Qualcomm Technologies International, Ltd.

ADK Audio Prompts

Application Note

80-CT418-1 Rev. AF

October 23, 2017

Revision history

Revision	Date	Description
1	FEB 2014	Original publication of this document. Alternative document number CS-00237358-AN.
2	FEB 2014	Updated for ADK 3.0 final release and to the new CSR™ style.
3	JUN 2015	Updated for ADK 4.0
4	SEP 2016	Updated to conform to QTI standards; no technical content was changed in this document revision.
AE	JUL 2017	Updated for ADK 6.0.1. DRN updated to use Agile number.
AF	OCT 2017	Reverted to content in Issue 4 and updated for ADK 4.3 release. Added to the Content Management System.

Contents

Tables

Figures

1 Introduction to Audio prompts

Audio prompts refer to audio data stored in internal flash or external SQIF. Audio prompts can be associated with Sink application events to enable more meaningful user feedback.

Details how audio prompts operate in Firmware and how they are generated and stored in the file system are described in:

- Audio prompts VM traps
- Audio prompt tool
- Configuring audio prompts
- Using external flash to store audio prompt files

2 Audio prompts VM traps

The VM `StreamFileSource` function enables data access from the file system using the streams interface, this can be either audio prompts stored in the internal `filesystem` or on an external SQIF with `filesystem` partitions.

The VM `TransformAdpcmDecode` function provides an extra transform to decompress IMA ADPCM encoded data from any source, including `StreamI2cSource`.

Other encoded audio prompt formats are supported using the DSP, these include SBC, MP3, and AAC encoded audio prompts, see Configuring audio prompts for further details.

3 Audio prompt tool

The ADK supports audio prompts in several compression formats. The Audio Prompt Tool is a command-line executable that generates header information and audio prompt files for the file system from an input configuration file.

Header information is required to provide a single point of indexing for audio prompts and to store information that is needed when playing back the audio prompts such as compression format and sample rate. The script generates separate output files for actual audio prompts data in the prompts folder (`.prm` files). The header information files (`.idx`) are generated in the headers folder.

The file system prompts can be downloaded by rebuilding the application image with the prompts and headers folders in the **image** folder and downloading the resultant `.xpv/xdv` files to the IC using **BlueFlash**.

NOTE Additional prompts may be stored in an external SQIF device file system partition (see) provided they are included in the prompts and header folder.

Table 3-1 shows how the header information for each prompt is stored.

Table 3-1 Storage of audio prompt header information

Byte	Data	Values
[0]	Storage Type	`0x00` - <Reserved> `0x01` – <Reserved> `0x02` – File System

[1]	`Audio prompt Index`	`0x0000 to 0xffff`
		`(Big Endian)`

[2]		
[3]	<Reserved>	-
[4]	Stereo flag (for uncompressed PCM audio prompts only)	`0x00` - <Mono> `0x01` - <Stereo>

[5]	Audio prompt size in Bytes	`0x00000000 to 0xffffffff`
		`(Big Endian)`

[6]		
[7]		
[8]		

Table 3-1 Storage of audio prompt header information (cont.)

Byte	Data	Values
[9]	Compression format	`0x00` – reserved `0x01` – IMA ADPCM `0x02` – SBC `0x03` – MP3 `0x04` – AAC `0x05` – PCM **NOTE** `0x00` value assumes Mono PCM at 8khz
[10]	Sample Rate	`0x0000` to `0xffff`
[11]		`(Big Endian)`

When running the Audio Prompt Tool python script, passing the input configuration file is mandatory, other options are available also, thatb is:

- `-i`: Sets the location of the image directory, this is where file system prompts are generated
- `-h and --help`: Displays the **help** window

NOTE The Audio Prompt Tool is invoked by ADK Configuration Tools. Therefore, if you use the Sink Configuration Tool to generate audio prompts you do not need to run this script directly.

3.1 Audio prompt configuration file

The input configuration file must include one line of comma-separated-values for each audio prompt to be configured in the format:

compression=`compression_format`,rate=`sample_rate`,channels=`channels`,file=`file_name`

Table 3-2 describes these parameters.

Table 3-2 Audio prompt configuration parameters

Parameter	Meaning	Values	Default
`compression_format`	Compression format of audio prompt file	adpcm sbc mp3 aac pcm	pcm
`sample_rate`	The sample rate of audio prompt file	8000 to 48000, see Table 3-3 for more information	8000

Table 3-2 Audio prompt configuration parameters (cont.)

Parameter	Meaning	Values	Default
`channels`	Number of channels (applicable to pcm compression format only)	mono stereo	mono
`file_name`	The name of the file that is to be used to generate the audio prompt data	Full path of the file containing the audio prompt data.	N/A

The file extension and content of each input file must match the specified configuration for each particular audio prompt. This is checked by the Audio Prompt Tool.

Further requirements may need to be met based on the compression format used for a particular audio prompt, see Table 3-3.

Table 3-3 Audio prompt input file requirements

Compression Format	Input File Extension	Content Format	Extra Requirements	Pre encoded by User
ADPCM	`.wav`	8 khz mono PCM wave file format	rate = 8000	No

SBC	`.sbc`	SBC encoded	rate = encoded sample rate
			rate = 16, 32, 44.1 or 48 khz
				Yes

MP3	`.mp3`	MPEG 1 or 2 layer III encoded format	rate = encoded sample rate
			rate = 16, 32, 44.1 or 48 Khz
				Yes

AAC	`.adts`	AAC encoded in ADTS bit stream format	rate = encoded sample rate
			rate = 8, 16, 32, 44.1, or 48 kHz
			stereo only
				Yes

PCM	`.wav`	Mono or Stereo 16-bit PCM wave file format	rate = wave file sample rate, rate must be between 8 khz and 48 khz stereo flag must match wave file number of channels	No

Table 3-3 shows that SBC, MP3, and AAC audio prompts must be pre-encoded by the user, while PCM and ADPCM audio prompts can be generated by the Audio Prompt Tool from the standard `.wav` file format.

The PCM and ADPCM audio prompts are mixable, that is,. they can be mixed with other audio streams including A2DP music or SCO voice channels if present. However, SBC, MP3 and AAC audio prompts are non-mixable, that is they suspend other audio streams during audio prompt playback.

