



CE Customization

ユーザ インターフェイスの拡張機能とマクロ (CE9.5 向け)

CE Customization では、Touch10 で操作するビデオ システムと DX シリーズ ユーザ インターフェイスにカスタム要素を追加することができます。そのようなユーザ インターフェイス拡張の例としては、照明やブラインドの室内制御や、外部制御システムによって完全に管理されるその他の周辺機器 (使用可能なビデオ ソース数を拡張する 1 つ以上のビデオ スイッチなど) の制御があります。

マクロによって、ビデオ エンドポイント動作の一部を自動化することができる JavaScript コードのスニペットを記述できます。

シスコのビデオ システム本体と周辺機器を Touch10/DX シリーズのユーザ インターフェイスから制御できるようになるため、会議室全体で一貫したユーザ エクスペリエンスを実現できます。

現在のバージョンの CE Customization ユーティリティは、バージョン CE9.5 以降のコラボレーション エンドポイント ソフトウェアを実行している SX、MX、DX、およびルーム シリーズのビデオ システムで利用できます。

WebEx 対応システムのマクロは単なる機能プレビューです。

SX10 ではマクロはサポートされないことに注意してください。

このドキュメントの内容

パート 1	
室内制御 (In-Room Control)	
用語の定義	4
室内制御	5
ユーザ インターフェイスの作成	
Touch10 用のユーザ インターフェイスの作成	7
室内制御エディタの起動	8
室内制御エディタについて	9
現在の設定のプレビュー	10
アプリケーション プログラム インターフェイス (API)	
室内制御のプログラミング用 API	12
ウィジェット	
ウィジェットの概要	17
スイッチ	18
スライダー	19
ボタン	20
グループ ボタン	21
アイコン ボタン	22
スピナー	23
テキスト	24
方向パッド	25
スペーサー	26
コマンド リファレンス	
イベント	28
コマンド	30
ステータス	31
インタラクティブ メッセージの作成	
インタラクティブ メッセージの仕組み(I)	33
インタラクティブ メッセージの仕組み(II)	34
トラブルシューティング	
トラブルシューティングのヒント	36
ヒントとテクニック	
推奨されるベスト プラクティス	38
室内制御エディタと拡張機能 API へのアクセス許可	40

パート 2	
室内シミュレータ	
シミュレータの実行	42
パート 3	
ビデオ スイッチの使用	
サードパーティ製ビデオ スイッチによる使用可能なビデオ ソース数の拡張	44
コマンドの詳細	45
ビデオ スイッチの例	46
パート 4	
マクロでの操作	
マクロの作成	48
マクロ エディタ パネル	49
確認事項	50
パート 5 NEW	
オーディオ コンソール	
オーディオ接続のカスタマイズ	52
オーディオ コンソール パネル	53

本ガイドの使用にあたって

JavaScript が有効なデバイスで本書をお読みいただくと、左側のメニュー バーと目次のエントリに設定されたすべてのハイパーリンクをご利用いただけます。クリックすると、そのトピックに移動します。

製品に関する資料

Cisco TelePresence Systems のユーザ ガイドとコンプライアンスおよび安全に関する情報は、<http://www.cisco.com/go/telepresence/docs> で入手できます。

定期的にシスコ Web サイトにアクセスし、このガイドの最新版を入手することをお勧めします。

エディタにアクセスできるユーザ

室内制御エディタにアクセスするには、管理者権限が必要です。

ただし、管理者は、室内制御用のユーザ アカウントを作成することができます。このアカウントを使用すると、コーデックにログインして室内制御エディタを実行できます。Web インターフェイスの他の部分には、このアカウントではアクセスできません。

SSH を使用してコーデックにログインした場合は、ごく一部の API セットのみアクセスできません。

パート 1 室内制御



用語の定義

このドキュメントでは、次の用語を使用します。

ビデオ システム。バージョン CE9.3 以降のコラボレーション エンドポイント ソフトウェアが実行されている Cisco TelePresence MX シリーズ、SX シリーズ、または DX シリーズのビデオ システムまたはコーデック。ビデオ デバイスとも呼ばれます。フル機能を実現するには、新しいバージョンの CE ソフトウェアが必要です。

制御システム。周辺機器用ハードウェア ドライバを含むサードパーティ製の制御システム (例: Crestron、AMX、Raspberry Pi)。

Touch10。MX シリーズおよび SX シリーズのビデオ システム向けのタッチ式制御デバイス。製品の正式名称は Cisco TelePresence Touch10 です。Touch10 コントローラ、Touch10 ユーザ インターフェイス、または Touch10 パネルとも呼ばれます。

室内制御パネル。室内にあるサードパーティ製の周辺機器のコントロールを備えたパネル。室内制御パネルを開くには、Touch10 のステータス バーで対応する室内制御アイコンをタップします。[「ユーザ インターフェイスの作成」](#)の章を参照してください。

室内制御エディタ。室内制御パネルを作成するための、使いやすいドラッグアンドドロップ エディタ。

xAPI。ビデオ システムの双方向 API。xAPI を使用すると、サードパーティ製アプリケーションとビデオ システムが相互に接続し、通信することができます。

ウィジェット。Touch10 の室内制御パネルの構築に使用できる、ボタン、スライダー、テキスト フィールドなどのユーザ インターフェイス要素。

室内制御とは

室内制御では、Touch10 のユーザ インターフェイスにカスタム要素を追加できます。そのようなユーザ インターフェイス拡張の例としては、照明やブラインドの制御や、外部制御システムによって完全に管理されるその他の周辺機器 (使用可能なビデオ ソース数を拡張する 1 つ以上のビデオ スイッチなど) の制御があります。

シスコのビデオ システムとその他の周辺機器を Touch10 ユーザ インターフェイスから制御できるようになるため、会議室全体で一貫したユーザ エクスペリエンスを実現できます。

このドキュメントで説明されている室内制御機能のバージョンは、バージョン CE9.6 以降のコラボレーション エンドポイント ソフトウェアが実行されている MX、SX (SX10 ではない)、および DX シリーズのビデオ システムで使用できます。

Touch10 / DX のユーザ インターフェイスをカスタマイズして、会議室にある周辺機器を制御できます。たとえば、サウンドやムービーの再生、照明やブラインドの制御などを行うことができます。

また、通話中や外部発信中のみ表示される、コンテンツ依存型のコントロールを追加することもできます。

パネルの最大数が無制限になりました。

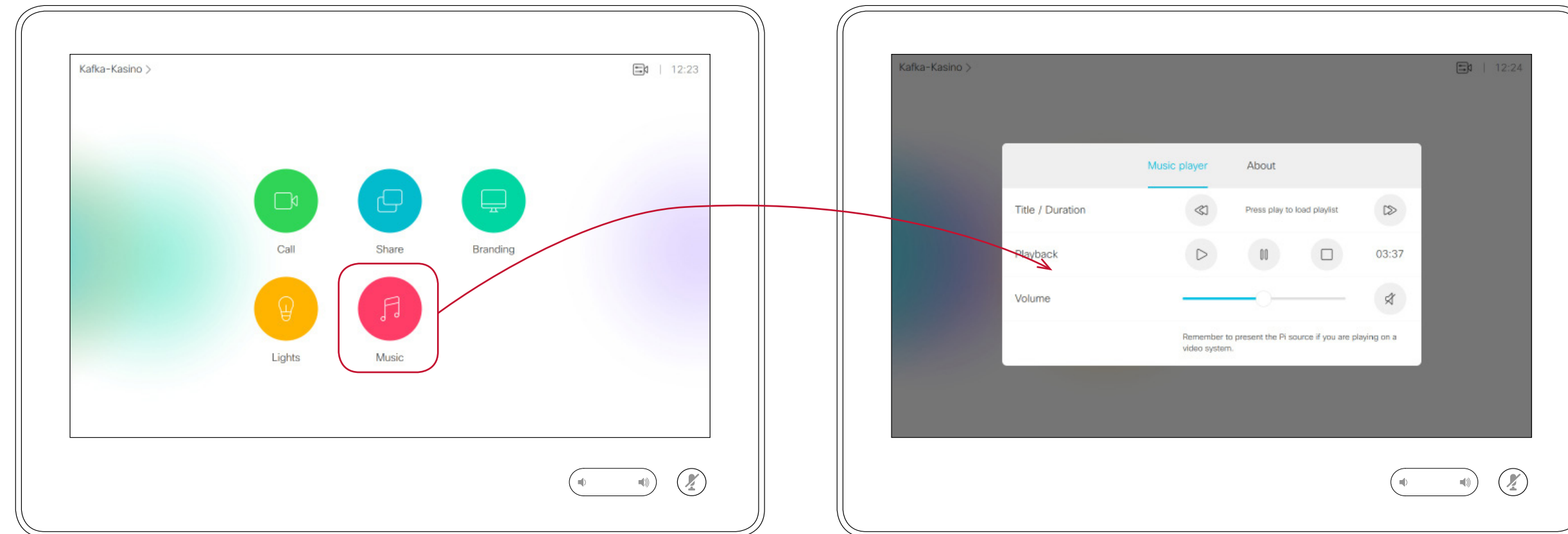
すべての実用的な目的のために、パネルの最大数はユーザビリティ要件、およびある程度のシステムリソースによって設定されることに注意してください。Touch10 / DX で導入する各ボタンは、それぞれ対応するパネルが必要になります。

すべてのボタンは、通常のコール ハンドリング ボタンとともに表示されるようになります。グローバル ボタンへのアクセスは、Touch10 / DX ディスプレイの最上部に位置しなくなります。

Touch10 / DX ディスプレイ上に十分なスペースが残っていない場合は、[その他 (More...)] ボタンが表示され、このボタンを使ってその他のボタンにアクセスできます。

つまり、合計 3 つのパネル セットを自由に使用できます。

- ・ [常時 (Always)] アイコン (ボタン) はいつでも表示されています。
- ・ [外部発信のみ (Out of call only)] アイコン (ボタン) は外部発信の場合にのみ表示されます。
- ・ [通話中のみ (In-Call only)] アイコン (ボタン) は通話中のみ表示されます。



Touch10 に表示される室内制御によって行えるカスタマイズの例。左の図に示されているアイコンをタップすると、右の図に示されているメニューが表示され、ミュージック プレイヤーの制御を行えます。

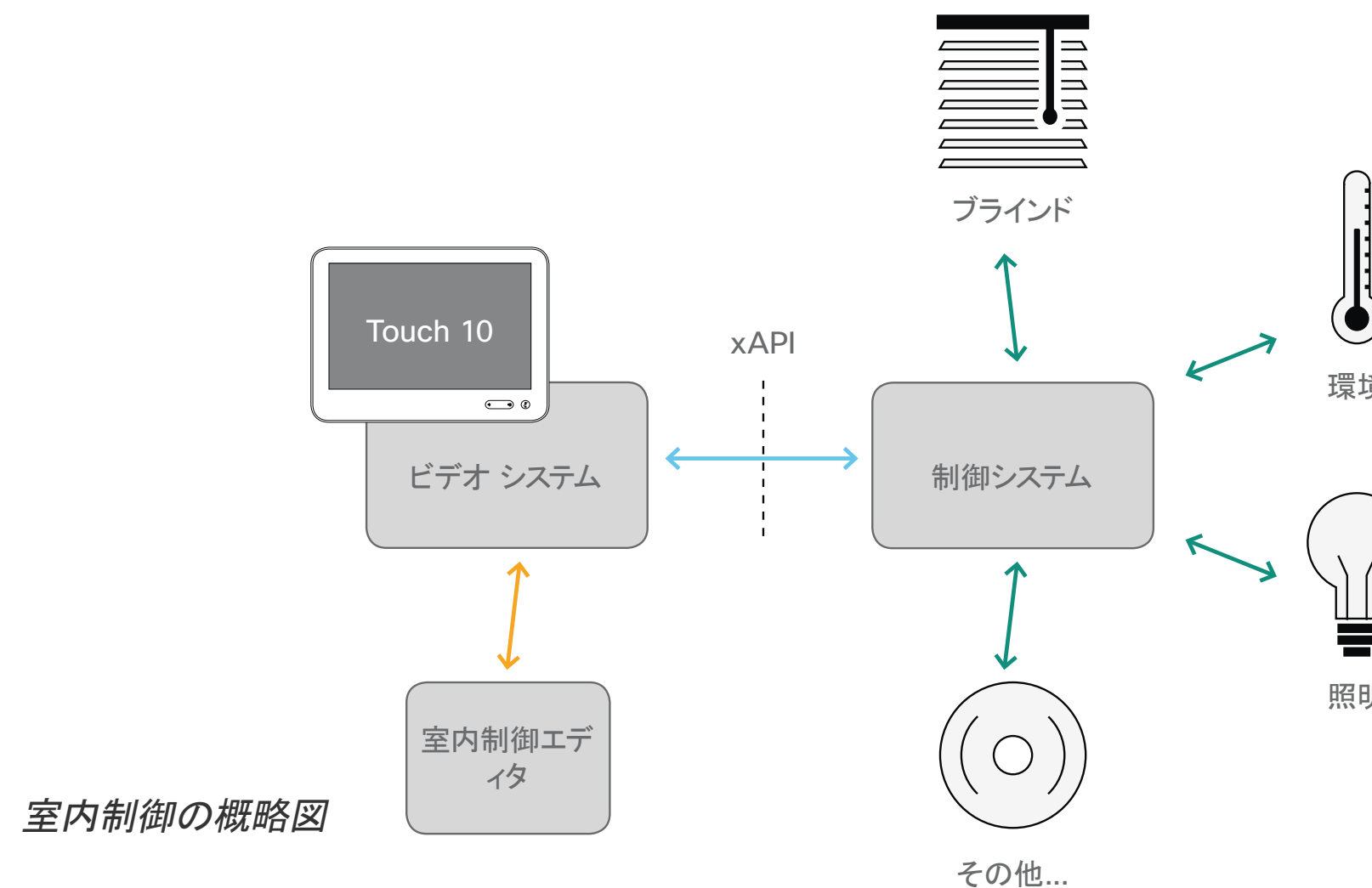
室内制御の機能を利用するには、シスコのビデオ システムに加え、Touch10/DX ユーザ インターフェイスとサードパーティ製の制御システム (例: Crestron、AMX など) が必要になります。

ビデオ システムの API は xAPI と呼ばれており、ビデオ システムと制御システム間のリンクとして機能します。制御システムをプログラムするときは、xAPI で公開されているイベントとコマンドを使用します。

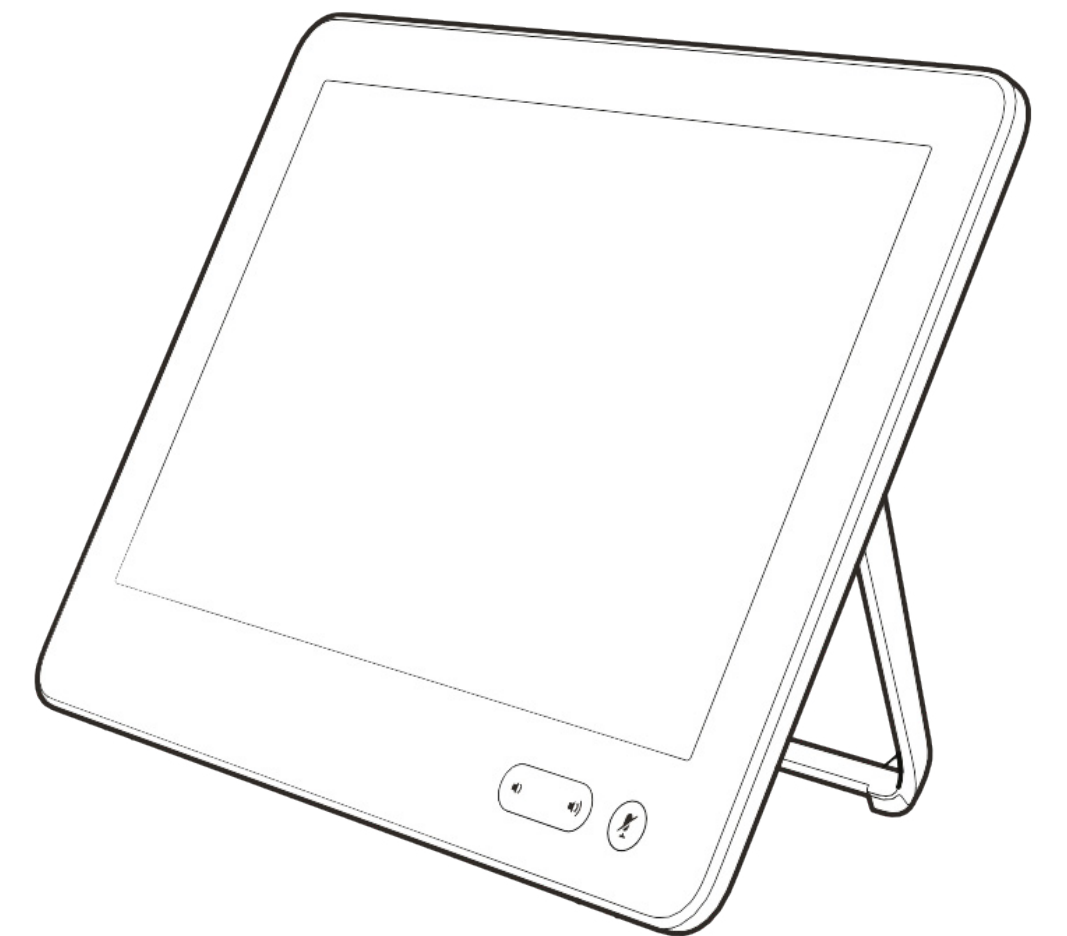
ドラッグアンドドロップで簡単に操作できる制御エディタには、ウィジェットと呼ばれるユーザ インターフェイス要素のライブラリが用意されています。これらのウィジェットを使用して、Touch10/DX ユーザ インターフェイス向けに独自の室内制御パネルを作成できます。

これらの機能により、制御システムの機能と使いやすい Touch10 / DX ユーザ インターフェイスを高度に連携させて利用することができます。

このドキュメントで示されているすべての例には Touch10 ユーザの場合のみを示しますが、2 つのインターフェイスは非常によく似ているため難しいことはありません。



ユーザ インターフェイスの作成



Touch10 のユーザ インターフェイスの作成

共有ファイルの形式

室内制御エディタを使用して、ビデオ システムの Touch10 / DX ユーザ インターフェイスで周辺機器を制御するためにカスタマイズしたパネルを作成します。

ビデオ システムへの接続

ビデオ システムにアクセスできる場合は、ビデオ システムの Web インターフェイスからエディタを起動できます。

室内制御パネルが Touch10/DX ですすでに作成されている場合は、パネルがエディタに自動的にロードされ、設計の開始点として使用できるようになります。

新しいパネルをビデオ システムにプッシュすると、すぐに結果が Touch10 / DX に反映されます。

オフライン

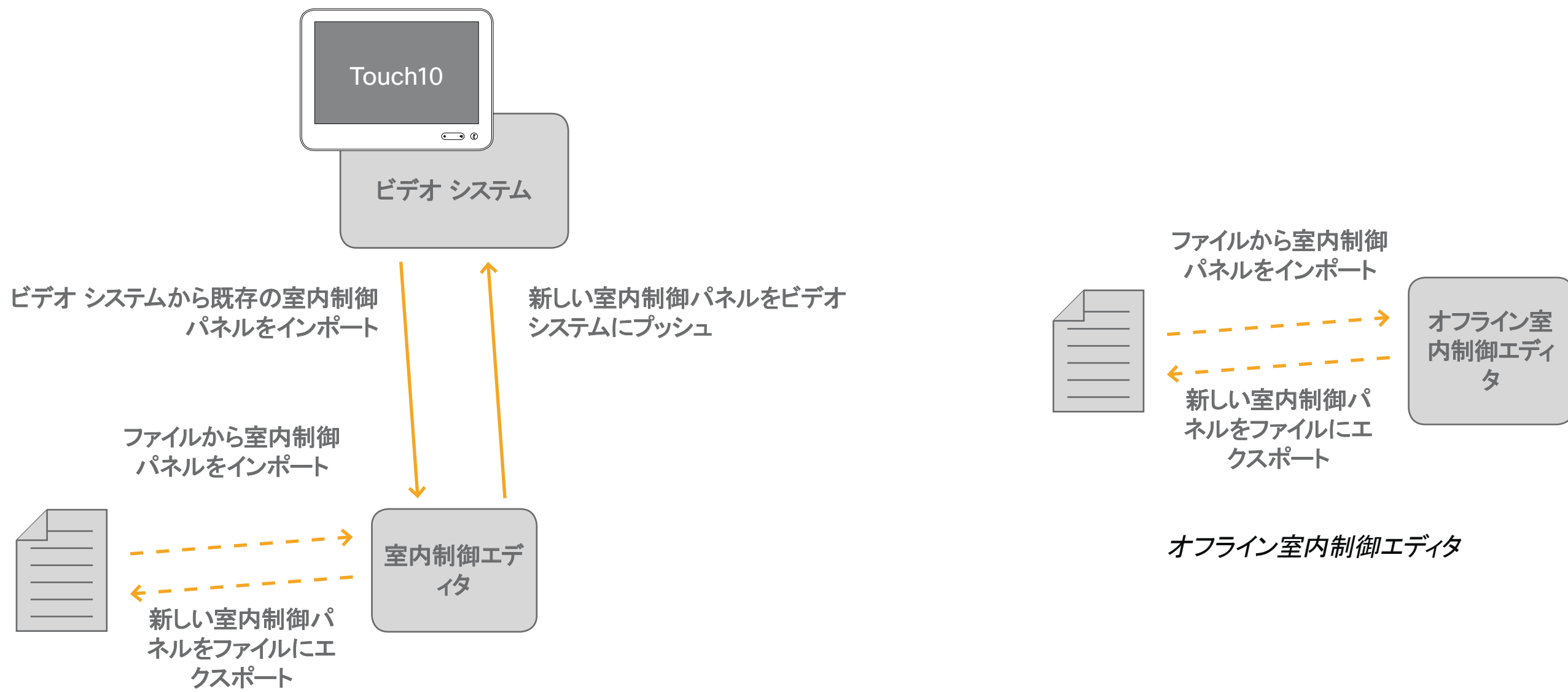
次の 2 つの場所から、オフライン エディタをダウンロードできます。

- ・ <http://www.cisco.com/go/in-room-control-docs> からダウンロード
- ・ または、ビデオ システムの Web インターフェイスに管理者クレデンシャルでサインインし、[統合(Integration)] > [室内制御(In-Room Control)] に移動して、[エディタのダウンロード(Download Editor)] をクリックします。

オフライン エディタをダウンロードする場合は、ダウンロードした zip ファイルからファイルを展開します。フォルダ構造は維持してください。

オフライン エディタを使用するときは、ビデオ システムおよび Touch10 / DX と直接通信するのではなく、ファイルを操作します。この点を除くと、オフライン エディタは完全な機能を備えています。

ビデオ システムの Web インターフェイスから起動したエディタとオフライン エディタは、同じファイル形式を共有しているため、一方のエディタで作成したファイルを、もう一方のエディタで開いて変更することができます。



ビデオ システムの Web インターフェイスから起動される室内制御エディタ

パート 1: 室内制御 > ユーザ インターフェイスの作成 室内制御エディタの起動

ビデオ システムの Web インターフェイスに管理者クレデンシャルでサインインし、[統合 (Integration)] > [室内制御 (In-Room Control)] に移動して、[エディタの起動 (Launch Editor)] をクリックします。

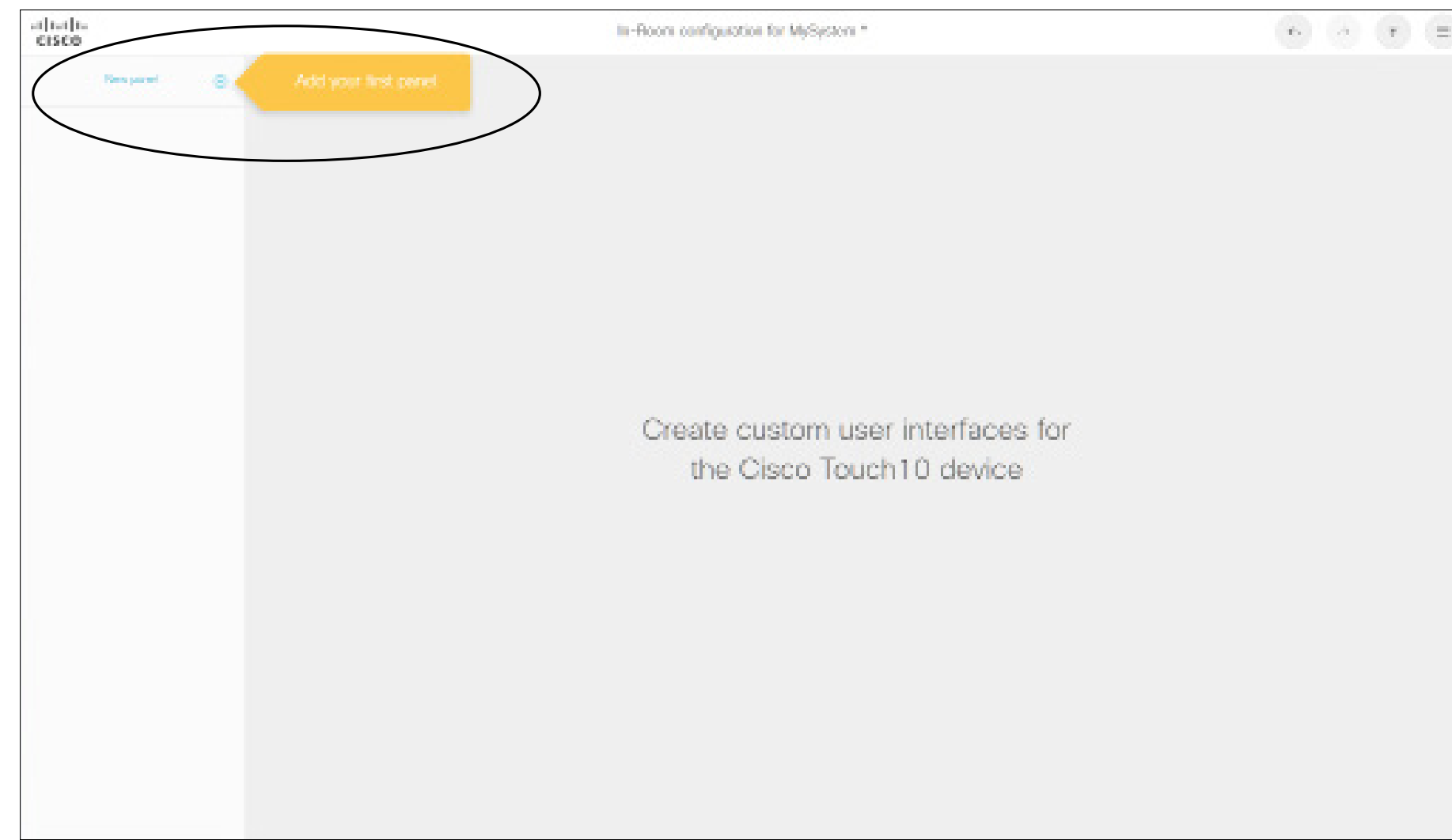
ビデオ システムに既存の室内制御パネルのセットがある場合は、それらがエディタに自動的にロードされます。パネルがすでに作成されている場合、[作成 (Create)] アイコン (下の例を参照) の代わりに [編集 (Edit)] アイコンが表示されます。

オフラインバージョン。 オフラインバージョンのエディタを使用している場合は、ブラウザを使用して **rceditor** フォルダにある **index.html** ファイルを開きます。

定義されているパネルがない場合、ユーザ インターフェイスは右の図のようになります。

[最初のパネルを追加してください (Add your first panel)] というテキストが指している ⊕ をクリックします。

最初のパネルのパネルタイプは **ホーム** になります (詳細については、前のページを参照してください)。この設定は変更できます (詳細については、次のページを参照してください)。



パート 1: 室内制御 > ユーザ インターフェイスの作成 室内制御エディタについて

(パネル プロパティ内の)パネル位置を使用すると、シーケンス内の特定パネルの位置を指定します(トップからボトムまで)。他のパネルはそれに応じて移動します。これにより、Touch10 のボタン表示順が決定されます。

パネル ボタン名、色およびアイコンがここに表示されます。名前を変更するには、テキストをクリックして編集可能にします。色とアイコンを指定するには、プロパティパネルを使用します。

ここから新しいパネルを作成します。

現在の設定をプレビューします。次のページも参照してください。

ここをクリックして、ページのプロパティにアクセスします。

元に戻す、やり直し、ビデオシステムへの設定のエクスポート。

その他のオプション

ファイルからインポートするときは、インポートおよびマージから選択します。マージを行うと、現在のパネル セットにパネルが追加されます。同じ名前の任意のパネルは、上書きされます。

これでウィジェットの幅を制御します。使用できる幅サイズは、ウィジェットのタイプによって異なります。

ID は、ウィジェットを参照するために使用されます。

テキストは、Touch10 に表示されるテキストです。

プロパティパネルには、黄色のフレームで選択またはハイライトした部分の設定が表示されます。選択肢は、[パネル (Panel)]、[ページ (Page)]、[行 (Row)] または [ウィジェット (Widget)] です。

ヒント: [元に戻す (Undo)] と [やり直し (Redo)] のアイコンに加えて、使い慣れたキーボードのショートカット コマンドによるコピーと貼り付けも使用できます (CTRL C と CTRL V または CMD C と CMD V)。

ウィジェットをページにドラッグして、挿入します。

パネルのプロパティ

パネル ID。

パネル順を変更するにはこれを使用します。詳細については左上にあるテキストを参照してください。

パネルが利用可能になるタイミングを指定します。

ここで選択するアイコンは、Touch10/DX デイスプレィ上のパネルの [室内制御 (In-Room Control)] ボタンに表示されるアイコンになります。

