Mobile Data Modem Audio Calibration Database

User Guide

80-VM407-6 A

January 25, 2013

Submit technical questions at:
https://support.cdmatech.com/

Confidential and Proprietary – Qualcomm Technologies, Inc.

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Contents

1 Introduction 5

1.1 Purpose 5
1.2 Scope 5
1.3 Conventions 5
1.4 References 5
1.5 Technical assistance 6
1.6 Acronyms 6

2 Offline Calibration 7

2.1 Calibration procedure 7
2.2 How this works 11

3 Online Calibration 13

3.1 Calibration procedure 13
3.2 How this works 14

4 Real-Time Calibration 15

4.1 Calibration Procedure 15
4.2 How this works 16

Tables

Table 1-1 Reference documents and standards 5

Revision history

Revision	Date	Description
A	Jan 2013	Initial release

1 Introduction

1.1 Purpose

This document provides an overview of various aspects related to Audio Calibration Database (ACDB) for MDM9625 targets, highlighting various calibration procedures and how they function on the device. The three types of calibration procedures discussed are:

- Offline calibration
- Online calibration
- Real-time calibration

1.2 Scope

Readers are expected to use [Q2] and [Q3] in association with this document. This document does not provide specific details related to ACDB APIs, ACPH commands, or how to use QACT.

1.3 Conventions

Function declarations, function names, type declarations, and code samples appear in a different font, e.g., `#include`.

Code variables appear in angle brackets, e.g., `<number>`.

Commands to be entered appear in a different font, e.g., **`copy a:*.* b:`**.

Button and key names appear in bold font, e.g., click **Save** or press **Enter**.

1.4 References

Reference documents are listed in Table 1-1. Reference documents that are no longer applicable are deleted from this table; therefore, reference numbers may not be sequential.

Table 1-1 Reference documents and standards

Ref.	Document	
Qualcomm Technologies		
Q1	*Application Note: Software Glossary for Customers*	CL93-V3077-1
Q2	*QACT v3.x.x User Guide*	80-VM407-5
Q3	*Audio Calibration Database (ACDB) API Interface Specification (B Family)*	80-NB127-3
Q4	*Audio Calibration Packet Handler (ACPH) API Interface Specification*	80-NF006-1

1.5 Technical assistance

For assistance or clarification on information in this document, submit a case to Qualcomm Technologies, Inc. (QTI) at https://support.cdmatech.com/.

If you do not have access to the CDMATech Support Service website, register for access or send email to support.cdmatech@qti.qualcomm.com.

1.6 Acronyms

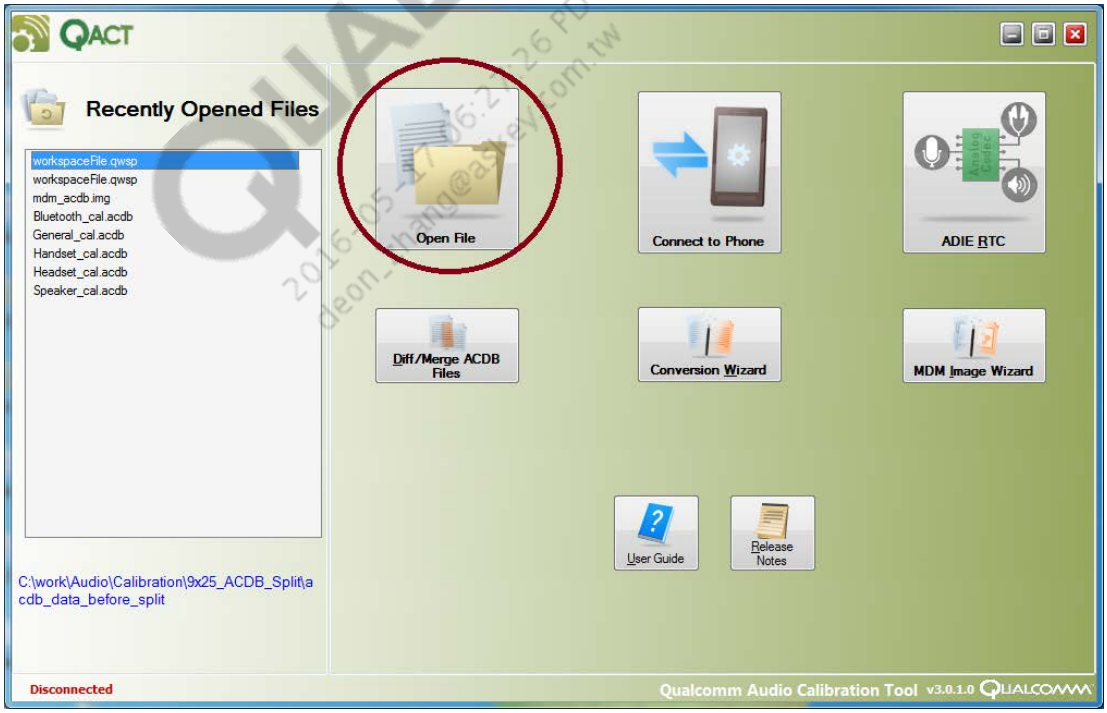
For definitions of terms and abbreviations, see [Q1].

