Archived:Vibration Control API on Symbian

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Warning: From S60 3rd Edition you should instead use the *HWRM Vibra API (HWRMVibra.h)*. From S60 3rd Edition this API has been deprecated

Introduced in S60 2nd Edition, this public interface allows you to control the device vibration feature. Vibration might be used in an application such as a game to signal a collision or an explosion, or for giving tactile feedback to other game events, such as resonance when a ball hits an object such as a racquet or a bat.

To use this API, include the vibractrl.h header in the source code and the vibractrl.lib in the .mmp file. The factory functions to construct an instance of the control object are:

```
CVibraControl* NewL()
CVibraControl* NewL(MVibraControlObserver *aCallback)
CVibraControl* NewLC(MVibraControlObserver *aCallback)
```

Some key methods are:

```
virtual TInt StartVibra(TUint16 aDuration) = 0
virtual TInt StopVibra(void) = 0
virtual TVibraModeState VibraSettings(void) const = 0
```

`StartVibra()` initiates the device vibration feedback, where `aDuration` is the interval in milliseconds. A value of 0 specifies that the vibration should continue indefinitely. Vibration can be stopped before the specified duration has elapsed with a call to `StopVibra()`.

The `StartVibra()` method does not block, but returns immediately, so that the vibration happens simultaneously as the application continues to run. If StartVibra() is called again, before the first vibration completes, then the first vibration is interrupted and the second vibrations starts immediately — the periods of vibration are not cumulative.

The vibration settings in the user profile must be active for the vibration to be activated via this API. Also, specific S60 devices may have implementation-defined or hardware-imposed limits to the duration of the vibration feature (to conserve power). In such circumstances any vibration will cut off at that limit, even if the duration parameter is greater than the limit.

Use `VibraSettings()` to retrieve the current vibration settings from the current user profile. Then, if vibration is not active but is needed by the application — typically a game — the user could be informed and offered the option to enable the vibration feature for the duration of the game. This method returns `TVibraModeState` with the possible states of `EVibraModeON`, `EVibraModeOFF` or, if an error occurs, `EVibraModeUnknown`.

The mixin interface MVibraControlObserver class declares:

```
virtual void VibraModeStatus(CVibraControl::TVibraModeState aStatus) = 0
virtual void VibraRequestStatus(CVibraControl::TVibraRequestStatus aStatus) = 0
```

`VibraModeStatus()` is called when the vibration setting in the user profile is changed, so your application can be aware of changes the user makes. `VibraRequestStatus()` is called when the device vibration feature is requested.

The argument `aStatus` is a `TVibraRequestStatus` and indicates the current `VibraControl` request status. In addition to the return value supplied by the `StartVibra()` method (Symbian OS standard error codes), more detailed status information is returned by the callback method `MVibraControlObserver::VibraRequestStatus()`. Possible values for aStatus are `EVibraRequestOK`, `EVibraRequestFail`, `EVibraRequestNotAllowed`, `EVibraRequestStopped`, `VibraRequestUnableToStop` and `EVibraRequestUnknown`.

(C) Copyright Nokia 2014. All rights reserved.

Note:

- The DLL that implements the API is not included in Nokia 6600.
- S60 2nd Edition, FP2 and FP3 devices include the VibraCtrl.dll and vibra functionality is supported.
- S60 3rd Edition provides a new Vibra Client API (CHWRMVibra).

(C) Copyright Nokia 2014. All rights reserved.