



Intel Accelerated DSP Software API Reference Manual

for Software Version 1.1

Automatically generated from sources, December 01, 2009.

Document Number: 323036-001

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's Web Site (<http://www.intel.com/>).

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Inside, Core Inside, i960, Intel, the Intel logo, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, the Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel Sponsors of Tomorrow., the Intel Sponsors of Tomorrow. logo, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, InTru, the InTru logo, InTru soundmark, Itanium, Itanium Inside, MCS, MMX, Pentium, Pentium Inside, skool, the skool logo, Sound Mark, The Journey Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others

Copyright © 2009, Intel Corporation. All rights reserved.

Table of Contents

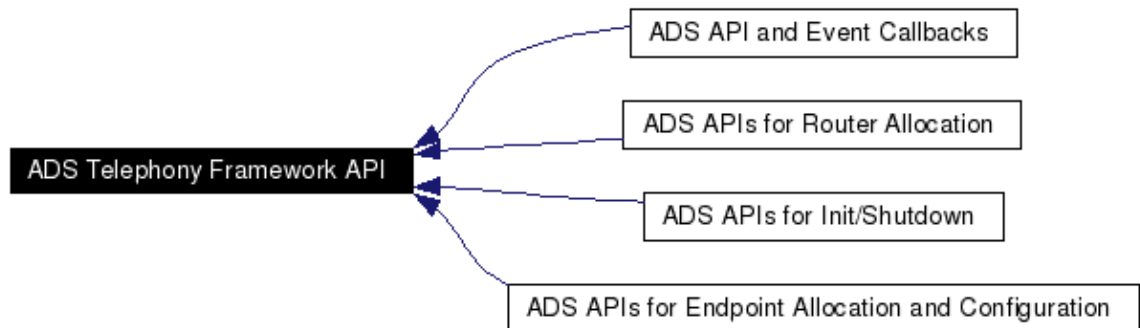
1 ADS Telephony Framework API.....	1
1.1 Modules.....	1
1.2 Data Structures.....	1
1.3 Enumerations.....	1
1.4 Detailed Description.....	2
1.5 Data Structure Documentation.....	2
1.5.1 AdsTelInit_t Struct Reference.....	2
1.6 Enumeration Type Documentation.....	3
2 ADS APIs for Init/Shutdown [ADS Telephony Framework API].....	5
2.1 Functions.....	5
2.2 Detailed Description.....	5
2.3 Function Documentation.....	5
3 ADS APIs for Endpoint Allocation and Configuration [ADS Telephony Framework API].....	8
3.1 Data Structures.....	8
3.2 Defines.....	8
3.3 Typedefs.....	18
3.4 Enumerations.....	18
3.5 Functions.....	19
3.6 Detailed Description.....	20
3.7 Data Structure Documentation.....	20
3.7.1 AdsTelMediaDesc_s Struct Reference.....	20
3.7.2 AdsTelToneSeg_t Struct Reference.....	21
3.7.3 AdsTelToneInfo_t Struct Reference.....	23
3.8 Define Documentation.....	24
3.9 Typedef Documentation.....	71
3.10 Enumeration Type Documentation.....	72
3.11 Function Documentation.....	74
4 ADS APIs for Router Allocation [ADS Telephony Framework API].....	94
4.1 Typedefs.....	94
4.2 Enumerations.....	94
4.3 Functions.....	94
4.4 Detailed Description.....	94
4.5 Typedef Documentation.....	95
4.6 Enumeration Type Documentation.....	95
4.7 Function Documentation.....	96
5 ADS API and Event Callbacks [ADS Telephony Framework API].....	99
5.1 Data Structures.....	99
5.2 Defines.....	99
5.3 Typedefs.....	99
5.4 Enumerations.....	100
5.5 Detailed Description.....	100
5.6 Data Structure Documentation.....	100
5.6.1 AdsTelRetBuf_t Struct Reference.....	100
5.6.2 AdsTelPacket_t Struct Reference.....	101
5.6.3 AdsTelPCMPacket_t Struct Reference.....	102
5.6.4 AdsTelRetAck_s Struct Reference.....	103
5.6.5 AdsTelRetTg_s Struct Reference.....	104
5.6.6 AdsTelRetTdFsk_s Struct Reference.....	105
5.6.7 AdsTelRetTd_s Struct Reference.....	106
5.6.8 AdsTelRetEv_s Struct Reference.....	107
5.6.9 AdsTelRetGetParams_s Struct Reference.....	108

Table of Contents

5 ADS API and Event Callbacks [ADS Telephony Framework API]	
5.6.10 AdsTelRetPlay_s Struct Reference.....	109
5.7 Define Documentation.....	110
5.8 Typedef Documentation.....	110
5.9 Enumeration Type Documentation.....	113

1 ADS Telephony Framework API

Collaboration diagram for ADS Telephony Framework API:



Public API for the ADS Telephony Framework.

1.1 Modules

- **ADS APIs for Init/Shutdown**
Initialization and shutdown functions for the ADS Telephony Framework.
- **ADS APIs for Endpoint Allocation and Configuration**
Pipeline instance allocation and deallocation functions for the ADS Telephony Framework.
- **ADS APIs for Router Allocation**
Router instance allocation and deallocation functions for the ADS Telephony Framework.
- **ADS API and Event Callbacks**
Functions and type definitions of the ADS Telephony Framework for API and events callbacks.

1.2 Data Structures

- struct **AdsTelInit_t**
This structure contains the Initialization parameters for ADS.

1.3 Enumerations

- enum **AdsTelStatus_t** {
 ADS_TEL_STATUS_SUCCESS,
 ADS_TEL_STATUS_FAIL,
 ADS_TEL_STATUS_INVALID_PARAM,
 ADS_TEL_STATUS_EVENT,
 ADS_TEL_STATUS_GET_PARAMS,
 ADS_TEL_STATUS_TG_CMPLT,
 ADS_TEL_STATUS_TD_RCV_CMPLT,
 ADS_TEL_STATUS_TD_RCV_FSK_CMPLT,
 ADS_TEL_STATUS_PLY_CMPLT,
 ADS_TEL_STATUS_UNDERFLOW,
 ADS_TEL_STATUS_TONE,
 ADS_TEL_STATUS_AUDIO,
 ADS_TEL_STATUS_AUDIO_TONE,
 ADS_TEL_STATUS_INVALID_FORMAT,
 ADS_TEL_STATUS_DELIMITER

1.3 Enumerations

}

AdsTel return type.

1.4 Detailed Description

Public API for the ADS Telephony Framework.

Purpose:

1.5 Data Structure Documentation

1.5.1 AdsTellnit_t Struct Reference

This structure contains the Initialization parameters for ADS.

1.5.1.1 Data Fields

- unsigned int **numEp**
The number of endpoints that will be supported by the framework.
- unsigned int **geo**
The geographical location that the framework is configured for.
- unsigned int **numRTTs**
The number of real-time threads that the system will use.
- unsigned int **ctrlPrior**
The priority of the control thread within the framework.
- unsigned int **rttPrior**
The priority of the real time threads within the framework.
- **adsTelFwCb_t fwCb**
The framework callback function.

1.5.1.2 Detailed Description

This structure contains the Initialization parameters for ADS.

Description:

This structure contains the initialization parameters for the ADS framework **adsTellnit()**.

Purpose:

Initialization paramters for ADS.

See also:

None.

1.5.1.3 Field Documentation

unsigned int **AdsTellnit_t::numEp**

The number of endpoints that will be supported by the framework.

The number of routers supported by the framework will be half the number of endpoints supported.

unsigned int **AdsTellnit_t::geo**

1.5.1 AdsTelInit_t Struct Reference

The geographical location that the framework is configured for.

0 = Peoples Republic Of China 1 = Japan 2 = USA

unsigned int **AdsTelInit_t::numRTTs**

The number of real-time threads that the system will use.

unsigned int **AdsTelInit_t::ctrlPrior**

The priority of the control thread within the framework.

The allowable range is 1-99 with higher numbers being higher priority

unsigned int **AdsTelInit_t::rttPrior**

The priority of the real time threads within the framework.

The allowable range is 1-99 with higher numbers being higher priority

adsTelFwCb_t AdsTelInit_t::fwCb

The framework callback function.

1.6 Enumeration Type Documentation

enum **AdsTelStatus_t**

AdsTel return type.

Description:

return types for adsTel API calls and **adsTelFwCb_t**.

Enumerator:

<i>ADS_TEL_STATUS_SUCCESS</i>	successful return status.
<i>ADS_TEL_STATUS_FAIL</i>	failure return status.
<i>ADS_TEL_STATUS_INVALID_PARAM</i>	invalid parameter return status.
<i>ADS_TEL_STATUS_EVENT</i>	event return status.
<i>ADS_TEL_STATUS_GET_PARAMS</i>	See adsTelFwCb_t .
<i>ADS_TEL_STATUS_TG_CMPLT</i>	adsTelFwCb_t return status for a call to adsTelEpCompParamsGet() .
<i>ADS_TEL_STATUS_TD_RCV_CMPLT</i>	adsTelFwCb_t return status for a call to adsTelEpTonePlay() or adsTelEpFskPlay() .
<i>ADS_TEL_STATUS_TD_RCV_FSK_CMPLT</i>	adsTelFwCb_t return status for a call to adsTelEpToneRcv() .
<i>ADS_TEL_STATUS_PLY_CMPLT</i>	adsTelFwCb_t return status for a call to adsTelEpFskRcv() .
<i>ADS_TEL_STATUS_UNDERFLOW</i>	adsTelFwCb_t return status for a call to adsTelEpPlayAudio() .
<i>ADS_TEL_STATUS_TONE</i>	adsTelEpDataCb_t status to be returned when no RTP packet is available.
<i>ADS_TEL_STATUS_AUDIO</i>	adsTelEpDataCb_t status to be returned when only a RFC2833 tone packet is available.
	adsTelEpDataCb_t status to be returned when an audio packet is available.

1.6 Enumeration Type Documentation

ADS_TEL_STATUS_AUDIO_TONE

adsTelEpDataCb_t status to be returned when both an audio packet and a RFC2833 tone packet is available.

ADS_TEL_STATUS_INVALID_FORMAT

invalid wav file format status.

ADS_TEL_STATUS_DELIMITER

delimiter for error checks.

2 ADS APIs for Init/Shutdown

[ADS Telephony Framework API]

Collaboration diagram for ADS APIs for Init/Shutdown:



Initialization and shutdown functions for the ADS Telephony Framework.

2.1 Functions

- **AdsTelStatus_t** **adsTelInit** (**AdsTelInit_t** adsTelInitParams)
 - **AdsTelStatus_t** **adsTelUnInit** (void)
 - void **adsTelFwTrigger** (void)
-

2.2 Detailed Description

Initialization and shutdown functions for the ADS Telephony Framework.

Purpose:

The functions in this group are responsible for all initialization and shutdown for the ADS Telephony Framework.

2.3 Function Documentation

AdsTelStatus_t **adsTelInit** (**AdsTelInit_t** *adsTelInitParams*)

Description:

This function initializes the ADS Telephony Framework and allocates any memory and threads that are needed.

Context:

Calling function thread.

Assumptions:

None.

Side-Effects:

None.

Parameters:

adsTelInitParams [IN] - Initialization parameters.

Blocking:

Yes.

Reentrant:

No.

2.3 Function Documentation

Thread-safe:

Yes.

Return values:

None.

Precondition:

None.

Postcondition:

None.

Note:

None.

See also:

None.

AdsTelStatus_t adsTelUnInit (void)

Description:

This function uninitializes the ADS Telephony Framework and any associated memory and threads.

Context:

Calling function thread.

Assumptions:

None.

Side-Effects:

None.

Blocking:

Yes.

Reentrant:

No.

Thread-safe:

Yes.

Return values:

None.

Precondition:

None.

Postcondition:

None.

Note:

None.

See also:

None.

2.3 Function Documentation

```
void adsTelFwTrigger ( void )
```

Description:

This function triggers the framework data processing threads.

Context:

Calling function thread.

Assumptions:

None.

Side-Effects:

None.

Blocking:

Yes.

Reentrant:

Yes.

Thread-safe:

Yes.

Return values:

None.

Precondition:

None.

Postcondition:

None.

Note:

None.

See also:

None.

3 ADS APIs for Endpoint Allocation and Configuration

[ADS Telephony Framework API]

Collaboration diagram for ADS APIs for Endpoint Allocation and Configuration:



Pipeline instance allocation and deallocation functions for the ADS Telephony Framework.

3.1 Data Structures

- struct **AdsTelMediaDesc_s**
Audio segment to play.
- struct **AdsTelToneSeg_t**
Tone segment structure.
- struct **AdsTelToneInfo_t**
Tone Information structure.

3.2 Defines

- #define **ADS_TEL_MAX_TONES**
Maximum number of tones that can be sent to or detected by a tone generator component.
- #define **ADS_TEL_MAX_TONE_ID**
Maximum tone ID.
- #define **ADS_TEL_MAX_FSK_BYTES**
Maximum number of FSK bytes that can be sent to or detected by a tone generator component.
- #define **ADS_TEL_MAX_PARAMS**
Maximum number of parameters that can be set or retrieved for any component.
- #define **ADS_TEL_MEDIA_SEG_EOP**
Identifier for the last segment to play.
- #define **ADS_TEL_PLY_MAX_SEGS**
Maximum number of segments that can be sent to a player component.
- #define **ADS_TEL_TG_LASTSEG**
Tone Generator last segment identifier.
- #define **ADS_TEL_TG_MODE_OVERWRITE**
Tone Generator segment output mode type.
- #define **ADS_TEL_TG_MODE_MIX**
Tone Generator segment output mode type.
- #define **ADS_TEL_TG_SINGLETONE**
Tone Generator segment tone type.
- #define **ADS_TEL_TG_DUALTONE**
Tone Generator segment tone type.
- #define **ADS_TEL_TG_AMTONE**
Tone Generator segment tone type.
- #define **ADS_TEL_TD_QUAL_CRITERIA_LOOSE**
Tone Detector qualification criteria type.
- #define **ADS_TEL_TD_QUAL_CRITERIA_MEDIUM**
Tone Detector qualification criteria type.
- #define **ADS_TEL_TD_QUAL_CRITERIA_TIGHT**
Tone Detector qualification criteria type.

3.2 Defines

- #define **ADS_TEL_TD_TONEATTR_EVTON**
Tone Detector event report attribute type.
- #define **ADS_TEL_TD_TONEATTR_EVTOFF**
Tone Detector event report attribute type.
- #define **ADS_TEL_TD_TONEATTR_EVTBOTH**
Tone Detector event report attribute type.
- #define **ADS_TEL_TD_SINGLETONE**
Tone Detector tone type.
- #define **ADS_TEL_TD_DUALTONE**
Tone Detector tone type.
- #define **RFC_TID_DTMF_0**
RFC DTMF 0.
- #define **RFC_TID_DTMF_1**
RFC DTMF 1.
- #define **RFC_TID_DTMF_2**
RFC DTMF 2.
- #define **RFC_TID_DTMF_3**
RFC DTMF 3.
- #define **RFC_TID_DTMF_4**
RFC DTMF 4.
- #define **RFC_TID_DTMF_5**
RFC DTMF 5.
- #define **RFC_TID_DTMF_6**
RFC DTMF 6.
- #define **RFC_TID_DTMF_7**
RFC DTMF 7.
- #define **RFC_TID_DTMF_8**
RFC DTMF 8.
- #define **RFC_TID_DTMF_9**
RFC DTMF 9.
- #define **RFC_TID_DTMF_STAR**
RFC DTMF 10.
- #define **RFC_TID_DTMF_POUND**
RFC DTMF 11.
- #define **RFC_TID_DTMF_A**
RFC DTMF 12.
- #define **RFC_TID_DTMF_B**
RFC DTMF 13.
- #define **RFC_TID_DTMF_C**
Tone Detector tone type.
- #define **RFC_TID_DTMF_D**
RFC DTMF 15.
- #define **RFC_TID_FAX_CED**
Fax CED Tone.
- #define **RFC_TID_FAX_CNG**
Fax CNG Tone.
- #define **RFC_TID_FAX_V21**
Fax V21 Tone.
- #define **RFC_CPT_START**
Call Progress Tone Base.
- #define **RFC_TID_DIAL**
RFC CPT - Dial Tone.
- #define **RFC_TID_PBX_DIAL**
RFC CPT - PBX Dial Tone.
- #define **RFC_TID_SP_DIAL**
RFC CPT - Special Dial Tone.

3.2 Defines

- #define **RFC_TID_2ND_DIAL**
RFC CPT - 2nd Dial Tone.
- #define **RFC_TID_RING**
RFC CPT - Ring Tone.
- #define **RFC_TID_SP_RING**
RFC CPT - Special Ring Tone.
- #define **RFC_TID_BUSY**
RFC CPT - Busy Tone.
- #define **RFC_TID_CONGESTION**
RFC CPT - Congestion Tone.
- #define **RFC_TID_SP_INFO**
RFC CPT - Special Information Tone.
- #define **RFC_TID_COMFORT**
RFC CPT - Comfort Tone.
- #define **RFC_TID_HOLD**
RFC CPT - Hold Tone.
- #define **RFC_TID_REC**
Fax Tone ID.
- #define **RFC_TID_CALLER_WT**
RFC CPT - Caller Wait Tone.
- #define **RFC_TID_CALL_WT**
RFC CPT - Call Waiting Tone.
- #define **RFC_TID_PAY**
RFC CPT - Pay Tone.
- #define **RFC_TID_POS_IND**
RFC CPT - Positive Indication Tone.
- #define **RFC_TID_NEG_IND**
RFC CPT - Negative Indication Tone.
- #define **RFC_TID_WARNING**
RFC CPT - Warning Tone.
- #define **RFC_TID_INTRUSION**
RFC CPT - Intrusion Tone.
- #define **RFC_TID_CAL_CARD**
RFC CPT - Calling Card Service Tone.
- #define **RFC_TID_PAYPHONE**
RFC CPT - Payphone Recognition Tone.
- #define **RFC_TID_CAS**
RFC CPT - CPE Alerting Signal.
- #define **RFC_TID_OFFHK_WARN**
RFC CPT - Off Hook Warning Tone.
- #define **RFC_TID_CONFIRMATION**
RFC CPT - Dial Tone.
- #define **RFC_TID_THR_PARTY**
RFC CPT - Three Party Remind Tone.
- #define **RFC_TID_UNAVAILABLE**
RFC CPT - Number Unavailable Tone.
- #define **RFC_TID_QUEUE**
RFC CPT - Queue Tone.
- #define **NTT_TID_DT**
Japan country code and pre-defined progress tones.
- #define **NTT_TID_RBT**
Japan country code and pre-defined progress tones.
- #define **NTT_TID_BT**
Japan country code and pre-defined progress tones.
- #define **NTT_TID_PDT**
Japan country code and pre-defined progress tones.