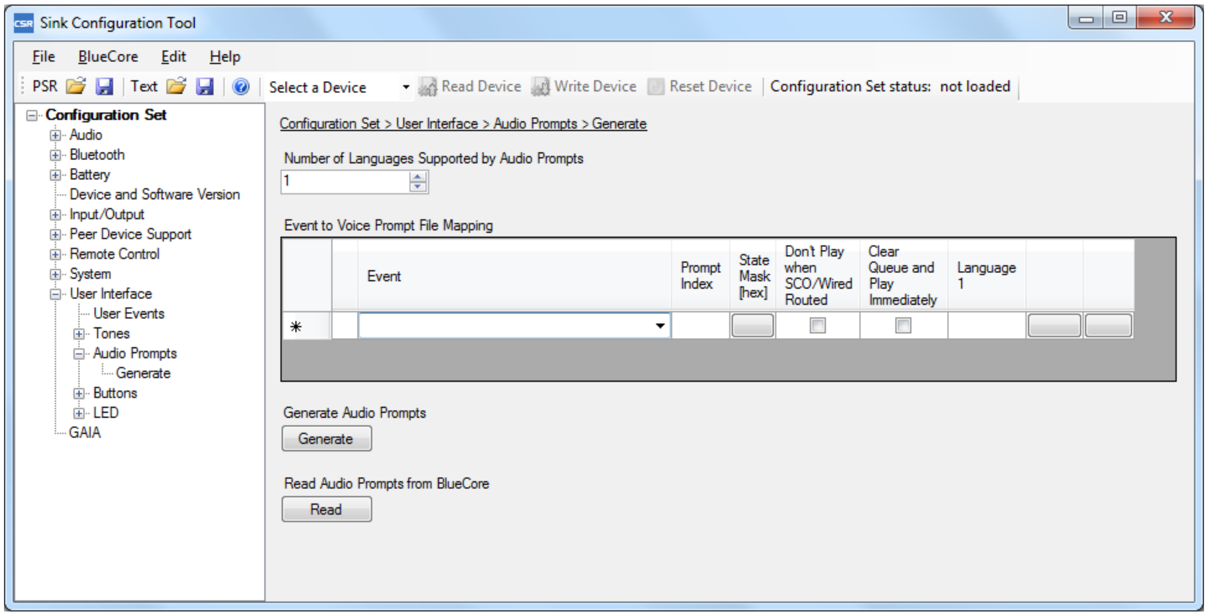
NOTE The ADK does not include tools to create MP3, AAC, or SBC encoded files.

4 Configuring audio prompts

Most users do not need to directly use the Audio Prompt Tool as described in Audio prompt tool, instead the ADK Sink Configuration Tool can used to add, and configure audio prompts for a Sink application.

Audio Prompts can be configured from the **User Interface/Audio Prompts/** > **Generate** node of the Tree View control, see Figure 4-1.

Figure 4-1 Audio prompt generation window



4.1 To configure an audio prompt for the power on event

This section uses an example to explains how to use the ADK Configuration Tool to configure a simple Audio Prompt. The example configures the Sink application to play the prompt “Power On” when the Power On event occurs.

Prerequisites

This example describes how to configure the application once the ADK configuration Tool is loaded and initialized with the `soundbar_with_subwoofer_CNS10001v4` PSR file.

Procedure

1. Navigate using the Tree View control to **User Interface/Audio Prompts/Generate**.
2. Select **Power On** from the **Event** drop down, see Figure 4-2.

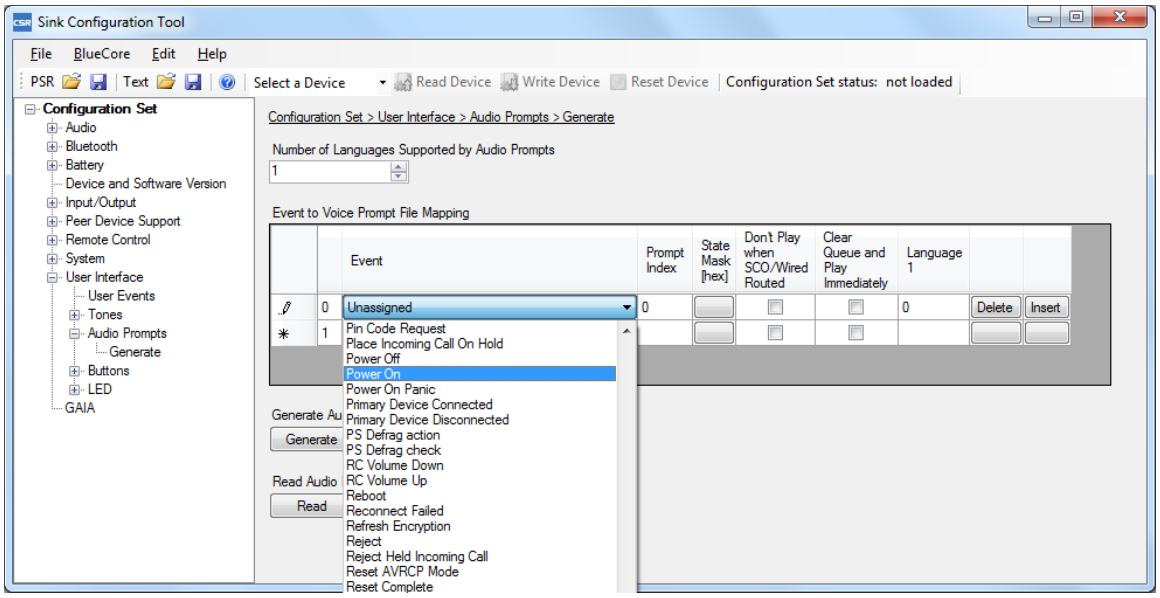


Figure 4-2 Configuring the audio prompt event mapping

This results in a new line being created in the **Event to Voice Prompt File** mapping table, see Figure 4-3.

