

EasyAudio Programming Guide

IEasyAudioCalibratedProfile Interface

Version: Juiwen Hsu@2013-0410

Contents

Contents	1
Introduction.....	1
Document Revision.....	1
Feature.....	2
Calibration Profile Storage Mechanism.....	3
Prepare development environment	3
IEasyAudioCalibratedProfile Interface.....	3
Methods.....	4
HRESULT GetCalibratedProfileFilename(LPTSTR filename);.....	4
HRESULT EnableCalibratedProfile(BOOL bVal);.....	4
HRESULT GetCurrentProfile(double & Frequency, double & Amp);.....	5
Example with IEasyAudioCalibratedProfile.....	6
Verification.....	7
Without hardware (no speaker, no MIC and no fixture).....	7
With hardware (include speaker, MIC and fixture)	8
EasyAudioCalibratedProfile.cfb	8
Where to find the sample file of EasyAudioCalibratedProfile.cfb?	8
Reference	9
EasyAVEngine-Programming-Guide.....	9
EasyAudio-Programming-Guide-IEasyAudioStdWaveGenerator.....	9
EasyAudio-Tone-Source-Calibration.....	9

Introduction

EasyAudio provides an **IEasyAudioCalibratedProfile** interface to output calibrated waveform with **EasyAudioSTDWaveGenerator**.

Document Revision

Feature

EasyAudio Calibration mechanism separate the calibrated waveform output to two parts.

First part is to generate the calibrated profile file “**EasyAudioCalibratedProfile.cfb**”

Second part is to apply the “**EasyAudioCalibratedProfile.cfb**” to output the calibrated waveform to hardware speaker.

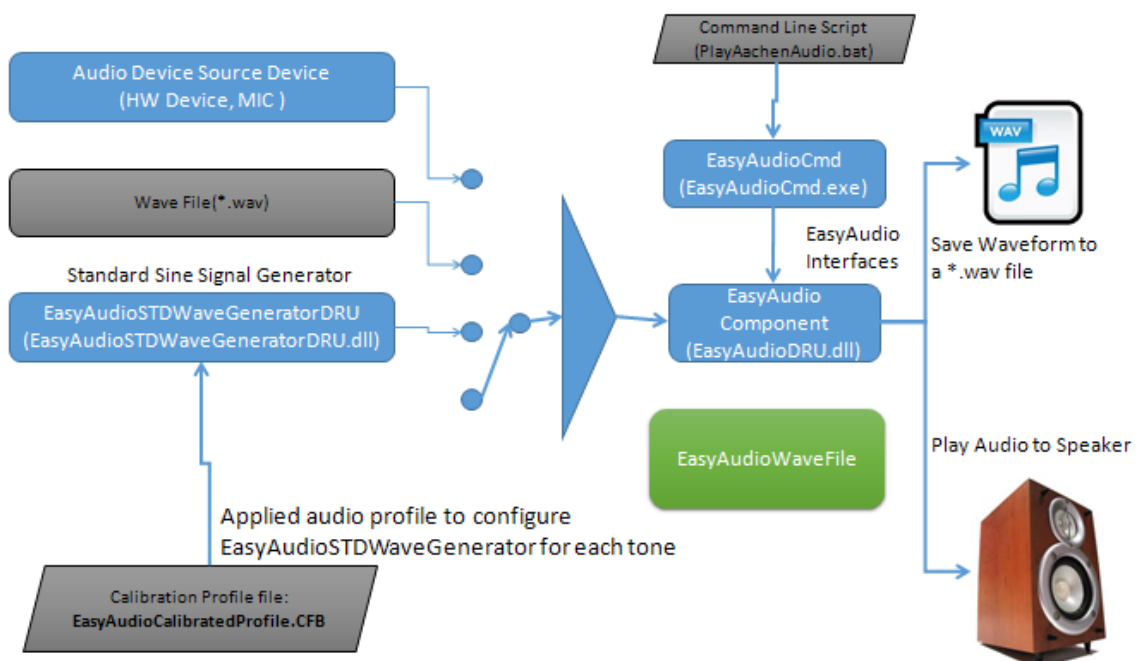
During development time, developers only need to take care how to enable the calibration mechanism.