3.2 Defines

- **#define NTT_TID_SDT**
Japan country code and pre-defined progress tones.
- **#define NTT_TID_CPT**
Japan country code and pre-defined progress tones.
- **#define NTT_TID_HST**
Japan country code and pre-defined progress tones.
- **#define NTT_TID_IIT**
Japan country code and pre-defined progress tones.
- **#define NTT_TID_SIIT**
Japan country code and pre-defined progress tones.
- **#define NTT_TID_HOW**
Japan country code and pre-defined progress tones.
- **#define US_TID_DIAL**
United States country code and pre-defined progress tones.
- **#define US_TID_RING**
United States country code and pre-defined progress tones.
- **#define US_TID_BUSY**
United States country code and pre-defined progress tones.
- **#define US_TID_RC_DIAL**
United States country code and pre-defined progress tones.
- **#define US_TID_PBX_DIAL**
United States country code and pre-defined progress tones.
- **#define US_TID_CONGESTION**
United States country code and pre-defined progress tones.
- **#define US_TID_CALL_WT**
United States country code and pre-defined progress tones.
- **#define US_TID_WARN_OPER**
United States country code and pre-defined progress tones.
- **#define PRC_TID_DIAL**
China country code and pre-defined progress tones.
- **#define PRC_TID_RING**
China country code and pre-defined progress tones.
- **#define PRC_TID_BUSY**
China country code and pre-defined progress tones.
- **#define PRC_TID_SP_DIAL**
China country code and pre-defined progress tones.
- **#define PRC_TID_CONGESTION**
China country code and pre-defined progress tones.
- **#define PRC_TID_UNAVAILABLE**
China country code and pre-defined progress tones.
- **#define PRC_TID_TOLL**
China country code and pre-defined progress tones.
- **#define PRC_TID_QUEUE**
China country code and pre-defined progress tones.
- **#define PRC_TID_CALL_WT**
China country code and pre-defined progress tones.
- **#define PRC_TID_THR_PARTY**
China country code and pre-defined progress tones.
- **#define PRC_TID_CONFIRMATION**
China country code and pre-defined progress tones.
- **#define PRC_TID_OFFHK_WARN**
China country code and pre-defined progress tones.
- **#define ADS_TEL_DEC_AUTOSW_OFF**
Decoder AutoSwitch Parameters.
- **#define ADS_TEL_DEC_AUTOSW_G711MU**
Decoder AutoSwitch Parameters.

3.2 Defines

- #define **ADS_TEL_DEC_AUTOSW_G711A**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G729A**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G723**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G722_64**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G726_40**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G726_32**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G726_24**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G726_16**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G729_1**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G722_56**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_G722_48**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_DEC_AUTOSW_ALL**
Decoder AutoSwitch Parameters.
- #define **ADS_TEL_TD_TIMEOUT_FOREVER**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_NONE**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_0**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_1**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_2**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_3**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_4**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_5**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_6**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_7**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_8**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_9**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_STAR**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_POUND**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_A**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_B**
Termination digits.

3.2 Defines

- #define **ADS_TEL_TD_TERM_DIGIT_C**
Termination digits.
- #define **ADS_TEL_TD_TERM_DIGIT_D**
Termination digits.
- #define **ADS_TEL_STOP_REASON_USER**
Stop Reasons.
- #define **ADS_TEL_STOP_REASON_EOD**
Stop Reasons.
- #define **ADS_TEL_STOP_REASON_TERM**
Stop Reasons.
- #define **ADS_TEL_STOP_REASON_TIMEOUT**
Stop Reasons.
- #define **ADS_TEL_STOP_REASON_ERROR**
Stop Reasons.
- #define **ADS_TEL_PARAM_ON**
Parameter Value.
- #define **ADS_TEL_PARAM_OFF**
Parameter Value.
- #define **ADS_TEL_PARAM_EC_NLP_ON**
Parameter Value.
- #define **ADS_TEL_PARAM_EC_NLP_SUB_ON**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_RPT_OFF**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_RPT_TONE_ON**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_RPT_TONE_OFF**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_RPT_TONE_ON_OFF**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_TC_FRAMES_MAX**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_NON_ADAPT_THRES**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_ADAPT_THRES**
Parameter Value.
- #define **ADS_TEL_PARAM_G726_PACK_LSB**
Parameter Value.
- #define **ADS_TEL_PARAM_G726_PACK_MSB**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_V23**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_B202**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_R1200**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_R600**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_R300**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_R150**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_R75**
Parameter Value.
- #define **ADS_TEL_PARAM_WB**
Parameter Value.

3.2 Defines

- #define **ADS_TEL_PARAM_NB**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_CS_US**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_MARK_US**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_CS_JP**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_MARK_JP**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_CS_PRC**
Parameter Value.
- #define **ADS_TEL_PARAM_TG_FSK_MARK_PRC**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_FSK_CS_US**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_FSK_MARK_US**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_FSK_CS_JP**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_FSK_MARK_JP**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_FSK_CS_PRC**
Parameter Value.
- #define **ADS_TEL_PARAM_TD_FSK_MARK_PRC**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_PASSTHRU**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G711MU_10MS**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G711A_10MS**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G729A**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G723**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G722_64**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G722_56**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G722_48**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G726_40**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G726_32**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G726_24**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G726_16**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G729_1**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G722_1**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_G728**
Parameter Value.

3.2 Defines

- #define **ADS_TEL_PARAM_CODER_TYPE_G729I**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_GSM_AMR**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_AMRWB**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_AMRWB_PLUS**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_GSM_FR**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_GSM_HR**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_GSM_EFR**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_QCELP8**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_QCELP13**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_SMV**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_ILBC**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_EVRC**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_SIREN14**
Parameter Value.
- #define **ADS_TEL_PARAM_CODER_TYPE_UNDEF**
Parameter Value.
- #define **ADS_TEL_PARAM_SRC_UPSAMPLE**
Parameter Value.
- #define **ADS_TEL_PARAM_SRC_DOWNSAMPLE**
Parameter Value.
- #define **ADS_TEL_PARAMID_EC_ECENABLE**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_ECTAIL**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_ECNLP**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_ECFREEZE**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_DELAYCOMP**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECENABLE**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECALG**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECTAIL**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECFREEZE**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECZCF**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECCNG**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECNLP**
Parameter ID.

3.2 Defines

- #define **ADS_TEL_PARAMID_EC_AECTD**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECHOWL**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECAP**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECNR**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECNRSMTH**
Parameter ID.
- #define **ADS_TEL_PARAMID_EC_AECDCLFLAG**
Parameter ID.
- #define **ADS_TEL_PARMID_HPF_ENABLE**
Parameter ID.
- #define **ADS_TEL_PARMID_SCL_VOL**
Parameter ID.
- #define **ADS_TEL_PARMID_AGC_ENABLE**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_CNG**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_CTYPE**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_EVT_PKT**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_EVT_PKTCHNG**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_AUTOSW**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_G726_40_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_G726_32_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_G726_24_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_G726_16_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_G726_PACK**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_G729_1_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_G722_64_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_G722_56_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_DEC_G722_48_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_VAD**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_CTYPE**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_MFPP**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_EVT_PKT**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G723_RATE**
Parameter ID.

3.2 Defines

- #define **ADS_TEL_PARMID_ENC_G726_40_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G726_32_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G726_24_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G726_16_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G726_PACK**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G729_1_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G729_1_RATE**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G722_64_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G722_56_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_ENC_G722_48_RTP_PLD**
Parameter ID.
- #define **ADS_TEL_PARMID_TG_VOL**
Parameter ID.
- #define **ADS_TEL_PARMID_TG_FSK_MOD**
Parameter ID.
- #define **ADS_TEL_PARMID_TG_FSK_CS**
Parameter ID.
- #define **ADS_TEL_PARMID_TG_FSK_MARK**
Parameter ID.
- #define **ADS_TEL_PARMID_TG_FSK_RATE**
Parameter ID.
- #define **ADS_TEL_PARMID_TG_RFC2833**
Parameter ID.
- #define **ADS_TEL_PARMID_TG_FSK_POSTMK**
Parameter ID.
- #define **ADS_TEL_PARMID_TG_PCM_MODE**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_TC**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_TC_FRAMES**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_RPT_EVENTS**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_RFC2833E_ENABLE**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_RFC2833E_UPDATERATE**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_RFC2833E_NUMEOE**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_RFC2833E_NUMBOE**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_RFC2833E_AUDIOSUPPRESS**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_RFC2833E_PAYLOADTYPE**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_FSK_CS**
Parameter ID.

3.3 Typedefs

- #define **ADS_TEL_PARMID_TD_FSK_MARK**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_FSK_STOP**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_FSK_RATE**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_FSK_ADAPT_THRESHOLD**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_FSK_RX_MIN_BYTES**
Parameter ID.
- #define **ADS_TEL_PARMID_TD_ENABLE**
Parameter ID.
- #define **ADS_TEL_PARMID_PLY_VOL**
Parameter ID.
- #define **ADS_TEL_PARMID_PLY_MODE**
Parameter ID.
- #define **ADS_TEL_PARMID_SRC_ENABLE**
Parameter ID.
- #define **ADS_TEL_PARMID_SRC_MODE**
Parameter ID.

3.3 Typedefs

- typedef unsigned int **AdsTelEpHandle_t**
Endpoint Handle.
- typedef short **AdsTelMediaHandle_t**
Media Handle.
- typedef **AdsTelMediaDesc_s** **AdsTelMediaDesc_t**
Audio segment to play.

3.4 Enumerations

- enum **AdsTelGeo_t** {
 ADS_TEL_GEO_PROC,
 ADS_TEL_GEO_JAPAN,
 ADS_TEL_GEO_USA,
 ADS_TEL_GEO_DELIMITER
}
 Call Progress Tone Geographical Location Settings.
- enum **AdsTelCallPrior_t** {
 ADS_TEL_CALL_PRIOR_NORMAL,
 ADS_TEL_CALL_PRIOR_911,
 ADS_TEL_CALL_PRIOR_DELIMITER
}
 AdsTel call/endpoint priority.
- enum **AdsTelPipeDir_t** {
 ADS_TEL_PIPE_DIR_IN,
 ADS_TEL_PIPE_DIR_OUT,
 ADS_TEL_PIPE_DIR_DELIMITER
}
 AdsTel endpoint pipeline direction.
- enum **AdsTelPipeConType_t** {
 ADS_TEL_PIPE_CONN_TYPE_PCM,
 ADS_TEL_PIPE_CONN_TYPE_IP,
 ADS_TEL_PIPE_CONN_TYPE_DELIMITER
}

3.4 Enumerations

- ```
}
 AdsTel endpoint connection type.
• enum AdsTelPcmData_t {
 ADS_TEL_PCM_S_BE_16B_WB_LINEAR,
 ADS_TEL_PCM_S_BE_16B_NB_LINEAR,
 ADS_TEL_PCM_S_LE_16B_WB_LINEAR,
 ADS_TEL_PCM_S_LE_16B_NB_LINEAR
}
 PCM data type .
• enum AdsTelEpConStatus_t {
 ADS_TEL_EP_CONN_NONE,
 ADS_TEL_EP_CONN_IN,
 ADS_TEL_EP_CONN_OUT,
 ADS_TEL_EP_CONN_BOTH,
 ADS_TEL_EP_CONN_DELIMITER
}
 AdsTel endpoint connection status.
• enum AdsTelTdCategory_t {
 ADS_TEL_TD_CAT_DTMF,
 ADS_TEL_TD_CAT_DELIMITER
}
 AdsTel Tone Detector tone category.
```

### 3.5 Functions

- **AdsTelStatus\_t adsTelTgToneBuild** (unsigned int toneId, unsigned int numSegs, **AdsTelToneSeg\_t** \*pToneSegs)  
Add a tone to the Tone Generator.
- **AdsTelStatus\_t adsTelTdToneBuild** (unsigned int toneId, **AdsTelToneInfo\_t** \*pToneInfo)  
Build a Tone Detector Tone.
- **AdsTelStatus\_t adsTelTdAmplitudeRangeSet** (**AdsTelTdCategory\_t** category, int ampMinF0, int ampMaxF0, int ampMinF1, int ampMaxF1)  
Set the amplitude detection range for the Tone Detector.
- **AdsTelStatus\_t adsTelTgDtmfParamSet** (int toneOn, int toneOff, int ampdBm)  
Set the tone parameters for the Tone Generator.
- **AdsTelStatus\_t adsTelEpCreate** (unsigned int \*pTransId, **AdsTelPipeType\_t** inPipe, **AdsTelPipeType\_t** outPipe, **AdsTelPcmData\_t** pcmDataType, **adsTelEpDataCb\_t** inCb, **adsTelEpDataCb\_t** outCb, **AdsTelCallPrior\_t** priority, **AdsTelRtrHandle\_t** rtrHandle, **AdsTelEpHandle\_t** \*pEpHandle)  
Create an endpoint.
- **AdsTelStatus\_t adsTelEpConStatusSet** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelEpConStatus\_t** conStatus)  
Set the connection status of an endpoint.
- **AdsTelStatus\_t adsTelEpCompParamsSet** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compId, unsigned int numParams, unsigned int \*pParamIDs, int \*pParamVals)  
Set multiple parameters for an endpoint component.
- **AdsTelStatus\_t adsTelEpCompParamsGet** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compId)  
Get all parameters values for an endpoint component.
- **AdsTelStatus\_t adsTelEpTonePlay** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compId, unsigned int numTones, unsigned int \*pToneIds)  
Play a tone or multiple tones on a tone generator component.
- **AdsTelStatus\_t adsTelEpFskPlay** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compId, unsigned int numBytes, char \*pFskData)  
Play FSK Data on a tone generator component.

## 3.5 Functions

- **AdsTelStatus\_t adsTelEpToneRecv** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compld, unsigned int totalTimeOut, unsigned int firstDigitTimeout, unsigned int interDigitTimeout, unsigned int termDigit, unsigned int numDigits)  
Receive a tone or multiple tones on a tone detector component.
  - **AdsTelStatus\_t adsTelEpFskRecv** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compld, unsigned int timeout)  
Receive FSK bytes on a tone detector component.
  - **AdsTelStatus\_t adsTelEpPlayAudio** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compld, **AdsTelMediaDesc\_t** \*pSegments, short nSegs)  
Play an audio file.
  - **AdsTelStatus\_t adsTelRegisterWavFile** (char \*filename, **AdsTelMediaHandle\_t** \*pMediaHandle)  
Register a wav file.
  - **AdsTelStatus\_t adsTelEpCompStart** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compld)  
Start a component.
  - **AdsTelStatus\_t adsTelEpCompStop** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compld)  
Stop a component.
  - **AdsTelStatus\_t adsTelEpCompReset** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle, **AdsTelPipeDir\_t** pipeDir, unsigned int compld)  
Reset a component.
  - **AdsTelStatus\_t adsTelEpDestroy** (unsigned int \*pTransId, **AdsTelEpHandle\_t** epHandle)  
Destroy an endpoint.
- 

## 3.6 Detailed Description

Pipeline instance allocation and deallocation functions for the ADS Telephony Framework.

### Purpose:

The functions in this group are responsible for all aspects of pipeline allocation for the ADS Telephony Framework.

---

## 3.7 Data Structure Documentation

### 3.7.1 AdsTelMediaDesc\_s Struct Reference

Audio segment to play.

#### 3.7.1.1 Data Fields

- int **offset**  
offset in bytes where player starts.
- int **length**  
length to play (in 10ms unit), 0 to play until the end.
- **AdsTelMediaHandle\_t** **handle**  
media handle to play.
- short **next**  
next segment, 0 to play segment in infinite loop, **ADS\_TEL\_MEDIA\_SEG\_EOP** for the last segment.



### 3.7.1 AdsTelMediaDesc\_s Struct Reference

#### 3.7.1.2 Detailed Description

Audio segment to play.

**Description:**

This structure is used to define the audio segments to play. It is passed as a parameter to **adsTelEpPlayAudio()**.

**Purpose:**

Parameters of the audio segment to play.

**See also:**

**AdsTelMediaHandle\_t**  
**adsTelEpPlayAudio()**.

#### 3.7.1.3 Field Documentation

int **AdsTelMediaDesc\_s::offset**

offset in bytes where player starts.

int **AdsTelMediaDesc\_s::length**

length to play (in 10ms unit), 0 to play until the end.

**AdsTelMediaHandle\_t** **AdsTelMediaDesc\_s::handle**

media handle to play.

short **AdsTelMediaDesc\_s::next**

next segment, 0 to play segment in infinite loop, **ADS\_TEL\_MEDIA\_SEG\_EOP** for the last segment.

---

### 3.7.2 AdsTelToneSeg\_t Struct Reference

Tone segment structure.