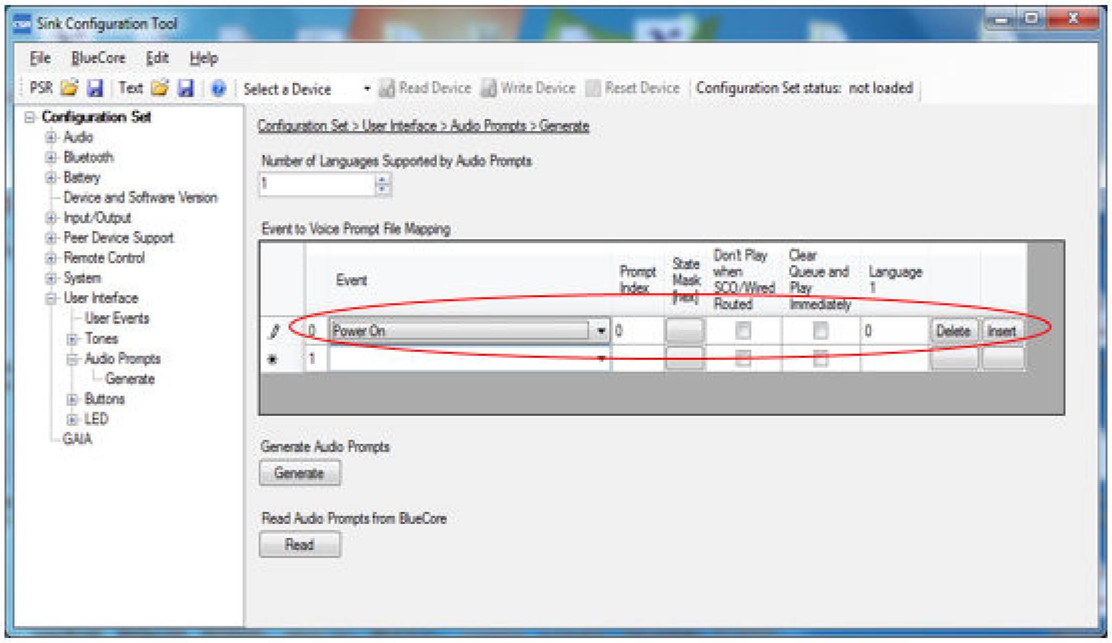
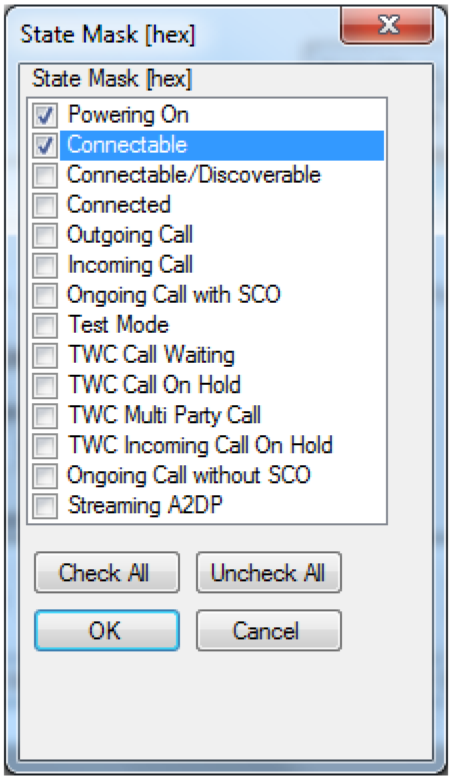

Sink Configuration Tool
File BlueCore Edit Help
PSR Text Select a Device Read Device Write Device Reset Device Configuration Set status: not loaded
Configuration Set
Audio
Bluetooth
Battery
Device and Software Version
Input/Output
Peer Device Support
Remote Control
System
User Interface
User Events
Tones
Audio Prompts
Generate
Buttons
LED
GAIA
Configuration Set > User Interface > Audio Prompts > Generate
Number of Languages Supported by Audio Prompts
1
Event to Voice Prompt File Mapping
Event
Prompt Index
State Mask [hex]
Don't Play when SCO/Wired Routed
Clear Queue and Play Immediately
Language 1
Power On
Delete
Insert
Generate Audio Prompts
Generate
Read Audio Prompts from BlueCore
Read

Figure 4-3 Configured event field of audio prompt mapping

3. Click on the **State Mask** cell for the row being configured. The **State Mask** window appears, see Figure 4-4.

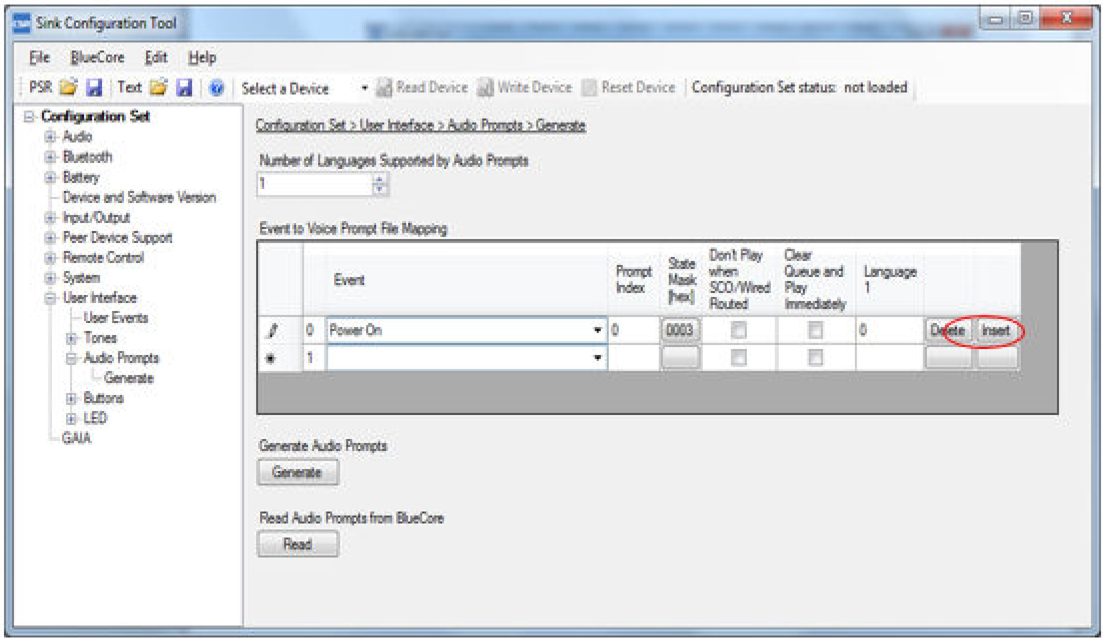


State Mask [hex]

- [✓] Powering On
- [✓] Connectable
- [ ] Connectable/Discoverable
- [ ] Connected
- [ ] Outgoing Call
- [ ] Incoming Call
- [ ] Ongoing Call with SCO
- [ ] Test Mode
- [ ] TWC Call Waiting
- [ ] TWC Call On Hold
- [ ] TWC Multi Party Call
- [ ] TWC Incoming Call On Hold
- [ ] Ongoing Call without SCO
- [ ] Streaming A2DP

Figure 4-4 : Select the audio prompt state mask

4. Check the **Powering On** and **Connectable** boxes.
   This means that the audio prompt is played when a Power On event occurs and the Sink application is in either the Powering On or Connectable state and results in the **Event to Voice Prompt File** mapping table being updated, see Figure 4-5.