And another task such as setup the fixture with speaker hardware to generate the calibrated profile is own by the end user.

To reduce the effort of development, developers don't need to care about generate the calibration profile during development time.

Software application developers only need to ask the file “**EasyAudioCalibratedProfile.cfb**” from TDE end user.

Tone Source Calibrated profile with EasyAudio

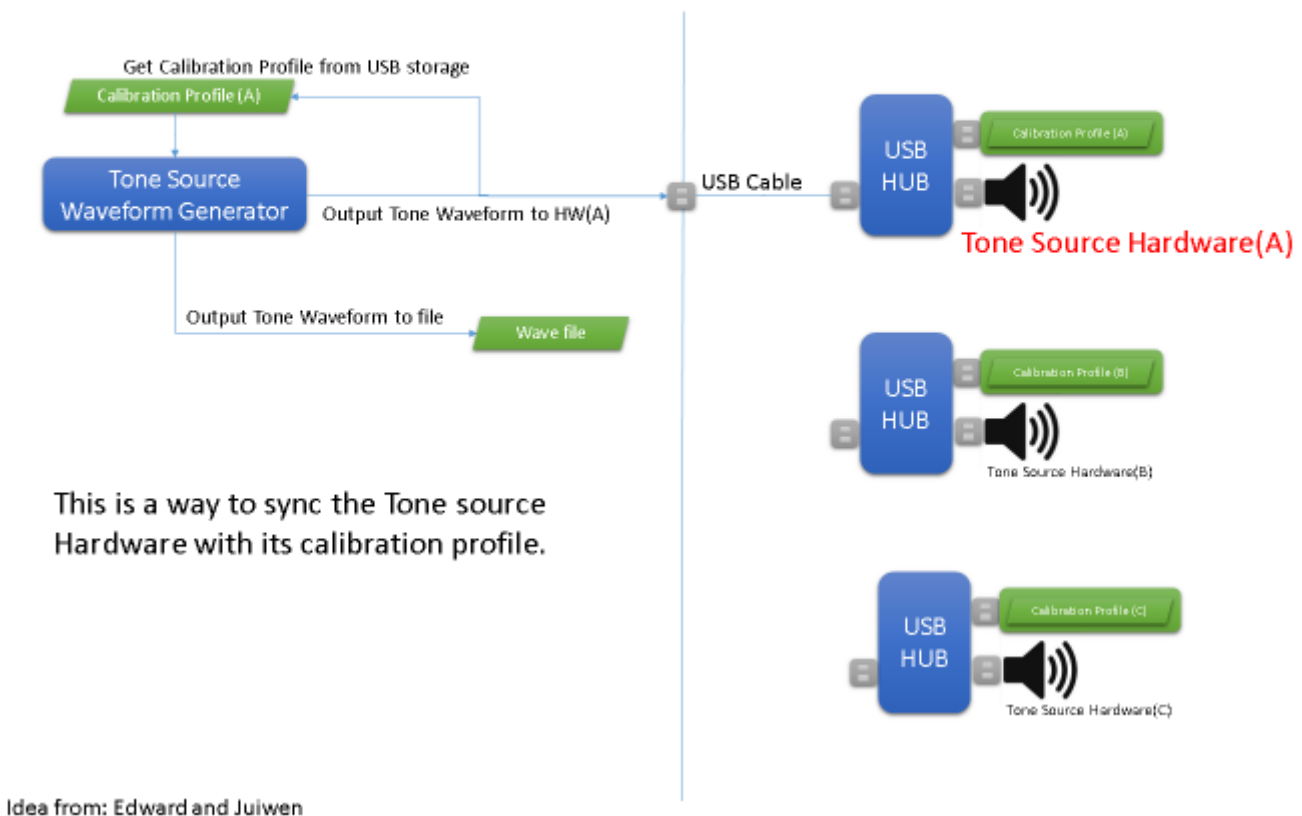


The EasyAudioCalibratedProfile.CFB can be generated from EasyAudio speaker Tone calibration tool

Calibration Profile Storage Mechanism

Calibration Profile Storage

EX: Work with **Tone Source HW(A)**



Prepare development environment

Please refer the document: “**EasyAVEngine-Programming-Guide**”

IEasyAudioCalibratedProfile Interface

IEasyAudioCalibratedProfile interface provides methods to let EasyAudio into calibration mechanism.

