**Multilingual Support in ChattyInfty3, the SAPI5 Version**

– How to Translate it into Another Language Version –

(November 1st, 2018)

This document is brief instructions for end users on how to customize ChattyInfty3 so that it can treat their local language other than English and Japanese.

**0. Basic Restrictions**

In the latest version (ChattyInfty3, Ver.3.21), please be sure there are some restrictions on multilingual support as shown below:

- ChattyInfty3 can be customized for many languages that are represented in Unicode including some languages using Grapheme Cluster such as Hindi and Tamil (they represent a single character with multiple codes). However, languages written from right to left such as Arabic are out of scope. For the present, we do not have a plan to correspond to them.

- ChattyInfty3, AITalk version is just Japanese-local, and you cannot customize it for the other languages. For the localization purpose, its SAPI5 version should be used.

**1. First Step of Localization**

1.1 Installing ChattyInfty3

To localize ChattyInfty3, please obtain its latest version from our web site:

http://www.sciaccess.net/en/

and install it. At the very beginning of the installation, you will be asked “which language should be used in the installation?” You may choose any language listed there, but in this instructions, we assume that you choose “English.”

ChattyInfty3, the SAPI5 version uses Microsoft Speech API, Ver.5 (SAPI5) as a text-to-speech (TTS) engine. You must install it for the local language together with ChattyInfty3. If it were not available, ChattyInfty3 could not read out contents. In such a case, however, if voices of “Microsoft Speech Platform” and “Windows10 OneCore” were available, you could use them as SAPI5 voices. Concerning this point, please read “SpeechFiveMagic.docx” included in this folder or see the web site: http://www.digitalnauts.co.jp/sfm/?lang=en.

1.2 Creating A New Language Folder

To localize ChattyInfty3 for another language, a new language folder should be prepared in the following manner.

1. Open the folder in which the executable file of ChattyInfty3 is located. It can be done by right-clicking (or pressing the Application key at) ChattyInfty3 icon on the desktop and clicking “Open File Location.” Incidentally, the full path to the folder is as follows:

C:\Program Files(x86)\sAccessNet\ChattyInfty3\.

1. Open the “lang” folder and in it, create a new folder named the code of the language that is newly added. The following table gives a list of the language code that Microsoft defines; Please be sure this table includes languages that cannot be treated in ChattyInfty3 for the present.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Code | Language | Code | Language | Code | Language | Code | Language |
| Af | Afrikaans | Fo | Faeroese | Lt | Lithuanian | Sk | Slovakian |
| Ar | Arabic | Fr | French | Lv | Latvian | Sl | Slovenian |
| As | Assamese | Gu | Gujerati | Mk | Macedonian | Sq | Albanian |
| Az | Azerbaijani | He | Hebrew | Ml | Malayalam | Sr | Croatian |
| Bg | Bulgarian | Hi | Hindi | Mr | Marathi | Sv | Swedish |
| Bn | Bangla | Hr | Croatian | Ms | Malayan | Sw | Swahili |
| Ca | Catalan | Hu | Hungarian | Ne | Nepali | Ta | Tamil |
| Cs | Czech | Hy | Armenian | Nl | Dutch | Te | Telugu |
| Da | Danish | Id | Indonesian | No | Norwegian | Th | Thai |
| De | German | Is | Icelandic | Or | Odia | Tr | Turkish |
| El | Greek | It | Italian | Pa | Punjabi | Tt | Tatar |
| En | English | Ja | Japanese | Pl | Polish | uk | Ukrainian |
| Es | Spanish | Ka | Georgian | Pt | Portuguese | ur | Urdu |
| Et | Estonian | Kk | Kazak | Ro | Romanian | uz | Uzbek |
| Eu | Basque | Kn | Kannada | Ru | Russian | Vi | Vietnamese |
| Fa | Persian | Ko | Korean | Sa | Sanskritic | Zh | Chinese |
| Fi | Finnish | Ks | Kashmiri | Sd | Sindhi |  |  |

1.3 Preparing Necessary Files

Next, copy and paste the following two (2) files and a folder (three in total), which are all included in the lang folder, into the new folder.

• resource.xml (file)

• CodeMenu.tbl (file)

• ReadSettings (folder)

Put a national-flag image also in the new folder. It is used as an icon in a dialog to select a language. Its format should be either ICO, PNG or BMP, and its name should be flag.\* such as flag.ico, flag.png, etc. There is no restriction concerning its file size, but we recommend you to prepare one of around 200pixel. An easy way to do it may be to get a national-flag image shown in an article in Wikipedia or somewhere and rename it to flag.png. If you would not need the special icon for the language, or if it were difficult to get an appropriate image, you might use the same image: flag.ico as one for English or Japanese; you can find it in the English (en)/Japanese (ja) folder. You might copy and paste it into the new folder.

**2. Second Step of Localization**

Next, you do need to author/translate the three definition files: “ReadSetting\_SAPI5\_EN.txt” in the ReadSettings folder, CodeMenu.tbl and resource.xml.

2.1 Authoring ReadSetting.txt

ReadSetting\_SAPI5\_EN.txt is a definition table of aloud reading for characters/symbols/ formulas in the math mode (in English). ChattyInfty3 is to read out math symbols/formulas in a manner defined in this text file. Please author/translate it in the following way.

Remark:

- ReadSetting\_SAPI5.txt should be encoded in Unicode, not UTF-8.

2.1.1 Before Translating

Before starting the translation, change the file name to an appropriate one for the new language such as ReadSetting\_SAPI5\_VI.txt for Vietnamese.

Open ReadSetting\_SAPI5\_XX.txt with a text editor. The first sector is brief instructions on how to edit this file. In this sector, all lines start by “;” mark. ;-mark is a command for comment-out, and all descriptions after it are disregarded by the program.

