/\*\*

 \* Бета-версия. Макрос исправляет глюк: cбрасывает нежелательную презентацию во время старта H323 звонка, в случае, когда имеет место Interworking.

 \* Автор: Юзов Евгений Борисович, ООО "Центр", 12.01.2022.

 \*/

import xapi from 'xapi';

const DEBUG = true;

const timeout = 10; // Количество секунд работы макроса после старта звонка, сравнивается с Call.Duration. Полезную презентацию можно подать после timeout секунд!

const index = 1; // Макс. значение счётчика, чтобы не попасть в бесконечный цикл.

const sleepTime = 2000; // sleepTime миллисекунд паузы после события изменения компоновки (ActiveLayout) и перед сбросом презентации.

/\*\*

 \* Держит паузу заданное количество миллисекунд.

 \*/

const sleep = (timeout) => new Promise((resolve) => {

  setTimeout(resolve, timeout);

});

/\*\*

 \* Старт локальной презентации, источник - видео с камеры. Потом дожидаемся, пока презентация установится, и сразу же её сбрасываем.

 \*/

async function StartStopPres() {

  await xapi.Command.Presentation.Start({Instance: '6', PresentationSource: '1', SendingMode: 'LocalRemote'})

  .then(await xapi.Command.Presentation.Stop({Instance: '6', PresentationSource: '1'}))

  .catch(console.log);

  if (DEBUG) {console.log ('Сработал сброс презентации!')}

}

/\*\*

 \* Срабатывает только при количестве активных звонков 1, и только в течение timeout секунд после установления звонка.

 \* Потом в течение всего звонка сбрасывать презентации не нужно, т.к. они несут полезную нагрузку!

 \* Если на кодеке терминируется несколько звонков, сброс презентации тоже не работает.

 \*/

async function CorrectScreen() {

  Promise.all([

  xapi.Status.SystemUnit.State.NumberOfActiveCalls.get(),

  xapi.Status.Call.Duration.get(),

  xapi.Status.Call.RemoteNumber.get(),

  xapi.Status.Video.Layout.CurrentLayouts.ActiveLayout.get(),

  xapi.Status.Call.Direction.get()

  ])

  .then(async([callCount, callDuration, remoteNumber, activeLayout, callDirection]) => {

    let i = 0; // Счётчик повторных попыток

    // Переменная для доп. условия: при вх. звонке - на номер звонящего (т.е. откуда звонят), например, iwf@10.100.40.11;

    // при исходящем звонке - на номер назначения (куда звонят), например, 217.198.12.14@ip.

    let numCondition = -1;

    // Поиск нужной подстроки в номере звонящего.

    const pos = [remoteNumber.indexOf('@10.100.40.38'), remoteNumber.indexOf('@cms.taif.ru'), remoteNumber.indexOf('@10.100.40.11'),

    remoteNumber.indexOf('@10.100.40.12'), remoteNumber.indexOf('@cucm.taif.ru'), remoteNumber.indexOf('@taif.loc'),

    remoteNumber.indexOf('@taif.ru'), remoteNumber.indexOf('iwf@'), remoteNumber.indexOf('@ip')];

    if (DEBUG) {

      console.log('callDirection: ' + callDirection);

      console.log('callDuration: ' + callDuration);

      console.log('callCount: ' + callCount);

      console.log('remoteNumber: ' + remoteNumber);

      console.log('activeLayout: ' + activeLayout);

      // console.log('remoteNumber @10.100.40.38 domain position is: ' + pos[0]);

      // console.log('remoteNumber @cms.taif.ru domain position is: ' + pos[1]);

      // console.log('remoteNumber @10.100.40.11 domain position is: ' + pos[2]);

      // console.log('remoteNumber @10.100.40.12 domain position is: ' + pos[3]);

      // console.log('remoteNumber @cucm.taif.ru domain position is: ' + pos[4]);

      // console.log('remoteNumber @taif.loc domain position is: ' + pos[5]);

      // console.log('remoteNumber @taif.ru domain position is: ' + pos[6]);

      console.log('remoteNumber iwf@ uri position is: ' + pos[7]);

      console.log('remoteNumber @ip domain position is: ' + pos[8]);

    }

    if (callDirection == 'Incoming') {numCondition = pos[7]}

    if (callDirection == 'Outgoing') {numCondition = pos[8]} // т.е. сброс срабатывает только при исходящем звонке на  ip-адрес, т.е. на номер, где присутствует @ip!

    if (callCount == 1 && callDuration <= timeout && numCondition !== -1) {

      // While здесь потому, что иногда с первого раза не срабатывает. Программа ждёт ещё sleepTime секунд и повторяет сброс презентации.

      while (i <= index) {

        await sleep(sleepTime); // Ждём sleepTime секунд после события изменения компоновки (ActiveLayout), т.к. сразу может не сработать.

        await xapi.Status.Video.Layout.CurrentLayouts.ActiveLayout.get()

        .then ((activeLayout) => {if (activeLayout !== '') {StartStopPres();}})

        i++;

      }

    }

  })

  .catch(console.log);

}

/\*\*

 \* Отличие H323 звонка от SIP звонка, с точки зрения кодека, в том, что при H323 звонке сразу появляется нежелательная презентация.

 \* Следовательно, меняется/появляется статус компоновки (ActiveLayout) этой презентации.

 \*/

xapi.status.on('Video Layout CurrentLayouts ActiveLayout', CorrectScreen);