Speech Translation SDK

Server Installation Guide

Version 1.0

Table of Contents

[1. About This Document 3](#_Toc374020903)

[1.1. Overview of the MCML system 4](#_Toc374020904)

[2. Installation Overview 6](#_Toc374020905)

[2.1. Installation Requirements 6](#_Toc374020906)

[3. Installing the Control Servers 7](#_Toc374020907)

[3.1. Installing the Control Servers 7](#_Toc374020908)

[3.2. Parameter setting for the Control Server 8](#_Toc374020909)

[3.3. Parameter setting for connecting Communication Servers 12](#_Toc374020910)

[4. Installing the Common Library for the Communication Servers 14](#_Toc374020911)

[4.1. Installing MCMLLib 14](#_Toc374020912)

[4.2. Installing ServerCommon 14](#_Toc374020913)

[4.3. Installing external libraries 14](#_Toc374020914)

[5. Installing the Common Library to the Communication Servers 15](#_Toc374020915)

[5.1. Installing the ASR Communication Server 15](#_Toc374020916)

[5.2. Parameter settings for the ASR Communication Server 16](#_Toc374020917)

[6. Installing the MT Communication Server 18](#_Toc374020918)

[6.1. Installing the MT Communication Server 18](#_Toc374020919)

[6.2. Parameter settings for the MT Communication Server 19](#_Toc374020920)

[7. Installing the TTS Communication Server 21](#_Toc374020921)

[7.1. Installing the TTS Communication Server 21](#_Toc374020922)

[7.2. Parameter settings for the TTS Communication Server 22](#_Toc374020923)

[8. Examples of Port Number Assignments 25](#_Toc374020924)

[8.1. Ranges of Port Numbers Assigned to Interface the Servers 25](#_Toc374020925)

Update History

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

# About This Document

The aim of this document is to help you to install the MCML system, a server system for Speech Translation SDK (Software Development Kit), and to connect the MCML system with your Automatic Speech Recognition (ASR), Machine Translation (MT) and Text-to-Speech Synthesis (TTS) engines.

## Overview of the MCML system

The MCML system consists of two Control Servers and three Communication Servers, i.e., ASR, MT and TTS Communication Servers (Figure 1‑1: Overview of the MCML system). 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. You may connect with other MCML systems by the use of additional, multilevel Control Servers.

Client

TCP/IP

TCP/IP

ASR

Engine

Other site

Control Server (Primary)

MT

Communication

Server

Control Server

(Secondary)

ASR

Engine

Control Server

(Secondary)

TCP/IP

User’s site

TTS

Communication

Server

TCP/IP

TCP/IP

TCP/IP

ASR

Engine

MT

Engine

TTS

Engine

User’s site

TCP/IP

TCP/IP

TCP/IP

TCP/IP

ASR

Communication

Server

ASR

Communication

Server

ASR

Communication

Server

http (https)

http (https)

Figure 1‑1: Overview of the MCML system

* + 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, as shown in Figure 1-1 ("Other site").
  + 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). Since an ASR Communication Server processes a single language, the same number of ASR Communication Servers is needed as that of the languages subjected to speech recognition.
  + 1. MT Communication Server
* The MT Communication Server transmits and receives process requests/responses between the MT Engine and the Control Server (Secondary). Since an MT Communication Server processes a single direction of translation, the same number of MT Communication Servers is needed as that of the language sets to be translated. For example, in translation between Japanese and English, two MT Communication Servers, one for J to E and another for E to J translations, are required.
  + 1. TTS Communication Server
* The TTS Communication Server transmits and receives process requests/responses between the TTS Engine and the Control Server (Secondary). Since a TTS Communication Server processes a single language, the same number of TTS Communication Servers is needed as that of the languages used for speech synthesis.

# Installation Overview

The following is a software block diagram for the MCML Server.

jre1.7

Linux CentOS 5.6

**TTSCommunication Server**

**MT Communication Server**

**ASR Communication Server**

**Control Server (Secondary)**

Control Server (Primary)

Tomcat 7

Figure 2‑1: Software block diagram

## Installation requirements

|  |  |
| --- | --- |
| Operating System | CentOS 5.6 (64 bit) |
| Java | jre 1.7 or later |
| Tomcat | Version 7 |

# Installing the Control Server

## Installing the Control Server

* + 1. Installing
* Create a WebArchive (war) file and copy it to the following directory for Tomcat setting.
  + [tomcat-install]/webapps/

(Note: [tomcat-install] indicates the Tomcat installation directory hereafter)

* Java heap size setting
  + Edit the property file for the Tomcat server.
    - Property file: [tomcat-install]/bin/catalina.sh

Add CATALINA\_OPTS="-Xms1024m –Xmx1024m"

**Caution:** Do not edit the property file when the Tomcat server is running.

* Editing the property files for the Control Server
  + Edit the following two property files:
    - ControlServer.properties

See "3.2. Parameter setting for the Control Server".