Configuration Set > User Interface > Audio Prompts > Generate

Number of Languages Supported by Audio Prompts: 1

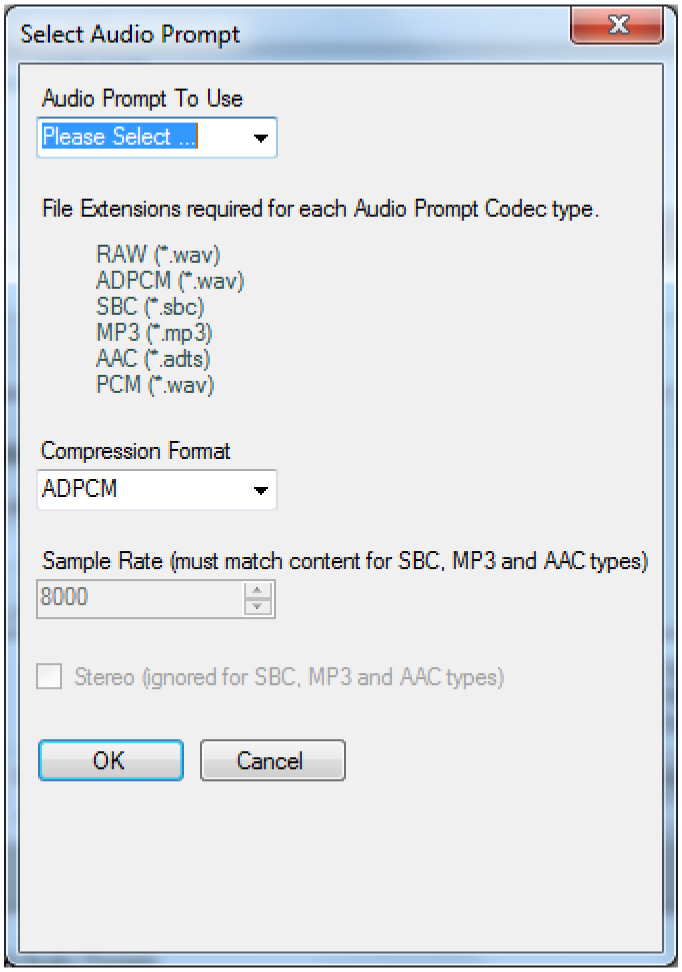
Event to Voice Prompt File Mapping

		Event	Prompt Index	State Mask [hex]	Don't Play when SCO/Wired Routed	Clear Queue and Play Immediately	Language 1		
	0	Power On	0	0003	☐	☐	0	Delete	Insert
	1				☐	☐			

Configuration Set status: not loaded

Figure 4-5 Configured audio prompt state mask

5. Click on the **Language 1** cell that is on the same row as the Power On Audio Prompt being configured, see Figure 4-5. A **Select Audio Prompt** window appears, see Figure 4-6.



Select Audio Prompt

Audio Prompt To Use: Please Select ...

File Extensions required for each Audio Prompt Codec type.

RAW (*.wav)
ADPCM (*.wav)
SBC (*.sbc)
MP3 (*.mp3)
AAC (*.adts)
PCM (*.wav)

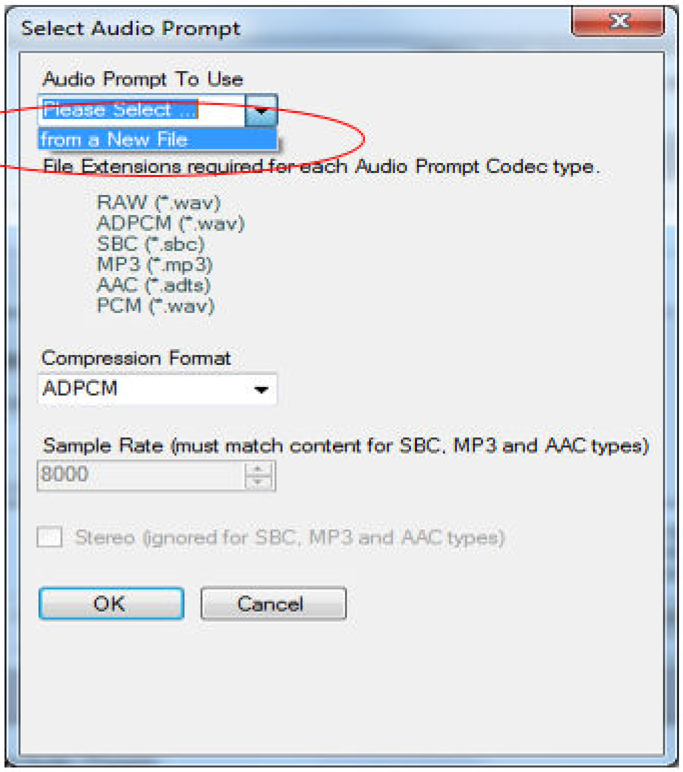
Compression Format: ADPCM

Sample Rate (must match content for SBC, MP3 and AAC types): 8000

Stereo (ignored for SBC, MP3 and AAC types)

Figure 4-6 Select audio prompt window

6. Select the Audio Prompt file to use from the **Audio Prompt To Use** drop down. Select **from a New File**, see Figure 4-7.



File Extensions required for each Audio Prompt Codec type.

RAW (*.wav)
ADPCM (*.wav)
SBC (*.sbc)
MP3 (*.mp3)
AAC (*.adts)
PCM (*.wav)

Compression Format: ADPCM

Sample Rate (must match content for SBC, MP3 and AAC types): 8000

Stereo (ignored for SBC, MP3 and AAC types)

Figure 4-7 To set an audio prompt from a new file

A file **Open** window appears, see Figure 4-8.

