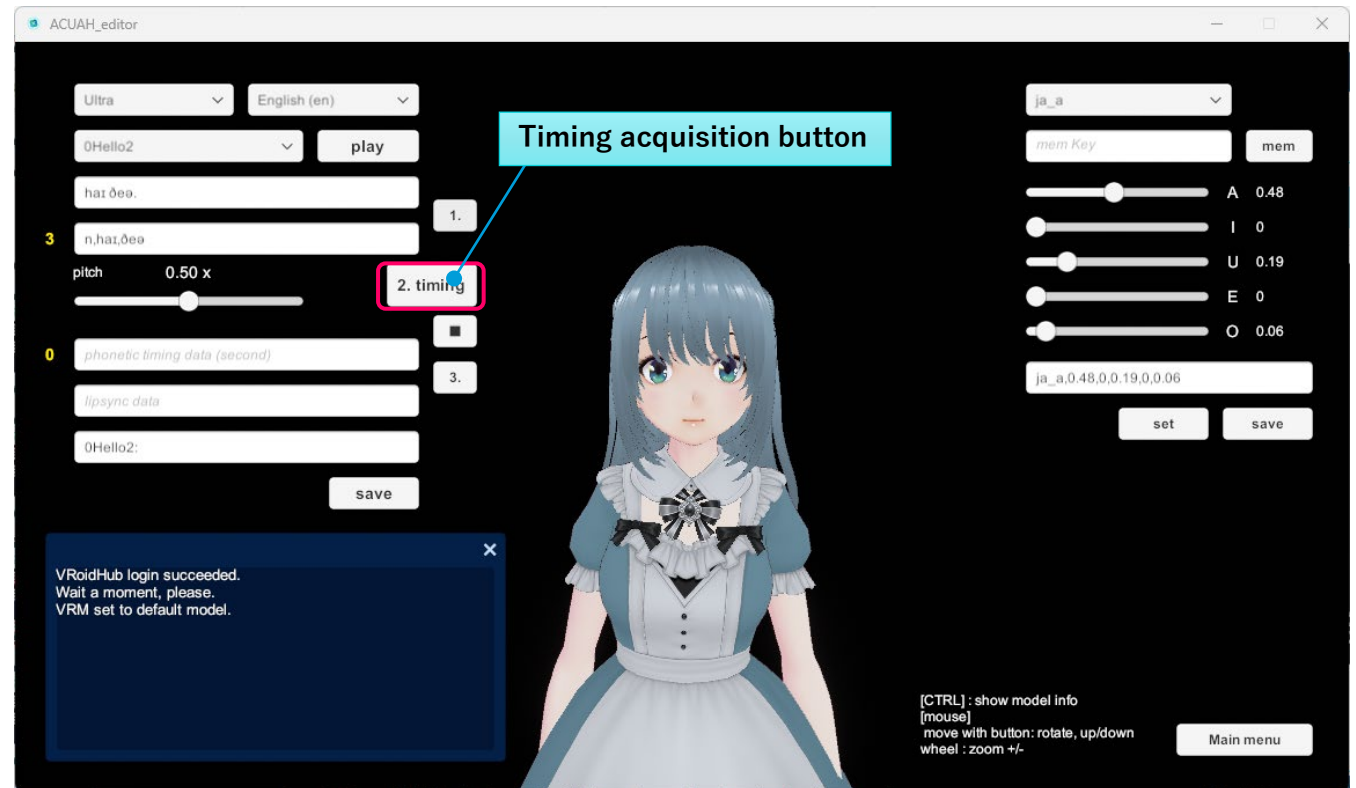
Click "2. timing" to start playback. (1)

Click "2. timing" when you hear the "Hi" (2)

Click "2. timing" when you hear the "there" (3)

Click a total of three times.

Try a few times until you think you've got it right.  
If you feel that the playback is too fast (too slow),  
adjust the playback speed by the pitch slider.



# Tutorial: 6. Edit lip-sync data

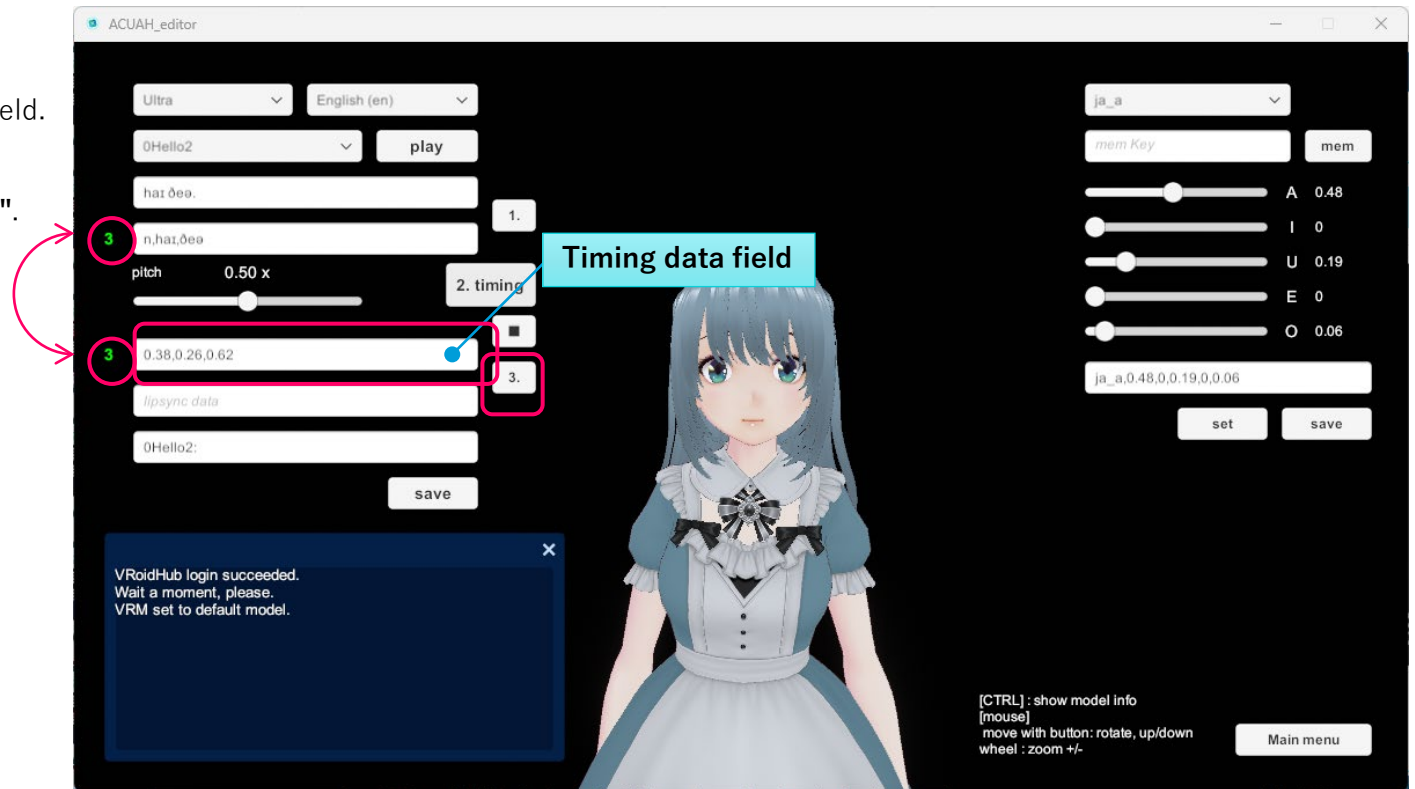
## (4) Acquisition of pronunciation timing "0Hello2"

If you can get the timing rightly,  
the time data is automatically entered in the timing data field.

Also, **if the number of clicks is correct,**  
**the number on the left of the timing data field marks "3".**

This number must be the same with  
the numbers on the left of the  
"Syllable-by-syllable diacritical marks data."  
(If they are the same, they will be green.)

**If it's OK, please click on the button "3."**



# Tutorial: 6. Edit lip-sync data

## (4) Acquisition of pronunciation timing "0Hello2"

When click on "3.",  
the value is displayed in the lip-sync data field.  
This is the lip sync data.

Lip-sync data field

Let's click the "voice data playback button" again.

The character should have lip-synced to the voice.

