

MCloud

Generated by Doxygen 1.8.6

Tue Jul 15 2014 09:26:09

Contents

1	Deprecated List	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Data Structure Documentation	7
4.1	MCloudPacket_S Struct Reference	7
4.1.1	Field Documentation	8
4.1.1.1	cmType	8
4.1.1.2	creator	8
4.1.1.3	dataType	8
4.1.1.4	doc	8
4.1.1.5	fingerPrint	8
4.1.1.6	packetType	8
4.1.1.7	revision	8
4.1.1.8	sessionID	8
4.1.1.9	start	8
4.1.1.10	startOffset	8
4.1.1.11	statusDescription	8
4.1.1.12	stop	8
4.1.1.13	stopOffset	9
4.1.1.14	streamID	9
4.1.1.15	userID	9
4.1.1.16	xmlString	9
4.2	MCloudWordToken_S Struct Reference	9
4.2.1	Field Documentation	9
4.2.1.1	confidence	9
4.2.1.2	index	9
4.2.1.3	internal	10

4.2.1.4	isFiller	10
4.2.1.5	spoken	10
4.2.1.6	startTime	10
4.2.1.7	stopTime	10
4.2.1.8	written	10
5	File Documentation	11
5.1	MCloud.h File Reference	11
5.1.1	Macro Definition Documentation	14
5.1.1.1	mcloudERROR	14
5.1.1.2	mcloudINFO	14
5.1.1.3	mcloudWARN	14
5.1.1.4	SHAREDLL	14
5.1.2	Typedef Documentation	14
5.1.2.1	MCloud	14
5.1.2.2	MCloudCallbackFct	15
5.1.2.3	MCloudPacket	15
5.1.2.4	MCloudPacketCallbackFct	15
5.1.2.5	MCloudWordToken	15
5.1.3	Enumeration Type Documentation	15
5.1.3.1	MCloudAttribute	15
5.1.3.2	MCloudCodec	15
5.1.3.3	MCloudType	16
5.1.3.4	S2S_Result	16
5.1.4	Function Documentation	16
5.1.4.1	base64_decode	16
5.1.4.2	base64_encode	16
5.1.4.3	mcloudAddFlowDescription	17
5.1.4.4	mcloudAddFlowDescription2	17
5.1.4.5	mcloudAddService	18
5.1.4.6	mcloudAnnounceOutputStream	18
5.1.4.7	mcloudBreak	18
5.1.4.8	mcloudConnect	19
5.1.4.9	mcloudCreate	19
5.1.4.10	mcloudDisconnect	19
5.1.4.11	mcloudFree	19
5.1.4.12	mcloudGetAttr	20
5.1.4.13	mcloudGetNextPacket	20
5.1.4.14	mcloudMsgHandler	20
5.1.4.15	mcloudPacketAddAudio	20

5.1.4.16	mcloudPacketDeinit	21
5.1.4.17	mcloudPacketGetAudio	21
5.1.4.18	mcloudPacketGetBinary	21
5.1.4.19	mcloudPacketGetText	22
5.1.4.20	mcloudPacketGetWordTokenA	22
5.1.4.21	mcloudPacketInitFromAudio	22
5.1.4.22	mcloudPacketInitFromBinary	23
5.1.4.23	mcloudPacketInitFromCmGet	23
5.1.4.24	mcloudPacketInitFromImage	24
5.1.4.25	mcloudPacketInitFromText	25
5.1.4.26	mcloudPacketInitFromWordTokenA	25
5.1.4.27	mcloudPacketReplaceText	26
5.1.4.28	mcloudPending	26
5.1.4.29	mcloudProcessDataAsync	26
5.1.4.30	mcloudRequestForDisplay	27
5.1.4.31	mcloudRequestInputStream	27
5.1.4.32	mcloudSendBinaryFile	27
5.1.4.33	mcloudSendBinaryFileAsync	28
5.1.4.34	mcloudSendDone	28
5.1.4.35	mcloudSendError	28
5.1.4.36	mcloudSendFlush	29
5.1.4.37	mcloudSendPacket	29
5.1.4.38	mcloudSendPacketAsync	29
5.1.4.39	mcloudSetAttr	30
5.1.4.40	mcloudSetAudioEncoder	30
5.1.4.41	mcloudSetAudioEncoder2	30
5.1.4.42	mcloudSetBreakCallback	31
5.1.4.43	mcloudSetCustomizationCallback	31
5.1.4.44	mcloudSetDataCallback	31
5.1.4.45	mcloudSetErrorCallback	31
5.1.4.46	mcloudSetFinalizeCallback	32
5.1.4.47	mcloudSetInitCallback	32
5.1.4.48	mcloudWaitFinish	32
5.1.4.49	mcloudWaitForClient	32
5.1.4.50	mcloudWordTokenArrayCreate	34
5.1.4.51	mcloudWordTokenArrayFree	34
5.1.4.52	url_decode	34
5.1.4.53	url_encode	34

Chapter 1

Deprecated List

Global `mcloudAddFlowDescription` (MCloud *cloudP, const char *password, int logging, const char *language, const char *name, const char *description)

This function has to be called after an MCloud object has been created and before connecting to the MCloud. A client can add more than one flow being just translations of the same descriptions. Therefore, the password and logging has to be the same over all flows. This call is deprecated and will be removed in future versions of the API

Global `mcloudPacketAddAudio` (MCloud *cloudP, MCloudPacket *p, const char *startTime, const char *stopTime, const char *fingerPrint, const short *sampleA, int sampleN, int isFinal)

This function can be used only for PCM s16le 16KHz audio both as input and as packet content!

Global `mcloudPacketInitFromAudio` (MCloud *cloudP, const char *startTime, const char *stopTime, const char *fingerPrint, const short *sampleA, int sampleN, int isFinal)

This function can be used only for PCM s16le 16KHz audio both as input and as packet content!

Global `mcloudSetAttr` (MCloud *cloudP, MCloudAttribute attr, const void *value)

Set the value of an attribute.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

MCloudPacket_S	7
MCloudWordToken_S	9

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

MCloud.h	11
------------------------------------	----

Chapter 4

Data Structure Documentation

4.1 MCloudPacket_S Struct Reference

```
#include <MCloud.h>
```

Data Fields

- [MCloudType packetType](#)
Type of the packet.
- [MCloudType dataType](#)
Type of the data included if packet is of type MCloudData.
- char * [sessionID](#)
The current sessionID which the packet belongs to.
- char * [streamID](#)
The current streamID which the packet belongs to.
- char * [fingerPrint](#)
Fingerprint of the packet.
- char * [creator](#)
Name of the creator of the packet.
- char * [start](#)
Human readable time stamp identifying the start time of the packet "dd/MM/YY-hh:mm:ss.mss".
- char * [stop](#)
Human readable time stamp identifying the end time of the packet "dd/MM/YY-hh:mm:ss.mss".
- unsigned int [startOffset](#)
Start time offset in ms relative to the beginning of the stream.
- unsigned int [stopOffset](#)
Stop time offset in ms relative to the beginning of the stream.
- char * [statusDescription](#)
Optional detailed status description in case of status messages.
- char * [userID](#)
- char * [cmType](#)
The current userID.
- char * [revision](#)
The CM-type of the packet.
- char * [xmlString](#)
The current revision of the packet.
- xmlDoc * [doc](#)
Reference to the whole XML document (libXML2 xmlDoc)

4.1.1 Field Documentation

4.1.1.1 `char* MCloudPacket_S::cmType`

The current userID.

4.1.1.2 `char* MCloudPacket_S::creator`

Name of the creator of the packet.

4.1.1.3 `MCloudType MCloudPacket_S::dataType`

Type of the data included if packet is of type MCloudData.

4.1.1.4 `xmlDoc* MCloudPacket_S::doc`

Reference to the whole XML document (libXML2 xmlDoc)

4.1.1.5 `char* MCloudPacket_S::fingerPrint`

Fingerprint of the packet.

4.1.1.6 `MCloudType MCloudPacket_S::packetType`

Type of the packet.

4.1.1.7 `char* MCloudPacket_S::revision`

The CM-type of the packet.

4.1.1.8 `char* MCloudPacket_S::sessionID`

The current sessionID which the packet belongs to.