#### 3.7.2.1 Data Fields

- unsigned short **repCount**  
Number of repeating the segment.
- unsigned short **segType**  
signal type (single, dual or AM tone )
- unsigned int **durationOn**  
tone on duration in milli seconds.
- unsigned int **durationOff**  
tone off duration in milli seconds.
- short **freqA**  
frequency A in Hz
- short **freqB**  
the 2nd frequency in Hz if dual or AM tone, ignored if single tone
- short **ampA**  
amplitude of A signal in dBm
- short **ampB**  
amplitude of B signal in dBm if dual tone, modulation rate in percentage if AM tone, ignored if single tone
- unsigned short **mode**

### 3.7.2 AdsTelToneSeg\_t Struct Reference

- mode, overwrite or mix over the decoder output
- short **nextSeg**  
the index of next segment relative to the current segment.

#### 3.7.2.2 Detailed Description

Tone segment structure.

**Description:**

This structure is used to define the segment data for tone generation template. It is passed as a parameter to **adsTelTgToneBuild()**.

**Purpose:**

Parameters of the tone to build for tone generation.

**See also:**

**adsTelTgToneBuild.**

#### 3.7.2.3 Field Documentation

unsigned short **AdsTelToneSeg\_t::repCount**

Number of repeating the segment.

0 means to repeat forever

unsigned short **AdsTelToneSeg\_t::segType**

signal type (single, dual or AM tone )

unsigned int **AdsTelToneSeg\_t::durationOn**

tone on duration in milli seconds.

unsigned int **AdsTelToneSeg\_t::durationOff**

tone off duration in milli seconds.

short **AdsTelToneSeg\_t::freqA**

frequency A in Hz

short **AdsTelToneSeg\_t::freqB**

the 2nd frequency in Hz if dual or AM tone, ignored if single tone

short **AdsTelToneSeg\_t::ampA**

amplitude of A signal in dBm

short **AdsTelToneSeg\_t::ampB**

amplitude of B signal in dBm if dual tone, modulation rate in percentage if AM tone, ignored if single tone

unsigned short **AdsTelToneSeg\_t::mode**

mode, overwrite or mix over the decoder output

short **AdsTelToneSeg\_t::nextSeg**

the index of next segment relative to the current segment.

### 3.7.3 AdsTelToneInfo\_t Struct Reference

e.g., 1 means to go the following segment, 0 means repeat the current segment, -2 means go back to previous 2 segments. ADS\_TEL\_TG\_LASTSEG means the last segment

---

### 3.7.3 AdsTelToneInfo\_t Struct Reference

Tone Information structure.

#### 3.7.3.1 Data Fields

- unsigned short **type**  
single or dual frequency tone
- unsigned short **criteria**  
loose, medium or tight
- unsigned short **freqLowA**  
low bound of the 1st frequency in Hz
- unsigned short **freqHighA**  
high bound of the 1st frequency in Hz
- unsigned short **freqLowB**  
low bound of the 2nd frequency in Hz
- unsigned short **freqHighB**  
high bound of the 2nd frequency in Hz
- short **ampLowA**  
low level of the 1st frequency in dBm
- short **ampHighA**  
high level of the 1st frequency in dBm
- short **ampLowB**  
low level of the 2nd frequency in dBm
- short **ampHighB**  
high level of the 2nd frequency in dBm
- char **attributes**  
attribute (how to report the tone can be)

#### 3.7.3.2 Detailed Description

Tone Information structure.

##### Description:

This structure is used to define the tone informations for tone detection template. It is passed as a parameter to **adsTelTdToneBuild()**.

##### Purpose:

Parameters of the tone to build for tone detection.

##### See also:

**adsTelTdToneBuild.**

#### 3.7.3.3 Field Documentation

unsigned short **AdsTelToneInfo\_t::type**  
single or dual frequency tone

unsigned short **AdsTelToneInfo\_t::criteria**  
loose, medium or tight

### 3.8 Define Documentation

unsigned short **AdsTelToneInfo\_t::freqLowA**

low bound of the 1st frequency in Hz

unsigned short **AdsTelToneInfo\_t::freqHighA**

high bound of the 1st frequency in Hz

unsigned short **AdsTelToneInfo\_t::freqLowB**

low bound of the 2nd frequency in Hz

unsigned short **AdsTelToneInfo\_t::freqHighB**

high bound of the 2nd frequency in Hz

short **AdsTelToneInfo\_t::ampLowA**

low level of the 1st frequency in dBm

short **AdsTelToneInfo\_t::ampHighA**

high level of the 1st frequency in dBm

short **AdsTelToneInfo\_t::ampLowB**

low level of the 2nd frequency in dBm

short **AdsTelToneInfo\_t::ampHighB**

high level of the 2nd frequency in dBm

char **AdsTelToneInfo\_t::attributes**

attribute (how to report the tone can be)

---

## 3.8 Define Documentation

**#define ADS\_TEL\_MAX\_TONES**

Maximum number of tones that can be sent to or detected by a tone generator component.

**Description:**

Used to define the maximum number of tones that can be sent to or detected by a tone generator component. See **adsTelEpTonePlay()**.

**#define ADS\_TEL\_MAX\_TONE\_ID**

Maximum tone ID.

**Description:**

Maximum tone ID that can be passed to the tone generator/detector component. See **adsTelEpTonePlay()**.

**#define ADS\_TEL\_MAX\_FSK\_BYTES**

Maximum number of FSK bytes that can be sent to or detected by a tone generator component.

**Description:**

Used to define the maximum number of FSK bytes that can be sent to or detected by a tone generator component. See **adsTelEpFskPlay()**.

### 3.8 Define Documentation

```
#define ADS_TEL_MAX_PARAMS
```

Maximum number of parameters that can be set or retrieved for any component.

**Description:**

Used to define the maximum number of parameters that can be set or retrieved for any component. See **adsTelEpCompParamsSet()** and

```
#define ADS_TEL_MEDIA_SEG_EOP
```

Identifier for the last segment to play.

**Description:**

Used to specify that this is the last segment to play. See **AdsTelMediaDesc\_t**

```
#define ADS_TEL_PLY_MAX_SEGS
```

Maximum number of segments that can be sent to a player component.

**Description:**

Used to define the maximum number of segments that can be sent to a player.

```
#define ADS_TEL_TG_LASTSEG
```

Tone Generator last segment identifier.

**Description:**

Used to define the last segment of a tone segment array buffer. See **AdsTelToneSeg\_t**.

```
#define ADS_TEL_TG_MODE_OVERWRITE
```

Tone Generator segment output mode type.

**Description:**

Used to define the output mode of a tone generator as overwrite. This means that the tone generator will overwrite the output of the decoder. See **AdsTelToneSeg\_t**.

```
#define ADS_TEL_TG_MODE_MIX
```

Tone Generator segment output mode type.

**Description:**

Used to define the output mode of a tone generator as mixer mode. This means that the output of the tone generator and the decoder will be mixed. See **AdsTelToneSeg\_t**.

```
#define ADS_TEL_TG_SINGLETONE
```

Tone Generator segment tone type.

**Description:**

Used to define the tone generator segment tone type as Single Tone. See **AdsTelToneSeg\_t**.

```
#define ADS_TEL_TG_DUALTONE
```

Tone Generator segment tone type.

**Description:**

### 3.8 Define Documentation

Used to define the tone generator segment tone type as Dual Tone. See **AdSTelToneSeg\_t**.

```
#define ADS_TEL_TG_AMTONE
```

Tone Generator segment tone type.

**Description:**

Used to define the tone generator segment tone type as Amplitude Modulated. See **AdSTelToneSeg\_t**.

```
#define ADS_TEL_TD_QUAL_CRITERIA_LOOSE
```

Tone Detector qualification criteria type.

**Description:**

Used to define the tone detector qualification criteria as loose. See **AdSTelToneInfo\_t**.

```
#define ADS_TEL_TD_QUAL_CRITERIA_MEDIUM
```

Tone Detector qualification criteria type.

**Description:**

Used to define the tone detector qualification criteria as medium. See **AdSTelToneInfo\_t**.

```
#define ADS_TEL_TD_QUAL_CRITERIA_TIGHT
```

Tone Detector qualification criteria type.

**Description:**

Used to define the tone detector qualification criteria as tight. See **AdSTelToneInfo\_t**.

```
#define ADS_TEL_TD_TONEATTR_EVTON
```

Tone Detector event report attribute type.

**Description:**

Used to define the tone detector event reporting attribute as tone on only. See **AdSTelToneInfo\_t**.

```
#define ADS_TEL_TD_TONEATTR_EVTOFF
```

Tone Detector event report attribute type.

**Description:**

Used to define the tone detector event reporting attribute as tone off only. See **AdSTelToneInfo\_t**.

```
#define ADS_TEL_TD_TONEATTR_EVTBOTH
```

Tone Detector event report attribute type.

**Description:**

Used to define the tone detector event reporting attribute as tone on and tone off. See **AdSTelToneInfo\_t**.

```
#define ADS_TEL_TD_SINGLETONE
```

### 3.8 Define Documentation

Tone Detector tone type.

**Description:**

Used to define the tone detector tone type as single tone. See **AdstelToneInfo\_t**.

```
#define ADS_TEL_TD_DUALTONE
```

Tone Detector tone type.

**Description:**

Used to define the tone detector tone type as dual tone. See **AdstelToneInfo\_t**.

```
#define RFC_TID_DTMF_0
```

RFC DTMF 0.

**Description:**

Used to define the RFC DTMF 0 tone.

```
#define RFC_TID_DTMF_1
```

RFC DTMF 1.

**Description:**

Used to define the RFC DTMF 1 tone.

```
#define RFC_TID_DTMF_2
```

RFC DTMF 2.

**Description:**

Used to define the RFC DTMF 2 tone.

```
#define RFC_TID_DTMF_3
```

RFC DTMF 3.

**Description:**

Used to define the RFC DTMF 3 tone.

```
#define RFC_TID_DTMF_4
```

RFC DTMF 4.

**Description:**

Used to define the RFC DTMF 4 tone.

```
#define RFC_TID_DTMF_5
```

RFC DTMF 5.

**Description:**

Used to define the RFC DTMF 5 tone.

### 3.8 Define Documentation

```
#define RFC_TID_DTMF_6
RFC DTMF 6.
```

**Description:**

Used to define the RFC DTMF  
6 tone.

```
#define RFC_TID_DTMF_7
RFC DTMF 7.
```

**Description:**

Used to define the RFC DTMF  
7 tone.

```
#define RFC_TID_DTMF_8
RFC DTMF 8.
```

**Description:**

Used to define the RFC DTMF  
8 tone.

```
#define RFC_TID_DTMF_9
RFC DTMF 9.
```

**Description:**

Used to define the RFC DTMF  
9 tone.

```
#define RFC_TID_DTMF_STAR
RFC DTMF 10.
```

**Description:**

Used to define the RFC DTMF  
10 tone.

```
#define RFC_TID_DTMF_POUND
RFC DTMF 11.
```

**Description:**

Used to define the RFC DTMF  
11 tone.

```
#define RFC_TID_DTMF_A
RFC DTMF 12.
```

**Description:**

Used to define the RFC DTMF  
12 tone.

```
#define RFC_TID_DTMF_B
RFC DTMF 13.
```

**Description:**



### 3.8 Define Documentation

Used to define the RFC DTMF  
13 tone.

```
#define RFC_TID_DTMF_C
```

Tone Detector tone type.

**Description:**

Used to define the tone detector tone type. See  
**AdstelToneInfo\_t**.

```
#define RFC_TID_DTMF_D
```

RFC DTMF 15.

**Description:**

Used to define the RFC DTMF  
15 tone.

```
#define RFC_TID_FAX_CED
```

Fax CED Tone.

**Description:**

Used to define a Fax  
CED tone.

```
#define RFC_TID_FAX_CNG
```

Fax CNG Tone.

**Description:**

Used to define a Fax  
CNG tone.

```
#define RFC_TID_FAX_V21
```

Fax V21 Tone.

**Description:**

Used to define a Fax  
V21 tone.

```
#define RFC_CPT_START
```

Call Progress Tone Base.

**Description:**

Used to the call progress  
tone base.

```
#define RFC_TID_DIAL
```

RFC CPT - Dial Tone.

**Description:**

Used to define the RFC CPT - Dial  
Tone.

```
#define RFC_TID_PBX_DIAL
```

### 3.8 Define Documentation

RFC CPT - PBX Dial Tone.

**Description:**

Used to define the RFC CPT - PBX Dial Tone.

```
#define RFC_TID_SP_DIAL
```

RFC CPT - Special Dial Tone.

**Description:**

Used to define the RFC CPT - Special Dial Tone.

```
#define RFC_TID_2ND_DIAL
```

RFC CPT - 2nd Dial Tone.

**Description:**

Used to define the RFC CPT - 2nd Dial Tone.

```
#define RFC_TID_RING
```

RFC CPT - Ring Tone.

**Description:**

Used to define the RFC CPT - Ring Tone.

```
#define RFC_TID_SP_RING
```

RFC CPT - Special Ring Tone.

**Description:**

Used to define the RFC CPT - Special Ring Tone.

```
#define RFC_TID_BUSY
```

RFC CPT - Busy Tone.

**Description:**

Used to define the RFC CPT - Busy Tone.

```
#define RFC_TID_CONGESTION
```

RFC CPT - Congestion Tone.

**Description:**

Used to define the RFC CPT - Congestion Tone.

```
#define RFC_TID_SP_INFO
```

RFC CPT - Special Information Tone.

**Description:**

Used to define the RFC CPT - Special Information Tone.

### 3.8 Define Documentation

```
#define RFC_TID_COMFORT
RFC CPT - Comfort Tone.
```

**Description:**

Used to define the RFC CPT - Comfort Tone.

```
#define RFC_TID_HOLD
RFC CPT - Hold Tone.
```

**Description:**

Used to define the RFC CPT - Hold Tone.

```
#define RFC_TID_REC
Fax Tone ID.
```

**Description:**

Used to define a Fax tone - record tone. See **AdstelToneInfo\_t**.

```
#define RFC_TID_CALLER_WT
RFC CPT - Caller Wait Tone.
```

**Description:**

Used to define the RFC CPT - Caller Wait Tone.

```
#define RFC_TID_CALL_WT
RFC CPT - Call Waiting Tone.
```

**Description:**

Used to define the RFC CPT - Call Waiting Tone.

```
#define RFC_TID_PAY
RFC CPT - Pay Tone.
```

**Description:**

Used to define the RFC CPT - Pay Tone.

```
#define RFC_TID_POS_IND
RFC CPT - Positive Indication Tone.
```

**Description:**

Used to define the RFC CPT - Positive Indication Tone.

```
#define RFC_TID_NEG_IND
RFC CPT - Negative Indication Tone.
```

**Description:**

### 3.8 Define Documentation

Used to define the RFC CPT - Negative Indication Tone.

```
#define RFC_TID_WARNING
```

RFC CPT - Warning Tone.

**Description:**

Used to define the RFC CPT - Warning Tone.

```
#define RFC_TID_INTRUSION
```

RFC CPT - Intrusion Tone.

**Description:**

Used to define the RFC CPT - Intrusion Tone.

```
#define RFC_TID_CAL_CARD
```

RFC CPT - Calling Card Service Tone.

**Description:**

Used to define the RFC CPT - Calling Card Service Tone.

```
#define RFC_TID_PAYPHONE
```

RFC CPT - Payphone Recognition Tone.

**Description:**

Used to define the RFC CPT - Payphone Recognition Tone.

```
#define RFC_TID_CAS
```

RFC CPT - CPE Alerting Signal.

**Description:**

Used to define the RFC CPT - CPE Alerting Signal.

```
#define RFC_TID_OFFHK_WARN
```

RFC CPT - Off Hook Warning Tone.

**Description:**

Used to define the RFC CPT - Off Hook Warning Tone.

```
#define RFC_TID_CONFIRMATION
```

RFC CPT - Dial Tone.

**Description:**

Used to define the RFC CPT - Confirmation Tone.

```
#define RFC_TID_THR_PARTY
```

### 3.8 Define Documentation

RFC CPT - Three Party Remind Tone.

**Description:**

Used to define the RFC CPT - Three Party Remind Tone.

```
#define RFC_TID_UNAVAILABLE
```

RFC CPT - Number Unavailable Tone.

**Description:**

Used to define the RFC CPT - Number Unavailable Tone.

```
#define RFC_TID_QUEUE
```

RFC CPT - Queue Tone.

**Description:**

Used to define the RFC CPT - Queue Tone.

```
#define NTT_TID_DT
```

Japan country code and pre-defined progress tones.

**Description:**

Dial tone.

```
#define NTT_TID_RBT
```

Japan country code and pre-defined progress tones.

**Description:**

Ring back tone.

```
#define NTT_TID_BT
```

Japan country code and pre-defined progress tones.

**Description:**

Busy tone.

```
#define NTT_TID_PDT
```

Japan country code and pre-defined progress tones.

**Description:**

Private dial tone.

```
#define NTT_TID_SDT
```

Japan country code and pre-defined progress tones.

**Description:**

Second dial tone.

```
#define NTT_TID_CPT
```

Japan country code and pre-defined progress tones.

### 3.8 Define Documentation

**Description:**

Acceptance tone.

```
#define NTT_TID_HST
```

Japan country code and pre-defined progress tones.

**Description:**

Hold service tone.

```
#define NTT_TID_IIT
```

Japan country code and pre-defined progress tones.

**Description:**

Incoming id tone.

```
#define NTT_TID_SIIT
```

Japan country code and pre-defined progress tones.

**Description:**

Special incoming id tone.

```
#define NTT_TID_HOW
```

Japan country code and pre-defined progress tones.

**Description:**

Howler tone.

```
#define US_TID_DIAL
```

United States country code and pre-defined progress tones.

**Description:**

Dial tone.

```
#define US_TID_RING
```

United States country code and pre-defined progress tones.