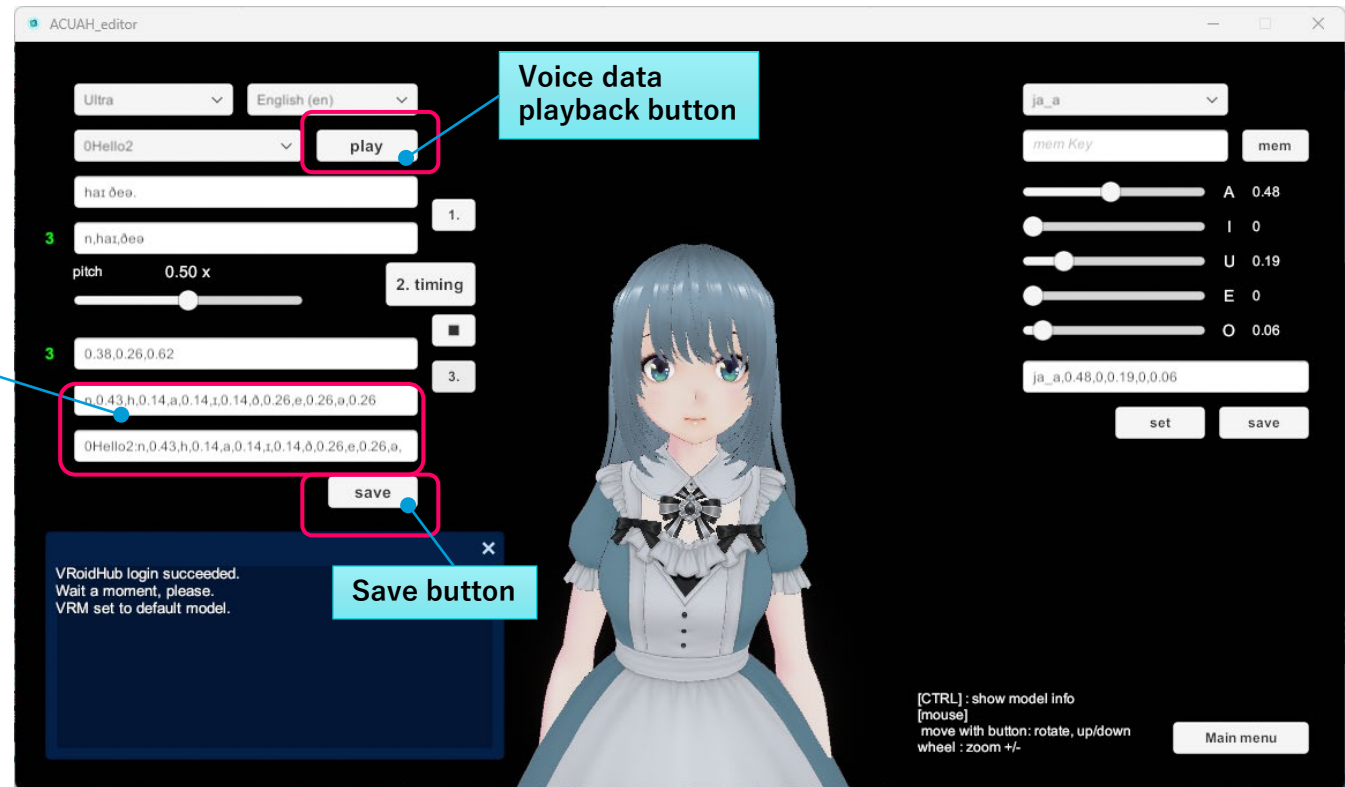
Watch the character's mouth move.

If you are OK, click the "save" button to save the file.

Save the lip sync data you have created.

(It will be written to "voice\_lipsync\_data.csv".)

The above is the lip-sync data for "Hello2" has been created.  
Continue to make lip-sync data for "Yawn".



Voice data  
playback button

Save button



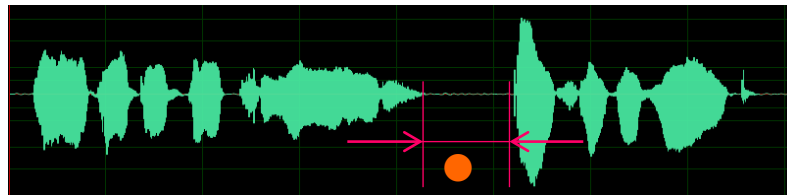
# Tutorial: 6. Edit lip-sync data

## (4) Acquisition of pronunciation timing "0Yawn"

The data of "0Yawn" is the same procedure as "0Hello2".  
But this data comes with a caveat.

"I got a bit tired. I'll take a break."

The voice data waveform is shown below.



I got a bit tired.

I'll take a break.

In the part of the ●, the character's mouth should be closed.  
This is the silent time.

**If there is a silent period where the character should have closed their mouth, you have to acquire the timing of that silence time.**



# Tutorial: 6. Edit lip-sync data

## (4) Acquisition of pronunciation timing "0Yawn"

### Text (pronunciation symbol) input field for voice data

If you play the voice data "0Yawn", you can hear "I got a bit tired. I'll take a break."

Change this phrase to the pronunciation symbol by the following pages and others.

<https://tophoneitics.com/>

"aɪ gɒt ə bɪt 'təɪəd, aɪl teɪk ə breɪk."

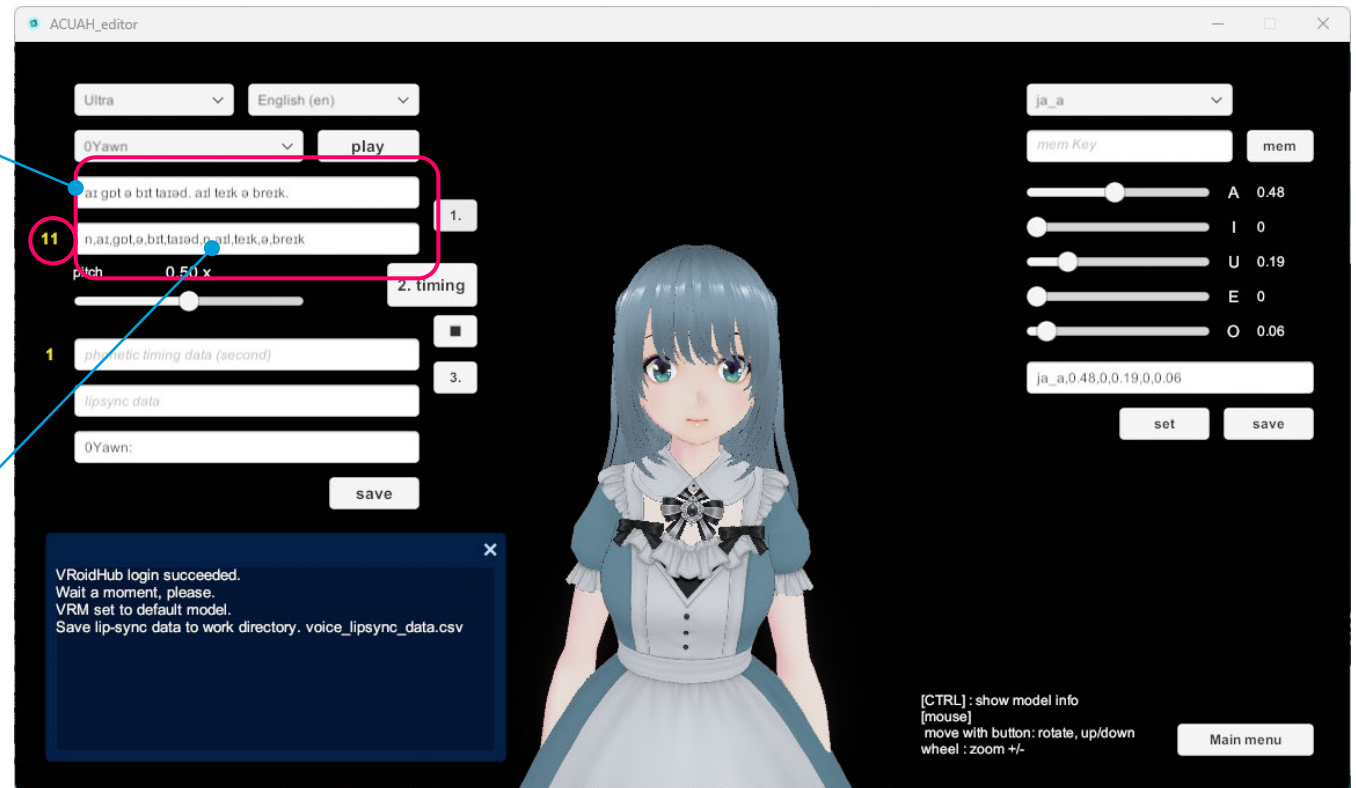
Then copy it to the text (pronunciation symbol) input field of the voice data and click on the button "1."

### Syllable-by-syllable diacritical marks data

"n,aɪ,gɒt,ə,bɪt,təɪəd,n,aɪl,teɪk,ə,breɪk"

If the pronunciation symbol text is

"n," (period and half-width space), it is replaced to "n,".



# Tutorial: 6. Edit lip-sync data

## (4) Acquisition of pronunciation timing "0Yawn"

Let's actually get the timing.

Listen to the voice data and try it a few times.

Click "2. timing" to start playback. (1)

Click "2. timing" when you hear the "I" (2)

Click "2. timing" when you hear the "got" (3)

Click "2. timing" when you hear the "a" (4)

Click "2. timing" when you hear the "bit" (5)

Click "2. timing" when you hear the "tired." (6)

Click "2. timing" when you hear [silence time] (7)  
(It is shown "●" on page 89 waveform.)

Click "2. timing" when you hear the "I'll" (8)

Click "2. timing" when you hear the "take" (9)

Click "2. timing" when you hear the "a" (10)

Click "2. timing" when you hear the "break." (11)

Click a total of eleven times.

Try a few times until you think you've got it right.  
If you feel that the playback is too fast (too slow),  
adjust the playback speed by the pitch slider.