4.1.1.9 `char* MCloudPacket_S::start`

Human readable time stamp identifying the start time of the packet "dd/MM/YY-hh:mm:ss.mss".

4.1.1.10 `unsigned int MCloudPacket_S::startOffset`

Start time offset in ms relative to the beginning of the stream.

4.1.1.11 `char* MCloudPacket_S::statusDescription`

Optional detailed status description in case of status messages.

4.1.1.12 `char* MCloudPacket_S::stop`

Human readable time stamp identifying the end time of the packet "dd/MM/YY-hh:mm:ss.mss".

4.1.1.13 unsigned int MCloudPacket_S::stopOffset

Stop time offset in ms relative to the beginning of the stream.

4.1.1.14 char* MCloudPacket_S::streamID

The current streamID which the packet belongs to.

4.1.1.15 char* MCloudPacket_S::userID

4.1.1.16 char* MCloudPacket_S::xmlString

The current revision of the packet.

Raw XML string

The documentation for this struct was generated from the following file:

- [MCloud.h](#)

4.2 MCloudWordToken_S Struct Reference

```
#include <MCloud.h>
```

Data Fields

- int [index](#)
The token index.
- char * [internal](#)
The internal form of the token.
- char * [written](#)
The written form of the token (can be NULL)
- char * [spoken](#)
The spoken form of the token (optional)
- float [confidence](#)
The confidence value in the interval [0,1].
- unsigned int [startTime](#)
The start time [ms] relative to the start of the stream.
- unsigned int [stopTime](#)
The end time [ms] relative to the start of the stream.
- int [isFiller](#)
This value is set to 1, if the token is a filler token and not a regular word.

4.2.1 Field Documentation

4.2.1.1 float MCloudWordToken_S::confidence

The confidence value in the interval [0,1].

4.2.1.2 int MCloudWordToken_S::index

The token index.

4.2.1.3 char* MCloudWordToken_S::internal

The internal form of the token.

4.2.1.4 int MCloudWordToken_S::isFiller

This value is set to 1, if the token is a filler token and not a regular word.

4.2.1.5 char* MCloudWordToken_S::spoken

The spoken form of the token (optional)

4.2.1.6 unsigned int MCloudWordToken_S::startTime

The start time [ms] relative to the start of the stream.

4.2.1.7 unsigned int MCloudWordToken_S::stopTime

The end time [ms] relative to the start of the stream.

4.2.1.8 char* MCloudWordToken_S::written

The written form of the token (can be NULL)

The documentation for this struct was generated from the following file:

- [MCloud.h](#)

Chapter 5

File Documentation

5.1 MCloud.h File Reference

```
#include <libxml/tree.h>
```

Data Structures

- struct [MCloudPacket_S](#)
- struct [MCloudWordToken_S](#)

Macros

- #define [SHAREDLL](#)
- #define [mcloudERROR](#)(...) [mcloudMsgHandler](#)(__FILE__, __LINE__, 1, __VA_ARGS__)
- #define [mcloudWARN](#)(...) [mcloudMsgHandler](#)(__FILE__, __LINE__, 2, __VA_ARGS__)
- #define [mcloudINFO](#)(...) [mcloudMsgHandler](#)(__FILE__, __LINE__, 3, __VA_ARGS__)

Typedefs

- typedef struct [MCloudPacket_S](#) [MCloudPacket](#)
- typedef struct [MCloudWordToken_S](#) [MCloudWordToken](#)
- typedef struct [MCloud_S](#) [MCloud](#)
- typedef int [MCloudCallbackFct](#) ([MCloud](#) *cloudP, void *userData)
MCloud general callback function type used for finalize, break, and error.
- typedef int [MCloudPacketCallbackFct](#) ([MCloud](#) *cloudP, [MCloudPacket](#) *p, void *userData)
MCloud packet callback function type used for data and init.

Enumerations

- enum [S2S_Result](#) { [S2S_Success](#) = 0, [S2S_Error](#) = 1 }
- enum [MCloudType](#) {
[MCloudModeWorker](#) = 1, [MCloudModeClient](#), [MCloudData](#), [MCloudDone](#),
[MCloudError](#), [MCloudReset](#), [MCloudFlush](#), [MCloudAudio](#),
[MCloudText](#), [MCloudImage](#), [MCloudMixed](#), [MCloudBinary](#),
[MCloudSendingQueue](#), [MCloudProcessingQueue](#), [MCloudCustomization](#) }

- enum `MCloudAttribute` {
`MCloudA_sAudioCodec`, `MCloudA_iSampleRate`, `MCloudA_iSampleSize`, `MCloudA_iChannelN`,
`MCloudA_iBitRate` }
- enum `MCloudCodec` {
`MCloudAC_UNK` = 0, `MCloudAC_PCM`, `MCloudAC_FLAC`, `MCloudAC_SPEEX`,
`MCloudAC_OPUS` }

Functions

- **SHAREDLL** void `mcloudMsgHandler` (const char *file, int line, int type, const char *format,...)
Message handler that should be only used with the macros defined above.
- **SHAREDLL** `MCloudWordToken` * `mcloudWordTokenArrayCreate` (int n)
Creates an array of MCloudWordTokens.
- **SHAREDLL** void `mcloudWordTokenArrayFree` (`MCloudWordToken` *tokenA, int n)
Free an array of MCloudWordTokens.
- **SHAREDLL** `MCloudPacket` * `mcloudPacketInitFromText` (`MCloud` *cloudP, const char *startTime, const char *stopTime, unsigned int startOffset, unsigned int stopOffset, const char *fingerPrint, const char *text)
Initialize a new packet from text for sending.
- **SHAREDLL** `MCloudPacket` * `mcloudPacketInitFromWordTokenA` (`MCloud` *cloudP, const char *startTime, const char *stopTime, unsigned int startOffset, unsigned int stopOffset, const char *fingerPrint, `MCloudWordToken` *tokenA, int tokenN)
Initialize a new packet from a MCloudWordToken array for sending.
- **SHAREDLL** `MCloudPacket` * `mcloudPacketInitFromAudio` (`MCloud` *cloudP, const char *startTime, const char *stopTime, const char *fingerPrint, const short *sampleA, int sampleN, int isFinal)
Initialize a new packet from audio for sending.
- **SHAREDLL** `MCloudPacket` * `mcloudPacketInitFromBinary` (`MCloud` *cP, const char *startTime, const char *stopTime, const char *fingerPrint, const char *filename, const char *mimetype, const uint8_t *bytes, int bytesN, int last)
Initialize a new packet from binary for sending.
- `MCloudPacket` * `mcloudPacketInitFromImage` (`MCloud` *cloudP, const char *startTime, const char *stopTime, const char *fingerPrint, int width, int height, const char *format, const char *buffer, int bufferN)
Initialize a new packet from image for sending.
- **SHAREDLL** `MCloudPacket` * `mcloudPacketAddAudio` (`MCloud` *cloudP, `MCloudPacket` *p, const char *startTime, const char *stopTime, const char *fingerPrint, const short *sampleA, int sampleN, int isFinal)
Add audio to an existing package for sending. Packet type will be changed to "mixed".
- **SHAREDLL** void `mcloudPacketDeinit` (`MCloudPacket` *p)
Free a packet.
- **SHAREDLL** `S2S_Result` `mcloudPacketGetText` (`MCloud` *cloudP, `MCloudPacket` *p, char **text)
Convenient function for extracting the string embedded in <text></text>.
- **SHAREDLL** `S2S_Result` `mcloudPacketReplaceText` (`MCloud` *cloudP, `MCloudPacket` *p, char *text)
Convenient function for replacing the string embedded in <text></text>.
- **SHAREDLL** `S2S_Result` `mcloudPacketGetWordTokenA` (`MCloud` *cloudP, `MCloudPacket` *p, `MCloudWordToken` **tokenA, int *tokenN)
Convenient function for extracting a word token array embedded in <wordtokens></wordtokens>.
- **SHAREDLL** `S2S_Result` `mcloudPacketGetAudio` (`MCloud` *cloudP, `MCloudPacket` *p, short **sampleA, int *sampleN)
Convenient function for extracting the data embedded in <audio></audio>.
- **SHAREDLL** `S2S_Result` `mcloudPacketGetBinary` (`MCloud` *cP, `MCloudPacket` *p, uint8_t **bA, int *bN, char **filename, char **mimetype, int *last)
Convenient function for extracting the data embedded in <audio></audio>
- **SHAREDLL** `MCloud` * `mcloudCreate` (const char *name, int mode)
Create an MCloud object with a given name and mode.

