Speech Translation SDK

Overview

Version 1.0

Table of Contents

[1. About This Document 3](#_Toc374435477)

[1.1. Overview 3](#_Toc374435478)

[2. Overview of the MCML System 4](#_Toc374435479)

[2.1. Overview of the MCML system 4](#_Toc374435480)

[2.2. Communication functions between the Communication Servers and the Engines 5](#_Toc374435481)

[3. Overview of the MCML Servers 6](#_Toc374435482)

[3.1. ASR Communication Server 6](#_Toc374435483)

[3.2. MT Communication Server 7](#_Toc374435484)

[3.3. TTS Communication Server 8](#_Toc374435485)

[4. Details of Communication Methods 9](#_Toc374435486)

[4.1. Communication between the ASR Communication Server and the ASR Engine 9](#_Toc374435487)

[4.2. Communication between the MT Communication Server and the MT Engine 11](#_Toc374435488)

[4.3. Communication between the TTS Communication Server and the TTS Engine 13](#_Toc374435489)

[A. Appendix 15](#_Toc374435490)

[A.1. Selection of servers connected by the Control Servers by requests and languages 15](#_Toc374435491)

[A.2. Pivot Translation by the MCML system 16](#_Toc374435492)

[A.3. Communication format 17](#_Toc374435493)

[A.4. Communication protocol and data format 23](#_Toc374435494)

Update History

|  |  |  |
| --- | --- | --- |
| Version | Release Date | Features |
| 1.0 | Dec. 20, 2013 | Released |

# About This Document

## Overview

This document is a guide to how to customize the MCML Servers, which constitute the server system of Speech Translation SDK (Software Development Kit), to connect to the Automatic Speech Recognition (ASR), Machine Translation (MT) and Text-to-Speech (TTS) Engines. It describes various running modes on each server in the MCML system.

This document includes the following topics:

* Communication functions in the MCML system
* Overview of the MCML servers
* Details of the communication methods
* Customization of the MCML servers to connect to the ASR Engine, the MT Engine and the TTS Engine
* Details of the MCML format

# Overview of the MCML System

## Overview of the MCML system

The MCML system consists of the Control Servers and three Communication Servers, i.e., ASR, MT and TTS Communication Servers (Figure 2‑1: MCML system overview). Process requests from clients are sent to the Communication Servers via the Control Servers and, as a response, results of processing are returned to clients..

Client

http (https)

TCP/IP

MT

Engine

ASR

Engine

Control Server (Primary)

Control Server (Secondary)

TTS

Communication

Server

TCP/IP

TCP/IP

MT

Communication

Server

ASR

Communication

Server

TTS

Engine

図2‑1：MCML system overview

* + 1. Control Server (Primary)
* The Control Server (Primary) transmits and receives process requests/responses between clients and the Control Servers (Secondary). According to processes (ASR, MT, or TTS) and languages, you may select a specific Control Server (Secondary) to connect. Control Servers (Secondary) in other MCML systems are also to be used.
  + 1. Control Server (Secondary)
* The Control Server (Secondary) transmits and receives process requests/responses between the Control Server (Primary) and the Communication Servers.
  + 1. ASR Communication Server
* The ASR Communication Server transmits and receives process requests/responses between the ASR Engine and the Control Server (Secondary).
  + 1. MT Communication Server
* The MT Communication Server transmits and receives process requests/responses between the MT Engine and the Control Server (Secondary).
  + 1. TTS Server
* The TTS Communication Server transmits and receives process requests/responses between the TTS Engine and the Control Server (Secondary).

## Communication functions between the Communication Servers and the Engines

Figure 2‑2 shows the communication between the servers and engines in the MCML system.

MT

Communication

Server

ASR

Communication

Server

TTS

Communication

Server

Communication

Function

MT

Engine

Communication

Function

Communication

Function

Command

TCP/IP

TCP/IP

TTS

Engine

ASR

Engine

Figure 2‑2: Communication between MCML servers and engines

# Overview of the MCML Servers

## ASR Communication Server

The ASR Communication Server has the following functions:

* To receive the sound of speech sent from the Control Server.
* To select ASR Engines used to recognize the language of the speech.
* To distribute a recognition process among different ASR Engines automatically by load balancing.
* To receive the recognized text from the ASR Engines.
* To send the recognized text to the Control Server in the MCML format.

