

Technical Disclosure Commons

Defensive Publications Series

September 06, 2016

AUTOMATIC AUDIOON MEDIA CONTENT STREAMING

Richard Rapp

Justin Lewis

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Rapp, Richard and Lewis, Justin, "AUTOMATIC AUDIOON MEDIA CONTENT STREAMING", Technical Disclosure Commons, (September 06, 2016)
http://www.tdcommons.org/dpubs_series/271



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

AUTOMATIC AUDIO-ON MEDIA CONTENT STREAMING

ABSTRACT

Disclosed herein is a method for automatically transitioning a video media content stream to an alternate stream that is audio-only when the streaming device connects to an audio-only playback mechanism. The method manifests as a feature within a video sharing application. The application is provided with a setting that automatically transitions the video content stream to an audio representation of the same stream when the client device connects via Bluetooth, Wi-Fi or headphone jack to a speaker device. Similarly, when the audio-only speaker device is disconnected from the client device the application resumes streaming the video stream. The method automatically enables the audio-only mode when the user is in the appropriate context (e.g., when driving). The method is also useful for other applications that are appropriate for audio-only playback including those related to sports, or news applications that work well with audio-only playback.

BACKGROUND

Video-sharing applications have developed into the world's largest music streaming platforms. Roughly half of the applications' watch time is for music and music videos. However, since the primary purpose was to share video content, such applications have always been built as a video streaming service. This means that all music content on the application has traditionally been served in video format, either as a music video or as an audio track placed onto a video that only consists of still images such as album art. The music industry has been moving more and more towards music videos. Music videos are now the most popular way to share music, which is extremely common on social platforms. This makes the video component extremely valuable to the industry and also makes it very likely

that competitors will soon begin to build video streaming functionality into their music streaming applications to allow for the playback of video.

Streaming components with added video overhead is not a major issue when it comes to desktop web, but poses an issue on mobile devices due to their limited bandwidth and data caps. In cases where the user wants to watch the video, streaming the video is clearly required. But if the user is only interested in the audio component, these video sharing applications become an inefficient choice for them, as they stream the video as well, wasting bandwidth, burning battery, and with slow connections, increasing buffering events, all of which result in a poorer user experience. Such issues are accentuated in emerging markets where bandwidth is even more limited, and devices tend to be less powerful. This has even lead to users pirating the audio from videos just so they can listen to music without wasting bandwidth.

Of late, the video sharing applications have provided features that allow for both audio-only streaming and audio playback in the background with the application minimized. Both of these features are very useful to users who want to listen to music without the accompanying video and the drawbacks that come along with it. For example, music streaming on mobile devices while driving is common and users are very aware of the wasted bandwidth, data usage, and negative effects on battery that come from streaming video in such cases. Additionally, video playback could be distracting for drivers. In the past, though users clearly were in favor of audio-only streaming, this was not possible on the application as all advertising required a visual component. Such issues have recently been removed with options for subscription-based ad-free experiences that are more in-line with other music streaming platforms. Even with the ability to change to audio-only streaming for mobile applications, users still may not be aware of how to use the feature, or even if they are, they

may be unable to use it in cases where they are the driver and are unable to interact with their device to change a setting.

DESCRIPTION

This disclosure presents a system and method for automatically transitioning a video media content stream to an alternate stream that is audio-only when the streaming device connects to an audio-only playback mechanism. This method manifests as a feature within the video sharing application. The system includes an application provided with a setting that automatically transitions the video content stream to an audio representation of the same stream when the client device connects via Bluetooth, wifi or headphone jack to a speaker device. Similarly, when the audio-only speaker device is disconnected from the client device the application resumes streaming the video stream.

In video sharing applications, the audio stream is not simply a side channel of the video stream. Each video can have several different audio streams which correspond to the language that the user is currently using to browse the application. Every time a video is played a playback configuration is sent to the client device. This playback configuration includes a reference link to the video stream, and also includes an alternate reference to an audio stream associated with this video and the corresponding user language setting.

Video sharing applications integrated with this process listen for two signals from the mobile operating system, a Bluetooth connection establishment event or a headset plug insert event. The application then uses two local variables to determine if the transition to an audio-only playback state should be automatic. The first of these two variables is a user setting which can be controlled by the user via a check box. The second setting “return_to_video” remembers the ephemeral context of the application so that the application does not transition back into video only mode when the user does not expect it. By default

this variable is set as false, and is reset to false every time the user explicitly toggles the audio-only setting. When either of these signals is received, the application determines which playback state (video or audio-only) the device is currently in. If the application is in video playback state then the “return_to_video” variable is set to true otherwise it is set to false. If the application is in video playback mode then it will transition to an audio playback mode.

Transitioning to audio playback mode involves not only changing a user state but also smoothly changing the streaming source. Meaning that the method downloads the first several seconds of the audio stream then kicks up the audio stream while simultaneously muting the video stream. Thus, even before the audio stream is buffered the video component of the video player is hidden to give the appearance that the device has already transitioned to an audio-only mode. However the audio stream and the video stream will remain synchronized (or included as an offset) so that they continue to remain in sync when the audio is played in-line with the video. When Bluetooth is disconnected or the application senses the headset jack being unplugged then the application consults the aforementioned two variables to determine if the application should return to video playback mode. If the user setting for automatic transitions is enabled and the “return_to_video” is set as true then the application will return the client device to a video playback state when the speaker device is disconnected. The method will typically be accompanied by a user interface notification that explains why the application transitioned to an audio-only mode.

The method automatically enables the audio-only mode when the user is in the appropriate context. The method is also useful for other applications that are appropriate for audio-only playback including those related to sports, news applications which work well with audio-only playback.