AIRENCE

Virtual Keyboard Mapper

Using Playout Plugins Feature

D&R Electronica Weesp BV
Rijnkade 15B
1382GS Weesp
The Netherlands
Phone: +31 (0)294-418014
Fax: +31 (0)294-416987
Website: http://www.d-r.nl
E-mail: info@d-r.nl

Version 1.1 - 2014-01-30

1 Table of contents

2 Installing a playout plugin

A playout plugin is nothing more than a XML file with specific content only available for the ‘Airence Virtual Keyboard Mapper’ application. For further information about the content of a plugin file or create your own plugin, refer to chapter 5.

If you have an existing plugin file available ready to use, one has to place the file into the following folder: \<*installationdirectory*>\plugins. Make sure the file has the *.xml* extension.

For example: C:\Program Files (x86)\Airence Virtual Keyboard Mapper\plugins

3 Loading a playout plugin

Once the ‘Airence Virtual Keyboard Mapper’ application is started, all the available plugins in the */plugins* folder will be shown in the selectionbox under ‘Choose Playout Plugin’ section. one of these installed plugins can be loaded for usage by selection one from the list.

In the fields below the selectionbox information about the selected plugin is displayed.

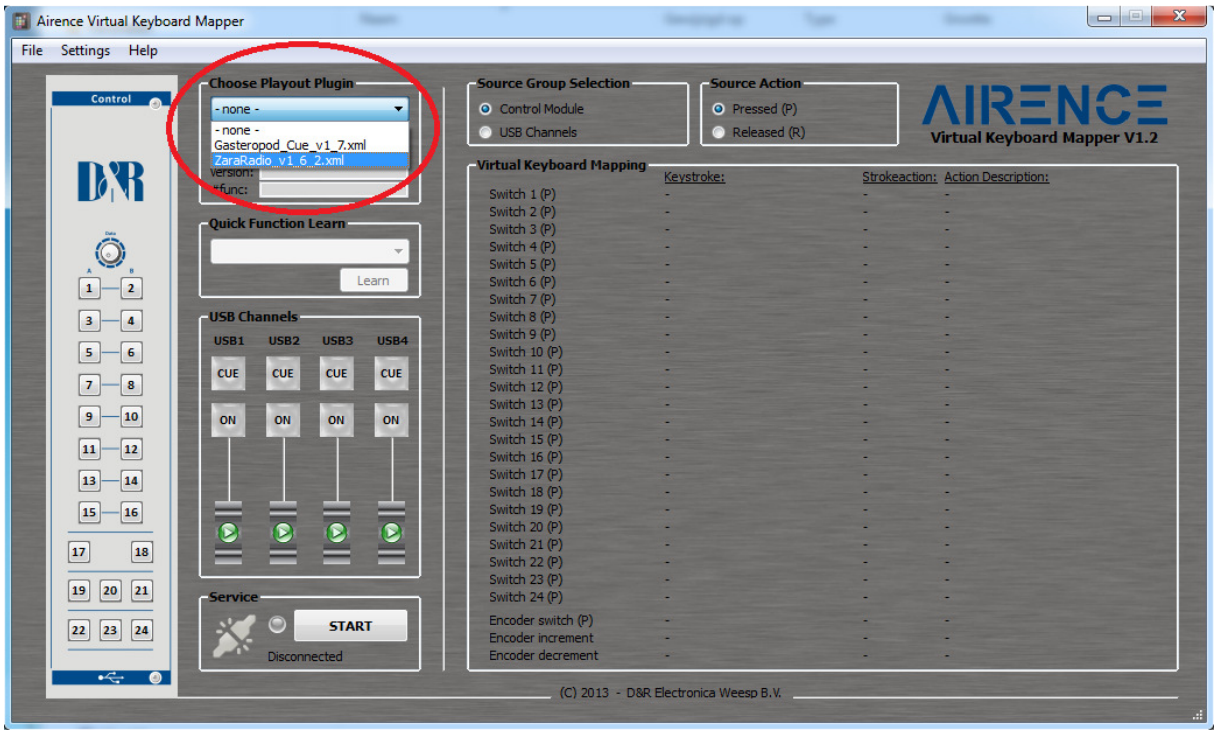

Figure 1. Selecting a playout plugin

4 Using the plugin

A playout plugin contains predefined functions which are mapped to keystrokes for specific playout software. The idea of using a plugin is to easily map such a function to a control signal coming from the Airence mixer. The mapping can be done manually or by the '*Quick Function Learn'* feature.