Methods

`HRESULT GetCalibratedProfileFilename(LPTSTR filename);`

Description:

Get the full path filename of EasyAudioCalibratedProfile.cfb.

Parameters:

filename: the zero-end string which contains the full path filename of EasyAudioCalibratedProfile.cfb.

If the filename contains empty string, it means “cannot found the calibrated file” and will not into calibration mechanism.

Return:

Returns an **HRESULT** value. Possible values include the following.

Return code	Description
E_FAIL	This method failed.
S_OK	Success.

Example:

`HRESULT EnableCalibratedProfile(BOOL bVal);`

Description:

Enable or disable the calibration profile mechanism.

Parameters:

bVal:

bVal=TRUE is enable the calibration profile mechanism.

bVal=FALSE is disable the calibration profile mechanism.

Return:

Returns an **HRESULT** value. Possible values include the following.

Return code	Description
-------------	-------------

E_FAIL	This method failed.
S_OK	Success.

HRESULT GetCurrentProfile(double & Frequency, double & Amp);

Description:

After the client enable the calibration profile mechanism via **EnableCalibratedProfile()**, the client application can get back those frequency and amplitude values by using GetCurrentProfile() method.

Parameters:

Frequency: the frequency value which set via **IEasyAudioSTDWaveGenerator** interface's method SetSTDWaveFormat(). This value is equal SignalFrequency's value.

HRESULT SetSTDWaveFormat(UINT SignalType, double SignalFrequency, double SignalAmplitude, double AmplitudeOffset);

Amp: if the Calibration mechanism is enable and also the file "EasyAudioCalibratedProfile.cfb" is appear. EasyAudio Calibration will look-up the correct amplitude value inside "**EasyAudioCalibratedProfile.cfb**" then set it to the amplitude of the sine waveform from EasyAudioSTDWaveGenerator.

Return:

Returns an **HRESULT** value. Possible values include the following.

Return code	Description
E_FAIL	This method failed.
S_OK	Success.

Example with IEasyAudioCalibratedProfile

EasyAudio COM Component

Step(1): Configure Sine Wave Generator via IEasyAudioSTDWaveGenerator interface with below methods

SetSTDWaveFormat(UINT SignalType, double SignalFrequency, double SignalAmplitude, double AmplitudeOffset)

Note: Please refer example inside function **SamplesEasyAudioSTDWaveGeneratorWithCalibration ()** in **EasyAudioATLConsoleApp** Project

Step(2): Configure Calibrated profile via IEasyAudioCalibratedProfile interface with below methods

GetCalibratedProfileFilename (LPTSTR filename);

EnableCalibratedProfile(BOOL bVal);

GetCurrentProfile(double & Frequency, double & amp);

Note: Please refer example inside function **SamplesEasyAudioSTDWaveGeneratorWithCalibration ()** in **EasyAudioATLConsoleApp** Project

Step(3): Configure and play audio via IEasyAudioControl interface with below methods

SetAudioSource (LPTSTR AIS, WORD nChannels=2,WORD wBitsPerSample=16, DWORD nSamplesPerSec=16000, double AIDVolume=1.0);

SetAudioOutDevice (LPTSTR AOD, long OutVol=0);

SetAudioOutWaveFile (LPTSTR AOWF, ULONG DelaySamplingCount, ULONG SamplingCount);

Play (BOOL bWaitCaptureTime=FALSE);

Note: Please refer example inside function **AppSTDWaveGen2WF_SPK()** in **EasyAudioATLConsoleApp** Project

The **EasyAudioATLConsoleApp** Project you can open it from
EasyAVEngine\Developers\Samples\Build\EasyAVEngine\Samples_VS2005.sln