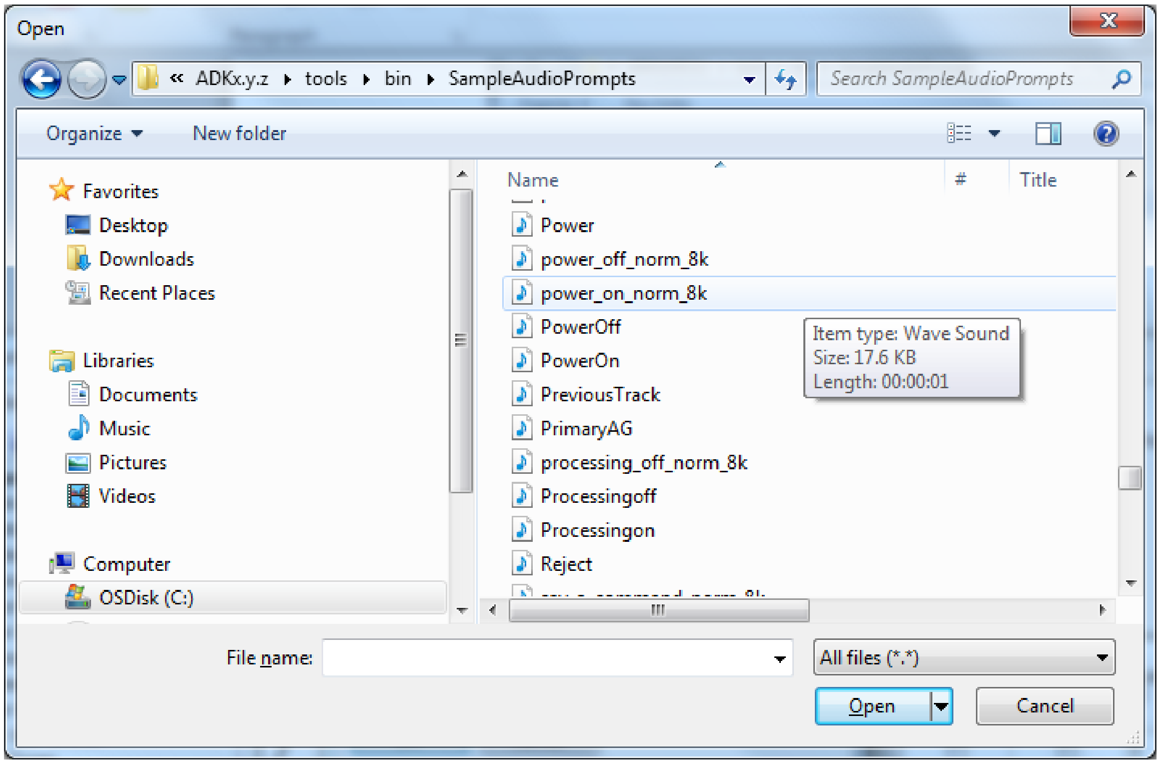
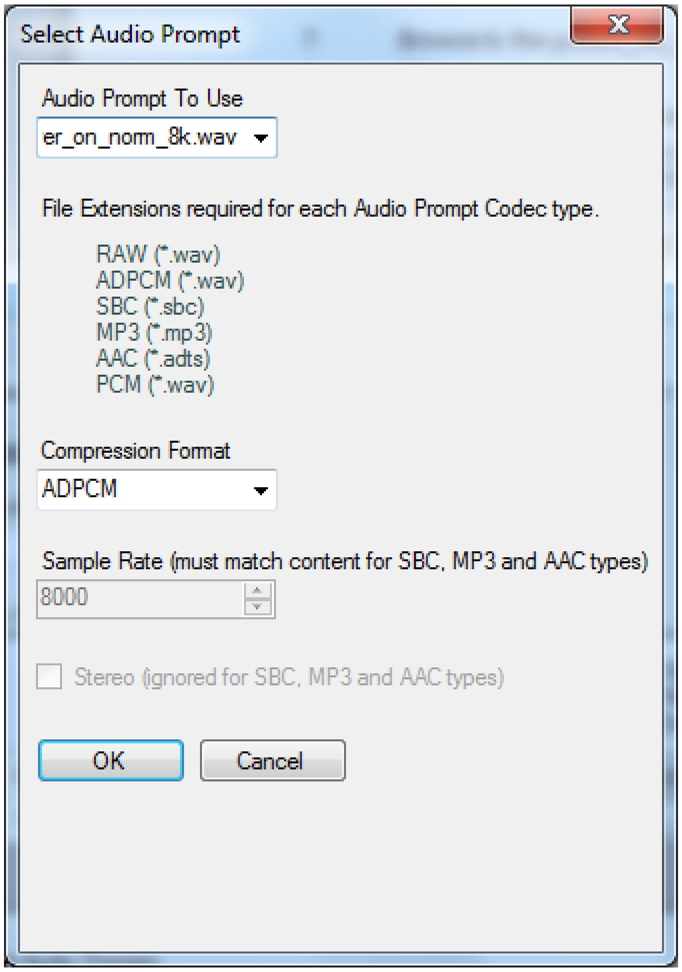


Figure 4-8 Select the power on audio prompt

7. Browse to the `power_on_norm_8k.wav` file in the **SampleAudioPrompts** folder and select it.
8. Click **Open**.
   The **Select Audio Prompt** window appears, see Figure 4-9.



Select Audio Prompt

Audio Prompt To Use: er_on_norm_8k.wav

File Extensions required for each Audio Prompt Codec type.

RAW (*.wav)
ADPCM (*.wav)
SBC (*.sbc)
MP3 (*.mp3)
AAC (*.adts)
PCM (*.wav)

Compression Format: ADPCM

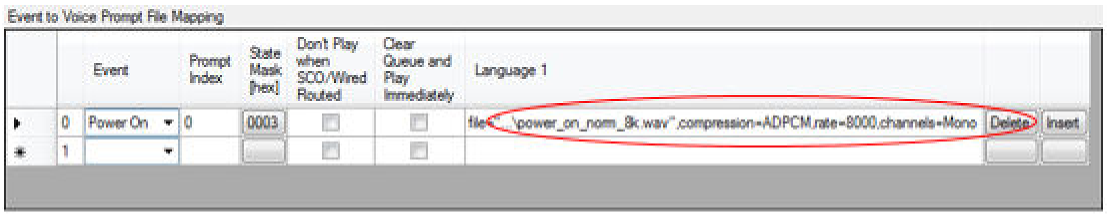
Sample Rate (must match content for SBC, MP3 and AAC types): 8000

☐ Stereo (ignored for SBC, MP3 and AAC types)

Figure 4-9 Default ADPCM audio prompt settings

NOTE In this example, there is no need to change the compression format, sample rate, or stereo flag as the default settings are sufficient in this case.

9. Click **OK** on the **Select Audio Prompt** window.
   The table is then updated with the information for the Audio Prompt being configured and is displayed as text in the **Language 1** cell, see Figure 4-10.



Event to Voice Prompt File Mapping

		Event	Prompt Index	State Mask [hex]	Don't Play when SCO/Wired Routed	Clear Queue and Play Immediately	Language 1		
▶	0	Power On	0	0003	☐	☐	file<...\power_on_norm_8k.wav",compression=ADPCM,rate=8000,channels=Mono	Delete	Insert
✱	1				☐	☐			