2 Offline Calibration

Offline calibration is the calibration of ACDB data in a file that is done completely on the PC without any connection required to the device. In this calibration, the user opens the ACDB using QACT, modifies the data, and saves it back to the database file on a PC. An ACDB is typically stored on PC as a workspace file that contains details about a collection of .acdb files that constitute an audio calibration database. For MDM9625, .acdb files in an ACDB workspace are later combined to make an ACDB image which is flashed onto the device.

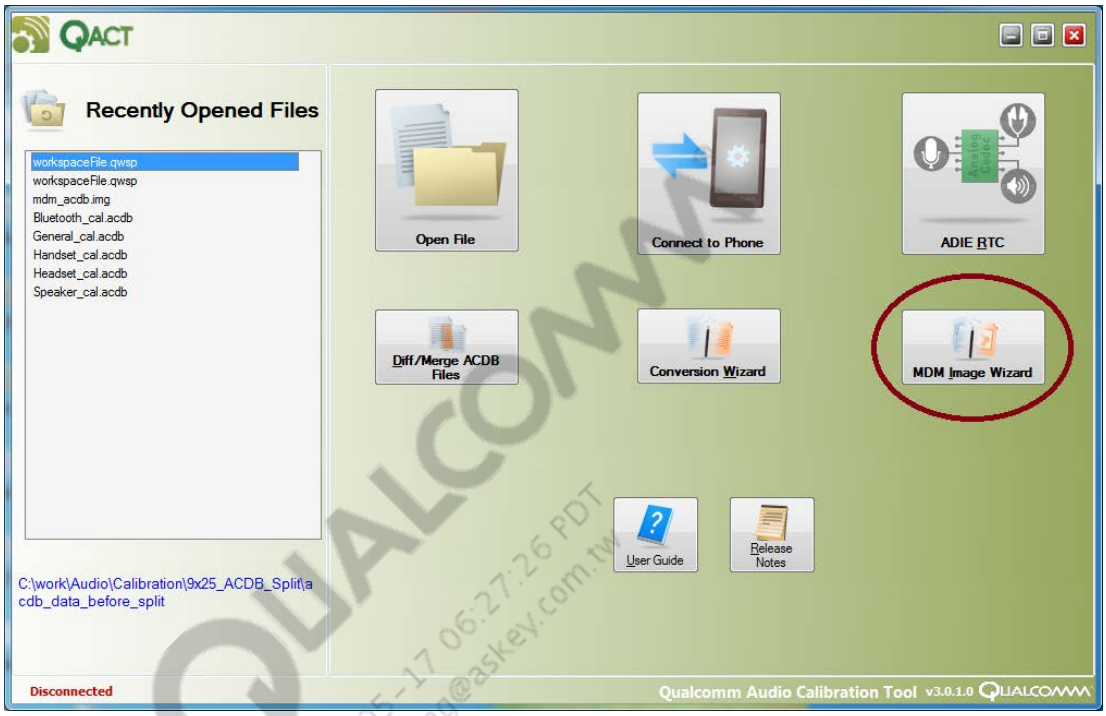
2.1 Calibration procedure

1. Open the QACT program and click **Open File**.



2. Once the required parameters have been modified, it is saved back as a .qwsp file. When an ACDB workspace is saved, QACT will save each individual .acdb file that is part of the workspace and the workspace itself.

3. These .acdb files are now ready to be loaded onto a device that supports EFS. In the case of MDM9625, not all variations support EFS, so we chose to use the image approach in loading the database onto the device. The next step is to combine all .acdb files to make a single ACDB image. That can be achieved using the MDM Image Wizard available with QACT.



