

The background features a vibrant, abstract design. On the left, there are horizontal, wavy bands of color in shades of red, orange, yellow, and green. On the right, a bright white light source emits a series of colorful rays in shades of blue, cyan, and yellow, creating a sunburst effect.

cisco *Live!*

Let's go

#CiscoLiveAPJC



The bridge to possible

Room Customization Journey

From Build to Deploy

Charlie Thorpe – Technical Marketing Engineer
BRKCOL-2199

CISCO *Live!*

#CiscoLiveAPJC

Cisco Webex App

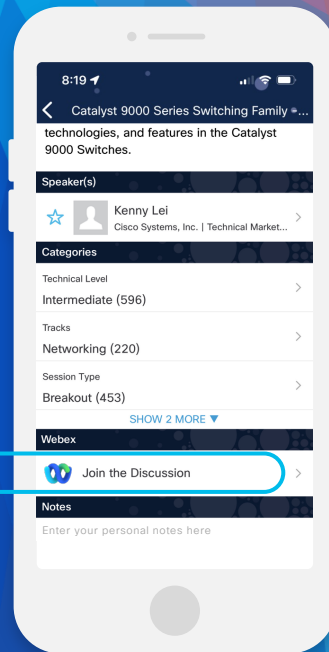
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until December 22, 2023.

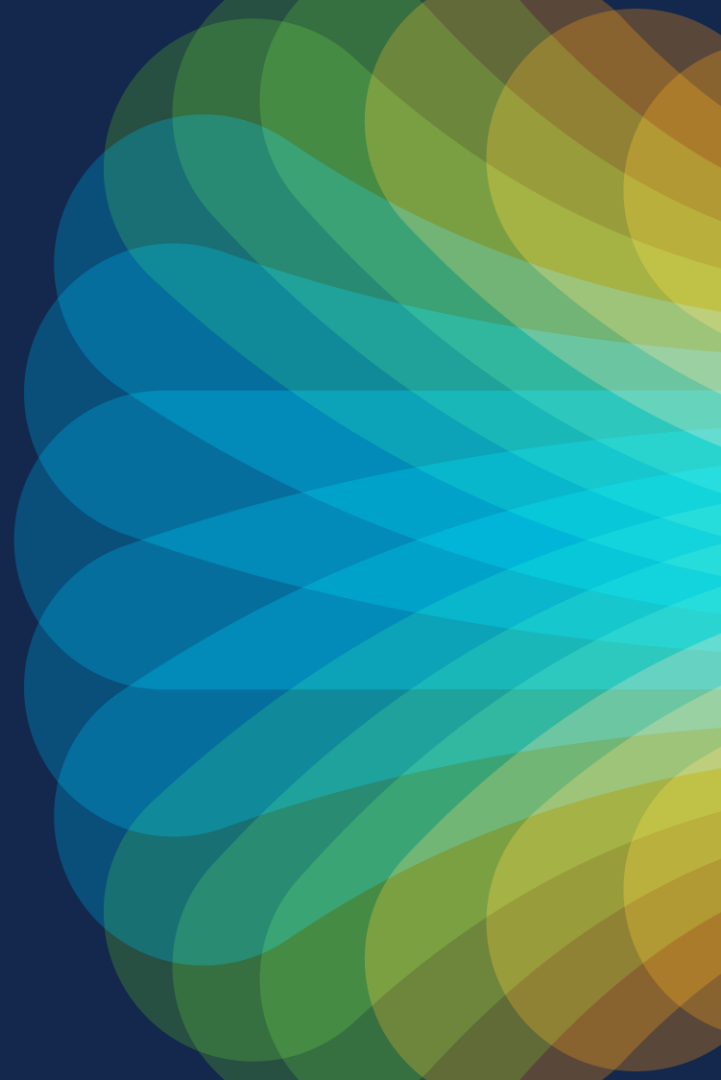


<https://ciscolive.ciscoevents.com/ciscolivebot/#BRKCOL-2199>

Agenda

- Introduction
- Room Devices
- Integration Tools
- Let's Build a Customization!
- Deployment Options
- Automation
- CE-Deploy

Room Devices



Cisco video devices

Desk Series



- All in one Desktop Solution
- Built in Camera, Microphones Speakers
- Personal Spaces, Huddle Rooms, Video Booths, Virtual Rounding

Room Series



- Integration Kits
- Bring your own Display
- Integrated or External Cameras
- Integrated or External Audio

Board Series



- All in one Collaboration Solution
- Built in Camera, Microphones Speakers
- Support for external Cameras, Audio and Displays

Cisco peripherals

Cameras



(A)



(B)

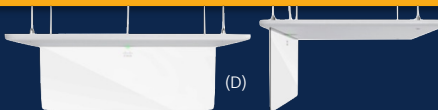
Quadcamera (A)

- Speakertracking, Frames, PeopleCounting, Integrated Speakers (ARC or Line In), HDMI 1080p@60, 5x Digital

PTZ4K (B)

- PresenterTracking, 20x Optical, 12x Digital, HDMI 1080p@60 or 4k@30, POE+

Microphones



(D)



(C)



(E)

Table Mic J/E (C)

- Balanced omnidirectional, boundary layer

Ceiling G2 (D)

- Balanced Directional Microphone

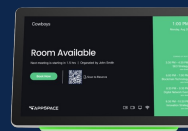
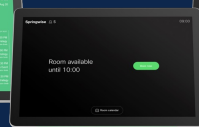
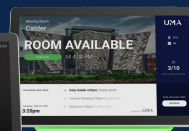
Table Mic Pro (E)

- AVoIP, POE, 4 Channel Directional

Touch Panels



(F)



(G)

Table Navigator (F)

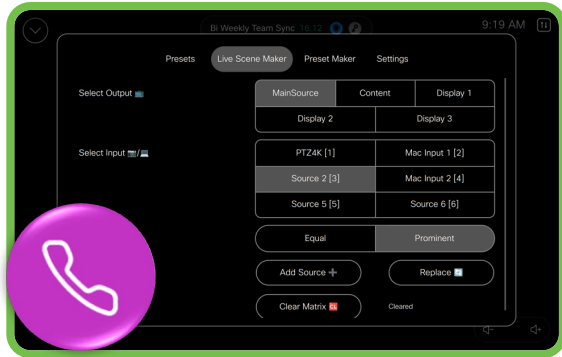
Wall Navigator (G)

- Both support
 - Controller, Booking and PWA Modes and POE

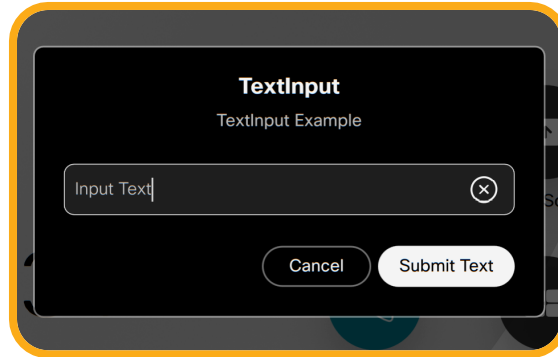


Integration Tools

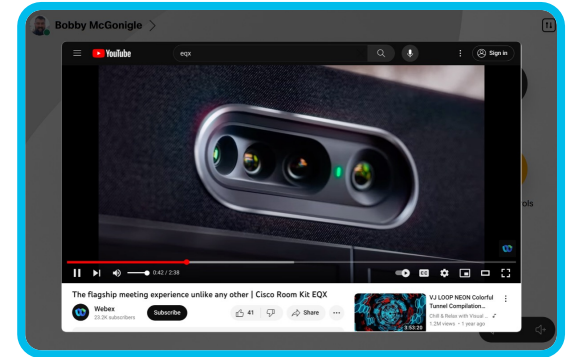
User Interface Elements



Extensions

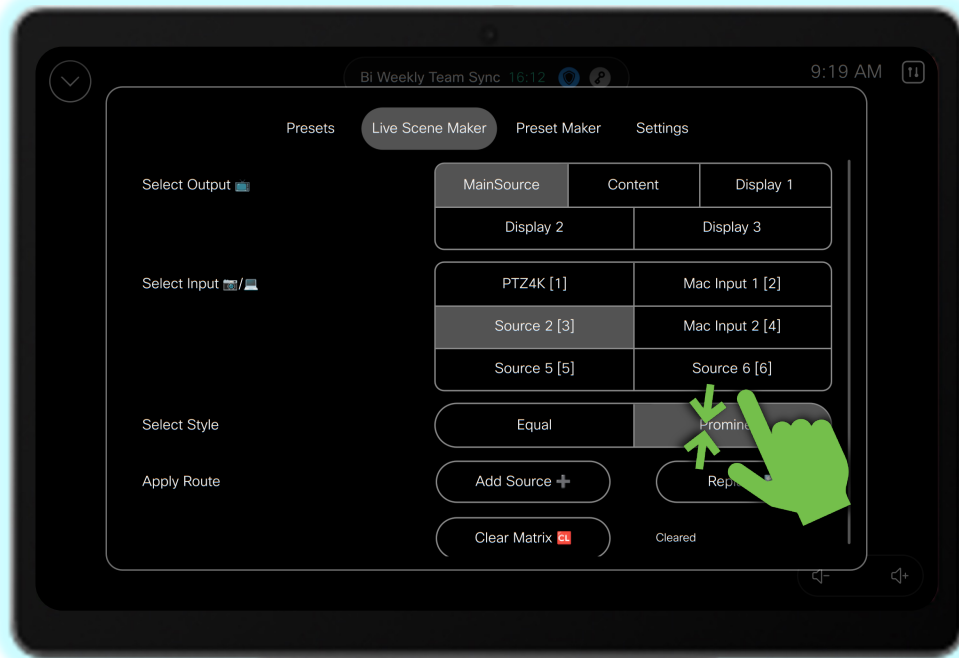


Messages



Web Content

UI Extensions



- UI Extensions produce Events on our xAPI
- Used to build new solutions that can unlock additional functionality or simply workflows

UI Extension Editor

- Built into each device
- Create custom interfaces

An action button usually works in tandem with a custom script from the macro editor to provide powerful features to the user (eg. a "Call for help" button).

Add a new UI Extension



Panel

+ Add

Create a panel with buttons and sliders. These can for example control the lights in the meeting room.



Action Button

+ Add

Create a button that runs a custom script. This is a great way to create quick dials for instance.

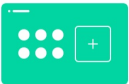


Web App

+ Add

Create a button that opens a web page on the device. Can be used on Webex Boards and Desk Pros to provide interactive web tools.*

*Requires that the Web Engine is enabled.



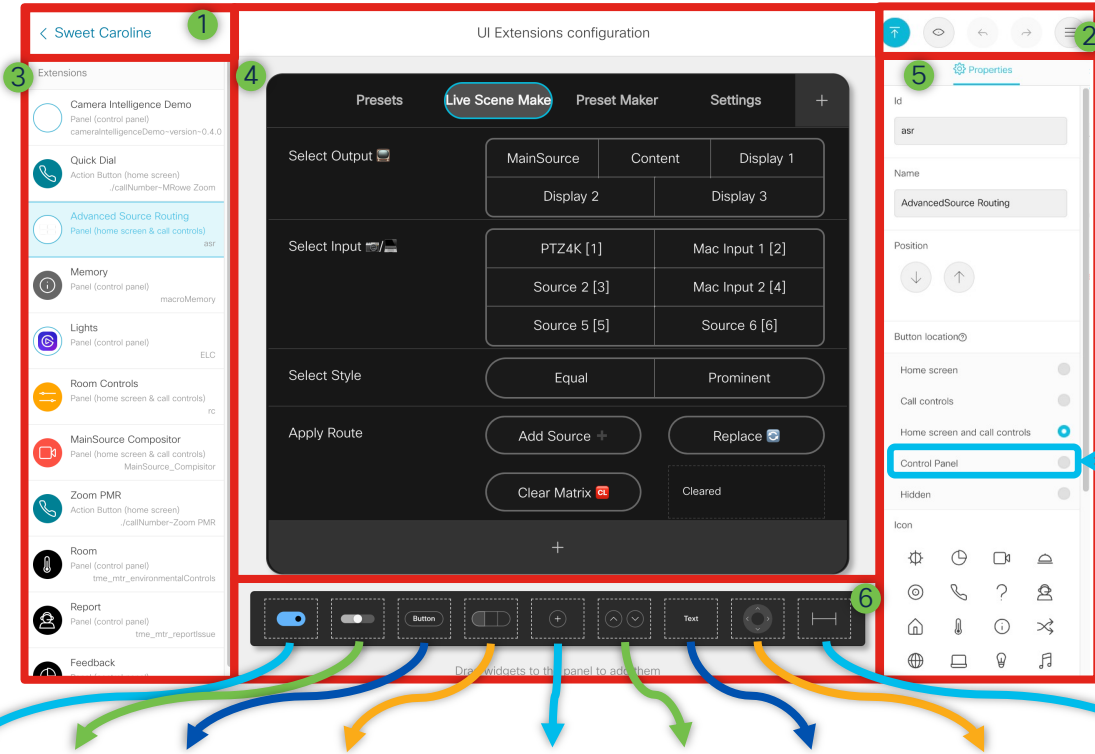
Web Widget

+ Add

Create a widget that shows a web page on the home screen. Can be used to provide information via the web.*

*Requires that the Web Engine is enabled.