Figure 4-10 0 Completed event to voice prompt mapping table

10. Click on the **Generate** button, see Figure 4-11.

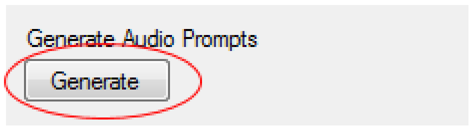
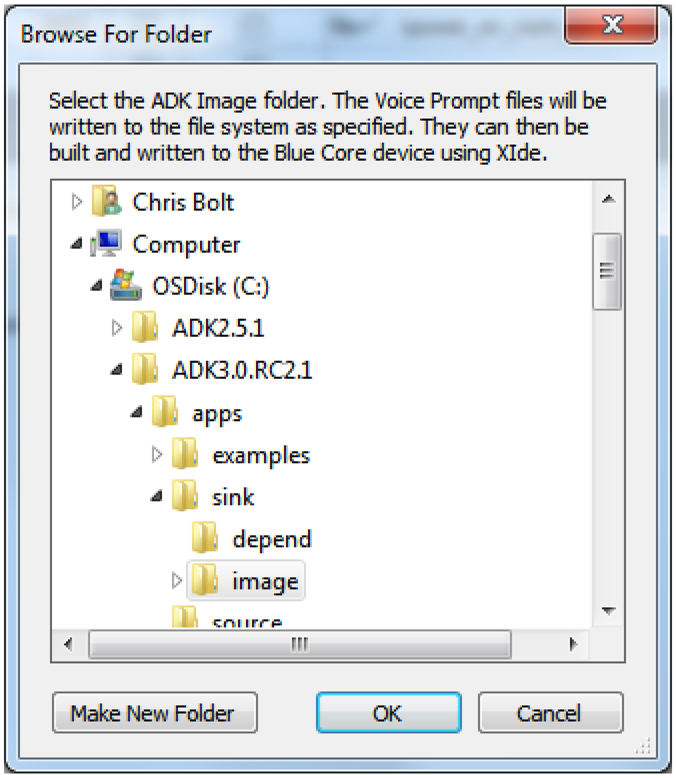
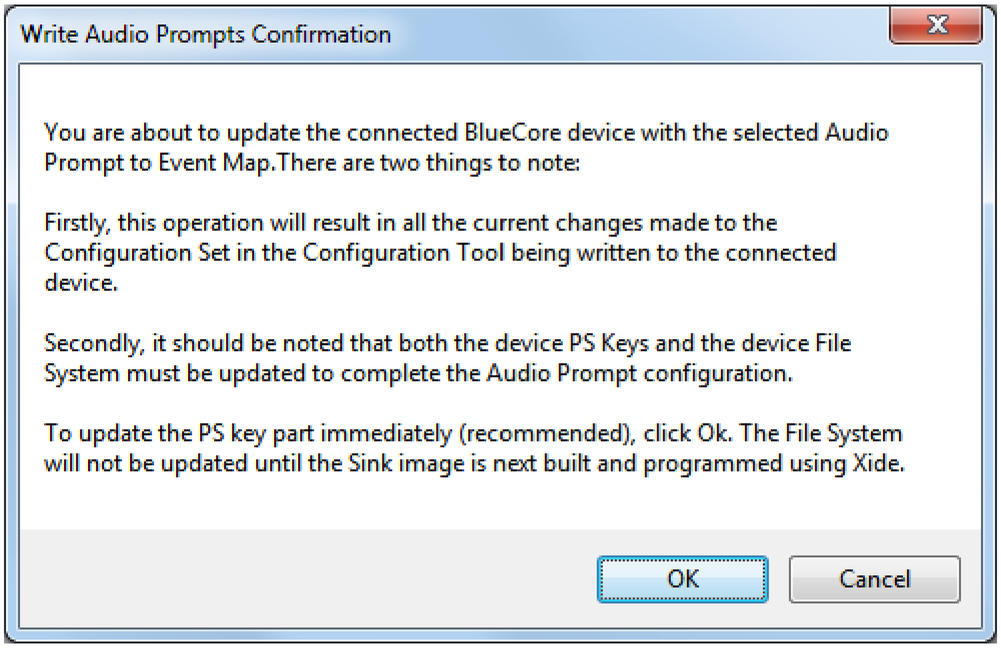


Figure 4-11 1 Generate Audio Prompts button

11. Specify the ADK image folder to be written to the Qualcomm® BlueCore™ device's internal flash when the Sink application is next built in **xIDE**, by selecting the directory from the **Browse For Folder** window that appears, see page 17.
    A **Write Audio Prompts Confirmation** window appears, see Figure 4-12.



Select the ADK Image folder. The Voice Prompt files will be written to the file system as specified. They can then be built and written to the Blue Core device using XIde.


Write Audio Prompts Confirmation

You are about to update the connected BlueCore device with the selected Audio Prompt to Event Map.There are two things to note:

Firstly, this operation will result in all the current changes made to the Configuration Set in the Configuration Tool being written to the connected device.

Secondly, it should be noted that both the device PS Keys and the device File System must be updated to complete the Audio Prompt configuration.

To update the PS key part immediately (recommended), click Ok. The File System will not be updated until the Sink image is next built and programmed using Xide.

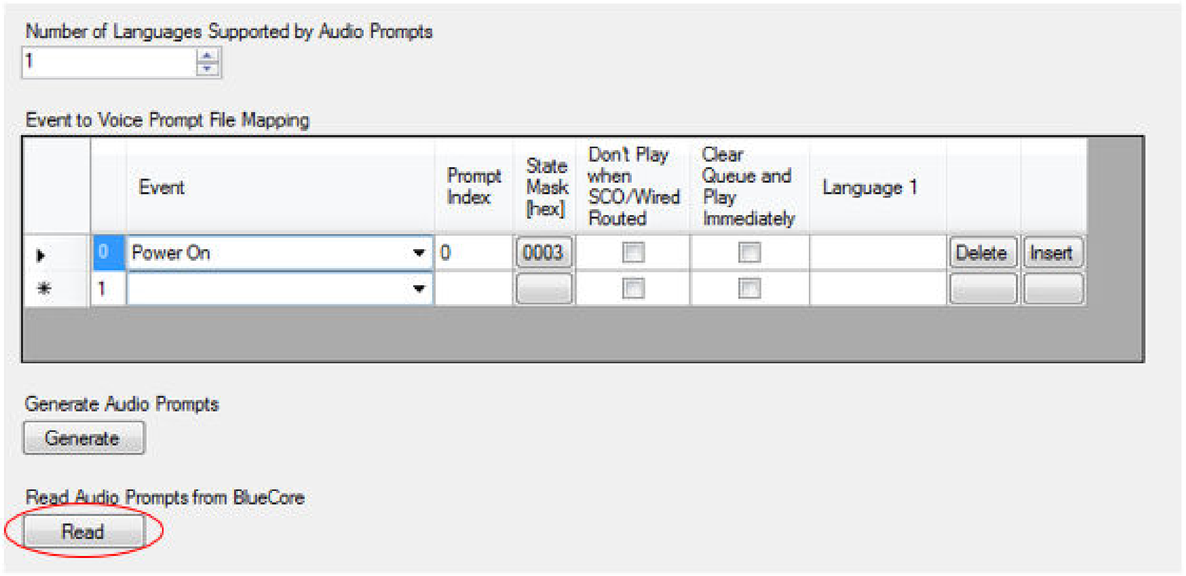
Figure 4-12 3Write Audio Prompts Confirmation window

12. Click **OK**.
13. In **xIDE**, build the audio Sink application by pressing **F5**.
    When the image is burnt to the BlueCore device, turning on the Sink results in the "Power On" audio prompt being played.

4.2 Reading configured audio prompts from a connected device

Using the ADK Configuration Tool it is possible to read from a connected BlueCore device to determine how the Audio Prompts are configured on it.

Because the Configuration Tool normal only reads data from the BlueCore PS Keys, when you read a connected device using the Toolstrip Icon, the Audio Prompts **Event to Voice Prompt File Mapping** table resembles Figure 4-13, (assuming the Power On Audio Prompt is the only configured audio prompt as described in To configure an audio prompt for the power on event).



Number of Languages Supported by Audio Prompts: 1

Event to Voice Prompt File Mapping

		Event	Prompt Index	State Mask [hex]	Don't Play when SCO/Wired Routed	Clear Queue and Play Immediately	Language 1		
▸	0	Power On	0	0003	☐	☐		Delete	Insert
*	1				☐	☐			

Generate Audio Prompts

Generate

Read Audio Prompts from BlueCore

Read

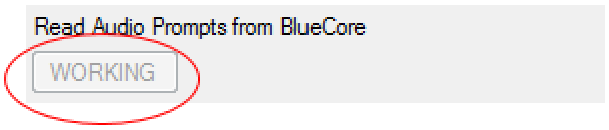
Figure 4-13   4 Reading audio prompts from a connected bluecore device

The **Language 1** entry is blank because the specific Audio Prompt information is stored in the device's internal flash, not the PS Keys. However the ADK Configuration Tool can dump the internal flash image, query it and discover the Audio Prompt configuration that was used.