2.1.2 [Settings] Sector

The next sector: [Settings], shows the imprint of this definition file. This information will be displayed on the dialog to choose a language. For instance, in Vietnamese, the description might be:

[Settings]

Version=1.6

Description=Vietnamese(SAPI5)

Language=Vietnamese

Date=2017.12.17

\* Version and Date should be given appropriate values to allow you to manage this file.

2.1.3 [MathReading] Sector

The [MathReading] defines how to read out math formulas.

In ChattyInfty3, three different types of “aloud reading” for math formulas can be selected. “Plain-Reading mode” is based on one which may be most widely used in English-speaking countries. It is natural; however, a spoken math expression is often ambiguous. We assume that people with low vision and dyslexia use it. In “Smooth-Reading mode,” minimum-necessary speech guides for blind users to grasp correctly the structure of a math formula are added. “Detailed-Reading mode” is assumed to be used when a blind user wants to know the math-formula structure in the most detail.

For instance, the following is the definition of aloud reading for a fraction in English.

[MathReading:frac]

{ // Standard

plain=“ ,%OVER, over, %UNDER, “

detail=“ ,fraction, numerator, %OVER, over, denominator, %UNDER frac end, “

smooth=“ ,frac, %OVER, over, %UNDER, “

}

\* As is explained in the first comment sector, %UNDER and %OVER are arguments at the numerator and denominator positions, respectively.

2.1.4 [ReadStrings] Sector

The [ReadStrings] sector gives a speech comment when a certain action occurs. For instance,

InputModeText = “Changed to text input mode.”

Defines a speech comment when changing to the text-input mode by pressing Ctrl + Space key.

EndDenominator =“ end of denominator, “

Gives a speech comment when the cursor moves to the end of numerator position by pressing the cursor (arrow) key.

2.1.5 [MathReadingCharacters] Sector

The [MathReadingCharacters] Sector defines how to read out characters/symbols. Again, you can assign three types of aloud reading to each character/symbol.

Each line in this sector consists of five fields enclosed by “-marks. For instance,

“Lambda”,,”cap lambda, “,”capital lambda- “,”cap lambda, “

The first field is its ID, and the second shows an attribute if necessary. Don’t change these two fields. The third, fourth and fifth fields give the plain, detailed and smooth readings, respectively. You can change them as you like. Punctuation marks included in these fields are put to control Pauses and intonation in aloud reading.

2.2 Authoring CodeMenu.tbl

CodeMenu.tbl defines the structure of symbol/character and formula lists which appear in the context menu. If you would need to change the names of <group> and <item> in the lists to local ones, you could do that.

To change the name, give an attribute: “caption” to it (<group> or <item>) and assign a name shown in the following example.

Original:

<group name=“Character Symbol”>

<group name=“Greek”>

<item code=“Gamma”/>

Changed:

<group name=“Character Symbol” caption=“Math Symbol”>

<group name=“Greek” caption=“Greek Character”>

<item code=“Gamma” caption=“Capital Gamma”/>

Then, the Greek letter: “Capital gamma” appears in the context menu in the following structure.

Math Symbols > Greek Characters > Capital Gamma,

While in the original,

Character Symbol > Greek > Gamma.

Remark:

CodeMenu.tbl is an xml file, and in it, “&”, “>“, etc. are regarded as a control command. If you need to write those characters, please use alternative expressions such as &amp;, &lt; which are standard in xml. Instead, it might be convenient for you to use an xml editor for authoring CodeMenu.tbl.

2.3 Authoring resource.xml

resource.xml gives labels for menu items and dialogs. If the English menu and dialogs were all right, you might not need to modify this file.

There are several sectors in this xml file, but in localization purpose, just the following five (5) sectors in <resource> could be modified.

<captions>, <descriptions>, <strings>, <messages> and <dialogs>.

Remark:

- The other sectors: <menu>, <popup>, <toolbar>, <accelerator> and <colors> are important elements for the software, itself. Do not modify them!

-Resource.xml is an xml file, and in it, “&”, “>“, etc. are regarded as a control command. If you need to write those characters, please use alternative expressions such as &amp;, &lt; which are standard in xml. Instead, it might be convenient for you to use an xml editor for authoring it.

2.3.1 <captions> Sector

Elements: “caption” in the <captions> sector are used to gives a label name to a menu item such as:

<caption id=“AppExit”>&amp;Quit</caption>

In the above descriptions, a character: Q following &amp; (&-mark) is regarded as a shortcut for the menu item: “Quit”.

2.3.2 <descriptions> Sector

Elements: “description” in the <descriptions> Sector are texts to be displayed for each command on the status bar.

2.3.3 <strings> and <messages> Sectors

Elements in <strings> and <messages> are character sequences or messages, which are displayed in the software, itself or a message box.

2.3.4 <dialogs> Sector

Elements: dialog in <dialogs> sector give label names of dialog boxes and a caption text for each button in the dialog boxes.

**3. How to Change a Language in ChattyInfty3**

To change a language actually in ChattyInfty3, after completing the job described above, you should change three (3) items in the Setting menu.

- Choose the (local) language at "LANG" in the Setting menu. At that time, after pressing the OK button, rebooting the software should be required.

- Select a local TTS voice at “Speech Setting“ in the menu. If you could not find it in the list, please see "SpeechFiveMagic.docx" included in this folder.

- Finally, to change ReadSetting\_SAPI5.txt to the local-language version, open "Select Reading Table" in the menu and choose an appropriate one.

Remark:

- In addition to the above points, please check the font setting at the Option tag in the Setting menu. If a font other than Times New Roman were chosen, change it to Times New Roman. If not, some local language might not be displayed properly.