* + - ServerListData.properties

　 See "3.3. Setting of Communication Servers to connect".

* + 1. Starting and Stopping the Control Server
* Starting the Control Server
  + Start the Tomcat server.

$ cd [tomcat-install]//bin

$ ./startup.sh

* + - When the Tomcat server is started, the Control Server will start.
* Stopping the Control Server
  + Stop the Tomcat server.

$ cd [tomcat-install]/bin

$ ./shutdown.sh

* + - When the Tomcat server is stopped, the Control Server will stop.

## Parameter settings for the Control Server

* + 1. Path of the property file
* Parameters for the Control Server are described in the following file:
  + [tomcat-install]/webapps/ControlServer/WEB-INF/classes/ControlServer.properties

server.maxnumber=30

server.corpuslog.clientip=off

listfilename=/tomcat/webapps/ControlServer/ServerListData.properties

timeout.milliseconds=10000

resultadoptionmode=2

　　　・

　　　・

　　　・

Note: In Section 3.2.2 and thereafter, capitals like "X" and "Y" used in parameters for each server indicate numerals, and more than one parameters can be defined (e.g., "serverX" = "server1", "server2", "server3"...).

* + 1. Common parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Detail | Value | Remarks |
| server.maxnumber | Maximum number of servers | Number |  |
| server.corpuslog.clientip | To determine whether or not to output client IP | on or off | Default: off |

* + 1. Parameters for result adoption function

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Detail | Value | Remarks |
| listfilename | Path of the property file for each communication server | String | e.g., [tomcat-install]/webapps/ControlServer/WEB-INF/classes/ServerListData.properties |
| timeout.milliseconds | Timeout period for ASR, MT or TTS process | Number (msec) | 0: No timeout (Default) |
| resultadoptionmode | Result adoption mode | 0 - 3 | 0: Adoption function not used (Default)  1: Respond to the most recent result  2: Order of list  3: Score in preferential order |
| timeout.limit.swicth | To enable or disable the timeout limit | 0 or 1 | 0: OFF (Default)  1: ON |
| timeout.limit.number | Maximum number of timeouts | Number |  |

* + 1. Corpus log for recognition

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Detail | Value | Remarks |
| server.corpuslog.asr.basedirectoryname | Path of the directory to save corpus log files | String | Make the directory before run. |
| server.corpuslog.asr.fileprefix | Name of the corpus log file | String | Specify a prefix string. |
| server.corpuslog.asr.utteranceinfo | To determine whether or not to output a log file with translation information | on or off | Default: on |
| server.corpuslog.asr.removedsymbol | List of symbols to remove from the corpus log file | String | Specify symbols (e.g., \n, \t).  Field separator: “,”  Null: off |
| server.corpuslog.asr.recognitionresult.utf8 | To determine whether or not to output a recognition result file in UTF-8 | on or off | Default: on |
| server.corpuslog.asr.recognitionresult.euc | To determine whether or not to output a recognition result file in EUC | on or off | Default: on |
| server.corpuslog.asr.inputspeechdata | To determine whether or not to output an input sound file | on or off | Default: on |
| server.corpuslog.asr.urlinfo | To determine whether or not to output URL information | on or off | Default: off |

* + 1. Corpus log for translation

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Detail | Value | Remarks |
| server.corpuslog.mt.basedirectoryname | Path of the directory to save corpus log files | String | Make the directory before run. |
| server.corpuslog.mt.fileprefix | Name of the corpus log file | String | Specify a prefix string. |
| server.corpuslog.mt.utteranceinfo | To determine whether or not to output a log file with translation information | on or off | Default: on |
| server.corpuslog.mt.removedsymbol | List of symbols to remove from the corpus log file | String | Specify symbols (e.g., \n, \t).  Field separator: “,”  Null: off |
| server.corpuslog.mt.urlinfo | To determine whether or not to output URL information | on or off | Default: off |