C:\Users\knozaki\Pictures\arrow45-001.gif

ASR

Communication

Server

ASR Engine 3

MCML

Control

Server

(Secondary)

ASR Engine 2

MCML

ASR Engine 1

C:\Users\knozaki\Pictures\arrow45-001.gif

MCML

Queue

Figure 3‑1: ASR Communication Server overview

## MT Communication Server

The MT Communication Server has the following functions:

* To receive text of the original speech sent from the Control Server in the MCML format.
* To select MT Engines used in translation into a requested language.
* To distribute a translation process among different MT Engines automatically by load balancing.
* To receive the translated text from the MT Engines.
* To send the translated text to the Control Server in the MCML format.

C:\Users\knozaki\Pictures\arrow45-001.gif

MT

Communication

Server

MT Engine 3

MCML

Control

Server

(Secondary)

MT Engine 1

MT Engine 2

MCML

C:\Users\knozaki\Pictures\arrow45-001.gif

MCML

Queue

Figure 3‑2: MT Communication Server overview

## TTS Communication Server

The TTS Communication Server has the following functions:

* To receive text for speech synthesis sent from the Control Server in the MCML format .
* To select TTS Engines used to synthesize speech in a requested language.
* To distribute a synthesizing process among different TTS Engines automatically by load balancing.
* To receive the synthesized speech from the TTS Engines.
* To send the synthesized speech to the Control Server in the MCML format.

C:\Users\knozaki\Pictures\arrow45-001.gif

TTS

Communication

Server

TTS Engine 3

MCML

Control

Server

(Secondary)

TTS Engine 1

TTS Engine 2

MCML

C:\Users\knozaki\Pictures\arrow45-001.gif

MCML

Queue

Figure 3‑3: TTS Communication Server overview

# Details of Communication Methods

## Communication between the ASR Communication Server and the ASR Engine

The flow of communication between the ASR Communication Server and the ASR Engine is shown in Figure 4-1. The process is run by the function “processRequest()” described in “ASREngineCtrl.java”.

Start

Get Input Data

Change the language mode.

Get an input profile type of the request.

Get input data from MCML request data.

Get audio metadata.

Get Audio Type

Input User Profile

Change Language Mode

**Customizable**

**Get binary data from the client and send it to the ASR Engine.**

Process

Speech Data

Process N-Best

**Parse N-Best.**

**Create output data in the MCML format.**

Output MCML

End

Figure 4‑1: Communication flow in the recognition process

* + 1. Communication method

The ASR Communication Server and the ASR Engine communicate using TCP/IP. Separate TCP/IP ports are used for transmission of speech data to the Engine and for reception of recognition results from the Engine.

Process

Speech Data

FrameSync

ASR Engine

ASR Communication Server

TCP/IP

N-Best

Process N-Best

Figure ‑: Data flow in the recognition process

* + 1. Communication data
       - Audio data of speech is sent to the ASR Engine using the FrameSync format (Refer to A.4.3: FrameSync data).
         * Audio data format

|  |  |
| --- | --- |
| Audio format | Linear PCM |
| Sampling frequency | 16 kHz |
| Sampling bit rate | 16 bit |
| Endian | Big |

* Results of speech recognition are received in the N-Best text format.

## Communication between the MT Communication Server and the MT Engine

The flow of communication between the MT Communication Server and the MT Engine is shown in Figure 4-3. The process is run by the function “processRequest()” described in MTEngineCtrl.java.

Start

Get Input data

Set the translation direction.

Get language information.

Get input data from MCML request data.

**Parse the N-Best.**

**Receive translation results from the Engine.**

**Send the text to the Engine.**

Get Translation Target Language

Set Translation Direction

Get Source Text

for Translate

Get the source text for translation.

**Customizable**

Filter

**Filter the source text.**

Send Text

Receive Result

Parse N-Best

**Create output data in the MCML format.**

Output MCML

End

Figure 4‑3: Communication flow in the translation process

* + 1. Communication method

The MT Communication Server initiates the MT Engine by using an external command upon its startup, sends text for translation to the standard input of the Engine, and receives translated results from the standard output of the Engine.

MT Communication Server

Send Text

MT Engine

Command

Receive Result

Standard out

Figure 4‑4: Data flow in the translation process

## Communication between the TTS Communication Server and the TTS Engine

The flow of communication between the TTS Communication Server and the TTS Engine is shown in Figure 4-5. The process is run by the function “processRequest()” described in TTSEngineCtrl.java.

Start

Get Input data

Get gender and age of the speaker.

Get text for speech synthesis from the request.

Get input data from MCML request data.

