Technical note

Keyboard channel activation with PlugnCast G2

(Version 002A_en)

Introduction

This document explains how to play a new content thanks to keyboard key in PlugnCast G2 environment.

Prerequisites

Gekkota 3.12.19 (or above) with player supporting USB port (DMB300 for example)

Plugncast server 2.50.31 (or above)

Plugncast Studio 2.50.41 (or above)

Use with PlugnCast G2

Keyboard

A standard USB keyboard or a keypad can be used as well.

Default debouncing filter is 3 seconds. This is set up to avoid unexpected behavior when too many keypress are done in a short period of time.

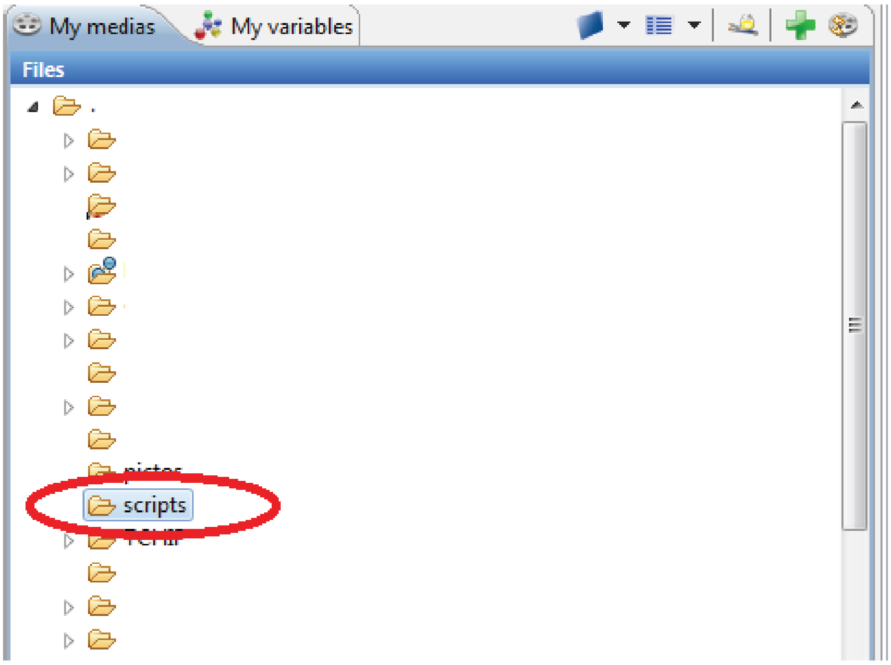
Script

Given that PlugnCast Studio does not have interface to manage simply a keyboard, an additional script keyboard_channel.js needs to be used.

The script is available on site INNES (PlugnCast_and_Screen_Composer_G2/Application Notes) or CDROM PCS.ON.WIN.P5 V2.50BQ

Once this script is available on your workstation:

- open PlugnCast Studio
- and add this script into a new folder of your media library (ex: folder name ‘scripts’)



Principle

- New channels will be defined and linked to a specific key on the Keyboard.
- Each time a key will be pressed (and released), the current channel(s) will be replaced with the one linked to the key (the script launch the new channel only when the key is released)
    - In case the new keyboard channel is an audio channel: the current audio channel is replaced.
    - In case the new keyboard channel is an audio-visual channel : the current audio-visual channel is replaced
- If the same key is pressed (and released) before the end of the media, the players will go back to the main channel

Note 2: the current audio channel can be stopped as well by using the property channel\type (explained just after).

Create Channels

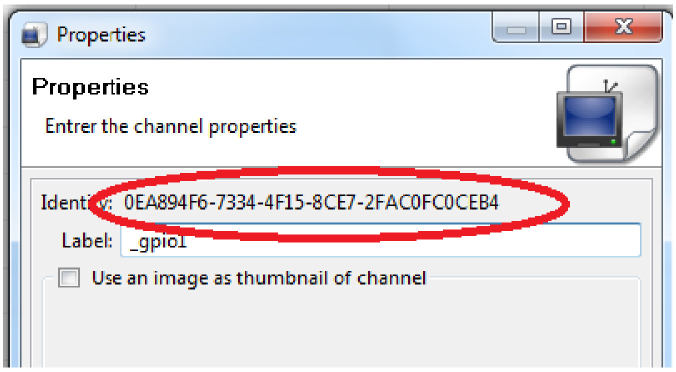
Create as many new replacement channels as required for your needs.

