



DX80 受付端末のつくり方

シスコシステムズ合同会社

岩岸 優希 <http://cs.co/yiwagish>

2018/8/29

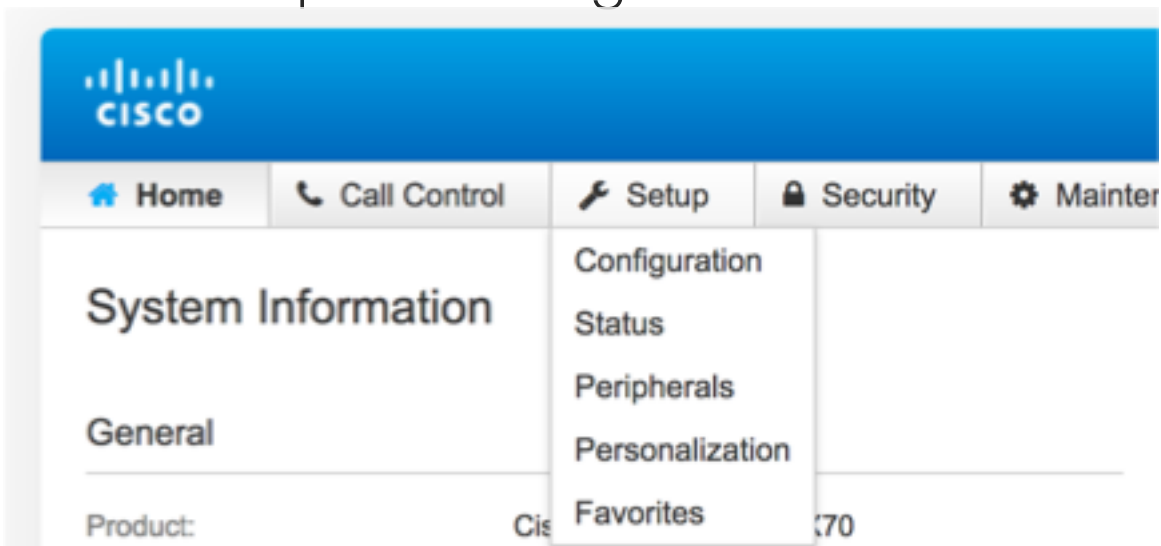
受付端末



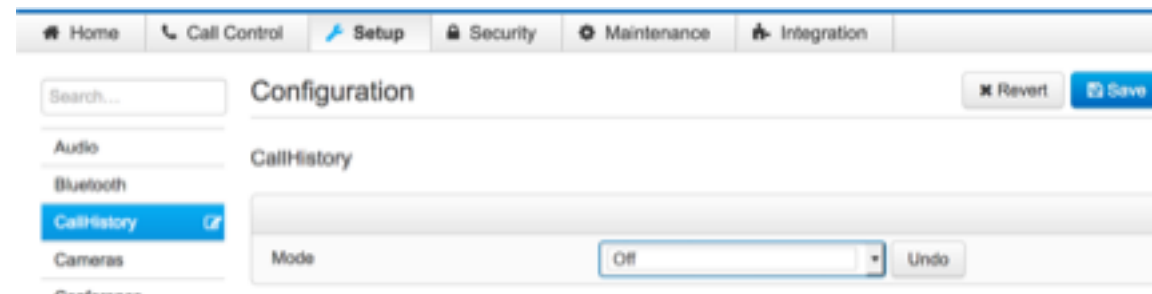
- ・通常は HDMI InにPC/サイネージプレイヤーを接続しサイネージ再生
- ・ご用の方は右上にタッチしてくださいと表示
- ・タップすると「発信」「受付」を表示
- ・「発信」からはフォルダ構造の電話帳表示
- ・「受付」は直接受付担当に発信