**Create output data in the MCML format.**

**Receive the synthesized data from the TTS Engine.**

**Send the SSML data to the TTS Engine.**

**Create data in the SSML format for the TTS Engine.**

Adjust delimiters in the text.

Get Text

Information

Get Voice

Information

Adjust Delimiter

**Customizable**

Create SSML

Send Data

Receive Data

Output MCML

End

Figure 4‑5: Communication flow in the speech synthesis process

* + 1. Communication method

The TTS Communication Server and the TTS Engine communicate using TCP/IP. Requests of speech synthesis are sent to the TTS Engine in the SSML (W3C) format, and synthesized data are returned from the Engine.

Send Data

TTS Engine

TTS Communication Server

SSML

TCP/IP

SSML

Receive Data

Figure 4‑6: Data flow in the speech synthesis process

# Appendix

## Selection of servers connected by the Control Servers by requests and languages

You may select servers to connect from the Control Servers according to processes (ASR, MT, or TTS) and languages. Different combinations of settings will make it possible to perform speech translation by utilizing multiple MCML systems. Figure A-1 represents translation from Japanese to Chinese, which is intermediated by Japanese-to-English and English-to-Chinese translations. The translation processes for Japanese and English are achieved by separate servers, as shown by "Japanese system" and "English system" in Figure A-1, respectively. The processes include:

* Recognition of speech in Japanese by the ASR servers in the Japanese system (Figure A-1: 1, 2)
* Translation of Japanese text to English by the MT servers in the Japanese system (Figure A-1: 3, 4)
* Translation of English text to Chinese by the MT servers in the English system (Figure A-1: 5, 6)
* Speech synthesis in Chinese by the TTS servers in the Chinese system (Figure A-1: 7, 8)

8

Control

Server

(Secondary)

Chinese

Japanese system

ASR

(JA)

2

3

MT

(JA-EN)

Control

Server

(Secondary)

TTS

(ZH)

5

4

ASR

(ZH)

7

MT

(EN-ZH)

6

TTS

(JA)

Japanese

Primary system

Control

Server

(Primary)

Chinese system



1

**Figure A‑1: Example of translation using multiple MCML systems**

## Pivot Translation by the MCML system

The MCML system is capable of pivot translation by appropriately setting the Control Server. By specifying an pivot language and enabling the pivot translation function, the Control Server will automatically choose either of the following two translation protocols, and perform translation.

* + 1. Direct translation

If the MT Communication Server and the MT Engine support the set of languages used by both clients, the system will perform direct translation.



Japanese

English

JA EN

MT

TTS

ASR

Figure A‑2: Direct translation

* + 1. Pivot translation

If the MT Communication Server and the MT Engine do not support the set of languages of both clients, the system will evaluate the possibility of pivot translation through an interlanguage and, if possible, will proceed with pivot translation.



Chinese

JA EN

ASR

MT

Japanese

TTS

EN ZH

MT

Figure A‑3: Pivot translation

## Communication format

* + 1. MCML format in the ASR process

The following are example messages of ASR requests/responses in the MCML format transmitted between the Control Server and the ASR Communication Server.

* + - 1. ASR request

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<User>

<Transmitter>

<Device>

<Location>

<URI>iPhone4</URI>

</Location>

</Device>

<UserProfile ID="tarou" Gender="Male" Age="29" />

</Transmitter>

<Receiver>

<Device>

<Location>

<URI>androidA</URI>

</Location>

</Device>

<UserProfile ID="hanako" Gender="Female" Age="28" />

</Receiver>

</User>

<Server>

<Request Service="ASR" ProcessOrder="1">

<InputUserProfile ID="INPUT" Gender="Male" Age="29">

<InputModality>

<Speaking>

<Language ID="ja" Fluency="0" />

</Speaking>

</InputModality>

</InputUserProfile>

<TargetOutput>

<HypothesisFormatNofN-best="5" />

<LanguageType ID="ja" />

</TargetOutput>

<Input>

<Data>

<Audio ChannelID="1" >

<ModelType>

<Domain>Trabel</Domain>

<Task>Dictation</Task>

</ModelType>

<Signal SamplingRate="16000" ValueType="integer" AudioFormat="raw PCM" BitRate="24" Endian="Little" ChannelQty="0" />