Please save back the unique identifier of each of the channels. They will be required to complete the configuration

How to get the unique identifier of a channel?

- go to the ‘Channel’ tab

- select the appropriate channel
- right click on it
- select properties
- and copy the channel_id



Properties

Properties

Entrer the channel properties

Identity: 0EA894F6-7334-4F15-8CE7-2FAC0FC0CEB4

Label: _gpio1

□ Use an image as thumbnail of channel

To ensure the proper working of this function. For new Channels to be created :

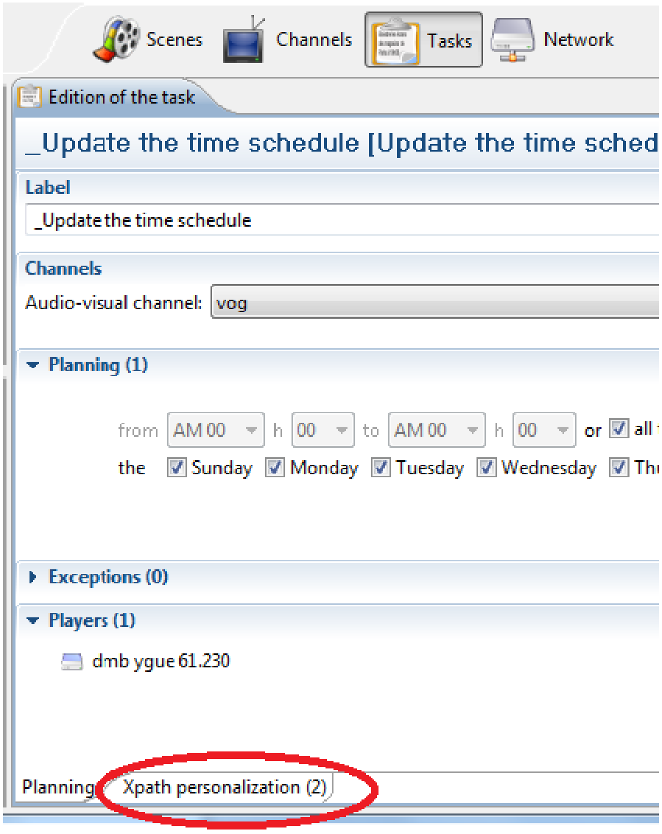
- The default scene cannot be used
- **one** scene must be used in this channel
- The calendars events start time and end time won’t be taken into account

Task configuration and xpath personalization

The task ‘Update the time schedule’ must be configured so that the player can download:

- the script ‘keyboard_channel.js’
- All the new keyboard channels.

Go to tab ‘Xpath personalization’



_Update the time schedule [Update the time sched

Label
_Update the time schedule

Channels
Audio-visual channel: vog

Planning (1)

from AM 00 h 00 to AM 00 h 00 or ☑ all

the ☑ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thu

Exceptions (0)

Players (1)

dmb ygue 61.230

Planning | Xpath personalization (2)

Several elements need to be configured

Important :

- **Respect the order of declaration**
- **Make sure to use the straight double quote : "**

(in the following order) :

1. First element: script configuration content:

```
//xpf:player

<script xmlns="http://www.innes.fr/2007/XPF10/Language">

// Set to true, if a scene launched by a key can be interrupted by another (not required - default = false)

const interrupt_scene = true;

// Set to true, if the same scene must be restarted when pressing the same button before the scene has ended (not required - default = false)

const restart_on_interrupt = false;

// The number of key configured (not required - default = 10)

const key_count=4;
```

```
// The channel id for the key 1 to key_count (required)
const key<1..key_count>_channel_id="<channel_id>";
// The keyboard key for the key 1 to key_count (required)
// example : const key1_key=«A”;
const key<1..key_count>_key="<key>";
// Must be set only if the normal audio channel must be stopped
const key<1..key_count>_channel_type="audio-video";
</script>
```

You can change the first value interrupt_scene in case a keyboard scene need to be interrupted by another one

You can change the number of keys that you want to control by changing the value key_count

Add as many parts const key<1..key_count>_channel_id= as required (until key_count keys)

For each of this part, define the identifier of the channel to be played

Add as many parts const key<1..key_count>_channel_type="audio-video" as required (until key_count keys) in case the audio channel of the normal source must be

stopped.

2. Second element: script loading content:

```
//xpf:player
<script src=".medias/scripts/keyboard_channel.js"
xmlns="http://www.innes.fr/2007/XPF10/Language"/>
```

- The part ".medias/scripts/keyboard_channel.js" is the relative script location path.