You can find this sample code from visual studio solution file:

EasyAVEngine\Developers\Samples\Build\EasyAVEngine\Samples_VS2005.sln

With project name: **EasyAudioATLConsoleApp**

Inside the sample function: **SamplesEasyAudioSTDWaveGeneratorWithCalibration()** will show up how to use **IEasyAudioCalibratedProfile**.

```
void SamplesEasyAudioSTDWaveGeneratorWithCalibration(void)
{
    CEasyAudioHelper eat;
    //AID audio waveform format

    =====

    eat.m_nChannels=1; // EasyAudio standard wave generator support 1 channel.
    eat.m_wBitsPerSample=16;
    eat.m_nSamplesPerSec=16000;
    eat.m_AIDVolume=1.0;
    //AOD properties =====
    eat.m_AODVolume=0; // 0 means full value
    //WF Out =====
    eat.m_DelaySamplingCount=000; /* avoid samples after play */
    eat.m_SamplingCount=24000; /* samples to get */
    //
```

```

eat.SetWaveFileSource(_T(""));
//
eat.SetAudioOutDeviceName(_T("")); //the output device name
//=====
// Assign the source name as "EasyAudioSTDWaveGenerator"
eat.SetMicDeviceName(_T("EasyAudioSTDWaveGenerator"));
//Assign wave ouput filename
eat.SetWaveOutFileName(_T("EasyAudioFormat(16000Hz-16Bits-1Ch)_Signal(Sine_800Hz)_Out(WF_SPK).wav")
);
// signal type, SignalFrequency, SignalAmplitude, AmplitudeOffset
#define SINEWAVE 0
UINT SignalType=SINEWAVE;
eat.pSWG->SetSTDWaveFormat(SignalType, 100, 0.5, 0.0);

// below section is work for calibration profile
{
    CComPtr<EasyAudio::IEasyAudioCalibratedProfile> cpf;
    hr=eat.pSWG.QueryInterface(&cpf);
    CString str;
    cpf->EnableCalibratedProfile(TRUE);
    LPTSTR pbf=str.GetBufferSetLength(255);RtlZeroMemory(pbf,255);
    cpf->GetCalibratedProfileFilename(pbf);
    str.ReleaseBuffer();
    if(!str.IsEmpty())
    {
        _tprintf(_T("\nCalibrated Profile Filename=%s"),str);
        double f,a;
        cpf->GetCurrentProfile(f,a);
        _tprintf(_T("\nFrequency=%f, Amplitude=%f"),f,a);
    }
}

eat.AppSTDWaveGen2WF_SPK(eat.GetEasyAudioControl()); //STDWaveGen in and output to speaker
}

```

Verification

Without hardware (no speaker, no MIC and no fixture)

Remove the “**EasyAudioCalibratedProfile.cfb**” from the system then see the amplitude changed.

Add the file “**EasyAudioCalibratedProfile.cfb**” into the system then see the amplitude changed.
Change the sine waveform frequency, generate *.wav file then see the amplitude changed.

With hardware (include speaker, MIC and fixture)

Add the file “**EasyAudioCalibratedProfile.cfb**” into the system.

Change the sine waveform frequency and play out to the hardware speaker.

Capture waveform from a standard MIC.

See the waveform amplitude for each different frequency, each frequency’s amplitude should approximate to 1000Hz’s amplitude.

If not, please ask a new “**EasyAudioCalibratedProfile.cfb**” for this speaker.