To do this click the **Read** button, see Figure 4-13.

NOTE Because the entire internal flash image has to be dumped over the USB-SPI interface, this command is quite slow, and takes approximately 30 seconds to execute.

While it is running the cursor changes to the Windows wait cursor, the button is grayed out and shows the text **WORKING**, see Figure 4-14.

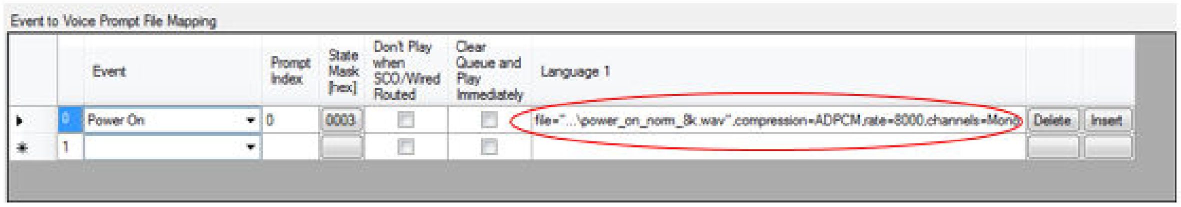


Read Audio Prompts from BlueCore

WORKING

Figure 4-14   5 Read configured audio prompts in progress

When the Read Audio Prompts operation is completed the **Event to Voice Prompt File Mapping** table is updated with the results, see Figure 4-15.



Event to Voice Prompt File Mapping

		Event	Prompt Index	State Mask [hex]	Don't Play when SCO/Wired Routed	Clear Queue and Play Immediately	Language 1		
▸	0	Power On	0	0003	☐	☐	file="...\power_on_norm_8k.wav",compression=ADPCM,rate=8000,channels=Mono	Delete	Insert
*	1				☐	☐			

Figure 4-15   6 The result of the read audio prompt operation

5 Using external flash to store audio prompt files

The Sink application supports the use of an external SQIF to store audio prompts. This allows more storage to enable the use of more prompts, languages or to use higher-quality samples. It is also possible to update audio prompts in the external SQIF over the air allowing end users to customize or update the product.

5.1 Union file system overview

The union file system merges the view of more than one physical file system and presents the VM on-chip application a single coherent file system. Practically for the ADK this relates to merging the file system on the internal flash with one or more file system partitions stored in an external SQIF device.

It is possible for same directory/file to be present on more than one file system. Conflict resolution in such cases follows the laws of UnionFS (Union file system on Linux).

- If the same directory path exists in more than one file system, then they are merged together.
- If the same file name (when full path is included) exists on more than one file system, then priority order is enforced that is, the file in the higher priority file system is presented to the user and the one or more files in other file systems remain hidden.
- If there is a conflict between two directories with same name the entry in the higher priority file system is used i.e. in such cases if a file is found in a higher priority file system, then the whole directory (and sub directories under it) in a lesser priority file system remain hidden, if it is a directory in higher priority file system then the file directory remains visible and the file in a lesser priority file system is hidden.

5.2 Basic SQIF use for audio prompts

The easiest way to store audio prompts in the external SQIF is to create a single file system partition, which is then merged with the internal file system. This allows most efficient use of file system space as the prompts can be split across both internal and external flash.

To do this:

1. Erase the SQIF:

   ```
   nvscmd.exe –usb 0 erase
   ```

2. Create a single file system partition config file, for example:

   ```
   # vp.ptn, File system with initial audio prompts
   0, *, RO, ptn1.xuv
   ```

3. Create the audio prompts using the Sink Configuration Tool. See To configure an audio prompt for the power on event .
4. Pack the audio prompts directories into a file system image, for example, where **audioprompts** contains the **headers** and **prompts** folders generated by the Configuration Tool:

   ```
   packfile.exe audioprompts ptn1.xuv
   ```
5. Write the audio prompts file system to the SQIF:

   ```
   nvscmd.exe -usb 0 burn vp.ptn all
   ```
6. Mount the SQIF file system partition using PSKEY_FSTAB, to mount the internal file system as a single SQIF partition, set the key to:

   ```
   0000 1000
   ```

See *Writing to Serial Flash From a VM Application* for further information on using SQIF tools.

5.3 Over The Air updates

Firmware functions exist so that a VM application can write to a SQIF partition and mount them at run time, see *Writing to Serial Flash From a VM Application* for an overview of these functions.

Combined with a transport mechanism, for example SPP, it is possible for the user to implement a system to update partitions Over The Air (OTA) and therefore update audio prompts on the device.

The ADK Sink application contains a mechanism to update audio prompts OTA using GAIA, this section describes how to do this.

5.3.1 Sink application OTA updates

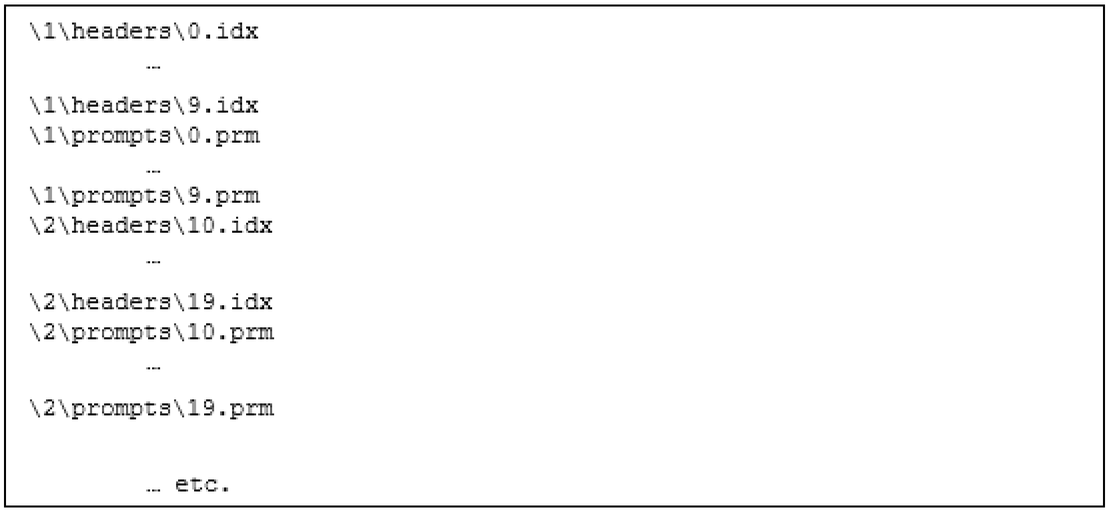
The Sink application allows audio prompts to be updated OTA using GAIA without rebooting. To allow this, the audio prompt language system is used with each language stored in a separate partition in the SQIF.