**Description:**

Ring back tone.

```
#define US_TID_BUSY
```

United States country code and pre-defined progress tones.

**Description:**

Busy tone.

```
#define US_TID_RC_DIAL
```

United States country code and pre-defined progress tones.

**Description:**

Recall dial tone.

```
#define US_TID_PBX_DIAL
```

United States country code and pre-defined progress tones.

### 3.8 Define Documentation

**Description:**

PBX dial tone.

```
#define US_TID_CONGESTION
```

United States country code and pre-defined progress tones.

**Description:**

Congestion tone.

```
#define US_TID_CALL_WT
```

United States country code and pre-defined progress tones.

**Description:**

Call waiting tone.

```
#define US_TID_WARN_OPER
```

United States country code and pre-defined progress tones.

**Description:**

Operator intervening tone.

```
#define PRC_TID_DIAL
```

China country code and pre-defined progress tones.

**Description:**

Dial tone.

```
#define PRC_TID_RING
```

China country code and pre-defined progress tones.

**Description:**

Ring back tone.

```
#define PRC_TID_BUSY
```

China country code and pre-defined progress tones.

**Description:**

Busy tone.

```
#define PRC_TID_SP_DIAL
```

China country code and pre-defined progress tones.

**Description:**

Special dial tone.

```
#define PRC_TID_CONGESTION
```

China country code and pre-defined progress tones.

**Description:**

Congestion tone.

```
#define PRC_TID_UNAVAILABLE
```

China country code and pre-defined progress tones.

### 3.8 Define Documentation

**Description:**

Unavailable tone.

```
#define PRC_TID_TOLL
```

China country code and pre-defined progress tones.

**Description:**

Long distance tone.

```
#define PRC_TID_QUEUE
```

China country code and pre-defined progress tones.

**Description:**

Queue tone.

```
#define PRC_TID_CALL_WT
```

China country code and pre-defined progress tones.

**Description:**

Call waiting tone.

```
#define PRC_TID_THR_PARTY
```

China country code and pre-defined progress tones.

**Description:**

3 party remind tone. See **AdsTelToneInfo\_t**.

```
#define PRC_TID_CONFIRMATION
```

China country code and pre-defined progress tones.

**Description:**

Confirmation tone.

```
#define PRC_TID_OFFHK_WARN
```

China country code and pre-defined progress tones.

**Description:**

Howler tone.

```
#define ADS_TEL_DEC_AUTOSW_OFF
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type  
- OFF.

```
#define ADS_TEL_DEC_AUTOSW_G711MU
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type -  
G711MU.



### 3.8 Define Documentation

```
#define ADS_TEL_DEC_AUTOSW_G711A
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type - G711A.

```
#define ADS_TEL_DEC_AUTOSW_G729A
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type - G729A.

```
#define ADS_TEL_DEC_AUTOSW_G723
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type - G723.

```
#define ADS_TEL_DEC_AUTOSW_G722_64
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type - G722.

```
#define ADS_TEL_DEC_AUTOSW_G726_40
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type - G726\_40.

```
#define ADS_TEL_DEC_AUTOSW_G726_32
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type - G726\_32.

```
#define ADS_TEL_DEC_AUTOSW_G726_24
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type - G726\_24.

```
#define ADS_TEL_DEC_AUTOSW_G726_16
```

Decoder AutoSwitch Parameters.

**Description:**

### 3.8 Define Documentation

Mask bits used to specify the coder type -  
G726\_16.

```
#define ADS_TEL_DEC_AUTOSW_G729_1
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type -  
G729\_1.

```
#define ADS_TEL_DEC_AUTOSW_G722_56
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type -  
G722\_56.

```
#define ADS_TEL_DEC_AUTOSW_G722_48
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type -  
G722\_48.

```
#define ADS_TEL_DEC_AUTOSW_ALL
```

Decoder AutoSwitch Parameters.

**Description:**

Mask bits used to specify the coder type  
- ALL.

```
#define ADS_TEL_TD_TIMEOUT_FOREVER
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See  
**adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_NONE
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See  
**adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_0
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See  
**adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_1
```

### 3.8 Define Documentation

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_2
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_3
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_4
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_5
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_6
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_7
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_8
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

### 3.8 Define Documentation

```
#define ADS_TEL_TD_TERM_DIGIT_9
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_STAR
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_POUND
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_A
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_B
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_C
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_TD_TERM_DIGIT_D
```

Termination digits.

**Description:**

Mask bits used to specify the termination digit for TD. See **adsTelEpToneRecv()**.

```
#define ADS_TEL_STOP_REASON_USER
```

Stop Reasons.

**Description:**

### 3.8 Define Documentation

Stopped by user. See **AdsTelRetTg\_t AdsTelRetTd\_t.**  
**AdsTelRetTdFsk\_t**

```
#define ADS_TEL_STOP_REASON_EOD
```

Stop Reasons.

**Description:**

End of Data. See **AdsTelRetTg\_t AdsTelRetTd\_t.**  
**AdsTelRetTdFsk\_t**

```
#define ADS_TEL_STOP_REASON_TERM
```

Stop Reasons.

**Description:**

Stopped by a termination condition. See **AdsTelRetTg\_t AdsTelRetTd\_t.**  
**AdsTelRetTdFsk\_t**

```
#define ADS_TEL_STOP_REASON_TIMEOUT
```

Stop Reasons.

**Description:**

Time out. See **AdsTelRetTg\_t AdsTelRetTd\_t.**  
**AdsTelRetTdFsk\_t**

```
#define ADS_TEL_STOP_REASON_ERROR
```

Stop Reasons.

**Description:**

General error. See **AdsTelRetTg\_t AdsTelRetTd\_t.**  
**AdsTelRetTdFsk\_t**

```
#define ADS_TEL_PARAM_ON
```

Parameter Value.

**Description:**

Enable a parameter.

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_OFF
```

Parameter Value.

**Description:**

Disable a parameter.

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_EC_NLP_ON
```

Parameter Value.

### 3.8 Define Documentation

**Description:**

EC Non Linear  
Processing On

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_EC_NLP_SUB_ON
```

Parameter Value.

**Description:**

EC Non Linear Processing Sub  
band on.

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_RPT_OFF
```

Parameter Value.

**Description:**

Tone Detector Report  
Tone On

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_RPT_TONE_ON
```

Parameter Value.

**Description:**

Tone Detector Report  
Tone On

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_RPT_TONE_OFF
```

Parameter Value.

**Description:**

Tone Detector Report  
Tone off

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_RPT_TONE_ON_OFF
```

Parameter Value.

### 3.8 Define Documentation

**Description:**

Tone Detector Report  
Tone on/off

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_TC_FRAMES_MAX
```

Parameter Value.

**Description:**

Tone Detector TC  
Frame Max

```
#define ADS_TEL_PARAM_TD_NON_ADAPT_THRES
```

Parameter Value.

**Description:**

Tone Detector non adaptive  
threshold.

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_ADAPT_THRES
```

Parameter Value.

**Description:**

Tone Detector adaptive  
threshold.

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_G726_PACK_LSB
```

Parameter Value.

**Description:**

G726 Pack LSB

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_G726_PACK_MSB
```

Parameter Value.

**Description:**

G726 Pack MSB

**See also:**

### 3.8 Define Documentation

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_V23
```

Parameter Value.

**Description:**

Tone Generator FSK  
V23

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_B202
```

Parameter Value.

**Description:**

Tone Generator FSK  
B202

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_R1200
```

Parameter Value.

**Description:**

Tone Generator FSK  
R1200

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_R600
```

Parameter Value.

**Description:**

Tone Generator FSK  
R600

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_R300
```

Parameter Value.

**Description:**

Tone Generator FSK  
R300



### 3.8 Define Documentation

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_R150
```

Parameter Value.

**Description:**

Tone Generator FSK  
R150

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_R75
```

Parameter Value.

**Description:**

Tone Generator FSK  
R75

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_WB
```

Parameter Value.

**Description:**

Tone Generator/Player  
wideband mode

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_NB
```

Parameter Value.

**Description:**

Tone Generator/Player  
narrowband mode

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_CS_US
```

Parameter Value.

**Description:**

Country Specific Tone Generator FSK  
CS US

### 3.8 Define Documentation

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_MARK_US
```

Parameter Value.

**Description:**

Country Specific Tone Generator FSK  
Mark US

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_CS_JP
```

Parameter Value.

**Description:**

Country Specific Tone Generator FSK CS  
Japan

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_MARK_JP
```

Parameter Value.

**Description:**

Country Specific Tone Generator FSK Mark  
Japan

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_CS_PRC
```

Parameter Value.

**Description:**

Country Specific Tone Generator FSK CS Peoples Republic of  
China

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TG_FSK_MARK_PRC
```

Parameter Value.

**Description:**

Country Specific Tone Generator FSK Mark Peoples Republic of  
China

### 3.8 Define Documentation

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_FSK_CS_US
Parameter Value.
```

**Description:**

Country Specific Tone Detector FSK  
CS US

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_FSK_MARK_US
Parameter Value.
```

**Description:**

Country Specific Tone Detector FSK  
Mark US

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_FSK_CS_JP
Parameter Value.
```

**Description:**

Country Specific Tone Detector FSK  
CS JP

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_FSK_MARK_JP
Parameter Value.
```

**Description:**

Country Specific Tone Detector FSK  
Mark JP

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_FSK_CS_PRC
Parameter Value.
```

**Description:**

Country Specific Tone Detector FSK CS Peoples Republic of  
China

### 3.8 Define Documentation

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_TD_FSK_MARK_PRC
```

Parameter Value.

**Description:**

Country Specific Tone Detector FSK Mark Peoples Republic of China

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_PASSTHRU
```

Parameter Value.

**Description:**

Codec Type Pass  
Through

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G711MU_10MS
```

Parameter Value.

**Description:**

Codec Type G711 MU  
Law 10ms

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G711A_10MS
```

Parameter Value.

**Description:**

Codec Type G711 A  
Law 10ms

**See also:**

**adsTelEpCompParamsSet()  
adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G729A
```

Parameter Value.

**Description:**

Codec Type G729ab

### 3.8 Define Documentation

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G723
```

Parameter Value.

**Description:**

Codec Type G723.1

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G722_64
```

Parameter Value.

**Description:**

Codec Type G722  
64Kbps

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G722_56
```

Parameter Value.

**Description:**

Codec Type G722  
56Kbps

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G722_48
```

Parameter Value.

**Description:**

Codec Type G722  
48kbps

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G726_40
```

Parameter Value.

**Description:**

Codec Type G726 40

**See also:**

### 3.8 Define Documentation

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G726_32
```

Parameter Value.

**Description:**

Codec Type G726 32

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G726_24
```

Parameter Value.

**Description:**

Codec Type G726 24

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G726_16
```

Parameter Value.

**Description:**

Codec Type G726 16

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G729_1
```

Parameter Value.

**Description:**

Codec Type G729 1

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G722_1
```

Parameter Value.

**Description:**

Codec Type G722.1

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G728
```

### 3.8 Define Documentation

Parameter Value.

**Description:**

Codec Type G728

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_G729I
```

Parameter Value.

**Description:**

Codec Type G729I  
Integrated Mode

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_GSM_AMR
```

Parameter Value.

**Description:**

Codec Type GSM AMR

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_AMRWB
```

Parameter Value.

**Description:**

Codec Type ITU G722.2, 3GPP  
AMRWB

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_AMRWB_PLUS
```

Parameter Value.

**Description:**

Codec Type AMRWB+, 3GPP AMRWB Enhanced  
codec

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_GSM_FR
```

Parameter Value.

### 3.8 Define Documentation

**Description:**

Codec Type GSM Full  
Rate

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_GSM_HR
```

Parameter Value.

**Description:**

Codec Type GSM Half  
Rate

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_GSM_EFR
```

Parameter Value.

**Description:**

Codec Type GSM Enhanced Full Rate  
Codec

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_QCELP8
```

Parameter Value.

**Description:**

Codec Type QCELP8, Qualcomm  
Codec

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_QCELP13
```

Parameter Value.

**Description:**

Codec Type QCELP13, Qualcomm  
Codec

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_SMV
```

Parameter Value.



### 3.8 Define Documentation

**Description:**

Codec Type Selectable Mode  
Vocoder

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_ILBC
```

Parameter Value.

**Description:**

Codec Type Internet Low Bit rate  
codec

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_EVRC
```

Parameter Value.

**Description:**

Codec Type Enhanced variable  
rate codec

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_SIREN14
```

Parameter Value.

**Description:**

Codec Type ITU  
G722.1C

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_CODER_TYPE_UNDEF
```

Parameter Value.

**Description:**

Codec Type Undefined

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_SRC_UPSAMPLE
```

Parameter Value.

### 3.8 Define Documentation

**Description:**

Upsample data from NB  
to WB

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAM_SRC_DOWNSAMPLE
```

Parameter Value.

**Description:**

downsample data from  
WB to NB

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_ECENABLE
```

Parameter ID.

**Description:**

EC Flag  
**ADS\_TEL\_PARAM\_ON** -  
enable  
**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_ECTAIL
```

Parameter ID.

**Description:**

EC tail length

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_ECMLP
```

Parameter ID.

**Description:**

EC NLP flag,  
**ADS\_TEL\_PARAM\_OFF** - disable  
**ADS\_TEL\_PARAM\_EC\_NLP\_ON** - NLP on & Suppr  
Off  
**ADS\_TEL\_PARAM\_EC\_NLP\_SUB\_ON** - NLP on &  
Suppr on

**See also:**

### 3.8 Define Documentation

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_ECFREEZE
```

Parameter ID.

**Description:**

EC freeze flag

**ADS\_TEL\_PARAM\_ON** -

freeze

**ADS\_TEL\_PARAM\_OFF** - not

freeze

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_DELAYCOMP
```

Parameter ID.

**Description:**

EC delay componesation

0 ~ 40

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECENABLE
```

Parameter ID.

**Description:**

AEC flag

**ADS\_TEL\_PARAM\_ON** -

enable

**ADS\_TEL\_PARAM\_OFF** -

disable

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECALG
```

Parameter ID.

**Description:**

AEC ALG flag

0 - Full band

1 - Sub-band

2 - Sub band fast

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

### 3.8 Define Documentation

```
#define ADS_TEL_PARAMID_EC_AECTAIL
```

Parameter ID.

**Description:**

AEC Tail Length 1 to  
200 ms

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECFREEZE
```

Parameter ID.

**Description:**

AEC freeze flag  
**ADS\_TEL\_PARAM\_ON** -  
freeze  
**ADS\_TEL\_PARAM\_OFF** - not  
freeze

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECZCF
```

Parameter ID.

**Description:**

AEC Use of Zero Coeff  
**ADS\_TEL\_PARAM\_ON** -  
enable  
**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECCNG
```

Parameter ID.

**Description:**

AEC CNG  
**ADS\_TEL\_PARAM\_ON** -  
enable  
**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECNLP
```

### 3.8 Define Documentation

Parameter ID.

**Description:**

AEC NLP  
0 - disable  
1 - enable general  
2 - enable sbf specific algo  
3 - enable gen with initial anti-howling supressor,  
sbf only

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECTD
```

Parameter ID.

**Description:**

AEC TD flg  
**ADS\_TEL\_PARAM\_ON** -  
enable  
**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECHOWL
```

Parameter ID.

**Description:**

AEC AntiHowling  
0 - disable anti-howling  
1 - spectra-based HD  
2 - energy-based HD

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECAP
```

Parameter ID.

**Description:**

affine projection  
order,1...

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECNR
```

Parameter ID.

### 3.8 Define Documentation

**Description:**

IPP NR noise mitigation  
level 0,...,5

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECNRSMTH
```

Parameter ID.

**Description:**

IPP NR smoothing  
mode 0,...,2

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_EC_AECDCLFLAG
```

Parameter ID.

**Description:**

IPP NR DC  
compensation flag

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_HPF_ENABLE
```

Parameter ID.

**Description:**

High Pass Filter Flag  
**ADS\_TEL\_PARAM\_ON** -  
enable  
**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARAMID_SCL_VOL
```

Parameter ID.

**Description:**

SCL Volume

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

### 3.8 Define Documentation

```
#define ADS_TEL_PARMID_AGC_ENABLE
```

Parameter ID.

**Description:**

AGC flag

**ADS\_TEL\_PARAM\_ON** -  
enable

**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_CNG
```

Parameter ID.

**Description:**

G711 CNG flag

**ADS\_TEL\_PARAM\_ON** -  
enable

**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_CTYPE
```

Parameter ID.

**Description:**

codec type. See ADS ADS\_TEL\_PARAM\_CODER\_TYPE\_\*  
defines.

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_EVT_PKT
```

Parameter ID.

**Description:**

packet-loss report

**ADS\_TEL\_PARAM\_ON** -  
enable

**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_EVT_PKTCHNG
```

### 3.8 Define Documentation

Parameter ID.

**Description:**

report payload type change

**ADS\_TEL\_PARAM\_ON** -

enable

**ADS\_TEL\_PARAM\_OFF** -

disable

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_AUTOSW
```

Parameter ID.

**Description:**

Auto coder switch mask, Or'd bits. See ADS\_TEL\_DEC\_AUTOSW\_\* defines.

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_G726_40_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.726-40

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_G726_32_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.726-32

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_G726_24_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.726-24

**See also:**



### 3.8 Define Documentation

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_G726_16_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.726-16

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_G726_PACK
```

Parameter ID.

**Description:**

G.726 packing format  
**ADS\_TEL\_PARAM\_G726\_PACK\_LSB**  
**ADS\_TEL\_PARAM\_G726\_PACK\_MSB**

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_G729_1_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.729.1

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_G722_64_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.722-64

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_G722_56_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.722-56

### 3.8 Define Documentation

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_DEC_G722_48_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.722-48

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_VAD
```

Parameter ID.

**Description:**

VAD flag  
**ADS\_TEL\_PARAM\_ON** -  
enable  
**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_CTYPE
```

Parameter ID.