3. Specific element(s) for keyboard channel loading: audio-visual channels content:

```
//xpf:player
<channel height="100%" id="<channel_id>"
left="0px" top="0px" transition="none" transitionDuration="3000"
type="visual" width="100%"
xmlns="http://www.innes.fr/2007/XPF10/Language" zindex="1"/>
```

Replace the part `<channel_id>` with the channel identifier of the keyboard channel

If you want to activate inter-medias transition,

`change transition="none"` by `transition="opacity"`, and

`change transitionDuration` if required (duration is in milliseconds, 3000 maximum)

4. Specific element(s) for keyboard channel loading: audio channels content:

```
//xpf:player

 <channel id="<channel_id>"

type="audio" xmlns="http://www.innes.fr/2007/XPF10/Language"
zindex="1"/>
```

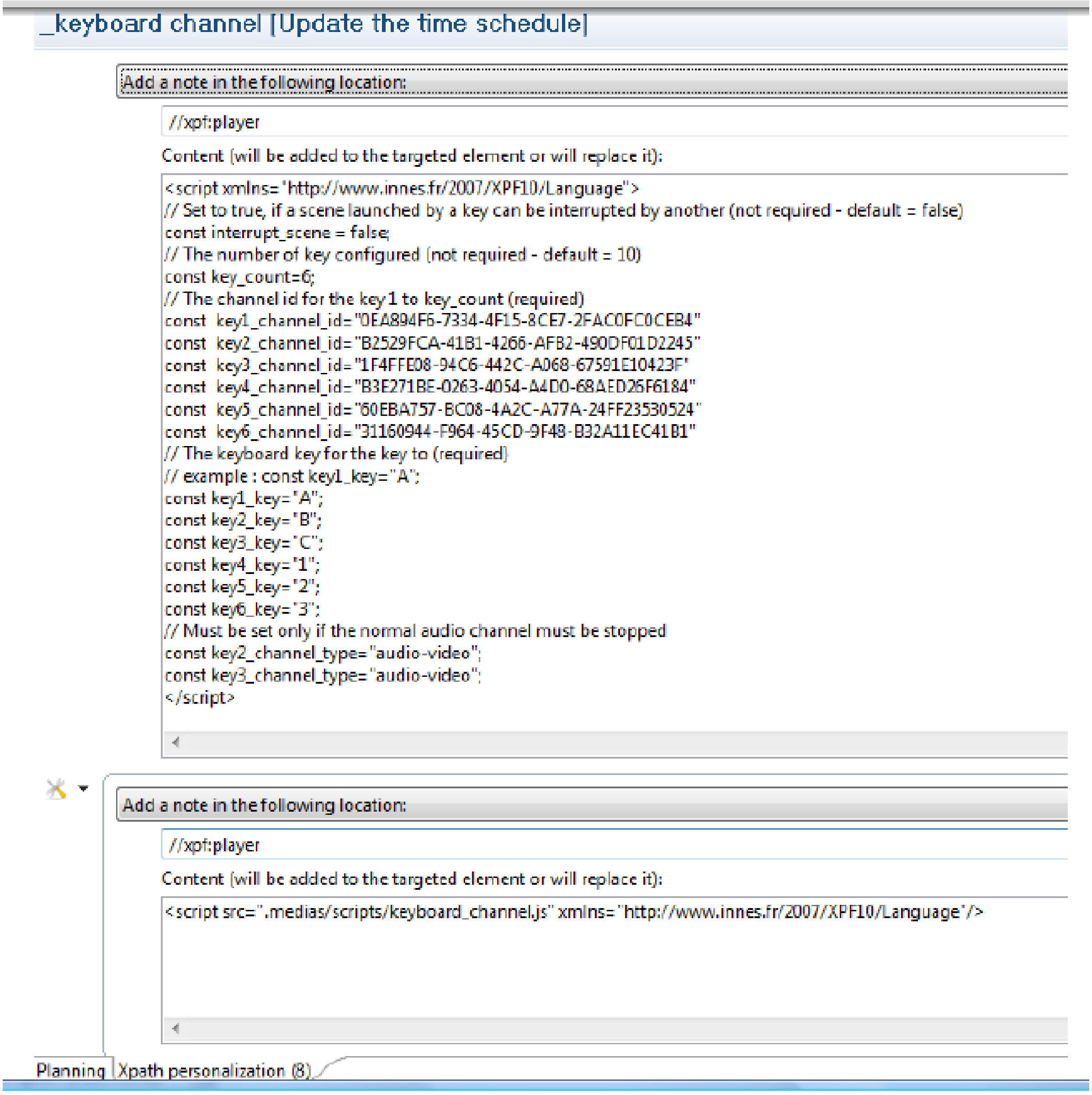
Replace the part `<channel_id>` with the channel identifier of the keyboard channel