発信履歴を残さない設定

- Setup > Configuration



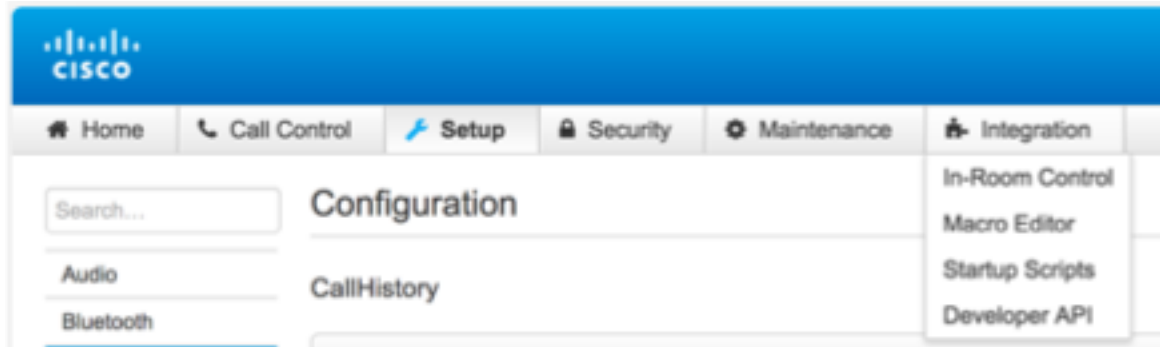
- Call History > Mode: Off



以前の来訪者の発信履歴を残さないようにする

受付ボタン設置

- Integration > In-Room Control



- Launch Editor をクリック

In-Room Control

You can customize our user interface to allow control of peripherals in a meeting room, for example lights and blinds. This allows for the powerful combination of an external control system's functionality and the Cisco user interface.

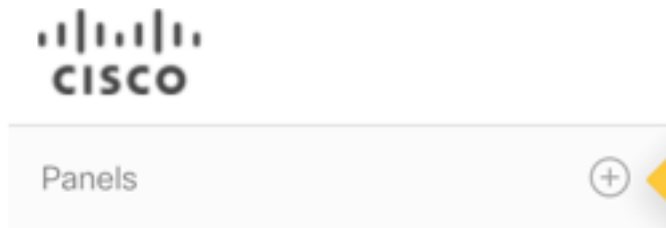
Create/Edit Interfaces

Launch Editor

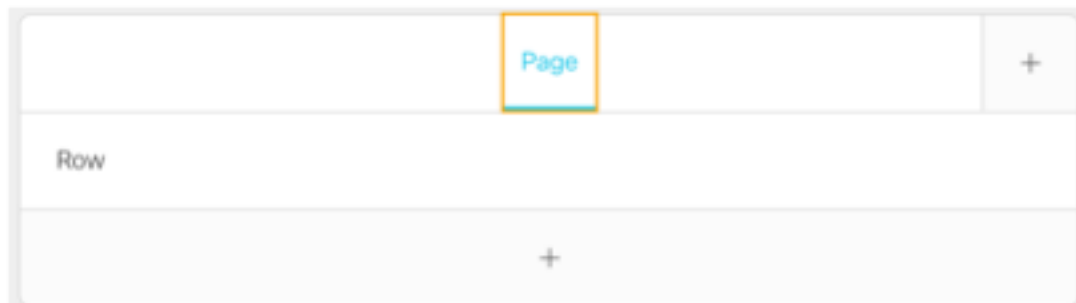
Launch the editor to create custom user interface panels. Export them to the video system to make them appear in the user interface, or save the designs in local files for later.