Touch10/DX の画面に表示される室内制御ボタンの色を指定することができます。標準ボタンに対して使用される限定的なカラーパレットがエディタで使用可能です。色の選択時、次に示すように、Cisco が使用している色についてコンテキストの小さい詳細が表示されます。

[スペーサ (Spacer)] は、レイアウトを調節するのに役立ちます。

[方向パッド (Directional pad)] は、Apple TVなどを制御するために使用できます。

上記のウィジェット ID は、(青色のテキストで示される)アクティブパネルに属しています。

パネルのページに入力し始めると、使用中のウィジェット ID が上に表示され、簡単な概要を確認できます。

パネルの最大数は 20 に増加しました。実際の制限はユーザビリティ要件、およびある程度のシステムリソースによって設定されます。Touch10/DX で導入する各ボタンは、それぞれ対応するパネルが必要になります。パネルは、次の 3 つのグループのいずれかに属します。

- In-call only (通話中のみ表示されます)
- Outside calls only (外部コールのみ表示されます)
- Always (つねに表示されます)

Touch10 パネル(または DX の画面)が対応できるよりも多いパネル(ボタンなど)を作成する場合、はみ出したボタンへのアクセスを提供する、[More...] と呼ばれるボタンが作成されます。

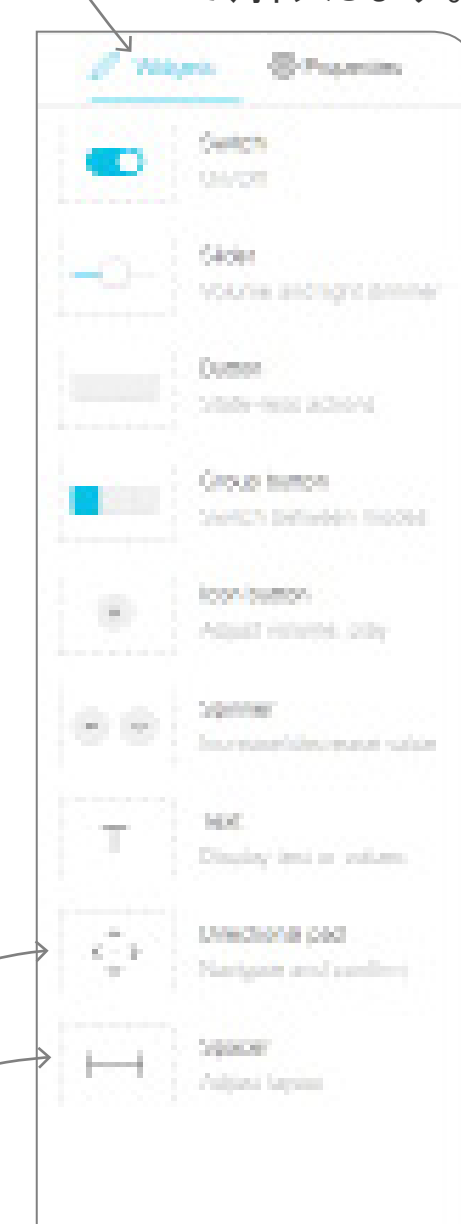
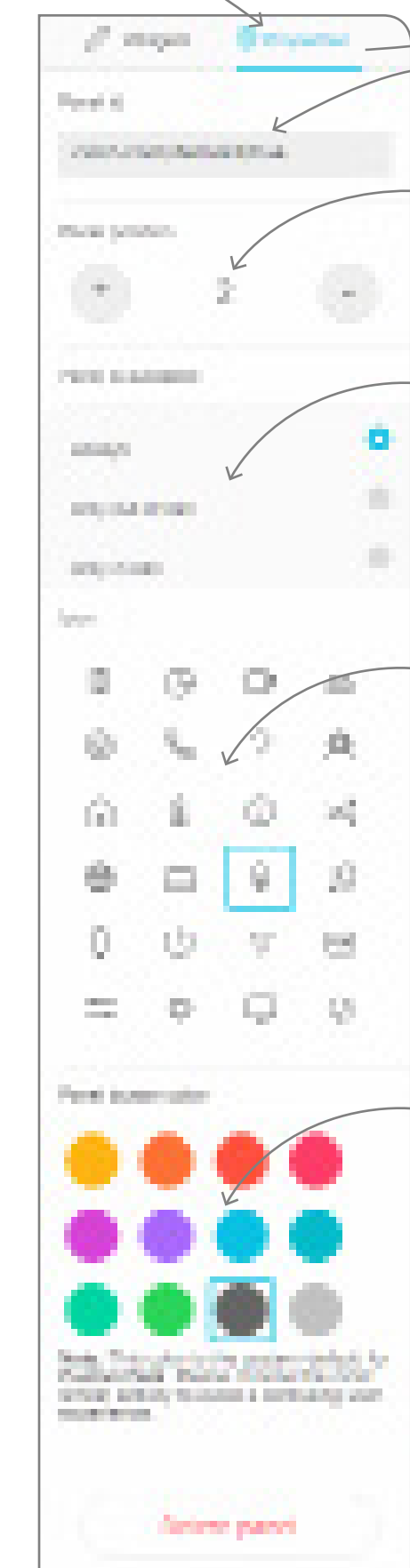
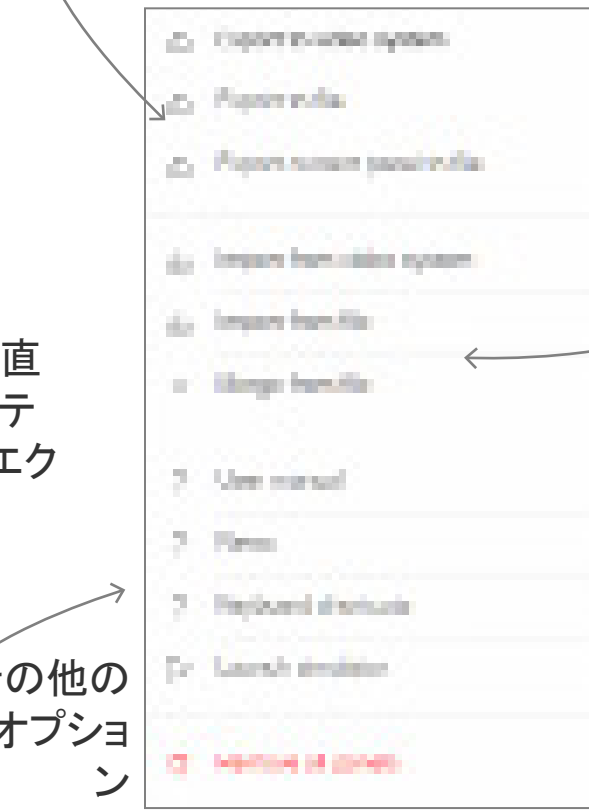
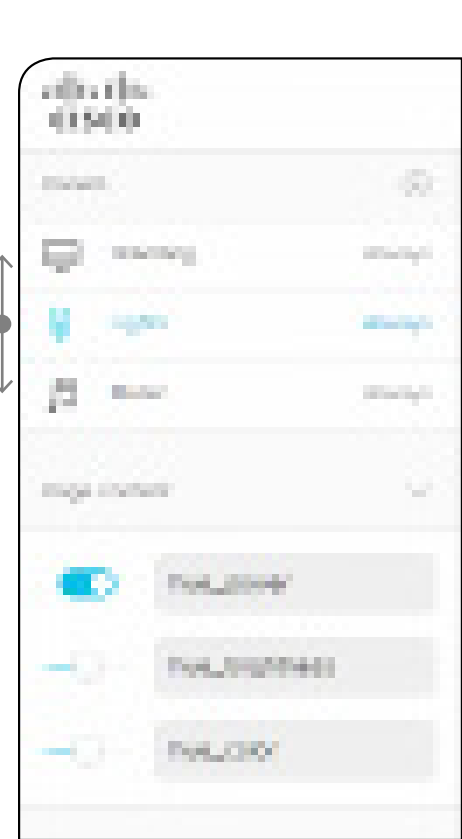
室内制御パネルは複数のページで構成可能です。各ページは 1 つ以上の行で構成され、テキストや、ウィジェットと呼ばれるユーザ インターフェイス要素を入力できます。

パネルごとの最大ページ数は 50 です。

ウィジェットは 4 列のグリッドに配置されます。ウィジェットは次のルールに従ってグリッドに配置されます。

- ウィジェットはサイズに応じて、1 ~ 4 列を占めます。
- 行は右揃えになります。
- 1 行に収まらない数のウィジェットを追加すると、ウィジェットは同じ行内で折り返されます。

ヒント: イベントを作成するがページを表示しないボタンが必要な場合があります。新しいパネルを作成し、パネル ID を指定した後、[ページの削除 (Delete page)] をクリックすると、パネルにページが含まれなくなります。パネル ID はイベントを生成するために必要です。この動作を使用する典型的な例として、スピードダイヤル ボタンやワンボタン(OBTP)があります。

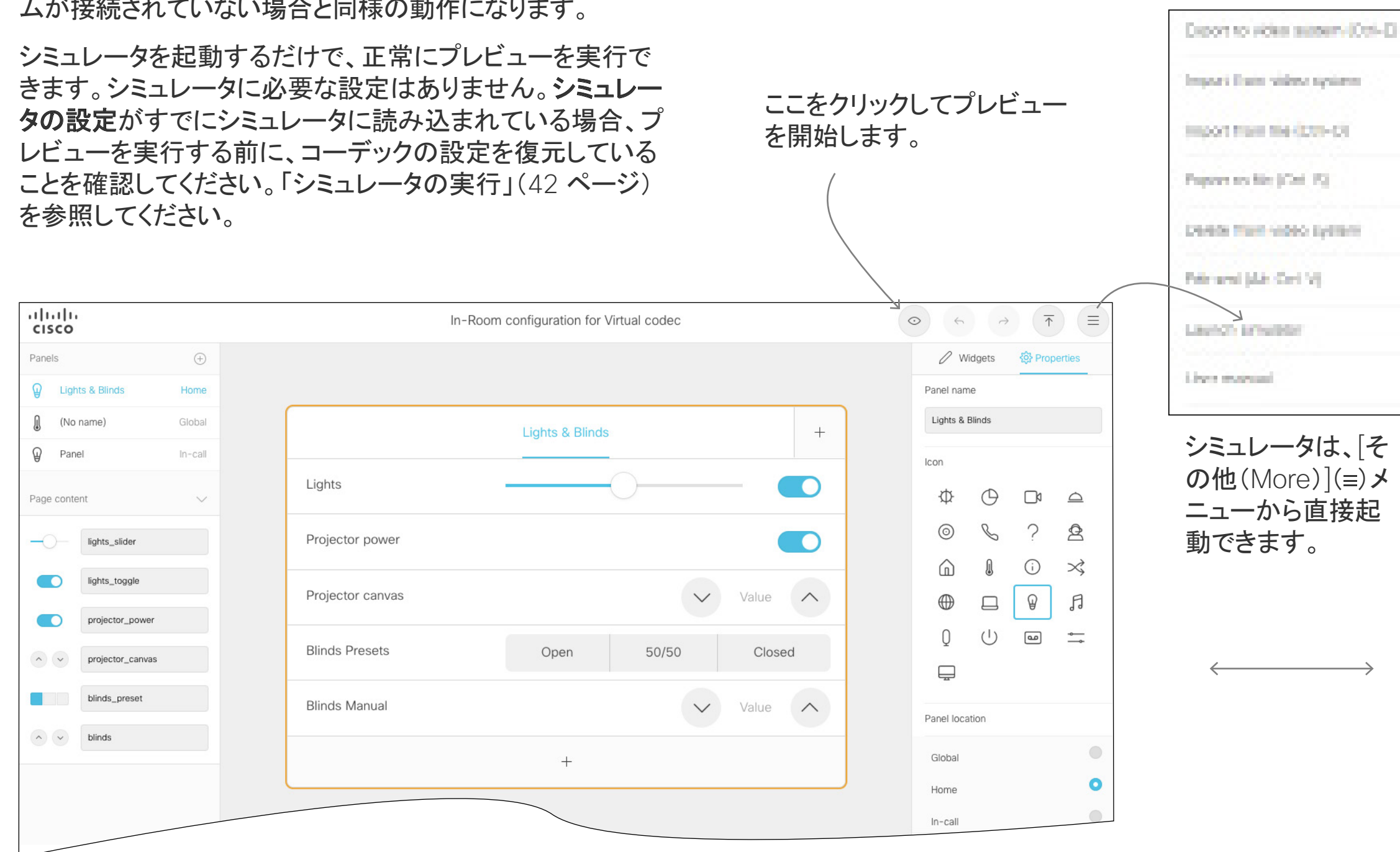


パート 1: 室内制御 > ユーザ インターフェイスの作成 現在の設定のプレビュー

設定を展開する前に、プレビューして確認することができます。

注: 実際の設定どおりのプレビューで確認するには、ここで説明されているプレビューを実行する前にシミュレータを起動しておく必要があります。そうしないと、Touch10 は制御システムが接続されていない場合と同様の動作になります。

シミュレータを起動するだけで、正常にプレビューを実行できます。シミュレータに必要な設定はありません。シミュレータの設定がすでにシミュレータに読み込まれている場合、プレビューを実行する前に、コーデックの設定を復元していることを確認してください。「シミュレータの実行」(42 ページ)を参照してください。



シミュレータは、[その他(More)](≡)メニューから直接起動できます。



上記の図は、シミュレートされたサードパーティ製の制御システムが接続されている、シミュレートされたビューを示しています。

設定を実装する際(実際の状況シナリオ)、可能な場合は必ず制御システムが SetValue コマンドを送信するように設定してください。

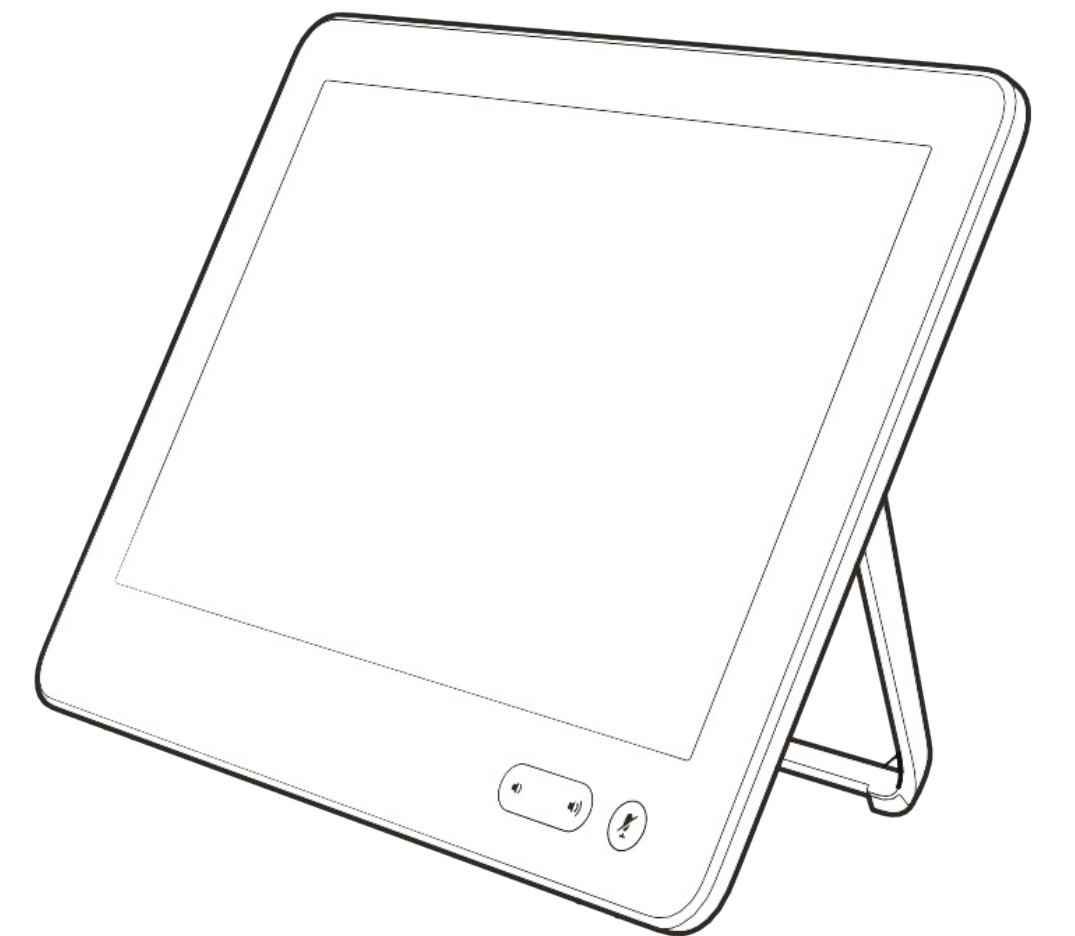
例: 実際の状況で [照明(Lights)] を [オン(On)] に設定する場合、Touch10 は照明が実際にオンに切り替えられたことを確認するフィードバックを受信する必要があります。これが行われるようにするには、コントローラは照明をオンにしてから SetValue を送信して、照明の設定が変更されたことを確認する必要があります。上記の例の右側のペインには、Touch10 が制御システムに送信した内容と、それに対して制御システムが Touch10 に返信した内容を示すシミュレーションが示されています。

実際の状況では、誰かが会議室の壁の照明スイッチを操作したときにはいつでも、制御システムが SetValue を Touch10 に送信していることを確認する必要があります。

注: このプレビューを実行する前にシミュレータをアクティブにしないと、Touch10 が SetValue コマンドを受信しない場合と同様に動作します。このフィードバックを受信しないと、Touch10 の [照明(Lights)] ボタンはまもなく [オフ(Off)] 設定に戻ります。

これに関して詳しくは、次のページを参照してください。

アプリケーション プログラム インターフェイス (API)



室内制御のプログラミング用 API

ビデオ システムへの接続

ビデオ システムの API (xAPI と呼ばれます) は、AMX や Crestron などのサードパーティ製の制御システムとの双方向通信を可能にします。次のような複数の方法で xAPI にアクセスできます。

- [Telnet]
- SSH
- HTTP/HTTPS
- RS-232/シリアル接続

どの方法を選択しても、xAPI の構造は同じです。使用するアプリケーションおよびビデオ システムに最適なアクセス方式を選択してください。

使用可能なアクセス方式と xAPI の使用方法の詳細については、使用するビデオ システムの API ガイドを参照してください。

次にアクセスしてください。

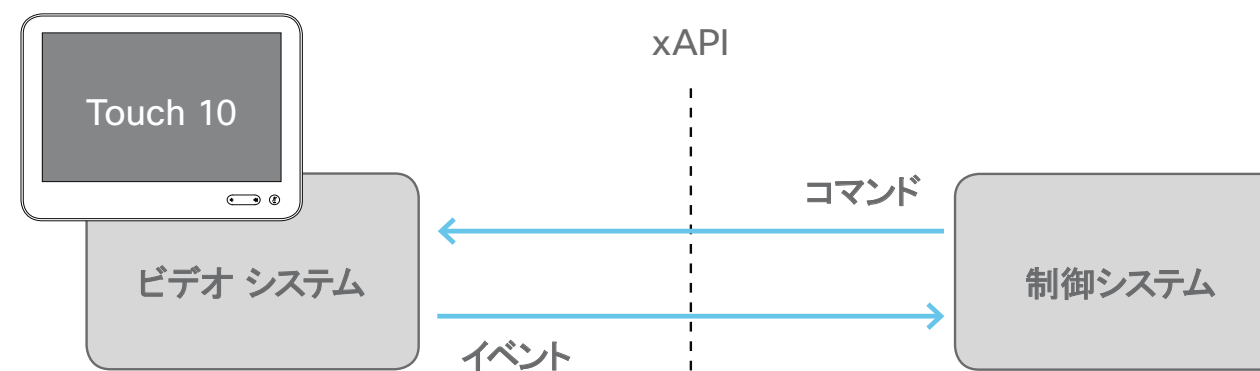
- <http://www.cisco.com/go/sx-docs> (SX シリーズの場合)
- <http://www.cisco.com/go/mx-docs> (MX シリーズの場合)
- <http://www.cisco.com/go/dx-docs> (DX シリーズの場合)

その後、[リファレンス ガイド (Reference Guides)] > [コマンド リファレンス (Command References)] をクリックして API ガイドを見つけます。

API を介した通信

ビデオ システムと制御システムは xAPI を介してメッセージを交換し、Touch10/DX 室内制御パネルが室内の実際の状態を常に反映するようにします。

ビデオ システムは、ユーザが Touch10/DX の室内制御パネルでいずれかのコントロールを使用するときに、1 つ以上のイベントを送信し、制御システムは、室内の設定が変更されたときにビデオ システムにコマンドを送信する必要があります。



ビデオ システムと制御システムは、xAPI を介してメッセージを交換します。

例:

- ユーザが Touch10/DX の [点灯 (Lights On)] ボタンをタップすると、ビデオ システムによって関連イベントが送信されます。制御システムは、これらのイベントに応答するために室内の照明をオンにして、対応するコマンドをビデオ システムに送り返す必要があります。
- ユーザが室内の照明をオンにした場合、制御システムはビデオ システムにコマンドを送信し、ビデオ システムが Touch10/DX 室内制御パネルを更新して照明がオンであることを反映できるようにする必要があります。

室内制御に関連するすべてのイベント、コマンド、およびステータスの概要については、「[Command reference](#)」の章を参照してください。

ビデオ システムと制御システムのペアリング

ビデオ システムに接続されている周辺機器として、制御システムを登録できます。

xCommand Peripherals Connect ID: "ID" Type: ControlSystem

ここで、ID は 制御システムに固有の ID です (通常は MAC アドレス)。

このコマンドと、コマンド オプションの詳細については、API ガイドを参照してください。

ハートビート。 制御システムは、ビデオ システムにハートビートを送信して、制御システムが接続されていることをビデオ システムに知らせる必要があります。ビデオ システムが制御システムからこれらのハートビートを受信している限り、その制御システムは、接続済みデバイス リスト (xStatus Peripherals ConnectedDevice を参照) にとどまります。

xCommand Peripherals HeartBeat ID: "ID" [Timeout: Timeout]

ここで、ID は制御システムに固有の ID (通常は MAC アドレス) で、Timeout は各ハートビート間の秒数です。Timeout が指定されていない場合は、60 秒に設定されます。

注: 接続されたユニットがハートビート送信を停止した場合に、ビデオ システムがハートビートの欠落を検出するまでに少しの時間 (最大 2、3 分) が経過します。

逆の場合も同様で、新たなハートビートがコーデックによって検出されるまでに最大で数分経過することがあります。

室内制御のプログラミング用 API(続き)

ウィジェット アクションのイベント

ビデオ システムは、ユーザが Touch10/DX 室内制御パネル上のコントロールを使用するときに、次のような 1 つ以上のイベントを送信します。

- Pressed:ウィジェットを最初に押したときに送信されます。
- Changed:ウィジェットの値を変更したときに送信されます(トグルボタンとスライダにのみ適用されます)。
- Released:ウィジェットから指を離れたときに送信されます(ウィジェット外に指を移動してから離れたときも送信されます)。
- Clicked:ウィジェットをクリックしたとき(ウィジェットを押してからウィジェット外に指を移動しないで離れたとき)に送信されます。

これらのイベントは、次の 2 つの方式で送信されます。

- UserInterface Extensions Event:端末出力モードに適しています。
- UserInterface Extensions Widget:XML 出力モードに適しています。

登録する制御システムに最適な方式を確認するには、右の表を参照してください。

これらのイベントがトリガーされるウィジェット(ユーザ インターフェイス要素)と、そのタイミングについては「[Widgets](#)」の章で説明しています。

UserInterface Extensions Event(端末出力モード向け)	UserInterface Extensions Widget(XML 出力モード向け)
単一の文字列に、アクションのタイプ、イベントをトリガーしたウィジェット(ウィジェット ID で識別されます)、およびウィジェットの値に関する情報が含まれます。	アクションのタイプ、イベントをトリガーしたウィジェット(ウィジェット ID で識別されます)、およびウィジェットの値は、XML ツリー内の別々の要素として含まれます。
<p>登録方法:</p> <pre>xfeedback register event/UserInterface/Extensions/Event</pre> <p>例:</p> <pre>*e UserInterface Extensions Event Pressed Signal: "<u>WidgetId</u>Value" ** end *e UserInterface Extensions Event Changed Signal: "<u>WidgetId</u>Value" ** end *e UserInterface Extensions Event Released Signal: "<u>WidgetId</u>Value" ** end *e UserInterface Extensions Event Clicked Signal: "<u>WidgetId</u>Value" ** end</pre>	<p>登録方法:</p> <pre>xfeedback register event/UserInterface/Extensions/Widget</pre> <p>例:</p> <pre><Event> <UserInterface item="1"> <Extensions item="1"> <Widget item="1"> <Action item="1"> <WidgetId item="1"><u>WidgetId</u></WidgetId> <Value item="1">Value</Value> <Type item="1"><u>Type</u></Type> </Action> </Widget> </Extensions> </UserInterface> </Event></pre>

制御システムが登録できる 2 つのイベント方式:それぞれ、端末出力モードと XML 出力モードに適しています。

室内制御のプログラミング用 API(続き)

パネル更新のイベント

新しい室内制御パネルが適用されると、ビデオ システムは次のイベントを送信します。

LayoutUpdated: Touch10/10 の新しい室内制御パネルがビデオ システムにエクスポートされたときに送信されます。

このイベントへの応答として、制御システムは、すべてのウィジェットを初期化するコマンドを送信して、室内設定の実際の状態を反映させる必要があります。

登録方法:

```
xfeedback register event/UserInterface/Extensions/Widget/
  LayoutUpdated
```

例:

端末出力モード:

```
*e UserInterface Extensions Widget LayoutUpdated
** end
```

XML 出力モード:

```
<Event>
  <UserInterface item="1">
    <Extensions item="1">
      <Widget item="1">
        <LayoutUpdated item="1"/>
      </Widget>
    </Extensions>
  </UserInterface>
</Event>
```

ページを開く/閉じるときのイベント

それぞれのページに固有のページ ID を与えた場合は、ページの開/閉時にシステムがイベントを送信できます。

EventPageOpened: ページが開かれたときに送信されます

EventPageClosed: ページが閉じられたときに送信されます

ページの動作はラジオ ボタンと似ていて、別のページを開くと現在のページが閉じます。その場合は EventPageClosed と EventPageOpened の両方が発行されます。

登録方法:

```
xfeedback register event/UserInterface/Extensions/PageOpened
xfeedback register event/UserInterface/Extensions/PageClosed
```

例:

端末出力モード:

```
*e UserInterface Extensions Event PageOpened PageId: "appletvpage"
*e UserInterface Extensions Event PageClosed PageId: "appletvpage"
```

XML 出力モード:

```
<Event>
  <UserInterface item="1">
    <Extensions item="1">
      <Page item="1">
        <Action item="1">
          <PageId item="1">appletvpage</PageId>
          <Type item="1">Opened</Type>
        </Action>
      </Page>
    </Extensions>
  </UserInterface>
</Event>
```

PageClosed の例は、左の例の Opened を単に Closed に置き換えたものになります。このイベントを使用する標準的な方法は、イベントに基づきコントローラに何らかのアクションを実行させることです(この場合は AppleTV ボックスを自動的にオン/オフにする)。

室内制御のプログラミング用 API(続き)

コマンドとステータス

ウィジェットの値を設定する SetValue コマンドは、室内制御の操作に不可欠です。

```
xCommand UserInterface Extensions Widget SetValue Value:
Value WidgetId: WidgetId
```

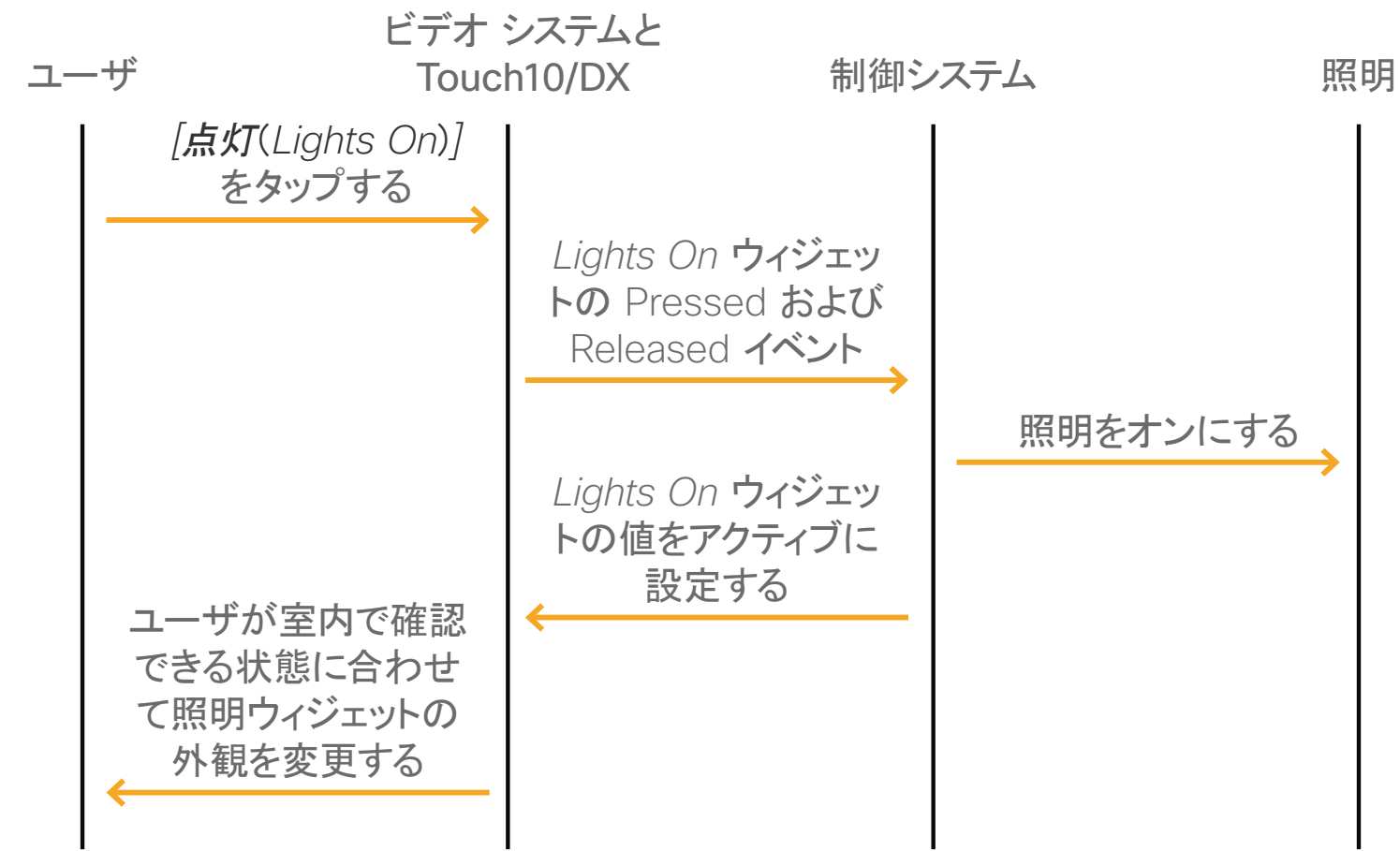
ビデオ システムが SetValue コマンドを受信すると、ビデオ システムのステータスと Touch10/DX 室内制御パネルがそれに応じて更新されます。