Each keyboard channel is associated to a script element

For example, when 6 different keys need to be used, 8 elements (2 + 6) need to be inserted inside Xpath Personalization

WARNING: Ensure that all the elements are in the position ‘Add a note in the following location:’ (else the script cannot work properly).

Example of final rendering:



_keyboard channel [Update the time schedule]

Add a note in the following location:

//xpf:player

Content (will be added to the targeted element or will replace it):

```
<script xmlns="http://www.innes.fr/2007/XPF10/Language">
// Set to true, if a scene launched by a key can be interrupted by another (not required - default = false)
const interrupt_scene = false;
// The number of key configured (not required - default = 10)
const key_count=6;
// The channel id for the key 1 to key_count (required)
const  key1_channel_id="0EA894F6-7334-4F15-8CE7-2FAC0FC0CEB4"
const  key2_channel_id="B2529FCA-41B1-4266-AFB2-490DF01D2245"
const  key3_channel_id="1F4FFE08-94C6-442C-A068-67591E10423F"
const  key4_channel_id="B3E271BE-0263-4054-A4D0-68AED26F6184"
const  key5_channel_id="60EBA757-BC08-4A2C-A77A-24FF23530524"
const  key6_channel_id="31160944-F964-45CD-9F48-B32A11EC41B1"
// The keyboard key for the key to (required)
// example : const key1_key="A";
const key1_key="A";
const key2_key="B";
const key3_key="C";
const key4_key="1";
const key5_key="2";
const key6_key="3";
// Must be set only if the normal audio channel must be stopped
const key2_channel_type="audio-video";
const key3_channel_type="audio-video";
</script>
```

Add a note in the following location:

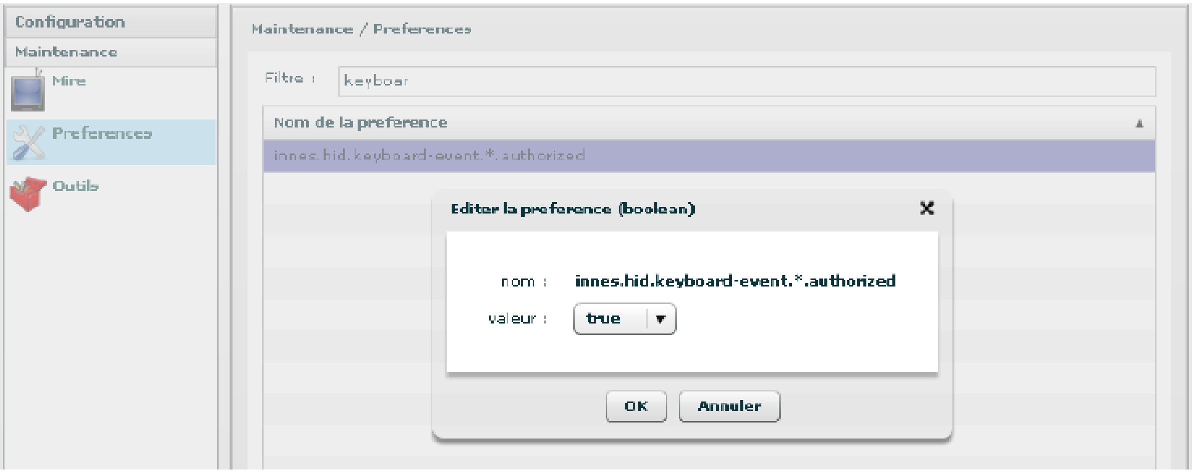
//xpf:player

Content (will be added to the targeted element or will replace it):

```
<script src=".medias/scripts/keyboard_channel.js" xmlns="http://www.innes.fr/2007/XPF10/Language"/>
```

Use with Gekkota/elinux

In case of using Gekkota/elinux, the preference of player: innes.hid.keyboard-event.*.authorized must be on True



Filtre : keyboar

Nom de la preference

innes.hid.keyboard-event.*.authorized

Editer la preference (boolean)

nom : innes.hid.keyboard-event.*.authorized

valeur : true