アイコンだけ残す設定

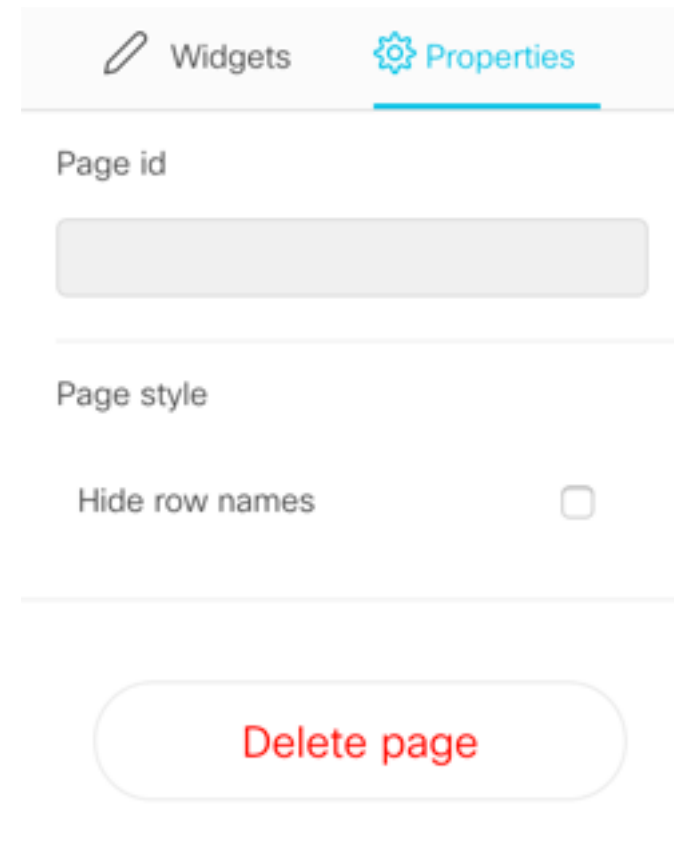
- Panesをクリック



- Page をクリック



- Delete Page をクリック

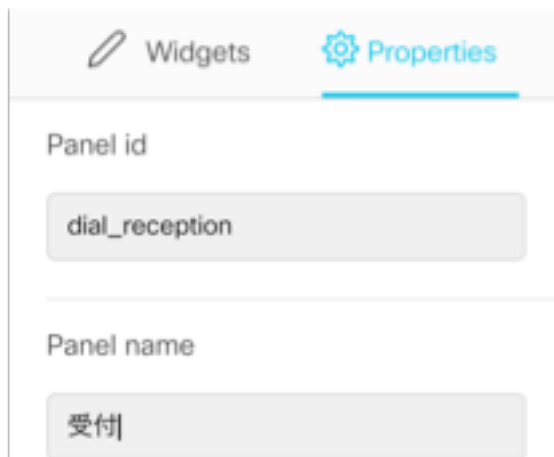


受付アイコンの設定

- Panel をクリック

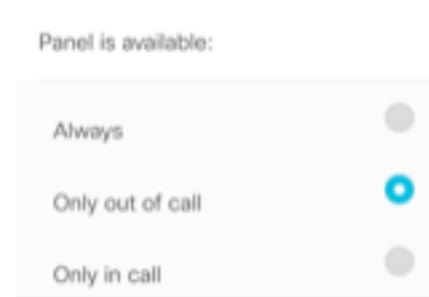


- Properties を選択
 - Panel id: dial_reception
 - Panel name: 受付

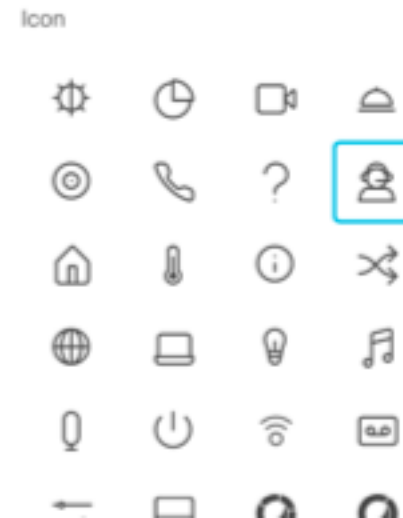


Panel idはタップした際にイベントが発生しマクロに渡されます

- Panel is available
 - Only out of call を選択



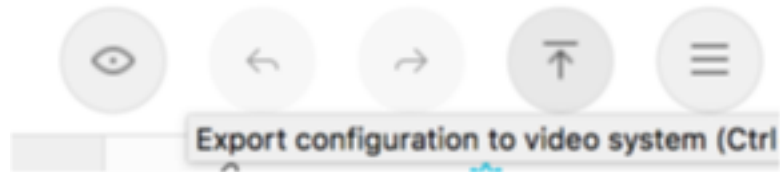