# Tutorial: 6. Edit lip-sync data

## (4) Acquisition of pronunciation timing "0Yawn"

When click on "3.",  
the value is displayed in the lip-sync data field.  
This is the lip sync data.

Lip-sync data field

Let's click the "voice data playback button" again.

The character should have lip-synced to the voice.

Watch the character's mouth move.

If you are OK, click the "save" button to save the file.

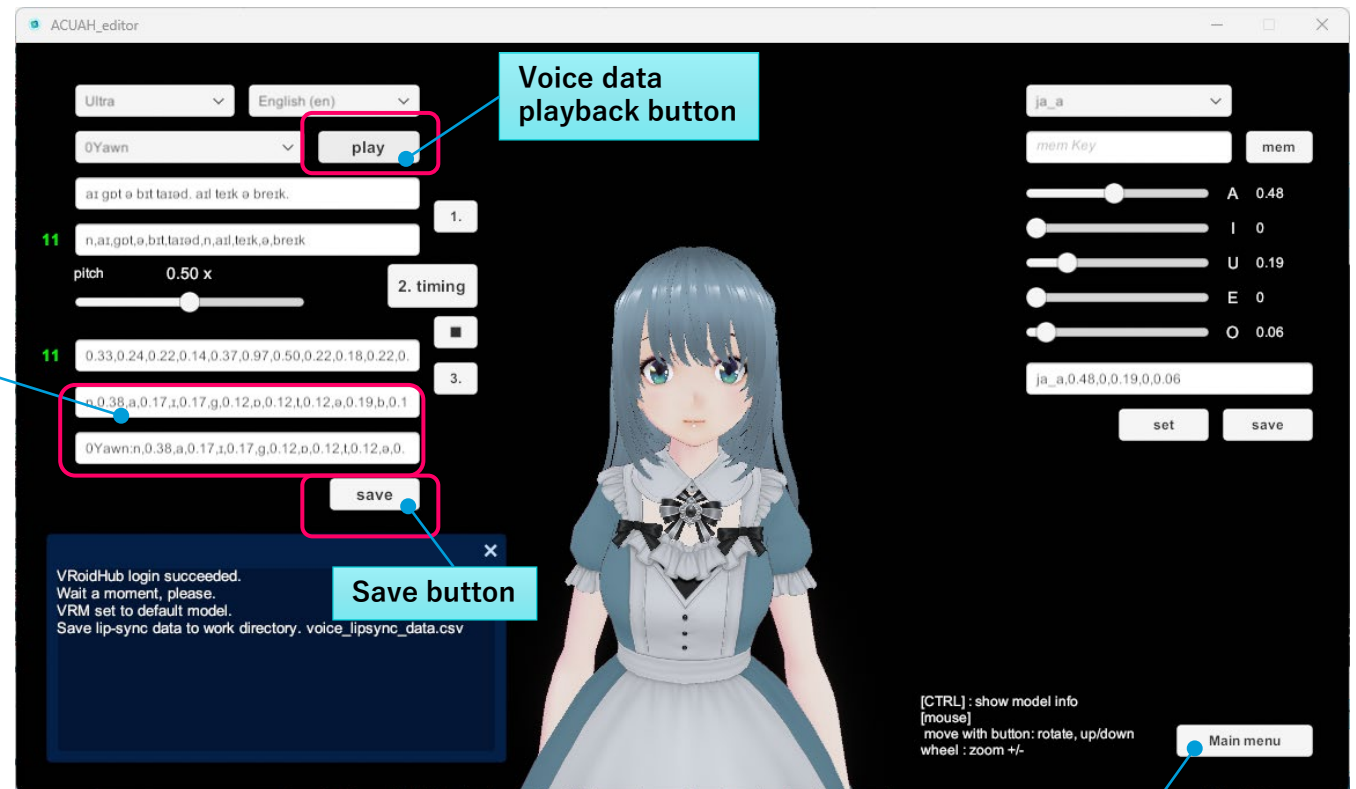
Save the lip sync data you have created.

(It will be written to "voice\_lipsync\_data.csv".)

The above is the lip-sync data for "Yawn" has been created.

This completes the editing of the lip sync data.

Click the "Main menu screen" button to return to the main menu.



# Tutorial: 6. Edit lip-sync data

This completes the editing of the lip sync data.

- In the tutorial, we created lip-sync data with two (English) voice data, but you can create original voice data by the same procedure.
- The timing of pronunciation and clicking should be roughly the same. Listen to the voice data several times and try to click in rhythm.

# Tutorial

## 7. Check the voice data and dialogue scenario files



# Tutorial: 7. Check the voice data and dialogue scenario files

## (1) Check the consistency of the voice data and the lip sync data.

### Lip-sync data language settings

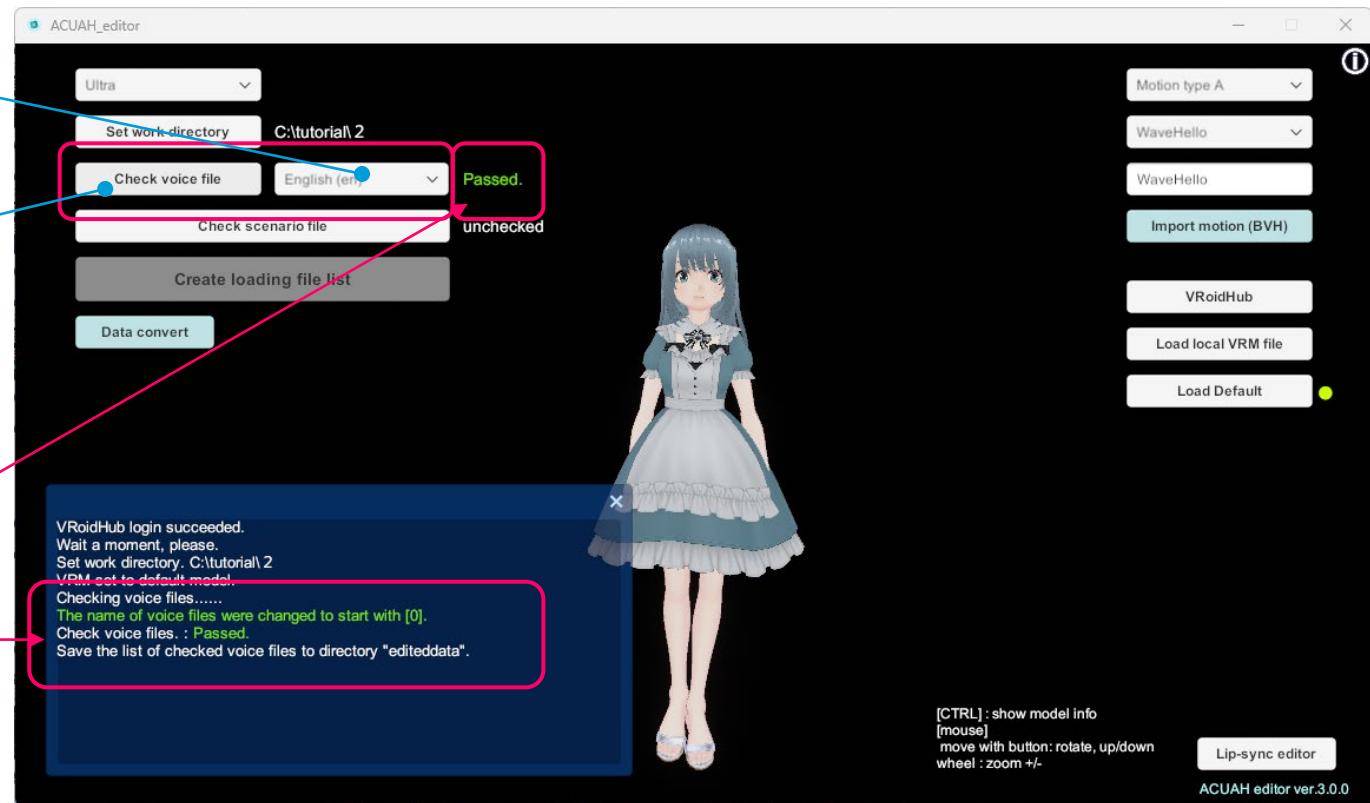
Make sure that the language setting is the same as in lip-sync editor screen.  
Please make sure that "English (en)" is selected.

### Voice data playback button

To check for consistency of the voice data and the lip sync data you created.

Click on "Check voice file."

"Passed." will be displayed.