</Audio>

</Data>

<AttachedBinaryChannelID="1" DataID="ja" />

</Input>

</Request>

</Server>

</MCML>

* + - 1. Normal ASR response

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<User>

# Same as request of the ASR Communication Server

</User>

<Server>

<Response Service="ASR" ProcessOrder="1">

<Output>

<Data>

<Text ChannelID="1" >

<ModelType>

<Language ID="ja" />

<Domain>Travel</Domain>

<Task>Dictation </Task>

<Personality ID="EF050A,12.345" Gender="Male" Age="29" />

</ModelType>

<SentenceSequence Order="1" Score="2.44" N-BestRank=1">

<Sentence Order="1">

<Function>text</Function>

<Surface Delimiter="|">おはよう|ござい|ます</Surface>

<Chunk Order="1">

<Surface>おはよう</Surface>

</Chunk>

<Chunk Order="2">

<Surface>ござい</Surface>

</Chunk>

<Chunk Order="3">

<Surface>ます</Surface>

</Chunk>

</Sentence>

</SentenceSequence>

</Text>

</Data>

</Output>

</Response>

</Server>

</MCML>

* + - 1. Error ASR response

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<User>

# Same as request of the ASR Communication Server

</User>

<Server>

<Response Service="ASR" ProcessOrder="1">

<Error Code="String" Message="String" Service="String" />

</Response>

</Server>

</MCML>

* + 1. MCML format in the MT process

The following are example messages of MT requests/responses in the MCML format transmitted between the Control Server and the MT Communication Server.

* + - 1. MT request

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<User>

# Same as request of the ASR Communication Server

</User>

<Server>

<Request Service="MT" ProcessOrder="2">

<InputUserProfile ID="EF050A,12.345" Gender="Male" Age="29">

<InputModality>

<Speaking>

<Language ID="ja" Fluency="0" />

</Speaking>

</InputModality>

</InputUserProfile>

<TargetOutput>

<HypothesisFormatNofN-best="1" />

<LanguageType ID="en" />

</TargetOutput>

<Input>

<Data>

# Same as Output/Data response of the ASR Communication Server

</Data>

</Input>

</Request>

</Server>

</MCML>

* + - 1. Normal MT response

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<User>

# Same as request of the ASR Communication Server

</User>

<Server>

<Response Service="MT" ProcessOrder="2">

<Output>

<Data>

<Text ChannelID="1>

<ModelType>

<Language ID="en" />

<Domain>Travel</Domain>

<Task>Dictation </Task>

<Personality ID="EF050A,12.345" Gender="Male" Age="29" />

</ModelType>

<SentenceSequence Order="8374" Score="2.44" N-BestRank=1">

<Sentence Order="1">

<Function>text</Function>

<Surface Delimiter="|">good|morning</Surface>

<Chunk Order="1">

<Surface>good</Surface>

</Chunk>

<Chunk Order="2">

<Surface>morning</Surface>

</Chunk>

</Sentence>

</SentenceSequence>

</Text>

</Data>

</Output>

</Response>

</Server>

</MCML>

* + - 1. Error MT response

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<User>

# Same as request of the ASR Communication Server

</User>

<Server>

<Response Service="MT" ProcessOrder="1">

<Error Code="String" Message="String" Service="String" />

</Response>

</Server>

</MCML>

* + 1. MCML format in the TTS process

The followings are example messages of TTS requests/responses in the MCML format transmitted between the Control Server and the TTS Communication Server.

* + - 1. TTS request

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<User>

# Same as request of the ASR Communication Server

</User>

<Server>

<Request Service="TTS" ProcessOrder="3">

<InputUserProfile ID="EF050A,12.345" Gender="Male" Age="29">

<InputModality>

<Speaking>

<Language ID="ja" Fluency="0" />

</Speaking>

</InputModality>

</InputUserProfile>

<TargetOutput>

<LanguageType ID="en" />

</TargetOutput>

<Input>

<Data>

<Text ChannelID="1" >

<ModelType>

<Language ID="ja" />

<Domain>Travel</Domain>

<Task>Dictation </Task>

<Personality ID="EF050A,12.345" Gender="Male" Age="29" />

</ModelType>

<SentenceSequence Order="1" Score="2.44" N-BestRank=1">

<Sentence Order="1">

<Function>text</Function>

<Surface Delimiter="|">おはよう|ござい|ます</Surface>

<Chunk Order="1">

<Surface>おはよう</Surface>

</Chunk>

<Chunk Order="2">

<Surface>ござい</Surface>

</Chunk>

<Chunk Order="3">

<Surface>ます</Surface>

</Chunk>

</Sentence>

</SentenceSequence>

</Text>

</Data>

</Input>

</Request>

</Server>

</MCML>

* + - 1. Normal TTS response

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<User>

# Same as request of the ASR Communication Server

</User>

<Server>

<Response Service="TTS" ProcessOrder="3" >

<Output>

<Data>

<Audio ChannelID="1" >

<Signal SamplingRate="16000" ValueType="integer" AudioFormat="raw PCM" BitRate="24" Endian="big" ChannelQty="0" />

