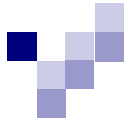


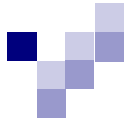


# Locosto Audio Entity Introduction



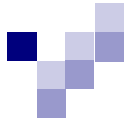
# Overview

The AUDIO services provided by the AUDIO entity can be accessed by any entity running on the mobile.



# Audio API

All the services provided by the audio entity can be accessed via direct function call.



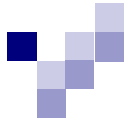
# Audio API

All the functions return an immediate value, providing information on the success or the failure of the function call. In some (most of the) cases, extra processing time is needed to perform the action requested when calling the function. In this case, the function is exit and later on, one or several EVENTS are sent back by the AUDIO entity.



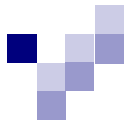
# Audio API

The AUDIO entity use the EVENT format and the return path method defined in Riviera Environment. Basically, in order to send information back, the AUDIO entity sends EVENTS to the MMI.

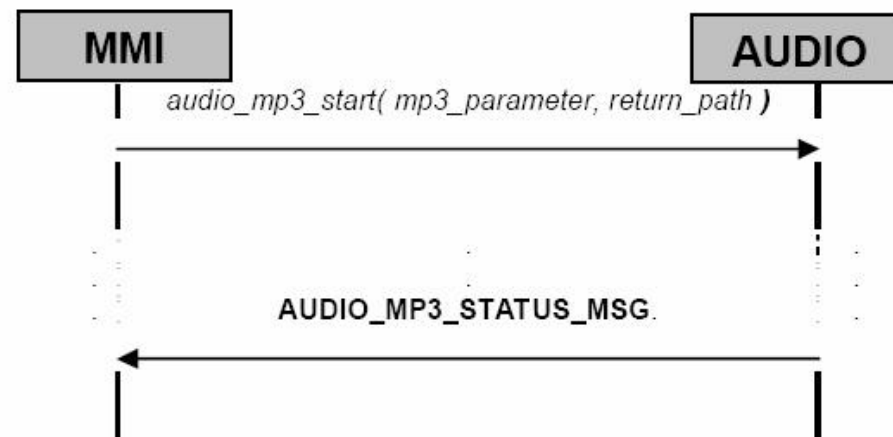


# Audio API

MMI have two ways to get access to the EVENTS: Call back functions or message posted in its mailbox.



# Audio API

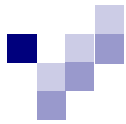




# Audio-Modem Incompatibilities

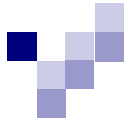
	GSM		Dedicated Non-AMR					Dedicated AMR		GPRS		GSM->GPRS	GPRS->GSM
	Idle	SMS	FACCH	Ringer	Speech	TCH/Data	IDS	Ringer	Speech	Idle	Transfer		
Keybeep	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tones	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Melody_E1	✓	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓	✓	✓
Melody_E2	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓
Voice Memo	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
AMR MMS	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓
Speech Reco	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
TTY	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
FIR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
AEC	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Audio Mode	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TIDE/DCO	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓





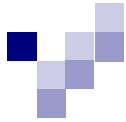
# Audio Task Compatibilities

	Keybeep	Tones	Melody E1	Melody E2	Voice Memo	AMRMMS	Speech Reco	TTY	FIR	AEC	Audio Mode	TIDEDCO
Keybeep	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tones	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Melody E1	✓	✗	see (*)	✗	✗	✗	✗	✗	✓	✓	✓	✓
Melody E2	✓	✗	✗	see (*)	✗	✗	✗	✗	✓	✓	✓	✓
Voice Memo	✓	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓
AMRMMS	✓	✓	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓
Speech Reco	✓	✓	✗	✗	✗	✗	✓	✗	✓	✓	✓	✓
TTY	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓
FIR	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
AEC	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓
Audio Mode	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
TIDEDCO	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗



# Full Access Family

*T\_AUDIO\_RET audio\_full\_access\_write (*  
*T\_AUDIO\_FULL\_ACCESS\_WRITE*  
*\*p\_parameter,*  
*T\_RV\_RETURN*  
*return\_path)*



# Full Access Family

**T\_AUDIO\_FULL\_ACCESS\_WRITE**

typedef struct {

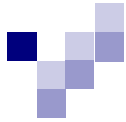
UINT8 variable\_identifier;

// identifier of the variable to configure

void \*data;

// data corresponding to the variable to set

}T\_AUDIO\_FULL\_ACCESS\_WRITE;



# Full Access Family

Example:

AUDIO_MICROPHONE_MODE (1)	INT8
AUDIO_MICROPHONE_GAIN (2)	INT8
AUDIO_MICROPHONE_EXTRA_GAIN (1)	INT8
AUDIO_MICROPHONE_OUTPUT_BIAS (1)	INT8
AUDIO_MICROPHONE_FIR	typedef struct { UINT16 coefficient[31]; } T AUDIO_FIR_COEF;
AUDIO_MICROPHONE_ANR (6)	T AUDIO_ANR_CFG (see <a href="#">28.6.2.1.1.3.1</a> )
AUDIO_MICROPHONE_ES (6)	T AUDIO_ES_CFG (see <a href="#">28.6.2.1.1.3.2</a> )
AUDIO_SPEAKER_MODE (1)	INT8
AUDIO_SPEAKER_GAIN (2)	INT8
AUDIO_SPEAKER_EXTRA_GAIN (1)	INT8
AUDIO_SPEAKER_FILTER (1)	INT8