# Tutorial: 7. Check the voice data and dialogue scenario files

## (2) Check the consistency of dialogue scenario files

### Check scenario file

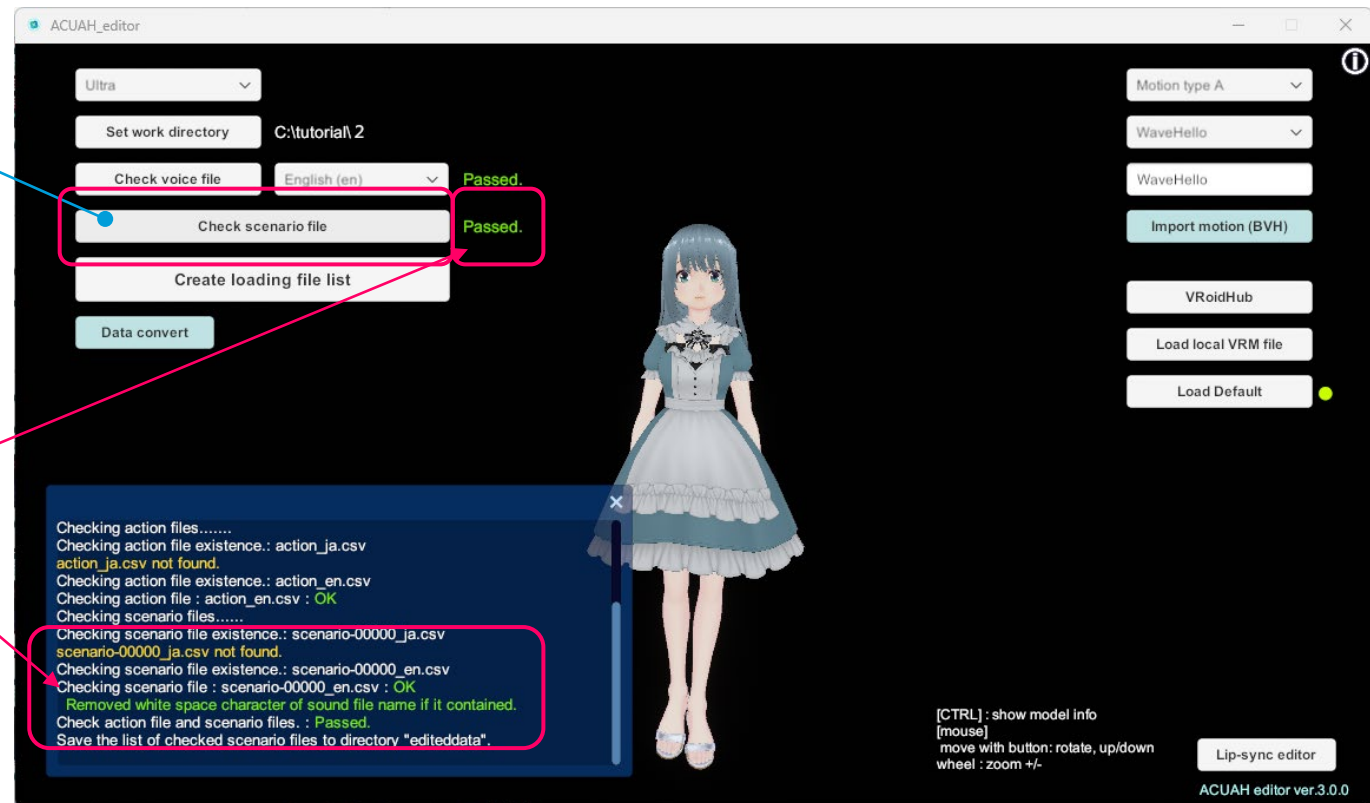
consistency check of voice data and dialogue scenario files.

When click here it is checked mainly about below.

- Whether there are all the files you needed.
  - Is the file format correct?
  - Are there any prohibited characters used?
  - Is it consistent with the voice data?
- and so on.

For this tutorial.  
In the dialogue scenario file  
"Passed." should appear on the screen.

If the dialogue scenario file is incomplete,  
an error message will be displayed.  
Please check and correct it.





# Tutorial: 7. Check the voice data and dialogue scenario files

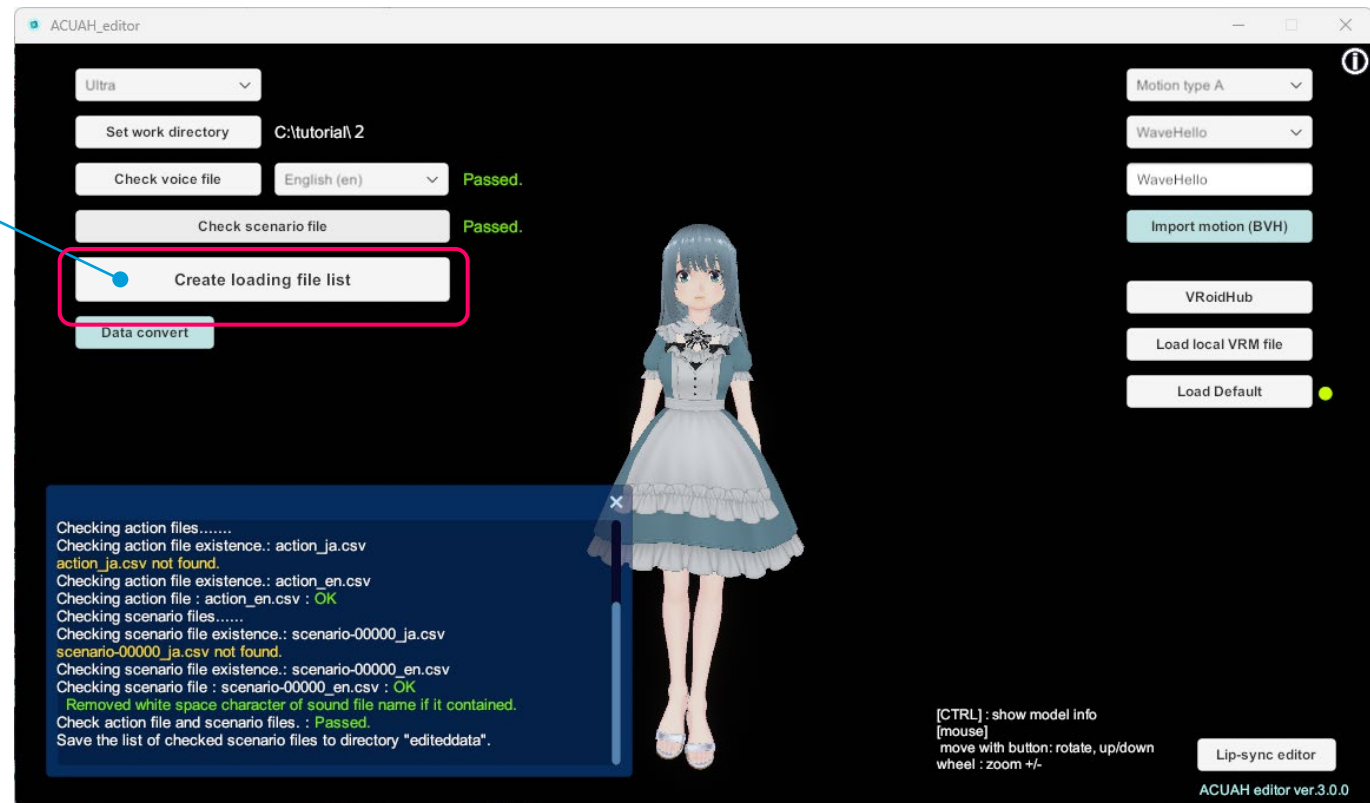
## (3) Check the character model.

**Check loading file list**  
Create editeddata folder  
for loading ACUAH file list

If "Check voice file" and "Check scenario file" are both passed, the "Create loading file list" button will be clickable.

You are able to proceed to create a "acuah\_local\_list" file.  
And you can also make a set of below files.

- Dialogue scenario file
- Voice data
- Character model



If you want to make a set with the character model, make sure you select the appropriate character model here.  
If it is a VRM file, please select the corresponding file on PC. (You can check the character model information by pressing the 「CTRL」 key.)

## Tutorial: 7. Check the voice data and dialogue scenario files

This completes the check of the voice data and dialogue scenario files.



# Tutorial

## 8. Create a local loading list file for ACUAH



# Tutorial: 8. Create a local loading list file for ACUAH

## (1) Confirm and agree to the Terms of Use.

If you click "Create loading file list.", a window on the right will appear.