<ModelType>

<Language ID="en" />

<Domain>Travel</Domain>

<Personality ID="EF050A,12.345" Gender="Male" Age="29" />

</ModelType>

</Audio>

</Data>

<AttachedBinaryChannelID="1" DataID="en" />

</Output>

</Response>

</Server>

</MCML>

* + - 1. Error TTS response

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<User>

# Same as request of the ASR Communication Server

</User>

<Server>

<Response Service="TTS" ProcessOrder="1">

Error Code="String" Message="String" Service="String" />

</Response>

</Server>

</MCML>

## Communication protocol and data format

* + 1. Communication protocol and data format

Communication data in the MCML system consists of:

* XML

Tag data in the MCML format in the first frame

* Binary data

Audio data for ASR and TTS in the second and later frames.

Binary data does not exist in some cases.

* Last frame

The content is 0 (no data), indicating the terminal frame.



* + 1. Specifications of MIME and socket communication

The XML contents are the same as those of the MCML specification.

Character coding uses UTF8 as well.

The binary data is the same as the MIME binary section.

Socket Communication Specifications

MIME Specifications

MIME-Version: 1.0

Content-Type: multipart/mixed;

Content-Type: multipart/mixed; boundary="=\_NextPart\_000\_00E4\_01C0B1EB.69378110"

Content-Transfer-Encoding: 7bit

--=\_NextPart\_000\_00E4\_01C0B1EB.69378110--Content-Type: application/xml;charset="utf-8"

Content-Transfer-Encoding: 7bit

<?xml version="1.0" encoding="UTF-8"?>

Data Size

<MCML Version="1" xsi:noNamespaceSchemaLocation="MCML4ITU\_Sep6-1.xsd"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

①XML

<User>

</User>

<Server>

<Request Service="ASR" ProcessOrder="1">

</Request>

</Server>

</MCML>

--=\_NextPart\_000\_00E4\_01C0B1EB.69378110--Content-Type: application/octet-stream;

Content-Transfer-Encoding: binary;

Data Size

aW5nPTANCkJvdW5kc0NoZWNrPTANCk92ZXJmbG93Q2hlY2s9MA0KRmxQb2ludENoZWNrPTANCkZE

SVZDaGVjaz0wDQpVbnJvdW5kZWRGUD0wDQpTdGFydE1vZGU9MA0KVW5hdHRlbmRlZD0wDQpUaHJl

②Binary Data

YWRQZXJPYmplY3Q9MA0KTWF4TnVtYmVyT2ZUaHJlYWRzPTENCg0KW01TIFRyYW5zYWN0aW9uIFNl

cnZlcl0NCkF1dG9SZWZyZXNoPTENCg==

--=\_NextPart\_000\_00E4\_01C0B1EB.69378110--

0

* + 1. FrameSync Files

FrameSync files are formatted to enable data exchange when multiple ASR Engines are running in the constructed MCML system. The data consists of fixed-length frames, each of which includes a 4-byte header (the size can be changed) and a data body (10 msec per frame).

* + - 1. Header

Header tags provided in each frame are as follows.

|  |  |
| --- | --- |
| Tag type | Description |
| TOF | Top of file |
| EOF | End of file |
| START | Start of utterance |
| END | End of utterance |
| DATA | Data body |

* + - 1. Example of file organization

An example of file organization of a FrameSync file is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| Frame No. | Header | Data | Description |
| 1 | TOF | (None) | Top of file |
| 2 | START | (None) | Start of utterance |
| 3 | DATA | Data for 10 msec | Valid data |
| 4 | DATA | Data for 10 msec | Valid data |
| 5 | END | (None) | End of utterance |
| 6 | START | (None) | Start of utterance |
| 7 | DATA | Data for 10 msec | Valid data |
| 8 | DATA | Data for 10 msec | Valid data |
| 9 | DATA | Data for 10 msec | Valid data |
| 10 | END | (None) | End of utterance |
| 11 | EOF | (None) | End of file |

More than one utterance sequences can be combined into the same file, e.g., Frames 2-5 and Frames 6-10 in the above table. Also, only the frames with the "DATA" header tag can include valid data; those containing header tags other than "DATA" are errors.