4. The wizard will generate an .img file, e.g., mdm_acdb.img, from the .qwsp workspace file. However, the .img itself is not ready to be loaded onto the device. This must be converted to a loadable image, i.e., an .mbn file. This consists of the .img file generated above, prepended with a small header that contains details about the size of the data and location where the data must be loaded into memory.

5. An .mbn file can be generated as follows:

```
copy /B acdb_header.mbn+mdm_acdb.img acdb.mbn
```

The acdb_header.mbn can be obtained from the corresponding build's build/ms/bin/<flavor>. The folder also contains an acdb.mbn file that is the equivalent .mbn file for the mdm_acdb.img present in the build's apps_proc/multimedia/audio/audcal/acdbdata/image folder. So, alternatively, .mbn can also be generated from .img by placing the file as mdm_acdb.img under the abovementioned folder and issuing a build command. The build command will generate acdb.mbn.

6. The .mbn file is now ready to be loaded onto the device and can be loaded using QPST or JTAG.

Using QPST

a. Before flashing, please ensure the Sahara Configuration is set as follows:

i In the QPST Configuration window, **Shift + right-click** on the port on which the phone is present.

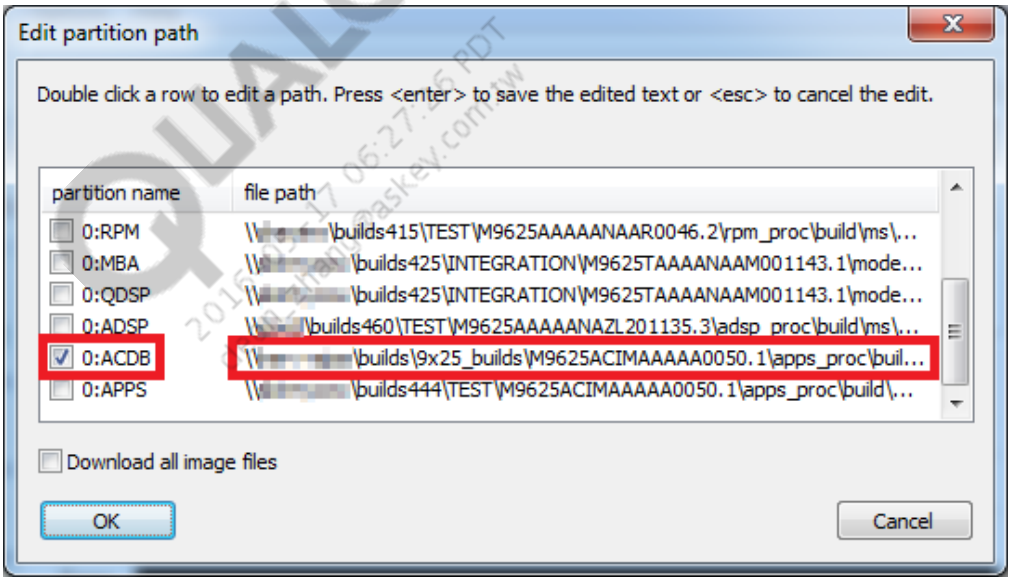
ii Click **Sahara Configuration**.

iii In the Sahara Configuration Dialog window, "Upon receiving Sahara Hello" (at the bottom of the window), set the third option "If a Ram DUMP request, perform download instead".

b. Open QPST Software Download Tool.