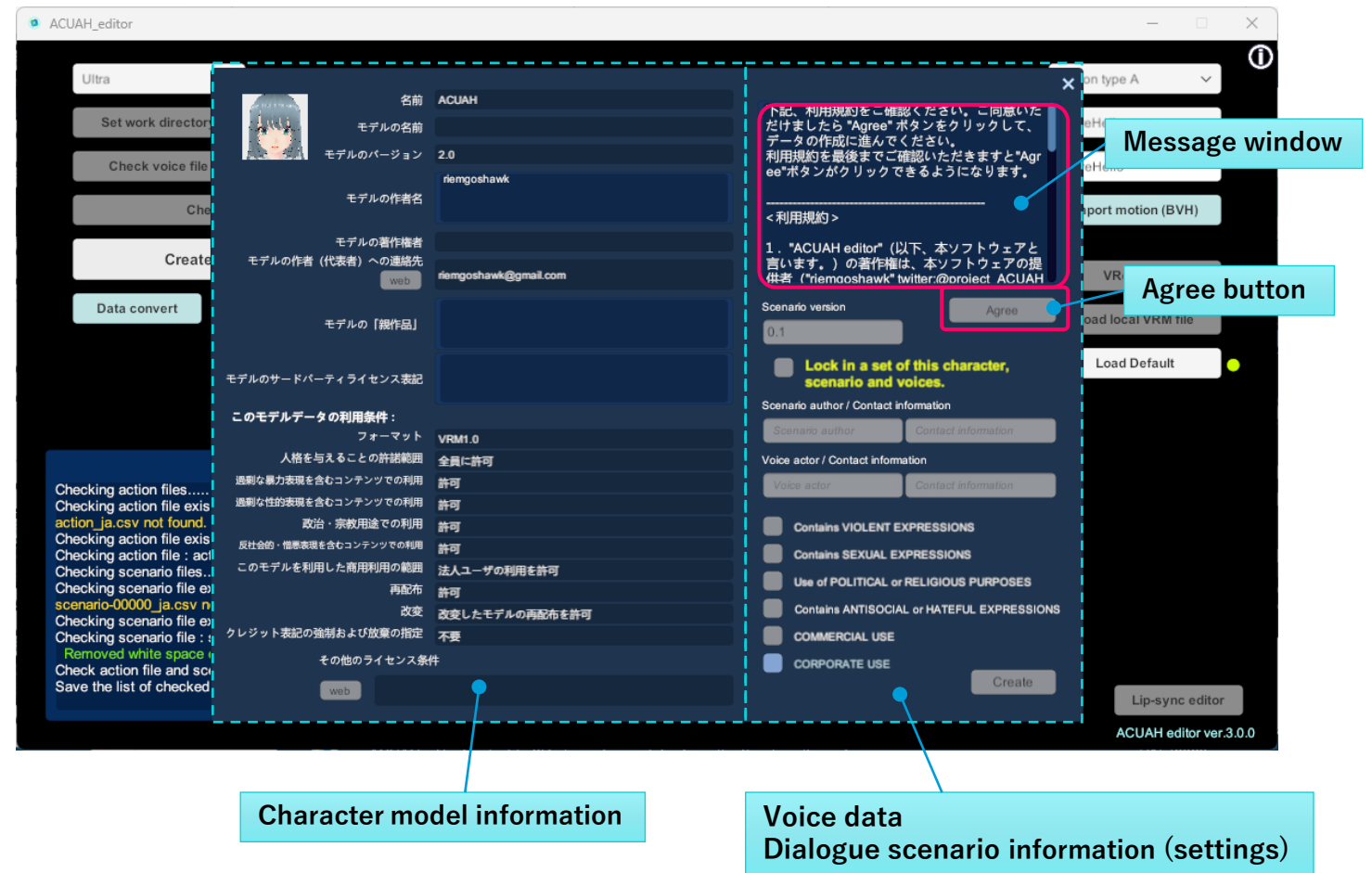
Character model information on the left and  
On the right is the voice data and  
dialogue scenario information (settings).

In the message window on the upper right shows  
Terms of Use. It is written by Japanese, but  
the contents are the same as on p.3 of this manual.

Please read the Terms of Use to the bottom  
of the page and you will be able to click on  
the "Agree" button.

**If you agree to the Terms of Use,  
please click on the "Agree" button.**

(The date and time when the you agree  
will be recorded.)



# Tutorial: 8. Create a local loading list file for ACUAH

## (2) Input the voice data and dialogue scenario information.

If you agree, you will be able to enter the following items.  
Please **enter the appropriate value**.

**In this tutorial, please enter your information.**

Scenario version (default 0.1)  
Version information of the scenario you created (number)  
(Any string)

Scenario author / Contact Information \*  
(Any string)

Voice actor / Contact Information \*  
"Ah-ya" / "twitter @aya\_voicer"

\* **Caution.**  
Contact information is optional,  
but you should enter if you use an original data,

Please comply with the intellectual property rights  
and terms of use of each data owner and enter appropriate values.  
( Note that this information is not collected by the app provider. )

The screenshot shows the ACUAH\_editor application window. A central dialog box is open for creating a new scenario. The dialog has a dark theme and contains the following fields and sections:

- Model Information:**
  - 名前 (Name): ACUAH
  - モデルの名前 (Model Name): [empty]
  - モデルのバージョン (Model Version): 2.0
  - モデルの作者名 (Model Author Name): riemgoshawk
  - モデルの著作権者 (Model Copyright Holder): [empty]
  - モデルの作者 (代表者) への連絡先 (Contact for Model Author (Representative)): web, riemgoshawk@gmail.com
  - モデルの「観作品」 (Model's "Observed Works"): [empty]
  - モデルのサードパーティライセンス表記 (Model's Third-Party License Statement): [empty]
- Terms and Conditions:**
  - このモデルデータの利用条件: VRM1.0
  - 人格を与えることの許諾範囲 (Scope of permission to give personality): 全員に許可 (Permitted to all)
  - 過剰な暴力表現を含むコンテンツでの利用 (Use in content containing excessive violence): 許可 (Permitted)
  - 過剰な性的表現を含むコンテンツでの利用 (Use in content containing excessive sexual expression): 許可 (Permitted)
  - 政治・宗教用途での利用 (Use for political/religious purposes): 許可 (Permitted)
  - 反社会的・憎悪表現を含むコンテンツでの利用 (Use in content containing antisocial/hate speech): 許可 (Permitted)
  - このモデルを利用した商用利用の範囲 (Scope of commercial use using this model): 法人ユーザーの利用を許可 (Permitted for corporate users)
  - 再配布 (Redistribution): 許可 (Permitted)
  - 変更 (Modification): 変更したモデルの再配布を許可 (Permitted to redistribute modified models)
  - クレジット表記の強制および放棄の指定 (Designation of mandatory credit and waiver): 不要 (Not required)
  - その他のライセンス条件 (Other license conditions): [empty]
- Agreement and Version:**
  - 7. 本ソフトウェアはユーザーに対して現状有姿で提供されるものであり、本ソフトウェア提供者は、本ソフトウェアにエラー、バグ等の不具合がないこと、ユーザーの利用目的に合致して有用であること、完全に動作することを含め、一切の保証は行わないものとします。
  - Please check the toggle properly which expressions the scenario and voice file contains.
  - Scenario version: 0.1
  - ☐ Lock in a set of this character, scenario and voices.
- Scenario author / Contact information:**
  - riemgoshawk, riemgoshawk@gmail.com
- Voice actor / Contact information:**
  - Ah-ya, twitter @aya\_voicer
- Consent checkboxes:**
  - ☐ Contains VIOLENT EXPRESSIONS
  - ☐ Contains SEXUAL EXPRESSIONS
  - ☐ Use of POLITICAL or RELIGIOUS PURPOSES
  - ☐ Contains ANTISOCIAL or HATEFUL EXPRESSIONS
  - ☐ COMMERCIAL USE
  - ☒ CORPORATE USE
- Buttons:** Agree, Create, Lip-sync editor

On the left side of the dialog, there is a log window showing the following messages:

- Checking action files.....
- Checking action file exists: action\_ja.csv not found.
- Checking action file exists: action\_ja.csv not found.
- Checking scenario files.....
- Checking scenario file exists: scenario-00000\_ja.csv not found.
- Checking scenario file exists: scenario-00000\_ja.csv not found.
- Checking scenario file exists: scenario-00000\_ja.csv not found.
- Checking scenario file exists: scenario-00000\_ja.csv not found.
- Removed white space from scenario file.
- Check action file and scenario file.
- Save the list of checked files.

# Tutorial: 8. Create a local loading list file for ACUAH

## (3) Set the usage conditions for the voice data and dialogue scenarios.

Please make the settings about the contents of the voice data and dialogue scenarios.

- dialogue scenarios
- voice data
- motion data

If they **contain violent expression**,  
**check the "Contains VIOLENT EXPRESSIONS."**

