Audio Recording APIs

The MMF (Multimedia Framework) provides a consolidated API for both simple audio playback and recording operations. The library used for audio recording is the same as the one used for audio playback; however, the classes used are different. For audio recording, the classes are:

- **CMdaAudioRecorderUtility** — The audio recorder API supporting audio file manipulation and recording operations in MMF.
- **CMdaAudioPlayerUtility** — The audio player API that supports audio playback operations and simple metadata retrieval operations.
- **MMdaObjectStateChangeObserver** — The callback API for the CMdaAudioRecorderUtility class that reports errors and state-change events during record and playback operations.

These classes are found in the **MediaClientAudio.lib** library.

Note that CMdaAudioPlayerUtility is also listed as a class used for audio recording operations, however its use is optional. The CMdaAudioRecorderUtility class has limited audio playback functionality, so where functions such as setting the volume ramp or metadata retrieval are required, CMdaAudioPlayerUtility will need to be used.

The CMdaAudioRecorderUtility class inherits the same set of basic audio operations, such as PlayL(), Stop(), and OpenFile(), as supported in the CMdaAudioPlayerUtility class, but with the addition of methods for audio clip manipulations, such as Crop() and Record().

Similarly to audio playback, when implementing with the CMdaAudioRecorderUtility class it is necessary to inherit and derive the abstract observer class, which consists of a single method:

```
void MoscoStateChangeEvent(CBase* aObject,
                           TInt aPreviousState,
                           TInt aCurrentState,
                           TInt aErrorCode);
```

This callback function will be invoked whenever there is a state change in the CMdaAudioRecorderUtility instance created.

Steps involved in Record Operation:

```
//Create an instance of CMdaAudioRecorderUtility class
CMdaAudioRecorderUtility iRecorder = CMdaAudioRecorderUtility::NewL(*this);

//Open file either by using a filename with full path infor or a descriptor
iRecorder->OpenFileL( aFilename );
//iRecorder->OpenDesL( aDescriptor );

//Recording is started
iRecorder->RecordL();
....
....
//Stop Recording
iRecorder->Stop();
//Close the recording session
iRecorder->Close();
```

Note: During creation of the CMdaAudioRecorderUtility class instance, the priority and preferences for the instance can be set. These arguments are optional and the default values are EMdaPriorityNormal and EMdaPriorityPreferenceTimeAndQuality. It is strongly recommended to use higher priority than the default. Priority of 80 (100 is maximum) is usable for recording.

It should be noted that the standard MMF functions do not allow recording to be undertaken during a phone call.

(C) Copyright Nokia 2014. All rights reserved.

Links

- Playing audio files
- Audio Playback APIs
- Audio playing, recording and conversion overview
- Audio and Video Documents

--

(C) Copyright Nokia 2014. All rights reserved.