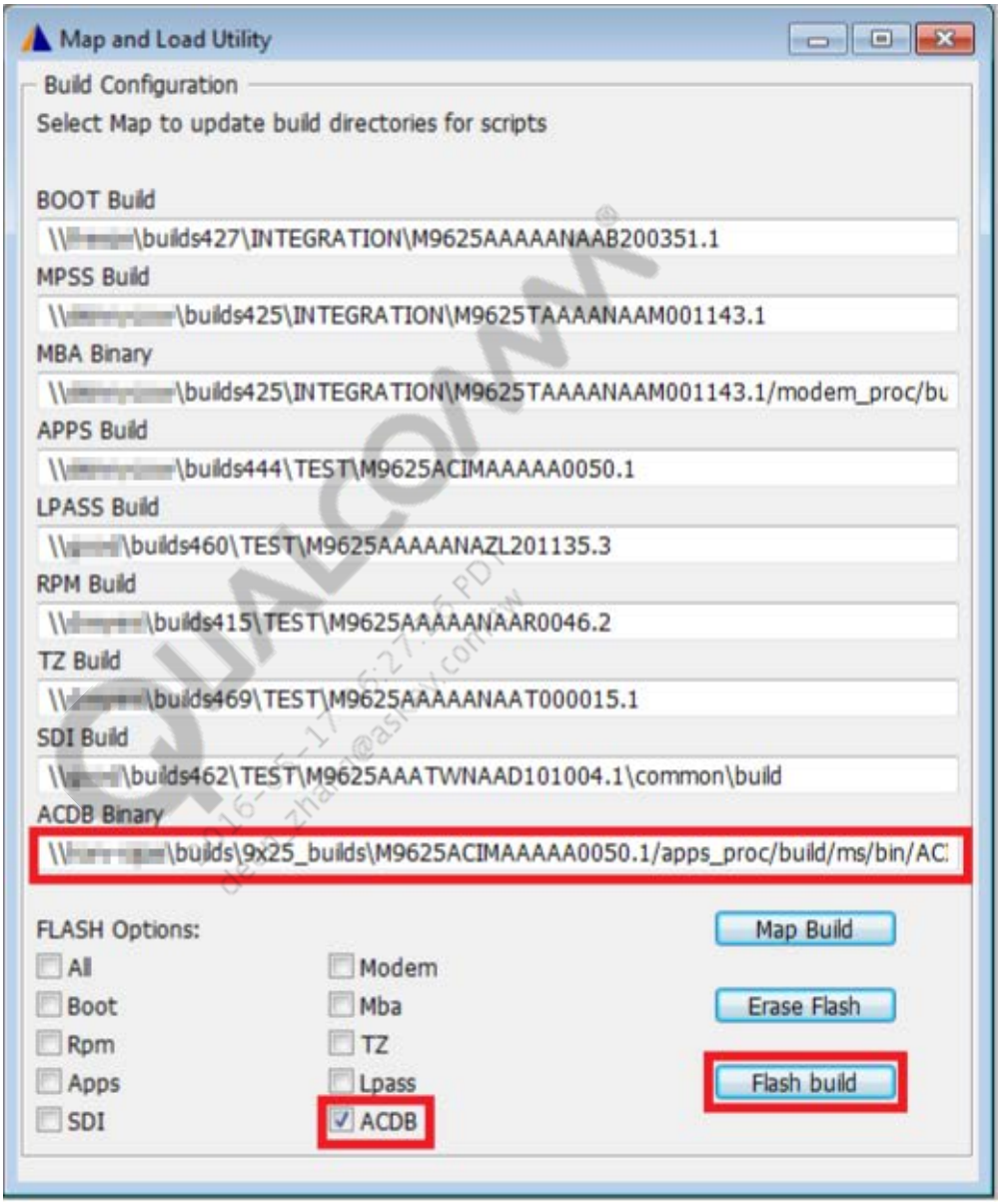
c. On the **SB 3.0** tab, select content.xml from the Metabuild root directory by clicking **Browse**.

