How to use the EqualizerControl in the Java ME AMMS API

This article explains how to use the EqualizerControl in the AMMS API.

02 Mar 2014

Introduction

When playing back music or any other sound, it is sometimes nice adjust the sound, for example for a little more bass for that techno beat. For these adjustments and equalizer is used and the AMMS API provides an `EqualizerControl` that can be used to filter your sound trough an Equalizer.

The EqualizerControl can be used in three different ways depending on how complex adjustments your application needs to make or how complicated interface you would like to present to the user.

1. Presets - Choose from a preset list
2. Bass and Treble - Adjust bass and treble, like on stereo set
3. Bands - Adjust the individual frequency bands

Getting started

The `EqualizerControl` is not specific to your `Player`, but is accessed from a `GlobalManager` class.

First import the AMMS classes you need

```
import javax.microedition.amms.*;
import javax.microedition.amms.control.*;
import javax.microedition.amms.control.audioeffect.*;
```

Then in your code request the Controller:

```
 private EqualizerControl equalizerControl = null;
 ...

equalizerControl=(EqualizerControl)GlobalManager.getControl("javax.microedition.amms.control.audioeffect.EqualizerControl");
 if (equalizerControl!=null) {
  // Enable the Equalization
  equalizerControl.setEnabled(true);
 } else {
  // Device does not support EqualizerControl
 }
```

As you can see from the code above, you call the static `getControl()` method of the `GlobalManager` class and request the Controller you are looking for. The controller we are looking for is named **javax.microedition.amms.control.audioeffect.EqualizerControl**.

As default the Equalizer is not enabled, so to be able to hear any difference in the sound we need to enable it with `setEnabled()`.

Now you can start playing around with the settings of the controller.

Presets

This is the most easiest setting, where you set the Equalization from a predefined list on the device. You can query the list of available preset with `getPresetNames()`, query the current preset with `getPreset()` and set the preset with `setPreset()`. The `getPresetNames()` methods returns an array of strings, you use one of these strings as the argument to the setPreset() method of the control.

Available presets

The available presets can vary from device to device, but for example the Asha 311 returns this list:

1. normal
2. pop
3. rock
4. jazz
5. classical

Example code snippets

Code snippet showing how to get the available equalizer presets and adding them to a ChoiceGroup.

```
private String [] presets;
private ChoiceGroup presetChoice;
...
presets = equalizerControl.getPresetNames();
presetChoice = new ChoiceGroup("Presets", Choice.EXCLUSIVE);
if (presets != null && presets.length > 0) {
for (int i = 0; i < presets.length; ++i)
 presetChoice.append(presets[i], null);
}
presetForm.append(presetChoice);
presetForm.setItemStateListener(this);
```


(C) Copyright Nokia 2014. All rights reserved.

Handling the ChoiceGroup selections in a itemStateChanged listener:

```
...
if (item == presetChoice) {
 equalizerControl.setPreset(presetChoice.getString(presetChoice.getSelectedIndex()));
}
...
```

Bass and Treble

Bass and Treble are set with the `setBass()` and `setTreble()` methods. The methods both take only one integer value as argument, with the range 0-100. The value 50 is the neutral setting, a value of 0 will apply the maximum attenuation the frequencies and 100 will apply maximum amplification to the frequencies.

```
 equalizerControl.setBass(bass);
 equalizerControl.setTreble(treble);
```

You can query the current bass or treble using the `getBass()` and `getTreble()` methods. If the equalizer settings have been adjusted in any other way (preset, bands) the value `EqualizerControl.UNDEFINED` will be returned.

Example code snippets

As the values go from 0-100 you can easily use a Gauge directly for the values. A 100 steps might be a bit too much, so scaling that down to perhaps 10 might be good.

```
bassGauge = new Gauge("Bass", true, 10, 5);
trebleGauge = new Gauge("Treble", true, 10, 5);
...
public void itemStateChanged(Item item) {
        if (item == bassGauge) {
            equalizerControl.setBass(bassGauge.getValue()*10);
        } else if (item == trebleGauge) {
            equalizerControl.setTreble(trebleGauge.getValue()*10);
        }
}
```

Frequency bands

For total control of the sound you can use the frequency band controls. The amount of bands (using the getNumberOfBands() method), the band they effect (the getCenterFreq() method) and range you can adjust (the getMinBandLevel() and getMaxBandLevel() methods) varies from device to device. You must therefore query the control for these values before use.

```
int bands = equalizerControl.getNumberOfBands();
int bandMinLevel = equalizerControl.getMinBandLevel();
int bandMaxLevel = equalizerControl.getMaxBandLevel();
```

To get the frequency band that a particular band index adjusts, use getCenterFreq(index) method.

```
int freq=equalizerControl.getCenterFreq(i);
```

The current value of a band can get queried with getBandLevel(band) and a new value can be set with setBandLevel(level, band) method. The level value when setting must be in the min-max range.

```
level = equalizerControl.getBandLevel(band);
...
equalizerControl.setBandLevel(level, band);
```


(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.