- Icon : 人型のものを選択



この他に色を選択可能です

DX80への反映

- Export アイコンをクリックし、DX80に反映

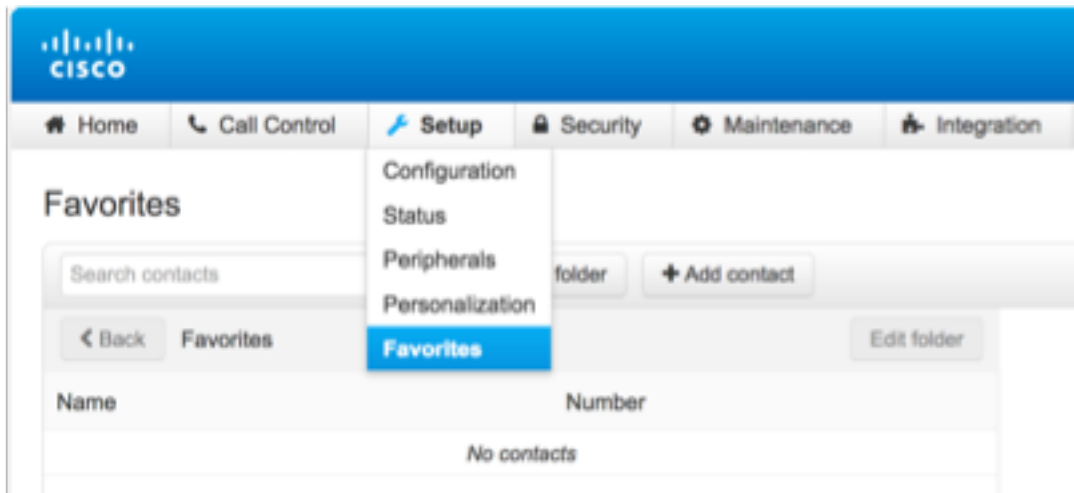


- DX80 発信の横に「受付」が表示
- 受付をクリックし、何も起きないことを確認



部署代表電話帳作成 - フォルダ作成

- Setup>Favorites



- Add folderを選択

Favorites

- Folder name: フォルダ名を入力

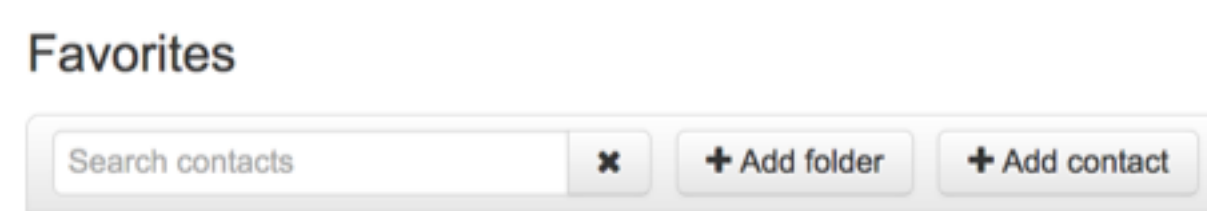
Folder name

Parent name

最後に"Save"
フォルダを多段にすることも可能

部署代表電話帳作成 – フォルダ作成

- Add contactを選択



- Name: 部署名
- Folder: 親フォルダを選択
- Number: 電話番号



最後に”Save”