* + 1. Corpus log for speech synthesis

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Detail | Value | Remarks |
| server.corpuslog.tts.basedirectoryname | Path of the directory to save corpus log files | String | Make the directory before run. |
| server.corpuslog.tts.fileprefix | Name of the corpus log file | String | Define a prefix string. |
| server.corpuslog.tts.utteranceinfo | To determine whether or not to output a log file with translation information | on or off | Default: on |
| server.corpuslog.tts.removedsymbol | List of symbols to remove from the corpus log file | String | Specify symbols (e.g., \n, \t).  Field separator: “,”  Null: off |
| server.corpuslog.tts.urlinfo | To determine whether or not to output URL information | on or off | Default: off |

## Parameter settings for Communication Servers to connect to

* + 1. Path of the property file
* Parameters to configure Communication Servers to connect to are described in the following file:
  + [tomcat-install]/webapps/ControlServer/WEB-INF/classes/ServerListData.properties

# Common Parameters

server.maxnumber=30

via.connect=on

via.maxnumber=2

via1.languages=en

via2.languages=ja

　　　・

　　　・

　　　・

* + 1. Common parameters

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | 詳細 | 値 | 備考 |
| server.maxnumber | Maximum number of servers to connect to | Number | Default: 0  0 - 2147483647(Largest 32-bit signed integer) |
| via.connect | To determine whether or not to perform pivot translation | on or off | Default: off |
| via.maxnumber | Maximum number of languages for pivot translation | Number | 0 - INTMAX |
| viaX.languages | Language used in pivot translation | String | Specify language codes (e.g., en, ja).  Two or more codes must be separated by commas. |

* + 1. Parameters for connected servers

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | 詳細 | 値 | 備考 |
| serverX.destination1 | Address of the server that is connected | String | e.g., Control Server (Secondary):  http://127.0.0.1:8080/ControlServer2/ControlServer  Communication Servers:  127.0.0.1:50000 |
| serverX.service | Type of the service of the connected server | String | ASR, MT, or TTS |
| serverX.languages | Target language | String | Specify language codes.  ASR/TTS: 1 code  MT: 2 codes separated by a comma |
| serverX.coefficient.priority | Priority coefficient | Number | 0.0 or more |
| serverX.company1 | Name of the organization of connected servers | NICT |  |

# Installing the Common Library for the Communication Servers

Install MCMLLib, ServerCommon, and external libraries that are commonly used for ASR, MT, and TTS Communication Servers.

## Installing MCMLLib

* + 1. Installing
* Create a directory for server installation and copy the "MCMLLib" folder to it.
  + [server-install]/

(Note: [server-install] indicates the installation directory for each server hereafter.)

MCMLLib/

## Installing ServerCommon

* + 1. Installing
* Copy the "ServerCommon" folder to the server installation directory.
  + [server-install]/

ServerCommon/

## Installing external libraries

* + 1. Installing
* Create a "JAR" folder in the server installation directory and copy appropriate library files to it.
  + [server-install]/

JAR/

activation.jar

mail.jar

# Installing the ASR Communication Server

## Installing the ASR Communication Server

* + 1. Installing
* Copy the “ASRServer” folder to the server installation directory.
  + [server-install]/

ASRServer/

* Editing the property file for ASR Communication Server
  + Edit the following property file:
    - ASRServer.properties
      * One property file per language is needed for recognition.

(See "5.2. Parameter settings for the ASR Communication Server".)

* + 1. Starting and Stopping
* Starting the ASR Communication Server
  + Specify a class path and a property file path by using the "-propertyfilename" option, and execute the command:

$ java –cp [server-install]/ASRServer/bin \

[server-install]/MCMLLib/bin \

[server-install]/ServerCommon/bin \

[server-install]/JAR/activation.jar \

[server-install]/JAR/mail.jar \

jp.go.nict.mcml.server.asr.ASRServer \

-propertyfilename [server-install]/ASRServer/ASRServer.properties

* Stopping the ASR Communication Server
  + Specify a running process ID of the ASR Communication Server, and run the kill command:

$ ps –aux | grep ASRServer

….1234…

$ kill -9 1234

## Parameter settings for the ASR Communication Server

* + 1. Property files

Property files specify the settings used to interface between the ASR Communication Server and the ASR Engine.

