

Request #: HUTRR106
Title: Call State Management Control
Spec Release: 1.22
Requester: Matthew Williams
Company: Microsoft

Pages Affected: Generic Desktop (0x01)
Values checked: By chair (Matthew Williams)

Current Status: **Approved**
Priority: Normal

Required Voter: Apple
Required Voter: Google
Required Voter: Logitech

Voting Begins: 30th November 2021
Voting Ends: 9th December 2021
Voting Result: 4-0

Summary:

Add new Usages to the Generic Desktop Page (0x01) to support the control and display of a system-level 'call' state (for voice/video conference 'calls').

Additional:

System environments can offer rich integration of voice/video conferencing applications within the Shell and system-wide management of 'calls' by having a single system-wide 'call' state. This state can then be modified via a native Shell experience or through a brokered device ("Call State Management Control"). Such a device could be standalone or even integrated into keyboards as are some system-control devices today (e.g. for power/sleep). "Call Mute", is NOT intended to be a replacement for existing functionality on existing Audio/Telephony headsets. It is NOT expected, for "Call State Management Control" devices to be combined/integrated with Telephony Devices or Audio Input devices.

Unlike existing Telephony controls (which reflect the state of a specific audio device, (e.g. headset/speakerphone) associated with the controls) the below are for system consumption, and is up to the system to decide how best reflect the 'call' state across devices affected by a 'call' (e.g. Telephony headset, generic standalone microphone, webcam, non-HIDs, etc...).

Scenario:

A user has an active 'call' within a voice-conferencing application, which causes the system to activate the "Call Active LED". The user activates the "Call Mute Toggle" control to mute their channel (e.g. to sneeze/cough). The system updates its internal state to reflect the 'call' is now muted, and either informs the voice-conferencing application that it should mute the user's channel, or mute the channel itself (or take similar other steps). Once completed, the system can reflect the updated state to the device by activating the "Call Mute LED" control, indicating to the user the call has been muted. The user can then unmute by activating the "Call Mute Toggle" control again, or using software controls synchronized to the 'call' state (if supported by the system).

Proposal:

Add-to/modify Table 4.1: Generic Desktop Page

Usage Id	Usage Type	Usage Name
0x13	CA	Call State Management Control
0x14-0x2F		<i>Reserved</i>
--	--	--
0xD7-0xDF	-	<i>Reserved</i>
0xE0	OOC	Call Active LED
0xE1	OSC	Call Mute Toggle
0xE2	OOC	Call Mute LED
0xE3-0xEF	-	<i>Reserved for Call State Management Control.</i>
0xF0-0xFFFF	-	<i>Reserved</i>

New section to be added with the following text:-

Section 4.16:

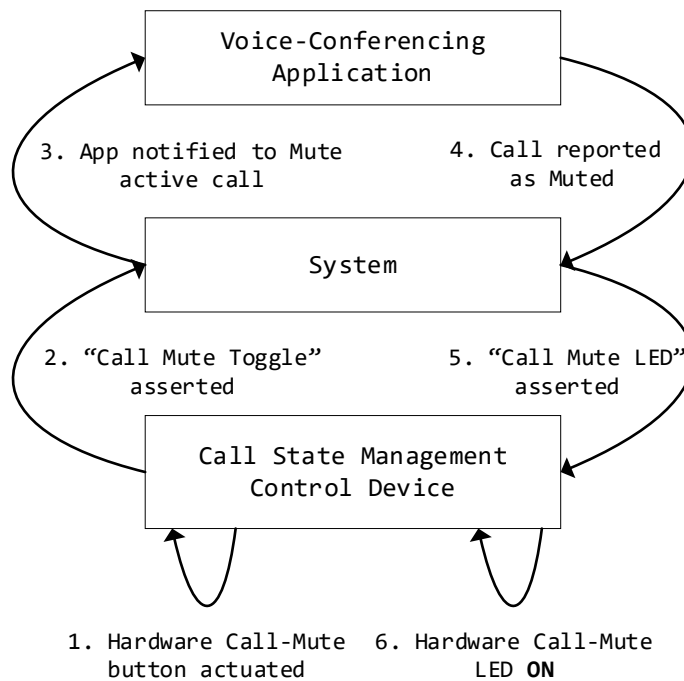
System environments can offer rich integration of voice/video conferencing applications within the Shell and system-wide management of ‘calls’ by having a single system-wide ‘call’ state. This state can then be modified via a native Shell experience or through a brokered device (“Call State Management Control”). Such a device could be standalone or even integrated into keyboards as are some system-control devices today (e.g. for power/sleep). “Call Mute”, is NOT intended to be a replacement for existing functionality on existing Audio/Telephony headsets. It is NOT expected, for “Call State Management Control” devices to be combined/integrated with Telephony Devices or Audio Input devices.

Unlike existing Telephony controls (which reflect the state of a specific audio device, (e.g. headset/speakerphone) associated with the controls) the below are for system consumption, and is up to the system to decide how best reflect the ‘call’ state across devices affected by a ‘call’ (e.g. Telephony headset, generic standalone microphone, webcam, non-HIDs, etc...).

Naturally, if there are multiple devices with these Usages, the system will be responsible for deciding how to broker and maintain state across devices (e.g. similar to a Keyboard’s CapsLock LED).