d. Click **Edit** to bring up the UI and select ACDB and its path as shown in the figure below:



Using JTAG

a. Flash ACDB Image using Map and Load Utility in Trace32 as shown in the figure below:



2.2 How this works

During boot time, the acdb.mbn flashed onto the device is loaded to a predetermined memory address that is present in its header before ACDB is initialized. On MDM9625, ACDB is initialized by CSD by calling csd_acdb_init(). This function will check for a proper ACDB Image loaded at this address. Once the check passes, it will parse the image header and identifies the list of files that are present and issues an ACDB_CMD_INITIALIZE_V2 command to ACDB for the list of files present in the image.

ACDB image header

```
struct img_file_format {
   uint32_t acdb_tag;
   uint16_t major_version;
   uint16_t minor_version;
   uint32_t num_files;
   struct file_entry file_list[0];
};
```

Member	Description
acdb_tag	A signature tag that is used to verify if a proper ACDB Image is loaded; value is *0xacdbacdb*
major_version	Major version of the ACDB image header
minor_version	Minor version of the ACDB image header
num_files	Total number of .acdb files present in the image
file_list	Array of structures with details to each file present in the device

File list structure

```
struct file_entry {
   uint32_t filenamelen;
   char filename[FILENAME_MAX_LEN];
   uint32_t offset;
   uint32_t size;
   uint32_t group_id;
   uint32_t flavor_id;
};
```

Member	Description
filenamelen	Length of the string representing filename
filename	A unicode string of maximum size FILENAME_MAX_LEN representing the filename; FILENAME_MAX_LEN is defined as 128
offset	Offset from the beginning of the Image where this file exists
size	Size of the data available for this file

Member	Description
group_id	Group ID to which this file belongs that helps identify what files need to be loaded for a particular group ID; but this is not used in MDM9625; by default, all files belong to one single group
flavor_id	Flavor ID helps identify which files constitute an ACDB database; this is introduced so that multiple ACDBs can be included in an ACDB image