**Description:**

codec type. See ADS ADS\_TEL\_PARAM\_CODER\_TYPE\_\*  
defines.

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_MFPP
```

Parameter ID.

**Description:**

number of frames per  
packet

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_EVT_PKT
```

Parameter ID.

### 3.8 Define Documentation

**Description:**

packet-loss report

**ADS\_TEL\_PARAM\_ON** -

enable

**ADS\_TEL\_PARAM\_OFF** -

disable

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_G723_RATE
```

Parameter ID.

**Description:**

G723 encoding rate either 5.3k

or 6.3k

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_G726_40_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for

G.726-40

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_G726_32_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for

G.726-32

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_G726_24_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for

G.726-24

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

### 3.8 Define Documentation

```
#define ADS_TEL_PARMID_ENC_G726_16_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.726-16

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_G726_PACK
```

Parameter ID.

**Description:**

G.726 packing format  
**ADS\_TEL\_PARAM\_G726\_PACK\_LSB**  
**ADS\_TEL\_PARAM\_G726\_PACK\_MSB**

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_G729_1_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.729.1

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_G729_1_RATE
```

Parameter ID.

**Description:**

G729.1 encoding rate

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_G722_64_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.722-64

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

### 3.8 Define Documentation

```
#define ADS_TEL_PARMID_ENC_G722_56_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.722-56

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_ENC_G722_48_RTP_PLD
```

Parameter ID.

**Description:**

RTP payload type for  
G.722-48

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TG_VOL
```

Parameter ID.

**Description:**

volume adjustment

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TG_FSK_MOD
```

Parameter ID.

**Description:**

FSK Modulation Type  
**ADS\_TEL\_PARAM\_TG\_FSK\_V23**  
**ADS\_TEL\_PARAM\_TG\_FSK\_B202**

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TG_FSK_CS
```

Parameter ID.

**Description:**

CS bit length

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

### 3.8 Define Documentation

```
#define ADS_TEL_PARMID_TG_FSK_MARK
```

Parameter ID.

**Description:**

MARK bit length

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TG_FSK_RATE
```

Parameter ID.

**Description:**

Baud rate

**ADS\_TEL\_PARAM\_TG\_FSK\_R1200**  
**ADS\_TEL\_PARAM\_TG\_FSK\_R600**  
**ADS\_TEL\_PARAM\_TG\_FSK\_R300**  
**ADS\_TEL\_PARAM\_TG\_FSK\_R150**  
**ADS\_TEL\_PARAM\_TG\_FSK\_R75**

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TG_RFC2833
```

Parameter ID.

**Description:**

RFC2833 flag

**ADS\_TEL\_PARAM\_ON** -  
enable

**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TG_FSK_POSTMK
```

Parameter ID.

**Description:**

post mark bit length

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TG_PCM_MODE
```

Parameter ID.

**Description:**

WB or NB mode in the

### 3.8 Define Documentation

PCM line  
**ADS\_TEL\_PARAM\_WB**  
**ADS\_TEL\_PARAM\_NB**

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_TC
```

Parameter ID.

**Description:**

Tone clamping flag

**ADS\_TEL\_PARAM\_ON** -  
enable

**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_TC_FRAMES
```

Parameter ID.

**Description:**

Number of frames to buffer in TC  
buffer

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_RPT_EVENTS
```

Parameter ID.

**Description:**

report event

**ADS\_TEL\_PARAM\_TD\_RPT\_OFF**

**ADS\_TEL\_PARAM\_TD\_RPT\_TONE\_ON**

**ADS\_TEL\_PARAM\_TD\_RPT\_TONE\_OFF**

**ADS\_TEL\_PARAM\_TD\_RPT\_TONE\_ON\_OFF**

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_RFC2833E_ENABLE
```

Parameter ID.

**Description:**

RFC2833E flag

**ADS\_TEL\_PARAM\_ON** -  
enable

**ADS\_TEL\_PARAM\_OFF** -

### 3.8 Define Documentation

disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_RFC2833E_UPDATERATE
```

Parameter ID.

**Description:**

Update rate of RFC2833 pkts. (in  
10-ms units)

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_RFC2833E_NUMEOE
```

Parameter ID.

**Description:**

End of Event  
Redundancy

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_RFC2833E_NUMBOE
```

Parameter ID.

**Description:**

Begin of Event  
Redundancy

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_RFC2833E_AUDIOSUPPRESS
```

Parameter ID.

**Description:**

Suppress encoder when sending RFC2833 RTP packets  
**ADS\_TEL\_PARAM\_ON** - enable  
**ADS\_TEL\_PARAM\_OFF** - disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

**Note:**

This parameter is not currently supported and audio data is always transmitted.



### 3.8 Define Documentation

```
#define ADS_TEL_PARMID_TD_RFC2833E_PAYLOADTYPE
Parameter ID.
```

**Description:**

RFC2833 payload type

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_FSK_CS
Parameter ID.
```

**Description:**

minimum CS-bit length required by FSK  
receiver

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_FSK_MARK
Parameter ID.
```

**Description:**

minimum mark-bit length required by FSK  
receiver

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_FSK_STOP
Parameter ID.
```

**Description:**

extra stop-bit length allowed between  
the data

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_FSK_RATE
Parameter ID.
```

**Description:**

FSK receiver Baud rate

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_FSK_ADAPT_THRESHOLD
```

### 3.8 Define Documentation

Parameter ID.

**Description:**

FSK detector with adaptive or non-adaptive threshold

**ADS\_TEL\_PARAM\_TD\_NON\_ADAPT\_THRES**

**ADS\_TEL\_PARAM\_TD\_ADAPT\_THRES**

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_FSK_RX_MIN_BYTES
```

Parameter ID.

**Description:**

Minimum number of bytes to be received by the FSK detector

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_TD_ENABLE
```

Parameter ID.

**Description:**

Completely enable/disable the Tone Detector

**ADS\_TEL\_PARAM\_ON** - enable

**ADS\_TEL\_PARAM\_OFF** - disable

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_PLY_VOL
```

Parameter ID.

**Description:**

Player Volume control in dB

**See also:**

**adsTelEpCompParamsSet()**

**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_PLY_MODE
```

Parameter ID.

**Description:**

Player Mode

**ADS\_TEL\_PARAM\_WB**

**ADS\_TEL\_PARAM\_NB**

**See also:**

### 3.9 Typedef Documentation

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_SRC_ENABLE
```

Parameter ID.

**Description:**

SRC Enable

**ADS\_TEL\_PARAM\_ON** -  
enable

**ADS\_TEL\_PARAM\_OFF** -  
disable

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

```
#define ADS_TEL_PARMID_SRC_MODE
```

Parameter ID.

**Description:**

SRC Mode

**ADS\_TEL\_PARAM\_SRC\_UPSAMPLE**

**ADS\_TEL\_PARAM\_SRC\_DOWNSAMPLE**

**See also:**

**adsTelEpCompParamsSet()**  
**adsTelEpCompParamsGet().**

---

## 3.9 Typedef Documentation

```
typedef unsigned int AdsTelEpHandle_t
```

Endpoint Handle.

**Description:**

The handle used for controlling an endpoint instance.

**Purpose:**

Used to determine the endpoint instance being accessed.

**See also:**

**adsTelEpCreate()**  
**adsTelEpDestroy().**

```
typedef short AdsTelMediaHandle_t
```

Media Handle.

**Description:**

The handle used for playing an audio file.

### 3.10 Enumeration Type Documentation

**Purpose:**

Used to identify registered wav file.

**See also:**

**adsTelRegisterWavFile()**  
**adsTelEpPlayAudio().**

```
typedef struct AdsTelMediaDesc_s AdsTelMediaDesc_t
```

Audio segment to play.

**Description:**

This structure is used to define the audio segments to play. It is passed as a parameter to **adsTelEpPlayAudio()**.

**Purpose:**

Parameters of the audio segment to play.

**See also:**

**AdsTelMediaHandle\_t**  
**adsTelEpPlayAudio().**

---

## 3.10 Enumeration Type Documentation

```
enum AdsTelGeo_t
```

Call Progress Tone Geographical Location Settings.

**Description:**

Selects the geographical specific call progress tones. See **AdsTelInit\_t**.

**Enumerator:**

|                              |                                                       |
|------------------------------|-------------------------------------------------------|
| <i>ADS_TEL_GEO_PROC</i>      | Peoples Republic of China call progress tone setting. |
| <i>ADS_TEL_GEO_JAPAN</i>     | Japanese call progress tone setting.                  |
| <i>ADS_TEL_GEO_USA</i>       | United States of America call progress tone setting.  |
| <i>ADS_TEL_GEO_DELIMITER</i> | delimiter for error checks.                           |

```
enum AdsTelCallPrior_t
```

AdsTel call/endpoint priority.

**Description:**

specifies the priority of an endpoint/router. A 911 priority endpoint/router will always be set up. If needs be a low priority endpoint/router will be destroyed to accomodate the creation of the 911 priority endpoint/router.

**Note:**

Call Priority is not currently supported. All calls must be configured with **ADS\_TEL\_CALL\_PRIOR\_NORMAL**.

**Enumerator:**

|                                  |                      |
|----------------------------------|----------------------|
| <i>ADS_TEL_CALL_PRIOR_NORMAL</i> | normal call priority |
| <i>ADS_TEL_CALL_PRIOR_911</i>    | 911 call priority    |

### 3.10 Enumeration Type Documentation

*ADS\_TEL\_CALL\_PRIOR\_DELIMITER* delimiter for error checks.

#### enum **AdsTelPipeDir\_t**

AdsTel endpoint pipeline direction.

##### **Description:**

Select a pipeline within an endpoint.

##### **Enumerator:**

|                                   |                             |
|-----------------------------------|-----------------------------|
| <i>ADS_TEL_PIPE_DIR_IN</i>        | inbound pipeline.           |
| <i>ADS_TEL_PIPE_DIR_OUT</i>       | outbound pipeline.          |
| <i>ADS_TEL_PIPE_DIR_DELIMITER</i> | delimiter for error checks. |

#### enum **AdsTelPipeConType\_t**

AdsTel endpoint connection type.

##### **Description:**

Selects the type of the endpoint being created. See **adsTelEpCreate()**.

##### **Enumerator:**

|                                         |                             |
|-----------------------------------------|-----------------------------|
| <i>ADS_TEL_PIPE_CONN_TYPE_PCM</i>       | PCM pipeline.               |
| <i>ADS_TEL_PIPE_CONN_TYPE_IP</i>        | IP pipeline.                |
| <i>ADS_TEL_PIPE_CONN_TYPE_DELIMITER</i> | delimiter for error checks. |

#### enum **AdsTelPcmData\_t**

PCM data type .

##### **Description:**

Selects the sampling format for the PCM endpoint being created. See **adsTelEpCreate()**.

##### **Enumerator:**

|                                       |                                                |
|---------------------------------------|------------------------------------------------|
| <i>ADS_TEL_PCM_S_BE_16B_WB_LINEAR</i> | signed, Big Endian, 16 bit, 16 KHz, Linear.    |
| <i>ADS_TEL_PCM_S_BE_16B_NB_LINEAR</i> | signed, Big Endian, 8 bit, 8 KHz, Linear.      |
| <i>ADS_TEL_PCM_S_LE_16B_WB_LINEAR</i> | signed, Little Endian, 16 bit, 16 KHz, Linear. |
| <i>ADS_TEL_PCM_S_LE_16B_NB_LINEAR</i> | signed, Little Endian, 8 bit, 8 KHz, Linear.   |

#### enum **AdsTelEpConStatus\_t**

AdsTel endpoint connection status.

##### **Description:**

specifies the connection status of endpoint inbound and outbound pipelines. See **adsTelEpConStatusSet()**.

##### **Enumerator:**

|                             |                                            |
|-----------------------------|--------------------------------------------|
| <i>ADS_TEL_EP_CONN_NONE</i> | Endpoint pipelines disconnected.           |
| <i>ADS_TEL_EP_CONN_IN</i>   | Endpoint inbound pipeline connected only.  |
| <i>ADS_TEL_EP_CONN_OUT</i>  | Endpoint outbound pipeline connected only. |

### 3.11 Function Documentation

*ADS\_TEL\_EP\_CONN\_BOTH* Endpoint inbound and outbound pipelines connected.  
*ADS\_TEL\_EP\_CONN\_DELIMITER* delimiter for error checks.

#### enum **AdsTelTdCategory\_t**

AdsTel Tone Detector tone category.

#### **Description:**

specifies the tone category for the Tone Detector. See **adsTelTdAmplitudeRangeSet()**.

#### **Enumerator:**

*ADS\_TEL\_TD\_CAT\_DTMF* Tone Detector DTMF category.  
*ADS\_TEL\_TD\_CAT\_DELIMITER* delimiter for error checks.

---

## 3.11 Function Documentation

```
AdsTelStatus_t adsTelTgToneBuild (unsigned int toneld,
 unsigned int numSegs,
 AdsTelToneSeg_t * pToneSegs
)
```

Add a tone to the Tone Generator.

#### **Description:**

This function adds a new tone which can be played by the tone generator components.

#### **Context:**

Calling function thread.

#### **Assumptions:**

None.

#### **Side-Effects:**

None.

#### **Blocking:**

Yes.

#### **Reentrant:**

No.

#### **Thread-safe:**

Yes.

#### **Parameters:**

*toneld* [IN] - The tone ID to use. The allowable range is 0-255;  
*numSegs* [IN] - The number of segments in the tone.  
*pToneSegs* [IN] - A pointer to an array of the tone segments.

#### **Return values:**

*ADS\_TEL\_STATUS\_SUCCESS* The function executed successfully.  
*ADS\_TEL\_STATUS\_FAIL* The function did not execute successfully.

#### **Precondition:**

### 3.11 Function Documentation

The framework must have been previously initialized.

**Postcondition:**

None.

**Note:**

If a call progress tone ID is used then the call progress tone will be overwritten with the current tone information.

**See also:**

None.

```
AdsTelStatus_t adsTelTdToneBuild (unsigned int toneld,
 AdsTelToneInfo_t * pToneInfo
)
```

Build a Tone Detector Tone.

**Description:**

This function adds a new tone which can be detected by the tone detector components. This API can only be called before the framework has been initialized.

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

Yes.

**Reentrant:**

No.

**Thread-safe:**

Yes.

**Parameters:**

*toneld* [IN] - The tone ID to use. The allowable range is 16-255;

*pToneInfo* [IN] - A pointer to the tone detection criterion information.

**Return values:**

*ADS\_TEL\_STATUS\_SUCCESS* The function executed successfully.

*ADS\_TEL\_STATUS\_FAIL* The function did not execute successfully.

**Precondition:**

The framework must be uninitialized.

**Postcondition:**

None.

**Note:**

None.

### 3.11 Function Documentation

**See also:**

None.

```
AdsTelStatus_t adsTelTdAmplitudeRangeSet (AdsTelTdCategory_t category,
 int ampMinF0,
 int ampMaxF0,
 int ampMinF1,
 int ampMaxF1
)
```

Set the amplitude detection range for the Tone Detector.

**Description:**

The Tone Detector is able to detect the pre-defined DTMF tones in the full amplitude level range of +3 ~ -45 dBm. The application can use this function to set a specific amplitude range. Only the signals within this amplitude range can be detected as the DTMF tones. This API can only be called before the framework has been initialized.

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

Yes.

**Reentrant:**

No.

**Thread-safe:**

Yes.

**Parameters:**

*category* [IN] - The tone ID category. Only DTMF is currently supported.  
*ampMinF0* [IN] - Minimum amplitude of the low frequency, +3 ~ -45 in dBm.  
*ampMaxF0* [IN] - Maximum amplitude of the low frequency, +3 ~ -45 in dBm.  
*ampMinF1* [IN] - Minimum amplitude of the high frequency, +3 ~ -45 in dBm.  
*ampMaxF1* [IN] - Maximum amplitude of the high frequency, +3 ~ -45 in dBm.

**Return values:**

*ADS\_TEL\_STATUS\_SUCCESS* The function executed successfully.  
*ADS\_TEL\_STATUS\_FAIL* The function did not execute successfully.

**Precondition:**

The framework must be uninitialized.

**Postcondition:**

None.

**Note:**

None.



### 3.11 Function Documentation

**See also:**

None.

```
AdSTelStatus_t adSTelTgDtmfParamSet (int toneOn,
 int toneOff,
 int ampdBm
)
```

Set the tone parameters for the Tone Generator.

**Description:**

The DTMF tone generation has the default parameters of 100 ms tone-on and tone-off duration and -3dBm level. This function allows the users to change the default parameters. This API can only be called after the framework has been initialized.

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

Yes.

**Reentrant:**

No.

**Thread-safe:**

Yes.

**Parameters:**

*toneOn* [IN] - tone on duration in ms. Range 1 ~ FFFFFFFF.

*toneOff* [IN] - tone off duration in ms. Range 1 ~ FFFFFFFF.

*ampdBm* [IN] - total tone level in dBm, must be in 0 ~ -45 range.

**Return values:**

*ADS\_TEL\_STATUS\_SUCCESS* The function executed successfully.

*ADS\_TEL\_STATUS\_FAIL* The function did not execute successfully.

**Precondition:**

The framework must be initialized.

**Postcondition:**

None.

**Note:**

None.

**See also:**

None.