次のような状況では、制御システムが SetValue コマンドを送信することが重要です。それにより、Touch10/DX 室内制御パネルが室内の実際の状態を反映するようになります。

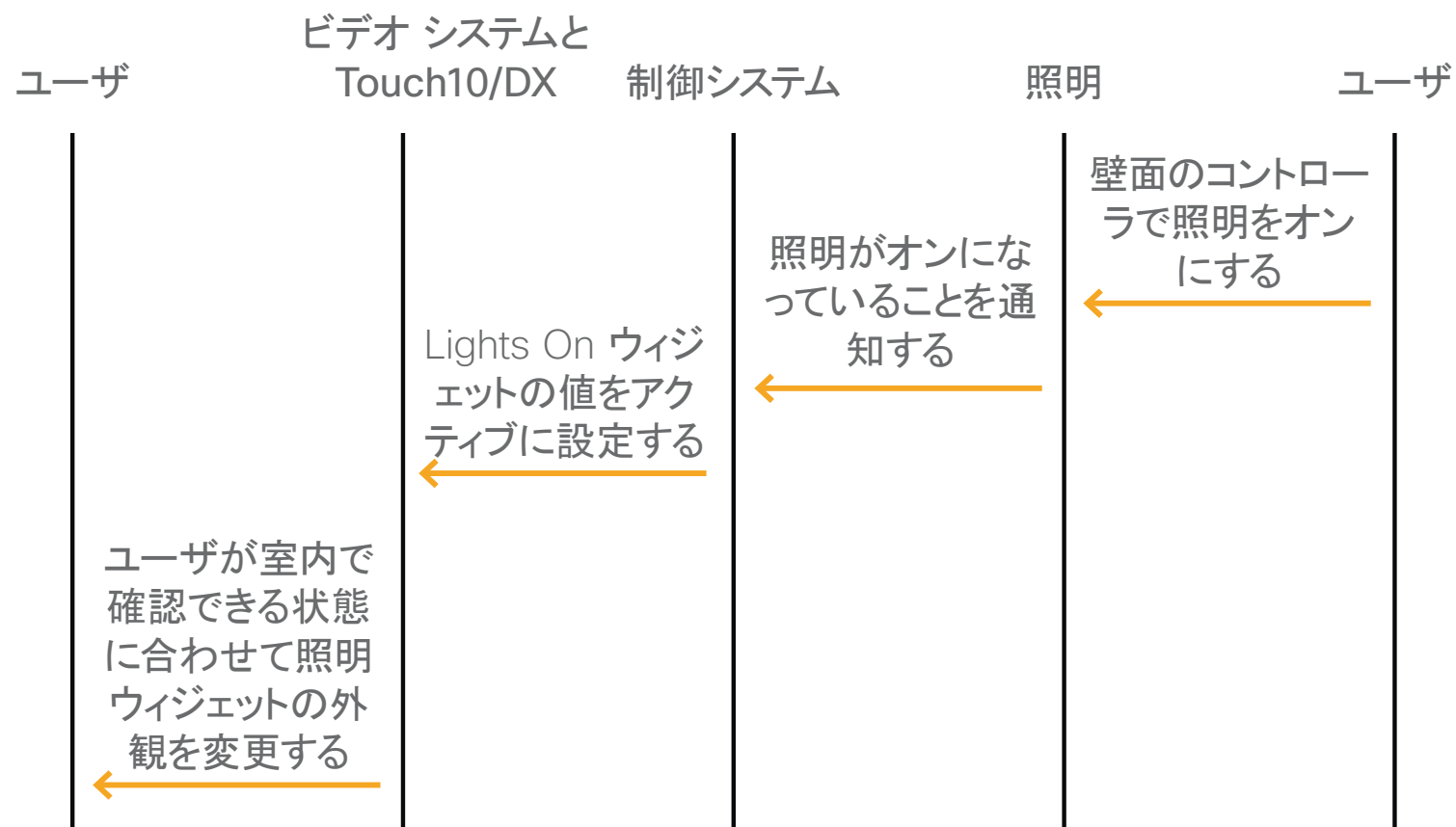
- ・ 制御システムがビデオ システムに最初に接続したとき。
- ・ ビデオ システムが再起動したとき。
- ・ 制御システムが再起動したとき。
- ・ 新しい室内制御パネルが(LayoutUpdated イベントへの応答として)室内制御エディタからビデオ システムにエクスポートされたとき。
- ・ 室内にある制御対象の機器の状態が物理的に変更されたとき(壁面のコントローラを使用して照明をオンにするなど)。
- ・ イベントへの応答時。たとえば、ユーザがTouch10/DX 室内制御パネルの [点灯(Lights On)] ボタンをタップしたとき。
- ・ さらに、制御システムは、室内で必要なすべての操作を実行し、Touch10/DX 室内制御パネルでのアクション(実際に照明をオンにするなど)を反映する必要があります。

さまざまなウィジェット(ユーザ インターフェイス要素)に適用されるコマンドの詳細については、「ウィジェット」の章を参照してください。Widgets。

例

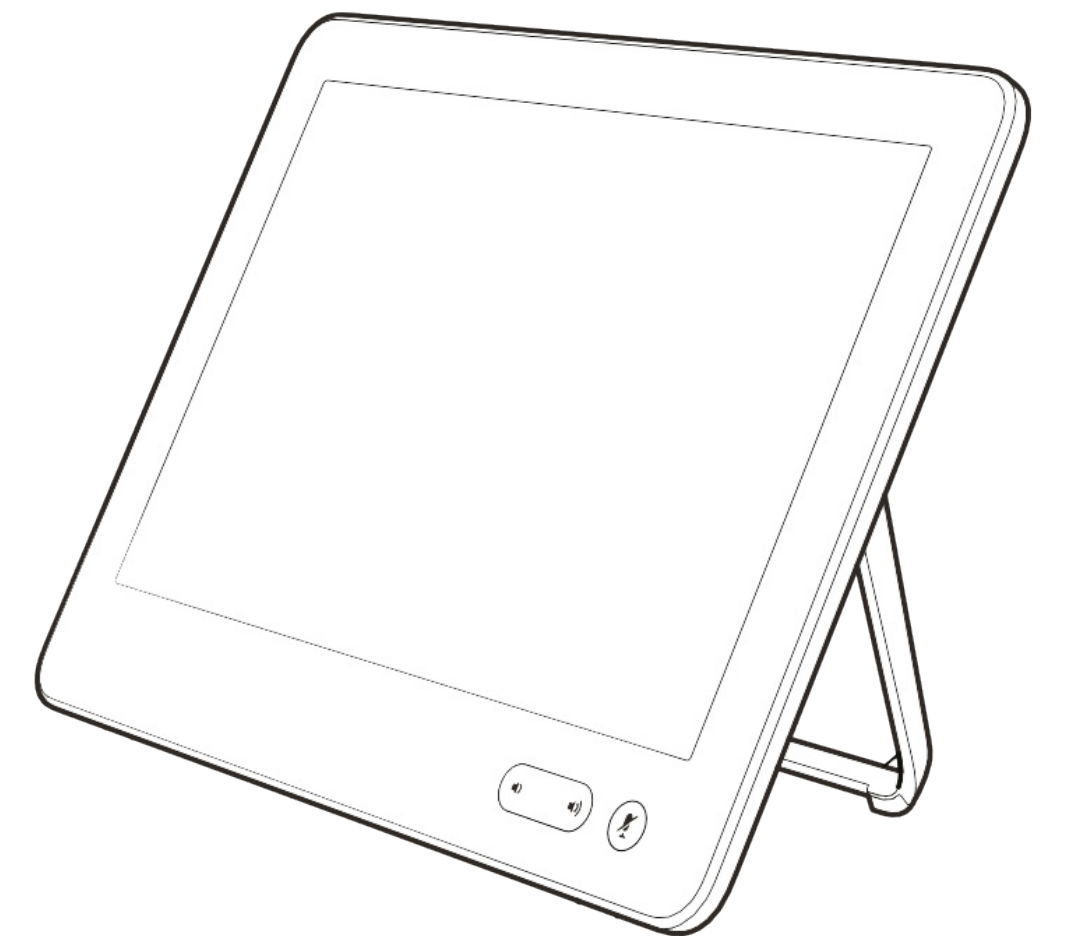


メッセージ フロー: Touch10/DX のコントロールを使用して照明をオンにする



メッセージ フロー: 壁面のコントローラを使用して照明をオンにする

ウィジェット



ウィジェットについて

Touch10/DX 室内制御パネルは、ウィジェットと呼ばれるユーザ インターフェイス要素で構成されています。室内制御エディタの右ペインに、完全なウィジェット ライブラリがあります。

[全般 (General)] タブ: カスタム テキスト付きのボタン、グループ ボタン、トグル ボタン、スライダ、テキスト フィールドなど。

[アイコン (Icons)] タブ: ホーム、電源、上下左右の矢印、カメラ コントロール、スピーカー コントロール、マイク コントロール、メディア プレーヤー コントロール など、一般的な記号が表示されたボタン。

利用可能な各ウィジェット タイプについては、後続のページで以下の点に重点を置いて説明します。

- ・ ウィジェットの値を変更するコマンド
- ・ 送信されるイベント (Pressed、Changed、Released、Clicked) と、これらのイベントをトリガーするアクション
- ・ 端末出力モードと XML 出力モードでのコマンドとイベントの例。

室内制御 (ユーザ インターフェイス拡張) に関するすべてのイベント、コマンド、およびステータスの構文とセマンティクスは、「Command reference」の章で説明されています。

ウィジェットの識別子

Touch10/DX 室内制御パネル上のすべてのウィジェットには、固有識別子 (ウィジェット ID) が必要です。ウィジェット ID はユーザが定義するか、自動的に割り当てることができます。ウィジェット ID には任意の名前や番号を使用できます。特殊文字を含めずにわかりやすい名前を使用することをお勧めします。最大文字数は 255 です。

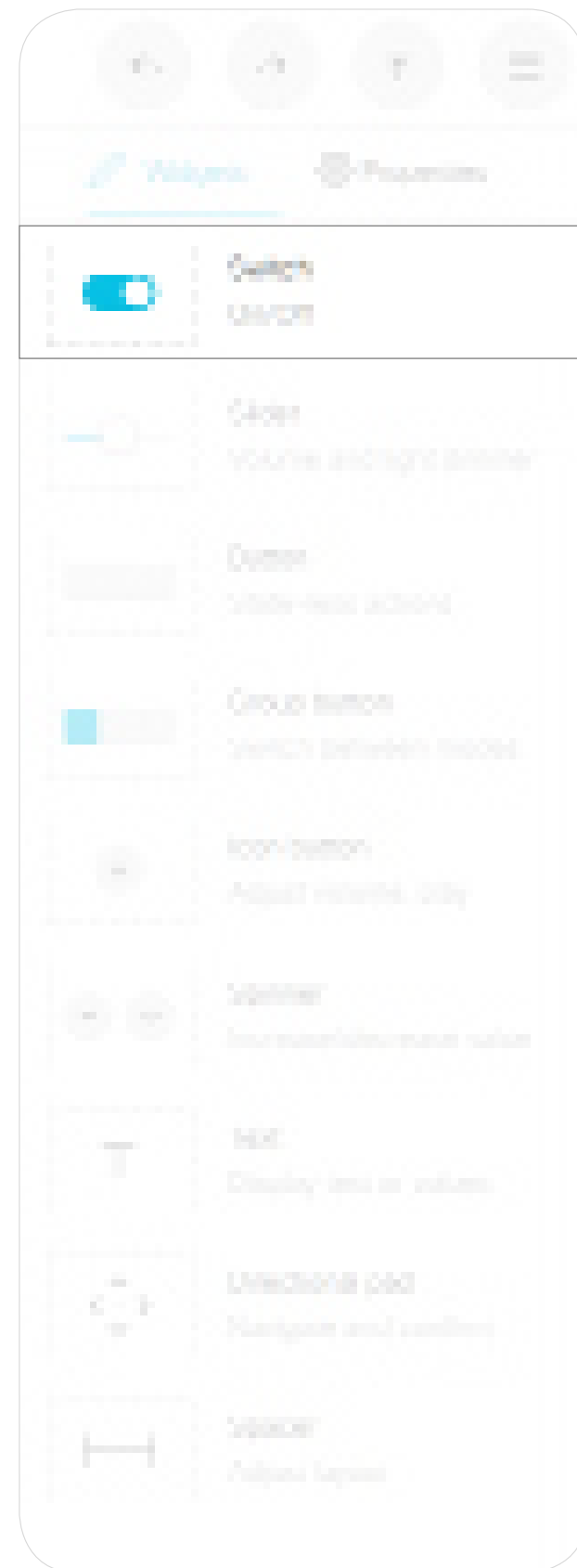
ウィジェット ID は、Touch10/DX (ビデオ システム) と制御システムの間をプログラムでリンクするために使用されます。ウィジェット ID は、ウィジェットに関連付けられたすべてのイベントに含まれます。制御システム用に記述したコードを使用してそのウィジェットにコマンドを送信するときは、同じ ID を使用する必要があります。

グループの識別子

ウィジェットの 1 つであるグループ ボタンには、2 種類の識別子があります。ウィジェット ID はボタンのグループ全体を表します。グループ ID はグループ内にある個々のボタンの固有識別子です。



グループ ID は自動的に割り当てられますが、ユーザが定義することもできます。グループ ID には任意の名前や番号を使用できます。特殊文字を含めずにわかりやすい名前を使用することをお勧めします。最大文字数は 255 です。



2つの状態を切り替えるスイッチで、オンまたはオフのいずれかを示します。

Event

Changed: ボタンを離すとトリガーされます。

値: <on/off>

コマンド

ボタンの外観は、タップするとすぐに変化します。ただし、制御システムは、ボタンの *on* と *off* が切り替わるときに、常に SetValue コマンドをビデオシステムに送信する必要があります。これにより、ステータスが確実に更新されます。

例: WidgetId = "togglebutton" のスイッチで "on" を押します。

端末モード

```
*e UserInterface Extensions Event Changed Signal: "togglebutton:on"  
** end
```

XML モード

```
<Event>  
  <UserInterface item="1">  
    <Extensions item="1">  
      <Widget item="1">  
        <Action item="1">  
          <WidgetId item="1">togglebutton</WidgetId>  
          <Value item="1">on</Value>  
          <Type item="1">changed</Type>  
        </Action>  
      </Widget>  
    </Extensions>  
  </UserInterface>  
</Event>
```

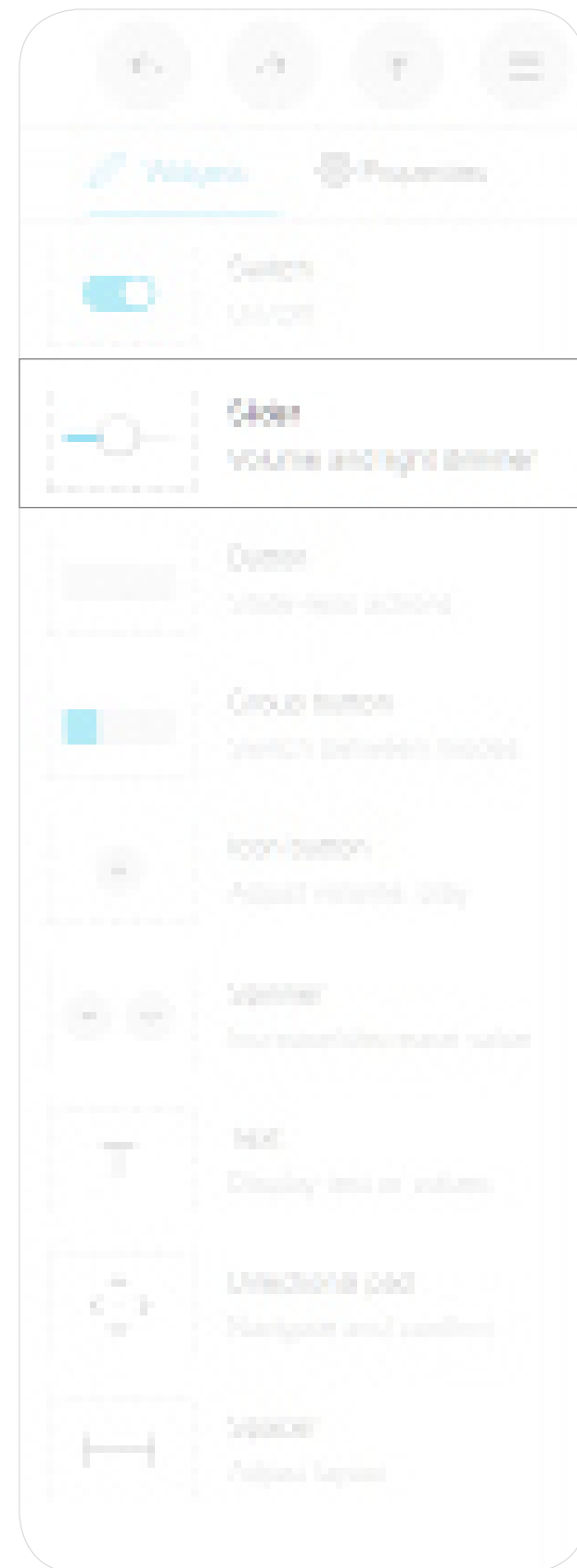
例: WidgetId = "togglebutton" のボタンを "on" に設定します。

```
xCommand UserInterface Extensions Widget SetValue WidgetId: "togglebutton" Value: "on"
```

スイッチとは、2つの状態を切り替えるボタンで、オンまたはオフのいずれかを示します。

使用例: 照明、ファン、プロジェクターなど、オンまたはオフに切り替えることができる機器。

明るさを調整する照明には、スライダと併用するトグル ボタンとして使用することもできます。



Event

- Pressed スライダを押すとトリガーされます。
値: 該当なし
- Changed スライダを押したまま移動させて、スライダを離れたときにトリガーされます。
値: 0 ~ 255
- Released スライダを離すとトリガーされます。
値: 0 ~ 255

例: WidgetId = "slider" のスライダを押したまま新しい位置 ("68") に移動させて、スライダを離します。

端末モード

```
*e UserInterface Extensions Event Pressed Signal: "slider"
** end
*e UserInterface Extensions Event Changed Signal: "slider:32"
** end
*e UserInterface Extensions Event Changed Signal: "slider:68"
** end
*e UserInterface Extensions Event Released Signal: "slider:68"
** end
```

XML モード

```
<Event>
  <UserInterface item="1">
    <Extensions item="1">
      <Widget item="1">
        <Action item="1">
          <WidgetId item="1">slider</WidgetId>
          <Value item="1">68</Value>
          <Type item="1">released</Type>
        </Action>
      </Widget>
    </Extensions>
  </UserInterface>
</Event>
```

スライダを使用すると、設定範囲内の値を選択できます。最小値は 0、最大値は 255 で表されます。スライダを押したまま移動させると、1 秒間に最大 5 回のイベントが送信されます。

バーをタップすると、スライダはすぐにその新しい位置に移動します。

使用例: 明るさを調整できる照明、音量コントロール。

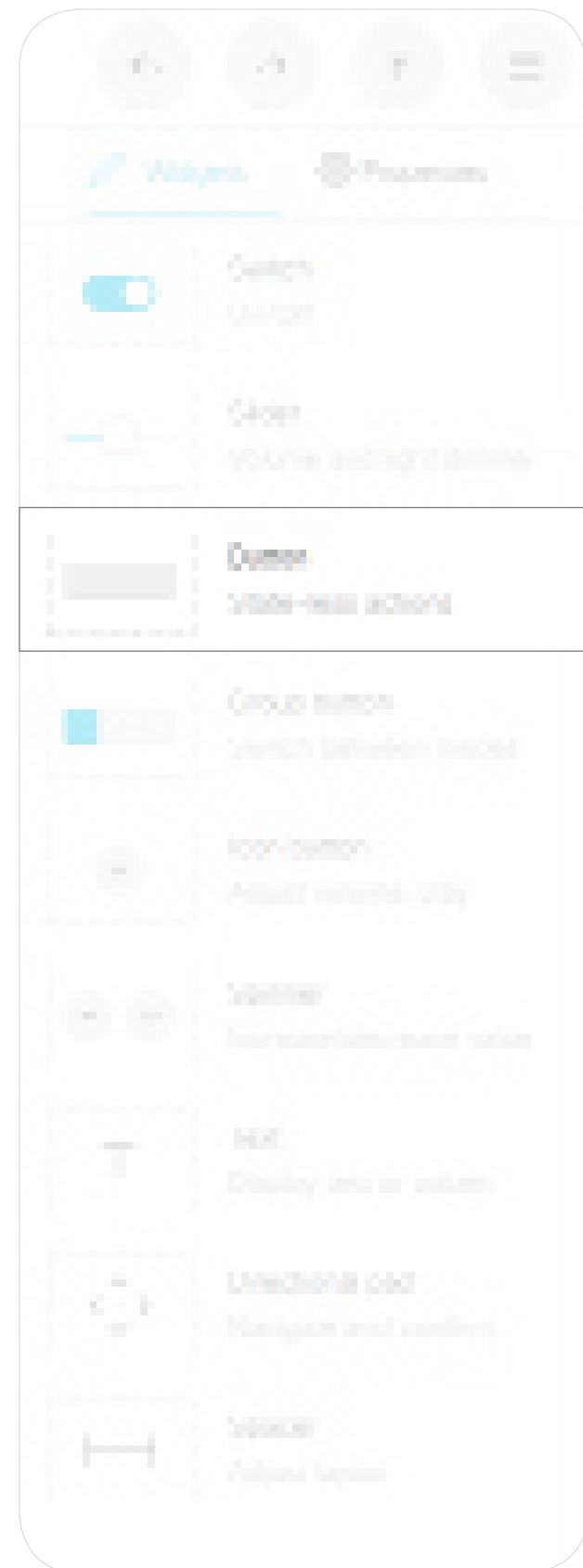
コマンド

スライダの外観は、タップまたはスライドするとすぐに変化します。ただし、制御システムは常に SetValue コマンドをビデオ システムに送信して、スライダの新しい位置を知らせる必要があります。これにより、ステータスが確実に更新されます。

例: WidgetId = "slider" のスライダを位置 "98" に設定します。

```
xCommand UserInterface Extensions Widget SetValue WidgetId: "slider" Value: "98"
```

ボタン



Event

- Pressed** ボタンを押すとトリガーされます。値: 該当なし
- Changed** ボタンを離すとトリガーされます。値: 該当なし
- Released** ボタンを離すとトリガーされます。値: 該当なし

例: WidgetId = "button" のボタンを押してから離します。

端末モード

```
*e UserInterface Extensions Event Pressed Signal: "button"
** end
*e UserInterface Extensions Event Released Signal: "button"
** end
*e UserInterface Extensions Event Clicked Signal: "button"
** end
```

XML モード

```
Event>
<UserInterface item="1">
  <Extensions item="1">
    <Widget item="1">
      <Action item="1">
        <WidgetId item="1">button</WidgetId>
        <Value item="1"></Value>
        <Type item="1">clicked</Type>
      </Action>
    </Widget>
  </Extensions>
</UserInterface>
</Event>
```

カスタム テキスト付きのボタンは、さまざまなサイズで配置できます。ボタンのサイズによって、追加できる最大文字数が決まります。テキストは次の行に折り返されません。SetValue コマンドを使用して、テキストを動的に変更することはできません。

ボタンには、active と inactive という 2 種類の状態があります。ユーザがボタンをタップしたときに、そのボタンを active 状態に設定する必要はありません。ボタンは、外観を変更せずに信号を送信するためだけに使用できます。

一度に 1 つのボタンのみを選択できるように関係付けられた複数のボタン(ラジオ ボタン)が必要な場合は、グループ ボタン(次のページ)の使用を検討してください。

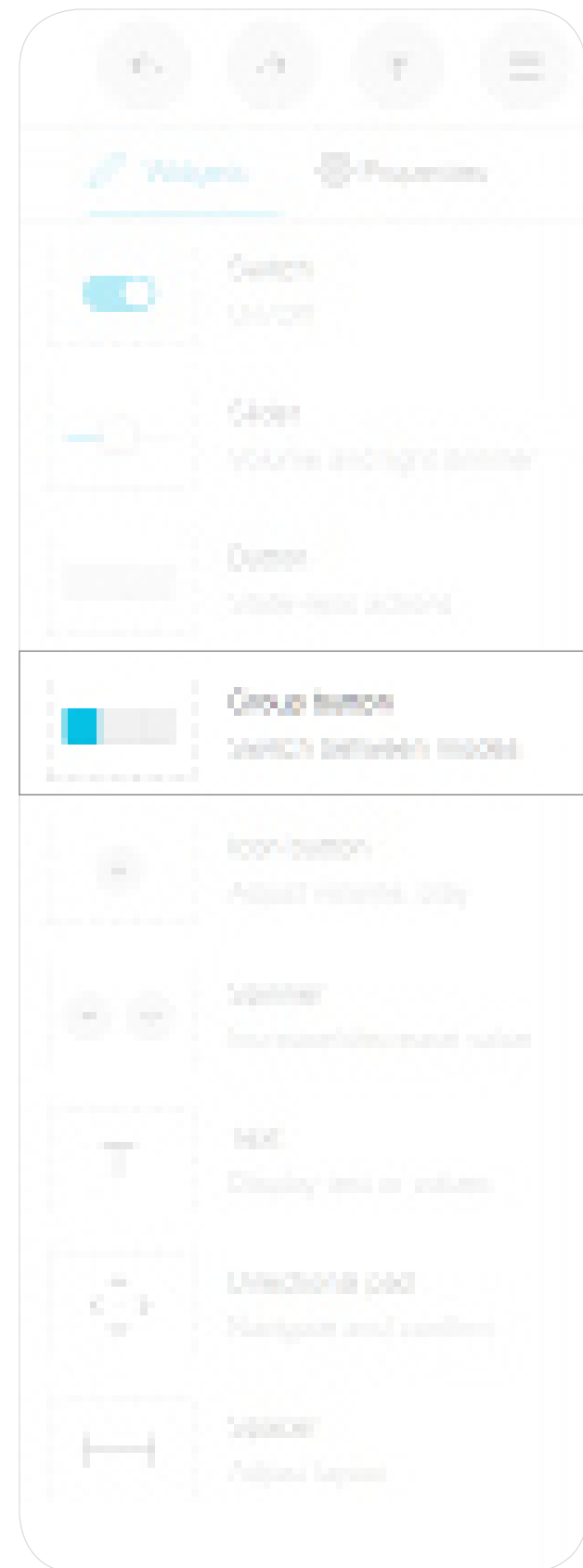
使用例: 対象のオンとオフを切り替えます。

コマンド

ユーザ インターフェイスのボタンをハイライトするかどうかを指定するには、SetValue コマンドを使用します。値が "active" の場合はボタンがハイライトされ、値が "inactive" の場合はハイライトが解除されます。

例: WidgetId = "button" のボタンをハイライトします (active 状態に設定)。

```
xCommand UserInterface Extensions Widget SetValue WidgetId: "button" Value: "active"
```



Event

- Pressed** いずれかのボタンを押すとトリガーされます。値: 押されたボタン(グループ内)のグループ ID。
- Released** いずれかのボタンを離すとトリガーされます。値: 離れたボタン(グループ内)のグループ ID

例: WidgetId = "groupbutton" のグループに 4 つのボタンがあります。グループ ID = "two" のボタンを押します。

端末モード

```
*e UserInterface Extensions Event Pressed Signal: "groupbutton:two"
** end
*e UserInterface Extensions Event Released Signal: "groupbutton:two"
** end
```

XML モード

```
<Event>
  <UserInterface item="1">
    <Extensions item="1">
      <Widget item="1">
        <Action item="1">
          <WidgetId item="1">groupbutton</WidgetId>
          <Value item="1">two</Value>
          <Type item="1">released</Type>
        </Action>
      </Widget>
    </Extensions>
  </UserInterface>
</Event>
```

グループボタンは、1 行だけではなく、マトリックスとして作成できるようになりました。これは、ラジオボタンについて最大 4 つという制限がなくなったことを意味します。