- **SHAREDLL** void **mcloudFree** (MCloud *cloudP)
Free an MCloud object.
- **SHAREDLL** S2S_Result **mcloudGetAttr** (MCloud *cloudP, MCloudAttribute attr, void *value)
Get the value of an attribute.
- **SHAREDLL** S2S_Result **mcloudSetAttr** (MCloud *cloudP, MCloudAttribute attr, const void *value)
Set the value of an attribute.
- **SHAREDLL** S2S_Result **mcloudSetAudioEncoder** (MCloud *cp, MCloudCodec codec, int sampleRate, int bitRate, int channels)
Set the audio codec.
- **SHAREDLL** S2S_Result **mcloudSetAudioEncoder2** (MCloud *cp, char *codec, int sampleRate, int bitRate, int channels)
Set the audio codec.
- **SHAREDLL** S2S_Result **mcloudConnect** (MCloud *cloudP, const char *host, int port)
Connect to the MCloud server running on the host at port given.
- **SHAREDLL** S2S_Result **mcloudDisconnect** (MCloud *cloudP)
Disconnect from the MCloud server.
- **SHAREDLL** S2S_Result **mcloudAddService** (MCloud *cloudP, const char *name, const char *service, const char *inputFingerPrint, const char *inputType, const char *outputFingerPrint, const char *outputType, const char *specifier)
Add a service description of a worker to an MCloud object.
- **SHAREDLL** S2S_Result **mcloudAddFlowDescription2** (MCloud *cloudP, const char *username, const char *password, int logging, const char *language, const char *name, const char *description)
Add a flow description of a client to an MCloud object.
- **SHAREDLL** S2S_Result **mcloudAddFlowDescription** (MCloud *cloudP, const char *password, int logging, const char *language, const char *name, const char *description)
Add a flow description of a client to an MCloud object.
- **SHAREDLL** S2S_Result **mcloudAnnounceOutputStream** (MCloud *cloudP, const char *type, const char *fingerPrint, const char *streamID, const char *specifier)
Announce an output stream of a client to an MCloud object.
- **SHAREDLL** S2S_Result **mcloudRequestInputStream** (MCloud *cloudP, const char *type, const char *fingerPrint, const char *streamID, char *info, int infoN)
Request an input stream of a client from an MCloud object.
- **SHAREDLL** S2S_Result **mcloudRequestForDisplay** (MCloud *cloudP)
Request the display of an output stream.
- **SHAREDLL** S2S_Result **mcloudWaitForClient** (MCloud *cloudP, char **streamID)
Wait for a service request to process.
- **SHAREDLL** MCloudPacket * **mcloudGetNextPacket** (MCloud *cloudP)
Wait for the next data package.
- **SHAREDLL** S2S_Result **mcloudSendPacket** (MCloud *cloudP, MCloudPacket *p)
Send a packet.
- **SHAREDLL** S2S_Result **mcloudSendPacketAsync** (MCloud *cloudP, MCloudPacket *p, void *userData)
Send a packet asynchronously.
- **SHAREDLL** S2S_Result **mcloudSendBinaryFile** (MCloud *cP, FILE *f, int chunkSize, char *filename, char *mimeType, char *fingerPrint)
Convenience function for sending the content of a whole file.
- **SHAREDLL** S2S_Result **mcloudSendBinaryFileAsync** (MCloud *cP, FILE *f, int chunkSize, char *filename, char *mimeType, char *fingerPrint, void *userData)
Convenience function for sending the content of a whole file asynchronously.
- **SHAREDLL** S2S_Result **mcloudSendDone** (MCloud *cloudP)
Inform a client or a worker that there is no more data to receive.
- **SHAREDLL** S2S_Result **mcloudSendError** (MCloud *cloudP, const char *description)
Inform a client or a worker that an error occurred during processing.

- **SHAREDLL S2S_Result mcloudSendFlush** (MCloud *cloudP)
Inform subsequent worker to flush their output buffers.
- **SHAREDLL S2S_Result mcloudProcessDataAsync** (MCloud *cloudP, MCloudPacket *p, void *userData)
Process received packages asynchronously.
- **SHAREDLL int mcloudPending** (MCloud *cloudP, MCloudType queueType)
Return number of pending packages in queue.
- **SHAREDLL S2S_Result mcloudWaitFinish** (MCloud *cloudP, MCloudType queueType, int done)
Wait until all pending packages have been processed/ sent.
- **SHAREDLL S2S_Result mcloudBreak** (MCloud *cloudP, MCloudType queueType)
Stop processing, sending pending packages immediately, and reset queue.
- **SHAREDLL void mcloudSetInitCallback** (MCloud *cloudP, MCloudPacketCallbackFct *callback, void *userData)
Set an init callback function.
- **SHAREDLL void mcloudSetFinalizeCallback** (MCloud *cloudP, MCloudCallbackFct *callback, void *userData)
Set a finalize callback function.
- **SHAREDLL void mcloudSetBreakCallback** (MCloud *cloudP, MCloudType queueType, MCloudCallbackFct *callback, void *userData)
Set a break callback function.
- **SHAREDLL void mcloudSetErrorCallback** (MCloud *cloudP, MCloudType queueType, MCloudCallbackFct *callback, void *userData)
Set an error callback function.
- **SHAREDLL void mcloudSetDataCallback** (MCloud *cloudP, MCloudPacketCallbackFct *callback)
Set a data callback function.
- **SHAREDLL void mcloudSetCustomizationCallback** (MCloud *cP, MCloudPacketCallbackFct *callback, void *userData)
Set a customization callback function.
- **SHAREDLL MCloudPacket * mcloudPacketInitFromCmGet** (MCloud *cP, const char *sessionId, const char *userId, const char *fingerPrint, const char *type, const char *revision)
Initialize a new packet from a CmGet for sending.
- **SHAREDLL char * base64_encode** (const char *data, size_t input_length, size_t *output_length)
Convenient function for encoding the data in base64.
- **SHAREDLL char * base64_decode** (const char *data, size_t input_length, size_t *output_length)
Convenient function for decoding the data from base64.
- **SHAREDLL char * url_encode** (const char *str)
Convenient function that returns a url-encoded version of a string.
- **SHAREDLL char * url_decode** (const char *str)
Convenient function that returns a url-decoded version of a string.

5.1.1 Macro Definition Documentation

5.1.1.1 **#define mcloudERROR(...) mcloudMsgHandler(__FILE__, __LINE__, 1, __VA_ARGS__)**

5.1.1.2 **#define mcloudINFO(...) mcloudMsgHandler(__FILE__, __LINE__, 3, __VA_ARGS__)**

5.1.1.3 **#define mcloudWARN(...) mcloudMsgHandler(__FILE__, __LINE__, 2, __VA_ARGS__)**

5.1.1.4 **#define SHAREDLL**

5.1.2 Typedef Documentation

5.1.2.1 **typedef struct MCloud_S MCloud**

MCloud object.