```

AdsTelStatus_t adsTelEpCreate (unsigned int * pTransId,
 AdsTelPipeType_t inPipe,
 AdsTelPipeType_t outPipe,
 AdsTelPcmData_t pcmDataType,
 adsTelEpDataCb_t inCb,
 adsTelEpDataCb_t outCb,
 AdsTelCallPrior_t priority,
 AdsTelRtrHandle_t rtrHandle,
 AdsTelEpHandle_t * pEpHandle
)

```

Create an endpoint.

#### Description:

This function is responsible for creating an endpoint and connecting it to a router. The endpoint can be either PCM or IP. The components in both the inbound and outbound pipelines will be initialised with default values. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_SUCCESS**

**ADS\_TEL\_STATUS\_FAIL.**

See **adsTelFwCb\_t** for the return buffer format for each return status.

#### Context:

Calling function thread.

#### Assumptions:

None.

#### Side-Effects:

None.

#### Blocking:

No.

#### Reentrant:

Yes.

#### Thread-safe:

Yes.

#### Parameters:

|                    |                                                                                                                                                                                                                           |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pTransId</i>    | [OUT] - Tracking number for the API return call.                                                                                                                                                                          |
| <i>inPipe</i>      | [IN] - Inbound pipeline type.                                                                                                                                                                                             |
| <i>outPipe</i>     | [IN] - Outbound pipeline type.                                                                                                                                                                                            |
| <i>pcmDataType</i> | [IN] - The PCM data input format. This parameter is only valid for PCM endpoints and can be set to NULL for IP endpoints.                                                                                                 |
| <i>inCb</i>        | [IN] - The function that the framework will call when requesting input data.                                                                                                                                              |
| <i>outCb</i>       | [IN] - The function that the framework will call when providing output data.                                                                                                                                              |
| <i>priority</i>    | [IN] - Priority of setup. A priority of 911 means that the if the max number of endpoints is already setup then one will be destroyed to create this one. This functionality is not currently supported in the framework. |
| <i>rtrHandle</i>   | [IN] - The handle of the router that the pipeline connect to.                                                                                                                                                             |
| <i>pEpHandle</i>   | [OUT] - The handle for the pipeline.                                                                                                                                                                                      |

#### Return values:

**ADS\_TEL\_STATUS\_SUCCESS**

|                                           |                                                                                                |
|-------------------------------------------|------------------------------------------------------------------------------------------------|
|                                           | The function executed successfully and the operation was sent to the framework for processing. |
| <code>ADS_TEL_STATUS_FAIL</code>          | The function did not execute successfully.                                                     |
| <code>ADS_TEL_STATUS_INVALID_PARAM</code> | An invalid input parameter was supplied.                                                       |

**Precondition:**

The associated router must have already been created.

**Postcondition:**

None.

**Note:**

Both the inbound and outbound callbacks will be disconnected by default on endpoint creation. See `adsTelEpConStatusSet()`.

**See also:**

`adsTelRouterCreate()`.

```

AdsTelStatus_t adsTelEpConStatusSet (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelEpConStatus_t conStatus
)

```

Set the connection status of an endpoint.

**Description:**

This function enables or disables the outbound and inbound callbacks for an endpoint. The reply from the framework will be posted via `adsTelFwCb_t`. The possible `adsTelFwCb_t` return statuses are:

`ADS_TEL_STATUS_SUCCESS`

`ADS_TEL_STATUS_FAIL`

See `adsTelFwCb_t` for the return buffer format for each return status.

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Parameters:**

`pTransId` [OUT] - Tracking number for the API return call.

`epHandle` [IN] - The endpoint to set the connection status for.

`conStatus` [IN] - The status to set the connection to.

**Blocking:**

No.

**Reentrant:**

Yes.

**Thread-safe:**

Yes.

### 3.11 Function Documentation

**Return values:**

**ADS\_TEL\_STATUS\_SUCCESS** on success or else the appropriate error code.

**Precondition:**

The adsTel framework must be initialized.

**Postcondition:**

None.

**Note:**

None.

**See also:**

**adsTelInit()**.

```
AdsTelStatus_t adsTelEpCompParamsSet (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compId,
 unsigned int numParams,
 unsigned int * pParamIDs,
 int * pParamVals
)
```

Set multiple parameters for an endpoint component.

**Description:**

This function sets a number of component parameters up to **ADS\_TEL\_MAX\_PARAMS**. When an endpoint is created all component parameters are initialised with default values. This function can be used to change the configuration of any component. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_SUCCESS**

**ADS\_TEL\_STATUS\_FAIL**

See **adsTelFwCb\_t** for the return buffer format for each return status.

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

No.

**Reentrant:**

Yes.

**Thread-safe:**

Yes.

**Parameters:**

*pTransId* [IN] - Tracking number for the API return call.  
*epHandle* [IN] - Endpoint handle.  
*pipeDir* [IN] - Pipeline direction (in/out).

### 3.11 Function Documentation

*compld* [IN] - The component location within the the pipeline. The index is 0 based.  
*numParams* [IN] - The number of parameters being set.  
*pParamIDs* [IN] - Pointer to an array of parameter IDs of max size **ADS\_TEL\_MAX\_PARAMS**.  
*pParamVals* [IN] - Pointer to an array of parameter values of max size **ADS\_TEL\_MAX\_PARAMS**.

#### Return values:

*ADS\_TEL\_STATUS\_SUCCESS* The function executed successfully and the operation was sent to the framework for processing.  
*ADS\_TEL\_STATUS\_FAIL* The function did not execute successfully.  
*ADS\_TEL\_STATUS\_INVALID\_PARAM* An invalid input parameter was supplied.

#### Precondition:

The endpoint must have already been created.

#### Postcondition:

None.

#### Note:

None.

#### See also:

**adsTelEpCreate()**.

```
AdsTelStatus_t adsTelEpCompParamsGet (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compld
)
```

Get all parameters values for an endpoint component.

#### Description:

This function return all the parameter values for a component via the framework callback. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_GET\_PARAMS**

See **adsTelFwCb\_t** for the return buffer format for each return status.

#### Context:

Calling function thread.

#### Assumptions:

None.

#### Side-Effects:

None.

#### Blocking:

No.

#### Reentrant:

Yes.

#### Thread-safe:

Yes.

### 3.11 Function Documentation

#### Parameters:

*pTransld* [IN] - Tracking number for the API return call.  
*epHandle* [IN] - Endpoint handle.  
*pipeDir* [IN] - Pipeline direction (in/out).  
*compld* [IN] - The component location within the the pipeline. The index is 0 based.

#### Return values:

*ADS\_TEL\_STATUS\_SUCCESS* The function executed successfully and the operation was sent to the framework for processing.  
*ADS\_TEL\_STATUS\_FAIL* The function did not execute successfully.  
*ADS\_TEL\_STATUS\_INVALID\_PARAM* An invalid input parameter was supplied.

#### Precondition:

The endpoint must have already been created.

#### Postcondition:

None.

#### Note:

None.

#### See also:

**adsTelEpCreate()**  
**adsTelEpCompParamsSet().**

```
AdsTelStatus_t adsTelEpTonePlay (unsigned int * pTransld,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compld,
 unsigned int numTones,
 unsigned int * pTonelds
)
```

Play a tone or multiple tones on a tone generator component.

#### Description:

This function will play out all the supplied tones on a tone generator. Note that if a non DTMF tone is played it will play continuously until the tone generator is explicitly stopped with **adsTelEpCompStop()**. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_TG\_CMPLT**

**ADS\_TEL\_STATUS\_FAIL**

See **adsTelFwCb\_t** for the return buffer format for each return status.

#### Context:

Calling function thread.

#### Assumptions:

None.

#### Side-Effects:

None.

#### Blocking:

No.

### 3.11 Function Documentation

**Reentrant:**

Yes.

**Thread-safe:**

Yes.

**Parameters:**

*pTransId* [OUT] - Tracking number for the API return call.  
*epHandle* [IN] - Endpoint handle.  
*pipeDir* [IN] - Pipeline direction (in/out).  
*compld* [IN] - The component location within the the pipeline. The index is 0 based.  
*numTones* [IN] - The number of tones to be played up to a maximum of **ADS\_TEL\_MAX\_TONES**.  
*pToneIds* [IN] - A pointer to an array (of size numTones) of the tone IDs to be played. See for a list of the valid tone IDs.

**Return values:**

**ADS\_TEL\_STATUS\_SUCCESS** The function executed successfully and the operation was sent to the framework for processing.  
**ADS\_TEL\_STATUS\_FAIL** The function did not execute successfully.  
**ADS\_TEL\_STATUS\_INVALID\_PARAM** An invalid input parameter was supplied.

**Precondition:**

The endpoint must have already been created.

**Postcondition:**

None.

**Note:**

None.

**See also:**

**adsTelEpCreate()**.

```
AdsTelStatus_t adsTelEpFskPlay (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compld,
 unsigned int numBytes,
 char * pFskData
)
```

Play FSK Data on a tone generator component.

**Description:**

This function will play out all the supplied FSK data bytes on a tone generator. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_TG\_CMPLT**

**ADS\_TEL\_STATUS\_FAIL**

See **adsTelFwCb\_t** for the return buffer format for each return status.

**Context:**

Calling function thread.

**Assumptions:**

None.

### 3.11 Function Documentation

**Side-Effects:**

None.

**Blocking:**

No.

**Reentrant:**

Yes.

**Thread-safe:**

Yes.

**Parameters:**

*pTransId* [OUT] - Tracking number for the API return call.  
*epHandle* [IN] - Endpoint handle.  
*pipeDir* [IN] - Pipeline direction (in/out).  
*compld* [IN] - The component location within the the pipeline. The index is 0 based.  
*numBytes* [IN] - The number of FSK Bytes to be played up to a maximum of **ADS\_TEL\_MAX\_FSK\_BYTES**.  
*pFskData* [IN] - A pointer to an array (of size numBytes) of the FSK Bytes to be sent.

**Return values:**

**ADS\_TEL\_STATUS\_SUCCESS** The function executed successfully and the operation was sent to the framework for processing.  
**ADS\_TEL\_STATUS\_FAIL** The function did not execute successfully.  
**ADS\_TEL\_STATUS\_INVALID\_PARAM** An invalid input parameter was supplied.

**Precondition:**

The endpoint must have already been created.

**Postcondition:**

None.

**Note:**

None.

**See also:**

**adsTelEpCreate()**.

```
AdsTelStatus_t adsTelEpToneRecv (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compld,
 unsigned int totalTimeout,
 unsigned int firstDigitTimeout,
 unsigned int interDigitTimeout,
 unsigned int termDigit,
 unsigned int numDigits
)
```



### 3.11 Function Documentation

Receive a tone or multiple tones on a tone detector component.

**Description:**

This function will attempt to detect the number of expected tones on a tone detector unless one of the timeout conditions is met. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_TD\_RCV\_CMPLT**

**ADS\_TEL\_STATUS\_FAIL**. See **adsTelFwCb\_t** for the return buffer format for each return status.

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

No.

**Reentrant:**

Yes.

**Thread-safe:**

Yes.

**Parameters:**

|                          |                                                                          |
|--------------------------|--------------------------------------------------------------------------|
| <i>pTransId</i>          | [OUT] - Tracking number for the API return call.                         |
| <i>epHandle</i>          | [IN] - Endpoint handle.                                                  |
| <i>pipeDir</i>           | [IN] - Pipeline direction (in/out).                                      |
| <i>compld</i>            | [IN] - The component location within the pipeline. The index is 0 based. |
| <i>totalTimeout</i>      | [IN] - Total time out (in 10ms units).                                   |
| <i>firstDigitTimeout</i> | [IN] - First digit time out (in 10ms units).                             |
| <i>interDigitTimeout</i> | [IN] - Inter digit time out (in 10ms units).                             |
| <i>termDigit</i>         | [IN] - OR'd terminate digit bits.                                        |
| <i>numDigits</i>         | [IN] - number of digits to receive.                                      |

**Return values:**

|                                     |                                                                                                |
|-------------------------------------|------------------------------------------------------------------------------------------------|
| <i>ADS_TEL_STATUS_SUCCESS</i>       | The function executed successfully and the operation was sent to the framework for processing. |
| <i>ADS_TEL_STATUS_FAIL</i>          | The function did not execute successfully.                                                     |
| <i>ADS_TEL_STATUS_INVALID_PARAM</i> | An invalid input parameter was supplied.                                                       |

**Precondition:**

The endpoint must have already been created.

**Postcondition:**

None.

**Note:**

None.

**See also:**

### 3.11 Function Documentation

**adsTelFwCb\_t**  
**adsTelEpCreate().**

```
AdsTelStatus_t adsTelEpFskRecv (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compld,
 unsigned int timeout
)
```

Receive FSK bytes on a tone detector component.

#### Description:

This function will attempt to detect FSK data on a tone detector component up to **ADS\_TEL\_MAX\_FSK\_BYTES** unless the operation times out first. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are: **ADS\_TEL\_STATUS\_TD\_RCV\_FSK\_CMPLT** **ADS\_TEL\_STATUS\_FAIL**. See **adsTelFwCb\_t** for the return buffer format for each return status.

#### Context:

Calling function thread.

#### Assumptions:

None.

#### Side-Effects:

None.

#### Blocking:

No.

#### Reentrant:

Yes.

#### Thread-safe:

Yes.

#### Parameters:

*pTransId* [OUT] - Tracking number for the API return call.  
*epHandle* [IN] - Endpoint handle.  
*pipeDir* [IN] - Pipeline direction (in/out).  
*compld* [IN] - The component location within the the pipeline. The index is 0 based.  
*timeout* [IN] - Total time out (in 10ms units).

#### Return values:

|                                     |                                                                                                |
|-------------------------------------|------------------------------------------------------------------------------------------------|
| <b>ADS_TEL_STATUS_SUCCESS</b>       | The function executed successfully and the operation was sent to the framework for processing. |
| <b>ADS_TEL_STATUS_FAIL</b>          | The function did not execute successfully.                                                     |
| <b>ADS_TEL_STATUS_INVALID_PARAM</b> | An invalid input parameter was supplied.                                                       |

#### Precondition:

The endpoint must have already been created.

#### Postcondition:

None.

### 3.11 Function Documentation

**Note:**

None.

**See also:**

**adsTelFwCb\_t**  
**adsTelEpCreate().**

```
AdsTelStatus_t adsTelEpPlayAudio (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compld,
 AdsTelMediaDesc_t * pSegments,
 short nSegs
)
```

Play an audio file.

**Description:**

This function will play an audio file on a player component. The audio file to be played must have been registered with **adsTelRegisterWavFile()**. To stop the audio playing use **adsTelEpCompStop()**. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible return statuses are:

**ADS\_TEL\_STATUS\_SUCCESS**  
**ADS\_TEL\_STATUS\_INVALID\_PARAM**  
**ADS\_TEL\_STATUS\_FAIL**

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

No.

**Reentrant:**

Yes.

**Thread-safe:**

Yes.

**Parameters:**

|                  |                                                                                                |
|------------------|------------------------------------------------------------------------------------------------|
| <i>pTransId</i>  | [OUT] - Tracking number for the API return call.                                               |
| <i>epHandle</i>  | [IN] - Endpoint handle.                                                                        |
| <i>pipeDir</i>   | [IN] - Pipeline direction (in/out).                                                            |
| <i>compld</i>    | [IN] - The component location within the the pipeline. The index is 0 based.                   |
| <i>pSegments</i> | [IN] - Pointer to an array of audio segments to play of max size <b>ADS_TEL_PLY_MAX_SEGS</b> . |
| <i>nSegs</i>     | [IN] - The number of segments to play.                                                         |

**Return values:**

|                               |                                                                                                |
|-------------------------------|------------------------------------------------------------------------------------------------|
| <b>ADS_TEL_STATUS_SUCCESS</b> | The function executed successfully and the operation was sent to the framework for processing. |
|-------------------------------|------------------------------------------------------------------------------------------------|

### 3.11 Function Documentation

**ADS\_TEL\_STATUS\_FAIL** The function did not execute successfully.  
**ADS\_TEL\_STATUS\_INVALID\_PARAM** An invalid input parameter was supplied.

**Precondition:**

The endpoint must have already been created.

**Postcondition:**

None.

**Note:**

None.

**See also:**

**adsTelEpCreate()**  
**adsTelRegisterWavFile()**  
**adsTelEpCompStop().**

```
AdsTelStatus_t adsTelRegisterWavFile (char * filename,
 AdsTelMediaHandle_t * pMediaHandle
)
```

Register a wav file.

**Description:**

This function will register a wav file into the ADS Telephony Framework. A maximum of 32 files can be registered. The possible return statuses are:

**ADS\_TEL\_STATUS\_SUCCESS**  
**ADS\_TEL\_STATUS\_INVALID\_FORMAT**  
**ADS\_TEL\_STATUS\_FAIL**

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

Yes.

**Reentrant:**

Yes.

**Thread-safe:**

Yes.

**Parameters:**

*filename* [IN] - The name of the wav file to register.  
*pMediaHandle* [OUT] - The media handle returned by the framework. This handle is valid if the function returns ADS\_TEL\_STATUS\_SUCCESS.

**Return values:**

**ADS\_TEL\_STATUS\_SUCCESS** The function executed successfully and the operation was sent to the framework for processing.

### 3.11 Function Documentation

**ADS\_TEL\_STATUS\_FAIL**      The function did not execute successfully.

**Precondition:**

The adsTel framework must be initialized.

**Postcondition:**

None.

**Note:**

None.

**See also:**

**adsTelEpPlayAudio()**  
**adsTelEpCompStop().**

```
AdsTelStatus_t adsTelEpCompStart (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compld
)
```

Start a component.

**Description:**

This function will start a component.

The reply from the framework will be posted via **adsTelFwCb\_t** The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_SUCCESS**

**ADS\_TEL\_STATUS\_FAIL**

See **adsTelFwCb\_t** for the return buffer format for each return status.

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

No.

**Reentrant:**

Yes.

**Thread-safe:**

Yes.

**Parameters:**

*pTransId* [OUT] - Tracking number for the API return call.

*epHandle* [IN] - Endpoint handle.

*pipeDir* [IN] - Pipeline direction (in/out).

*compld* [IN] - The component location within the the pipeline. The index is 0 based.

### 3.11 Function Documentation

**Return values:**

|                                     |                                                                                                |
|-------------------------------------|------------------------------------------------------------------------------------------------|
| <b>ADS_TEL_STATUS_SUCCESS</b>       | The function executed successfully and the operation was sent to the framework for processing. |
| <b>ADS_TEL_STATUS_FAIL</b>          | The function did not execute successfully.                                                     |
| <b>ADS_TEL_STATUS_INVALID_PARAM</b> | An invalid input parameter was supplied.                                                       |

**Precondition:**

The endpoint must have already been created.

**Postcondition:**

None.

**Note:**

This API is not currently supported by any component.

**See also:**

**adsTelFwCb\_t**  
**adsTelEpCreate()**  
**adsTelEpCompStop().**

```
AdsTelStatus_t adsTelEpCompStop (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compld
)
```

Stop a component.

**Description:**

This function will stop a component. It is only applicable to the following components:

Tone Detector  
Tone generator  
Player

The reply from the framework will be posted via **adsTelFwCb\_t** The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_SUCCESS**

**ADS\_TEL\_STATUS\_FAIL.**

See **adsTelFwCb\_t** for the return buffer format for each return status.

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

No.

**Reentrant:**

Yes.

**Thread-safe:**

Yes.

### 3.11 Function Documentation

#### Parameters:

*pTransId* [OUT] - Tracking number for the API return call.  
*epHandle* [IN] - Endpoint handle.  
*pipeDir* [IN] - Pipeline direction (in/out).  
*compld* [IN] - The component location within the the pipeline.

#### Return values:

*ADS\_TEL\_STATUS\_SUCCESS* The function executed successfully and the operation was sent to the framework for processing.  
*ADS\_TEL\_STATUS\_FAIL* The function did not execute successfully.  
*ADS\_TEL\_STATUS\_INVALID\_PARAM* An invalid input parameter was supplied.

#### Precondition:

The endpoint must have already been created.

#### Postcondition:

None.

#### Note:

None.

#### See also:

**adsTelFwCb\_t**  
**adsTelEpCreate()**  
**adsTelEpCompStart()**.