マトリックスは最大 4 列と必要な数の行で構成されます。

まず、マトリックスが構成する列の数(1、2、3、または 4)を定義することから開始します。これはマトリックス全体(すべての行など)に適用されるグローバル設定であり、1 行あたりのボタンの最大数を定義します。

ただし、行は最大数より少ないボタンを含むことが可能です。その後にボタンの自動サイジングが行われます。ボタンはつねに利用可能なスペースに収まります。

例: 3 列のマトリックスを定義し、ボタンが 7 つ必要だとします(例として 3 行)。システムはその後、ボタン 3 つを最初の行に、次の 3 つを 2 番目の行に、そして最後のボタンを 3 番目の行に配置します。3 番目の行に配置される 1 つのボタンは、スペースを埋めるように(3 列にまたがるように)自動サイジングされます。

ボタンのサイズによって、追加できる最大文字数が決まります。テキストは次の行に折り返されません。

SetValue コマンドを使用して、テキストを動的に変更することはできません。

使用例: 互いに排他的なルーム プリセット。たとえば、[暗い(Dark)]、[クール(Cool)]、[明るい(Bright)] から選択できるルーム プリセット。プリセットが有効でなくなった場合は(壁面のコントローラや Touch 10 のスライダーで照明を操作した場合など)、プリセットを選択解除(解放)してください。

もう 1 つの使用例: UI の言語を別の言語に変更します。

コマンド

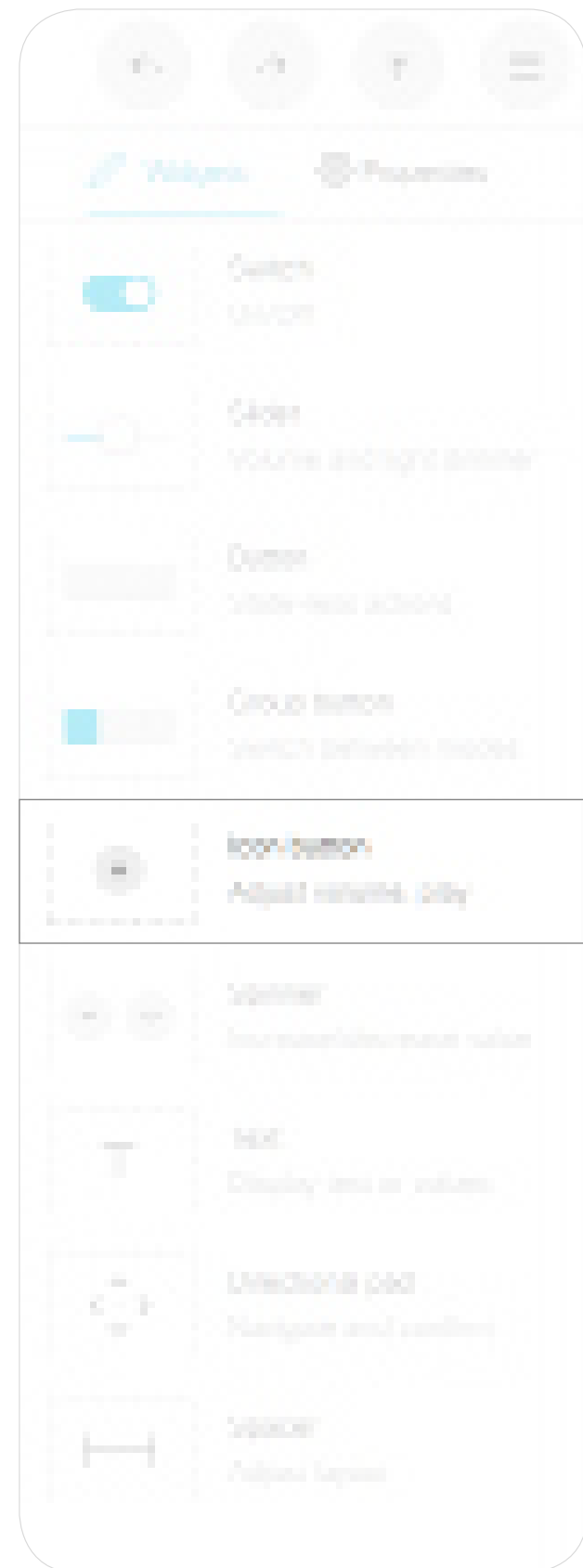
ボタンの外観は、タップするとすぐに変化します。ただし、制御システムは、いずれかのボタンがタップされたときは必ず、ビデオシステムに SetValue コマンドを送信する必要があります。これにより、ステータスが確実に更新されます。

例: WidgetId = "groupbutton" のグループ内にあるグループ ID = "one" のボタンを選択(ハイライト)します。次に、すべてのボタンを解放します(すべてのボタンのハイライトが解除されます)。

UnSetValue コマンドを使用すると、グループ内のすべてのボタンを解放して、どのボタンもハイライトしていない状態にできます。

```
xCommand UserInterface Extensions Widget SetValue
WidgetId: "groupbutton" Value: "one"
```

```
xCommand UserInterface Extensions Widget UnSetValue
WidgetId: "groupbutton"
```



Event

- Pressed** ボタンを押すとトリガーされます。値: 該当なし
- Released** ボタンを離すとトリガーされます。値: 該当なし
- Clicked** ボタンを離すとトリガーされます。値: 該当なし

例: WidgetId = "symbol" のボタンを押してから離します。

端末モード

```
*e UserInterface Extensions Event Pressed Signal: "symbol"
** end
*e UserInterface Extensions Event Released Signal: "symbol"
** end
*e UserInterface Extensions Event Clicked Signal: "symbol"
** end
```

XML モード

```
<Event>
  <UserInterface item="1">
    <Extensions item="1">
      <Widget item="1">
        <Action item="1">
          <WidgetId item="1">symbol</WidgetId>
          <Value item="1"></Value>
          <Type item="1">clicked</Type>
        </Action>
      </Widget>
    </Extensions>
  </UserInterface>
</Event>
```

アイコン ボタンは、カスタム テキストを持つボタンと動作を共有します。

ボタンには、active と inactive という 2 種類の状態があります。ユーザがボタンをタップしたときに、そのボタンを active 状態に設定する必要はありません。ボタンは、外観を変更せずに信号を送信するためだけに使用できます。

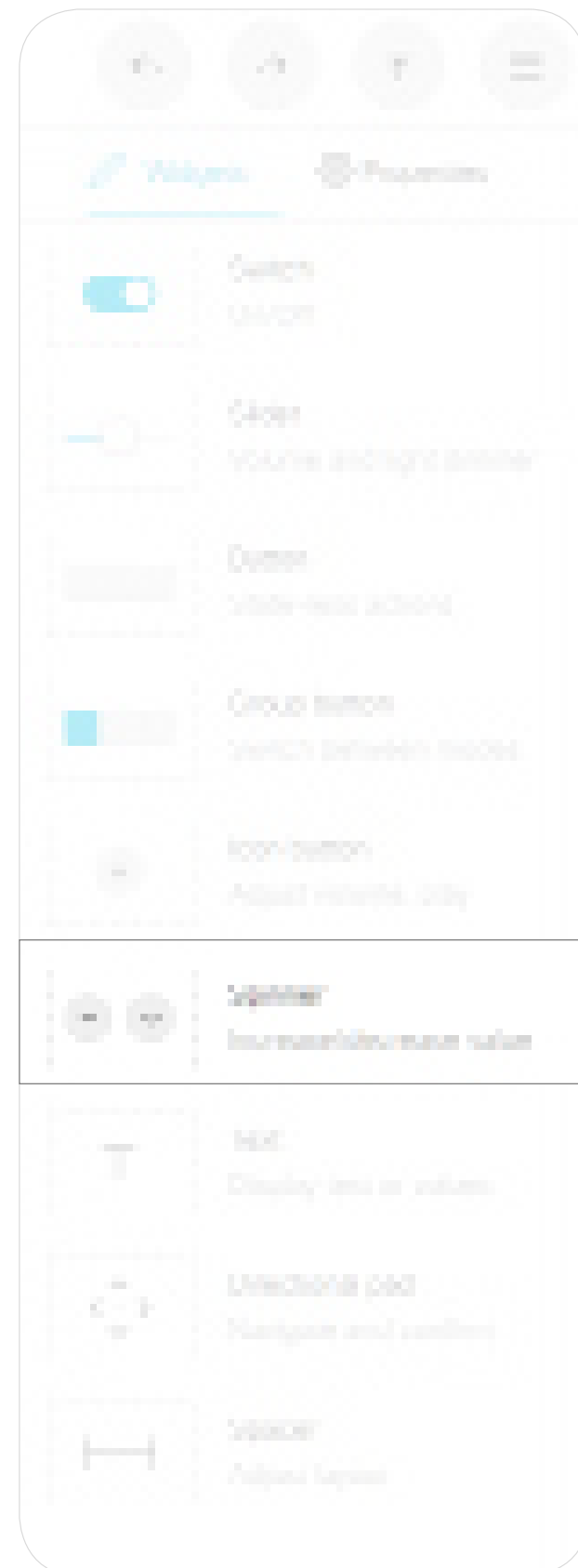
使用例: メディア プレーヤーなど、開始、停止、一時停止できるデバイスを制御します。

コマンド

ユーザ インターフェイスのボタンをハイライトするかどうかを指定するには、SetValue コマンドを使用します。値が "active" の場合はボタンがハイライトされ、値が "inactive" の場合はハイライトが解除されます。

例: WidgetId = "symbol" のボタンをハイライトします (active 状態に設定)。

```
xCommand UserInterface Extensions Widget SetValue
WidgetId: "symbol" Value: "active"
```



Event

- Pressed** スピン ボックスのいずれかのボタンを押すとトリガーされます。値: <increment/decrement>
- Released** スピン ボックスのいずれかのボタンを離すとトリガーされます。値: <increment/decrement>
- Clicked** スピン ボックスのいずれかのボタンを離すとトリガーされます。値: <increment/decrement>

例: WidgetId = "spinner" のスピン ボックスの減少(下矢印)ボタンを押してから離します。

端末モード

```
*e UserInterface Extensions Event Pressed Signal: "spinner:decrement"
** end
*e UserInterface Extensions Event Released Signal: "spinner:decrement"
** end
*e UserInterface Extensions Event Clicked Signal: "spinner:decrement"
** end
```

XML モード

```
<Event>
  <UserInterface item="1">
    <Extensions item="1">
      <Widget item="1">
        <Action item="1">
          <WidgetId item="1">spinner</WidgetId>
          <Value item="1">decrement</Value>
          <Type item="1">clicked</Type>
        </Action>
      </Widget>
    </Extensions>
  </UserInterface>
</Event>
```

スピン ボックスを使用すると、リストから値を選択できます。2 つのボタンを使用して、数値を増減したり、オプションのリストから項目を選択したりできます。

ボタンの間に表示されるテキストを追加するには、SetValue コマンドを使用します。

使用例: 室内の温度を適切に設定します。

コマンド

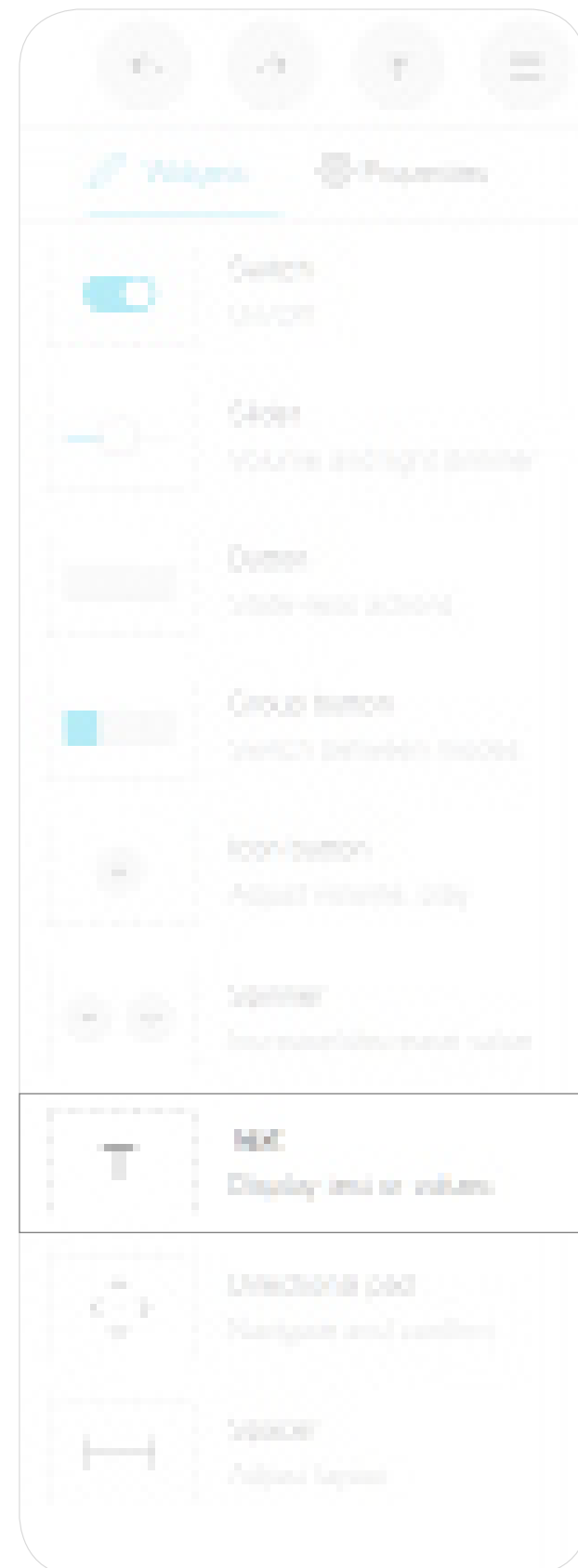
2 つのボタンの間に表示されるテキストを追加または更新するには、SetValue コマンドを使用します。

例: WidgetId = "spinner" のスピン ボックスで、増加(上矢印)ボタンと減少(下矢印)ボタンの間にテキスト "22" を追加します。

```
xCommand UserInterface Extensions Widget SetValue WidgetId: "spinner" Value: "22"
```

テキスト (Text)

Event
なし



コマンド

テキスト ボックスにテキストを設定するには、SetValue コマンドを使用します。

例: WidgetId = "textbox" のテキスト ボックスに「プロジェクターはウォームアップ中です」というテキストを設定します。

```
xCommand UserInterface Extensions Widget SetValue WidgetId: "textbox" Value:
"The projector is warming up."
```

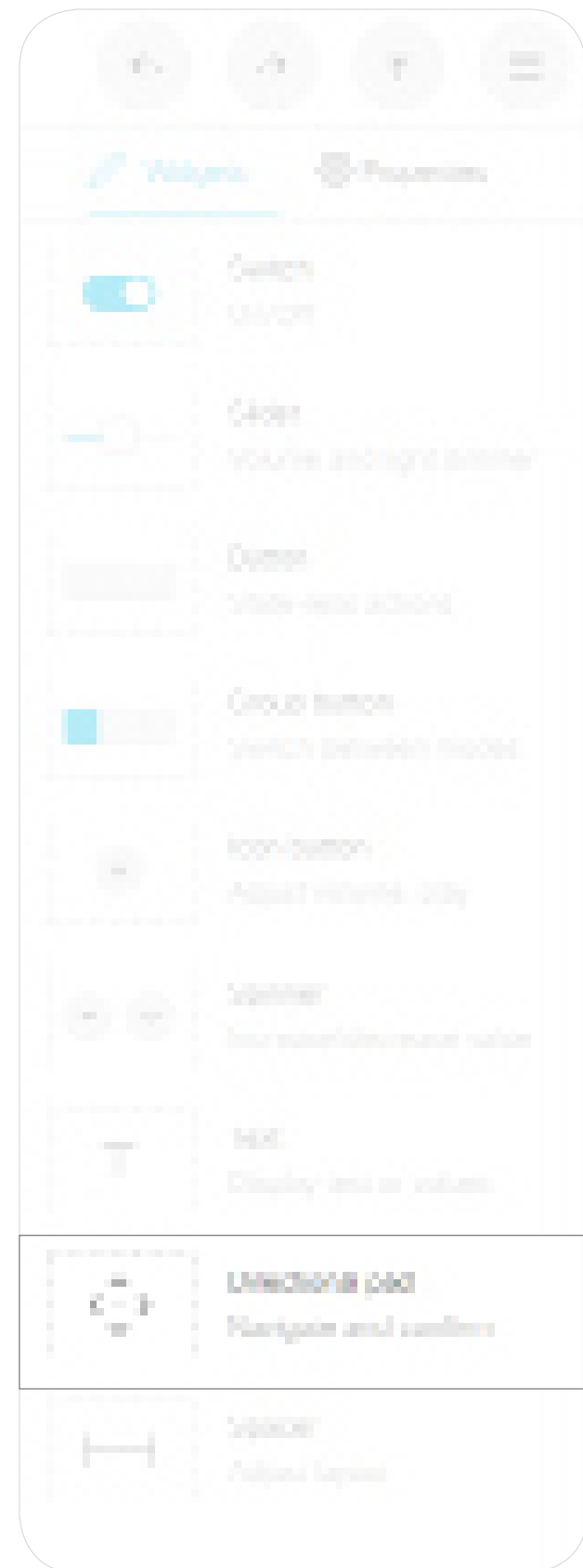
テキスト ボックスは、さまざまなサイズで配置できます。最大 2 行までのテキストを追加することができ、テキストは次の行に自動的に折り返されます。

フォント サイズが大きい、ライン ラップなしの小さなテキスト ボックスも使用できます。

エディタでテキスト ボックスに最初のテキストを指定し、後から SetValue コマンドを使用してテキストを動的に入力することができます。

使用例: ヘルプ テキスト、操作の指示、各種のプリセットの説明、制御システムから通知される情報テキスト(「プロジェクターはウォームアップ中です」など)。

フォント サイズが大きいテキスト ボックスは、主に現在の室温などのステータス値を示すために使用されます。



Event

- Pressed ボタンを押すとトリガーされます。値: 該当なし
- Changed ボタンを離すとトリガーされます。値: 該当なし
- Released ボタンを離すとトリガーされます。値: 該当なし

例: WidgetId = "dirpad" のボタンを押してから離します。

端末モード

```
*e UserInterface Extensions Event Pressed Signal: "dirpad:up"
** end
*e UserInterface Extensions Event Released Signal: "dirpad:up"
** end
*e UserInterface Extensions Event Clicked Signal: "dirpad:up"
** end
```

XML モード

```
<Event>
  <UserInterface item="1">
    <Extensions item="1">
      <Widget item="1">
        <Action item="1">
          <WidgetId item="1">dirpad</WidgetId>
          <Value item="up"></Value>
          <Type item="1">clicked</Type>
        </Action>
      </Widget>
    </Extensions>
  </UserInterface>
</Event>
```

方向パッドは、5つのボタン、つまり4つの方向ボタンと中央のボタンのセットと見なすことができます。

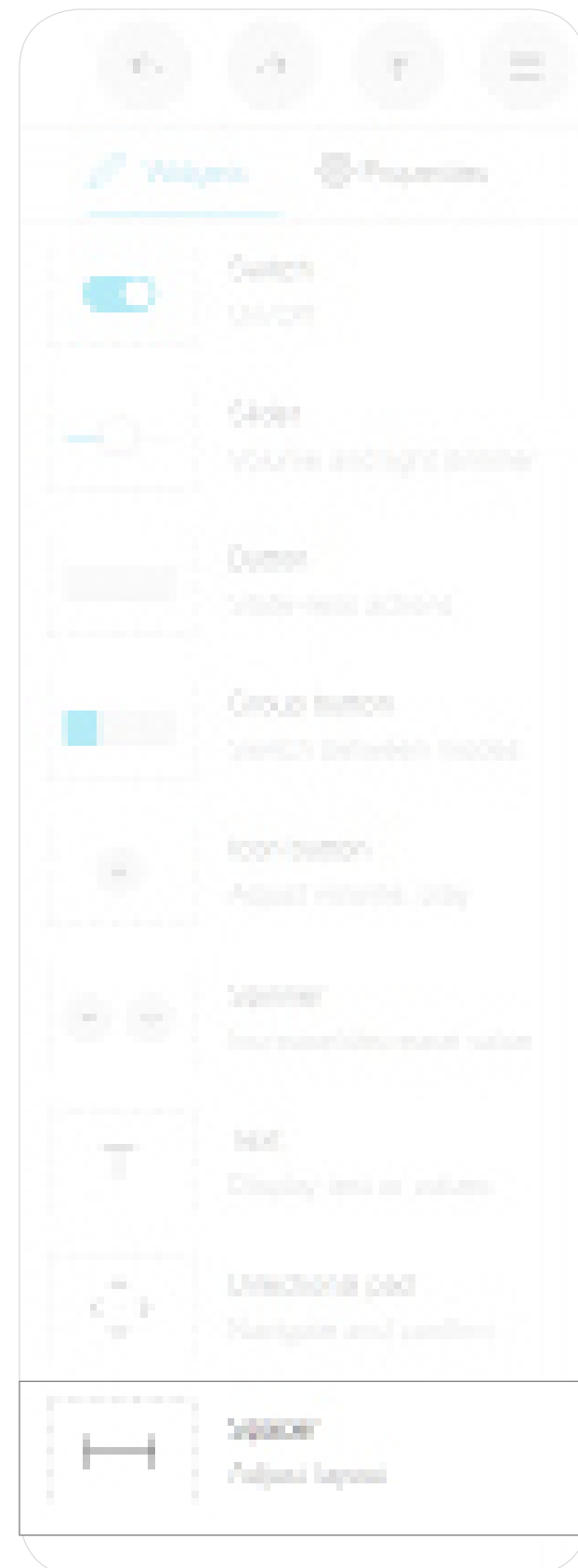
左の例から分かるように、イベントは次の形式を取ります。

“<WidgetId><the_button_pushed>”

ここで、the_button_pushed は次の値を取ります。

up、down、left、right、center

使用例: AppleTV の制御

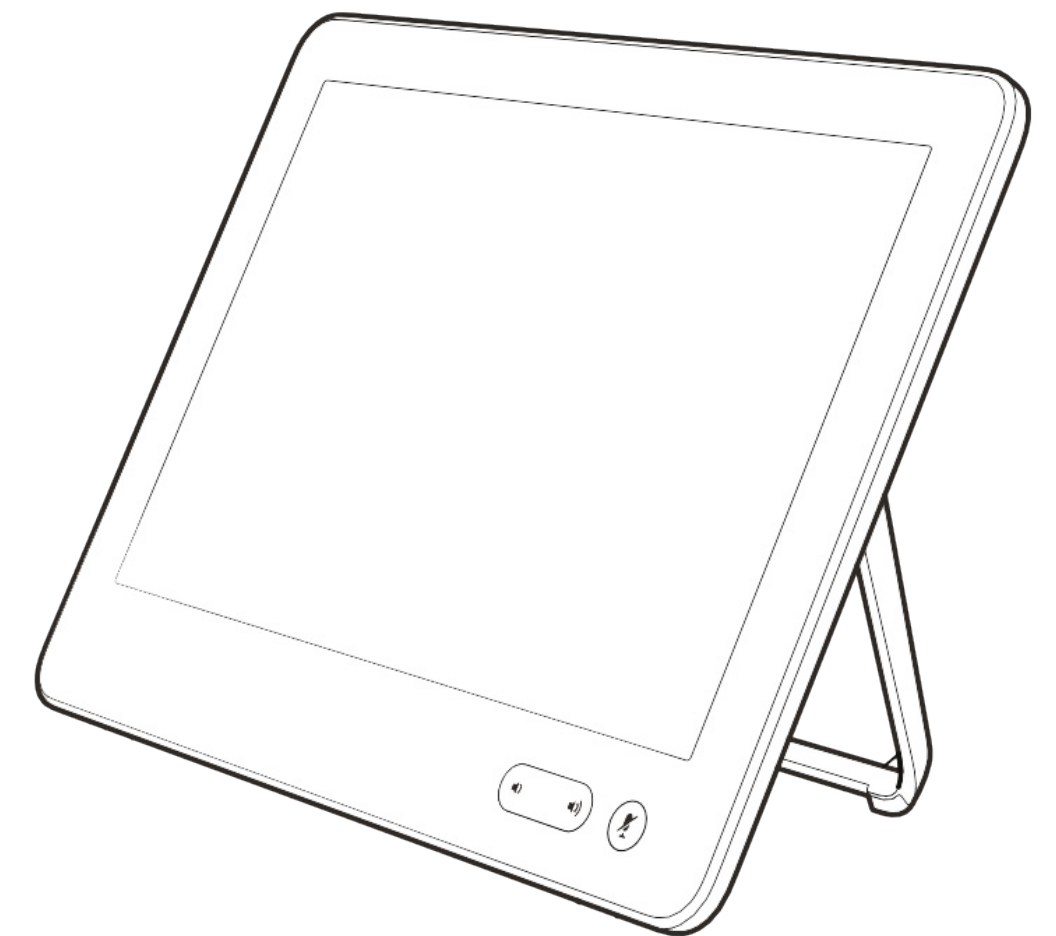


スペーサはレイアウト ツールにすぎません。したがって、スペーサに関連付けられたイベントまたはコマンドはありません。

スペーサを使用すると、ウィジェット間、またはウィジェットの後にスペースを追加できます。これはレイアウト ツールにすぎません。

スペーサの幅は調節可能です(1 ~ 4)。最大値に設定した場合、その行を占有します。また、垂直スペーサとしても使用できます。

コマンド リファレンス



イベント

UserInterface Extensions Event Pressed

ウィジェットを最初に押したときにビデオ システムによって送信されます。

“Pressed” タイプの UserInterface Extensions Widget Action イベントに相当します。

*e UserInterface Extensions Event Pressed Signal: [*Signal*](#)

ここで、

[*Signal*](#): String (0, 255)

文字列の形式は “<WidgetId>:<Value>” です。ここで、<WidgetId> はイベントをトリガーするウィジェットの固有識別子で、<Value> はウィジェットの値です。有効な値の範囲は、ウィジェット タイプによって異なります。

UserInterface Extensions Event Changed

ウィジェットの値を変更するときにビデオ システムによって送信されます (トグル ボタンとスライダにのみ適用されます)。

“Changed” タイプの UserInterface Extensions Widget Action イベントに相当します。

*e UserInterface Extensions Event Changed Signal: [*Signal*](#)

ここで、

[*Signal*](#): String (0, 255)

文字列の形式は “<WidgetId>:<Value>” です。ここで、<WidgetId> はイベントをトリガーするウィジェットの固有識別子で、<Value> はウィジェットの値です。有効な値の範囲は、ウィジェット タイプによって異なります。

UserInterface Extensions Event Released

ウィジェットから指を離れたときにビデオ システムによって送信されます (ウィジェット外に指を移動してから離しても有効になります)。

“Released” タイプの UserInterface Extensions Widget Action イベントに相当します。

*e UserInterface Extensions Event Released Signal: [*Signal*](#)

ここで、

[*Signal*](#): String (0, 255)

文字列の形式は “<WidgetId>:<Value>” です。ここで、<WidgetId> はイベントをトリガーするウィジェットの固有識別子で、<Value> はウィジェットの値です。有効な値の範囲は、ウィジェット タイプによって異なります。

UserInterface Extensions Event Clicked

ウィジェットをクリックしたときにビデオ システムによって送信されます (ウィジェット外に指を移動しないで押して離れた場合)。

“Clicked” タイプの UserInterface Extensions Widget Action イベントに相当します。

*e UserInterface Extensions Event Clicked Signal: [*Signal*](#)

ここで、

[*Signal*](#): String (0, 255)

文字列の形式は “<WidgetId>:<Value>” です。ここで、<WidgetId> はイベントをトリガーするウィジェットの固有識別子で、<Value> はウィジェットの値です。有効な値の範囲は、ウィジェット タイプによって異なります。

イベント(続き)

UserInterface Extensions Widget Action

ユーザ インターフェイス(室内制御パネル)上のいずれかのコントロールをユーザが使用すると、ビデオ システムによって送信されます。

UserInterface Extensions Event *Type* イベントに相当します。

アクション タイプに応じて異なりますが、このイベントは次のイベントのいずれかに相当します。

- UserInterface Extensions Event Pressed
- UserInterface Extensions Event Changed
- UserInterface Extensions Event Released
- UserInterface Extensions Event Clicked Events

```
<Event>
  <UserInterface item="1">
    <Extensions item="1">
      <Widget item="1">
        <Action item="1">
          <WidgetId item="1">WidgetId</WidgetId>
          <Value item="1">Value</Value>
          <Type item="1">Type</Type>
        </Action>
      </Widget>
    </Extensions>
  </UserInterface>
</Event>
```

ここで、

WidgetId: String (0, 40)

イベントをトリガーしたウィジェットの固有識別子。

Value: String (0, 255)

ウィジェットの値。有効な値の範囲は、ウィジェット タイプによって異なります。

Type: <Pressed/Changed/Released/Clicked>

Pressed: ウィジェットを最初に押したときに送信されます。

Changed: ウィジェットの値を変更したときに送信されます(トグル ボタンとスライダのみ)。

Released: ウィジェットから指を離れたときに送信されます(ウィジェット外に指を移動してから離しても送信されます)。

Clicked: ウィジェットをクリックしたとき(ウィジェットを押してウィジェット外に指を移動しないで離れたとき)に送信されます。

UserInterface Extensions Widget LayoutUpdated

ユーザ インターフェイス拡張の設定ファイルが更新されたとき(つまり、室内制御エディタからビデオ システムに新しい設定をエクスポートするとき)に、ビデオ システムによって送信されます。

*e UserInterface Extensions Widget LayoutUpdated
または

```
<Event>
  <UserInterface item="1">
    <Extensions item="1">
      <Widget item="1">
        <LayoutUpdated item="1"/>
      </Widget>
    </Extensions>
  </UserInterface>
</Event>
```