部署代表電話帳表示例

- 部署

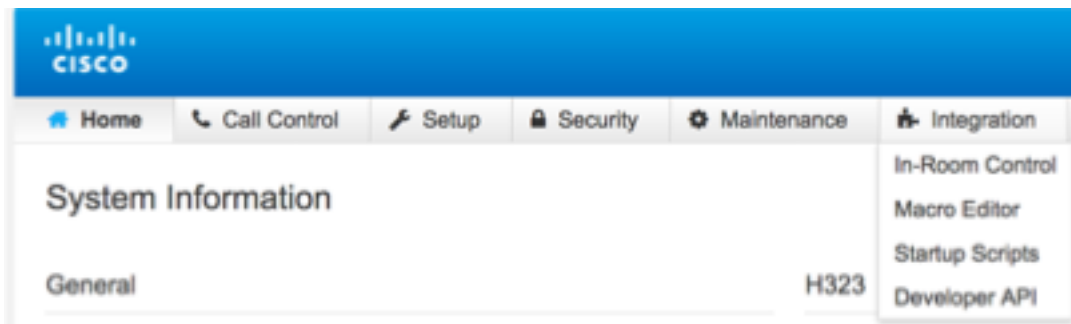


- 代表番号選択

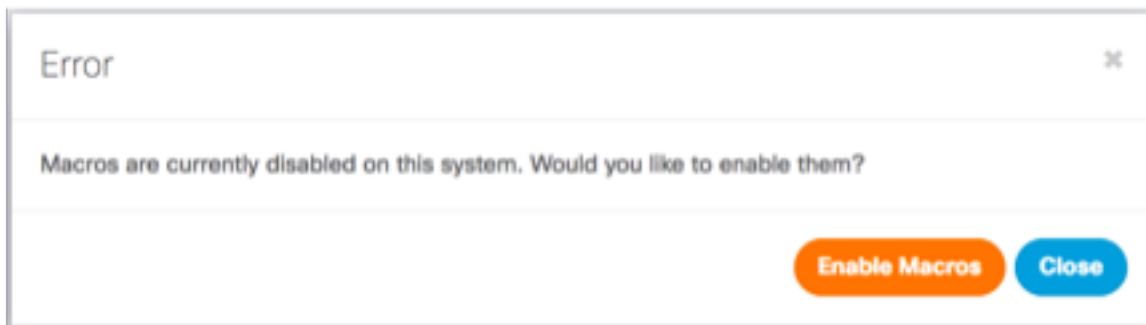


受付発信およびインストラクション表示マクロ

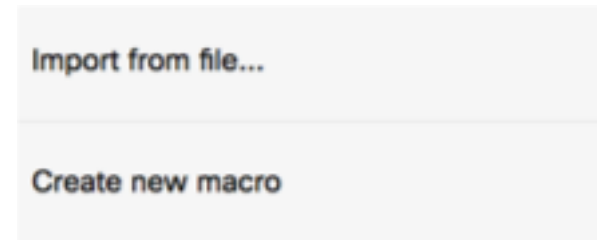
- Integration > Macro Editor



- Enable Macros をクリック



- Create new macro をクリック



- "Untitled1*" をクリックしマクロ名を"Reception"に変更



Macro- 受付発信処理

```
/**
 * A small Direct Call from ICON
 */
const xapi = require('xapi');

// These match the widget ids of the In-Room control buttons

const numbers = {
  dial_reception: '1000',
};

function dial(number) {
  console.log('dial', number);
  xapi.command('dial', { Number: number });
}

function listenToGui() {
```

```
xapi.event.on('UserInterface Extensions Panel Clicked', (event) => {
  const number = numbers[event.PaneId];
  if (number) dial(number);
});
}

listenToGui();
```