5.1.2.2 typedef int MCloudCallbackFct(MCloud *cloudP, void *userData)

MCloud general callback function type used for finalize, break, and error.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>userData</i>	user data

Returns

0 if success

5.1.2.3 typedef struct MCloudPacket_S MCloudPacket

5.1.2.4 typedef int MCloudPacketCallbackFct(MCloud *cloudP, MCloudPacket *p, void *userData)

MCloud packet callback function type used for data and init.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>p</i>	packet containing the data to process
in	<i>userData</i>	user data

Returns

0 if success

5.1.2.5 typedef struct MCloudWordToken_S MCloudWordToken

5.1.3 Enumeration Type Documentation

5.1.3.1 enum MCloudAttribute

Enumerator

MCloudA_sAudioCodec Audio codec type [RPCM](#) ([raw PCM](#)), [SPEEX](#) ([Speex](#)), [OPUS](#) ([Opus](#)), [FLAC](#) ([Flac](#)) (default: [RPCM](#))

MCloudA_iSampleRate input/ output sample rate in Hz (get/set) (default: 16000)

MCloudA_iSampleSize input/ output sample size in bits (get/set) (default: 16)

MCloudA_iChannelIN input/ output number of channels (get/set) (default: 1)

MCloudA_iBitRate input / output bit rate in bits/sec

5.1.3.2 enum MCloudCodec

Defines the supported encoding

Enumerator

MCloudAC_UNK

MCloudAC_PCM

MCloudAC_FLAC

MCloudAC_SPEEX

MCloudAC_OPUS

5.1.3.3 enum MCloudType

Enumerator

MCloudModeWorker worker mode
MCloudModeClient client mode
MCloudData message of type data
MCloudDone status message of type done
MCloudError status message of type error
MCloudReset status message of type reset
MCloudFlush status message of type flush
MCloudAudio data packet of type audio
MCloudText data packet of type text
MCloudImage data packet of type image
MCloudMixed data packet of type mixed
MCloudBinary data packet of type binary
MCloudSendingQueue sending queue
MCloudProcessingQueue receiving/ processing queue
MCloudCustomization message with customizations

5.1.3.4 enum S2S_Result

Enumerator

S2S_Success success
S2S_Error error

5.1.4 Function Documentation

5.1.4.1 SHAREDLL char* base64_decode (const char * data, size_t input_length, size_t * output_length)

Convenient function for decoding the data from base64.

Parameters

in	<i>data</i>	data to be decoded
in	<i>input_length</i>	length of input data stream
in	<i>output_length</i>	length of output data stream

Returns

decoded data

5.1.4.2 SHAREDLL char* base64_encode (const char * data, size_t input_length, size_t * output_length)

Convenient function for encoding the data in base64.

Parameters

in	<i>data</i>	data to be encoded
in	<i>input_length</i>	length of input data stream
in	<i>output_length</i>	length of output data stream

Returns

encoded data

5.1.4.3 SHAREDLL S2S_Result mcloudAddFlowDescription (MCloud * *cloudP*, const char * *password*, int *logging*, const char * *language*, const char * *name*, const char * *description*)

Add a flow description of a client to an MCloud object.

Deprecated This function has to be called after an MCloud object has been created and before connecting to the MCloud. A client can add more than one flow being just translations of the same descriptions. Therefore, the password and logging has to be the same over all flows. This call is deprecated and will be removed in future versions of the API

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>password</i>	an optional password which has to be used in order to subscribe to this flow (display server), NULL for no password
in	<i>logging</i>	if set to 0, the flow will not be logged in the database
in	<i>language</i>	descriptive language identifier of the flow, e.g. English
in	<i>name</i>	name of the flow, e.g. title of a talk
in	<i>description</i>	additional description of the flow, e.g. abstract

Returns

S2S_Success if no error occurs

5.1.4.4 SHAREDLL S2S_Result mcloudAddFlowDescription2 (MCloud * *cloudP*, const char * *username*, const char * *password*, int *logging*, const char * *language*, const char * *name*, const char * *description*)

Add a flow description of a client to an MCloud object.

This function has to be called after an MCloud object has been created and before connecting to the MCloud. A client can add more than one flow being just translations of the same descriptions. Therefore, the password and logging has to be the same over all flows.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>username</i>	an optional username which has to be used in order to subscribe to this flow (display server), NULL for no username
in	<i>password</i>	an optional password which has to be used in order to subscribe to this flow (display server), NULL for no password
in	<i>logging</i>	if set to 0, the flow will not be logged in the database
in	<i>language</i>	descriptive language identifier of the flow, e.g. English

in	<i>name</i>	name of the flow, e.g. title of a talk
in	<i>description</i>	additional description of the flow, e.g. abstract

Returns

S2S_Success if no error occurs

5.1.4.5 SHAREDLL S2S_Result mcloudAddService (MCloud * *cloudP*, const char * *name*, const char * *service*, const char * *inputFingerPrint*, const char * *inputType*, const char * *outputFingerPrint*, const char * *outputType*, const char * *specifier*)

Add a service description of a worker to an MCloud object.

This function has to be called after an MCloud object has been created and before connecting to the MCloud.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>name</i>	name of the worker
in	<i>service</i>	name of the service (asr, smt, tts, ...)
in	<i>inputFingerPrint</i>	service input finger print
in	<i>inputType</i>	data input type (audio, text)
in	<i>outputFingerPrint</i>	service output finger print
in	<i>outputType</i>	data output type (audio, text)
in	<i>specifier</i>	an additional specifier, i.e. a speaker identifier

Returns

S2S_Success if no error occurs

5.1.4.6 SHAREDLL S2S_Result mcloudAnnounceOutputStream (MCloud * *cloudP*, const char * *type*, const char * *fingerPrint*, const char * *streamID*, const char * *specifier*)

Announce an output stream of a client to an MCloud object.

This function has to be called after the client has been connected to the MCloud.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>type</i>	data type (audio, text, image)
in	<i>fingerPrint</i>	finger print of the data stream
in	<i>streamID</i>	unique stream identifier
in	<i>specifier</i>	an additional specifier, i.e. a speaker identifier

Returns

S2S_Success if no error occurs

5.1.4.7 SHAREDLL S2S_Result mcloudBreak (MCloud * *cloudP*, MCloudType *queueType*)

Stop processing, sending pending packages immediately, and reset queue.

This function can be used to stop further processing or sending packages in the queue specified.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>queueType</i>	type of queue MCloudSendingQueue, or MCloudProcessingQueue

Returns

S2S_Success if no error occurs

5.1.4.8 SHAREDLL S2S_Result mcloudConnect (MCloud * *cloudP*, const char * *host*, int *port*)

Connect to the MCloud server running on the host at port given.

This function has to be called after an MCloud object has been created and before waiting for a client or worker.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>host</i>	host name
in	<i>port</i>	port number

Returns

S2S_Success if no error occurs

5.1.4.9 SHAREDLL MCloud* mcloudCreate (const char * *name*, int *mode*)

Create an MCloud object with a given name and mode.

Parameters

in	<i>name</i>	descriptive name of the worker or client (used as 'creator' in XML messages)
in	<i>mode</i>	working mode, i.e. MCloudModeWorker, or MCloudModeClient

Returns

reference to an MCloud object or NULL if failed

5.1.4.10 SHAREDLL S2S_Result mcloudDisconnect (MCloud * *cloudP*)

Disconnect from the MCloud server.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
----	---------------	-------------------------------

Returns

S2S_Success if no error occurs

5.1.4.11 SHAREDLL void mcloudFree (MCloud * *cloudP*)

Free an MCloud object.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
----	---------------	-------------------------------

5.1.4.12 SHAREDLL S2S_Result mcloudGetAttr (MCloud * *cloudP*, MCloudAttribute *attr*, void * *value*)

Get the value of an attribute.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>attr</i>	an MCloud attribute
out	<i>value</i>	reference to the returned value

Returns

S2S_Success if no error occurs

5.1.4.13 SHAREDLL MCloudPacket* mcloudGetNextPacket (MCloud * *cloudP*)

Wait for the next data package.

This function has to be called in order to wait for the next data packet. The function times out after an internally specified amount of time.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
----	---------------	-------------------------------

Returns

S2S_Success if no error occurs

5.1.4.14 SHAREDLL void mcloudMsgHandler (const char * *file*, int *line*, int *type*, const char * *format*, ...)

Message handler that should be only used with the macros defined above.

Parameters

in	<i>file</i>	source code file name FILE
in	<i>line</i>	source code line LINE
in	<i>type</i>	message type
in	<i>format</i>	printf format
in	...	variable list of additional arguments

5.1.4.15 SHAREDLL MCloudPacket* mcloudPacketAddAudio (MCloud * *cloudP*, MCloudPacket * *p*, const char * *startTime*, const char * *stopTime*, const char * *fingerprint*, const short * *sampleA*, int *sampleN*, int *isFinal*)

Add audio to an existing package for sending. Packet type will be changed to "mixed".

Deprecated This function can be used only for PCM s16le 16KHz audio both as input and as packet content!