```
AdsTelStatus_t adsTelEpCompReset (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle,
 AdsTelPipeDir_t pipeDir,
 unsigned int compld
)
```

Reset a component.

#### Description:

This function will reset a component. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are:  
**ADS\_TEL\_STATUS\_SUCCESS**  
**ADS\_TEL\_STATUS\_FAIL**  
See **adsTelFwCb\_t** for the return buffer format for each return status.

#### Context:

Calling function thread.

#### Assumptions:

None.

#### Side-Effects:

None.

#### Blocking:

No.

#### Reentrant:

Yes.

### 3.11 Function Documentation

**Thread-safe:**

Yes.

**Parameters:**

*pTransId* [OUT] - Tracking number for the API return call.  
*epHandle* [IN] - Endpoint handle.  
*pipeDir* [IN] - Pipeline direction (in/out).  
*compld* [IN] - The component location within the the pipeline. The index is 0 based.

**Return values:**

*ADS\_TEL\_STATUS\_SUCCESS* The function executed successfully and the operation was sent to the framework for processing.  
*ADS\_TEL\_STATUS\_FAIL* The function did not execute successfully.  
*ADS\_TEL\_STATUS\_INVALID\_PARAM* An invalid input parameter was supplied.

**Precondition:**

The endpoint must have already been created.

**Postcondition:**

None.

**Note:**

This API is not currently supported by any components.

**See also:**

**adsTelFwCb\_t**  
**adsTelEpCreate()**

```
AdsTelStatus_t adsTelEpDestroy (unsigned int * pTransId,
 AdsTelEpHandle_t epHandle
)
```

Destroy an endpoint.

**Description:**

This function will remove an endpoint from its associated router and then destroy the endpoint. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_SUCCESS**

**ADS\_TEL\_STATUS\_FAIL**

See **adsTelFwCb\_t** for the return buffer format for each return status.

**Context:**

Calling function thread.

**Assumptions:**

None.

**Side-Effects:**

None.

**Blocking:**

No.

**Reentrant:**

Yes.



### 3.11 Function Documentation

**Thread-safe:**

Yes.

**Parameters:**

*pTransId* [OUT] - Tracking number for the API return call.

*epHandle* [IN] - The endpoint to destroy.

**Return values:**

*ADS\_TEL\_STATUS\_SUCCESS* The function executed successfully and the operation was sent to the framework for processing.

*ADS\_TEL\_STATUS\_FAIL* The function did not execute successfully.

*ADS\_TEL\_STATUS\_INVALID\_PARAM* An invalid input parameter was supplied.

**Precondition:**

The endpoint must have already been created.

**Postcondition:**

None.

**Note:**

None.

**See also:**

**adsTelFwCb\_t**

**adsTelEpCreate()**

## 4 ADS APIs for Router Allocation

### [ADS Telephony Framework API]

Collaboration diagram for ADS APIs for Router Allocation:



Router instance allocation and deallocation functions for the ADS Telephony Framework.

### 4.1 Typedefs

- typedef unsigned int **AdsTelRtrHandle\_t**  
Router Handle.

### 4.2 Enumerations

- enum **AdsTelRtrType\_t** {  
    **ADS\_TEL\_RTR\_NORMAL**,  
    **ADS\_TEL\_RTR\_CONF**,  
    **ADS\_TEL\_RTR\_LECTURE**,  
    **ADS\_TEL\_RTR\_DELIMITER**  
}  
    AdsTel router type.
- enum **AdsTelPipeType\_t** {  
    **ADS\_TEL\_PIPE\_IP\_OUT**,  
    **ADS\_TEL\_PIPE\_IP\_IN**,  
    **ADS\_TEL\_PIPE\_PCM\_OUT**,  
    **ADS\_TEL\_PIPE\_PCM\_IN**,  
    **ADS\_TEL\_PIPE\_DELIMITER**  
}  
    AdsTel pipeline types.

### 4.3 Functions

- **AdsTelStatus\_t adsTelRouterCreate** (unsigned int \*pTransId, **AdsTelCallPrior\_t** priority, **AdsTelRtrType\_t** rtrType, **AdsTelRtrHandle\_t** \*pRtrHandle)  
    Create a router instance.
  - **AdsTelStatus\_t adsTelRouterDestroy** (unsigned int \*pTransId, **AdsTelRtrHandle\_t** rtrHandle)  
    Destroy a router instance.
- 

### 4.4 Detailed Description

Router instance allocation and deallocation functions for the ADS Telephony Framework.

#### Purpose:

The functions in this group are responsible for all aspects of router allocation for the ADS Telephony Framework.

---

## 4.5 Typedef Documentation

typedef unsigned int **AdsTelRtrHandle\_t**

Router Handle.

**Description:**

The handle used for controlling an router instance.

**Purpose:**

Used to determine the router instance being accessed.

**See also:**

**adsTelRouterCreate()**  
**adsTelRouterDestroy().**

---

## 4.6 Enumeration Type Documentation

enum **AdsTelRtrType\_t**

AdsTel router type.

**Description:**

specifies the type of a router.

**Note:**

router type is not currently supported and all routers should be created with the ADS\_TEL\_RTR\_NORMAL type.

**Enumerator:**

|                              |                             |
|------------------------------|-----------------------------|
| <i>ADS_TEL_RTR_NORMAL</i>    | standard router.            |
| <i>ADS_TEL_RTR_CONF</i>      | conference router.          |
| <i>ADS_TEL_RTR_Lecture</i>   | lecture router.             |
| <i>ADS_TEL_RTR_DELIMITER</i> | delimiter for error checks. |

enum **AdsTelPipeType\_t**

AdsTel pipeline types.

**Description:**

specifies the type of a pipeline. See **adsTelEpCreate()**.

**Enumerator:**

|                             |                             |
|-----------------------------|-----------------------------|
| <i>ADS_TEL_PIPE_IP_OUT</i>  | IP outbound pipeline type.  |
| <i>ADS_TEL_PIPE_IP_IN</i>   | IP inbound pipeline type.   |
| <i>ADS_TEL_PIPE_PCM_OUT</i> | PCM outbound pipeline type. |
| <i>ADS_TEL_PIPE_PCM_IN</i>  | PCM inbound pipeline type.  |

*ADS\_TEL\_PIPE\_DELIMITER* delimiter for error checks.

## 4.7 Function Documentation

```
AdsTelStatus_t adsTelRouterCreate (unsigned int * pTransId,
 AdsTelCallPrior_t priority,
 AdsTelRtrType_t rtrType,
 AdsTelRtrHandle_t * pRtrHandle
)
```

Create a router instance.

### Description:

Create a router instance. By setting the priority to 911 the router will be set up even if the max number of routers has already been set up by deleting an existing router instance. The router type should be set to **ADS\_TEL\_RTR\_NORMAL** for calls of up to three endpoints. For conference calls of greater than three endpoints **ADS\_TEL\_RTR\_CONF** type should be used. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are:

**ADS\_TEL\_STATUS\_SUCCESS**

**ADS\_TEL\_STATUS\_FAIL**

See **adsTelFwCb\_t** for the return buffer format for each return status.

### Context:

Calling function thread.

### Assumptions:

None.

### Side-Effects:

None.

### Blocking:

No.

### Reentrant:

Yes.

### Thread-safe:

Yes.

### Parameters:

|                   |                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------|
| <i>pTransId</i>   | [OUT] - Tracks the API return call in non blocking mode. This can be set to NULL in blocking mode. |
| <i>priority</i>   | [IN] - The priority of the call.                                                                   |
| <i>rtrType</i>    | [IN] - The router type.                                                                            |
| <i>pRtrHandle</i> | [OUT] - The returned router handle.                                                                |

### Return values:

|                                     |                                                                                                |
|-------------------------------------|------------------------------------------------------------------------------------------------|
| <i>ADS_TEL_STATUS_SUCCESS</i>       | The function executed successfully and the operation was sent to the framework for processing. |
| <i>ADS_TEL_STATUS_FAIL</i>          | The function did not execute successfully.                                                     |
| <i>ADS_TEL_STATUS_INVALID_PARAM</i> | An invalid input parameter was supplied.                                                       |

### Precondition:

## 4.7 Function Documentation

The adsTel framework must be initialized.

### Postcondition:

None.

### Note:

Call priority is not currently supported and all calls should be created with **ADS\_TEL\_CALL\_PRIOR\_NORMAL**. Router type is not currently supported and all routers should be set up with **ADS\_TEL\_RTR\_NORMAL** router type setting.

### See also:

**adsTelFwCb\_t**  
**adsTelInit()**