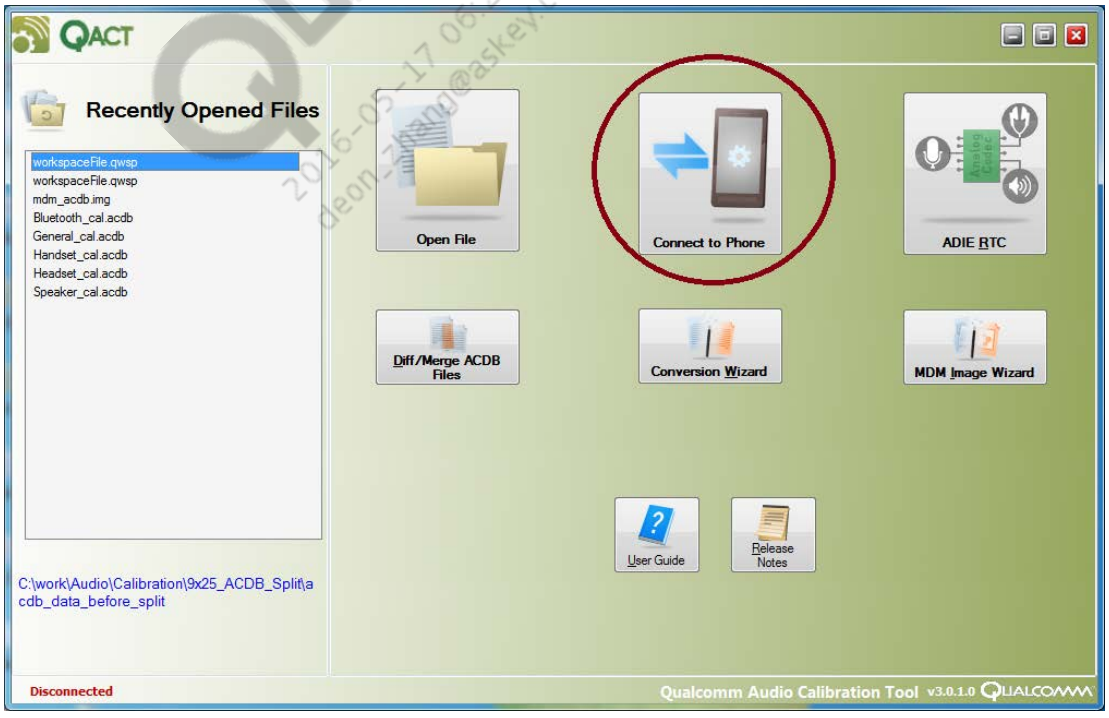
ACDB in turn requests these files to be opened by calling AcdbGetFileData() for each file. This function, implemented specifically for MDM9625, looks up the ACDB Image header, computes the memory address where the file is loaded in memory by adding offset of the filename to the ACDB Image base address, and returns the same.

3 Online Calibration

Online calibration is the calibration of ACDB data currently present and being used by a device. In this calibration, the user opens ACDB data on the device from a PC via QACT and modifies and saves it back to the device while the device is connected. The changes applied will be effective from the next voice call after calibration change; e.g., if a voice call is ongoing and if online calibration is done, the change will not be effective on the ongoing call; unlike RTC, the changes will be effective only from the next call. If the device is already in a voice call, the changes are effective from the next voice call. If the user wants the changes to persist, the data must be stored as a database on the PC and can later be flashed to the device as described in Chapter 2.

3.1 Calibration procedure

1. Connect to ACDB on the phone using QACT.



2. Various parameters can be queried and updated instantly using QACT.
3. As the changes made above only last until power-cycle, users may want to save it as a workspace and follow the procedure specified in Chapter 2 to persist these changes to the device.

3.2 How this works

QACT communicates with ACDB on a connected device via the Audio Calibration Package Handler (ACPH) on that device by issuing the following commands to query the required data:

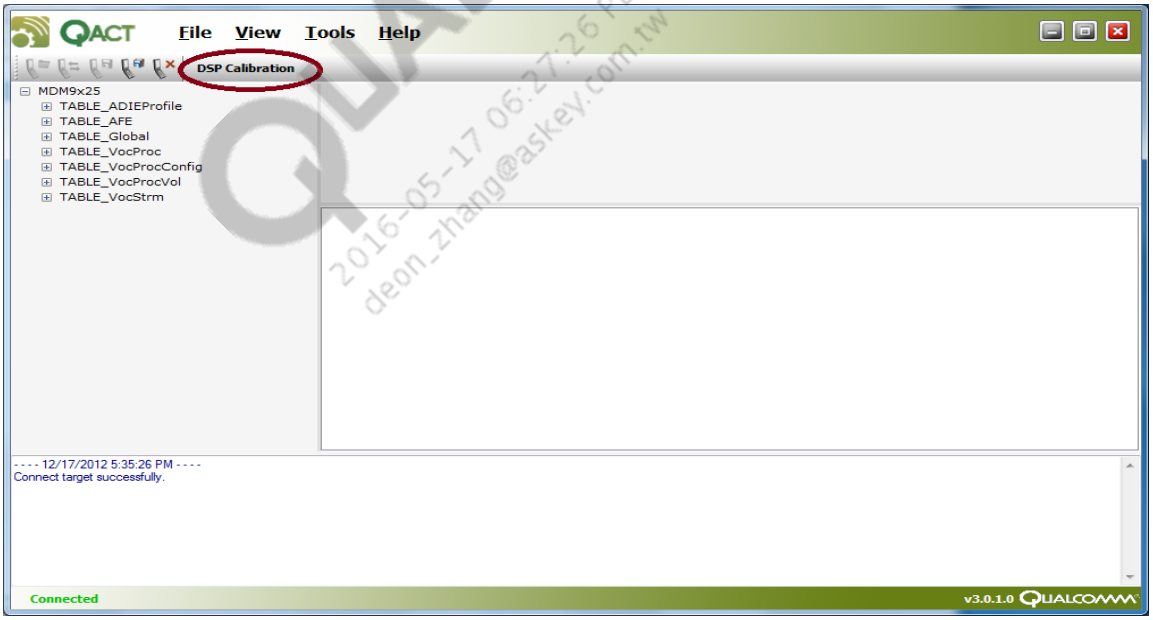
1. ACPH_CMD_GET_TARGET_VERSION (0x0001)
2. ACPH_CMD_CHECK_CONNECTION (0x0002)
3. ACPH_CMD_GET_ACDB_FILES_INFO (0x0003)
4. ACPH_CMD_QUERY_MAX_BUFFER_LENGTH (0x0004)
5. ACPH_CMD_GET_ACDB_FILE (0x0005)
6. ACPH_CMD_GET_NO_OF_TBL_ENTRIES_ON_HEAP (0x0006)
7. ACPH_CMD_GET_TBL_ENTRIES_ON_HEAP (0x0007)
8. ACPH_CMD_GET_ACDB_DATA (0x0008)
9. ACPH_CMD_SET_ACDB_DATA (0x0009)
10. ACPH_CMD_QUERY_ONLINE_VERSION (0x000A)

4 Real-Time Calibration

Real-time calibration is calibration of data that is currently set and being used by DSP on the connected device via QACT. This procedure is performed while the DSP on the connected device is active, such as during an audio playback or a voice call. The changes are effective immediately until the DSP session ends. Using this procedure, users can modify certain parameters and immediately observe the effect and hence calibrate their device.

4.1 Calibration Procedure

1. Connect to ACDB on the phone using QACT as shown in Section 3.1.
2. Open real-time calibration data on phone as shown



3. Various parameters can be queried and updated instantly using QACT.
4. Once the changes are final, the user can copy the real-time calibration data to ACDB then save it as a workspace and follow the procedure specified in Chapter 2 to persist the changes to device.

4.2 How this works

QACT communicates with ACDB data on the DSP of the connected device via ACPH on that device by issuing a different set of commands:

1. ACPH_CMD_QUERY_AUD_DEVICE_COPP_HANDLES (0x00FB)
2. ACPH_CMD_QUERY_AUD_COPP_STREAM_HANDLES (0x00FC)
3. ACPH_CMD_QUERY_VOC_ALL_ACTIVE_STREAMS (0x00FD)
4. ACPH_CMD_QUERY_VOC_VS_COPP_HANDLES (0x00FE)
5. ACPH_CMD_QUERY_VOC_VC_DEVICES (0x00FF)
6. ACPH_CMD_RTC_GET_CAL_DATA (0x0100)
7. ACPH_CMD_RTC_SET_CAL_DATA (0x0101)
8. ACPH_CMD_QUERY_AUD_TOPOLOGY_COPP_HANDLES (0x0102)
9. ACPH_CMD_QUERY_VOC_VC_TOPOLOGIES (0x0103)
10. ACPH_CMD_QUERY_DSP_RTC_VERSION (0x0104)