The method for preparing the SQIF for this is similar to that described in Basic SQIF use for audio prompts except that each language must be separated into its own partition.

Preparing SQIF for OTA updates

To prepare SQIF for OTA updates.

1. Split the audio prompts into their relevant sets based on the number of events configured; so if ten events are configured files 0-9 belong to set one, 10-19 to set two, and so on.
2. Separate the header and prompt files into individual folders, see Figure 5-1.



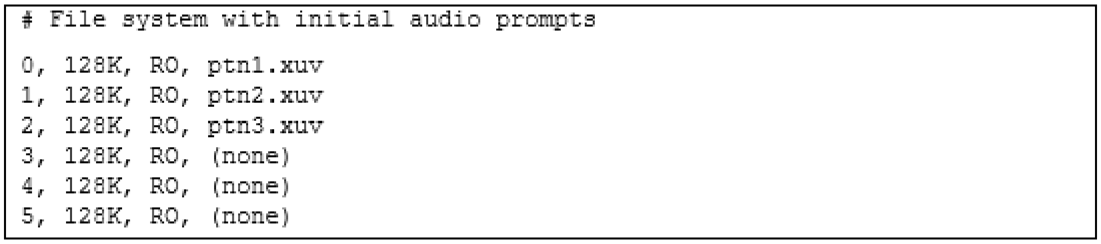
```
\1\headers\0.idx
        …
\1\headers\9.idx
\1\prompts\0.prm
        …
\1\prompts\9.prm
\2\headers\10.idx
        …
\2\headers\19.idx
\2\prompts\10.prm
        …
\2\prompts\19.prm


        … etc.
```

Figure 5-1 Example partition audio prompt folder structure

3. Ensure there are no audio prompt files in the internal flash file system as these take priority over any stored in external SQIF
4. Use the **packfile** utility to make these into partition images. For example:

   ```
   packfile.exe 1 ptn1.xuv
   ```

5. Configure the partitions in the SQIF, in this example six partitions are being created (one for each audio prompt set), the first three contain audio prompts, the last three are empty ready for updates, see Figure 5-2.



```
# File system with initial audio prompts

0, 128K, RO, ptn1.xuv
1, 128K, RO, ptn2.xuv
2, 128K, RO, ptn3.xuv
3, 128K, RO, (none)
4, 128K, RO, (none)
5, 128K, RO, (none)
```

Figure 5-2 Example partition configuration file

A `.ptn` file is created.

NOTE Thought should be given when initializing the partition sizes as they cannot be changed after creation without entirely erasing the SQIF. This cannot be done to a device without development tools so it is not something that should be expected in a production device.

6. The partitions can then be written to the SQIF, see Basic SQIF use for audio prompts.

Configuring Sink Application for OTA Updates

To Configure the Sink application for OTA updates:

1. Add the define `ENABLE_SQIFVP` to the Sink application **Project Properties**.
2. Configure PSKEY_USR47 to inform the application which partitions are free for use. The PS Key is formatted as a bitfield.
   For example `0x0038` (that is, 00111000 binary) indicates partitions 3, 4 and 5 are unused.
3. The currently configured audio prompt sets (languages) may be selected in the application by the user event `EventSelectTTSLanguageMode`.

NOTE The application mounts partitions on demand and skips over any audio prompt partitions marked as free in PSKEY_USR47.
The maximum number of possible languages supported with the required configuration should be set regardless of whether the partition they would reside in is marked as empty, this value is configured in PSKEY_USR9, see *ADK Audio Sink Configuration PS Key Bit Fields.*

5.3.2 Updating audio prompt partitions

Any partition that is not currently mounted by the application may be updated. The implementation in the Sink application only mounts a partition that contains audio prompts currently selected for use.

NOTE When a partition has been mounted in the file system, it may not be unmounted until the CSR8670 is rebooted. Therefore the number of updates that maybe performed without a reboot depends the number of free or unmounted partitions.

For further information on how to write to a SQIF partition from the VM, see *Writing to Serial Flash From a VM Application.*

5.3.3 Partition handling with GAIA

Table 5-1 lists the commands a host application developer can use to update SQIF partitions using GAIA.

See *Overview of the Qualcomm GAIA Ecosystem* and *ADK 4.3 Qualcomm GAIA Command Reference Document* for more information.

Table 5-1 GAIA commands to update SQIF partitions

GAIA Command	Description
`GAIA_COMMAND_GET_MOUNTED_PARTITIONS`	Find partitions currently mounted by the Sink application.
`GAIA_COMMAND_GET_STORAGE_PARTITION_STATUS`	Query Partition Status. Partition's mounted state, type and size can be queried.
`GAIA_COMMAND_OPEN_STORAGE_PARTITION`	Open Partition for Write. Open a sink to stream partition data to.
`GAIA_COMMAND_WRITE_STORAGE_PARTITION`	Write partition data.
`GAIA_COMMAND_CLOSE_STORAGE_PARTITION`	Close partition

Table 5-1 GAIA commands to update SQIF partitions (cont.)

GAIA Command	Description
`GAIA_COMMAND_MOUNT_STORAGE_PARTITION`	Mount Partition. **NOTE** For audio prompt updates, it is not necessary with the current implementation to use this command as it is done automatically by the application.
`GAIA_COMMAND_STORE_PS_KEY`	Update available partitions data. **NOTE** After updating audio prompts the PSKEY_USR47 should be updated. It is expected that at least one partition should be marked as free to allow future updates, but this is decision is left to the user. Any user PS Key can be updated with this GAIA command.

Document references

Document	Reference
Writing to Serial Flash From a VM Application	80-CE754-1/CS-00227835-AN
Overview of the Qualcomm GAIA Ecosystem	80-CT408-1/CS-00211724-DC
ADK 4.3 Qualcomm GAIA Sink Command Reference	80-CF422-1/CS-00406554-DC
ADK 4.3 Source Configuration Tool User Guide	80-CF419-1/CS-00406798-UG

Terms and definitions

Term	Definition
AAC	Advanced Audio Codec
ADK	Audio or Application Development Kit
ADPCM	Adaptive Differential Pulse Code Modulation
ADTS	Audio Data Transport Stream
BlueCore	Group term for the range of QTIL Bluetooth wireless technology ICs
Bluetooth	Set of technologies providing audio and data transfer over short-range radio connections
DSP	Digital Signal Processor
GAIA	Generic Application Interface Architecture
I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
IMA	Interactive Multimedia Association
IO	Input Output
OTA	Over The Air
PCM	Pulse Code Modulation
PS	Persistent Store
QTIL	Qualcomm Technologies International, Ltd.
ROM	Read Only Memory
SBC	Sub Band Codec
SPI	Serial Peripheral Interface
SQIF	Serial Quad I/O Flash
TTS	Text-To-Speech
VM	Virtual Machine