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>p</i>	MCloudPacket to which the audio should be added
in	<i>startTime</i>	human readable starting time stamp
in	<i>stopTime</i>	human readable ending time stamp
in	<i>fingerPrint</i>	finger print of the audio
in	<i>sampleA</i>	reference to an array of audio samples
in	<i>sampleN</i>	number of samples in array
in	<i>isFinal</i>	indicates whether the sample array given is the final one

Returns

reference to the created package or NULL if failed

5.1.4.16 SHAREDLL void mcloudPacketDeinit (MCloudPacket * *p*)

Free a packet.

Parameters

in	<i>p</i>	reference to an MCloud package
----	----------	--------------------------------

5.1.4.17 SHAREDLL S2S_Result mcloudPacketGetAudio (MCloud * *cloudP*, MCloudPacket * *p*, short ** *sampleA*, int * *sampleN*)

Convenient function for extracting the data embedded in <audio></audio>.

Don't forget to free the memory allocated for sampleA.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>p</i>	reference to an MCloud package
out	<i>sampleA</i>	returned array of samples
out	<i>sampleN</i>	returned number of samples in array

Returns

S2S_Success if no error occurred

5.1.4.18 SHAREDLL S2S_Result mcloudPacketGetBinary (MCloud * *cP*, MCloudPacket * *p*, uint8_t ** *bA*, int * *bN*, char ** *filename*, char ** *mimetype*, int * *last*)

Convenient function for extracting the data embedded in <audio></audio>

Don't forget to free the memory allocated for sampleA.

Parameters

in	<i>cP</i>	reference to an MCloud object
in	<i>p</i>	reference to an MCloud package
out	<i>bA</i>	returned array of binary samples

out	<i>bN</i>	returned number of binary samples in array
out	<i>filename</i>	return the declared name for the file or NULL if not specified
out	<i>mimetype</i>	return the declared mimetype for the file or NULL if not specified
out	<i>Isat</i>	1 if this is the last packet for the file, 0 if more packet are incoming, -1 if not specified

Returns

S2S_Success if no error occurred

5.1.4.19 SHAREDLL S2S_Result mcloudPacketGetText (MCloud * *cloudP*, MCloudPacket * *p*, char ** *text*)

Convenient function for extracting the string embedded in <text></text>.

Don't forget to free the memory allocated for text.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>p</i>	reference to an MCloud package
out	<i>text</i>	returned text

Returns

S2S_Success if no error occurred

5.1.4.20 SHAREDLL S2S_Result mcloudPacketGetWordTokenA (MCloud * *cloudP*, MCloudPacket * *p*, MCloudWordToken ** *tokenA*, int * *tokenN*)

Convenient function for extracting a word token array embedded in <wordtokens></wordtokens>.

Don't forget to free the memory allocated for tokenA.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>p</i>	reference to an MCloud package
out	<i>tokenA</i>	returned array of word tokens
out	<i>tokenN</i>	number of word tokens returned

Returns

S2S_Success if no error occurred

5.1.4.21 SHAREDLL MCloudPacket* mcloudPacketInitFromAudio (MCloud * *cloudP*, const char * *startTime*, const char * *stopTime*, const char * *fingerPrint*, const short * *sampleA*, int *sampleN*, int *isFinal*)

Initialize a new packet from audio for sending.

Deprecated This function can be used only for PCM s16le 16KHz audio both as input and as packet content!

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>startTime</i>	human readable starting time stamp
in	<i>stopTime</i>	human readable ending time stamp
in	<i>fingerPrint</i>	finger print of the audio
in	<i>sampleA</i>	reference to an array of audio samples
in	<i>sampleN</i>	number of samples in array
in	<i>isFinal</i>	indicates whether the sample array given is the final one

Returns

reference to the created package or NULL if failed

5.1.4.22 SHAREDLL MCloudPacket* mcloudPacketInitFromBinary (MCloud * *cP*, const char * *startTime*, const char * *stopTime*, const char * *fingerPrint*, const char * *filename*, const char * *mimetype*, const uint8_t * *bytes*, int *bytesN*, int *last*)

Initialize a new packet from binary for sending.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>startTime</i>	human readable starting time stamp (set to NULL if not relevant)
in	<i>stopTime</i>	human readable ending time stamp (set to NULL if not relevant)
in	<i>fingerPrint</i>	finger print of the binary data
in	<i>filename</i>	name of the file being sent (set to NULL if not relevant)
in	<i>mimetype</i>	MIME type of the file being sent (set to NULL if not relevant)
in	<i>bytes</i>	reference to a buffer of bytes
in	<i>bytesN</i>	number of bytes
in	<i>last</i>	whether this packet is the last for the current file (i.e. following packets with the same filename should be treated as a different file)

Returns

reference to the created package or NULL if failed

5.1.4.23 SHAREDLL MCloudPacket* mcloudPacketInitFromCmGet (MCloud * *cP*, const char * *sessionId*, const char * *userId*, const char * *fingerPrint*, const char * *type*, const char * *revision*)

Initialize a new packet from a CmGet for sending.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>sessionId</i>	current sessionID which the packet belongs to
in	<i>userId</i>	current userID
in	<i>fingerPrint</i>	current finger print
in	<i>type</i>	current type
in	<i>revision</i>	current revision

Returns

reference to the created package or NULL if failed

5.1.4.24 MCloudPacket* **mcloudPacketInitFromImage** (**MCloud** * *cloudP*, **const char *** *startTime*, **const char *** *stopTime*, **const char *** *fingerPrint*, **int** *width*, **int** *height*, **const char *** *format*, **const char *** *buffer*, **int** *bufferN*)

Initialize a new packet from image for sending.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>startTime</i>	human readable starting time stamp
in	<i>stopTime</i>	human readable ending time stamp
in	<i>fingerPrint</i>	language finger print of the image (slide text language)
in	<i>width</i>	image width
in	<i>height</i>	image height
in	<i>format</i>	image format, i.e. PNG
in	<i>buffer</i>	reference to a buffer that keeps the image
in	<i>bufferN</i>	length of buffer in bytes

Returns

reference to the created package or NULL if failed

5.1.4.25 SHARED DLL MCloudPacket* mcloudPacketInitFromText (MCloud * *cloudP*, const char * *startTime*, const char * *stopTime*, unsigned int *startOffset*, unsigned int *stopOffset*, const char * *fingerPrint*, const char * *text*)

Initialize a new packet from text for sending.

startOffset and *stopOffset* do not necessarily correspond with *startTime* and *stopTime*. While *startTime* and *stopTime* define the absolute time of the packet, *startOffset* and *stopOffset* define that start and stop time stamp of e.g. the speech transcription within the packet (still relative to the beginning of the stream. For example: Packet from *startTime*=14:01:00.000 to *stopTime*=14:02:00.000 in a stream that started at *startTime*=14:00:00.000. However, speech within this packet starts 10 seconds ahead from the beginning of the packet because there is some silence to first 10 seconds and ends 1 second before the end of the packet. This means we need to define a *startOffset*=70000 (1 min since start of packet + 10 seconds of silence) and a *stopOffset*=119000.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>startTime</i>	human readable starting time stamp
in	<i>stopTime</i>	human readable ending time stamp
in	<i>startOffset</i>	start time offset in ms relative to the beginning of the stream
in	<i>stopOffset</i>	stop time offset in ms relative to the beginning of the stream
in	<i>fingerPrint</i>	finger print of the text
in	<i>text</i>	the text string

Returns

reference to the created package or NULL if failed

5.1.4.26 SHARED DLL MCloudPacket* mcloudPacketInitFromWordTokenA (MCloud * *cloudP*, const char * *startTime*, const char * *stopTime*, unsigned int *startOffset*, unsigned int *stopOffset*, const char * *fingerPrint*, MCloudWordToken * *tokenA*, int *tokenN*)

Initialize a new packet from a MCloudWordToken array for sending.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>startTime</i>	human readable starting time stamp

in	<i>stopTime</i>	human readable ending time stamp
in	<i>startOffset</i>	start time offset in ms relative to the beginning of the stream
in	<i>stopOffset</i>	stop time offset in ms relative to the beginning of the stream
in	<i>fingerPrint</i>	finger print of the text
in	<i>tokenA</i>	reference to an MCloudWordToken array
in	<i>tokenN</i>	number of word tokens in array

Returns

reference to the created package or NULL if failed

5.1.4.27 SHAREDLL S2S_Result mcloudPacketReplaceText (MCloud * cloudP, MCloudPacket * p, char * text)

Convenient function for replacing the string embedded in <text></text>.