UI Extension Editor



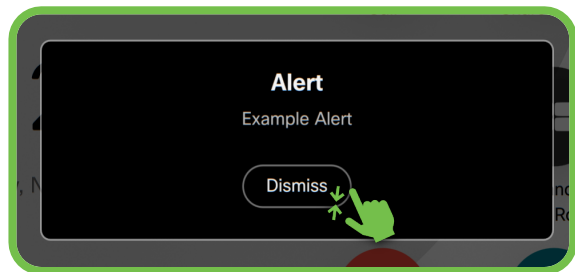
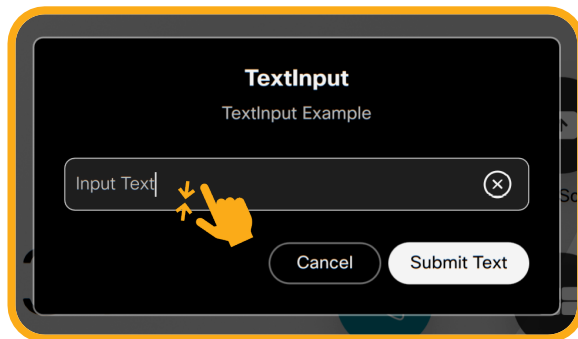
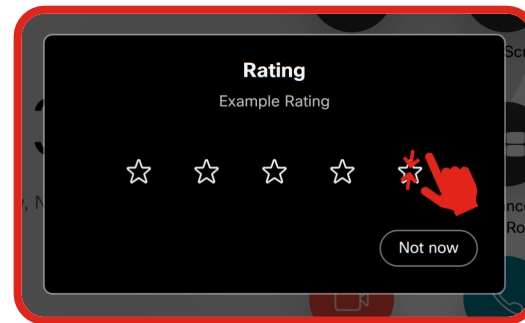
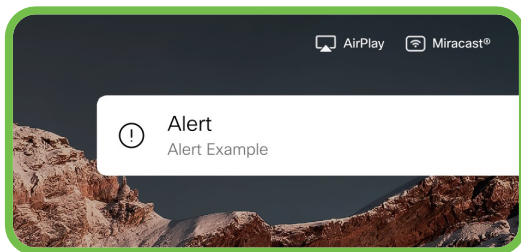
1. Device Home Page
2. Extension Toolbar
 - Upload
 - Preview
 - Undo/Redo
 - Menu
3. Extension Sidebar

4. Editor Canvas
5. Properties Menu
6. Widget Select Menu

This is new!

Toggle || Slider || Button || Group Button || Icon Button || Spinner || Textbox || Directional Pad || Spacer

User Interface Messages



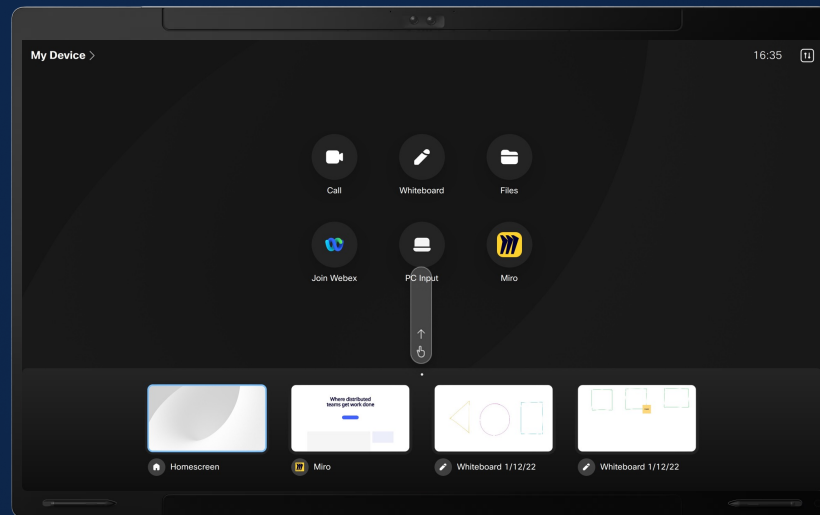
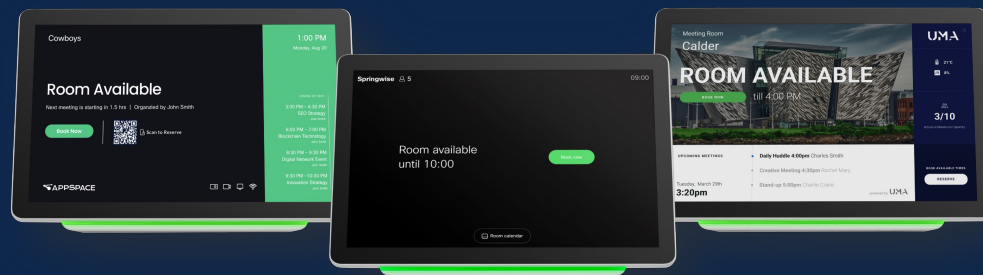
User Interface Messages



User Interface Elements

Web Applications

- Render Web pages on a Touch OSD or Touch Controller
- Great for simple agnostic applications such as Weather, Stocks, Visual Collaboration Apps, booking solutions
- Leveraging our xAPI, can serve as a fully customized front end and control our devices



Integration Methods

Take advantage of our vast xAPI

Macros



- Built into Every Device
- Dedicated Runtime Environment
- ES6 Javascript
- Supports HTTP Control for accessing services/devices on the network

Room Controllers



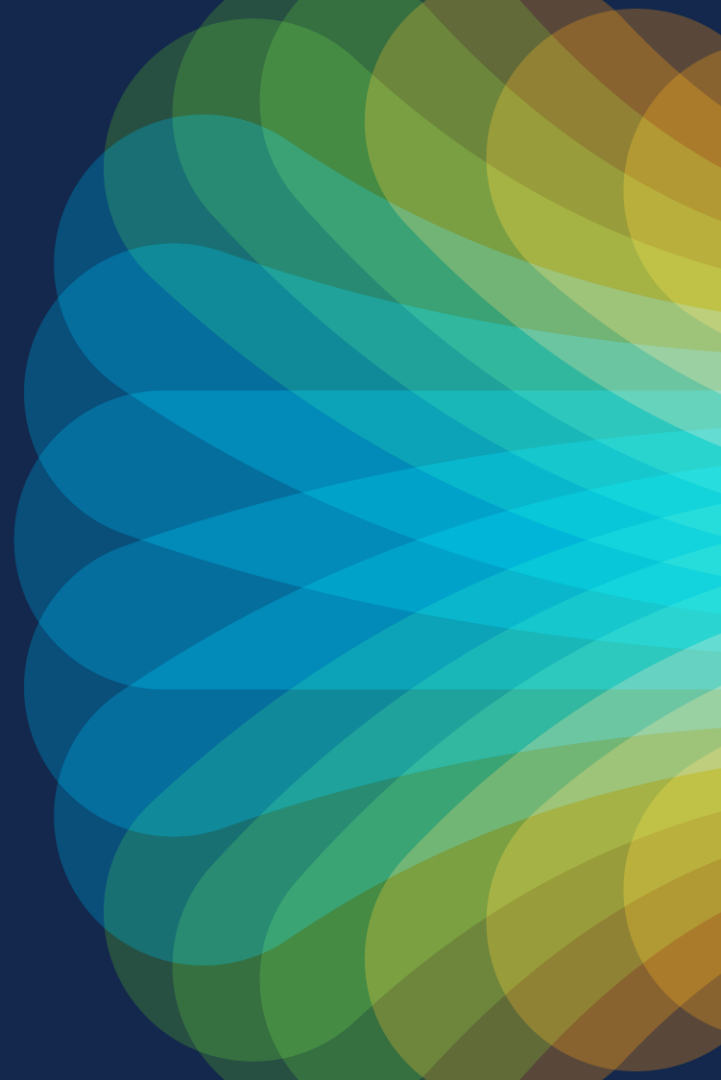
- Can extend a devices access to other integrations
 - Rs232/485, Relays, TCP/UDP, Etc
- Can help coordinate vast Audio Visual arrangements

External Services



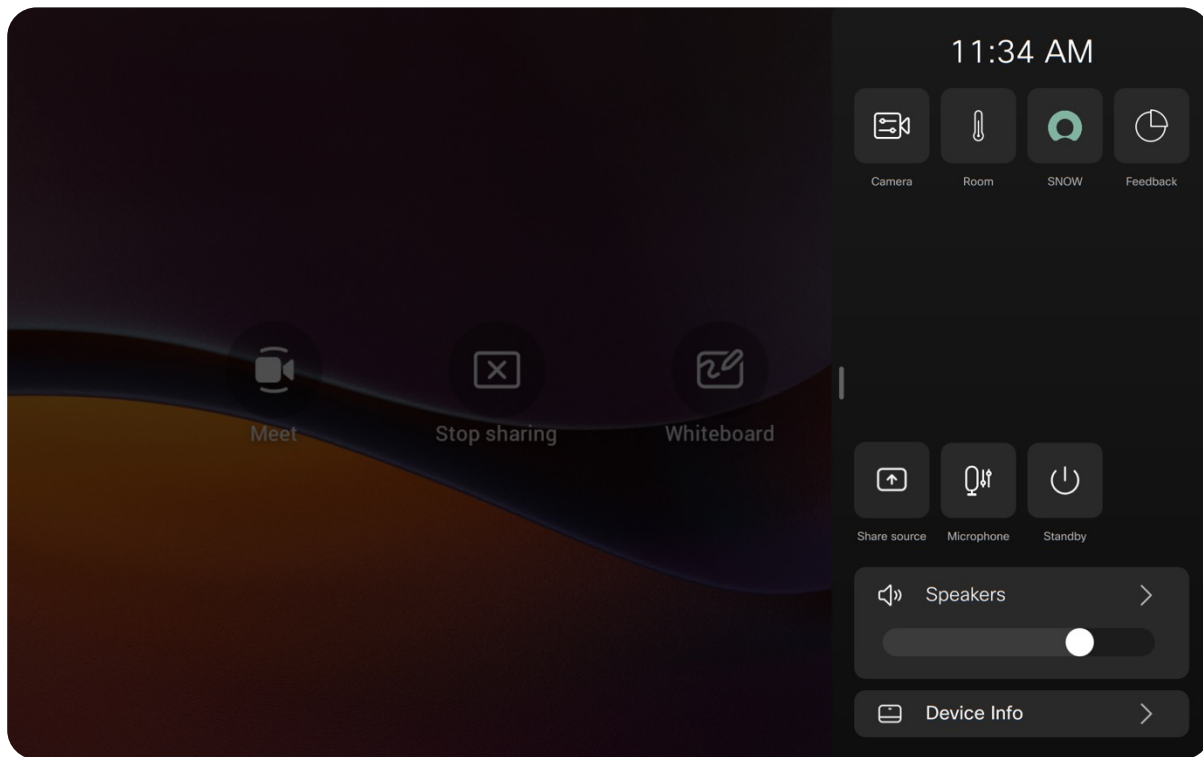
- Can extend a device's access to other services in and out of your environment
- Can simplify, centralize and scale your solutions
- Common Tools include
 - NodeJs and the JsxApi Module
 - PutXML Requests
 - Workspace Integrations

Device xAPI, UI Extension and Microsoft Teams Rooms



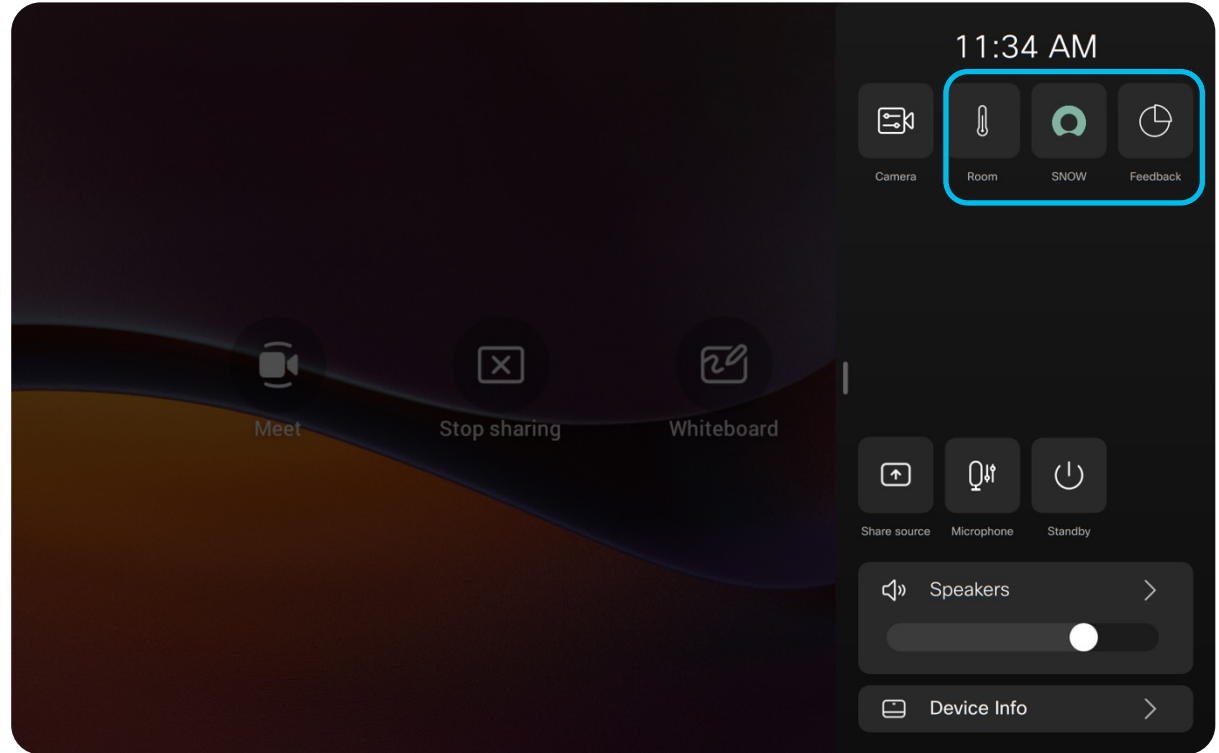
What is available?