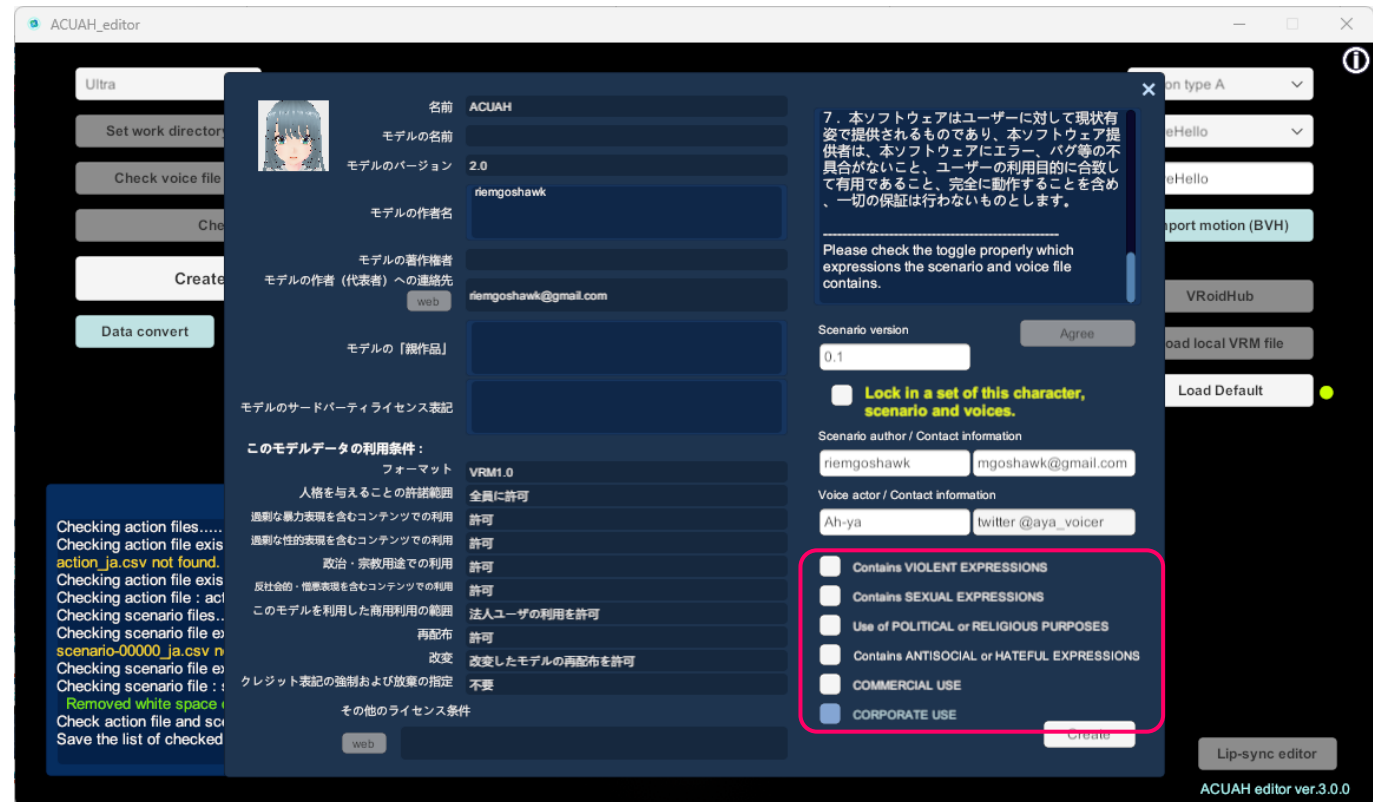
If they **contain sexually explicit material**,  
**check the "Contains SEXUAL EXPRESSIONS."**

If they are **used for political or religious purposes**,  
**check the "Use of POLITICAL or RELIGIOUS PURPOSES."**

If they **contain antisocial or hateful expressions**,  
**check the "Contains ANTISOCIAL or HATEFUL EXPRESSIONS."**

If they are used **for commercial**,  
**check the "COMMERCIAL USE."**

Please make appropriate settings.



"CORPORATE USE" cannot be checked manually.  
Only those who are using a corporate use license will have this checked automatically.

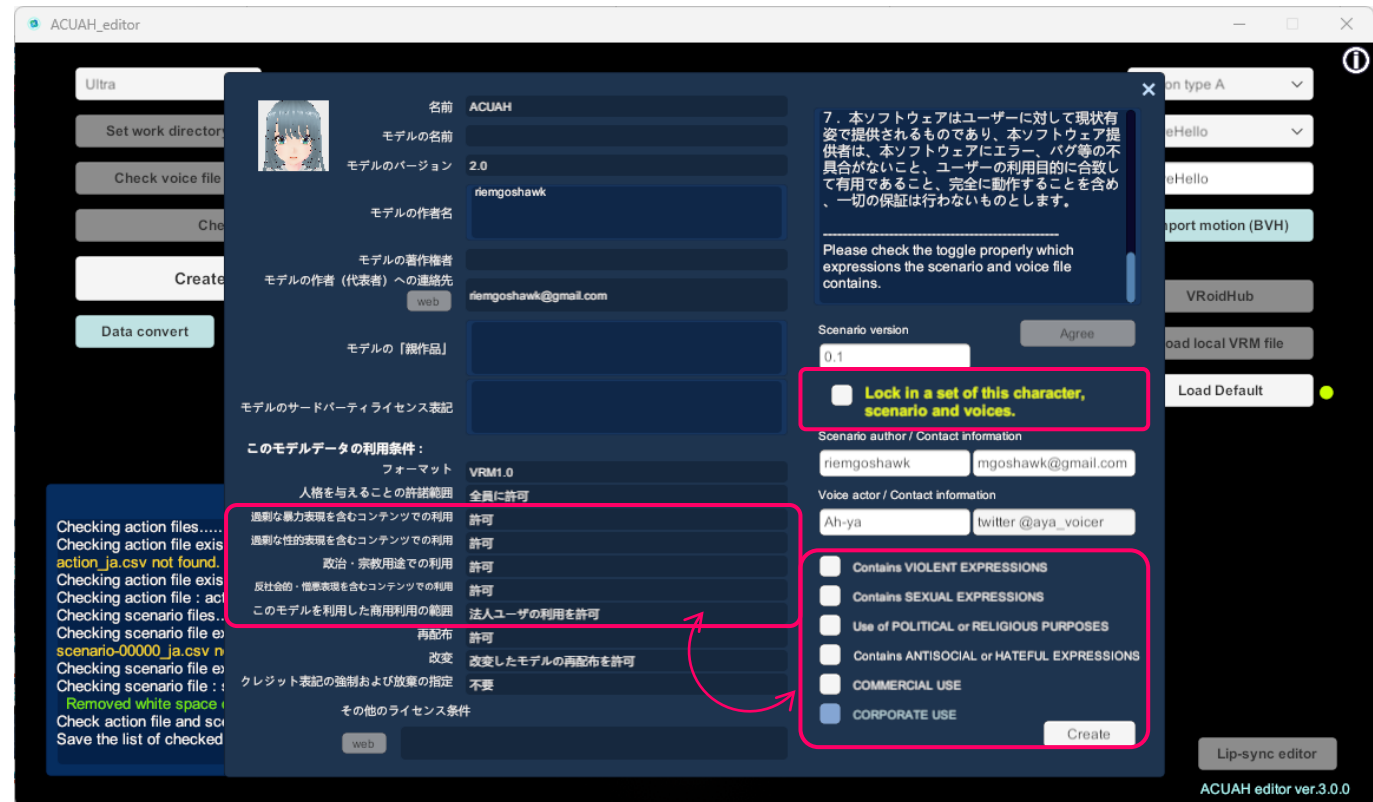
# Tutorial: 8. Create a local loading list file for ACUAH

## (3) Set the usage conditions for the voice data and dialogue scenarios.

**"Lock in a set of this character, scenarios and voices."**

If you check this box, the Voice data, dialogue scenarios, and character model as a set and they will be fixed.(locked-in)

**The User of ACUAH will not be able to select character model if it is locked-in.**



# Tutorial: 8. Create a local loading list file for ACUAH

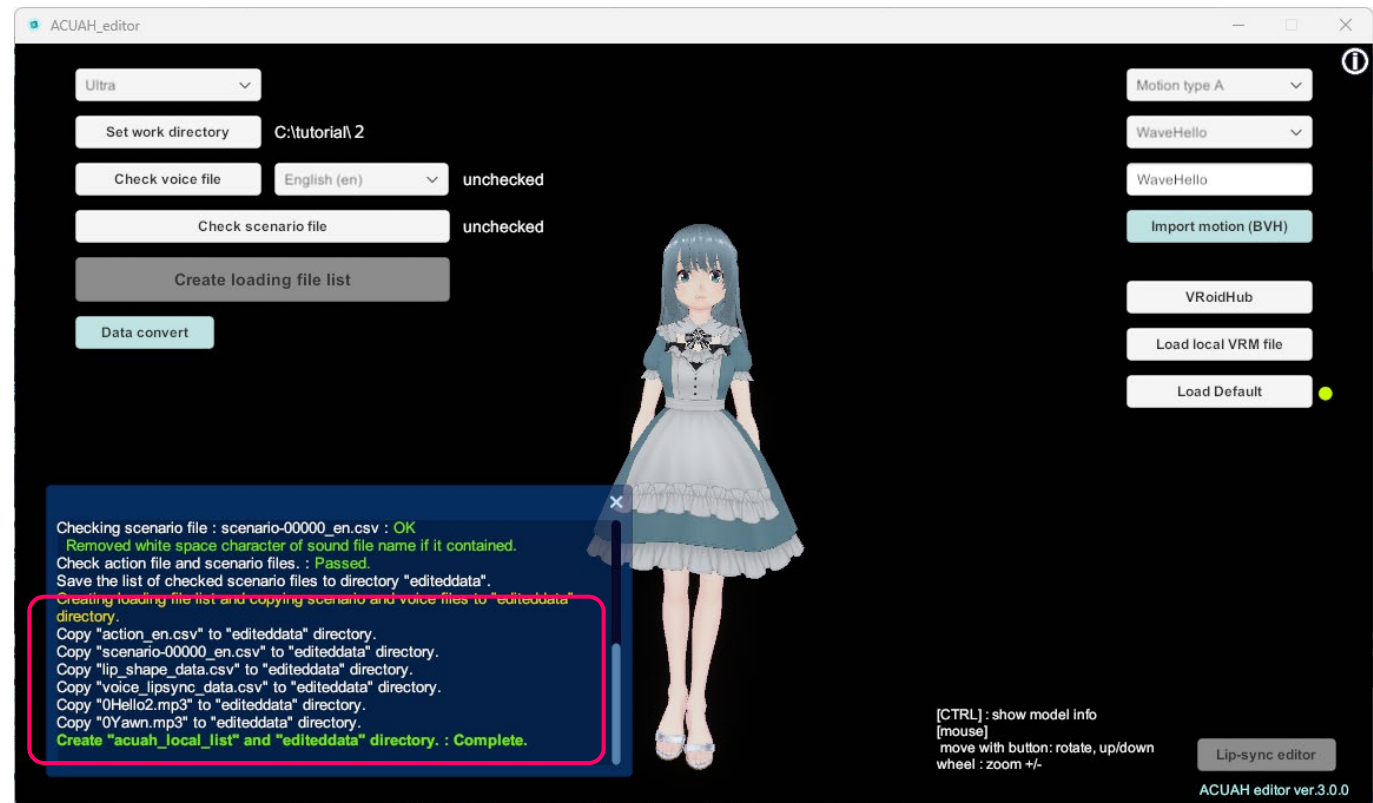
## (4) Create a local loading list for ACUAH.

