

Audio/video tracks identification

Since build [2.0.186](#) there is the examples how to identify a published audio/video tracks.

On the screenshot below, the track names `mic1` for audio and `cam1` for video tracks are displayed in SFU Player example

SFU Player

Server url

wss://test1.flashphoner.com:8443

Room name

ROOM1

Player

Player1-4f27

Stop

ESTABLISHED

Published by: Publisher1-1c49

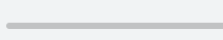

320x180

cam1

360p send 720p send 180p send



Mute mic1

|| 2:41  

Analyzing example code

Let's review SFU SDK examples source code available on [GitHub](#) with tag `8287dd9`.

1. Track types setting in `config.json` file

audio track type code

```
{
  ...,
  "media": {
    "audio": {
      "tracks": [{
        "source": "mic",
        "channels": 2,
        "type": "mic1"
      }]
    },
    ...
  }
}
```

video track type code

```
{
  ...,
  "media": {
    ...
    "video": {
      "tracks": [
        {
          "source": "camera",
          "width": 1280,
          "height": 720,
          "codec": "H264",
          ...,
          "type": "cam1"
        }
      ]
    }
  }
}
```

2. Adding track types to WebRTC configuration object before publishing

`Room.join` code

```
let streams = await getVideoStreams(mainConfig);
let audioStreams = await getAudioStreams(mainConfig);
if (state.isConnected() && state.isActive()) {
  //combine local video streams with audio streams
  streams.push.apply(streams, audioStreams);
  let config = {};
  //add our local streams to the room (to PeerConnection)
  streams.forEach(function (s) {
    let contentType = s.type || s.source;
    //add local stream to local display
    localDisplay.add(s.stream.id, $("#" + state.inputId()).val(),
s.stream, contentType);
    //add each track to PeerConnection
```

```

    s.stream.getTracks().forEach((track) => {
      config[track.id] = contentType;
      addTrackToPeerConnection(state.pc, s.stream, track, s.encodings);
      subscribeTrackToEndedEvent(state.room, track, state.pc);
    });
  });
  //start WebRTC negotiation
  state.waitFor(state.room.join(state.pc, null, config), MAX_AWAIT_MS);
}

```

3. Receiving `ADD_TRACKS` event at player side

`SFU_ROOM_EVENT.ADD_TRACKS`, `setTrackInfo()` code

```

room.on(constants.SFU_ROOM_EVENT.ADD_TRACKS, function(e) {
  console.log("Received ADD_TRACKS");
  ...
  for (const pTrack of e.info.info) {
    let createDisplay = true;
    for (let i = 0; i < participant.displays.length; i++) {
      let display = participant.displays[i];
      if (pTrack.type === "VIDEO") {
        if (display.hasVideo()) {
          continue;
        }
        display.videoMid = pTrack.mid;
        display.setTrackInfo(pTrack);
        createDisplay = false;
        break;
      } else if (pTrack.type === "AUDIO") {
        if (display.hasAudio()) {
          continue;
        }
        display.audioMid = pTrack.mid;
        display.setTrackInfo(pTrack);
        createDisplay = false;
        break;
      }
    }
    if (!createDisplay) {
      continue;
    }
    let display = createRemoteDisplay(participant.nickName,
    participant.nickName, mainDiv, displayOptions);
    participant.displays.push(display);
    if (pTrack.type === "VIDEO") {
      display.videoMid = pTrack.mid;
      display.setTrackInfo(pTrack);
    } else if (pTrack.type === "AUDIO") {
      display.audioMid = pTrack.mid;
      display.setTrackInfo(pTrack);
    }
  }
})...;

```

4. Track identifier displaying

`AddRemoveTracks.info.info.contentType`, `setTrackInfo()` code

```
setTrackInfo: function(trackInfo) {
  if (trackInfo) {
    ...
    if (trackInfo.type) {
      contentType = trackInfo.contentType || "";
      if (trackInfo.type == "VIDEO" && displayOptions.type &&
contentType != "") {
        showItem(videoTypeDisplay);
        videoTypeDisplay.innerHTML = contentType;
      }
      if (trackInfo.type == "AUDIO") {
        audioStateButton.setContentType(contentType);
      }
    }
  }
},
...
```