- The same points of integration
- UI Extensions (in Beta today, soon to be generally available)
- A filtered subset of device xAPI's



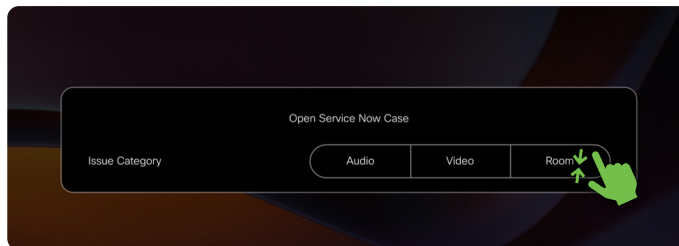
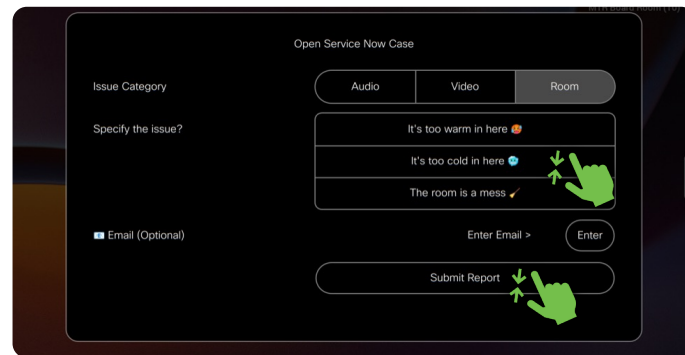
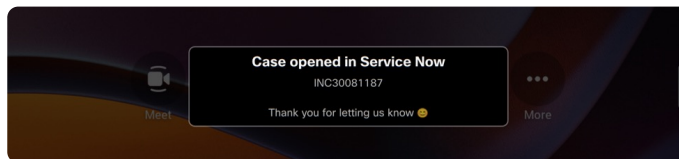
UI Extensions – important factors

- This is a Microsoft Teams Room
 - you cannot modify the MTR interface!
- UI extensions available in Control Panel
- In Call UI extensions will be on show in Webex meetings

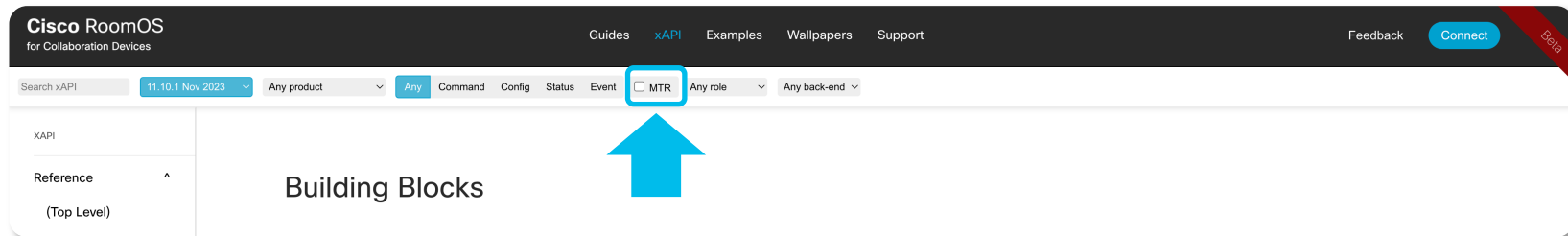


UI Extensions

- UI Extensions:
 - Limited space to show all buttons – More button will appear to overflow page
 - Simple toggle button
 - Open an interactive panel



MTR and filtered device xAPI??



- Some device xAPI have no value when MTR is configured
 - For example, xCommand Phonebook Contact Add
- Some xAPI only apply to MTR configured devices
 - For example, xStatus SystemUnit Extensions Microsoft InCall

Tips for xAPI integrations on MTR devices

- Do not assume a device xAPI solution working on a RoomOS configured device works on an MTR configured device!
- Do not assume a device xAPI solution working on MTR configured device works on a RoomOS configured device!
- Do not assume all features available on RoomOS are available or behave in the same way when MTR is configured.
 - Example: `xConfiguration Video Output Connector[N] MonitorRole`
 - The xAPI only applies to a native Webex meeting running on an MTR configured device. It would not apply when the same device is in a Microsoft Teams Meeting
 - MTR configured devices must be dual registered (Webex Control Hub and Teams Admin Centre)



Let's Build a Customization!



Where to start



Outline your project

Who's your audience?

- Business Analysts



What's are the challenges?

- The Touch Panel interface is “difficult” to operate, which disrupts their daily standup call
- We don't have a compatible Calendar Service for OBTP



Which integration tool is best to solve with?

- UI Extensions and a Macro

Outline your project

Who's your audience?

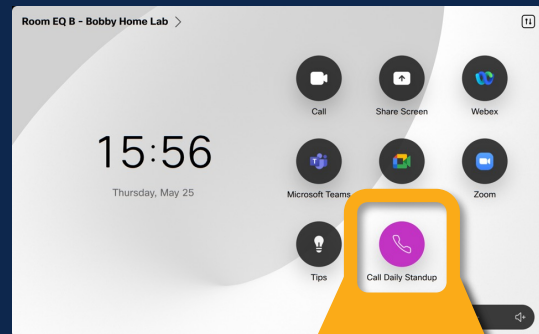
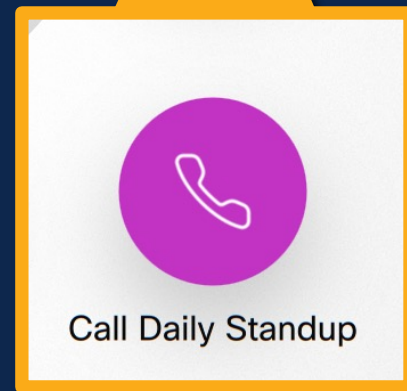
- Business Analysts

What's are the challenges?

- The Touch Panel interface is “difficult” to operate, which disrupts their daily standup call
- We don't have a compatible Calendar Service for OBTP

Which integration tool is best to solve with?

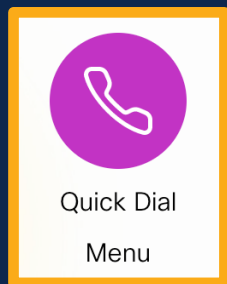
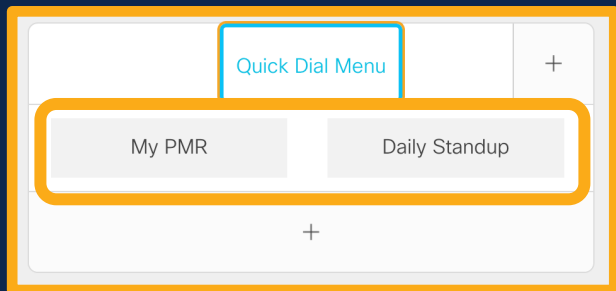
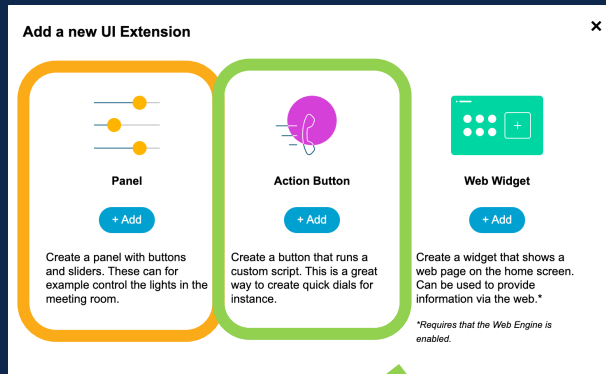
- UI Extensions and a Macro



The Quick Dial Macro

Designing the UX

Quick Dial Macro



Cisco Webex
Local Device Controls

Find page

Desk Pro

Extensions

- Quick Settings
Panel (home screen & call controls)
Quick_Settings
- WebWidget
Web Widget (home screen)
panel_1
- MRowe Zoom
Action Button (home screen)
./callNumber-MRowe Zoom
- Codec Monitor
Action Button (home screen & call controls)
codecMonitor-visible
- Call Daily Standup**
Action Button (home screen)
myQuickDial

Hidden Extensions

- Call Assistant
Panel (opened by scripts)
Adaptive_Auto_Call_Assistant_Hidden
- Codec Monitor
Panel (opened by scripts)
codecMonitor
- Config
Panel (opened by scripts)
codecConfig