Click the "Create" button at the bottom right.

If it is created successfully, In the message window

"Create "acuah\_local\_list" and "editeddata" directory. : Complete."

will be displayed.

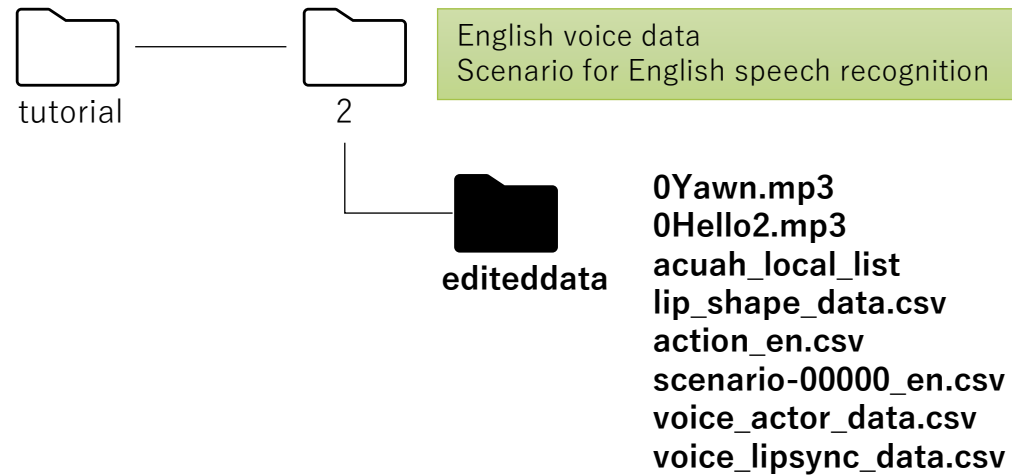




# Tutorial: 8. Create a local loading list file for ACUAH

## (5) Check the “editeddata” folder.

. ¥tutorial¥2 and look in the editeddata folder.  
You should now have the following files saved.



# Tutorial: 8. Create a local loading list file for ACUAH


## (6) Check the operation with ACUAH.

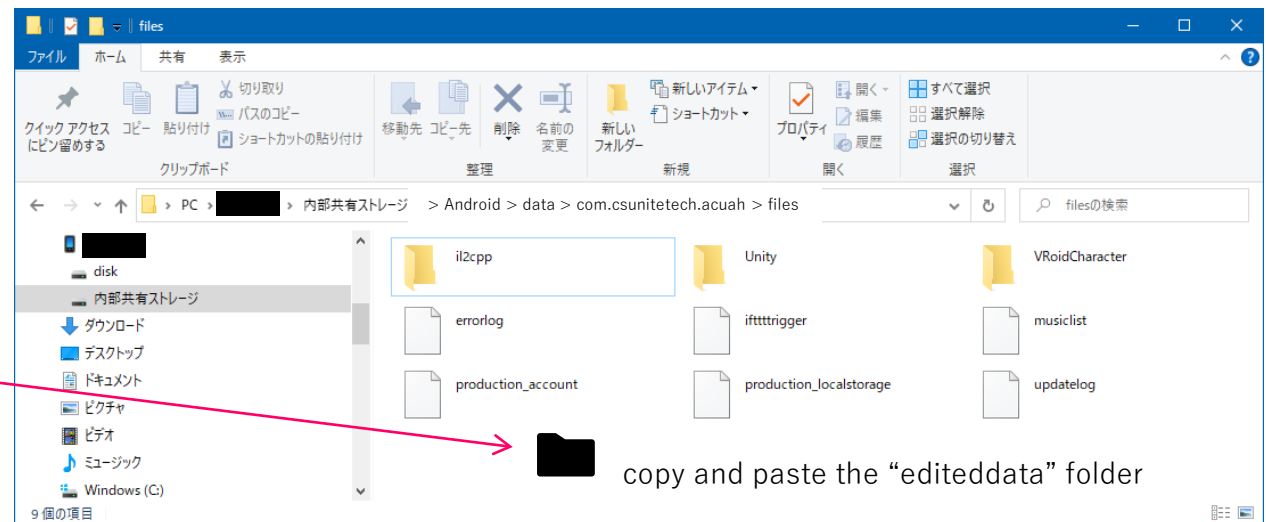
Connect your PC and smartphone with a USB cable or Bluetooth and save the whole editeddata folder.  
Copy the folder to the following directory on your smartphone.

**Internal Shared Storage: /Android/data/com.csunitetech.acuah/files**

If ACUAH can load them correctly, please **check the voice data and dialogue scenario**.

ACUAH operation manual are available at <https://acuah.info>.  
Please refer to the "ACUAH Application manual".

 **editeddata**  
0Yawn.mp3  
0Hello2.mp3  
acuah\_local\_list  
lip\_shape\_data.csv  
action\_en.csv  
scenario-00000\_en.csv  
voice\_actor\_data.csv  
voice\_lipsync\_data.csv



## Tutorial: 8. Create a local loading list file for ACUAH

We have finished creating the file list for loading ACUAH and checking the operation with ACUAH.  
This concludes the tutorial on how to use the ACUAH editor.

Please create your original voice data and dialogue scenario data.



# Additional Tutorial

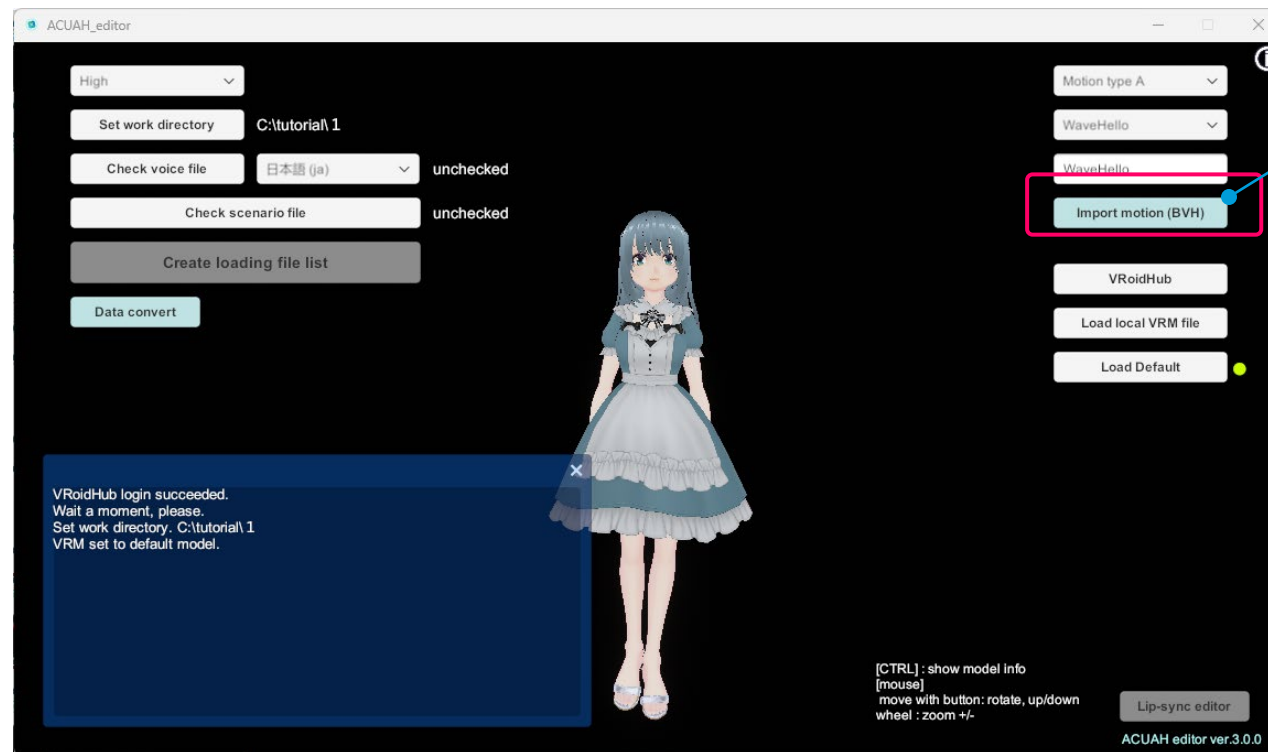
## 9. Import motion data (BVH)



# Additional Tutorial: 9. Import motion data (BVH)

## (1) Save motion data (BVH) in the working directory

- Motion data in BVH format can be imported along with dialogue scenarios and voice data.
- Save the motion data file in BVH format directly under the working directory.  
Multiple files can be imported at once, so please save all the target files.
- Once saved, click the "Import motion (BVH)" button.