Don't forget to free the memory allocated for text.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>p</i>	reference to an MCloud package
out	<i>text</i>	returned text

Returns

S2S_Success if no error occurred

5.1.4.28 SHAREDLL int mcloudPending (MCloud * cloudP, MCloudType queueType)

Return number of pending packages in queue.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>queueType</i>	type of queue MCloudSendingQueue, or MCloudProcessingQueue

Returns

number of pending packages

5.1.4.29 SHAREDLL S2S_Result mcloudProcessDataAsync (MCloud * cloudP, MCloudPacket * p, void * userData)

Process received packages asynchronously.

This function can be used to process packets asynchronously. The packages will be placed into an internal queue and processed in the background by calling mcloudDataCallback. Callback functions are used to forward status messages such as errors. Use mcloudBreak to stop processing pending packages. Use mcloudWaitFinish to wait until the last package has been processed. Packages are freed automatically after they have been sent. While processing, the data callback function is called for the next pending package. As soon as no more packages are pending and mcloudWaitFinish has been called, the finalize callback function is called. The error callback function may be called in case of errors, and the break callback function if mcloudBreak has been called.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>p</i>	an MCloud packet
in	<i>userData</i>	additional user data, that is passed to the callback functions

Returns

S2S_Success if no error occurs

5.1.4.30 SHAREDLL S2S_Result mcloudRequestForDisplay (MCloud * *cloudP*)

Request the display of an output stream.

By calling this function, the client requests the display of the output stream on the display server. For cancelling the request for display, the client needs to disconnect.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
----	---------------	-------------------------------

Returns

S2S_Success if no error occurs

5.1.4.31 SHAREDLL S2S_Result mcloudRequestInputStream (MCloud * *cloudP*, const char * *type*, const char * *fingerprint*, const char * *streamID*, char * *info*, int *infoN*)

Request an input stream of a client from an MCloud object.

This function has to be called in order to request a specific input stream from the MCloud such as ASR or MT results. Otherwise, the client will not receive any data. This function has to be called after the client has been connected to the MCloud.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>type</i>	data type (audio, text, image)
in	<i>fingerprint</i>	finger print of the data stream
in	<i>streamID</i>	stream identifier of the (output) stream which the requested data stream belongs to
in, out	<i>info</i>	additional information received from the MCloud
in	<i>infoN</i>	length of info field in bytes

Returns

S2S_Success if no error occurs

5.1.4.32 SHAREDLL S2S_Result mcloudSendBinaryFile (MCloud * *cP*, FILE * *f*, int *chunkSize*, char * *filename*, char * *contentType*, char * *fingerprint*)

Convenience function for sending the content of a whole file.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>f</i>	the FILE stream to send
in	<i>chunkSize</i>	bytes read from the file to be inserted in 1 packet
in	<i>filename</i>	name of the file being sent (set to NULL if not relevant)
in	<i>mimetype</i>	MIME type of the file being sent (set to NULL if not relevant)
in	<i>fingerPrint</i>	finger print of the file

Returns

S2S_Success if no error occurs

5.1.4.33 SHAREDLL S2S_Result mcloudSendBinaryFileAsync (MCloud * *cP*, FILE * *f*, int *chunkSize*, char * *filename*, char * *mimeType*, char * *fingerPrint*, void * *userData*)

Convenience function for sending the content of a whole file asynchronously.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>f</i>	the FILE stream to send
in	<i>chunkSize</i>	bytes read from the file to be inserted in 1 packet
in	<i>filename</i>	name of the file being sent (set to NULL if not relevant)
in	<i>mimetype</i>	MIME type of the file being sent (set to NULL if not relevant)
in	<i>fingerPrint</i>	finger print of the file
in	<i>userData</i>	additional user data, that is passed to the callback functions for each packet

Returns

S2S_Success if no error occurs

5.1.4.34 SHAREDLL S2S_Result mcloudSendDone (MCloud * *cloudP*)

Inform a client or a worker that there is no more data to receive.

This function should be called to inform a worker that there is no more data to receive from a client, or to inform a client, that the worker finished processing of the data received. As soon as both, worker and client have sent a done, the session is terminated.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
----	---------------	-------------------------------

Returns

S2S_Success if no error occurs

5.1.4.35 SHAREDLL S2S_Result mcloudSendError (MCloud * *cloudP*, const char * *description*)

Inform a client or a worker that an error occurred during processing.

This function can be called if an error occurs during processing either in the worker or client in order to inform all other involved components. Typically this results in a termination of the current session.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>description</i>	error description

Returns

S2S_Success if no error occurs

5.1.4.36 SHAREDLL S2S_Result mcloudSendFlush (MCloud * *cloudP*)

Inform subsequent worker to flush their output buffers.

This function should be called to inform subsequent workers finalize processing data stored in the queue and to flush their buffers.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
----	---------------	-------------------------------

Returns

S2S_Success if no error occurs

5.1.4.37 SHAREDLL S2S_Result mcloudSendPacket (MCloud * *cloudP*, MCloudPacket * *p*)

Send a packet.

This function has to be called to send a data packet to the MCloud. The data packet has to be created in advance by using the packet handling functions and has to be freed after usage.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>p</i>	an MCloud packet

Returns

S2S_Success if no error occurs

5.1.4.38 SHAREDLL S2S_Result mcloudSendPacketAsync (MCloud * *cloudP*, MCloudPacket * *p*, void * *userData*)

Send a packet asynchronously.

This function can be used for sending data packets asynchronously. The packages will be placed into an internal queue and sent in the background. Callback functions are used to forward status messages such as errors. Use `mcloudBreak` to stop sending pending packages. Use `mcloudWaitFinish` to wait until the last package has been sent. Packages are freed automatically after they have been sent. While sending, the error callback function may be called in case of errors and the break callback function if `mcloudBreak` has been called. As soon as no more packages are pending and `mcloudWaitFinish` has been called, the finalize callback function is called.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>p</i>	an MCloud packet
in	<i>userData</i>	additional user data, that is passed to the callback functions

Returns

S2S_Success if no error occurs

5.1.4.39 SHAREDLL S2S_Result mcloudSetAttr (MCloud * *cloudP*, MCloudAttribute *attr*, const void * *value*)

Set the value of an attribute.

Deprecated Set the value of an attribute.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>attr</i>	an MCloud attribute
in	<i>value</i>	reference to the value to be set

Returns

S2S_Success if no error occurs

5.1.4.40 SHAREDLL S2S_Result mcloudSetAudioEncoder (MCloud * *cp*, MCloudCodec *codec*, int *sampleRate*, int *bitRate*, int *channels*)

Set the audio codec.

Parameters

in	<i>cp</i>	reference to an MCloud object
in	<i>codec</i>	MCloud codec used to transmit/receive data to/from the Mediator
in	<i>sampleRate</i>	sample rate used to transmit/receive data to/from the Mediator
in	<i>bitRate</i>	bit rate used to transmit/receive data to/from the Mediator
in	<i>channels</i>	channels used to transmit/receive data to/from the Mediator

Returns

S2S_Success if no error occurs

5.1.4.41 SHAREDLL S2S_Result mcloudSetAudioEncoder2 (MCloud * *cp*, char * *codec*, int *sampleRate*, int *bitRate*, int *channels*)

Set the audio codec.

Parameters

in	<i>cp</i>	reference to an MCloud object
in	<i>codec</i>	codec (string form) used to transmit/receive data to/from the Mediator

in	<i>sampleRate</i>	sample rate used to transmit/receive data to/from the Mediator
in	<i>bitRate</i>	bit rate used to transmit/receive data to/from the Mediator
in	<i>channels</i>	channels used to transmit/receive data to/from the Mediator

Returns

S2S_Success if no error occurs

5.1.4.42 SHAREDLL void mcloudSetBreakCallback (MCloud * cloudP, MCloudType queueType, MCloudCallbackFct * callback, void * userData)

Set a break callback function.

