Audio Proxy Server

Note: This API is not part of the public SDK. It can be found in the SDK API Plug-in.

Audio Proxy Server (APS) is a wrapper to the S60 sound device.

Its main purpose is to bypass the Vendor ID (VID) related restrictions on S60 3rd Edition and S60 3rd Edition, FP1 platforms, when streaming audio data in 3rd party applications.

The component is most suitable for VoIP developers who need to set the appropriate audio priority and preference values in order to utilize full duplex audio.

APS supports audio streaming using the following codecs:

- AMR
- G711
- G729
- iLBC
- PCM (APS v1.1 only)

Please note that some devices do not support all of the codecs mentioned above.

APS is available for S60 3rd Edition MR and S60 3rd Edition FP1 releases (since APS v2.0).

The latest version available: **2.43**

Note: * This API is now deprecated. While still functional on S60 3rd Edition FP2 devices it will not be maintained anymore. Developers are advised to adopt the new VoIP Audio Service API
- The VoIP Audio Service API however is not suitable for certain use cases which require custom audio priority and preference values.

Release notes

APS release 2.4.3

- Enabled underflow notifications for the playback stream. When play data queue underflows, client receives a message with status KErrUnderflow (-10) on play communication queue.

APS release 2.4.2

- Fixed an issue with playing audio to CS call uplink in S60 3.1 devices. APS clients can now enable CS uplink playback with following player initialization settings:

```
iPlayerSettings.iSettings.iCaps = TMdaAudioDataSettings::ERoutingControl;
iPlayerSettings.iSettings.iFlags = 0; // (CAudioOutput::ENoPreference)
```

When ERoutingControl capability is set by the client, APS takes the initial output routing preference from iFlags instead of using the default (CAudioOutput::EPrivate). (Note: These flags have no effect on S60 3.0 version of APS, which doesn't have this issue).

- APS installation package now includes a backup & restore registration file, and does not need to be manually reinstalled after restoring from a backup.

APS release 2.41

- Support for AMR encoder configuration (bitrate control, VAD)
- New enumeration values in TAPSCodecMode: EAMRNB4750bps .. EAMRNB12200bps

Known issues:

- VAD mode for AMR cannot be changed while the recording stream is running.
- Cannot playback audio to telephony uplink on S60 3.1 devices:
  - RAPSSession::ActivateLoudspeaker(EFalse) -> Playback is routed to local earpiece only (as expected, since APS sets the audio routing mode to 'private'.
  - RAPSSession::ActivateLoudspeaker(ETrue) -> Playback fails with -21 (KErrAccessDenied) error if there is an active voice call.

APS release 2.31

- Patch to crashing in the middle of an active call.
- Update over an existing APS 2.0 without "update error".

APS release 2.3

- Support for S60 3rd Edition and S60 3rd Edition FP1 devices.

(C) Copyright Nokia 2013. All rights reserved.

- The Server now reports completion of each stream initialization via new callbacks:

TAPSRequest::EAPSPlayerInitComplete and TAPSRequest::EAPSRecorderInitComplete

These callbacks must be caught and processed by CQueueHandler::RunL on the client side. The client should now create and initialize the recorder AFTER receiving TAPSRequest::EAPSPlayerInitComplete. Also, the APS server is ready for streaming only AFTER both callbacks are received by the client.

Client applications must utilize this new initialization procedure as their pre APS 2.3 compatible implementations will not work with APS 2.3 without these changes.

Developers should refer to QueueHandler.cpp file of the bundled example application and make sure their client's implementation is similar to that in CQueueHandler::RunL as well as CStreamerEngine::InputStreamInitialized and CStreamerEngine::OutputStreamInitialized in StreamerEngine.cpp.

This change was necessary as the initialization of the sound device and codecs was not done properly on earlier APS releases, making the initialization and configuration process prone to failures and resulting in potentially unstable device.

- Corrected ITC between the player and the recorder threads.
- Corrected microphone gain settings and returning MaxGain of the device.
- Handled underflow condition during stream playback to protect decoders

from terminating the streaming session when receiving an empty buffer.

- APS errors propagated to the client.
- APS 2.3 can now be installed over an existing APS without "update error".
- Updated streamer app
- creating player and recorder in sequenced fashion by utilizing

new callbacks from the APS (TAPSRequest::EAPSPlayerInitComplete and TAPSRequest::EAPSRecorderInitComplete)

- added Audio Settings mainly for gain and volume settings testing
- added more comments to help client app developers
- better error and invalid selections handling

Known issues: Previous APS 2.x installations must be uninstalled prior to installing APS 2.3. An 'Update error' -note may otherwise be shown on the screen and the installation will terminate.

APS release 2.2

- AMR codec support added. Note: The AMR configuration cannot be changed

due to lack of API support.

- PCM codec support removed from the menus. PCM hasn't been working with

APS 2.x releases due to 164-byte buffer limitation.

- Removed 2-3 sec delay between the audio recorded and played back.
- Fixed stream start failing when requested first time.
- Improved Test application funcionality.
- Fixed server crash on application exit without initializing.
- Fixed server crash on application exit after initializing but never

starting streaming.

- Improved documentation.

APS release 2.1 (First release from S60 R&D)

- Improved documentation (e.g. sequence diagrams included)
- Solves memory leak problems identified in APS 2.0

APS release 2.0 (Nokia Developer)

- Added support of G711, G29 and iLBC codecs
- Solves the delay (from 300ms to 30ms)

APS release 1.1 ("VID hack")

- Supports only S60 3.0 devices.
- Solves audio mixing problem with CS call
- Solves audio routing problem for VoIP, but does not solve the 300ms delay

caused by 4KB buffer in the adaptation layer.

Download

Audio Proxy Server API headers and libraries can be found in the SDK API Plug-in.

The latest version of the Audio Proxy Server SIS file can be downloaded from [[File:Apsserver2.zip]].

Links

- [[Archived:Audio routing limitations on S60 3rd Edition (Known Issue)]]

(C) Copyright Nokia 2013. All rights reserved.

--

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.