New

Properties

Id
myQuickDial

Name
Call Daily Standup

Position
↓ ↑

Button location ⓘ
Home screen

Call controls
Home screen and call controls
Hidden

Icon
⚙️ ⌚ 📺 🔔
📍 📞 ? 🗣️

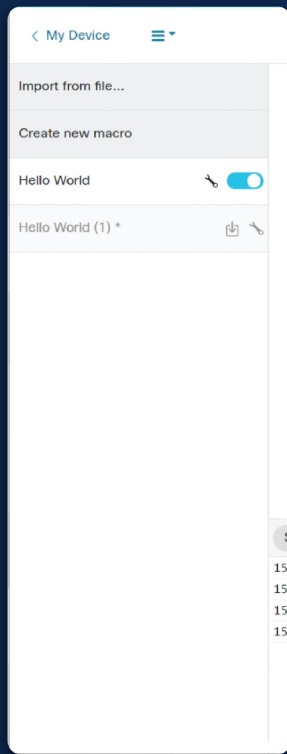
Assigning UI Properties

Quick Dial Macro

- Set Unique IDs for All Elements
- Assign a Location
- Customize the Icon

The Macro Editor IDE

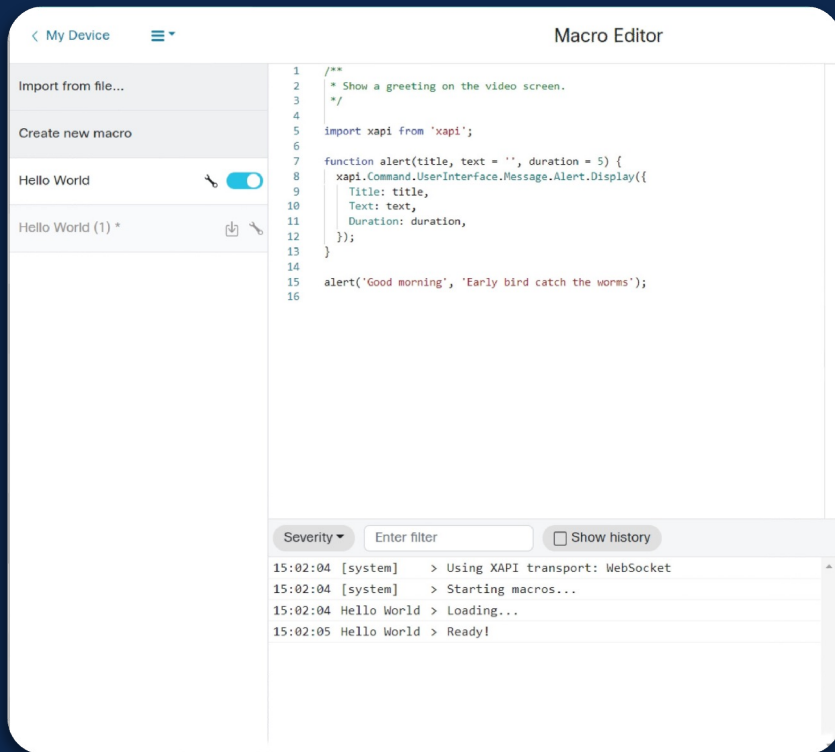
Quick Dial Macro



File Menu

The Macro Editor IDE

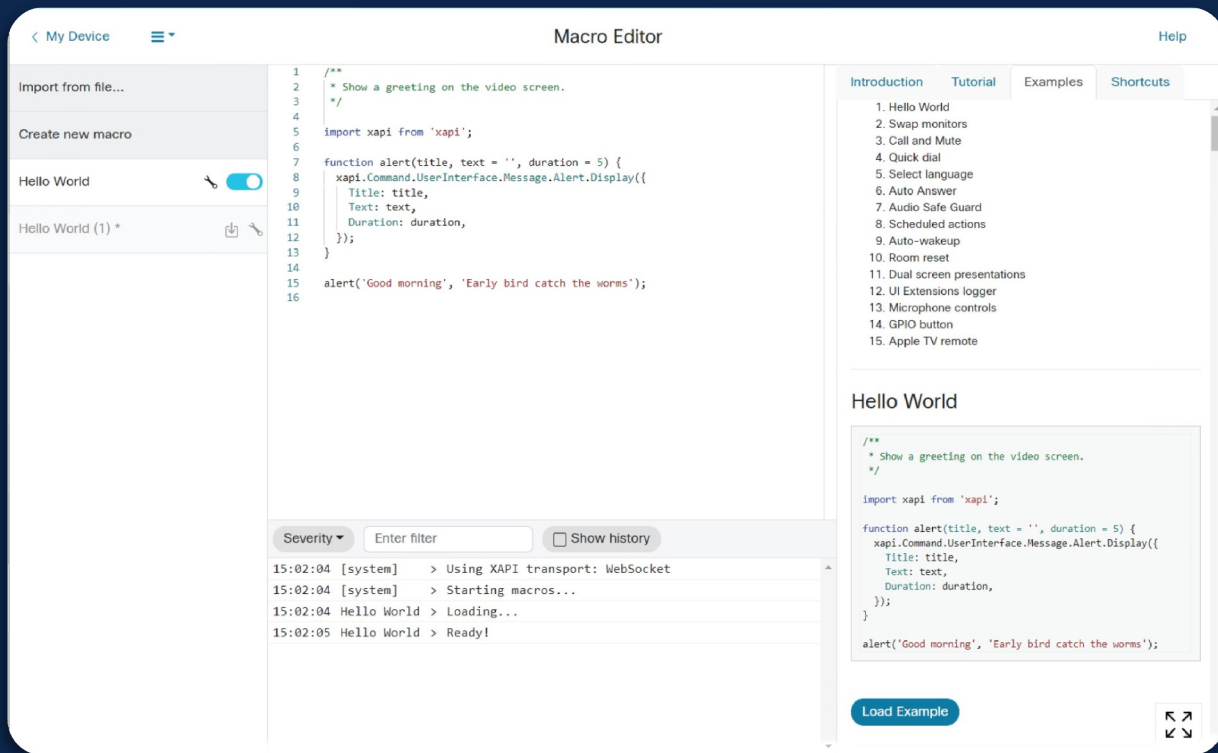
Quick Dial Macro



Integrated
Development
Environment

The Macro Editor IDE

Quick Dial Macro



Help
And
Examples

Writing the script

Quick Dial Macro

First Try

```
1| import xapi from `xapi`;
2|
3| xapi.Event.UserInterface.Extensions.Panel.Clicked.on(event => {
4|   if (event.PanelId == `myQuickDial`){
5|     xapi.Command.Dial({Number: `sampleNumber@domain.webex.com`})
6|   }
7| })
```

Provides function for a UI Extension Panel ID'd as myQuickDial

Simplified

```
1| import xapi from `xapi`;
2|
3| const quickDialNumber = `sampleNumber@domain.webex.com`
4|
5| //Skip a few dozen lines...
...
61| xapi.Event.UserInterface.Extensions.Panel.Clicked.on(event => {
62|   if (event.PanelId == `myQuickDial`){
63|     xapi.Command.Dial({Number: quickDialNumber})
64|   }
65| })
```

Provides the same function as above, but the number was elevated, making it easier to edit in the future as the need changes

Quick Dial Macro

Amplified

```
1| import xapi from `xapi`;
2|
3| const quickDial_ID_Header= `./callNumber~`;
4|
5| const quickDialNumbers = [
6|   { name: 'My PMR', uri: 'myPMR@domain.webex.com'},
7|   { name: 'Daily Standup', uri: 'dailyCall@webex.com'}
8| ];
9|
10|
11| async function init(){ await buildQuickDialPanels();}
12|
13| init();
14|
15| ...
16| //Skip a few dozen lines...
17|
18| async function buildQuickDialPanels() {
19|   let currentPanellist = await generateExistingPanelsList();
20|   let quickDialList = generateQuickDialPanellist();
21|   let panelsToAdd = quickDialList.filter(x => !currentPanelList.includes(x));
22|   let panelsToRemove = currentPanelList.filter(x => !quickDialList.includes(x));
23|   for (let i = 0; i < panelsToRemove.length; i++) {
24|     await xapi.Command.UserInterface.Extensions.Panel.Remove({ PanelId: panelsToRemove[i] });
25|     console.warn({ Message: `Panel [${panelsToRemove[i]}] removed` });
26|   }
27|   for (let j = 0; j < panelsToAdd.length; j++) {
28|     for (let q = 0; q < quickDialNumbers.length; q++) {
29|       if (panelsToAdd[j] == quickDialNumbers[q].panelId) {
30|         console.log({ Message: `Building new Quick Dial Panel [${panelsToAdd[j]}]` })
31|         await xapi.Command.UserInterface.Extensions.Panel.Save({
32|           PanelId: panelsToAdd[j]
33|         }, `<Extensions>
34|         <Panel>
35|         <Type>Home</Type>
36|         <Icon>Handset</Icon>
37|         <Color>#008094</Color>
38|         <Name>${quickDialNumbers[q].name}</Name>
39|         <ActivityType>Custom</ActivityType>
40|         </Panel>
41|       </Extensions>`);
42|       }
43|     }
44|   }
45| }
```

Provides the same quick dialing user experience, but with an expanded feature set

- Simpler Setup
- Multiple Quick Dial button support
- Automatic UI Extension Management
- Detailed Logging

The solution continues to grow

Quick Dial Macro

Amplified

```
1| import xapi from 'xapi';
2|
3| const quickDial_ID_Header = `./callNumber~`;
4|
5| const quickDialNumbers = [
6|   { name: 'My PMR', uri: 'myPMR@domain.webex.com' },
7|   { name: 'Daily Standup', uri: 'dailyCall@webex.com' }
8| ];
9|
10|
11| async function init() { await buildQuickDialPanels(); }
12|
13| init();
14|
15| ...
16| //Skip a few dozen lines...
17|
18| async function buildQuickDialPanels() {
19|   let currentPanelList = await generateExistingPanelsList();
20|   let quickDialList = generateQuickDialPanelsList();
21|   let panelsToAdd = quickDialList.filter(x => !currentPanelList.includes(x));
22|   let panelsToRemove = currentPanelList.filter(x => !quickDialList.includes(x));
23|   for (let i = 0; i < panelsToRemove.length; i++) {
24|     await xapi.Command.UserInterface.Extensions.Panel.Remove({ PanelId: panelsToRemove[i] });
25|     console.warn({ Message: `Panel [${panelsToRemove[i]}] removed` });
26|   }
27|   for (let j = 0; j < panelsToAdd.length; j++) {
28|     for (let q = 0; q < quickDialNumbers.length; q++) {
29|       if (panelsToAdd[j] === quickDialNumbers[q].panelId) {
30|         console.log({ Message: `Building new Quick Dial Panel [${panelsToAdd[j]}]` });
31|         await xapi.Command.UserInterface.Extensions.Panel.Save({
32|           PanelId: panelsToAdd[j],
33|           <Extensions>
34|           <Panel>
35|             <Type>Home</Type>
36|             <Icon>Handset</Icon>
37|             <Color>#008094</Color>
38|             <Name>${quickDialNumbers[q].name}</Name>
39|             <ActivityType>Custom</ActivityType>
40|           </Panel>
41|         </Extensions>
42|       );
43|     }
44|   }
45| }
46|
```

Simpler Setup with
support for Multiple Quick Dial Buttons

```
const quickDialNumbers = [
  { name: 'My PMR', uri: 'myPMR@dm.webex.com' },
  { name: 'Standup', uri: 'dailyCall@webex.com' }
];
```

Quick Dial Macro

Amplified

```
async function buildQuickDialPanels() {  
  let currentPanelList = await generateExistingPanelsList();  
  let quickDialList = generateQuickDialPanelList();  
  let panelsToAdd = quickDialList.filter(x => !currentPanelList.includes(x));  
  let panelsToRemove = currentPanelList.filter(x => !quickDialList.includes(x));  
  for (let i = 0; i < panelsToRemove.length; i++) {  
    await xapi.Command.UserInterface.Extensions.Panel.Remove({ PanelId: panelsToRemove[i] });  
    console.warn({ Message: `Panel [${panelsToRemove[i]}] removed` })  
  }  
  for (let j = 0; j < panelsToAdd.length; j++) {  
    for (let q = 0; q < quickDialNumbers.length; q++) {  
      if (panelsToAdd[j] == quickDialNumbers[q].panelId) {  
        console.log({ Message: `Building new Quick Dial Panel [${panelsToAdd[j]}]` })  
        await xapi.Command.UserInterface.Extensions.Panel.Save({  
          PanelId: panelsToAdd[j]  
        }, `  
        <Panel>  
          <Type>Home</Type>  
          <Icon>Handset</Icon>  
          <Color>#008094</Color>  
          <Name>${quickDialNumbers[q].name}</Name>  
          <ActivityType>Custom</ActivityType>  
        </Panel>  
      </Extensions>`);  
      }  
    }  
  }  
};
```

Automatic UI Extension Management

Compare existing UI against
quickDialNumbers object and
remove outliers

Build any new panels
introduced in the
quickDialNumbers object

Quick Dial Macro

Quick Dial Macro

Amplified

```
async function buildQuickDialPanels() {
  let currentPanelList = await generateExistingPanelsList();
  let quickDialList = generateQuickDialPanelList();
  let panelsToAdd = quickDialList.filter(x => !currentPanelList.includes(x));
  let panelsToRemove = currentPanelList.filter(x => !quickDialList.includes(x));
  for (let i = 0; i < panelsToRemove.length; i++) {
    await xapi.Command.UserInterface.Extensions.Panel.Remove({ PanelId: panelsToRemove[i] });
    console.warn({ Message: `Panel [${panelsToRemove[i]}] removed` });
  }
  for (let j = 0; j < panelsToAdd.length; j++) {
    for (let q = 0; q < quickDialNumbers.length; q++) {
      if (panelsToAdd[j] == quickDialNumbers[q].panelId) {
        console.log({ Message: `Building new Quick Dial Panel [${panelsToAdd[j]}]` });
        await xapi.Command.UserInterface.Extensions.Panel.Save({
          PanelId: panelsToAdd[j]
        }, `<Extensions>
<Panel>
  <Type>Home</Type>
  <Icon>Handset</Icon>
  <Color>#008094</Color>
  <Name>${quickDialNumbers[q].name}</Name>
  <ActivityType>Custom</ActivityType>
</Panel>
</Extensions>`);
      }
    }
  }
}
```

Detailed Logging

Congrats! You just built a button 🎉



What else is possible?



The new challenge

Campfire Room

Who's your audience?

- The executive leadership team

What's are the challenges?

- A conference room with a square table and a hollow center
- A want to see guests across the table and on the call
- Speaker tracking should capture all in room guests, but you have a 360 Degree view
 - A Codec can only support 1 Speakertracking Camera at a time...

Which integration tool is best to solve with?

- Macros



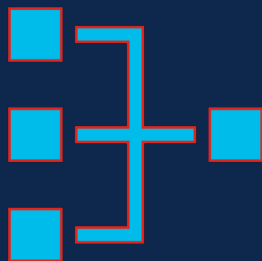
How do we solve for this space?

