HTML5
リモコン入力について

BRAVIA Professional Display

2025.12現在

キーイベント

ディスプレイのリモコンを押すと、キーイベント が通知されます。

HTML5アプリは、ホームボタンキー、チャンネル操作キー、矢印キーなどのキーイベントを受け取り、処理することができます。

それ以外のキーに関しても、manifest.json に指定することでアプリケーションへ通知することができます。

以下に、ホームボタン、チャンネルアップ、チャンネルダウン の各キーの指定例を示します。

```
"app": {
 "launch": {
  "local_path": "file://localhost/doc/index.html",
 },
 "attachkeys": [
  "VK_HOME",
  "VK_CH_UP",
  "VK_CH_DOWN"
 ]
}
```

以下に、アプリケーションにおけるキーイベントの受け取り方を示します。

```
document.onkeydown = function(e) {
 if (e.keyCode === VK_HOME) {
  // Please implement a home key behavior here.
 }
}
```

リモコンキーイベントの処理例のサンプルアプリケーションを活用ください。

リモコンのインテグレーション

ディスプレイのリモコンは、以下の表のようなキーイベントを通知します。

VK_* 値はウィンドウオブジェクトの定数プロパティです。 これらはシステムにより定義されたオブジェクトであり、ユーザーが定義する必要はありません。

リモコンキー	バーチャルキー
Left	VK_LEFT
Up	VK_UP
Right	VK_RIGHT
Down	VK_DOWN
Enter (Confirm)	VK_ENTER
Return	VK_BACK_SPACE
Blue	VK_BLUE
Red	VK_RED
Green	VK_GREEN
Yellow	VK_YELLOW
0	VK_0
1	VK_1
2	VK_2
3	VK_3
4	VK_4
5	VK_5
6	VK_6
7	VK_7
8	VK_8
9	VK_9
Play	VK_PLAY
Pause	VK_PAUSE
Stop	VK_STOP
Next	VK_TRACK_NEXT
Prev	VK_TRACK_PREV
FF (Fast-Forward)	VK_FAST_FWD
FR (Fast-Rewind)	VK_REWIND
Subtitle	VK_SUBTITLE
Information	VK_INFO
Exit	VK_EXIT

そのほかの定義済みキーイベント

以下の表のキーは、ウィンドウオブジェクトの定数プロパティですが、デフォルトでは通知されません。
イベントを受信するには、manifest.json に追加する必要があります。

モコンキー	バーチャルキー
Home	VK_HOME
CH+	VK_CH_UP
CH-	VK_CH_DOWN
Audio	VK_AUDIO_CHANGE
HDMI 1	VK_TV_INPUT_HDMI_1 [1]
HDMI 2	VK_TV_INPUT_HDMI_2 [1]
HDMI 3	VK_TV_INPUT_HDMI_3 [1]
HDMI 4	VK_TV_INPUT_HDMI_4 [1]
HELP	VK_HELP [1]
SYNC	VK_SYNC_MENU [1]

[1] 対応ディスプレイ: BZ53L, BZ50L, BZ40L, BZ35L, BZ30L, BZ40J, BU40J, BZ35J, BU35J, BZ30J, BU30J, BZ40H, BU40H

一部キーの取り扱いにおける追加対応

上記HDMI 1/HDMI 2/HDMI 3/HDMI 4/HELP/SYNC 等の一部キーは、manifest.jsonへの追加を行ってもキーイベントを完全にコントロールすることができません。
HTML5アプリ内で下記のpreventDefault()をコールし、該当のキーイベントをキャンセルする必要があります。

HTML5
Remote Commander

BRAVIA Professional Display

As of December 2025

Key Events

Key events are delivered when signaled from BRAVIA Professional Display remote commander. HTML5 Application can handle the “HOME”, “CH+/-“, arrow keys, and so on. An application can provide display features without launching BRAVIA Professional Display’s default applications.

The rest of keys can also be handled in an application when they are manually declared in manifest.json.

The example below shows how to define the HOME, CH+ and CH- keys.

```
"app": {
 "launch": {
  "local_path": "file://localhost/doc/index.html",
 },
 "attachkeys": [
  "VK_HOME",
  "VK_CH_UP",
  "VK_CH_DOWN"
 ]
}
```

The example below shows how to use them in an application.

```
document.onkeydown = function(e) {
 if (e.keyCode === VK_HOME) {
  // Please implement a home key behavior here.
 }
}
```

You can also download a sample application that shows how to handle key events delivered from a remote commander.

Integrate Remote Commander

Key events are delivered when signaled from the BRAVIA Professional Display remote commander as described in the table below.

VK_* values are the constant properties owned by the window object. These values are predefined objects by the system and users do not have to define them. These events are delivered by default.