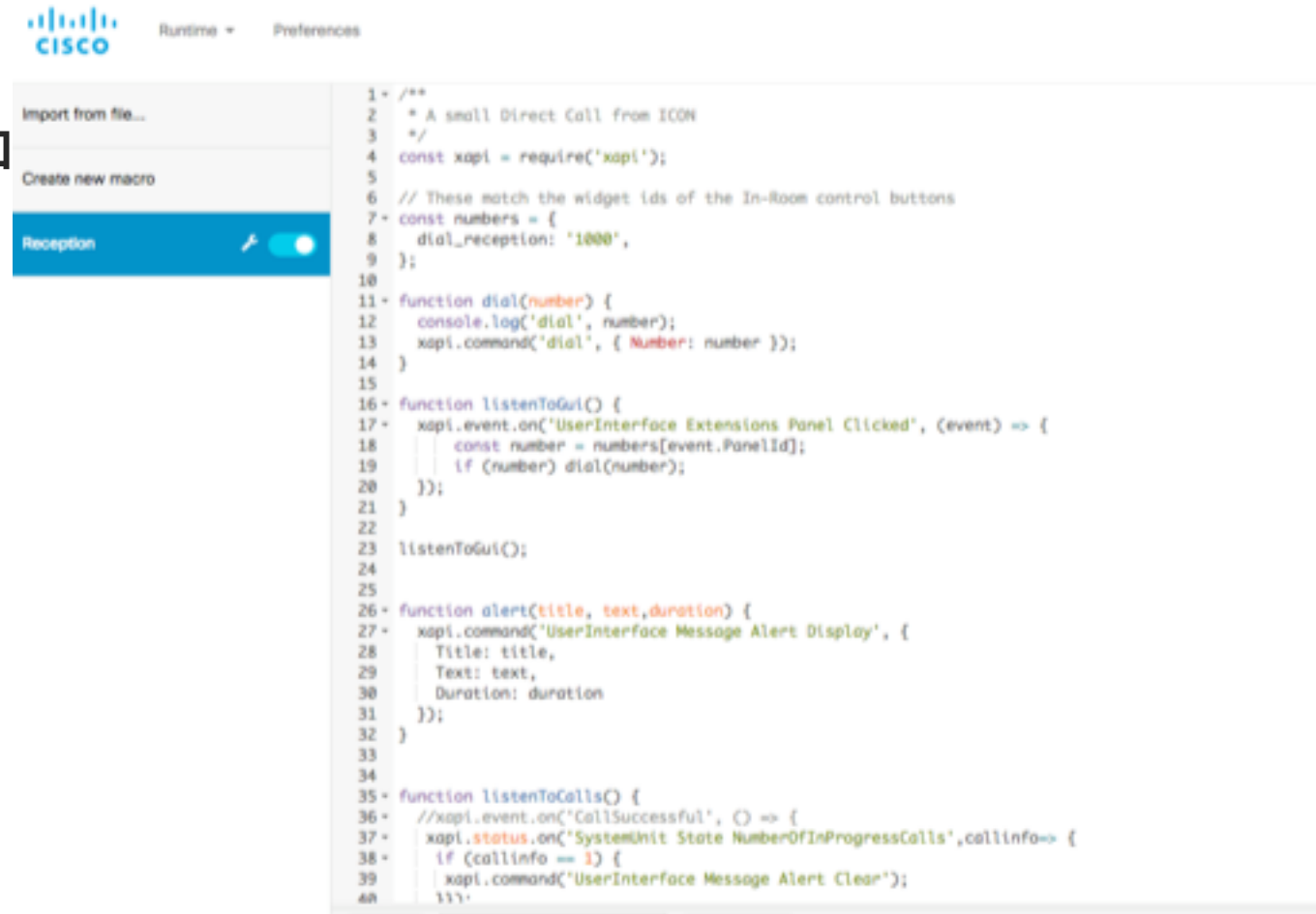
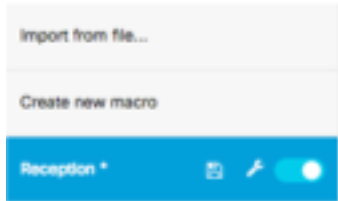
Macro- インストラクション表示

```
function alert(title, text,duration) {  
  
xapi.command('UserInterface Message Alert Display', {  
  
Title: title,  
  
Text: text,  
  
Duration: duration  
  
});  
  
}
```

```
function listenToCalls() {  
  
xapi.status.on('SystemUnit State NumberOfInProgressCalls',callinfo=> {  
  
if (callinfo == 1) {  
  
xapi.command('UserInterface Message Alert Clear');  
  
}});  
  
xapi.event.on('CallDisconnect', () => alert("画面をタッチし、受付を選択するか発信を選んで  
部署代表を選択してください","Please Touch screen.",0));  
  
}  
  
listenToCalls();
```

マクロの保存と有効化

- Save ボタンを押し、保存
- Enable/Disableをオンにし、マクロを有効化

A screenshot of the Cisco XAPI Runtime Preferences window. The window title is 'Runtime - Preferences' and it features the Cisco logo. The interface shows a list of macros on the left, with 'Reception' selected. The main area displays the JavaScript code for this macro. The code includes comments and functions for dialing a number, listening to GUI events, and displaying alerts.

```
1 * /**
2  * A small Direct Call from ICON
3  */
4 const xapi = require('xapi');
5
6 // These match the widget ids of the In-Room control buttons
7 * const numbers = {
8   dial_reception: '1000',
9 };
10
11 * function dial(number) {
12   console.log('dial', number);
13   xapi.command('dial', { Number: number });
14 }
15
16 * function listenToGui() {
17 *   xapi.event.on('UserInterface Extensions Panel Clicked', (event) => {
18     const number = numbers[event.PanelId];
19     if (number) dial(number);
20   });
21 }
22
23 listenToGui();
24
25
26 * function alert(title, text, duration) {
27 *   xapi.command('UserInterface Message Alert Display', {
28     Title: title,
29     Text: text,
30     Duration: duration
31   });
32 }
33
34
35 * function listenToCalls() {
36 *   //xapi.event.on('CallSuccessful', () => {
37 *   xapi.status.on('SystemUnit State NumberOfInProgressCalls', callinfo => {
38 *     if (callinfo == 1) {
39 *       xapi.command('UserInterface Message Alert Clear');
40 *     }
41 *   });
42 }
```