コマンド

UserInterface Extensions Widget SetValue

このコマンドは、指定されたウィジェットの値を設定します。それに応じて UserInterface Extensions のステータスが更新されます。範囲外の値を設定しようとすると、エラーが返されます。

使用方法:

```
xCommand UserInterface Extensions Widget SetValue Value: Value
WidgetId: WidgetId
```

ここで、

Value: String (0, 255)

ウィジェットの値。値の範囲は、ウィジェット タイプによって異なります。

WidgetId: String (0, 40)

ウィジェットの固有識別子。

UserInterface Extensions Widget UnsetValue

このコマンドは、指定されたウィジェットの値を空にします。それに応じて UserInterface Extensions のステータスが更新されます。そのウィジェットの選択が解除されたことがユーザ インターフェイスに通知されます。

使用方法:

```
xCommand UserInterface Extensions Widget UnsetValue WidgetId:
WidgetId
```

ここで、

WidgetId: String (0, 40)

ウィジェットの固有識別子。

UserInterface Extensions Clear

このコマンドは、ビデオ システムからすべてのユーザ インターフェイス拡張 (ウィジェット) を削除します。

使用方法:

```
xCommand UserInterface Extensions Clear
```

UserInterface Extensions List

このコマンドを使用して、ビデオ システムに存在するすべてのユーザ インターフェイス拡張 (ウィジェット) を一覧表示します。

使用方法:

```
xCommand UserInterface Extensions List
```

ステータス

UserInterface Extensions Widget [n] WidgetId

UserInterface Extensions Widget [n] Value

このステータスは、ウィジェットの識別子 (WidgetId) と現在の値を返します。

UserInterface Extensions Widget SetValue コマンドを使用して値が設定されるまで、値は空の文字列です。

使用方法:

```
xStatus UserInterface Extensions
```

返される結果の値スペース:

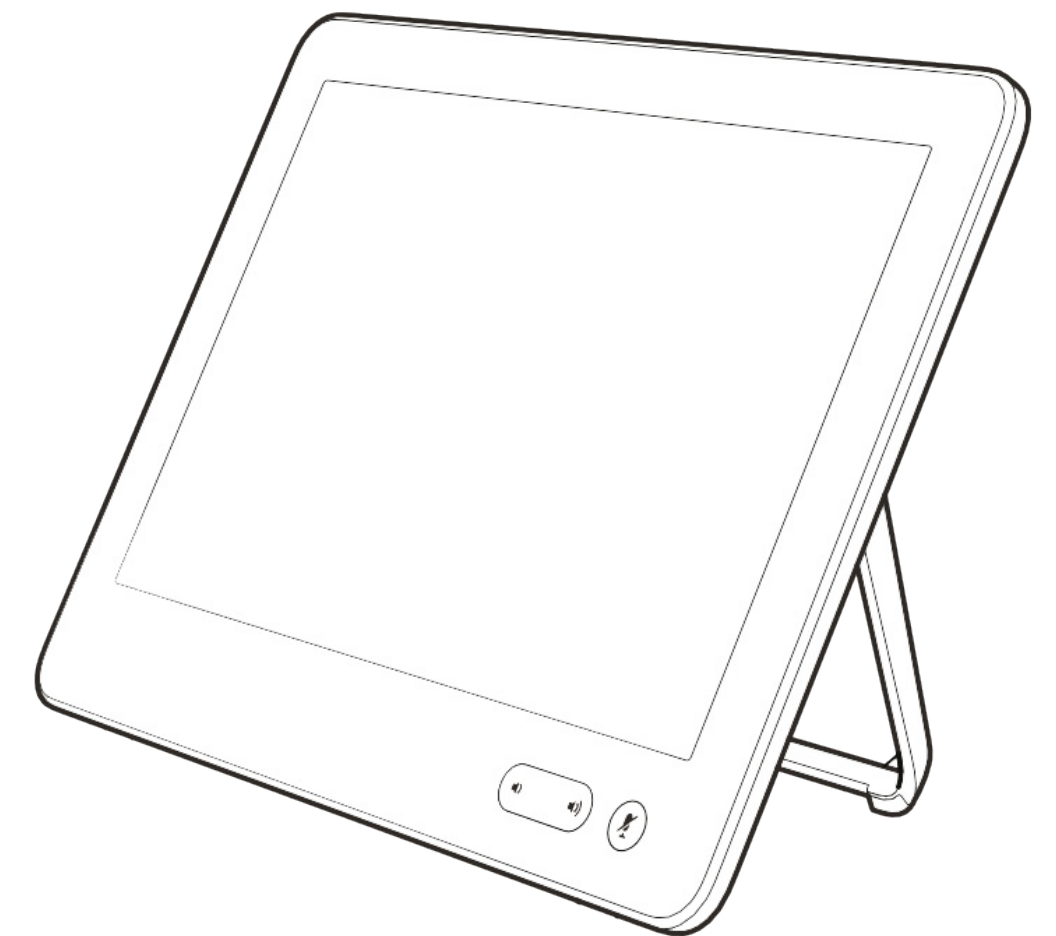
値: ウィジェットの値。ウィジェット タイプによって異なります。文字列 (0、255)

WidgetId: ウィジェットの固有識別子。文字列 (0、40)。

例:

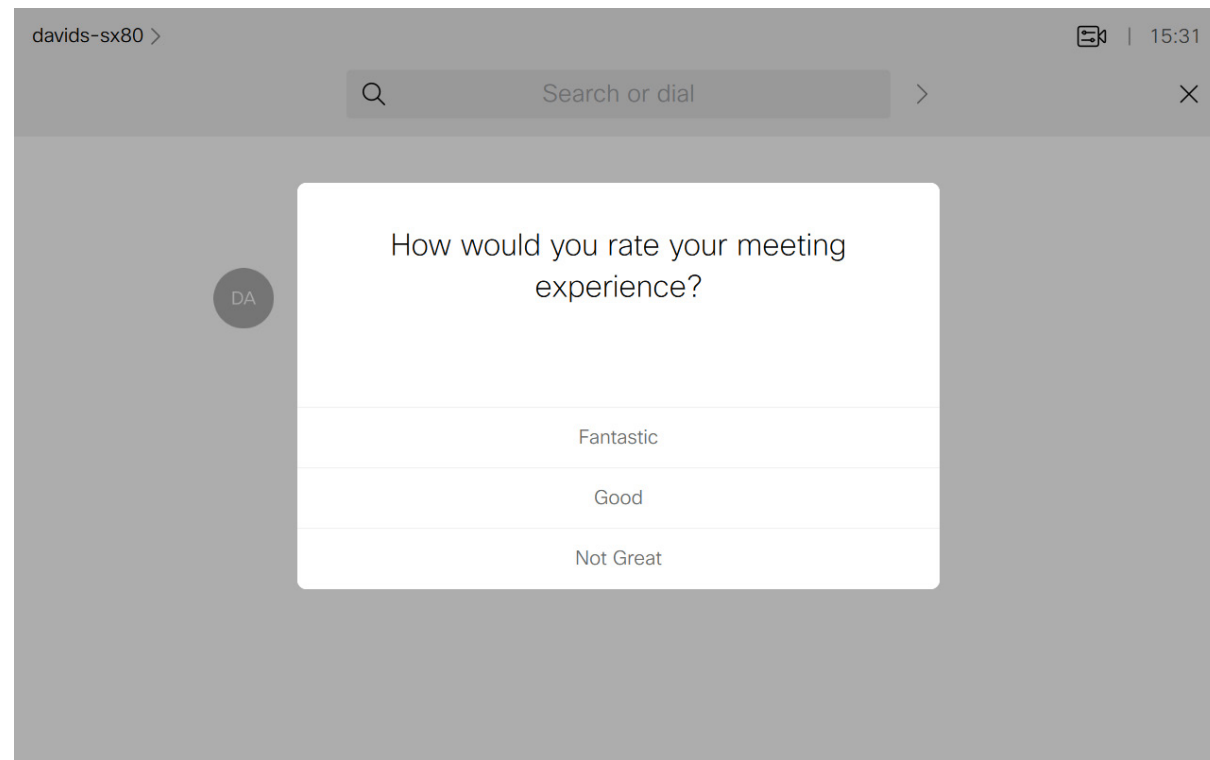
```
xstatus UserInterface Extensions
*s UserInterface Extensions Widget 1 Value: "on"
*s UserInterface Extensions Widget 1 WidgetId: "togglebutton"
*s UserInterface Extensions Widget 2 Value: "255"
*s UserInterface Extensions Widget 2 WidgetId: "slider"
*s UserInterface Extensions Widget 3 Value: "Blinds"
*s UserInterface Extensions Widget 3 WidgetId: "spinner"
*s UserInterface Extensions Widget 4 Value: "inactive"
*s UserInterface Extensions Widget 4 WidgetId: "button"
*s UserInterface Extensions Widget 5 Value: "2"
*s UserInterface Extensions Widget 5 WidgetId: "groupbutton"
*s UserInterface Extensions Widget 6 Value: "Projector is ready"
*s UserInterface Extensions Widget 6 WidgetId: "textfield"
** end
```

インタラクティブ メッセージの作成



インタラクティブ メッセージの仕組み (I)

例 1: エクスペリエンスを評価する



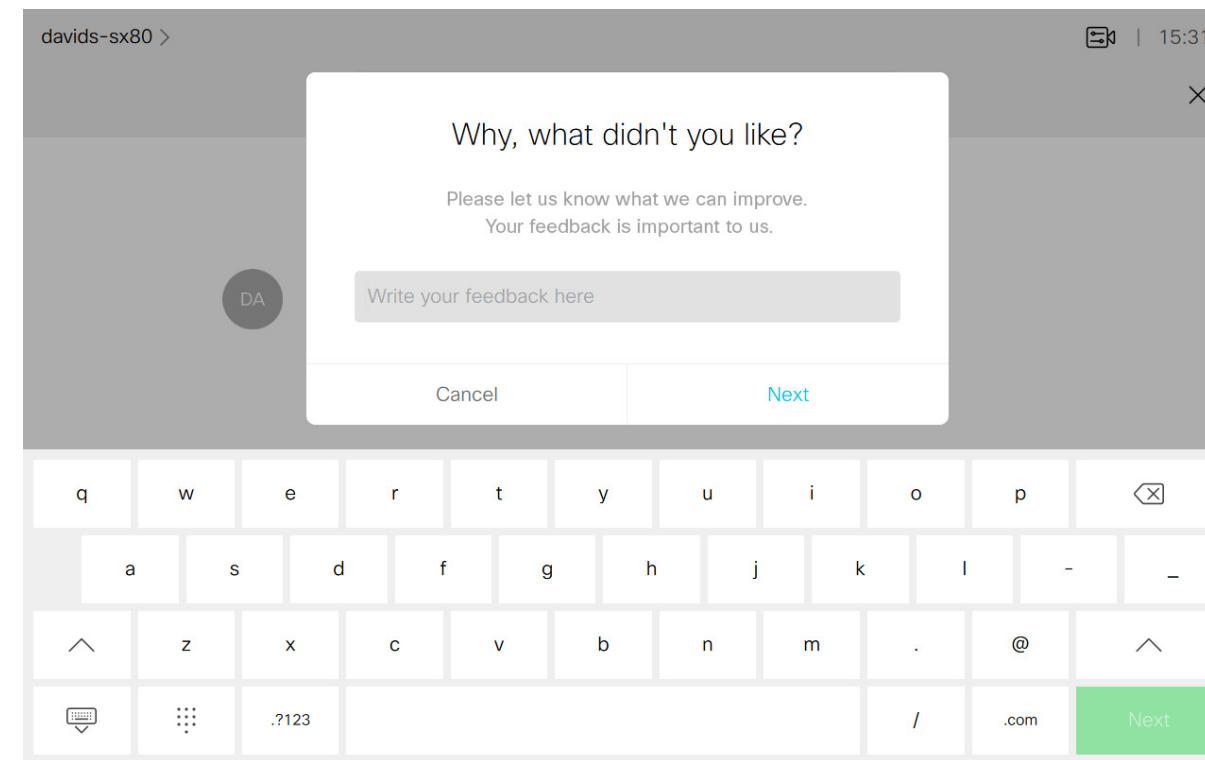
xCommand UserInterface Message Prompt Display
 FeedbackId: "MeetingExperience" Text: "" Title: "今回の会議エクスペリエンスをどのように評価いただけますか?" Option.1: "素晴らしい" Option.2: "良かった" Option.3: "良くなかった"

*e UserInterface Message Prompt Response FeedbackId: "MeetingExperience"

*e UserInterface Message Prompt Response OptionId: 1

```
<XmlDoc resultId="">
<Event>
  <UserInterface item="1">
    <Message item="1">
      <Prompt item="1">
        <Response item="1">
          <FeedbackId item="1">MeetingExperience</FeedbackId>
          <OptionId item="1">1</OptionId>
        </Response>
      </Prompt>
    </Message>
  </UserInterface>
</Event>
</XmlDoc>
```

例 2: ここにフィードバックを書き込む



xCommand UserInterface Message TextInput Display
 FeedbackId: "MeetingFeedback" Placeholder: "ここにご意見をお書きください" SubmitText: "次へ" Title: "お気に召さなかった点とその理由をお書きください。" Text: "弊社の改善のためのご意見をお寄せください。お客様のご意見は非常に重要です。"

*e UserInterface Message TextInput Response FeedbackId: "MeetingFeedback"

*e UserInterface Message TextInput Response Text: "解像度が低い"

```
<XmlDoc resultId="">
<Event>
  <UserInterface item="1">
    <Message item="1">
      <TextInput item="1">
        <Response item="1">
          <FeedbackId item="1">MeetingFeedback</FeedbackId>
          <Text item="1">Low resolution</Text>
        </Response>
      </TextInput>
    </Message>
  </UserInterface>
</Event>
</XmlDoc>
```

メッセージ機能は、Touch10/DX の画面上で警告および/またはインタラクティブなメッセージの作成を行い、ユーザに対して適切な行動を行うよう促すことができます。

以前のメッセージでの選択に応じて次のメッセージを表示する、一連のメッセージ群を作成したい場合、そのようなイベント作成を行うマクロの使用を推奨します。または、作成されたイベントに反応する外部のコントロール デバイスを使用することもできます。

ユーザからの入力を送信するには、HttpFeedback の使用に習熟する必要があります。

HttpFeedbackによって、API 状態の変更(ステータス、イベント、および設定の更新)時に HTTP フィードバック メッセージ(Webhook)を送信するようなデバイスの設定が可能です。HTTP Post フィードバック メッセージは、指定されている ServerURL に送信されます。

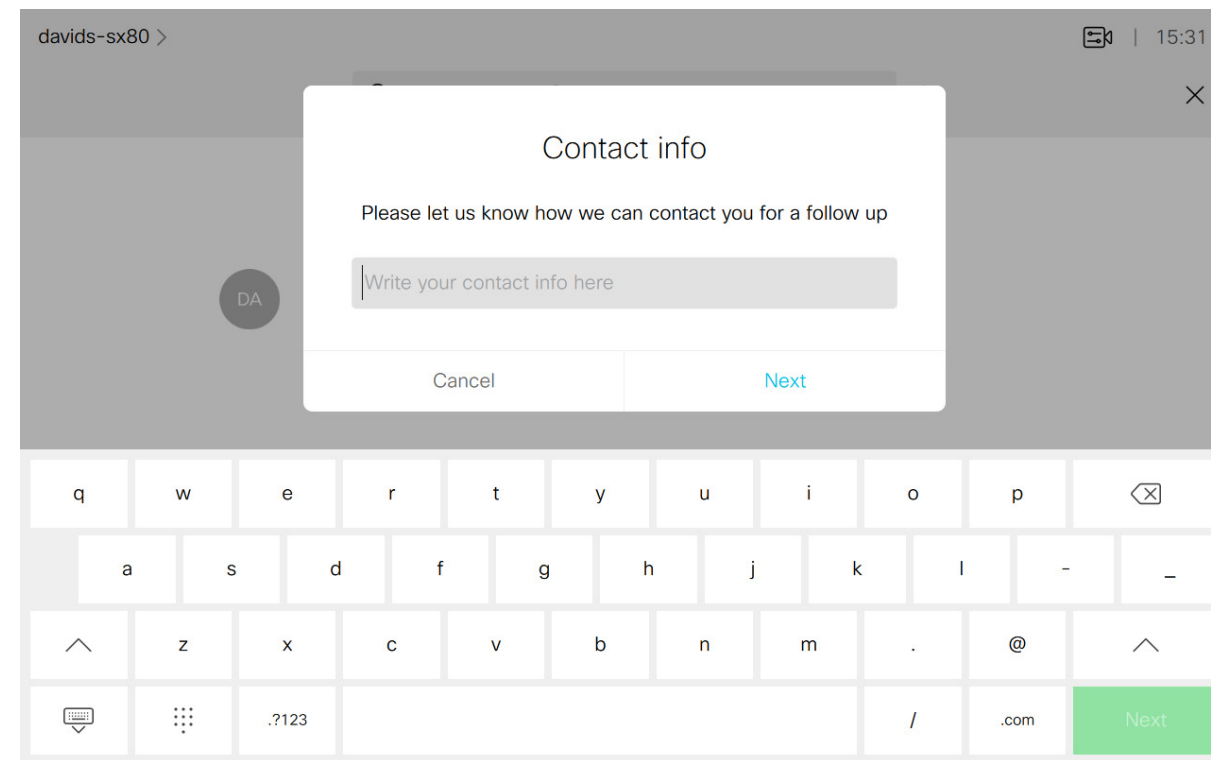
詳細については、API リファレンス ガイドで確認してください。

左にある例 2 に示すようにメッセージの作成を行う際、[次へ (Next)] ボタン上のテキストを指定できる点に注意してください。ただし、[キャンセル (Cancel)] ボタンはデフォルトで表示され、テキストは変更できません。

例 4 (次ページ)に示すようなメッセージアラートを使用したメッセージの作成を行う際、[却下 (Dismiss)] ボタンがデフォルトで表示されます。このボタンのテキストは変更できません。

インタラクティブ メッセージの仕組み(II)

例 3: 連絡を取る



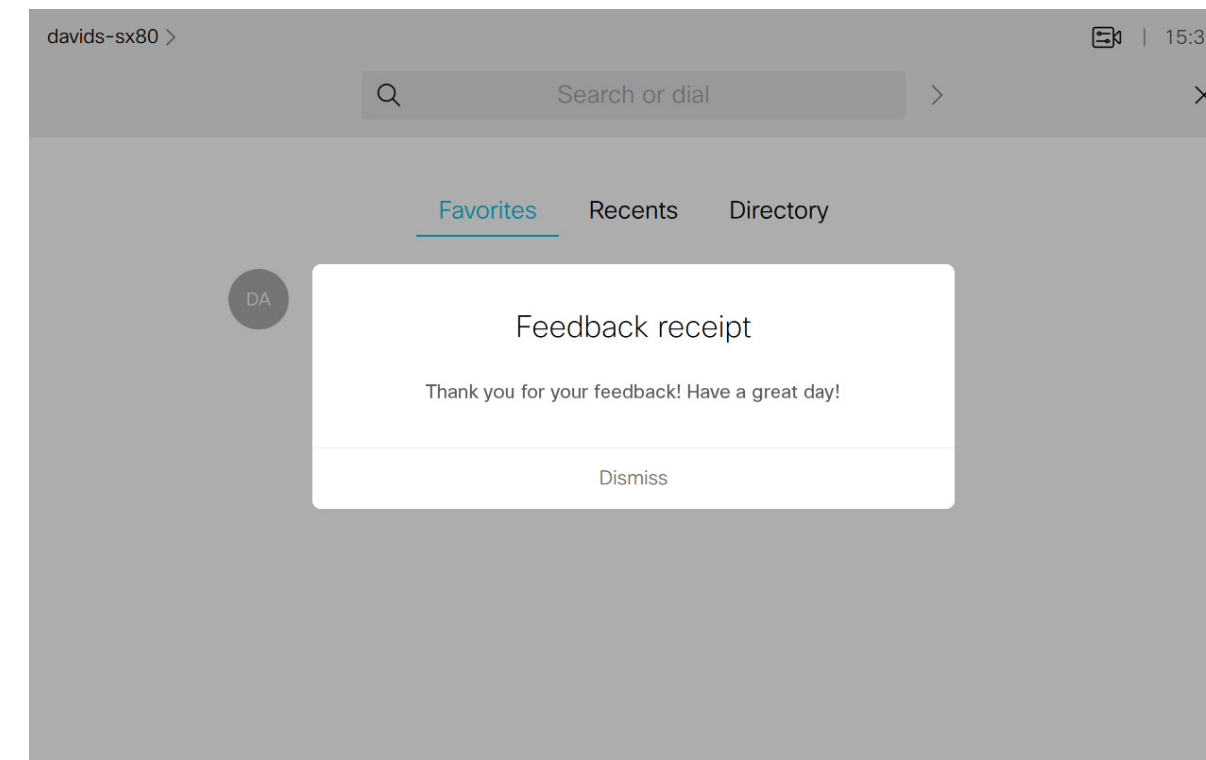
xCommand UserInterface Message TextInput Display
 FeedbackId: "ContactInfo" Placeholder: "連絡先をご記入ください" SubmitText: "次へ" Title: "連絡先情報" Text: "フォローアップのためにお客様への連絡手段をお知らせください"

*e UserInterface Message TextInput Response FeedbackId: "ContactInfo"

*e UserInterface Message TextInput Response Text: "john@go.webex.com"

```
<XmlDoc resultId="">
<Event>
  <UserInterface item="1">
    <Message item="1">
      <TextInput item="1">
        <Response item="1">
          <FeedbackId item="1">ContactInfo</FeedbackId>
          <Text item="1">john@go.webex.com</Text>
        </Response>
      </TextInput>
    </Message>
  </UserInterface>
</Event>
</XmlDoc>
```

例 4: フィードバックの受信

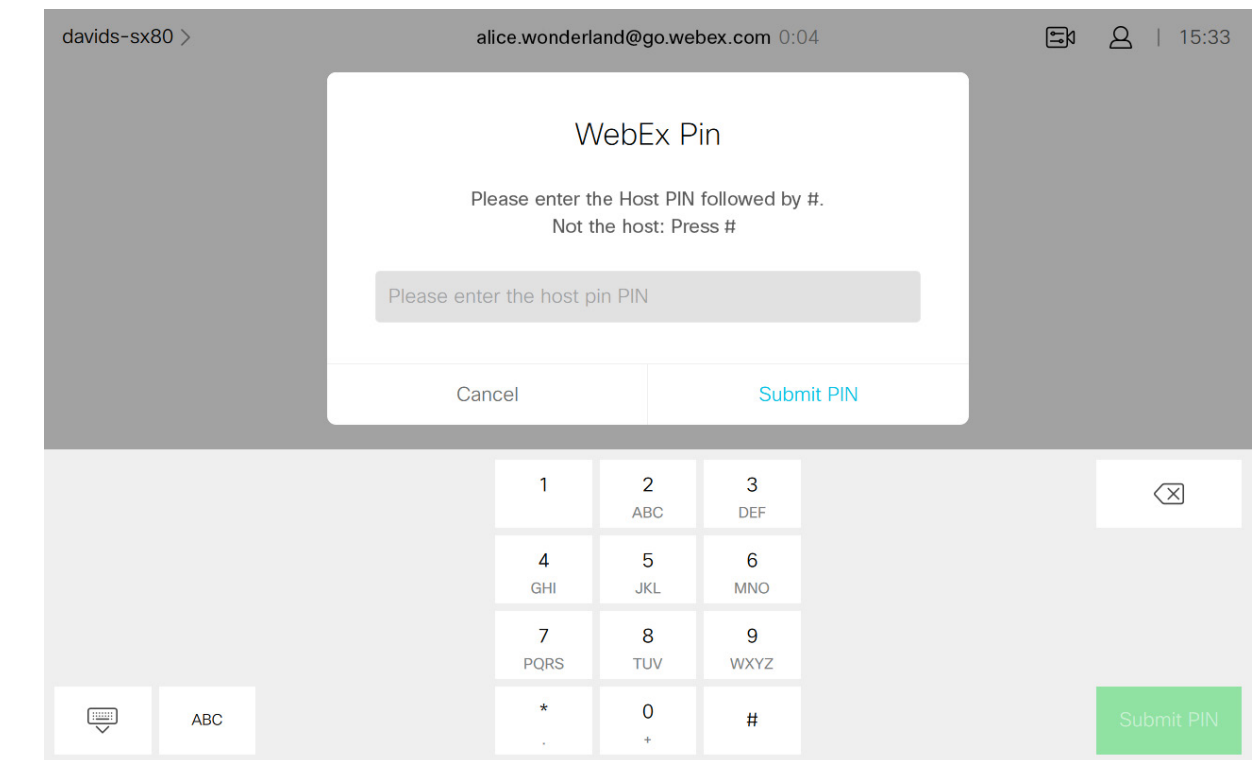


xCommand UserInterface Message Alert Display Title: "フィードバック受信" text: "ご意見を承りました!ありがとうございました。"

*e UserInterface Message Alert Cleared

```
<XmlDoc resultId="">
<Event>
  <UserInterface item="1">
    <Message item="1">
      <Alert item="1">
        <Cleared item="1"/>
      </Alert>
    </Message>
  </UserInterface>
</Event>
</XmlDoc>
```

例 5: WebEx 暗証番号の入力



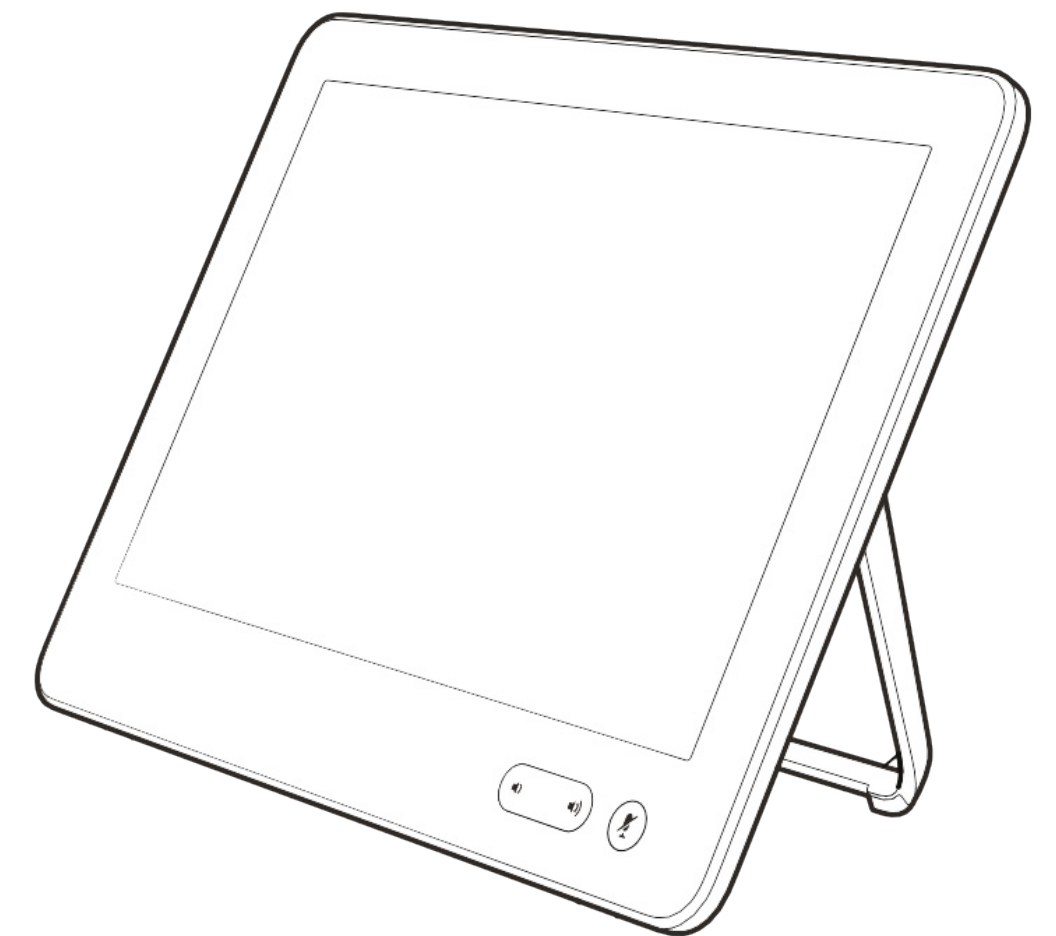
xCommand UserInterface Message TextInput Display
 FeedbackId: "WebExPin" InputType: Numeric Placeholder: "暗証番号を入力してください" SubmitText: "Submit PIN" Text: "ホストの PIN をご入力ください。入力が終わりましたら # を押してください。ホストでない場合、# を押してください" Title: "WebEx Pin"

*e UserInterface Message TextInput Response FeedbackId: "WebExPin"

*e UserInterface Message TextInput Response Text: "1122#"

```
<XmlDoc resultId="">
<Event>
  <UserInterface item="1">
    <Message item="1">
      <TextInput item="1">
        <Response item="1">
          <FeedbackId item="1">WebExPin</FeedbackId>
          <Text item="1">1122#</Text>
        </Response>
      </TextInput>
    </Message>
  </UserInterface>
</Event>
</XmlDoc>
```

トラブルシューティング



トラブルシューティングを行う際のヒント

サインイン

ビデオ システムの Web インターフェイスに管理者クレデンシャルでサインインし、**[統合 (Integration)] > [室内制御 (In-Room Control)]** に移動します。矢印をクリックして、**[開発ツール (Development Tools)]** を表示します。

すべてのウィジェットとそのステータスの概要

[ウィジェット状態の概要 (Widget State Overview)] ウィンドウには、すべてのウィジェットとそのステータスが一覧表示されます。ステータスは、**[現在の値 (Current Value)]** 列に表示されます。