EasyAudioCalibratedProfile.cfb

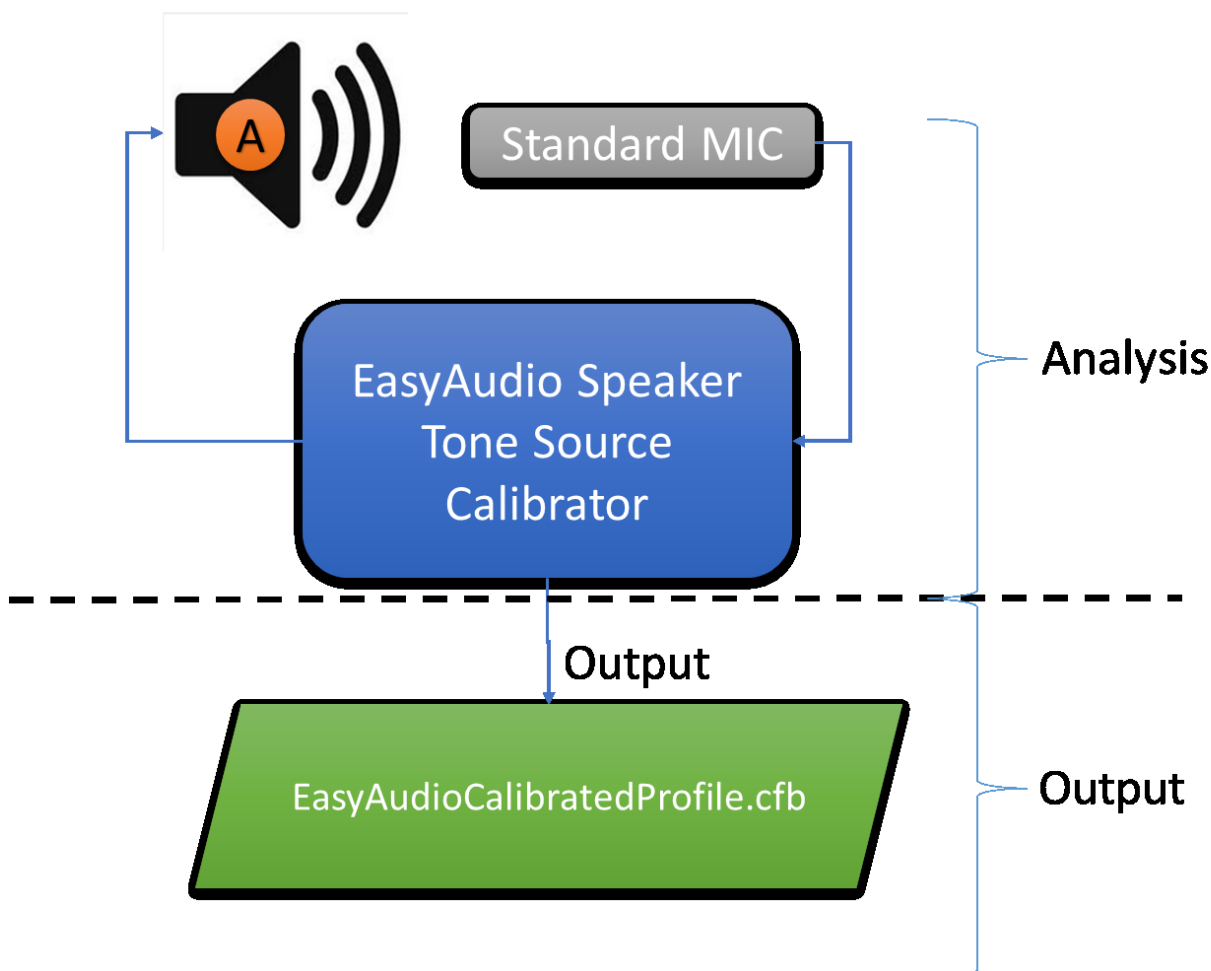
Where to find the sample file of EasyAudioCalibratedProfile.cfb?

You can find it in one of the location:

\\us01f01\video\video\Hardware_Engineering\TDE_Collections\EasyAVEngine\EndUsers\Docs\EasyAudio-Tone-Source-Calibration\

or

\\corp.logitech.com\Video\TDE_TRANSFER_DRIVE\EasyAVEngine\EndUsers\Docs\EasyAudio-Tone-Source-Calibration\



Reference

[EasyAVEngine-Programming-Guide](#)

[EasyAudio-Programming-Guide-I](#)[EasyAudioStdWaveGenerator](#)

[EasyAudio-Tone-Source-Calibration](#)