Campfire Room

Room Kit EQ



Room Kit Pro



Settings

Configurations Statuses Room Types Audio and Video

Cameras Video Inputs and Outputs Microphones Presenter Track All Peripherals

Microphone 1 - Active	<div><div></div></div>
Microphone 2 - Active	<div><div></div></div>
Microphone 3 - Active	<div><div></div></div>



Table Mic Pro

Campfire Room



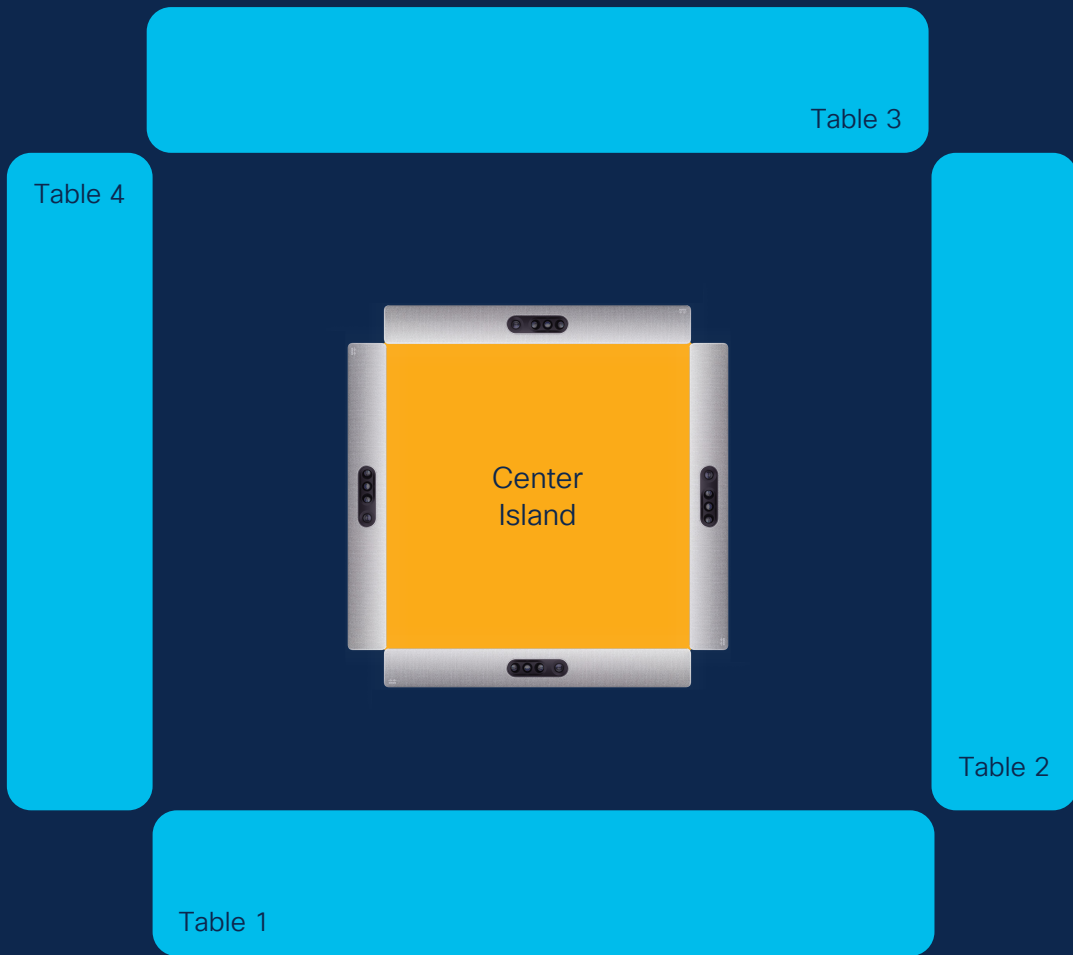
Quad Camera

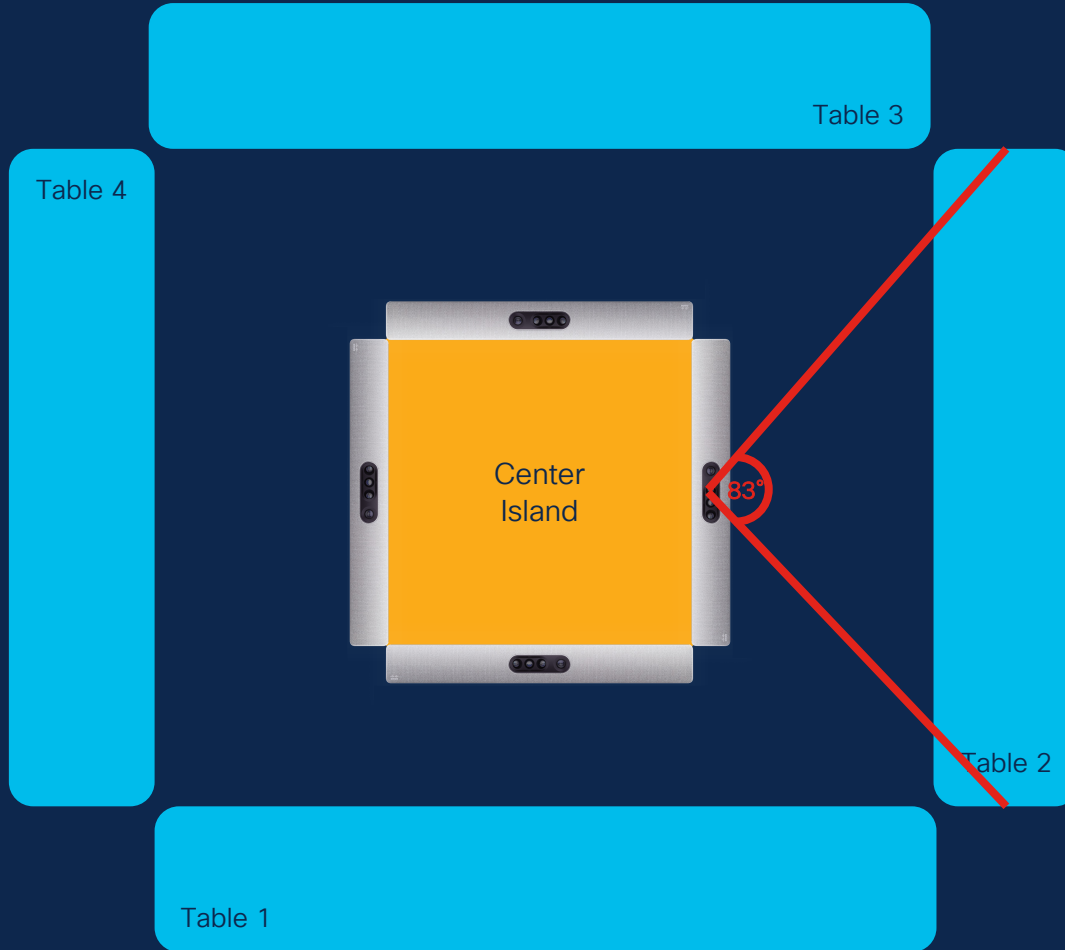


Center
Island



Table 1





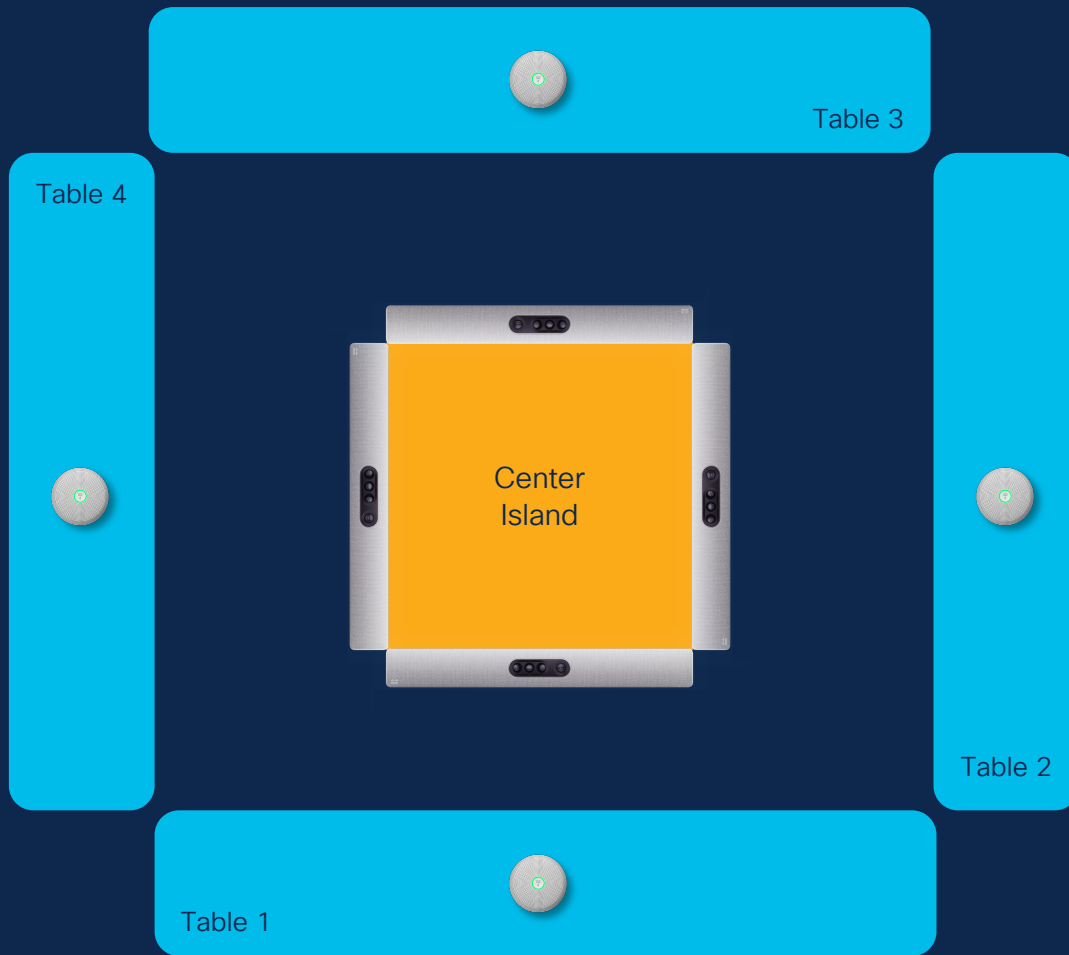
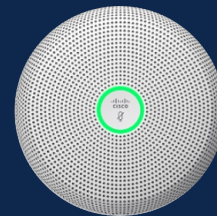


Table Microphone Pro

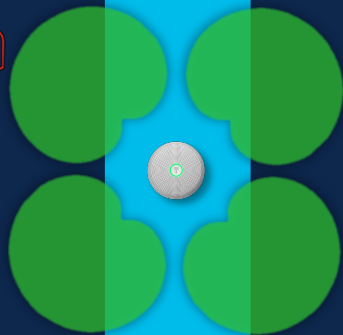
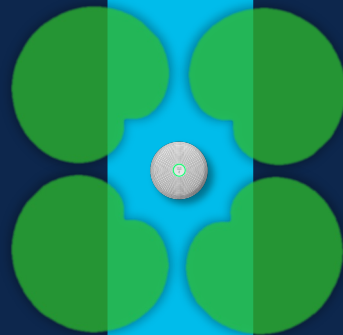
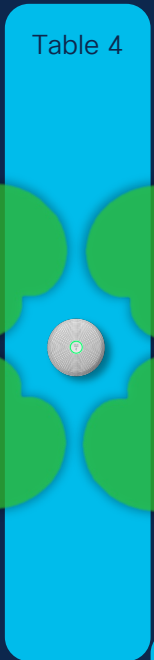
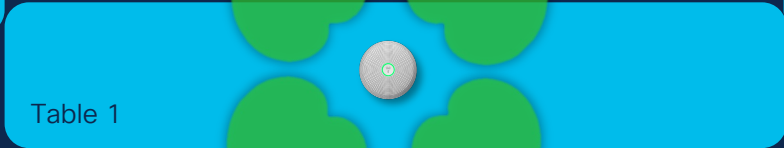
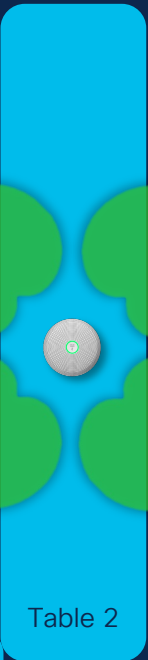
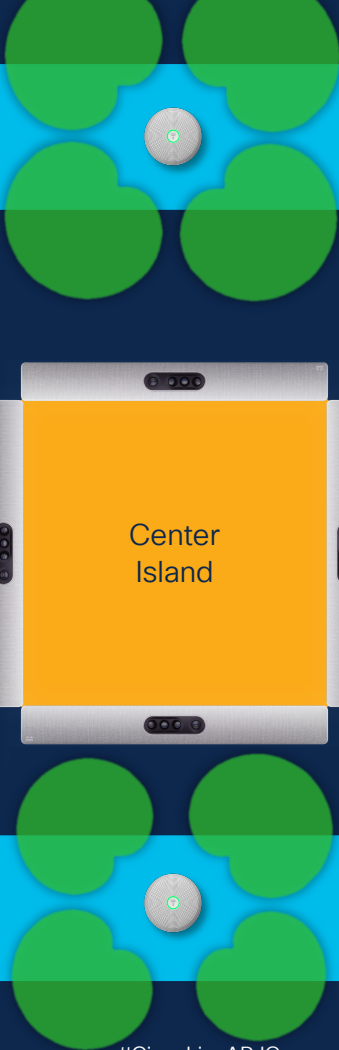