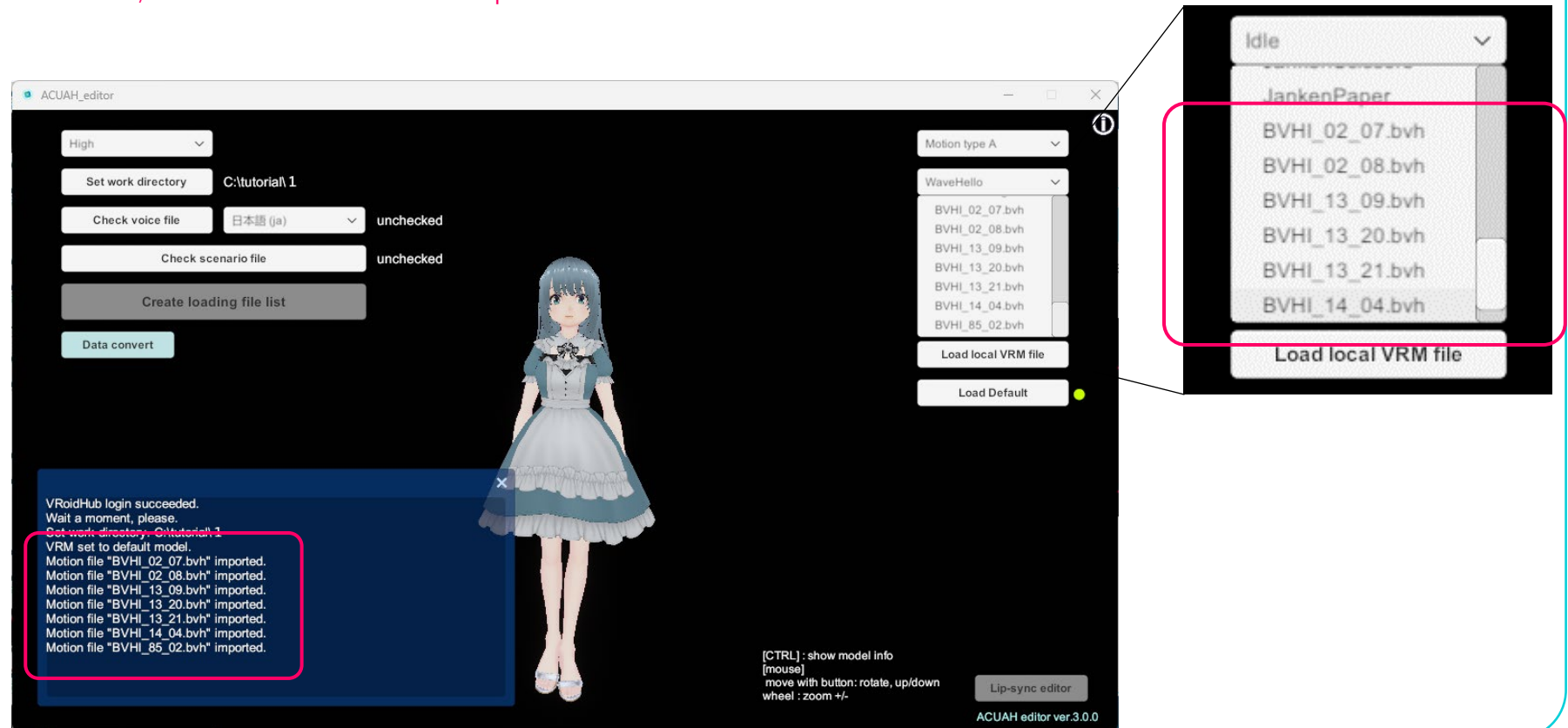


Import motion (BVH)  
button

# Additional Tutorial: 9. Import motion data (BVH)

## (2) Import motion data (BVH)

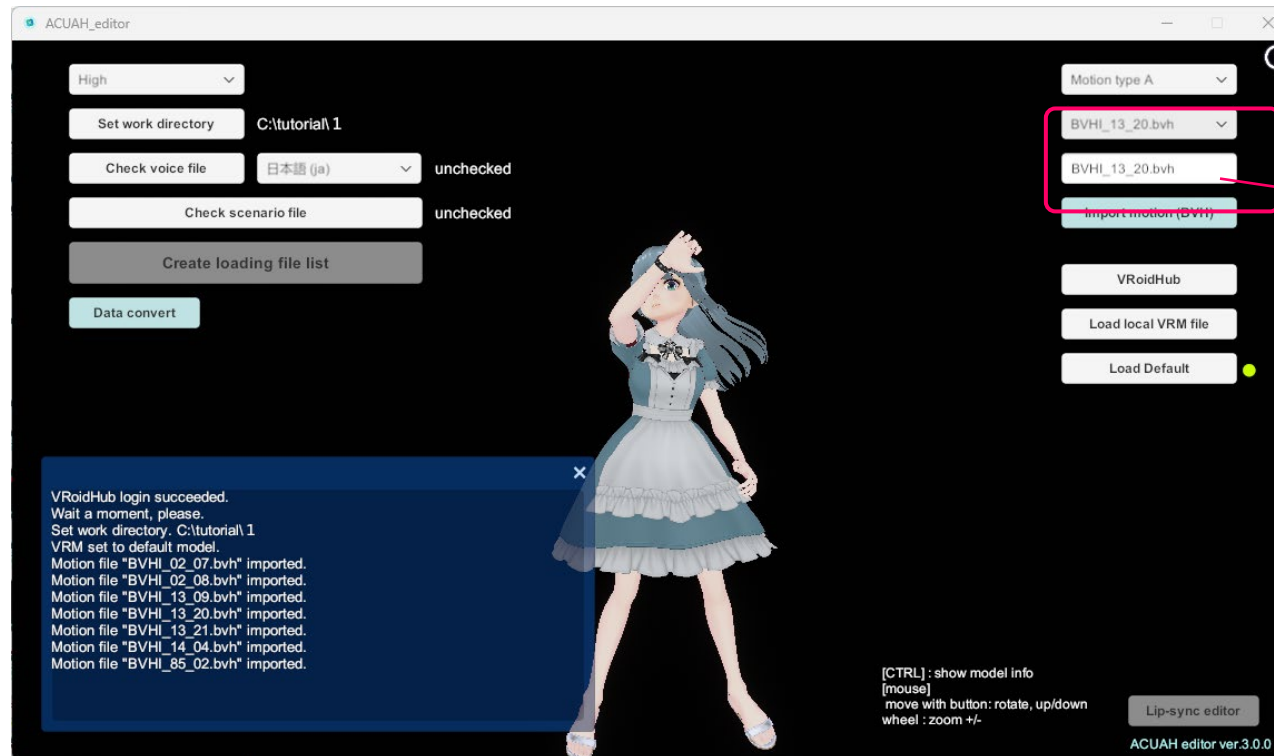
- After successful import, the message window will display "Motion file "BVHI\_xxx.bvh" imported.
- When importing a file, the string "BVHI\_" is automatically added to the beginning of the file name.
- In the drop-down list for motion, the file name of the imported motion data is added.



# Additional Tutorial: 9. Import motion data (BVH)

## (3) Confirm motion data (BVH) and specify it in the dialogue scenario

- Select the imported motion data from the motion drop-down list to confirm the character's movement.
  - The file name appears in the motion name string copy field.
- Please enter **this string** in **column 5 [animation trigger]** of the **dialogue scenario file**.
- This string (filename), like the other [animation trigger] specifications, **corresponds to the enclosing characters**.



motion name  
string copy

5	6
[animation trigger]	[o
BVHI_13_20.bvh	
WaveHello	

**When specifying with enclosing characters**

**%BVHI\_13\_20.bvh:WaveHello%.**

"BVHI\_13\_20.bvh", "WaveHello".  
is selected at random.

# About ACUAH editor / ACUAH

Please visit the following website for more information.

ACUAH website	<a href="https://acuah.info">https://acuah.info</a>
BOOTH (riemgoshawk)	<a href="https://riemgoshawk.booth.pm/">https://riemgoshawk.booth.pm/</a>

If you have any questions, such as how to use the software or bugs, please contact us at the email address below.  
[riemgoshawk@gmail.com](mailto:riemgoshawk@gmail.com)





# Thanks

I would like to express our gratitude to the following people for agreeing to distribute the voice data for this tutorial.  
Thank you once again.

Nana Shiina (椎奈なな 様 twitter @C7na7)

Tsukiyashiro Rei (月社零 様 twitter @keiiiiin\_orz)

Ah-ya@ (twitter @aya\_voicer)