servlet.acceptport=50000

engine.maxnumber=2

engine.srtimeout.time=50000

engine.srtimeout.ratio=2000

　　　・

　　　・

　　　・

* + 1. Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Detail | Value | Remarks |
| servlet.acceptport | Port number for socket connections with the Control Server | Number | 0 - 65535 |
| engine.maxnumber | Maximum number of ASR Engine connections | Number | 1 - 2147483647(Largest 32-bit signed integer) |
| engine.srtimeout.time | Recognition timeout period, in milliseconds | Number | 0 - 9223372036854775807(Largest 64-bit signed integer) |
| engine.srtimeout.ratio | Recognition timeout period, in percentage | Number | 0 - 9223372036854775807(Largest 64-bit signed integer) |
| engine.receivetimeout | Reception timeout period for a recognition request | Number | 0 - 9223372036854775807(Largest 64-bit signed integer) |
| engine.corpuslog.basedirectryname | Path of the directory to save corpus log files | String | Make the directory before run. |
| engine.corpuslog.fileprefix | Name of the corpus log file | String | Specify a prefix string. |
| engine.corpuslog.utteranceinfo | To determine whether or not to output a log file of recognition information | on or off | Default: on |
| engine.corpuslog.recognitionresult.utf8 | To determine whether or not to output a recognition results file in UTF-8 | on or off | Default: on |
| engine.corpuslog.recognitionresult.euc | To determine whether or not to output a recognition results file in EUC | on or off | Default: on |
| engine.corpuslog.inputspeechdata | To determine whether or not to output an input sound file | on or off | Default: on |
| engine.language1 | Language for recognition by the Engine | String | Specify language codes. |
| engine.language1.delimiter | Delimiter for recognition results | String | e.g., \*, \\... |
| engine.language1.stringcode | Character encoding used for recognition results | String | Specify character codes (e.g., UTF8, EUC-JP, EUC-CN…). |
| engine.connect.retrytimes | Maximum number of retrying to connect the Engine | Number | 0 - 2147483647(Largest 32-bit signed integer) |
| engine.connect.retryinterval | Interval of retrying to connect the engine (msec) | Number | 0 - 2147483647(Largest 32-bit signed integer) |
| engine.asrX.host | Host name or IP address of the connected ASR Engine | String |  |
| engine.asrX.nbestreceiveport | Port number to accept recognition results from the ASR Engine | Number | 0 - 65535 |
| engine.asrX.speechinputport | Port number to input a sound to the ASR Engine | Number | 0 - 65535 |
| engine.asrX.framesyncsize | Payload size of the FrameSync data | Number | 0 - 2147483647(Largest 32-bit signed integer) |
| log.basefilename | Path name of the log file directory followed by the prefix of the log file name | String | Make the directory before run. |
| log.level | Log levels | 0 - 4 | 0: No input  1: ERROR  2: WARNING  3: TRACE  4: DEBUG |
| engine.textfilterfile | Path of the rule file to correct recognition results | Filename with absolute path | e.g., [server-install]/ASRServer/filter/output\_text\_filter.ja |

# Installing the MT Communication Server

## Installing the MT Communication Server

* + 1. Installing
* Copy the “MTServer” folder to the server installation directory.
  + [server-install]/

MTServer/

* Editing the property file for the MT Communication Server
  + Edit the following property file:
    - MTServer.properties
      * One property file per language set is needed for translation.

(See "6.2. Parameter settings for the MT Communication Server".)

* + 1. Starting and Stopping
* Starting the MT Communication Server
  + Specify a class path and a property file path by using the "-propertyfilename" option, and execute the command:

$ java-cp [server-install]/MTServer/bin \

[server-install]/MCMLLib/bin \

[server-install]/ServerCommon/bin \

[server-install]/JAR/activation.jar \

[server-install]/JAR/mail.jar \

jp.go.nict.mcml.server.mt.MTServer \

-propertyfilename [server-install]/MTServer/MTServer.properties

* Stopping the MT Communication Server
  + Specify a running process ID of the MT Communication Server, and run the kill command:

$ ps –aux | grep MTServer

….1234….