Table 4

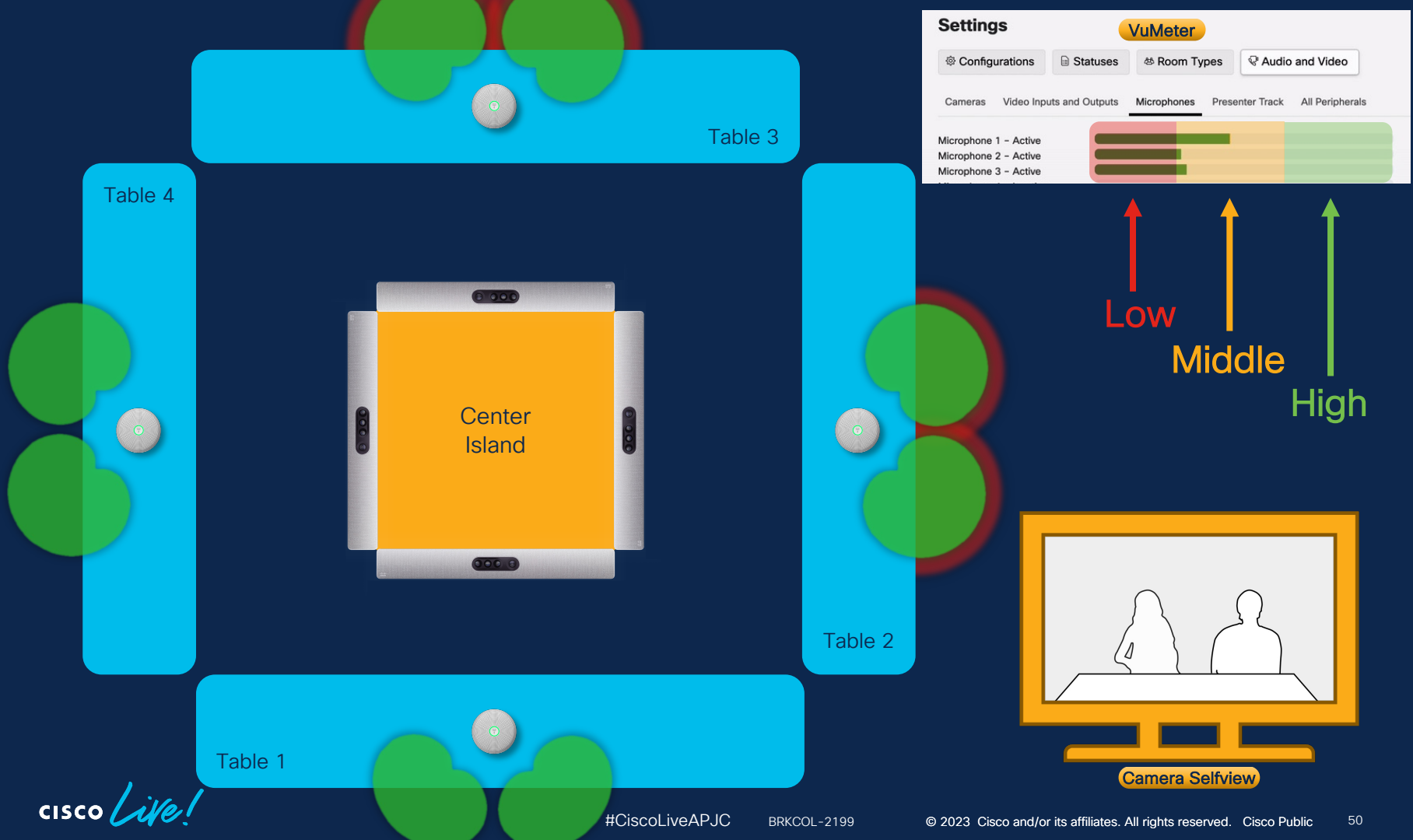
Table 3

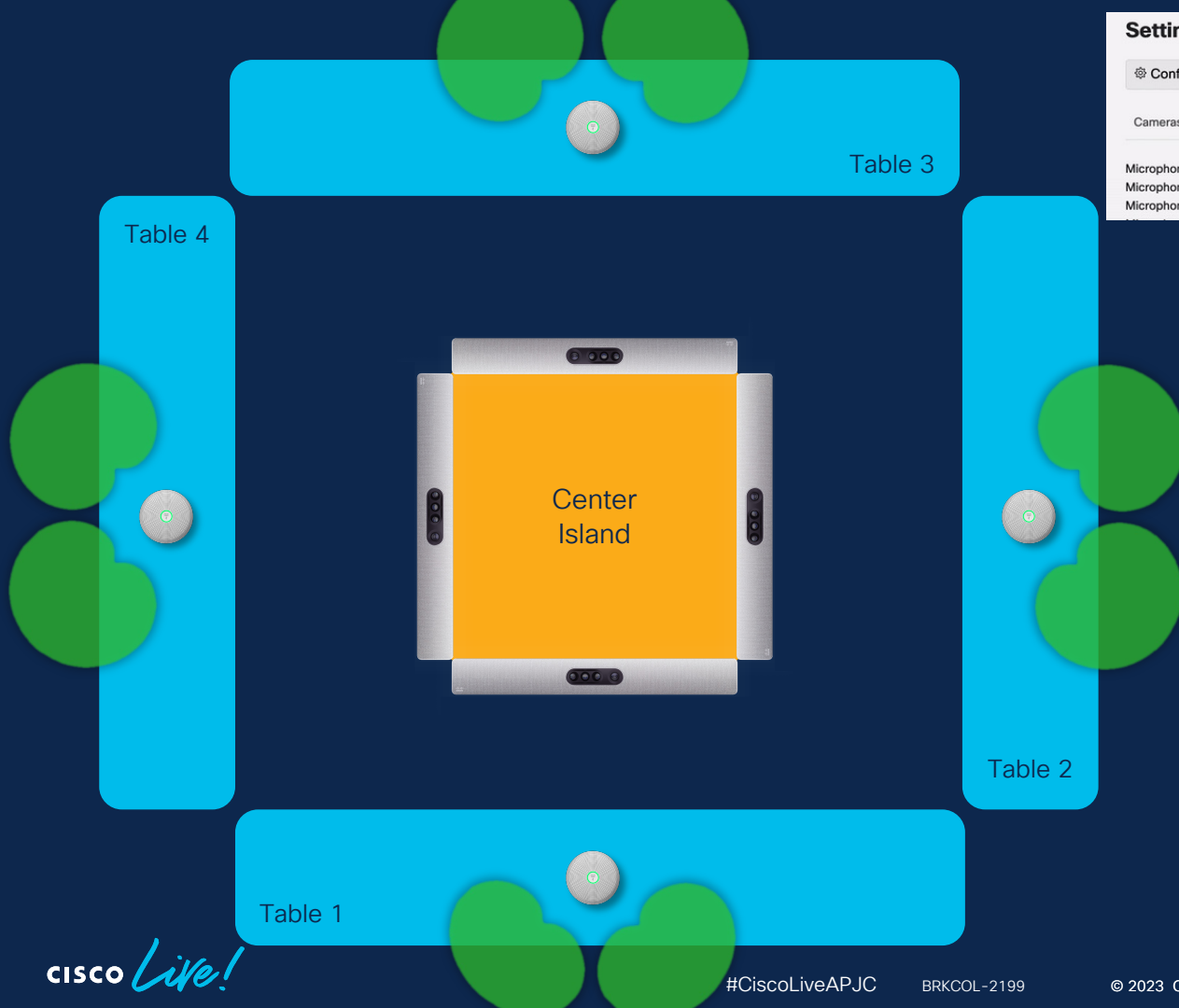
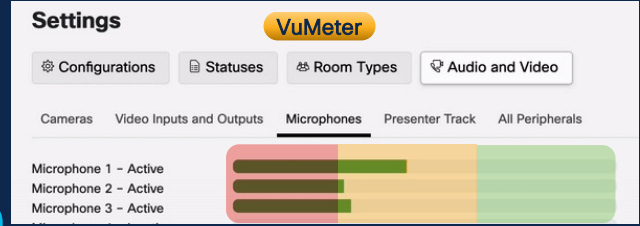
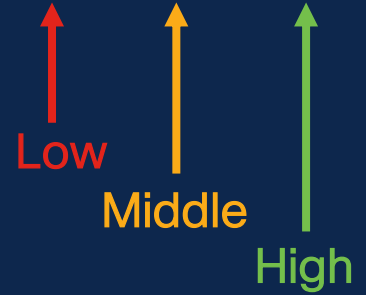
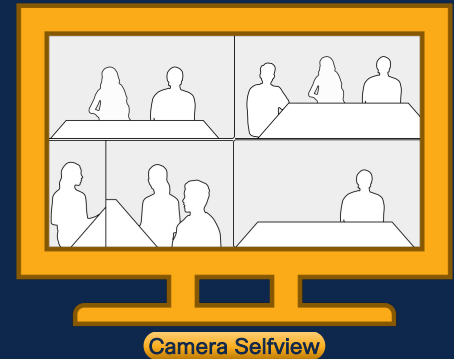
Table 2

Table 1



Center
Island





Ready to Deploy?



Customization Deployment Options



Control Hub

Endpoint API's

Device Backups

Automation, CE-Deploy and 3rd Party

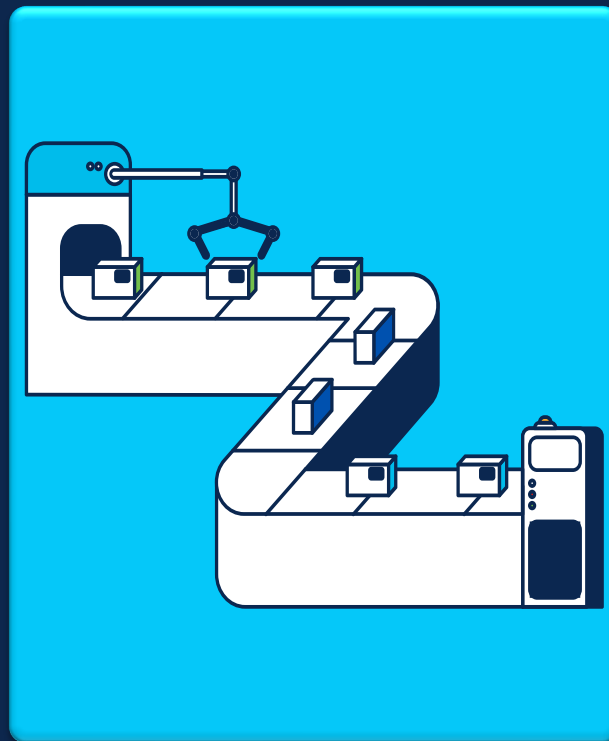
Customization Deployment Assets

Commands

Enable Macro Engine
Push Macros
Activate Macros
Push Extensions
Viewable Extensions

Data

Macro JavaScript
Custom Extension XML



Customization Deployment Options

Configurations

Device Web Portal
Control Hub(Bulk)
Cloud xAPI
Endpoint xAPI
CE-Deploy(Bulk)

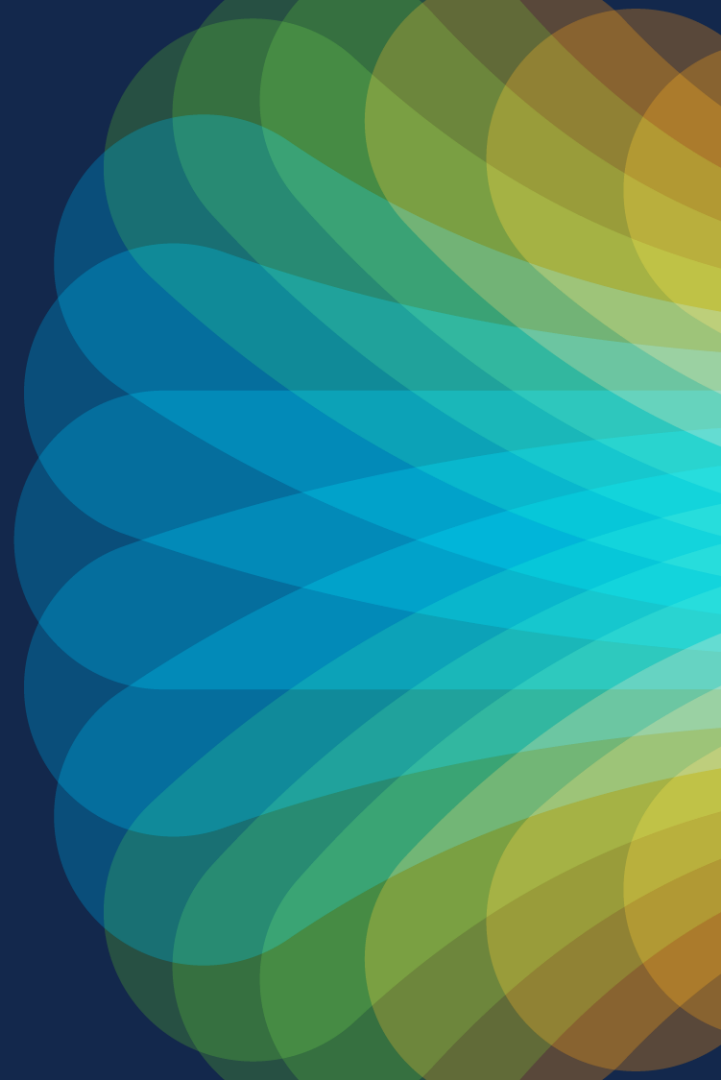
Macro's

Device Web Portal
Control Hub(single)
Cloud xAPI
Endpoint xAPI
CE-Deploy(Bulk)

UI Extensions

Device Web Portal
Cloud xAPI
Endpoint xAPI
CE-Deploy(Bulk)

Control Hub



Control Hub Device Templates

- Applied across multiple devices
- Templates based on use case
 - General settings
 - Divisible room
 - Single configuration
- One template applied at a time
- Templates settings can be undone

Apply configuration template Selected Devices 2

Select template Review Summary

Select template
Select a template to apply to the device selection.