[現在の値 (Current Value)] 列が空の場合、ウィジェットは初期化されておらず、値が設定されていません。制御システムが最初にビデオ システムに接続するとき、すべてのウィジェットを初期化することをお勧めします。

更新された値のビデオ システムへの送信

制御システムは、**SetValue** コマンドをビデオ システムに送信し、ウィジェットを更新するように指示します。テスト目的で、**[ウィジェット状態の概要 (Widget State Overview)]** ウィンドウの **[値の更新 (Update Value)]** 列を使用して、制御システムをシミュレートすることができます。

入力フィールドの 1 つに値を入力して、対応する **SetValue** コマンドをビデオ システムにただちに送信します。**[現在の値 (Current Value)]** 列 (ステータス) が更新され、それに応じて Touch10 室内制御パネルが変更されます。

ウィジェットの値をクリアするには、**[設定解除 (Unset)]** をクリックします (**UnsetValue** コマンドを送信します)。

制御システムがビデオ システムに接続されている場合、**[現在の値 (Current Value)]** 列と **[値の更新 (Update Value)]** 列が同期しなくなることがあります。**[現在の値 (Current Value)]** 列には常に現在のステータスが表示されません。**SetValue** コマンドが実際の制御システムまたは **[値の更新 (Update Value)]** 列から送信されているかどうかは関係ありません。

イベントとステータスの更新の確認

ウィジェットに関連するすべてのイベントとステータスの更新は、ただちに **[ログ (Log)]** ウィンドウに表示されます。イベントには、*e という接頭辞が付き、ステータスには *s という接頭辞が付きます。

Touch 10 ユーザ インターフェイスでコントロールを使用するとイベントが表示され、ビデオ システムのステータスを変更するコマンドがビデオ システムに送信されると、ステータスが更新されます。

パネルのロードに失敗した場合

既存の室内制御パネルがエディタの起動時に自動的にロードできなかった場合は、コーデックからパネルを手動でインポートするか、オフライン エディタで作成したローカル ファイルをロードすることができます。

どの方法を使用しても、エディタで保存されていないデータは消去されますが、ビデオ システム上の既存の室内制御パネルは、新しいパネルがビデオ システムにエクスポートされるまで上書きも削除もされません。

マクロが原因ではないことを確認する

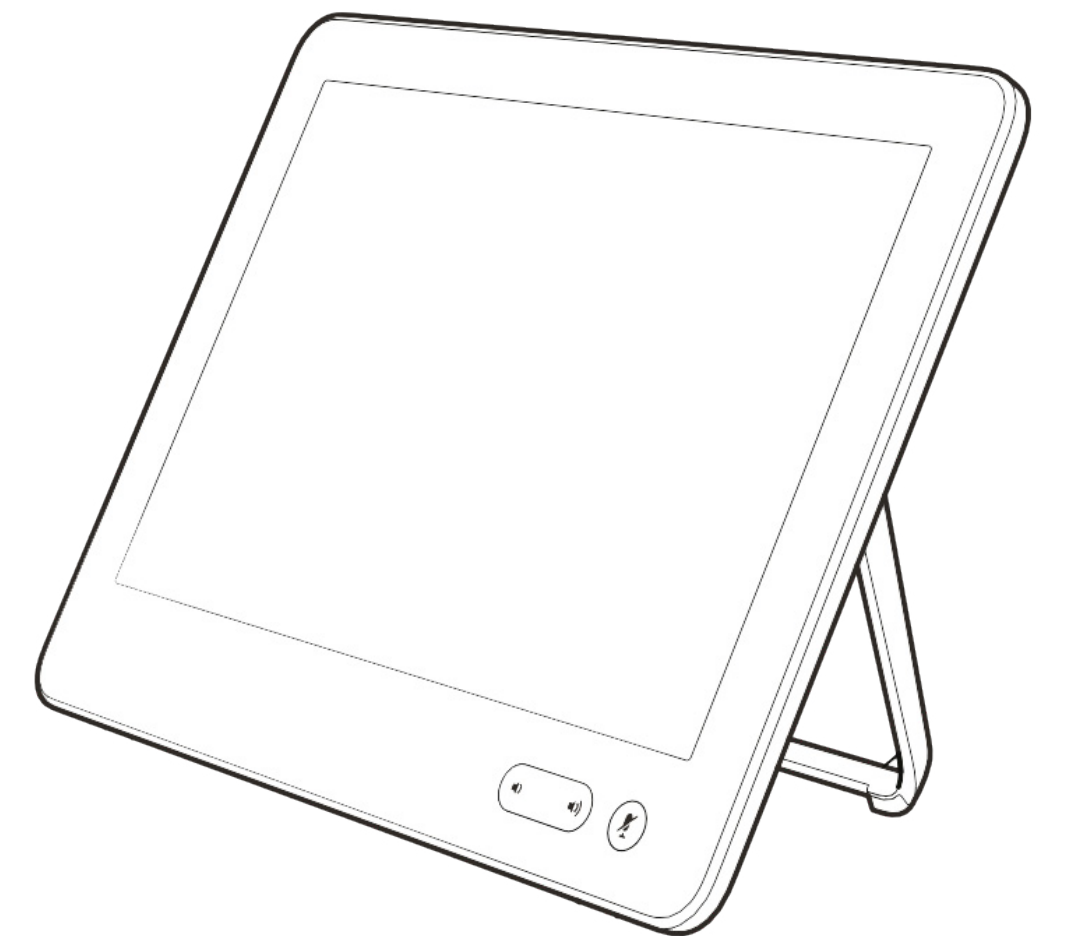
意図しない動作の変更が起こり、システムでマクロを実行する場合、トラブルシューティングを進める前に、必ずマクロを無効にします。

これを行うには、**xConfiguration Macros Mode: On/Off** を使用します。

マクロ フレームワークには、**macros.log** という名前の独自のログ ファイルがあります。

macros.log ファイルには、マクロ コンソールに出力されるものの多くが記録されます。コンソールに出力するようにマクロを設定できますが、これはログに保存されます。そのため、このファイルでカスタム ログ メッセージ (これは開発者が前もって作成することが必要です) を確認できることを覚えておいてください。

ヒントとテクニック



推奨されるベスト プラクティス

再起動後にフィードバックを取得するように再登録する

ビデオ システムまたは制御システムのいずれかが再起動した場合、制御システムは、ユーザが Touch10/DX の室内制御を使用するか新しい室内制御パネルを Touch10/DX にプッシュしたときにビデオ システムが送信するイベントに再登録する必要があります。

端末出力モードの場合:

```
xfeedback register event/UserInterface/Extensions/Widget
```

XML 出力モードの場合:

```
xfeedback register event/UserInterface/Extensions/Event
xfeedback register event/UserInterface/Extensions/Widget/
  LayoutUpdated
```

詳細については、[「API for in-room control」の章を参照してください。](#)

すべてのウィジェットを初期化する

次のような状況では、制御システムが Touch10/DX 室内制御パネル上のすべてのウィジェットを初期化していることを確認してください。

- ・ 制御システムがビデオ システムに最初に接続したとき。
- ・ ビデオ システムが再起動したとき。
- ・ 制御システムが再起動したとき。
- ・ 新しい室内制御パネルがビデオ システムにエクスポートされたとき (LayoutUpdated イベントへの応答として)。

初期化を行わないと、Touch10/DX が誤った値を表示し、室内の実際のステータスが反映されないことがあります。

初期値を設定するには、SetValue コマンドを使用します。

設定が変更されたら必ずビデオ システムに値を戻します。

予期しない動作や曖昧さを回避するため、制御システムは、設定が変更されたら必ず SetValue コマンドをビデオ システムに送信する必要があります。これは、Touch10/DX のコントロールを使用しているユーザが変更をトリガーした場合にも適用されます。

たとえば、照明を暗くするために Touch10/DX 室内制御パネルのスライダを使用しても、室内の物理的な調光器や別のタッチパネルを使用しても違いはありません。制御システムは常に、SetValue コマンドを使用して調光器の値をビデオ システムに返信する必要があります。

室内制御パネルを更新する

新しい室内制御パネルをビデオ システムにエクスポートすると、古いパネルが上書きされ、新しいパネルに置き換えられます。

更新するには、次のように実行します。

1. ビデオ システムの Web インターフェイスから室内制御エディタを起動します。
2. 必要な室内制御パネルを作成するか、または以前に保存したパネルをファイルからインポートします ([インポート (Import)] > [ファイルから (From file)])。
3. [エクスポート (Export)] > [コーデックへ (To codec)] をクリックします。

その他の便利なテクニック

以前の値を保存する。照明をオフにするときは (調光スライダとオン/オフのトグル ボタンが付いた照明など)、消灯時の状態を保存しておき、再度オンにするときにその値を使用します。

例: 明るさが 40% のときに照明をオフにして、再度オンにすると、ユーザは照明が (100% ではなく) 40% の明るさになると予想します。調光スライダの値が 0% になったときは、電源スイッチをオフに設定することも忘れないでください。

ブラインドの場合、方向矢印を短く押して、ブラインドを傾ける方法を検討してください。すでに最大限傾いている場合は、短く押すと、ブラインドがわずかに動きます。

方向矢印を長押しすると、ブラインドがその方向に動き始めます。完全に上がるか下がるまでは停止しません。

長押し後に動きを止めるには、いずれかのボタンを短く押します (停止ボタンは必要ありません)。

室内制御パネルとアイコン全体を削除する

ビデオ システムに室内制御パネルがある場合、対応する室内制御アイコンが、Touch 10/DX のステータス バー上に表示されるか、またはコール制御ボタンの右側にボタンとして表示されます。パネルが空で、ウィジェットを含まない場合でも、アイコンとパネルの両方が表示されます。

Touch10/DX から室内制御パネルとアイコンを削除するには、次のようにします。

1. ビデオ システムの Web インターフェイスから室内制御エディタを起動します。
2. 削除するパネルを選択します ([グローバル (Global)], [ホーム画面 (Homescreen)], または [通話中 (In-call)])。
3. [パネルの削除 (Delete panel)] をクリックします。

サードパーティ製の制御システムから CE への移行

サードパーティの制御システムをすでに使用している場合、本書の説明に従って CE の使用を開始するには、次のようにすることをお勧めします。

- ・ サードパーティ製の機器を制御するために作成したプログラミング コードをそのまま残します。
- ・ シスコのビデオ エンドポイントを制御するすべてのコードを削除します。このエンドポイントは、すでに Touch10/DX を介して制御されているためです。
- ・ サードパーティ制御システム パネルからのボタン押下によるシグナリングを再プログラムして、代わりに Touch10/DX からのボタン押下をリッスンするようにします。

多くの場合、このプログラミングは非常に簡単です。プログラミングを非常に容易に始められる室内制御用のモジュール/ドライバが主な制御システム製造元から提供されているためです。

推奨されるベスト プラクティス (続き)

ヒントとなる例

次の例は参考例として、またベスト プラクティスのガイダンスとして用いることができます。これらの例に示す方法でコントロールを設計して実装することは、決して必須ではありません。

ウィジェット ID

ウィジェット (ボタンなど) をページにドラッグする場合は、そのウィジェットにカスタマイズした ID を付与できます。ウィジェット ID は一意である必要はありません。ウィジェット間で ID を共有できます。しかし、その場合は同じタイプのウィジェットである必要があります。つまり、異なるパネルで「メイン照明」という名前の 2 つのスライダを作成することはできますが、1 つをスライダ、他方をトグル ボタンにして両方に「メイン照明」と名前を付けることはできません。

別のページまたはパネルの既存のウィジェットの複製を作成するには、コピーして貼り付けます。

同じ場所にあるものでグループを作成する

同じページにある同種のコントロールはグループ化することを検討します。室内制御エディタで作成したページは、制御パネルで別々のタブとして表示されます。

照明の制御

スライダとトグル ボタンを組み合わせ使用し、照明を制御することができます。トグル ボタンで照明のオン/オフを切り替えて、スライダを調光器として使用します。

次の方法を検討します。

- ・ ユーザが照明をオフにしたら、スライダを最小レベルに設定します。
- ・ ユーザがスライダを最小レベルに移動したら、トグル ボタンをオフに設定します。
- ・ 照明がオフになったときのスライダの値を保存しておき、再びオンになったときに元の値に戻すことができますようにします。
- ・ 明るさが 40 % のときに照明をオフにして、もう一度オンにすると、ユーザは照明が (最大の明るさではなく) 40 % の明るさになると予想します。
- ・ ユーザがグループ ボタン (照明のプリセット) でオプションの 1 つを選択したら、それに応じてスライダとトグル ボタンを設定します。
- ・ たとえば、スライダまたはトグル ボタンを操作して照明がプリセットから変更された場合は、グループ ボタンのすべてのオプションを選択解除します。

温度の制御

スピン ボックスとフォントが大きいテキスト ボックス (値) を組み合わせて使用し、温度を制御することができます。スピン ボックスを使用して希望の温度を設定し、フォントが大きいテキスト ボックスに現在の温度を表示します。

最良のユーザ エクスペリエンスを実現するには、以下の点を念頭に置いてください。

- ・ 室温が変化したら、フォントが大きいテキスト ボックスの表示内容を更新します。上矢印や下矢印がタップされたら、スピン ボックスのテキスト フィールドを更新します。
- ・ スピン ボックスのテキスト フィールドとフォントが大きいテキスト ボックスを更新する方法の詳細については、「[Widgets 1 の章を参照してください。](#)」

ブラインドの制御

スピン ボックスを使用するか、またはウィジェット ライブラリの [アイコン (Icons)] タブから上矢印と下矢印を使用することができます。

次の方法を検討します。

- ・ 方向矢印を短く押すと、ブラインドが傾きます。すでに最大限傾いている場合は、ブラインドが段階的に上下に動きます。
- ・ 方向矢印を長押しすると、ブラインドがその方向に動き始めます。完全に上がるか下がるまで停止しません。
- ・ 長押し後の動きを止めるには、いずれかのボタンを短く押します。別個の停止ボタンは必要ありません。

グループ ボタン

グループ ボタン (ラジオ ボタン) は、複数のボタンをリンクして同時に 1 つのボタンだけを選択できるようにする場合に最適です。たとえば、ルーム プリセットに使用できます。グループ内の個々のボタンが、それぞれの機能を説明するテキストには小さすぎる場合は、説明用のテキスト ボックスを別に使用することを検討してください。

スピード ダイアルまたはワン ボタン プッシュの作成

Touch10 でパネルや設定を表示せずに直接アクションを実行するボタンの作成を行いたい場合は、次のようにします。

- ・ 新しいパネルを作成し、パネル ID を指定した後、[\[ページの削除 \(Delete page\)\]](#) をクリックすると、パネルにページが含まれなくなります。パネル ID はイベントを生成するために必要です。

Touch10 でボタンをタップすることで直接イベントの作成が行われ、それを使ってアクションの開始に利用することができます。本機能における一般的な使用例として、スピード ダイアルまたはワン ボタン プッシュ ソリューションがあります。

Touch10 のボタンに、ユーザがわかりやすい名前を与えることを覚えておいてください。

室内制御エディタと拡張機能 API へのアクセス許可

室内制御エディタにアクセスするには、管理者権限が必要です。

ただし、管理者は、室内制御用のユーザ アカウントを作成することができます。このアカウントを使用すると、コーデックにログインして室内制御エディタを実行できます。Web インターフェイスの他の部分には、このアカウントではアクセスできません。

SSH を使用してコーデックにログインした場合は、ごく一部の API セットのみアクセスできます。

パート 2 室内シミュレータ



室内制御の機能を示す目的で、シミュレータが用意されています。シミュレータの目的は 2 つあります。販売ツールとしての目的もありますが、コマンドやイベントがどのように実行されるか知りたい場合にも役に立ちます。

シミュレータは、シミュレートした独自のサードパーティ製制御システムを組み込みます。

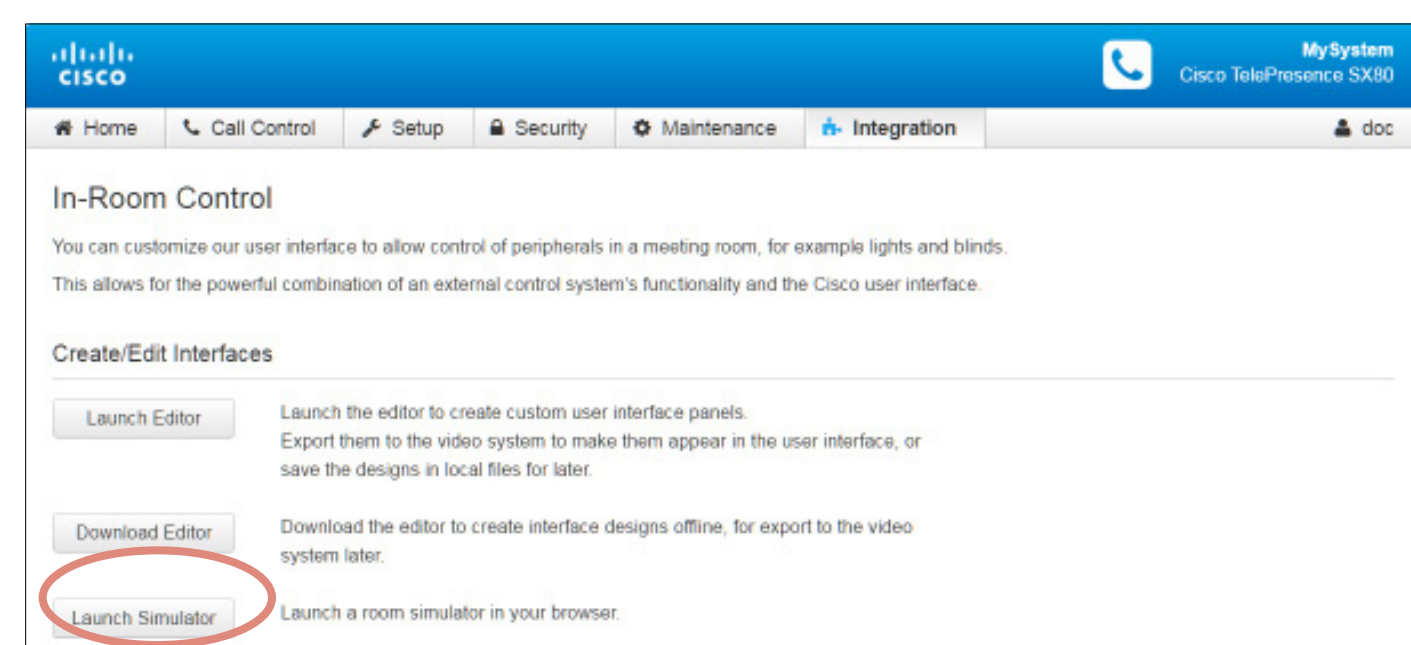
販売ツールとしての使用

全容を示すための販売ツールとして使用する場合、シミュレータは事前定義された室内制御の設定を使用します。これは変更できません。事前定義された設定は、コーデックの設定を上書きします。現在の設定はブラウザ内にバックアップされ、セッション後に再インストールされます(これを行ってもよいことを確認するよう求められます)。

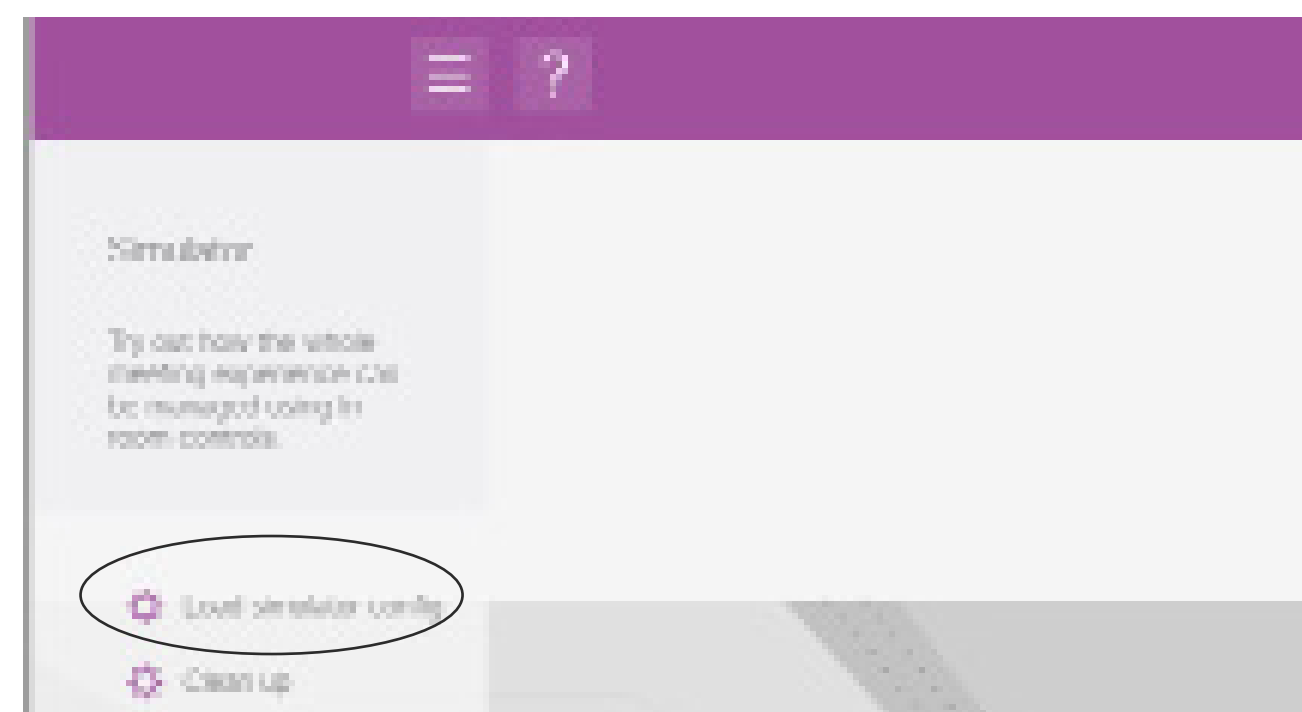
注: シミュレータは自己完結型であるため、競合を避けるために、シミュレータを実行するときはコーデックに接続されているすべてのコントローラを切断する必要があります。

ヒント: シミュレータを実行するときには Google Chrome ブラウザを使用することをお勧めします。他のブラウザでは、シミュレータが正しく稼働しない可能性があります。

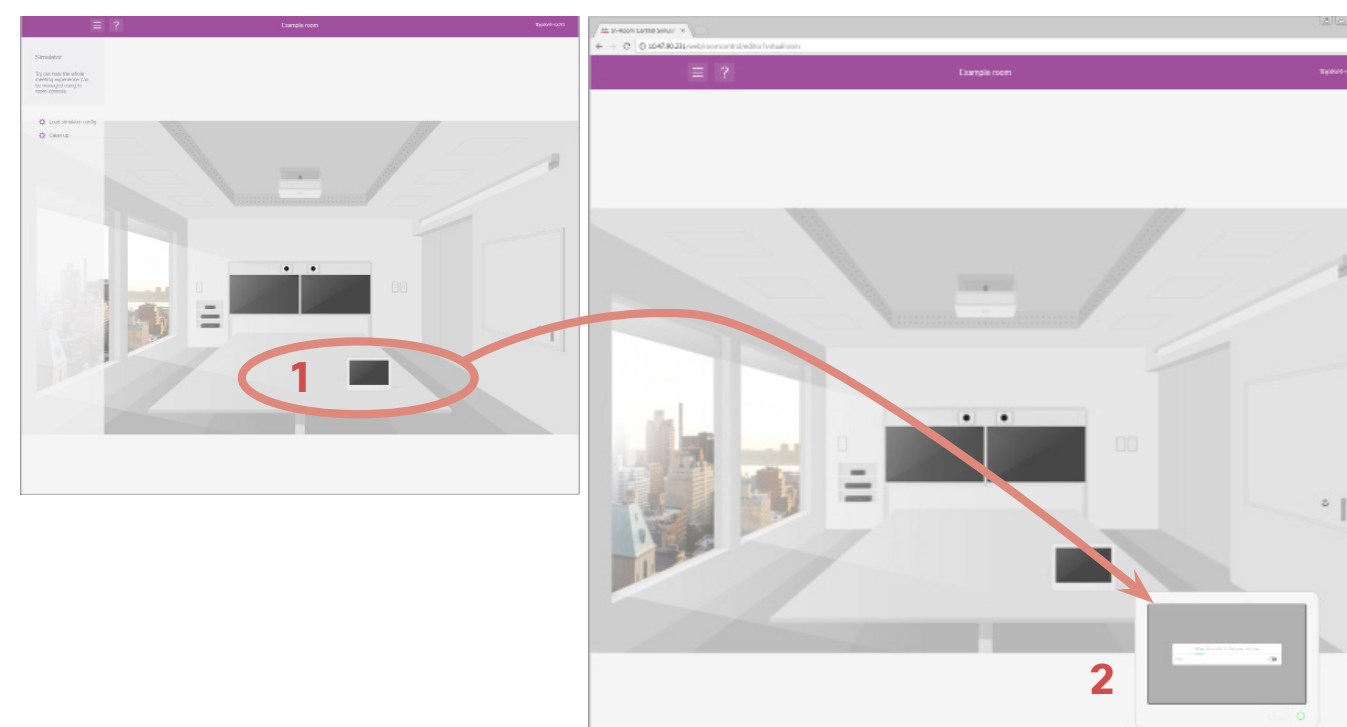
シミュレータを起動するには、Web インターフェイスで [統合 (Integration)] > [室内制御 (In-Room Control)] に移動し、[シミュレータの起動 (Launch Simulator)] を選択します。



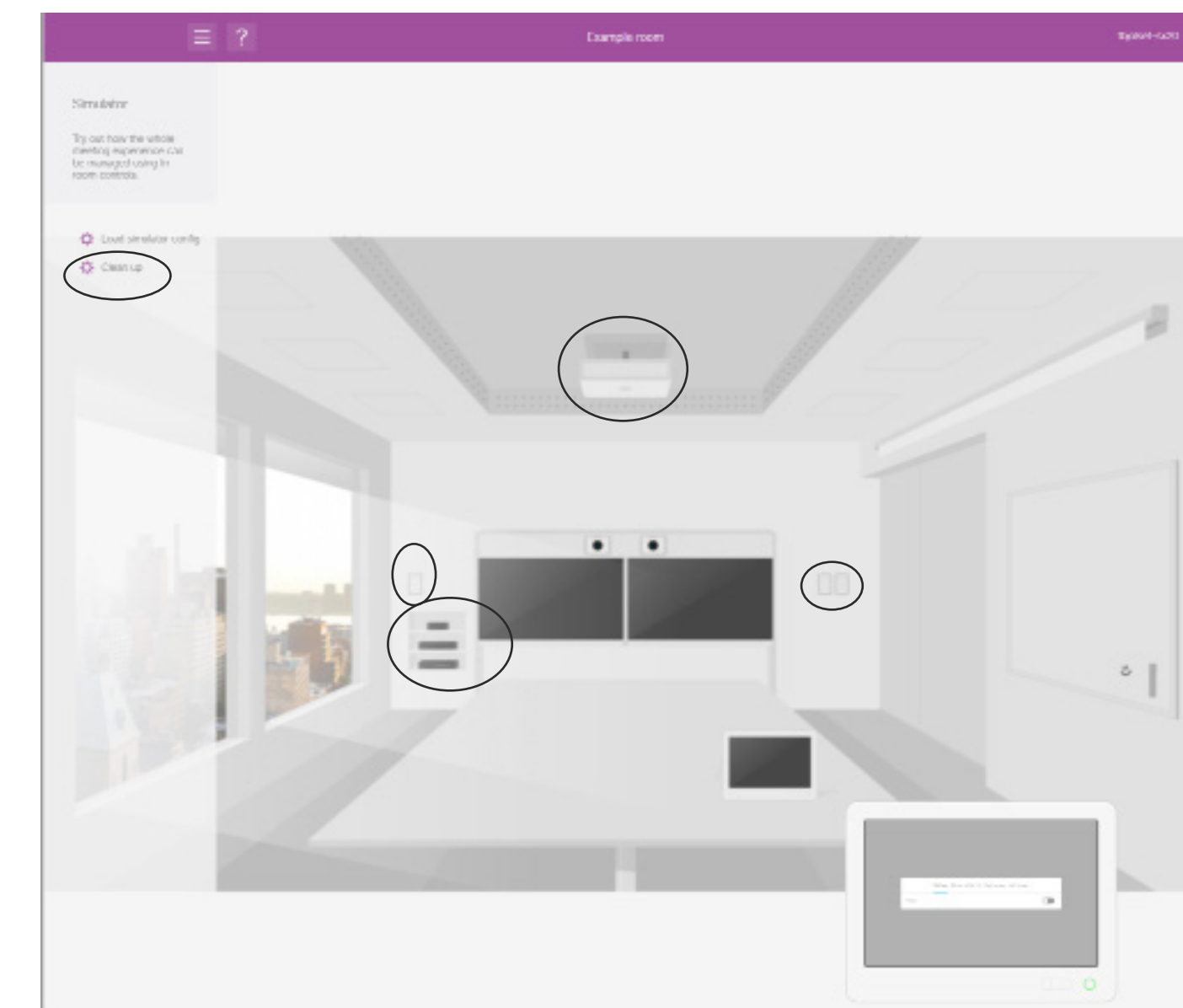
[サンプル ルーム (Example Room)] が表示されます。[シミュレータの設定を読み込む (Load Simulator Config)] をクリックします。読み込みに成功したら、必要に応じて任意の場所をクリックするとメッセージが除去されます。



Touch10(1)をクリックして拡大し、(2)に示されるようにクリックして室内制御を起動します。



これで、仮想 Touch 10 上のコントロールと室内のスイッチ自体を使って操作できるようになります。[クリーンアップ (Clean up)] をクリックして終了し、以前の設定を復元します。