This function is called when the worker should stop the processing as soon as possible. This callback is available for both the sending and the processing queue.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>queueType</i>	type of queue MCloudSendingQueue, or MCloudProcessingQueue
in	<i>callback</i>	reference to the function that has to be called
in	<i>userData</i>	reference to some data that is passed to the callback function

5.1.4.43 SHAREDLL void mcloudSetCustomizationCallback (MCloud * cP, MCloudPacketCallbackFct * callback, void * userData)

Set a customization callback function.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>callback</i>	reference to the function that has to be called
in	<i>userData</i>	reference to some data that is passed to the callback function

5.1.4.44 SHAREDLL void mcloudSetDataCallback (MCloud * cloudP, MCloudPacketCallbackFct * callback)

Set a data callback function.

This function is called for each incoming data package in a serial way, i.e. after a package has been processed it is called again if more packages are pending. Note that for this function no userData is given at the time of the set of the callback function. Instead, the userData is given per packet with mcloudProcessDataAsync or mcloudSendAsync. This callback is available for the processing queue only.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>callback</i>	reference to the function that has to be called

5.1.4.45 SHAREDLL void mcloudSetErrorCallback (MCloud * cloudP, MCloudType queueType, MCloudCallbackFct * callback, void * userData)

Set an error callback function.

This function is called as soon as an error occurs in the asynchronous processing. This callback is available for both the sending and the processing queue.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>queueType</i>	type of queue MCloudSendingQueue, or MCloudProcessingQueue
in	<i>callback</i>	reference to the function that has to be called
in	<i>userData</i>	reference to some data that is passed to the callback function

5.1.4.46 SHAREDLL void mcloudSetFinalizeCallback (MCloud * *cloudP*, MCloudCallbackFct * *callback*, void * *userData*)

Set a finalize callback function.

This function is called as soon as the processing of packets should be finalized, i.e. no more packets will follow and the worker should output the final results after all pending packets have been processed. This callback is only available for the processing queue.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>callback</i>	reference to the function that has to be called
in	<i>userData</i>	reference to some data that is passed to the callback function

5.1.4.47 SHAREDLL void mcloudSetInitCallback (MCloud * *cloudP*, MCloudPacketCallbackFct * *callback*, void * *userData*)

Set an init callback function.

This function is called as soon as an incoming service request has been accepted by the worker, i.e. in mcloudWaitForClient. The packet containing the service description is passed to the init callback function as argument. This callback is only available for the processing queue.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>callback</i>	reference to the function that has to be called
in	<i>userData</i>	reference to some data that is passed to the callback function

5.1.4.48 SHAREDLL S2S_Result mcloudWaitFinish (MCloud * *cloudP*, MCloudType *queueType*, int *done*)

Wait until all pending packages have been processed/ sent.

This function can be used to wait until all pending packages have been processed or sent in the queue specified.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
in	<i>queueType</i>	type of queue MCloudSendingQueue, or MCloudProcessingQueue
in	<i>done</i>	if set to 1, indicates that processing of the request has been completed

Returns

S2S_Success if no error occurs

5.1.4.49 SHAREDLL S2S_Result mcloudWaitForClient (MCloud * *cloudP*, char ** *streamID*)

Wait for a service request to process.

This function has to be called after the worker has been successfully connected to the MCloud in order to wait for an incoming service request to process.

Parameters

in	<i>cloudP</i>	reference to an MCloud object
out	<i>streamID</i>	ID of input stream

Returns

S2S_Success if no error occurs

5.1.4.50 SHAREDLL MCloudWordToken* mcloudWordTokenArrayCreate (int *n*)

Creates an array of MCloudWordTokens.

Parameters

in	<i>n</i>	number of elements
----	----------	--------------------

Returns

reference to the created token array or NULL if failed

5.1.4.51 SHAREDLL void mcloudWordTokenArrayFree (MCloudWordToken * *tokenA*, int *n*)

Free an array of MCloudWordTokens.

Parameters

in	<i>tokenA</i>	reference to an MCloudWorkToken array
in	<i>n</i>	number of elements

5.1.4.52 SHAREDLL char* url_decode (const char * *str*)

Convenient function that returns a url-decoded version of a string.

Don't forget to free() the returned string after use

Parameters

in	<i>str</i>	string to be decoded
----	------------	----------------------

Returns

url-decoded version of str

5.1.4.53 SHAREDLL char* url_encode (const char * *str*)

Convenient function that returns a url-encoded version of a string.

Don't forget to free() the returned string after use

Parameters

in	<i>str</i>	string to be encoded
----	------------	----------------------

Returns

url-encoded version of str

Index

- base64_decode
 - MCloud.h, [16](#)
- base64_encode
 - MCloud.h, [16](#)
- cmType
 - MCloudPacket_S, [8](#)
- confidence
 - MCloudWordToken_S, [9](#)
- creator
 - MCloudPacket_S, [8](#)
- dataType
 - MCloudPacket_S, [8](#)
- doc
 - MCloudPacket_S, [8](#)
- fingerPrint
 - MCloudPacket_S, [8](#)
- index
 - MCloudWordToken_S, [9](#)
- internal
 - MCloudWordToken_S, [9](#)
- isFiller
 - MCloudWordToken_S, [10](#)
- MCloud.h
 - MCloudA_iBitRate, [15](#)
 - MCloudA_iChannelIN, [15](#)
 - MCloudA_iSampleRate, [15](#)
 - MCloudA_iSampleSize, [15](#)
 - MCloudA_sAudioCodec, [15](#)
 - MCloudAC_FLAC, [15](#)
 - MCloudAC_OPUS, [15](#)
 - MCloudAC_PCM, [15](#)
 - MCloudAC_SPEEX, [15](#)
 - MCloudAC_UNK, [15](#)
 - MCloudAudio, [16](#)
 - MCloudBinary, [16](#)
 - MCloudCustomization, [16](#)
 - MCloudData, [16](#)
 - MCloudDone, [16](#)
 - MCloudError, [16](#)
 - MCloudFlush, [16](#)
 - MCloudImage, [16](#)
 - MCloudMixed, [16](#)
 - MCloudModeClient, [16](#)
 - MCloudModeWorker, [16](#)
 - MCloudProcessingQueue, [16](#)
 - MCloudReset, [16](#)
 - MCloudSendingQueue, [16](#)
 - MCloudText, [16](#)
 - S2S_Error, [16](#)
 - S2S_Success, [16](#)
 - MCloudA_iBitRate
 - MCloud.h, [15](#)
 - MCloudA_iChannelIN
 - MCloud.h, [15](#)
 - MCloudA_iSampleRate
 - MCloud.h, [15](#)
 - MCloudA_iSampleSize
 - MCloud.h, [15](#)
 - MCloudA_sAudioCodec
 - MCloud.h, [15](#)
 - MCloudAC_FLAC
 - MCloud.h, [15](#)
 - MCloudAC_OPUS
 - MCloud.h, [15](#)
 - MCloudAC_PCM
 - MCloud.h, [15](#)
 - MCloudAC_SPEEX
 - MCloud.h, [15](#)
 - MCloudAC_UNK
 - MCloud.h, [15](#)
 - MCloudAudio
 - MCloud.h, [16](#)
 - MCloudBinary
 - MCloud.h, [16](#)
 - MCloudCustomization
 - MCloud.h, [16](#)
 - MCloudData
 - MCloud.h, [16](#)
 - MCloudDone
 - MCloud.h, [16](#)
 - MCloudError
 - MCloud.h, [16](#)
 - MCloudFlush
 - MCloud.h, [16](#)
 - MCloudImage
 - MCloud.h, [16](#)
 - MCloudMixed
 - MCloud.h, [16](#)
 - MCloudModeClient
 - MCloud.h, [16](#)
 - MCloudModeWorker
 - MCloud.h, [16](#)
 - MCloudProcessingQueue
 - MCloud.h, [16](#)
 - MCloudReset