$ kill -9 1234

## Parameter settings for the MT Communication Server

* + 1. Property files

Property files specify the settings used to interface between the MT Communication Server and the MT Engine.

servlet.acceptport=54000

engine.maxnumber=2

engine.corpuslog.basedirectryname=/home/mcml/log/MTServer

engine.corpuslog.fileprefix=MTServerCorpuslogEN

　　　・

　　　・

　　　・

* + 1. Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Detail | Value | Remarks |
| servlet.acceptport | Port number for socket connections with the Control Server | Number | 0 - 65535 |
| engine.maxnumber | Maximum number of MT Engine connections | Number | 1 - 2147483647(Largest 32-bit signed integer) |
| engine.corpuslog.basedirectryname | Path of the directory to save corpus log files | String | Make the directory before run. |
| engine.corpuslog.fileprefix | Name of the corpus log file | String | Specify a prefix string. |
| engine.corpuslog.utteranceinfo | To determine whether or not to output a log file of recognition information | on or off | Default: on |
| engine.corpuslog.removedsymbol | List of symbols to remove from the corpus log file | String | Specify symbols (e.g., \n, \t).  Field separator: “,”  Null: off |
| engine.mtX.startupcommand | Run command for the MT Engine program (startup) | String |  |
| engine.mtX.controllercommand | Run command for the MT Engine program (controller) | String |  |
| log.basefilename | Path name of the log file directory followed by the prefix of the log file name | String | Make the directory before run. |
| log.level | Log levels | 0 - 4 | 0: No output  1: ERROR  2: WARNING  3: TRACE  4: DEBUG |
| engine.request.limit.threshold | Number of requests acceptable at a time | Number | 0 - 9223372036854775807(Largest 64-bit signed integer) |
| engine.request.timeout | Timeout period for a request (msec) | Number | 0 - 9223372036854775807(Largest 64-bit signed integer) |
| engine.request.timeout.interval | Intervals at which monitors request timeout (msec) | Number | 0 - 9223372036854775807(Largest 64-bit signed integer) |
| engine.language.maxnumber | Maximum number of languages processed by the MT Communication Server/MT Engine | Number | 2 - 2147483647(Largest 32-bit signed integer) |
| engine.languageY | Language processed by the MT Communication Server/MT Engine | String | Specify language codes. |
| engine.languageY.stringcode | Character encoding used in text for translation | String | Specify a character codes for each language (e.g., UTF8, EUC-JP, EUC-CN…). |
| engine.languageY.delimiter | Delimiter used in text for translation | String | e.g., \*, \\... |
| engine.languageY.textfilterfile | Path of the rule file used to correct translation results | String | e.g., [server-install]/MTServer/filter/output\_text\_filter.ja |
| engine.direction.commandlistfile | Path of the file used to command the direction of translation. | String | e.g., [server-install]/MTServer/DirectionCommandList.txt |

# Installing the TTS Communication Server

## Installing the TTS Communication Server

* + 1. Installing
* Copy the “TTSServer” folder to the server installation directory.
  + [server-install]/

TTSServer/

* Editing the property file for the TTS Communication Server
  + Edit the following property file:
    - TTSServer.properties
      * One property file per language is needed for synthesis.

(See "7.2. Parameter settings for the TTS Communication Server".)

* + 1. Starting and Stopping
* Starting the TTS Communication Server
  + Specify a class path and a property file path by using the "-propertyfilename" option, and execute the command:

$ java –cp [server-install]/TTSServer/bin \

[server-install]/MCMLLib/bin \

[server-install]/ServerCommon/bin \

[server-install]/JAR/activation.jar \

[server-install]/JAR/mail.jar \

jp.go.nict.mcml.server.tts.TTSServer \

-propertyfilename [server-install]/TTSServer/TTSServer.properties

* Stopping the TTS Communication Server
  + Specify a running process ID of the TTS Communication Server, and run the kill command:

$ ps –aux | grep TTSServer

….1234….

$ kill -9 1234

## Parameter settings for the TTS Communication Server

* + 1. Property files

Property files specify the settings used to interface between the TTS Communication Server and the TTS Engine.