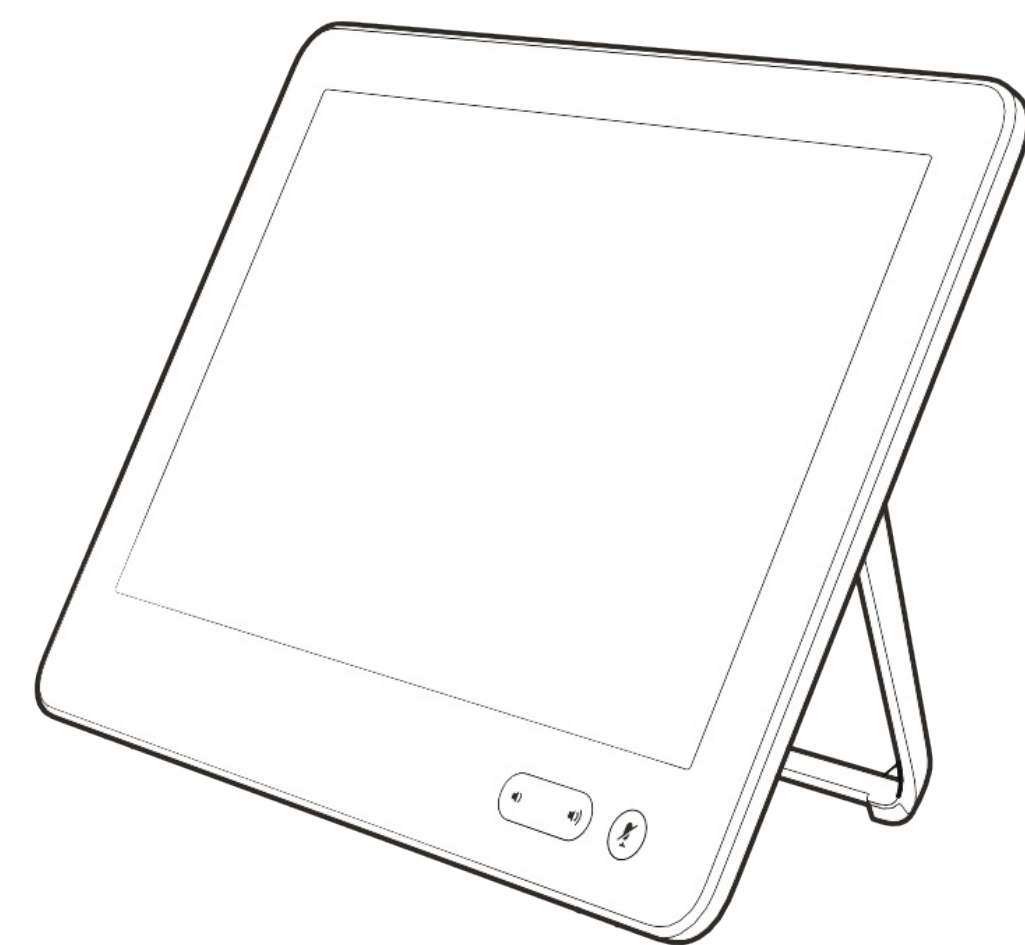


プレビュー用のシミュレータとして使用

設定の効果をプレビューしたり、Touch10 とシミュレートしたサードパーティ製の制御システムとの間の情報交換をモニタしたりするためにもシミュレータを使用できます。シミュレータにシミュレータ設定を読み込んでいないことを確認してください。必要に応じて、コーデック設定を復元します(上記のとおり [クリーンアップ (Clean up)] を使用します)。

シミュレータ プレビューの使用法については、“現在の設定のプレビュー”(ページ 10 ページ)で説明しています。

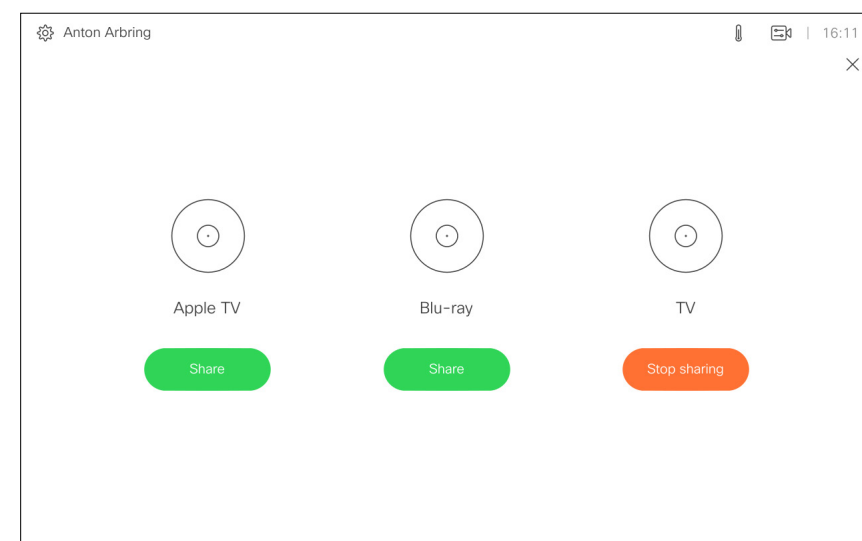
パート 3 ビデオ スイッチの使用



サードパーティ製ビデオ スイッチを使って使用可能なビデオ ソース数を拡張する

Cisco Touch10 パネルでは、サードパーティの外部ビデオ スイッチからのビデオ ソースを通常の [共有トレイ (Share Tray)] ビューに表示するよう設定することが可能になりました。

これらのソースは、コーデックに直接接続された他のビデオ ソースと同じように表示され、動作します。ユーザには、完全に透過的である、つまりビデオ スイッチが組み込まれていないように感じられるでしょう。



ビデオ スイッチ機能は、室内制御と同様にサードパーティの制御システムを必要とします。制御システムはコーデック API を使用し、いくつかの API イベントとコマンドを使ってビデオ スイッチと Touch10 ユーザ インターフェイスの間でソース状態を同期させます。

ユーザが Touch 10 でビデオ ソースを選択したときにこれが機能するためには、対応するイベントを発行するようコーデックを設定する必要があります。これにより、コントローラは適切なコマンドをビデオ スイッチとコーデックに送信できます。

このイベントが発行されるのは、コーデックから次の要求を行うよう、接続時にコントローラがコーデックに登録した場合だけです。

```
xFeedback register Event/UserInterface/Presentation/
ExternalSource
```

発行されるイベントは次のとおりです。

```
*e UserInterface Presentation ExternalSource Selected
SourceIdentifier: "XXXX"
```

ここで "XXXX" は、状態を選択または設定するときこのソースを識別するための固有の文字列 ID です。詳細については以降のページを参照してください。

さらに、システムを制御するために次の 6 つのコマンドを使用できます。

Add:ビデオ ソース識別子を追加します。これにはコネクタの ID、Touch10 に表示される名前、状態を選択または設定するときソースを識別するための固有文字列 ID、ソースごとに Touch10 に表示するアイコンの種類が含まれます。

List:現在の外部ソースのリストを返します。

Remove:リストからソースを削除します。

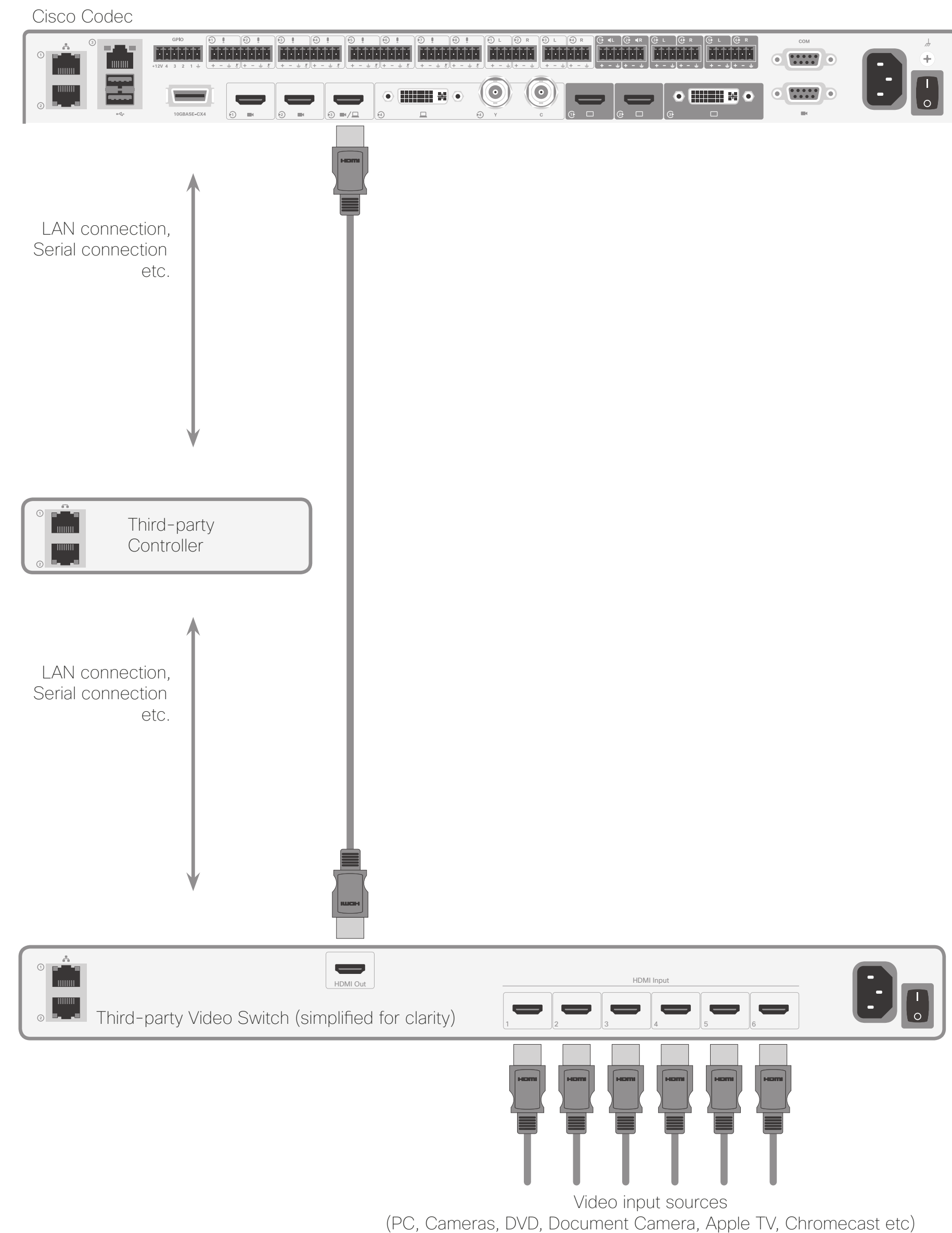
RemoveAll:リストからすべてのソースを削除します。

Select:特定のソースを選択します。

State Set:ソースの状態を変更します。

これらの詳細については、以降のページで説明します。

上記の設定を使用した簡単なセットアップの例が「[ビデオ スイッチの例](#)」(「[ビデオ スイッチの例](#)」(ページ 46 ページ))に掲載されています。



UserInterface Presentation ExternalSource Add コマンド

このコマンドは入力ソースを確立して定義します。

```
xcommand UserInterface Presentation ExternalSource Add
ConnectorId: ConnectorId Name: Name SourceIdentifier:
SourceIdentifier Type: Type
```

ここで、

ConnectorId:外部スイッチの接続先であるコーデック コネクタの ID

Name:Touch10 に表示される名前

SourceIdentifier:状態を選択または設定するときこのソースを識別するための固有の文字列 ID

Type:Touch10 に表示されるアイコンを決定します。<pc/camera/desktop/document_camera/mediaplayer/other/whiteboard> から選択してください。

例:

```
xcommand UserInterface Presentation ExternalSource Add
ConnectorId: 3 Name: "Blu-ray"
SourceIdentifier: bluray Type: mediaplayer
```

UserInterface Presentation ExternalSource List コマンド

このコマンドは、現在の外部ソースのリストを返します。

```
xcommand UserInterface Presentation ExternalSource List
```

UserInterface Presentation ExternalSource Remove コマンド

このコマンドは、リストからソースを削除します。

```
xcommand UserInterface Presentation ExternalSource Remove
SourceIdentifier: SourceIdentifier
```

ここで、

SourceIdentifier は、状態を選択または設定するときこのソースを識別するための固有の文字列 ID です。

UserInterface Presentation ExternalSource RemoveAll コマンド

このコマンドは、リストからすべてのソースを削除します。

```
xcommand UserInterface Presentation ExternalSource RemoveAll
```

UserInterface Presentation ExternalSource Select コマンド

選択したソースが準備完了状態で、有効な ConnectorId を持っている場合、選択されたソースを表示し始めます。また、アイテムを共有トレイに「表示中 (Presenting)」として表示します。

```
xcommand UserInterface Presentation ExternalSource Select
SourceIdentifier: SourceIdentifier
```

ここで、

SourceIdentifier は、状態を選択または設定するときこのソースを識別するための固有の文字列 ID です。

UserInterface Presentation ExternalSource State Set コマンド

SourceIdentifier を持つソースの状態を変更するために使われます。

```
xcommand UserInterface Presentation ExternalSource State Set
SourceIdentifier: SourceIdentifier State: State
[ErrorReason: ErrorReason]
```

ここで、

SourceIdentifier は、状態を選択または設定するときこのソースを識別するための固有の文字列 ID です。

State:<Error/Hidden/NotReady/Ready> 表示可能な状態は Ready(準備完了)のみです。リストには Hidden(非表示)が含まれていますが、共有トレイには表示されません。

ErrorReason:オプション。状態が Error に設定されている場合に、共有トレイに表示されます。

パート 3:ビデオ スイッチの使用 ビデオ スイッチの例

簡単な設定例を次に示します。

コントローラが次のものを送信します：

```
xcommand UserInterface Presentation ExternalSource Add ConnectorId: 3
  Name: "Blu-ray" SourceIdentifier: bluray Type: mediaplayer
```

```
xcommand UserInterface Presentation ExternalSource Add ConnectorId: 3
  Name: "Apple TV" SourceIdentifier: appletv Type: mediaplayer
```

```
xcommand UserInterface Presentation ExternalSource Add ConnectorId: 3
  Name: "TV" SourceIdentifier: tv Type: mediaplayer
```

デフォルト状態は NotReady です(図 1)

インテグレータは次のステップとして、これらを Ready に設定できます(図 2)。

```
xcommand UserInterface Presentation ExternalSource State Set State:
  Ready SourceIdentifier: bluray
```

```
xcommand UserInterface Presentation ExternalSource State Set State:
  Ready SourceIdentifier: appletv
```

```
xcommand UserInterface Presentation ExternalSource State Set State:
  Ready SourceIdentifier: tv
```

ビデオ スイッチでいずれかのソースが選択された場合、コントローラはそれに応じてコマンドを送信する必要があります：

```
xcommand UserInterface Presentation ExternalSource Select
  SourceIdentifier: tv
```

選択されたコネクタでスイッチが接続されると、表示が開始されます(図 3)。

ユーザが共有トレイで別のソース項目をクリックして別のソースを選択すると、コーデックは次のイベントを送信します。

```
*e UserInterface Presentation ExternalSource Selected
  SourceIdentifier: "appletv"
```

コントローラはこのイベントをリッスンして、選択されたソースを表示する必要があります。

注:更新 以下の設定がマニュアルになっている場合、プレゼンテーションは開始されません。

```
xconfiguration Video Input Connector [x]
  PresentationSelection: <AutoShare, Desktop, Manual, OnConnect>
```

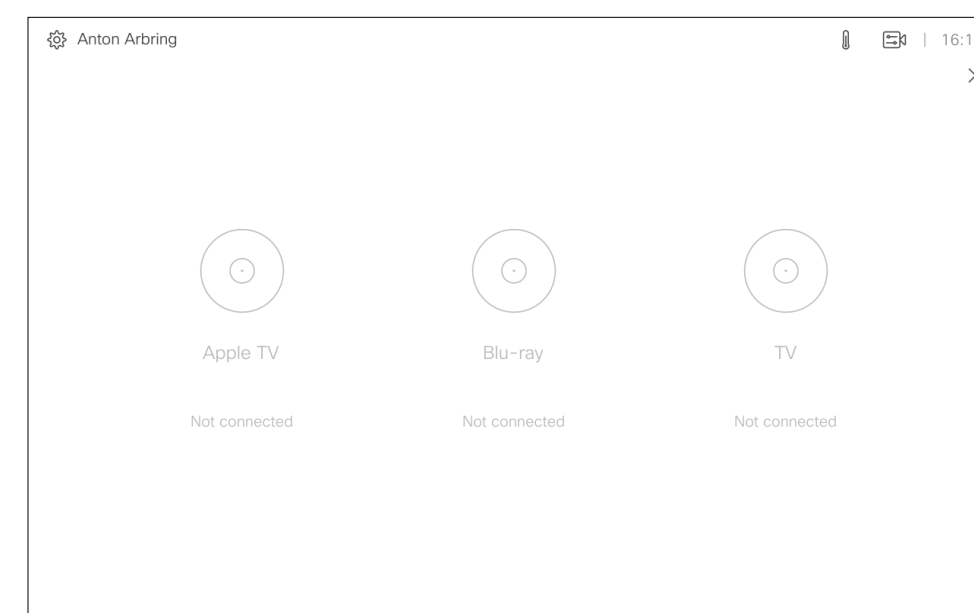


図 1: デフォルト状態は NotReady。

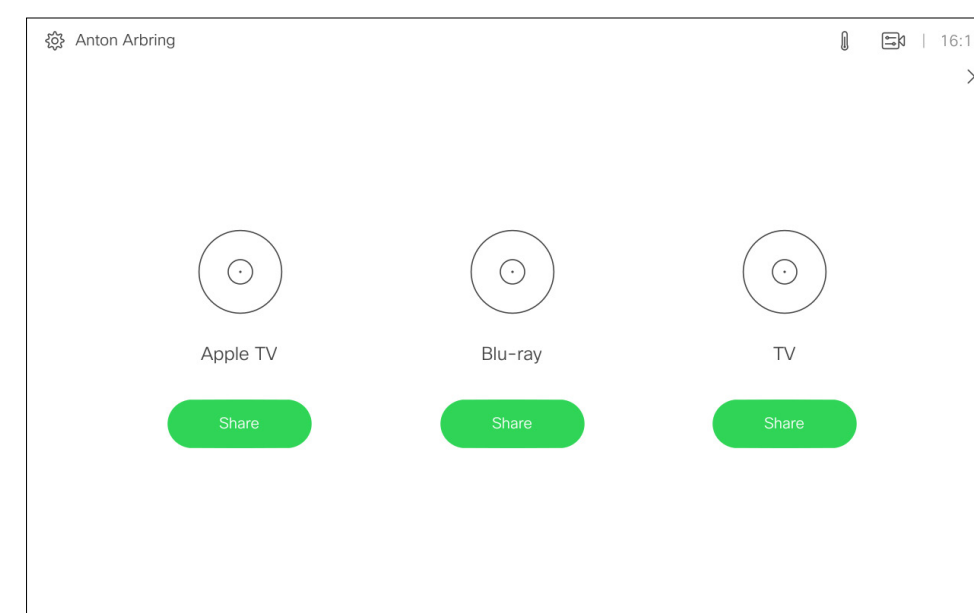


図 2: 入力ソースが Ready に設定されたところ。

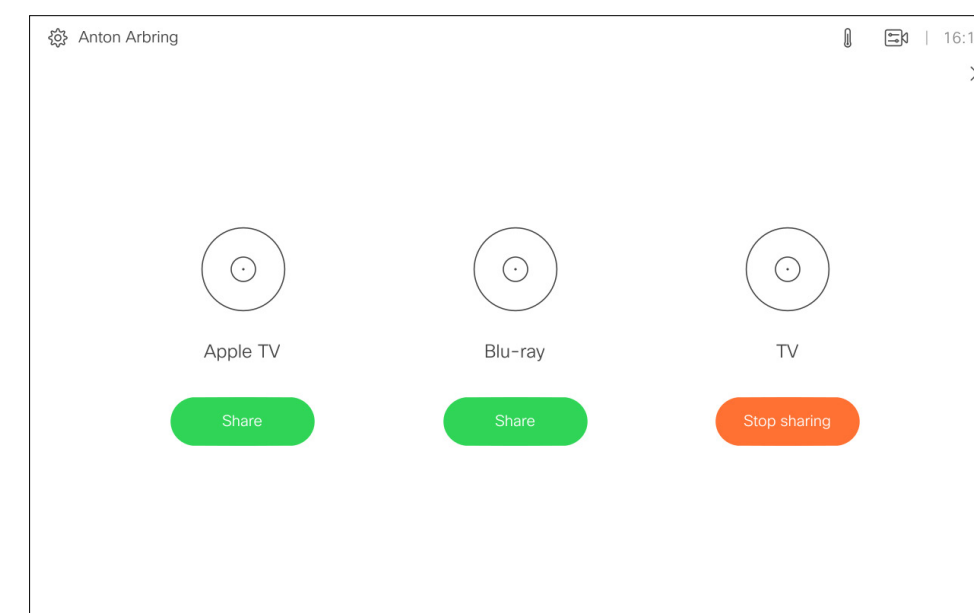
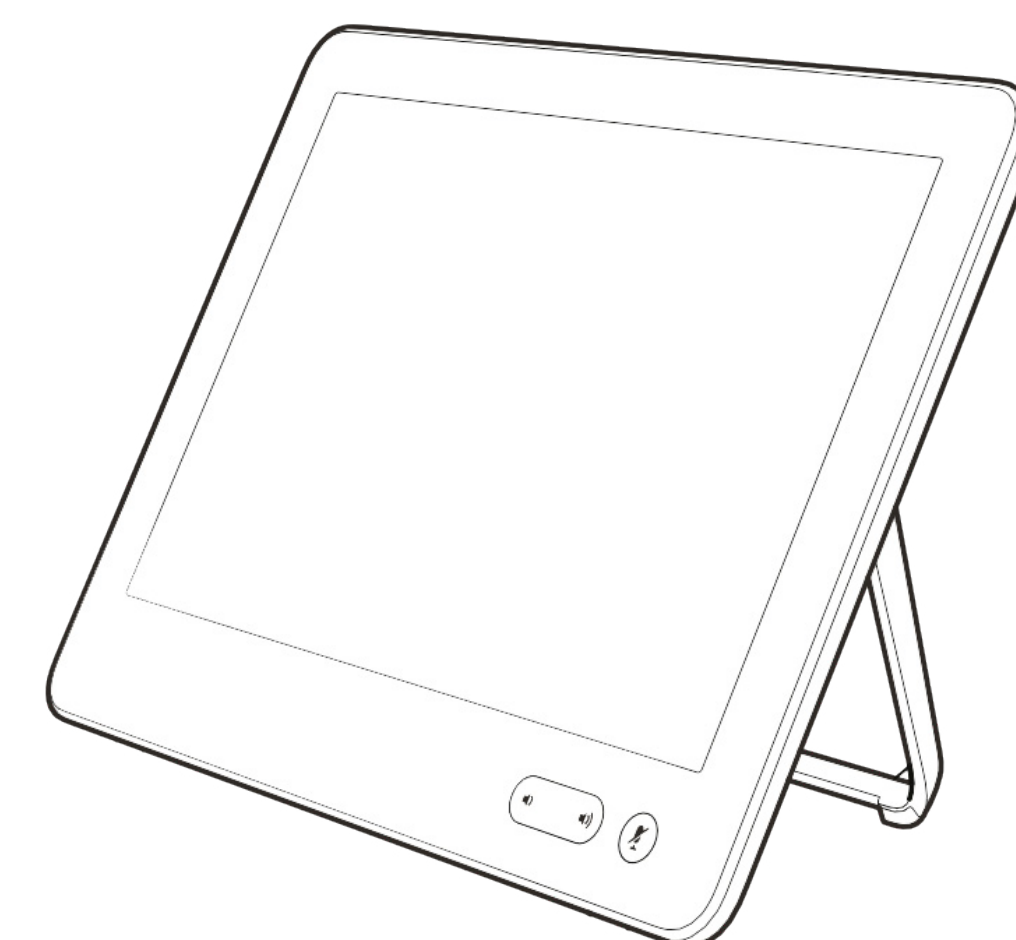


図 3: 選択されたコネクタでスイッチが接続されると、表示が始まる。

パート 4 マクロでの操作



パート 4: マクロでの操作

マクロの作成

マクロによって、ビデオ エンドポイントの一部を自動化することができます。JavaScript コードのスニペットを記述できます。このようにしてカスタム動作を作成します。

以下をご確認ください。

- ・ SX10 はマクロをサポートしていません。
- ・ Cisco WebEx(旧称 Cisco Spark)が有効になっているシステムでは、マクロは機能プレビューです。

ビデオ システムの Web インターフェイスに管理者クレデンシャルでサインインし、[統合(Integration)] > [マクロ エディタ(Macro Editor)] に移動します。

初めてこれをコードブックで実行するとき、このコードブックでマクロの使用を有効にするかどうか尋ねられます。

後でマクロ エディタからマクロの使用を無効にすることもできます。この方法では、マクロの実行は恒久的には無効にされません。コードブックが再起動されるたびに、マクロは自動的に再度有効にされます。

自動再起動を無効にするには、xConfiguration Macros Mode: Off を使用する必要があります。

システムによる予期しない動作が発生する場合に、このコマンドを使用することができます。このような場合、トラブルシューティングを進める前に、必ずマクロを無効にする必要があります。

マクロ フレームワークには多くの利点があります。インテグレータで以下を行えます。

- ・ 展開の調整。
- ・ 独自の「機能」、またはシスコが新しいソフトウェア機能の形で提供しない機能に対する「回避策」の作成。
- ・ シナリオまたは再構成の自動化。
- ・ カスタム テストまたはモニタリングの作成。

マクロを使用する場合、室内制御でローカル機能を有効にするための外部制御システムは不要になりました。

ただし、照明やブラインドといった他の機器を制御するなど、xAPI 経由でローカル アクションを実行する場合は、適切なサードパーティ製の制御システムが必要になります。

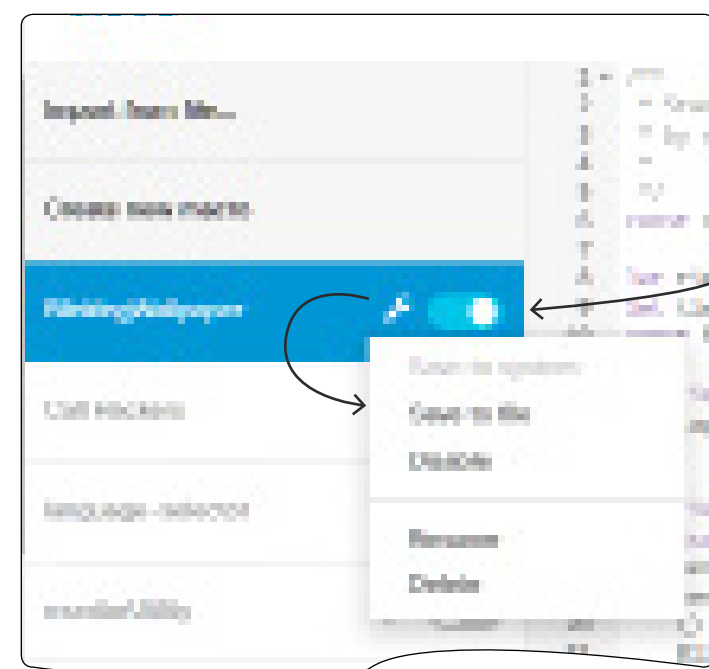
ローカル機能の例には、スピード ダイアルのため、またはすべての設定(入力ソース、カメラのプリセットなど)をデフォルトに戻す「ルーム リセット」をトリガーするための室内制御パネルがあります。

注: マクロ フレームワークは、ローカル xAPI の対話に限定されています。マクロ フレームワーク コードを介してデータをプッシュまたは受信するサーバへのリモート ネットワーク接続を確立することはできません。

免責事項: シスコはマクロ フレームワーク自体のみをサポートします。シスコでは、コンパイルに失敗するコード、または動作しないことが開発者の「意図」どおりであるコードはサポートしません。構文が正しいこと、また JavaScript でマクロを記述するための十分なコーディングスキルの取得は、コードを記述する人の責任になります。

開発者のパブリック フォーラム、問題になっている製品の API リファレンス ガイド、およびマクロ エディタのヘルプ セクションを参照してください。詳細について、以下のページも参照してください。