- MCloud.h, 16
- MCloudSendingQueue
 - MCloud.h, 16
- MCloudText
 - MCloud.h, 16
- MCloud
 - MCloud.h, 14
- MCloud.h, 11
 - base64_decode, 16
 - base64_encode, 16
 - MCloud, 14
 - MCloudAttribute, 15
 - MCloudCallbackFct, 14
 - MCloudCodec, 15
 - MCloudPacket, 15
 - MCloudPacketCallbackFct, 15
 - MCloudType, 15
 - MCloudWordToken, 15
 - mcloudAddFlowDescription, 17
 - mcloudAddFlowDescription2, 17
 - mcloudAddService, 18
 - mcloudAnnounceOutputStream, 18
 - mcloudBreak, 18
 - mcloudConnect, 19
 - mcloudCreate, 19
 - mcloudDisconnect, 19
 - mcloudERROR, 14
 - mcloudFree, 19
 - mcloudGetAttr, 20
 - mcloudGetNextPacket, 20
 - mcloudINFO, 14
 - mcloudMsgHandler, 20
 - mcloudPacketAddAudio, 20
 - mcloudPacketDeinit, 21
 - mcloudPacketGetAudio, 21
 - mcloudPacketGetBinary, 21
 - mcloudPacketGetText, 22
 - mcloudPacketGetWordTokenA, 22
 - mcloudPacketInitFromAudio, 22
 - mcloudPacketInitFromBinary, 23
 - mcloudPacketInitFromCmGet, 23
 - mcloudPacketInitFromImage, 23
 - mcloudPacketInitFromText, 25
 - mcloudPacketInitFromWordTokenA, 25
 - mcloudPacketReplaceText, 26
 - mcloudPending, 26
 - mcloudProcessDataAsync, 26
 - mcloudRequestForDisplay, 27
 - mcloudRequestInputStream, 27
 - mcloudSendBinaryFile, 27
 - mcloudSendBinaryFileAsync, 28
 - mcloudSendDone, 28
 - mcloudSendError, 28
 - mcloudSendFlush, 29
 - mcloudSendPacket, 29
 - mcloudSendPacketAsync, 29
 - mcloudSetAttr, 30
 - mcloudSetAudioEncoder, 30
 - mcloudSetAudioEncoder2, 30
 - mcloudSetBreakCallback, 31
 - mcloudSetCustomizationCallback, 31
 - mcloudSetDataCallback, 31
 - mcloudSetErrorCallback, 31
 - mcloudSetFinalizeCallback, 32
 - mcloudSetInitCallback, 32
 - mcloudWARN, 14
 - mcloudWaitFinish, 32
 - mcloudWaitForClient, 32
 - mcloudWordTokenArrayCreate, 34
 - mcloudWordTokenArrayFree, 34
 - S2S_Result, 16
 - SHARED DLL, 14
 - url_decode, 34
 - url_encode, 34
- MCloudAttribute
 - MCloud.h, 15
- MCloudCallbackFct
 - MCloud.h, 14
- MCloudCodec
 - MCloud.h, 15
- MCloudPacket
 - MCloud.h, 15
- MCloudPacket_S, 7
 - cmType, 8
 - creator, 8
 - dataType, 8
 - doc, 8
 - fingerprint, 8
 - packetType, 8
 - revision, 8
 - sessionId, 8
 - start, 8
 - startOffset, 8
 - statusDescription, 8
 - stop, 8
 - stopOffset, 8
 - streamID, 9
 - userID, 9
 - xmlString, 9
- MCloudPacketCallbackFct
 - MCloud.h, 15
- MCloudType
 - MCloud.h, 15
- MCloudWordToken
 - MCloud.h, 15
- MCloudWordToken_S, 9
 - confidence, 9
 - index, 9
 - internal, 9
 - isFiller, 10
 - spoken, 10
 - startTime, 10
 - stopTime, 10
 - written, 10
- mcloudAddFlowDescription
 - MCloud.h, 17

- mcloudAddFlowDescription2
 - MCloud.h, [17](#)
- mcloudAddService
 - MCloud.h, [18](#)
- mcloudAnnounceOutputStream
 - MCloud.h, [18](#)
- mcloudBreak
 - MCloud.h, [18](#)
- mcloudConnect
 - MCloud.h, [19](#)
- mcloudCreate
 - MCloud.h, [19](#)
- mcloudDisconnect
 - MCloud.h, [19](#)
- mcloudERROR
 - MCloud.h, [14](#)
- mcloudFree
 - MCloud.h, [19](#)
- mcloudGetAttr
 - MCloud.h, [20](#)
- mcloudGetNextPacket
 - MCloud.h, [20](#)
- mcloudINFO
 - MCloud.h, [14](#)
- mcloudMsgHandler
 - MCloud.h, [20](#)
- mcloudPacketAddAudio
 - MCloud.h, [20](#)
- mcloudPacketDeinit
 - MCloud.h, [21](#)
- mcloudPacketGetAudio
 - MCloud.h, [21](#)
- mcloudPacketGetBinary
 - MCloud.h, [21](#)
- mcloudPacketGetText
 - MCloud.h, [22](#)
- mcloudPacketGetWordTokenA
 - MCloud.h, [22](#)
- mcloudPacketInitFromAudio
 - MCloud.h, [22](#)
- mcloudPacketInitFromBinary
 - MCloud.h, [23](#)
- mcloudPacketInitFromCmGet
 - MCloud.h, [23](#)
- mcloudPacketInitFromImage
 - MCloud.h, [23](#)
- mcloudPacketInitFromText
 - MCloud.h, [25](#)
- mcloudPacketInitFromWordTokenA
 - MCloud.h, [25](#)
- mcloudPacketReplaceText
 - MCloud.h, [26](#)
- mcloudPending
 - MCloud.h, [26](#)
- mcloudProcessDataAsync
 - MCloud.h, [26](#)
- mcloudRequestForDisplay
 - MCloud.h, [27](#)
- mcloudRequestInputStream
 - MCloud.h, [27](#)
- mcloudSendBinaryFile
 - MCloud.h, [27](#)
- mcloudSendBinaryFileAsync
 - MCloud.h, [28](#)
- mcloudSendDone
 - MCloud.h, [28](#)
- mcloudSendError
 - MCloud.h, [28](#)
- mcloudSendFlush
 - MCloud.h, [29](#)
- mcloudSendPacket
 - MCloud.h, [29](#)
- mcloudSendPacketAsync
 - MCloud.h, [29](#)
- mcloudSetAttr
 - MCloud.h, [30](#)
- mcloudSetAudioEncoder
 - MCloud.h, [30](#)
- mcloudSetAudioEncoder2
 - MCloud.h, [30](#)
- mcloudSetBreakCallback
 - MCloud.h, [31](#)
- mcloudSetCustomizationCallback
 - MCloud.h, [31](#)
- mcloudSetDataCallback
 - MCloud.h, [31](#)
- mcloudSetErrorCallback
 - MCloud.h, [31](#)
- mcloudSetFinalizeCallback
 - MCloud.h, [32](#)
- mcloudSetInitCallback
 - MCloud.h, [32](#)
- mcloudWARN
 - MCloud.h, [14](#)
- mcloudWaitFinish
 - MCloud.h, [32](#)
- mcloudWaitForClient
 - MCloud.h, [32](#)
- mcloudWordTokenArrayCreate
 - MCloud.h, [34](#)
- mcloudWordTokenArrayFree
 - MCloud.h, [34](#)
- packetType
 - MCloudPacket_S, [8](#)
- revision
 - MCloudPacket_S, [8](#)
- S2S_Error
 - MCloud.h, [16](#)
- S2S_Success
 - MCloud.h, [16](#)
- S2S_Result
 - MCloud.h, [16](#)
- SHAREDLL
 - MCloud.h, [14](#)

sessionID
 MCloudPacket_S, 8

spoken
 MCloudWordToken_S, 10

start
 MCloudPacket_S, 8

startOffset
 MCloudPacket_S, 8

startTime
 MCloudWordToken_S, 10

statusDescription
 MCloudPacket_S, 8

stop
 MCloudPacket_S, 8

stopOffset
 MCloudPacket_S, 8

stopTime
 MCloudWordToken_S, 10

streamID
 MCloudPacket_S, 9

url_decode
 MCloud.h, 34

url_encode
 MCloud.h, 34

userID
 MCloudPacket_S, 9

written
 MCloudWordToken_S, 10

xmlString
 MCloudPacket_S, 9