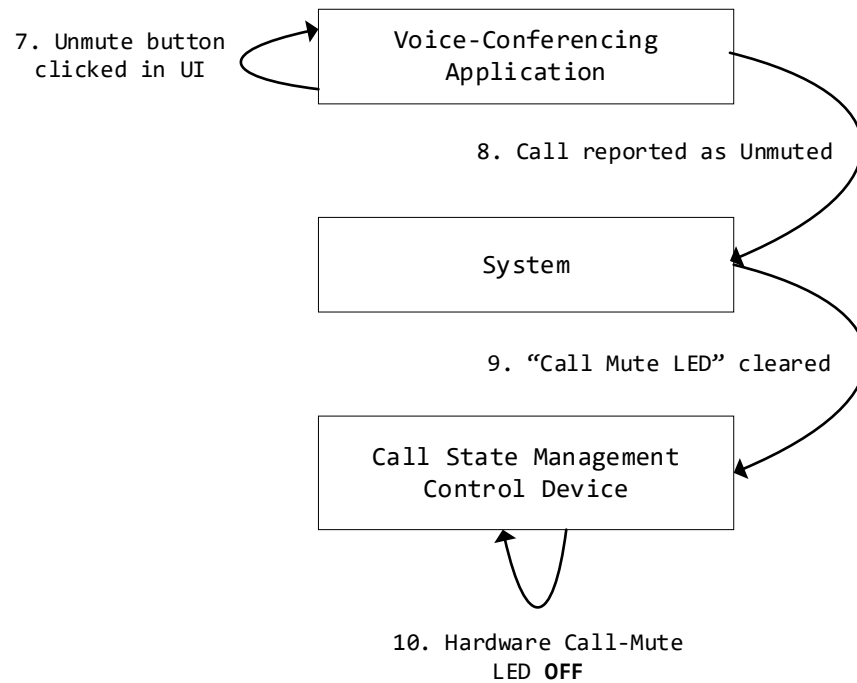
Example 1: User mutes an ongoing call (from device).

1. User actuates the hardware Call-Mute button to mute their channel.
2. "Call State Management Control" device asserts the "Call Mute Toggle" usage.
3. System informs the voice-conferencing application that it must mute the user's channel on the call.
4. Voice-conferencing application applies the channel mute and reports the new mute state to the System.
5. System asserts the "Call Mute LED" control on the "Call State Management Control" device.
6. Hardware Call-Mute LED illuminates, indicating to the User that their channel is muted.



Example 2: User unmutes an ongoing 'call' from a voice-conferencing application UI.

7. User clicks on the voice-conferencing application's unmute button to unmute their channel.
8. Voice-conferencing application removes the channel mute and reports the new mute state to the System.
9. System clears the "Call Mute LED" control on the "Call State Management Control" device.
10. Hardware Call-Mute LED darkens, indicating to the User that their channel is unmuted.



Usage Name	Usage Type	Usage Description
Call State Management Control	CA	System controls for managing the state of the (single) system 'call'.
Call Active LED	OOC	<p>Asserted by System to indicate there is a currently active system 'call'.</p> <p>Level Triggered.</p> <p><i>Note: The device must never depend on this value for internal state management other than LED state (i.e. display only). Do NOT tie this state to "Call Mute Toggle" or any other device state (e.g. audio). Doing so will create issues with system management of the device and system 'call' mute state.</i></p>
Call Mute Toggle	OSC	<p>Toggles the system's 'call' mute state, indicating to the system it must change its current 'call' mute state. (e.g. if system 'call' mute state is unmute, toggle it to mute, and vice-versa).</p> <p>Asserting this Usage does not indicate to turn-off/disable/mute system microphones (or-vice-versa). Rather, assertion indicates to the system that audio input should/should-not be rendered on the active 'call'. (e.g. background noise-cancelling algorithms in voice-conferencing applications can always access and process audio input.)</p> <p><i>Note: The device cannot derive the current system 'call' mute state from the lack-of or previous assertion of this Usage or any other Usage. (as system 'call' mute state may have been set by a user via system/app UI, or another device).</i></p>
Call Mute LED	OOC	<p>Asserted by System to indicate the system's 'call' mute state is muted.</p> <p>Level Triggered.</p> <p><i>Note: The device must never depend on this value for internal state management other than LED state (i.e. display only). Do NOT tie this state to "Call Mute Toggle" or any other device state (e.g. audio). Doing so will create issues with system management of the device and system 'call' mute state.</i></p>

Sample Report Descriptor:

“Call State Management Control” device with a “Call Mute Toggle” button and LEDs to show mute and active system ‘call’ states.

```
0x05, 0x01, // UsagePage(Generic Desktop[1])
0x09, 0x13, // UsageId(Call State Management Control[19])
0xA1, 0x01, // Collection(Application)
0x85, 0x01, // ReportId(1)
0x09, 0xE1, // UsageId(Call Mute Toggle[225])
0x15, 0x00, // LogicalMinimum(0)
0x25, 0x01, // LogicalMaximum(1)
0x95, 0x01, // ReportCount(1)
0x75, 0x01, // ReportSize(1)
0x81, 0x06, // Input(Data, Variable, Relative, NoWrap, Linear, PreferredState)
0x75, 0x07, // ReportSize(7)
0x81, 0x03, // Input(Constant, Variable, Absolute)
0x09, 0xE0, // UsageId(Call Active LED[224])
0x75, 0x01, // ReportSize(1)
0x91, 0x22, // Output(Data, Variable, Absolute, NoWrap, Linear, NoPreferredState)
0x09, 0xE2, // UsageId(Call Mute LED[226])
0x91, 0x22, // Output(Data, Variable, Absolute, NoWrap, Linear, NoPreferredState)
0x75, 0x06, // ReportSize(6)
0x91, 0x03, // Output(Constant, Variable, Absolute)
0xC0, // EndCollection()
```