servlet.acceptport=52000

engine.maxnumber=2

engine.corpuslog.basedirectryname=/home/mcml/log/TTSServer

engine.corpuslog.fileprefix=TTSServerCorpuslogEN

　　　・

　　　・

　　　・

* + 1. Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Detail | Value | Remarks |
| servlet.acceptport | Port number for socket connections with the Control Server | Number | 0 - 65535 |
| engine.maxnumber | Maximum number of TTS Engine connection | Number | 1 - 2147483647(Largest 32-bit signed integer) |
| engine.corpuslog.basedirectryname | Path of the directory to save corpus log files | String | Make the directory before run. |
| engine.corpuslog.fileprefix | Name of the corpus log file | String | Specify a prefix string. |
| engine.corpuslog.utteranceinfo | To determine whether or not to output a log file of recognition information | on or off | Default: on |
| engine.corpuslog.removedsymbol | List of symbols to remove from the corpus log file | String | Specify symbols (e.g., \n, \t).  Field separator: “,”  Null: off |
| engine.language1 | Language used in speech synthesis by the TTS Engine | String | Specify language codes. |
| engine.language1.delimiter | Delimiter used in text for synthesis | String | e.g. \*, \\... |
| engine.language1.stringcode | Character encoding used in text for synthesis | String | Specify character codes (e.g., UTF8, EUC-JP, EUC-CN…). |
| engine.language1.countrycode | Country code used in text for synthesis | String | Specify country codes (e.g., JP, US, …). |
| engine.connect.retrytimes | Maximum number of retrying to connect the Engine | Number | 0 - 2147483647(Largest 32-bit signed integer) |
| engine.connect.retryinterval | Interval of retrying to connect the Engine (msec) | Number | 0 - 2147483647(Largest 32-bit signed integer) |
| engine.ttsX.host | Host name or IP address of the connected TTS Engine | String |  |
| engine.ttsX.port | Port number for socket connections with the TTS Engine | Number | 0 - 65535 |
| engine.ttsX.bytesperframe | Capacity (byte per sound data frame) sent to the TTS Engine | Number | 0 - 500000000 |
| engine.cache | To determine whether or not to used the cache function | on or off | No default setting; make sure to set "on" or "off". |
| engine.cache.basedirectryname | Path of the directory to save cache data output files | String | e.g., [server-install]/TTSServer/cache |
| engine.cache.cachedatalistfilename | Name of the list file of cache data | String | e.g., CacheDataListFile.txt |
| engine.cache.maxcachecount | Maximum number of sound data files to be cashed | Number | 0 - 99999 |
| engine.cache.deletecount | Number of files to be deleted when the cache exceeds the maximum | Number | 0 - 99999 |
| log.basefilename | Path name of the log file directory followed by the prefix of the log file name | String | Make the directory before run. |
| log.level | Log levels | 0 - 4 | 0: No output  1: ERROR  2: WARNING  3: TRACE  4: DEBUG |
| engine.textfilterfile | Path of the rule file to correct synthesized text | String | e.g., [server-install]/TTSServer/filter/input\_text\_filter.ja |
| engine.changerate | To enable or disable the function to change the speaking rate of the TTS Engine | on or off | No default setting; make sure to set "on" or "off". |
| engine.rate | Speaking rate of the TTS Engine | Number | 0.5 - 2.0  Default: 0.0 |
| engine.audio.format | Audio file format of synthesized speech | String | Set the value of "MCML/Server/Response/Output/Data/Audio/Signal/AudioFormat" in the MCML format.  Default: "raw PCM" |
| engine.audio.endian | Endian of synthesized speech | String | Set the value ("big" or "little") of "MCML/Server/Response/Output/Data/Audio/Signal/Endian" in the MCML format.  Default: "big" |
| engine.audio.frequency | Sampling frequency of synthesized speech (Hz) | Number | Set the value (8000 or 16000) of "MCML/Server/Response/Output/Data/Audio/Signal/SamplingRate" in the MCML format.  Default: 16000 |
| engine.audio.valuetype | Data type of synthesized speech | String | Set the value ("integer" or " float") of "MCML/Server/Response/Output/Data/Audio/Signal/ValueType" in the MCML format.  Default: "integer" |
| engine.audio.bitrate | Bit rate of synthesized speech (bit) | Number | Set the value of "MCML/Server/Response/Output/Data/Audio/Signal/BitRate" in the MCML format.  Default: 16 |
| engine.audio.channelqty | Number of channels of synthesized speech | Number | Set the value of "MCML/Server/Response/Output/Data/Audio/ChannelQt" in the MCML format.  Default: 1 |

# Examples of Port Number Assignments

## Ranges of Port Numbers Assigned to Interface the Servers

* + 1. ControlServer(ClientSocket) <-->ASRServer(ServerSocket)
* 50000 - 50999
  + 1. ControlServer(ClientSocket) <-->MTServer(ServerSocket)
* 54000 - 59999
  + 1. ControlServer(ClientSocket) <-->TTSServer(ServerSocket)
* 52000 - 52999