Template

Name	AudioDefaults
Description	AudioRoomDefaults
Number of configurations ⓘ	2

Configuration	Value
Audio Default Volume	25 ⓘ
Audio Ultrasound Max Volume	42 ⓘ

Control Hub Macros

Macros

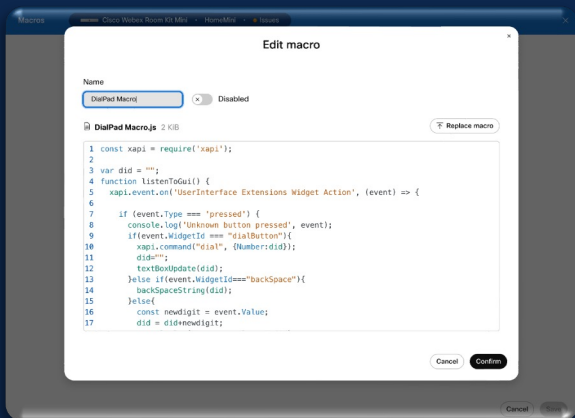
Control Hub management and provisioning of macros.

☒ Allow Control Hub to manage macros

Upload existing macros from devices in your organization to Control Hub, so they can be managed from the cloud.

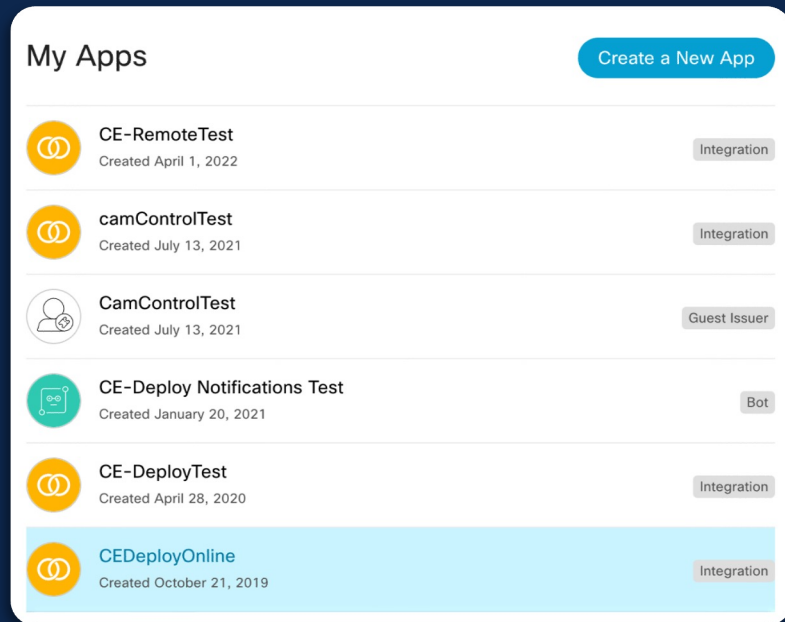
If enabled, Room, Board and Desk series devices in shared mode will upload their macros to Control Hub. Control Hub administrators may then edit macro source code, download local copies, and upload them to other devices. Devices in **personal mode** do not upload their macros to Control Hub due to privacy reasons.

Enable**



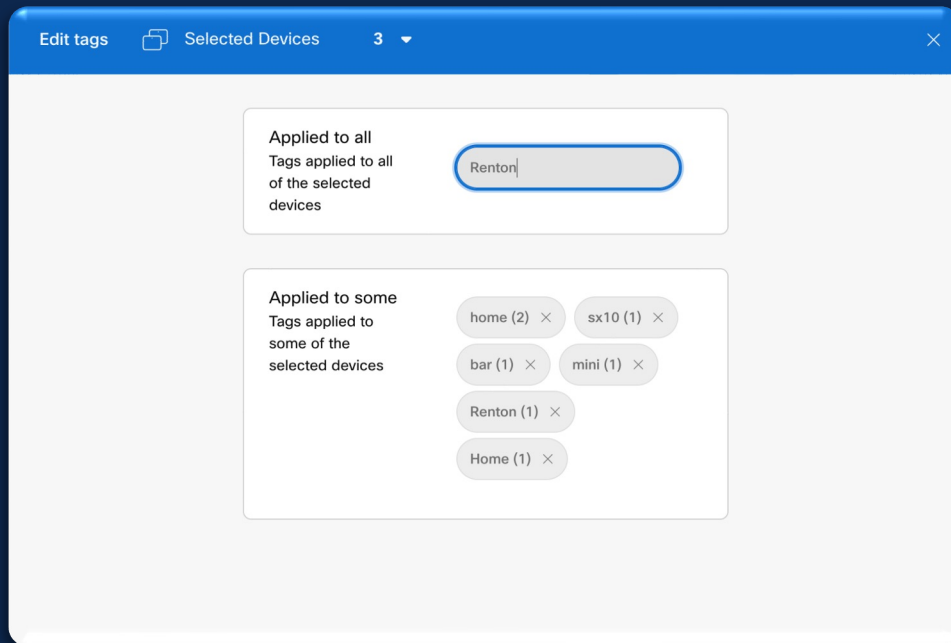
Control Hub Cloud xAPI

- Scoped Webex Integration
- Rate limit to 300 requests per minute
 - Conservative estimate
- Large body transactions can cause errors
 - Multiline commands
- Offline endpoints
 - Configurations stored by the cloud

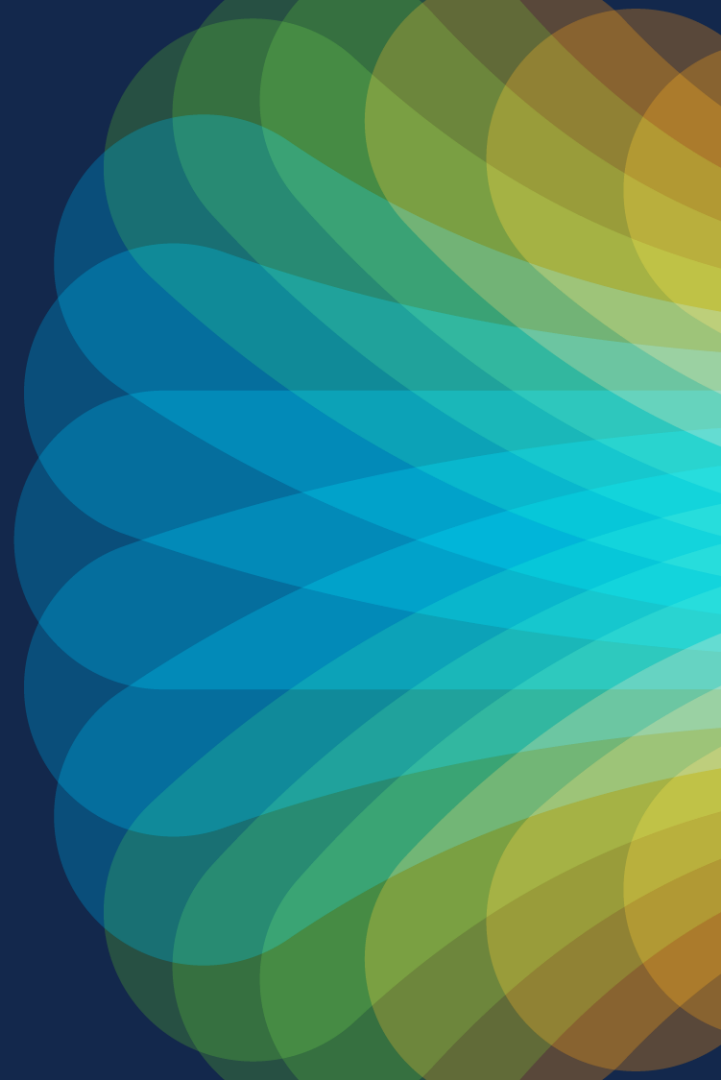


Control Hub Device Tags

- Have as many tags as you want!
- Avoid spaces in tags
 - camel Case or Pascal Case
- Examples of tags
 - Location based
 - Personal or shared device
 - Product type

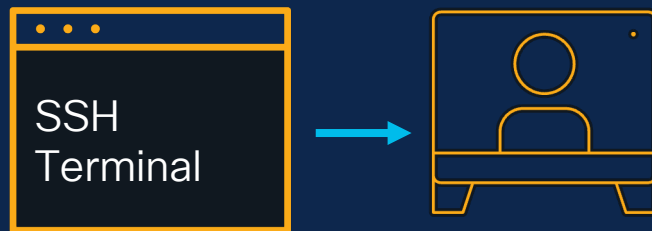
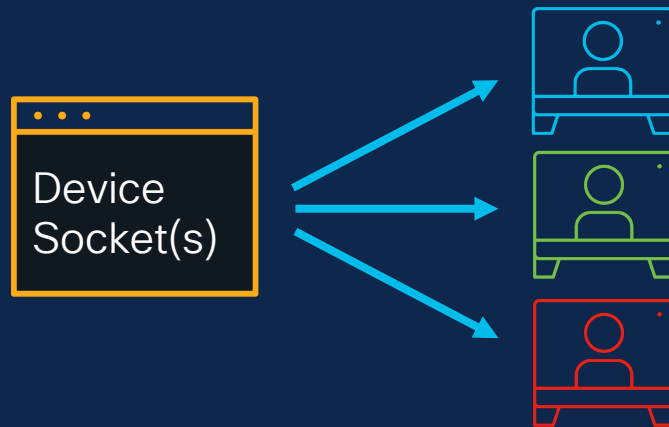


Endpoint API's



Endpoint API's for Deployment

- 3 Delivery Options
 - HTTP/HTTPS
 - SSH
 - WebSockets
- Websockets is the best option
 - jsxapi library support
 - JSON support
 - Easier to manage large scale projects



Device Backups

Device Web Admin Page Backup Templates

The screenshot shows the Cisco Webex Local Device Controls interface. The top header is teal with the Cisco Webex logo and 'Local Device Controls' text. A search bar and a user profile icon are on the right. The left sidebar is white with a teal header and contains navigation links: Home, Call, SETUP (Settings, Users, Security), CUSTOMIZATION (Personalization, UI Extensions Editor, Macro Editor, Developer API), and SYSTEM MAINTENANCE (Software, Issues and Diagnostics, Backup and Recovery). The 'Backup and Recovery' link is highlighted with a red box. The main content area is white with a teal header 'Backup and Recovery'. It contains four tabs: Backup, Restore, System Recovery, and Restart and Shutdown. The 'Backup' tab is active. Below the tabs is a section titled 'Download Backup' with a description: 'This page lets you backup a device's configurations. Configurations include xConfigurations, as well as the configurations for other features a device might support, like macros and UI Extensions.' Below this is a text prompt: 'Select items to include in your backup. Leave all items checked to generate a complete backup of the device.' A table lists various configuration items with checkboxes: Branding (Background, Branding, HalfwakeBranding), Favorites (No items installed), UI Extensions (MyPanel, MyButton), Macros (DialPad Macro, MyMacro, ResFix), Sign In Banner (Include), and Configuration (Include). At the bottom, there is a note about settings and a button to 'Remove system-specific configurations'.

Cisco Webex
Local Device Controls

192.168.128.87
Room Kit Mini

Home
Call

SETUP
Settings
Users
Security

CUSTOMIZATION
Personalization
UI Extensions Editor
Macro Editor
Developer API

SYSTEM MAINTENANCE
Software
Issues and Diagnostics
Backup and Recovery

Backup and Recovery

Backup Restore System Recovery Restart and Shutdown

Download Backup

This page lets you backup a device's configurations. Configurations include xConfigurations, as well as the configurations for other features a device might support, like macros and UI Extensions.

Select items to include in your backup. Leave all items checked to generate a complete backup of the device.

Branding	☒ Background ☒ Branding ☒ HalfwakeBranding
Favorites	No items installed
UI Extensions	☒ MyPanel ☒ MyButton
Macros	☒ DialPad Macro ☒ MyMacro ☒ ResFix
Sign In Banner	☒ Include
Configuration	☒ Include

The configuration listed below can be modified for a partial backup of the configuration.

Note: Settings such as Network, SIP URI and SystemUnit Name may specifically apply to a certain device. Applying a backup with these settings to a different device may result in it becoming unreachable on the network. To avoid this, consider removing those settings from the list below.

Remove system-specific configurations

Backup Template Creation

Remove system specific
Configuration and
download template

Remove system-specific configurations

```
Apps WallpaperBundles HalfwakeImage: Auto
Audio DefaultVolume: 50
Audio Input HDMI 1 Level: -5
Audio Input HDMI 1 Mode: On
Audio Input HDMI 1 VideoAssociation MuteOnInactiveVideo: On
Audio Input USBMicrophone 1 EchoControl Mode: On
Audio Input USBMicrophone 1 Level: 5
Audio Microphones Mute Enabled: True
Audio Microphones NoiseRemoval Mode: Enabled
Audio Microphones UsbPassthrough MuteButton: Active
```

☒ Select All



Removed 16 lines of system specific configurations.