```
AdsTelStatus_t adsTelRouterDestroy (unsigned int * pTransId,
 AdsTelRtrHandle_t rtrHandle
)
```

Destroy a router instance.

### Description:

Destroy a router instance. A router instance cannot be destroyed if there are still endpoints connected to the router. The reply from the framework will be posted via **adsTelFwCb\_t**. The possible **adsTelFwCb\_t** return statuses are: **ADS\_TEL\_STATUS\_SUCCESS**, **ADS\_TEL\_STATUS\_FAIL**. See **adsTelFwCb\_t** for the return buffer format for each return status.

### Context:

Calling function thread.

### Assumptions:

None.

### Side-Effects:

None.

### Blocking:

No.

### Reentrant:

Yes.

### Thread-safe:

Yes.

### Parameters:

*pTransId* [OUT] - Tracking number for the API return call.  
*rtrHandle* [IN] - The router to destroy.

### Return values:

|                                     |                                                                                                |
|-------------------------------------|------------------------------------------------------------------------------------------------|
| <b>ADS_TEL_STATUS_SUCCESS</b>       | The function executed successfully and the operation was sent to the framework for processing. |
| <b>ADS_TEL_STATUS_FAIL</b>          | The function did not execute successfully.                                                     |
| <b>ADS_TEL_STATUS_INVALID_PARAM</b> | An invalid input parameter was supplied.                                                       |

### Precondition:

## 4.7 Function Documentation

The router must have already been created.

**Postcondition:**

None.

**Note:**

None.

**See also:**

**adsTelFwCb\_t**  
**adsTelRouterCreate()**

# 5 ADS API and Event Callbacks

## [ADS Telephony Framework API]

Collaboration diagram for ADS API and Event Callbacks:



Functions and type definitions of the ADS Telephony Framework for API and events callbacks.

### 5.1 Data Structures

- struct **AdsTelRetBuf\_t**  
API return structure type.
- struct **AdsTelPacket\_t**  
ADS Framework Buffer format.
- struct **AdsTelPCMPacket\_t**  
ADS Framework Buffer format.
- struct **AdsTelRetAck\_s**  
**adsTelFwCb\_t** return buffer structure type.
- struct **AdsTelRetTg\_s**  
**adsTelFwCb\_t** return buffer structure type.
- struct **AdsTelRetTdFsk\_s**  
**adsTelFwCb\_t** return buffer structure type.
- struct **AdsTelRetTd\_s**  
**adsTelFwCb\_t** return buffer structure type.
- struct **AdsTelRetEv\_s**  
**adsTelFwCb\_t** return buffer structure type.
- struct **AdsTelRetGetParams\_s**  
**adsTelFwCb\_t** return buffer structure type.
- struct **AdsTelRetPlay\_s**  
**adsTelFwCb\_t** return buffer structure type.

### 5.2 Defines

- #define **ADS\_TEL\_MAX\_PAYLOAD\_SIZE**  
Maximum payload length.
- #define **ADS\_TEL\_RTP\_MD\_TYPE\_AUDIO**  
Audio packet packet payload type.
- #define **ADS\_TEL\_RTP\_MD\_TYPE\_RFC2833\_EVENT**  
RFC2833 Event packet payload type.

### 5.3 Typedefs

- typedef **AdsTelStatus\_t**(\* **adsTelEpDataCb\_t**)(**AdsTelEpHandle\_t** endPtHandle, void \*pBuffer1, void \*pBuffer2)  
adsTel Framework Data callback.
- typedef void(\* **adsTelFwCb\_t**)(unsigned int transId, void \*pData)  
adsTel Framework API and Event callback.
- typedef **AdsTelRetAck\_s** **AdsTelRetAck\_t**  
**adsTelFwCb\_t** return buffer structure type.
- typedef **AdsTelRetTg\_s** **AdsTelRetTg\_t**

## 5.3 Typedefs

- **adsTelFwCb\_t** return buffer structure type.
- typedef **AdsTelRetTdFsk\_s AdsTelRetTdFsk\_t**  
**adsTelFwCb\_t** return buffer structure type.
- typedef **AdsTelRetTd\_s AdsTelRetTd\_t**  
**adsTelFwCb\_t** return buffer structure type.
- typedef **AdsTelRetEv\_s AdsTelRetEv\_t**  
**adsTelFwCb\_t** return buffer structure type.
- typedef **AdsTelRetGetParams\_s AdsTelRetGetParams\_t**  
**adsTelFwCb\_t** return buffer structure type.
- typedef **AdsTelRetPlay\_s AdsTelRetPlay\_t**  
**adsTelFwCb\_t** return buffer structure type.

## 5.4 Enumerations

- enum **AdsTelEvent\_t** {  
    **ADS\_TEL\_EV\_TD\_TONE\_ON**,  
    **ADS\_TEL\_EV\_TD\_TONE\_OFF**,  
    **ADS\_TEL\_EV\_LOST\_PACKET**,  
    **ADS\_TEL\_EV\_DEC\_PACKET\_CHNG**  
}
- AdsTel event types.
- 

## 5.5 Detailed Description

Functions and type definitions of the ADS Telephony Framework for API and events callbacks.

### Purpose:

The functions and type definitions in this group are responsible for all aspects of API status and events reporting for the ADS Telephony Framework.

---

## 5.6 Data Structure Documentation

### 5.6.1 AdsTelRetBuf\_t Struct Reference

API return structure type.

#### 5.6.1.1 Data Fields

- **AdsTelStatus\_t status**  
the return status of the API call.
- void \* **buffer**  
return data of the API call.

#### 5.6.1.2 Detailed Description

API return structure type.

### Description:

Contains the status and associated data that is returned as part of an API call.

### Purpose:

Provide return data from an API call.



### 5.6.1 AdsTelRetBuf\_t Struct Reference

**See also:**

None.

#### 5.6.1.3 Field Documentation

**AdsTelStatus\_t AdsTelRetBuf\_t::status**

the return status of the API call.

**void\* AdsTelRetBuf\_t::buffer**

return data of the API call.

---

### 5.6.2 AdsTelPacket\_t Struct Reference

ADS Framework Buffer format.

#### 5.6.2.1 Data Fields

- unsigned short **channelID**  
channel ID
- unsigned short **payloadType**  
bit[0-6] payload type, bit[7] SID mark bit
- unsigned int **mediaType:4**  
media type
- unsigned int **payloadLen:12**  
payload length
- unsigned int **timeStamp**  
local or remote time stamp
- unsigned long **privateData1**  
Private data field untouched by the framework.
- unsigned long **privateData2**  
Private data field untouched by the framework.
- char **payload** [ADS\_TEL\_MAX\_PAYLOAD\_SIZE]  
payload

#### 5.6.2.2 Detailed Description

ADS Framework Buffer format.

**Description:**

The format of the buffer that is used to pass data between the ADS framework and the client application for IP data.

**Purpose:**

Format of the buffer used when passing data via the data callbacks.

**See also:**

**adsTelEpDataCb\_t()**.

#### 5.6.2.3 Field Documentation

unsigned short **AdsTelPacket\_t::channelID**

channel ID

### 5.6.2 AdsTelPacket\_t Struct Reference

unsigned short **AdsTelPacket\_t::payloadType**

bit[0-6] payload type, bit[7] SID mark bit

unsigned int **AdsTelPacket\_t::mediaType**

media type

unsigned int **AdsTelPacket\_t::payloadLen**

payload length

unsigned int **AdsTelPacket\_t::timeStamp**

local or remote time stamp

unsigned long **AdsTelPacket\_t::privateData1**

Private data field untouched by the framework.

unsigned long **AdsTelPacket\_t::privateData2**

Private data field untouched by the framework.

char **AdsTelPacket\_t::payload**[ADS\_TEL\_MAX\_PAYLOAD\_SIZE]

payload

---

### 5.6.3 AdsTelPCMPacket\_t Struct Reference

ADS Framework Buffer format.

#### 5.6.3.1 Data Fields

- unsigned short **channelID**  
channel ID
- unsigned short **payloadType**  
bit[0-6] payload type, bit[7] SID mark bit
- unsigned short **payloadLength**  
payload length in bytes
- unsigned int **timestamp**  
frame time stamp
- unsigned short **payloadValid**  
data is valid
- unsigned long **privateData1**  
Private data field untouched by the framework.
- unsigned long **privateData2**  
Private data field untouched by the framework.
- char **payload** [ADS\_TEL\_MAX\_PAYLOAD\_SIZE]  
payload

#### 5.6.3.2 Detailed Description

ADS Framework Buffer format.

##### Description:

The format of the buffer that is used to pass data between the ADS framework and the client application for PCM data.

### 5.6.3 AdsTelPCMPacket\_t Struct Reference

**Purpose:**

Format of the buffer used when passing data via the data callbacks.

**See also:**

**adsTelEpDataCb\_t()**.

#### 5.6.3.3 Field Documentation

unsigned short **AdsTelPCMPacket\_t::channelID**  
channel ID

unsigned short **AdsTelPCMPacket\_t::payloadType**  
bit[0-6] payload type, bit[7] SID mark bit

unsigned short **AdsTelPCMPacket\_t::payloadLength**  
payload length in bytes

unsigned int **AdsTelPCMPacket\_t::timestamp**  
frame time stamp

unsigned short **AdsTelPCMPacket\_t::payloadValid**  
data is valid

unsigned long **AdsTelPCMPacket\_t::privateData1**  
Private data field untouched by the framework.

unsigned long **AdsTelPCMPacket\_t::privateData2**  
Private data field untouched by the framework.

char **AdsTelPCMPacket\_t::payload**[ADS\_TEL\_MAX\_PAYLOAD\_SIZE]  
payload

---

### 5.6.4 AdsTelRetAck\_s Struct Reference

**adsTelFwCb\_t** return buffer structure type.

#### 5.6.4.1 Data Fields

- unsigned int **transId**  
the transaction ID that was returned from the original API call
- **AdsTelStatus\_t retStatus**  
the return status

#### 5.6.4.2 Detailed Description

**adsTelFwCb\_t** return buffer structure type.

**Description:**

This structure is the return buffer type of **adsTelFwCb\_t** when retStatus is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_SUCCESS**  
**ADS\_TEL\_STATUS\_FAIL**

#### 5.6.4 AdsTelRetAck\_s Struct Reference

**ADS\_TEL\_STATUS\_INVALID\_PARAM.**

**Purpose:**

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

**See also:**

**adsTelFwCb\_t**.

#### 5.6.4.3 Field Documentation

unsigned int **AdsTelRetAck\_s::transId**

the transaction ID that was returned from the original API call

**AdsTelStatus\_t AdsTelRetAck\_s::retStatus**

the return status

---

#### 5.6.5 AdsTelRetTg\_s Struct Reference

**adsTelFwCb\_t** return buffer structure type.

##### 5.6.5.1 Data Fields

- unsigned int **transId**  
the transaction ID that was returned from the original API call
- **AdsTelStatus\_t retStatus**  
the returned status
- unsigned int **reason**  
the reason for the TG play completion.
- unsigned int **numTones**  
the number of tones played.

##### 5.6.5.2 Detailed Description

**adsTelFwCb\_t** return buffer structure type.

**Description:**

This structure is the return buffer type of **adsTelFwCb\_t** when **retStatus** is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_TG\_CMPLT**.

**Purpose:**

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

**See also:**

**adsTelFwCb\_t**  
**adsTelEpTonePlay()**  
**adsTelEpFskPlay()**.

#### 5.6.5.3 Field Documentation

unsigned int **AdsTelRetTg\_s::transId**

the transaction ID that was returned from the original API call

### 5.6.5 AdsTelRetTg\_s Struct Reference

**AdsTelStatus\_t AdsTelRetTg\_s::retStatus**

the returned status

unsigned int **AdsTelRetTg\_s::reason**

the reason for the TG play completion.

unsigned int **AdsTelRetTg\_s::numTones**

the number of tones played.

0 if FSK data

---

### 5.6.6 AdsTelRetTdFsk\_s Struct Reference

**adsTelFwCb\_t** return buffer structure type.

#### 5.6.6.1 Data Fields

- unsigned int **transId**  
the transaction ID that was returned from the original API call
- **AdsTelStatus\_t retStatus**  
the returned status
- unsigned int **reason**  
the reason that for the TD FSK detect completion.
- unsigned int **numBytes**  
the number of FSK bytes received
- char **data** [ADS\_TEL\_MAX\_FSK\_BYTES]  
the FSK data bytes received

#### 5.6.6.2 Detailed Description

**adsTelFwCb\_t** return buffer structure type.

##### Description:

This structure is the return buffer type of **adsTelFwCb\_t** when **retStatus** is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_TD\_RCV\_FSK\_CMPLT**.

##### Purpose:

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

##### See also:

**adsTelFwCb\_t**  
**adsTelEpFskRecv()**.

#### 5.6.6.3 Field Documentation

unsigned int **AdsTelRetTdFsk\_s::transId**

the transaction ID that was returned from the original API call

**AdsTelStatus\_t AdsTelRetTdFsk\_s::retStatus**

the returned status

unsigned int **AdsTelRetTdFsk\_s::reason**

### 5.6.6 AdsTelRetTdFsk\_s Struct Reference

the reason that for the TD FSK detect completion.

unsigned int **AdsTelRetTdFsk\_s::numBytes**

the number of FSK bytes received

char **AdsTelRetTdFsk\_s::data**[ADS\_TEL\_MAX\_FSK\_BYTES]

the FSK data bytes received

---

## 5.6.7 AdsTelRetTd\_s Struct Reference

**adsTelFwCb\_t** return buffer structure type.

### 5.6.7.1 Data Fields

- unsigned int **transId**  
the transaction ID that was returned from the original API call.
- **AdsTelStatus\_t retStatus**  
the returned status.
- unsigned int **reason**  
the reason that for the TD detect completion.
- unsigned int **numTones**  
the number of tones detected.
- unsigned int **tones** [ADS\_TEL\_MAX\_TONES]  
the received tone IDs.

### 5.6.7.2 Detailed Description

**adsTelFwCb\_t** return buffer structure type.

#### Description:

This structure is the return buffer type of **adsTelFwCb\_t** when **retStatus** is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_TD\_RCV\_CMPLT**.

#### Purpose:

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

#### See also:

**adsTelFwCb\_t**  
**adsTelEpToneRecv()**.

### 5.6.7.3 Field Documentation

unsigned int **AdsTelRetTd\_s::transId**

the transaction ID that was returned from the original API call.

**AdsTelStatus\_t AdsTelRetTd\_s::retStatus**

the returned status.

unsigned int **AdsTelRetTd\_s::reason**

the reason that for the TD detect completion.

unsigned int **AdsTelRetTd\_s::numTones**

### 5.6.7 AdsTelRetTd\_s Struct Reference

the number of tones detected.

unsigned int **AdsTelRetTd\_s::tones**[ADS\_TEL\_MAX\_TONES]  
the received tone IDs.

---

### 5.6.8 AdsTelRetEv\_s Struct Reference

**adsTelFwCb\_t** return buffer structure type.

#### 5.6.8.1 Data Fields

- unsigned int **transId**  
the transaction ID - This will always be set to 0.
- **AdsTelStatus\_t retStatus**  
the returned status.
- **AdsTelEpHandle\_t epHandle**  
the endpoint on which the event occurred.
- **AdsTelPipeDir\_t pipeDir**  
the pipeline on which the event occurred.
- unsigned int **compId**  
the component location in the pipeline on which the event occurred.
- unsigned int **code**  
the event code.
- unsigned int **data1**  
the event data word 1
- unsigned int **data2**  
the event data word 1.

#### 5.6.8.2 Detailed Description

**adsTelFwCb\_t** return buffer structure type.

##### Description:

This structure is the return buffer type of **adsTelFwCb\_t** when **retStatus** is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_EVENT**.

##### Purpose:

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

##### See also:

**adsTelFwCb\_t**  
**adsTelEpToneRecv()**.

#### 5.6.8.3 Field Documentation

unsigned int **AdsTelRetEv\_s::transId**  
the transaction ID - This will always be set to 0.

**AdsTelStatus\_t AdsTelRetEv\_s::retStatus**  
the returned status.

**AdsTelEpHandle\_t AdsTelRetEv\_s::epHandle**

### 5.6.8 AdsTelRetEv\_s Struct Reference

the endpoint on which the event occurred.

**AdsTelPipeDir\_t AdsTelRetEv\_s::pipeDir**

the pipeline on which the event occurred.

unsigned int **AdsTelRetEv\_s::compld**

the component location in the pipeline on which the event occurred.

unsigned int **AdsTelRetEv\_s::code**

the event code.

See **AdsTelEvent\_t**.

unsigned int **AdsTelRetEv\_s::data1**

the event data word 1

unsigned int **AdsTelRetEv\_s::data2**

the event data word 1.

---

### 5.6.9 AdsTelRetGetParams\_s Struct Reference

**adsTelFwCb\_t** return buffer structure type.

#### 5.6.9.1 Data Fields

- unsigned int **transId**  
the transaction ID that was returned from the original API call.
- **AdsTelStatus\_t retStatus**  
the returned status.
- unsigned int **numParams**  
the number of parameters being returned.
- int **paramVals** [ADS\_TEL\_MAX\_PARAMS]  
the parameter values.

#### 5.6.9.2 Detailed Description

**adsTelFwCb\_t** return buffer structure type.

##### Description:

This structure is the return buffer type of **adsTelFwCb\_t** when **retStatus** is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_GET\_PARAMS**.

##### Purpose:

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

##### See also:

**adsTelFwCb\_t**  
**adsTelEpCompParamsGet()**.

#### 5.6.9.3 Field Documentation

unsigned int **AdsTelRetGetParams\_s::transId**



### 5.6.9 AdsTelRetGetParams\_s Struct Reference

the transaction ID that was returned from the original API call.

**AdsTelStatus\_t AdsTelRetGetParams\_s::retStatus**

the returned status.

unsigned int **AdsTelRetGetParams\_s::numParams**

the number of parameters being returned.

int **AdsTelRetGetParams\_s::paramVals[ADS\_TEL\_MAX\_PARAMS]**

the paramter values.

The index is the param ID

---

### 5.6.10 AdsTelRetPlay\_s Struct Reference

**adsTelFwCb\_t** return buffer structure type.

#### 5.6.10.1 Data Fields

- unsigned int **transId**  
the transaction ID that was returned from the original API call.
- **AdsTelStatus\_t retStatus**  
the returned status
- unsigned int **reason**  
the reson for the player completion.

#### 5.6.10.2 Detailed Description

**adsTelFwCb\_t** return buffer structure type.

##### Description:

This structure is the return buffer type of **adsTelFwCb\_t** when **retStatus** is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_PLY\_CMPLT**.

##### Purpose:

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

##### See also:

**adsTelFwCb\_t**  
**adsTelEpCompParamsGet()**.

#### 5.6.10.3 Field Documentation

unsigned int **AdsTelRetPlay\_s::transId**

the transaction ID that was returned from the original API call.

**AdsTelStatus\_t AdsTelRetPlay\_s::retStatus**

the returned status

unsigned int **AdsTelRetPlay\_s::reason**

the reson for the player completion.

---

## 5.7 Define Documentation

```
#define ADS_TEL_MAX_PAYLOAD_SIZE
```

Maximum payload length.

**Description:**

Used to define the maximum length of payload for **AdsTelPacket\_t** that can be passed to the framework via **adsTelEpDataCb\_t**.

```
#define ADS_TEL_RTP_MD_TYPE_AUDIO
```

Audio packet packet payload type.

**Description:**

Used to specify that the payload of a framework packet is audio.

```
#define ADS_TEL_RTP_MD_TYPE_RFC2833_EVENT
```

RFC2833 Event packet payload type.

**Description:**

Used to specify that the payload of a framework packet is a RFC2833 Event.

---

## 5.8 Typedef Documentation

```
typedef AdsTelStatus_t(* adsTelEpDataCb_t)(AdsTelEpHandle_t endPtHandle, void *pBuffer1, void *pBuffer2)
```

adsTel Framework Data callback.

**Description:**

Function prototype called by ADS when it is ready to receive/transmit data for a pipeline

**Context:****Assumptions:**

None

**Side-Effects:**

None

**Returns:**

void

**Precondition:**

None

**Postcondition:**

None

**Note:**

## 5.8 Typedef Documentation

None

### See also:

```
typedef void(* adsTelFwCb_t)(unsigned int transId, void *pData)
adsTel Framework API and Event callback.
```

### Description:

Prototype of the client's function to accept notification of API completion and event messages. This function is registered through the adsTel initialization function **adsTelInit()**. The format of the data buffer *pData* is dependent on the the value of status.

### Context:

Framework thread context.

### Assumptions:

None

### Side-Effects:

None

### Parameters:

|                |                                                                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>transId</i> | [OUT] - value that was returned to the client application from the original API call. Note this value will be 0 when an event is being returned. |
| <i>pData</i>   | [OUT] - The returned data buffer. The format of the buffer is dependent of the returned status.                                                  |

### Returns:

void.

### Precondition:

None.

### Postcondition:

None.

### Note:

None.

### See also:

**adsTelInit()**  
**AdsTelRetAck\_t**  
**AdsTelRetTg\_t**  
**AdsTelRetTd\_t**  
**AdsTelRetTdFsk\_t**  
**AdsTelRetEv\_t**  
**AdsTelRetGetParams\_t**  
**AdsTelRetPlay\_t.**

```
typedef struct AdsTelRetAck_s AdsTelRetAck_t
adsTelFwCb_t return buffer structure type.
```

### Description:

This structure is the return buffer type of **adsTelFwCb\_t** when *retStatus* is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_SUCCESS**

## 5.8 Typedef Documentation

**ADS\_TEL\_STATUS\_FAIL**  
**ADS\_TEL\_STATUS\_INVALID\_PARAM.**

**Purpose:**

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

**See also:**

**adsTelFwCb\_t**.

```
typedef struct AdsTelRetTg_s AdsTelRetTg_t
adsTelFwCb_t return buffer structure type.
```

**Description:**

This structure is the return buffer type of **adsTelFwCb\_t** when retStatus is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_TG\_CMPLT**.

**Purpose:**

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

**See also:**

**adsTelFwCb\_t**  
**adsTelEpTonePlay()**  
**adsTelEpFskPlay()**.

```
typedef struct AdsTelRetTdFsk_s AdsTelRetTdFsk_t
adsTelFwCb_t return buffer structure type.
```

**Description:**

This structure is the return buffer type of **adsTelFwCb\_t** when retStatus is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_TD\_RCV\_FSK\_CMPLT**.

**Purpose:**

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

**See also:**

**adsTelFwCb\_t**  
**adsTelEpFskRecv()**.

```
typedef struct AdsTelRetTd_s AdsTelRetTd_t
adsTelFwCb_t return buffer structure type.
```

**Description:**

This structure is the return buffer type of **adsTelFwCb\_t** when retStatus is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_TD\_RCV\_CMPLT**.

**Purpose:**

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

**See also:**

**adsTelFwCb\_t**  
**adsTelEpToneRecv()**.

```
typedef struct AdsTelRetEv_s AdsTelRetEv_t
adsTelFwCb_t return buffer structure type.
```

## 5.9 Enumeration Type Documentation

### Description:

This structure is the return buffer type of **adsTelFwCb\_t** when **retStatus** is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_EVENT**.

### Purpose:

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

### See also:

**adsTelFwCb\_t**  
**adsTelEpToneRecv()**.

```
typedef struct AdsTelRetGetParams_s AdsTelRetGetParams_t
adsTelFwCb_t return buffer structure type.
```

### Description:

This structure is the return buffer type of **adsTelFwCb\_t** when **retStatus** is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_GET\_PARAMS**.

### Purpose:

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

### See also:

**adsTelFwCb\_t**  
**adsTelEpCompParamsGet()**.

```
typedef struct AdsTelRetPlay_s AdsTelRetPlay_t
adsTelFwCb_t return buffer structure type.
```

### Description:

This structure is the return buffer type of **adsTelFwCb\_t** when **retStatus** is any of the following **AdsTelStatus\_t** members. **ADS\_TEL\_STATUS\_PLY\_CMPLT**.

### Purpose:

Return buffer structure type for API and event callback **adsTelFwCb\_t**.

### See also:

**adsTelFwCb\_t**  
**adsTelEpCompParamsGet()**.

---

## 5.9 Enumeration Type Documentation

```
enum AdsTelEvent_t
AdsTel event types.
```

### Description:

event types that can be returned by the framework via **adsTelFwCb\_t** return type **ADS\_TEL\_STATUS\_EVENT**.

### Enumerator:

|                                   |                                 |
|-----------------------------------|---------------------------------|
| <b>ADS_TEL_EV_TD_TONE_ON</b>      | tone on event.                  |
| <b>ADS_TEL_EV_TD_TONE_OFF</b>     | tone off event.                 |
| <b>ADS_TEL_EV_LOST_PACKET</b>     | lost packet event.              |
| <b>ADS_TEL_EV_DEC_PACKET_CHNG</b> | payload type of packek changed. |