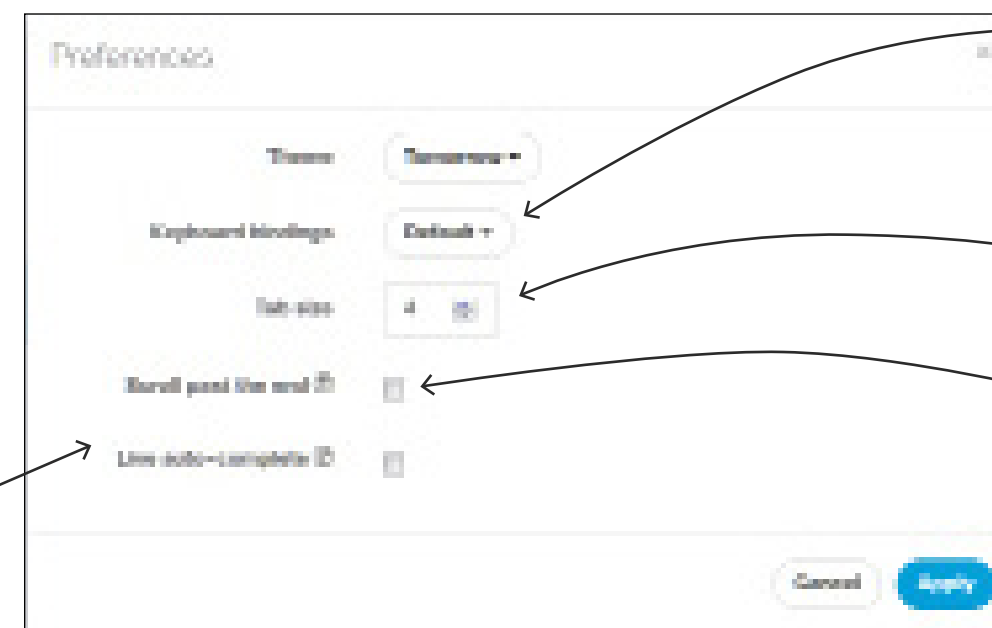
パート 4: マクロでの操作 マクロ エディタ パネル



マクロのアクティブ/非アクティブを切り替えます。

マクロの開始、停止、再開に使用します。再起動後、マクロは自動的に再度有効にされます。詳細については、前のページを参照してください。

[ログの切り替え (Toggle log)] は、ログ コンソールの表示/非表示を切り替えます。



使用するキーボード ショートカットのセットを定義します。[Vim] または [Emacs] のいずれかを選択します。

タブの幅を定義します (スペース数単位)。

[最後までスクロールする (Scroll past the end)] をチェックすると、マクロの末尾が画面の中央で維持されます。

上記は、新しいマクロのプログラミング セッションを開始する場所を示しています。[ファイルからインポート... (Import from file...)] をクリックして、ファイル (*.js) から既存のコードをインポートできます。または、[新規マクロの作成 (Create new macro)] をクリックして、空のマクロを作成できます。

[新規マクロの作成 (Create new macro)] をクリックすると、メインのエディタ ウィンドウがアクティブになります。ここからコーディングを開始できます。

マクロは、マクロの一覧に表示されます。

上に示す図のように、レンチ アイコンをクリックして、使用可能なオプションにアクセスできるようにします。

マクロの名前を変更するには、名前をクリックして編集し、Enter キーを押す方法もあります。

マクロの保存、削除、またはアクティブ/非アクティブの切り替えを行うと、ランタイム全体 (つまり、アクティブなマクロすべて) が再起動されます。ランタイムの詳細については、次のページを参照してください。

ログ コンソールの目的は、マクロを実行すると何が起るかを明らかにすることです。ここには、ランタイムのアクション、およびコンソールに出力することにした内容すべてが表示されます。

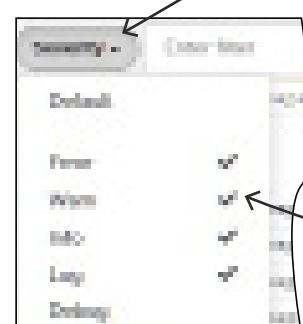
コンソール ログ ウィンドウに表示される内容の多くは、エンドポイントのログ バンドル内の macros.log ファイルにエクスポートされます。

これは、コードのエラーおよび例外を明らかにするために使用できます。次のコマンドを発行して、カスタム テキストをログに記録することもできます。

```
console.log('this is a log entry');
```

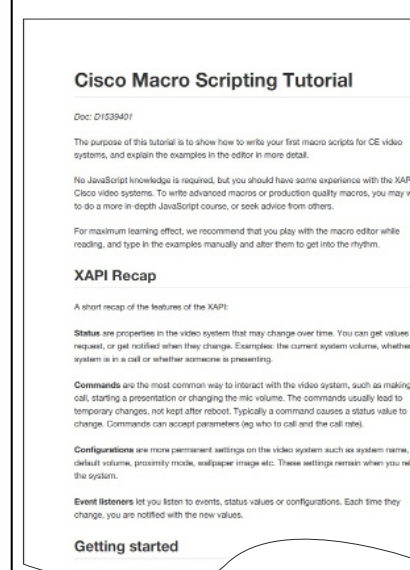
キー入力によりログを検索します。

ログ コンソール



ログに記録したいエラーや例外のタイプがチェックされていることをここで確認します。チェックされていない場合は、コンソール ログ ウィンドウに表示されません。

マクロの概要
マクロのチュートリアル



[チュートリアル (Tutorial)] をクリックすると、PDF をダウンロードするが尋ねられます。

ここで、JavaScript コードを記述します。JavaScript ライブラリが一部の領域で限られているように思える可能性があります。これは、特定のシナリオを防ぐために意図的に行われています。

エディタは自動的に構文エラーを検出し、スクリプトに検出されたエラーがある場合、コードを保存しないようにします。エラーの上にカーソルを動かし、詳細を確認します。

すべてのサンプル マクロについて、[サンプルのロード (Load Example)] をクリックし、メイン エディタ ウィンドウに直接コードを貼り付けることができます。

注: この操作を行うアクティブな編集セッションで、左側のメニューから [新規マクロの作成 (Create new macro)] をクリックし、[サンプルのロード (Load Example)] をクリックする必要があります。

キーボードのショートカットを使用できます。

パート 4: マクロでの操作 確認すべき事柄

マクロは、システムの期待どおりの動作を変更する場合があります。作成したマクロが通常のユーザを驚かせたり、混乱を生じさせたりする可能性があるならば、ビデオ システム画面の通知などを通してそれらの変更について知らせます。

マクロを他のマクロに依存させないようにすることをお勧めします。もちろん、複数のマクロのアクションが相互に依存しない限り、呼び出し状態など同じ xapi 値をリッスンする複数のマクロを作成することはできます。

大量のマクロを使用すると、重い負荷がかかりコーデックのパフォーマンスが低下する可能性があることに注意してください。

現在マクロで、部屋の照明を制御するなど、外部からデータを取得したり、外部に送信したりすることはできません。これを行うには、外部制御システムが必要になります。マクロと外部システムを組み合わせることによって、使用方法の幅を広げることができます。たとえば、単純な照明の制御に *Crestron* を使用し、プレゼンテーションやコールの状態に応じて照明を調整するなど、より高度な方法での調整にマクロを使用することができます。

Touch10 / DX に表示されるカスタマイズしたテキストをマクロに含めることができます。このテキストは、特定の機能のアクティブ化/非アクティブ化を確認するようユーザに警告したり、メッセージに従って行動するようユーザに注意を喚起したりするために使用できます。

このようなテキストによって純粋に情報を提供することができますが、情報を入力して応答するようユーザに求めることもできます。また、この情報に基づいて、ビデオ システムに直接動作させることもできます。マクロは外部からデータを取得したり、外部にデータを送信したりできないため、事前定義した URL などのコーデック外部に関連情報を送信するために、この機能を使用することはできません。

マクロ ランタイムについて

アクティブ化されたすべてのマクロは、マクロ ランタイムと呼ばれる、ビデオ エンドポイントの単一プロセスで実行されます。デフォルトで実行されますが、エディタから手動でそれを停止したり、開始したりすることができます。ビデオ エンドポイントを再起動した場合、xconfiguration macros autostart が On であれば、ランタイムは自動的に再開されます。

マクロが応答しなくなると(無限ループなどのために数秒間応答できなくなることがあります)、安全メカニズムによってランタイムが停止され、それにより、すべてのマクロも停止します。

ランタイムは、数秒後に自動的に再起動されます。この動作は継続しますが、ランタイムがシャット ダウンされるたびに、再起動までの時間が長くなります。これが一定回数以上発生すると、マクロに問題があることを通知するためにシステム診断が表示されます。

利用可能な学習リソース

マクロ機能の活用方法について学習する場合には、以下の資料をお勧めします。

- マクロ エディタのヘルプ セクションで見つけることができる「マクロの概要」をご覧ください。
- マクロ エディタにある「マクロのチュートリアル」をご覧ください。このチュートリアルは、cisco.com から無料でダウンロードすることもできます。
- マクロ エディタには、すぐに使用できるいくつかのサンプルも含まれています。これらすべてのサンプルは、エディタに読み込むことも、そこで学ぶことも、ご使用の設定でそのまま使用することもできます。もちろん、必要に応じて改良するための基盤として用いることもできます。

トラブルシューティングを行うときはマクロを無効にする

意図しない動作の変更が起これ、システムでマクロを実行する場合、トラブルシューティングを進める前に、必ずマクロを無効にします。

これを行うには、xConfiguration Macros Mode: On/Off を使用します。

マクロ フレームワークには、macros.log という名前の独自のログファイルがあります。

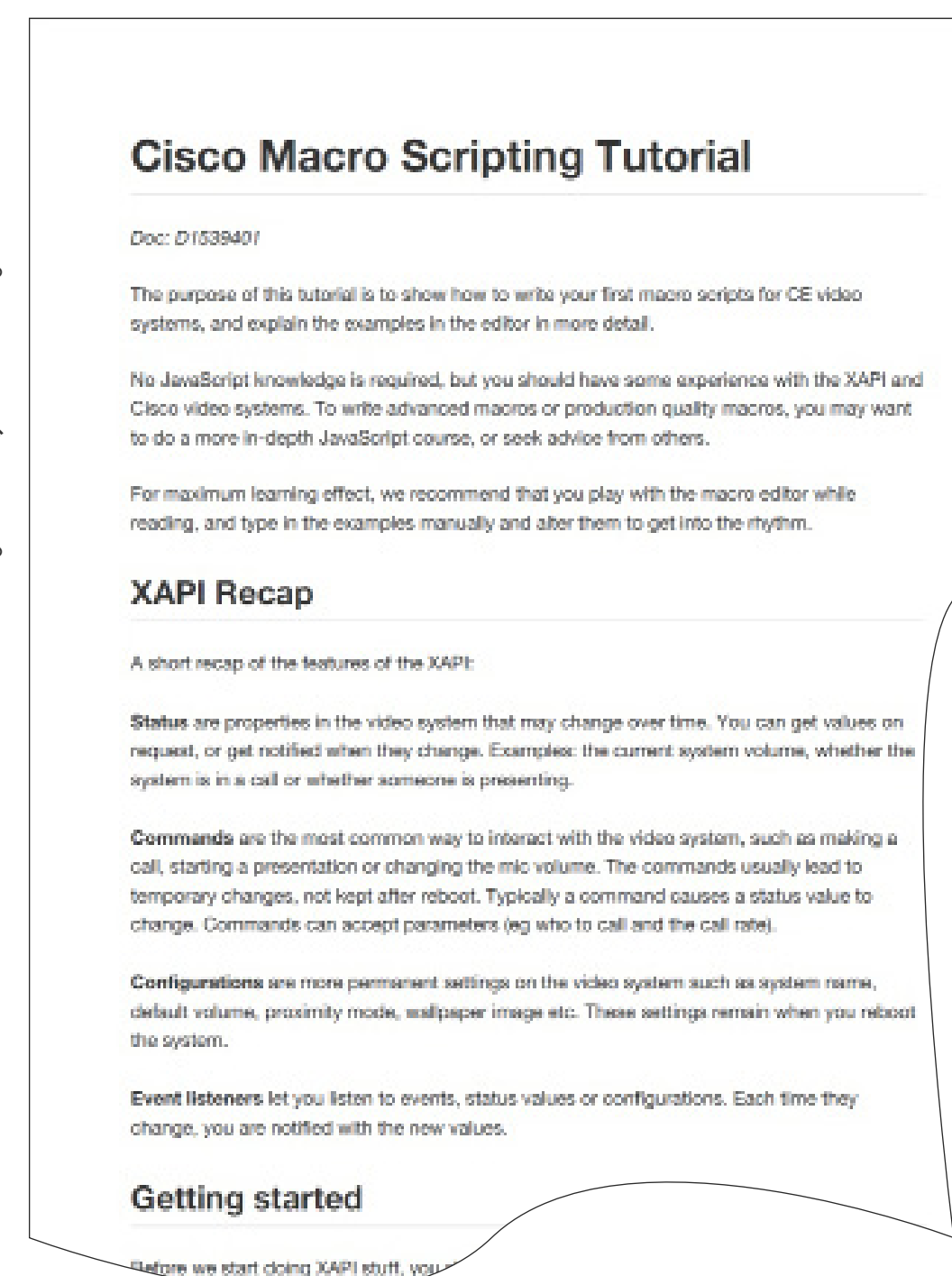
macros.log ファイルには、マクロ コンソールに出力されるものの多くが記録されます。コンソールに出力するようにマクロを設定できますが、これはログに保存されます。そのため、このファイルでカスタム ログ メッセージ(これは開発者が前もって作成することが必要です)を確認できることを覚えておいてください。

マクロ エディタのヘルプ セクションにある「マクロの概要」を参照してください。



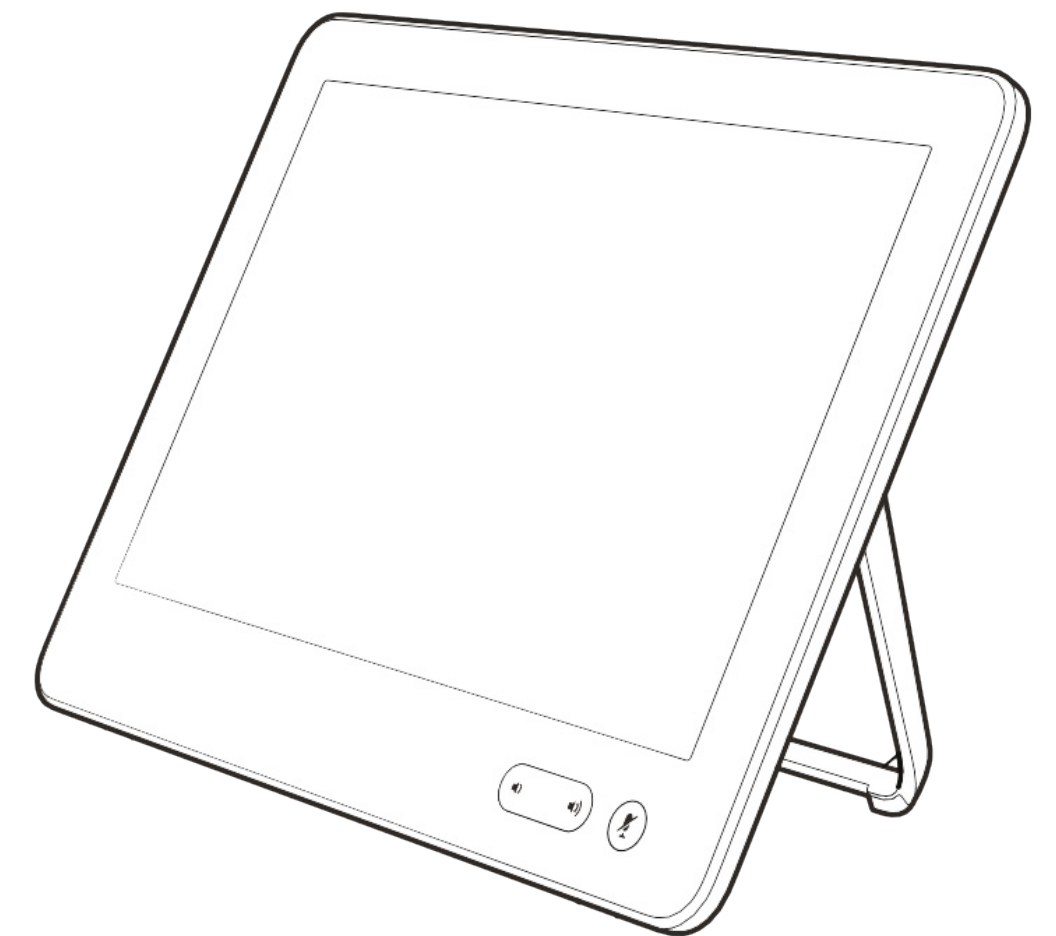
「チュートリアル」では、独自のマクロを作成するにあたり、知っておくべき情報のほとんどが提供されます。

[チュートリアル (Tutorial)] をクリックすると、PDF をダウンロードするか尋ねられます。



すべてのサンプル マクロについて、[サンプルのロード(Load Example)] をクリックし、メイン エディタ ウィンドウに直接コードを貼り付けることができます。詳細については、前のページを参照してください。

第 5 部 オーディオ コンソール



オーディオ コンソール ユーティリティを使用すると、簡単なドラッグ アンド ドロップを使用して、オーディオの入力と出力をどのように接続するかを定義できます。

現在のバージョンは、機能プレビューとお考えください。このバージョンではフル機能を使用できません。

注: このオーディオ コンソールの機能プレビューバージョンは、Codec SX80 と Codec Pro がスタンドアロン Codec として使用されている場合にしか使用できません。

したがって、MX700、MX800、Room 70 G2 では SX80 と Codec Pro を利用しますが、これらのエンドポイントでオーディオ コンソールは使用できません。

オーディオ コンソールは、ビデオ システムの Web インターフェイスの [セットアップ (Setup)] で見つかります。

論理入力グループと論理出力グループを定義することから始めます。その後で、これらの論理入力と論理出力に物理入力と物理出力を割り当てます。

いつでも新しい論理入力グループと論理出力グループを追加できます。同様に、いつでも論理モジュールを削除できます。

設定に適用された変更は直ちに有効になります。ただし、Codec を再起動すると、再起動前に Codec に保存していなかった設定のほとんどが削除されます。したがって、常に設定は保存するように強くお勧めします。

論理入力グループと論理出力グループには自由に名前を付けることができます。

マイクなどの物理入力は複数の入力に割り当てることができます。これは、ローカル聴衆者がマイクによる話を聞く必要がある講堂でビデオ システムを使用する場合に、ローカル補正を操作するのに便利です。

オーディオ コンソールのセットアップを使用すれば、遠端に送信されるマイク信号側のエコー制御を利用し、ローカルで使用される側のエコー制御を省略することができます (マイクを複数の論理グループに割り当てる方法を使用します)。

マイク信号にノイズ リダクションとイコライザの設定を適用することもできます。

機能プレビュー版には、イコライザ設定を定義できるツールはありません。当分の間は、このために xCommand を使用する必要があります。

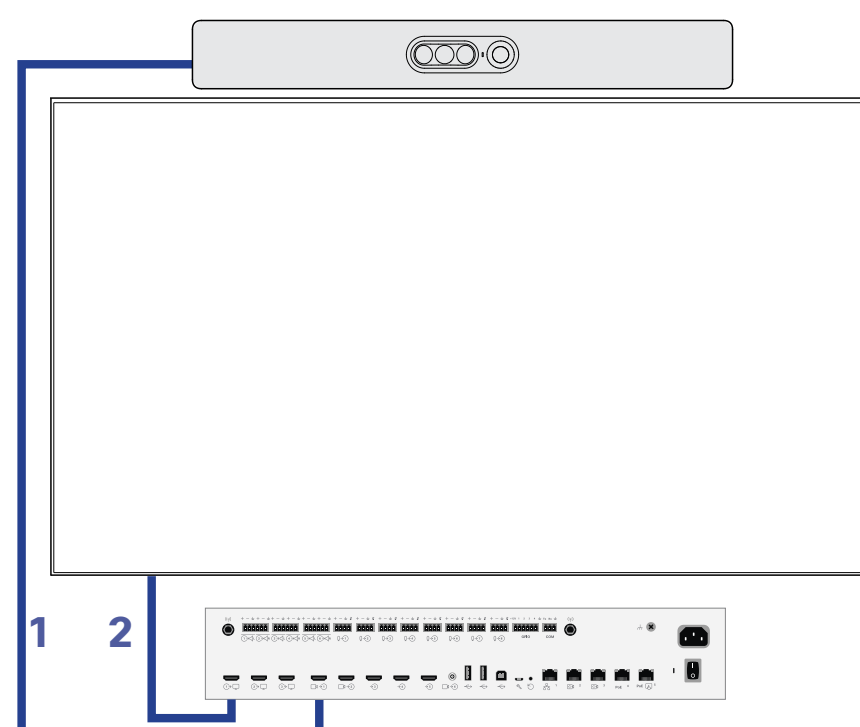
物理出力を複数の論理出力グループに割り当てることはできません。

Codec Pro では、マイク入力はファントム供給がアクティブになっているライン入力です。これにより、自由度が高まります。

オーディオ リターン チャンネルの使用

HDMI は、特定の環境下で、音声をどちらの方向にも送信できる能力を備えています。音声が逆方向に送信される場合をオーディオ リターン チャンネル (ARC) と言います。Codec Pro はこれをサポートしています。

下の図のような設定を考えてみます。HDMI 経由で接続された Cisco QuadCam (一番上)、モニタ (真ん中)、Codec Pro (一番下) が示されています。この設定では、QuadCam がカメラとサウンドバーの両方として機能します。



通常の用途では、HDMI1 が QuadCam ユニットのカメラからの映像を Codec Pro に提供するために使用され、同じ HDMI1 のオーディオ リターン チャンネルが音声を Codec Pro から QuadCam のスピーカーにルーティングするために使用されます。

一方、セットアップを単なるサウンドバー付きの TV として使用したい場合は、システムが HDMI 2 経由で音声をモニタから Codec Pro に送信し、Codec Pro が HDMI1 経由でその音声をさらに QuadCam に送信します。

これを可能にするには、モニタを CEC 対応にする必要があります。セットアップで 4k 映像を使用する場合は、モニタが 4k 形式の CEC をサポートすることを確認してください。

パート 5: オーディオ コンソール オーディオ コンソール パネル

オーディオ コンソールは、ビデオ システムの Web インターフェイスの [セットアップ (Setup)] で見つかります。

接続を削除するには、[ケーブル (cable)] を 2 回クリックします。

各接続のゲインがここで設定されます。

新しい設定をビデオ システムに保存します。

物理入力コネクタのプールが使用可能です。

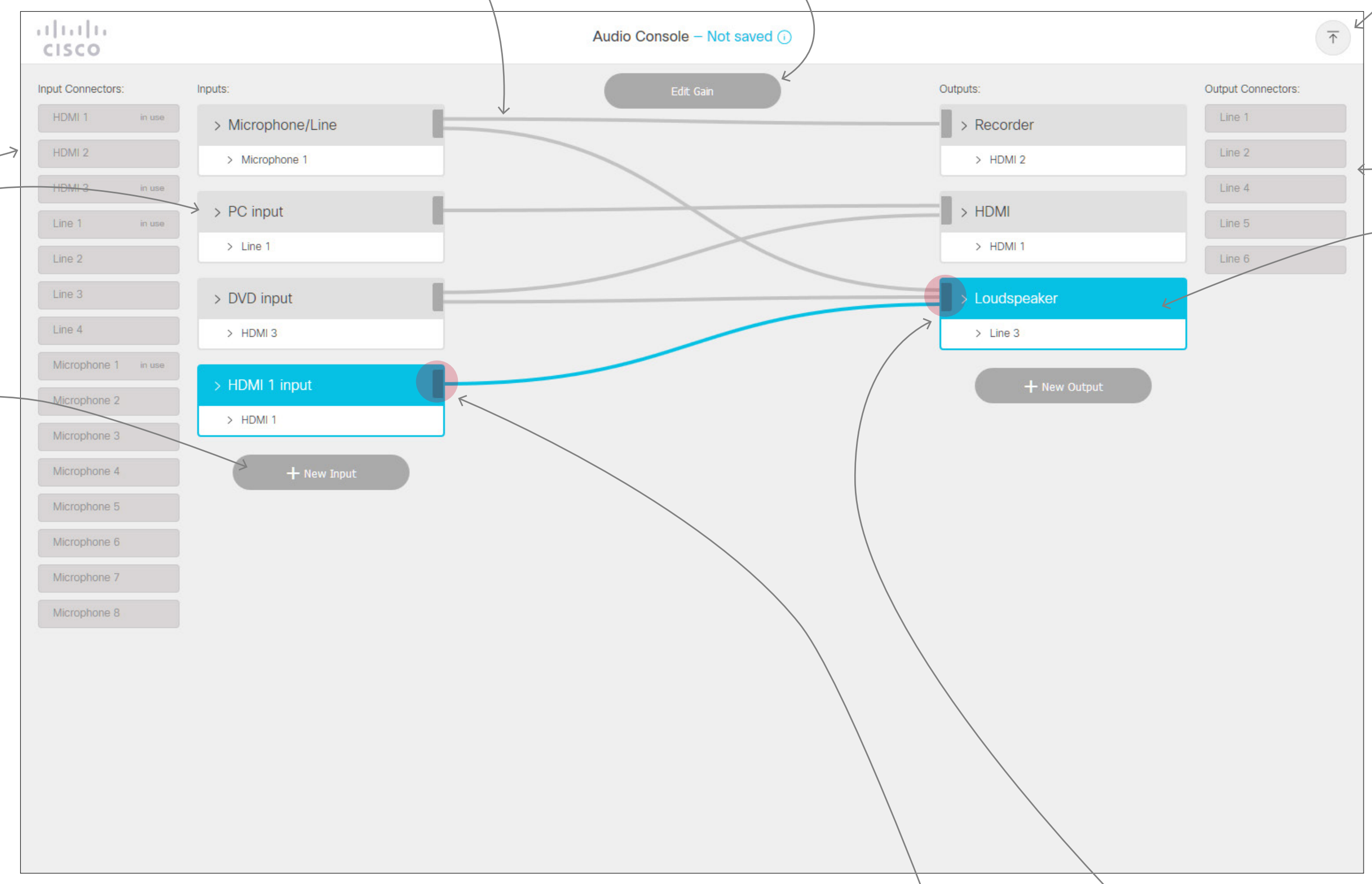
物理入力コネクタは論理入力グループに割り当てられます。これらの論理入力グループはユーザが作成します。

論理グループの名前は、[+ 新規入力 (+ New Input)] をクリックして指定します。

特定の物理入力コネクタを複数の論理グループに割り当てることもできます。ドラッグ アンド ドロップを使用します。

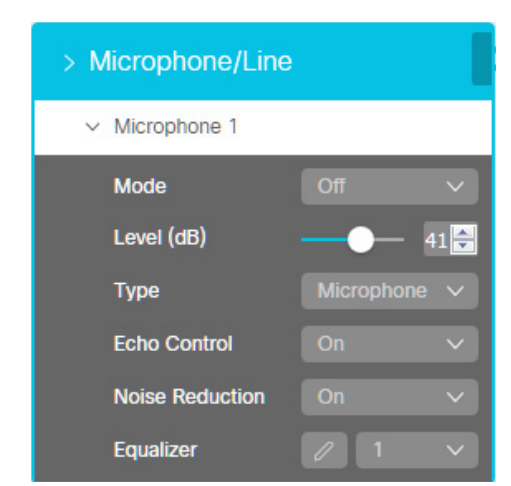
グループまたはグループのメンバーを削除するには、グループまたはメンバーにマウスをホバーします。X が表示されます。これをクリックすると、そのアイテムが削除されます。

論理グループは、セットアップの詳細にアクセスできるように拡張することもできます。



これは、まだ使用されていない物理出力コネクタのプールです。そのため、これらは非アクティブと見なされます。

論理出力グループの処理は論理入力グループと同様ですが、物理出力を複数の論理出力グループに割り当てることはできません。



物理入力は、マイクまたはラインに設定できます。

イコライザを使用すれば、最大 6 つの事前に定義されたイコライザ設定 (または「なし」つまり [オフ (Off)]) の中から選択することができます。

このバージョンでは、イコライザのセットアップに使用できるグラフィカル ツールはありません。編集は xCommand を使用して行う必要があります。

Section	Type	Gain	Frequency	Q
1	Peaking	7.813953488372093	111.95152029902204	4.0
2	Peaking	-11.906976744186046	283.8115043378185	4.0
3	Peaking	-10.418604651162791	796.2143411069947	4.0
4	Peaking	0	1000	4
5	Peaking	0	2000	4
6	Peaking	0	5000	4

ここをクリックしてこちらにドラッグし、入力と出力間の接続を確立します。

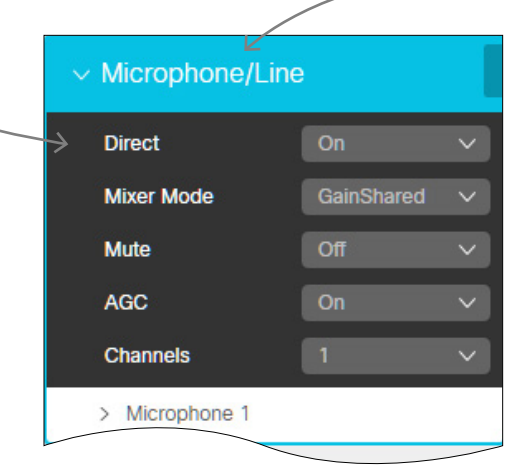
以下のことにご注意ください。

Codec Pro では追加でオーディオ リターン チャンネル (ARC) を選択できます。

Codec Pro では、マイク入力はファントム供給がアクティブになっているライン入力です。

ローカル補正のシナリオでは、エコー制御などの余分な処理がバイパスされるように、[マイク (Microphone(s))] が [ダイレクト (Direct)] に設定されていることを確認してください。

右に示すように、これにアクセスできるように論理グループを拡張します。



ローカル補正に使用する場合は、論理モジュールの個々のメンバーではなく、このモジュールを拡張します。

シスコのお問い合わせ先

シスコの Web サイトでは、シスコの世界各地のお問い合わせ先を確認できます。

URL:<http://www.cisco.com/go/offices>

本社
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134 USA

Intellectual property rights

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

シスコが導入する TCP ヘッダー圧縮は、カリフォルニア大学バークレー校 (UCB) により、UNIX オペレーティング システムの UCB パブリックドメイン バージョンの一部として開発されたプログラムを適応したものです。All rights reserved. Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワークポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

ハードコピーおよびソフトコピーの複製は公式版とみなされません。最新版はオンライン版を参照してください。

シスコは世界各国 200 箇所にオフィスを開設しています。各オフィスの住所、電話番号、FAX 番号については、シスコの Web サイトをご覧ください (www.cisco.com/go/offices)。

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. シスコの商標の一覧については、www.cisco.com/go/trademarks をご覧ください。掲載されている第三者の商標はそれぞれの権利者の財産です。The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)