Remote commander key	Virtual key
Left	VK_LEFT
Up	VK_UP
Right	VK_RIGHT
Down	VK_DOWN
Enter (Confirm)	VK_ENTER
Return	VK_BACK_SPACE
Blue	VK_BLUE
Red	VK_RED
Green	VK_GREEN
Yellow	VK_YELLOW
0	VK_0
1	VK_1
2	VK_2
3	VK_3
4	VK_4
5	VK_5
6	VK_6
7	VK_7
8	VK_8
9	VK_9
Play	VK_PLAY
Pause	VK_PAUSE
Stop	VK_STOP
Next	VK_TRACK_NEXT
Prev	VK_TRACK_PREV
FF (Fast-Forward)	VK_FAST_FWD
FR (Fast-Rewind)	VK_REWIND
Subtitle	VK_SUBTITLE
Information	VK_INFO
Exit	VK_EXIT

Predefined Key Events

The keys described in below table are predefined as the constant properties owned by the window object. These events are not delivered by default. These keys must be described in manifest.json to receive events.

Remote commander key	Virtual key
Home	VK_HOME
CH+	VK_CH_UP
CH-	VK_CH_DOWN
Audio	VK_AUDIO_CHANGE
HDMI 1	VK_TV_INPUT_HDMI_1 [1]
HDMI 2	VK_TV_INPUT_HDMI_2 [1]
HDMI 3	VK_TV_INPUT_HDMI_3 [1]
HDMI 4	VK_TV_INPUT_HDMI_4 [1]
HELP	VK_HELP [1]
SYNC	VK_SYNC_MENU [1]

[1] Supported display : BZ53L, BZ50L, BZ40L, BZ35L, BZ30L, BZ40J, BU40J, BZ35J, BU35J, BZ30J, BU30J, BZ40H, BU40H

Additional support in handling some keys

Some of the above keys such as HDMI 1/HDMI 2/HDMI 3/HDMI 4/HELP/SYNC cannot be fully controlled by adding them to manifest.json. It is necessary to call the following preventDefault() in the HTML5 app to cancel the corresponding key event.

HTML5
遥控指令

BRAVIA Professional Display

截至2025年12月

按键事件

当BRAVIA商用显示器遥控指令发出信号时，将发送按键事件。 HTML5应用程序可以处理“HOME”、“CH+/-”、箭头键等。 应用程序可以在不启动BRAVIA商用显示器的默认应用程序的情况下提供显示功能。
当在应用程序中手动声明其余键时，也可以在应用程序中处理它们 manifest.json.
下面的示例展示了如何定义HOME、CH+和CH-键。

```
"app": {
 "launch": {
  "local_path": "file://localhost/doc/index.html",
 },
 "attachkeys": [
  "VK_HOME",
  "VK_CH_UP",
  "VK_CH_DOWN"
 ]
}
```

下面的示例展示了如何在应用程序中使用它们。

```
document.onkeydown = function(e) {
 if (e.keyCode === VK_HOME) {
  // Please implement a home key behavior here.
 }
}
```

您同时也可以下载a sample application 这显示了如何处理来自远程指挥官的按键事件。

集成遥控指令

按键事件在BRAVIA商用显示器遥控发出信号时执行，如下表所示。 VK_* 值是窗口对象拥有的常量属性。 这些值是系统预定义的对象，用户不必定义它们。 默认情况下，这些事件是被执行的。

Remote commander key	Virtual key
Left	VK_LEFT
Up	VK_UP
Right	VK_RIGHT
Down	VK_DOWN
Enter (Confirm)	VK_ENTER
Return	VK_BACK_SPACE
Blue	VK_BLUE
Red	VK_RED
Green	VK_GREEN
Yellow	VK_YELLOW
0	VK_0
1	VK_1
2	VK_2
3	VK_3
4	VK_4
5	VK_5
6	VK_6
7	VK_7
8	VK_8
9	VK_9
Play	VK_PLAY
Pause	VK_PAUSE
Stop	VK_STOP
Next	VK_TRACK_NEXT
Prev	VK_TRACK_PREV
FF (Fast-Forward)	VK_FAST_FWD
FR (Fast-Rewind)	VK_REWIND
Subtitle	VK_SUBTITLE
Information	VK_INFO
Exit	VK_EXIT

预定义的按键事件

下表中描述的键被预定义为window对象拥有的常量属性。 默认情况下，这些事件不会被执行。 这些密钥必须在清单中描述。json接收事件。

Remote commander key	Virtual key
Home	VK_HOME
CH+	VK_CH_UP
CH-	VK_CH_DOWN
Audio	VK_AUDIO_CHANGE
HDMI 1	VK_TV_INPUT_HDMI_1 [1]
HDMI 2	VK_TV_INPUT_HDMI_2 [1]
HDMI 3	VK_TV_INPUT_HDMI_3 [1]
HDMI 4	VK_TV_INPUT_HDMI_4 [1]
HELP	VK_HELP [1]
SYNC	VK_SYNC_MENU [1]

[1] 支持的显示器: BZ53L, BZ50L, BZ40L, BZ35L, BZ30L, BZ40J, BU40J, BZ35J, BU35J, BZ30J, BU30J, BZ40H, BU40H

对处理某些按键的额外支持

对于某些按键，例如 HDMI 1/HDMI 2/HDMI 3/HDMI 4/HELP/SYNC，即使添加到manifest.json 中，也无法完全控制按键事件。
您需要在 HTML5 应用程序中调用下面的 PreventDefault() 来取消相应的按键事件。