Download

Restoring Backup Template

```
You are now connected via CE-Deploy $  
Welcome to  
Cisco Codec Release RoomOS 11.3.1.5 7762617dc56  
SW Release Date: 2023-03-09  
*r Login successful  
OK  
  
xCommand Provisioning Service Fetch Checksum: {value} ChecksumType: SHA512 Mode:  
Replace Origin: Other URL: http://fileserver.com/file
```

Send xAPI command



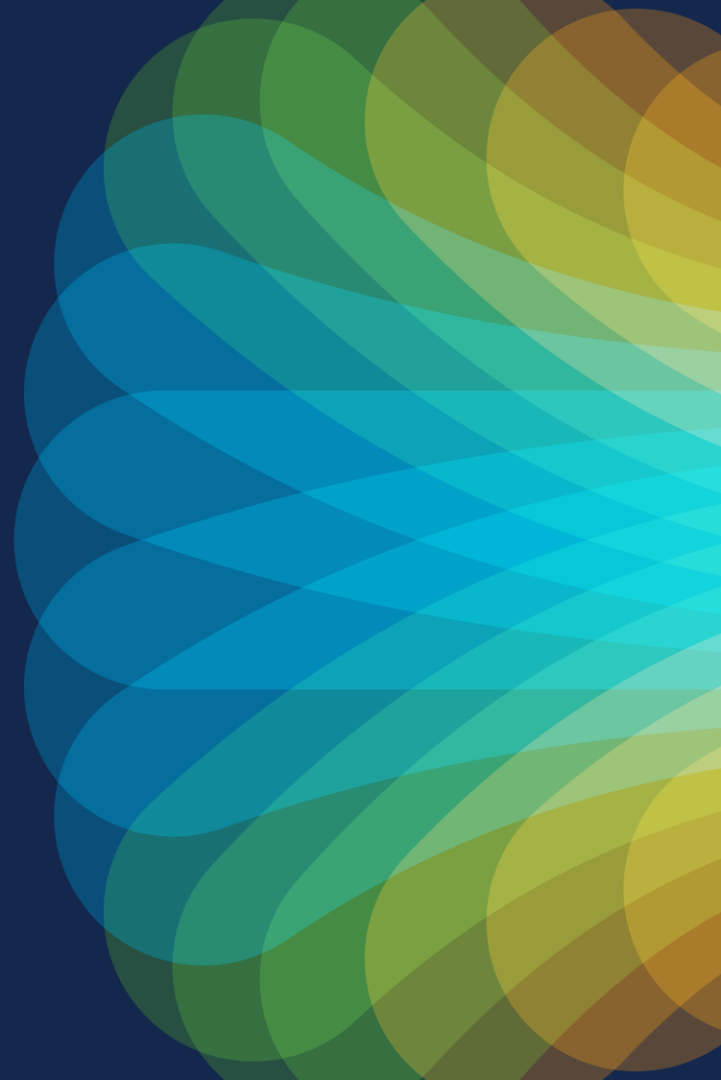
Hosted
file

Endpoint fetches
file



Calculates
Checksum and
performs restore

Automation



Automation Task



- Custom branding
- Macros
- Backups
- Fresh installs
- Configuration deployment
- Contacts
- Certificates

Cloud Automation

✓ Polling required for new endpoints

🖥️ Remote endpoints

👤 Personal endpoints

📊 Device tags in Control Hub

🔥 No firehose webhook for workspaces

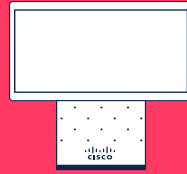
Workspaces Integration – new building foundation

Device Centric Model
1:1 Approach



Workspace integration
1:Many Approach

Macros



Webhooks / Long Poll Queues



Events



Location
Analytics



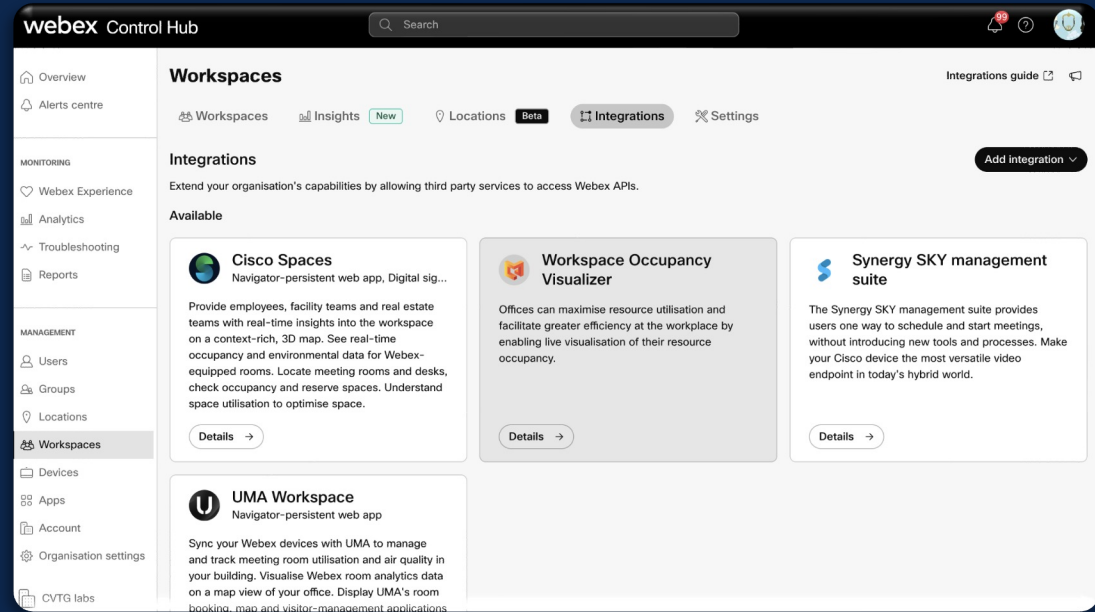
People
Experience



Real-time
monitoring

Workspace Integrations

- OAuth 2.0
- Identity of a machine account
- Receive webhook or queue-based notifications for status and events from org devices
- Small subset of the xStatus and xEvent
- Granular xAPI access control



Automation with Cloud xAPI's

- Monitoring for API congestion
- Queueing and carrying out commands
- xConfiguration read only commands via cloud xAPI
- Offline devices

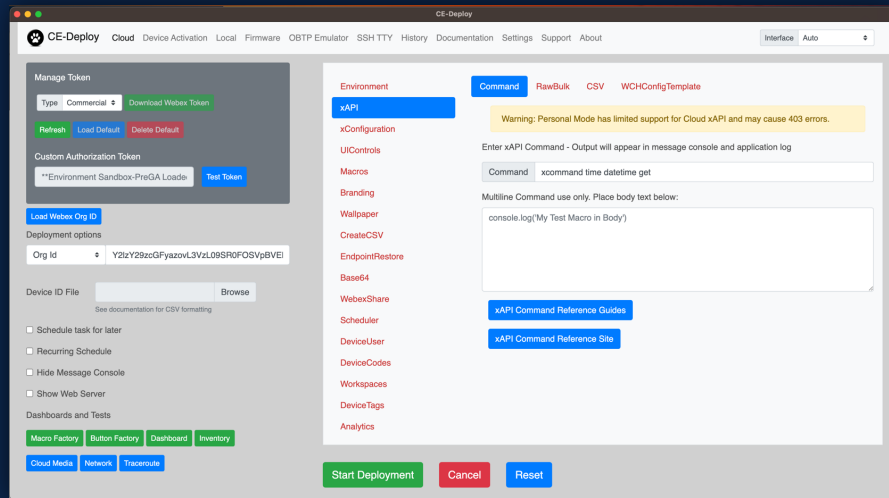
```
const limiter : Bottleneck = new Bottleneck( options: {  
  reservoir: 100, |  
  reservoirRefreshAmount: 100,  
  reservoirRefreshInterval: 60 * 1000,  
  
  // also use maxConcurrent and/or minTime for safety  
  maxConcurrent: 1,  
  minTime: 200  
});
```

CE-Deploy

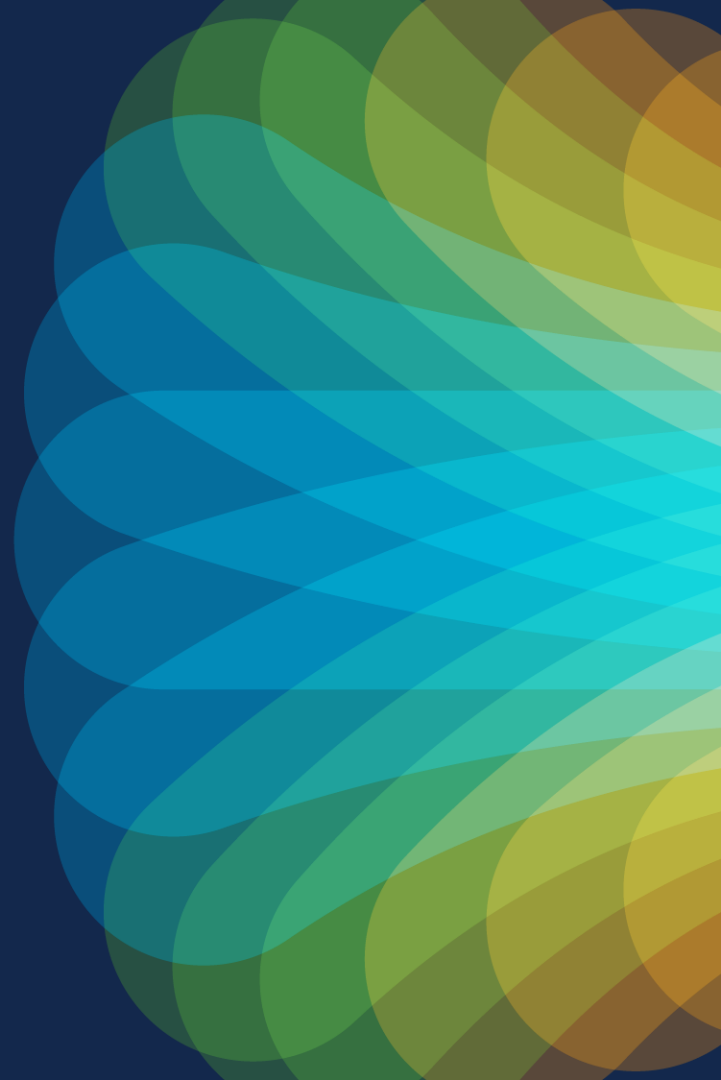


CE-Deploy

- Built based on customer feedback
- Not an automation tool
- Developed using common frameworks and libraries
 - Electron – App UI
 - JSxAPI – Accessing Endpoint API
 - Bottleneck – Job queueing
- Public API's
 - Cloud and endpoint xAPI's, putxml

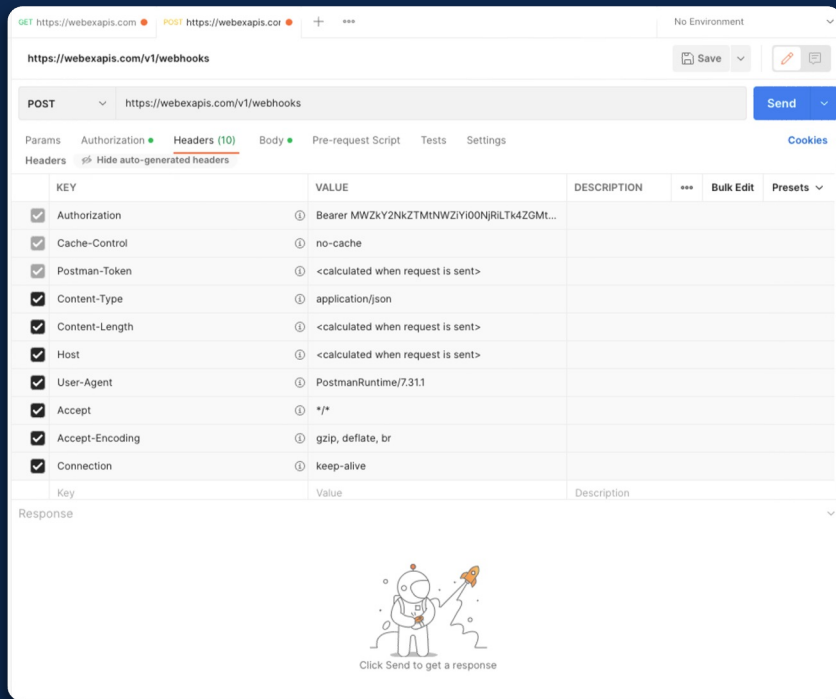


Deployment Tools



Tools

- Postman
- Coding tools
 - NodeJS
 - Visual Studio Code(free)
 - WebStorm(paid)
 - Python
 - Visual Studio Code(free)





Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.



Meet the Expert Opening Hours:

Tuesday	3:00pm – 7:00pm
Wednesday	11:15am – 7:00pm
Thursday	9:30am – 4:00pm
Friday	10:30am – 1:30pm

Session Surveys

We would love to know your feedback on this session!

- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt



Continue your education



- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Expert meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand



The bridge to possible

Thank you

CISCO *Live!*

#CiscoLiveAPJC

The background of the slide is a vibrant, abstract graphic. It features a series of overlapping, wavy bands of color in a rainbow spectrum, transitioning from red and orange on the left to yellow, green, and blue on the right. On the right side, there is a bright, multi-colored sunburst or starburst effect that radiates outwards, creating a sense of energy and movement. The overall composition is dynamic and celebratory.

cisco *Live!*

Let's go

#CiscoLiveAPJC