4.1 Quick Function Learn

Once a plugin is loaded, all the available functions from within that plugin are displayed in the Quick Function Learn selectionbox. The Quick Function Learn feature offers a fast way of mapping functions to control signals from the Airence mixer without worrying the underlying keystroke.

In order to map a function, select one of the list and press the *Learn* button.

Another window with a time progression bar will show up requesting you to trigger a control signal. A control signal can be one of the 24 switches, encoder, usb faderstarts/On/Cue signals. When triggered a control signal, the application will recognize it and maps the selected function to the control signal.

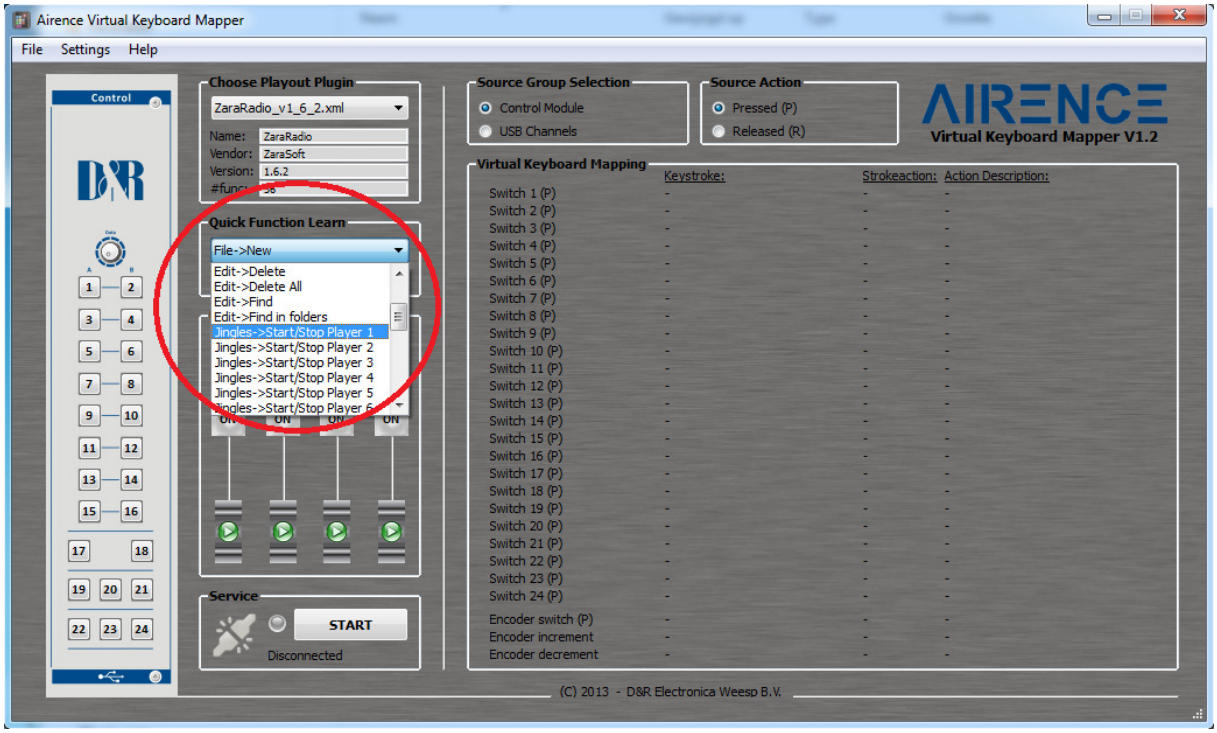

Figure 2. Selecting a function

Example: Mapping function *'Jingles->Start/Stop Player 1'* to switch 5 of the control section:

STEP 1: Select the function 'Jingles->Start/Stop Player 1' from the list.

STEP 2: Press the 'Learn' button.

STEP 3: Press switch 5 on the control section of the Airence mixer.

The above example shows how easy it is to map a playout function to a switch of the control section in just three steps! In the *Virtual Keyboard Mapping* area an overview of the mapped functions is shown (figure 4).

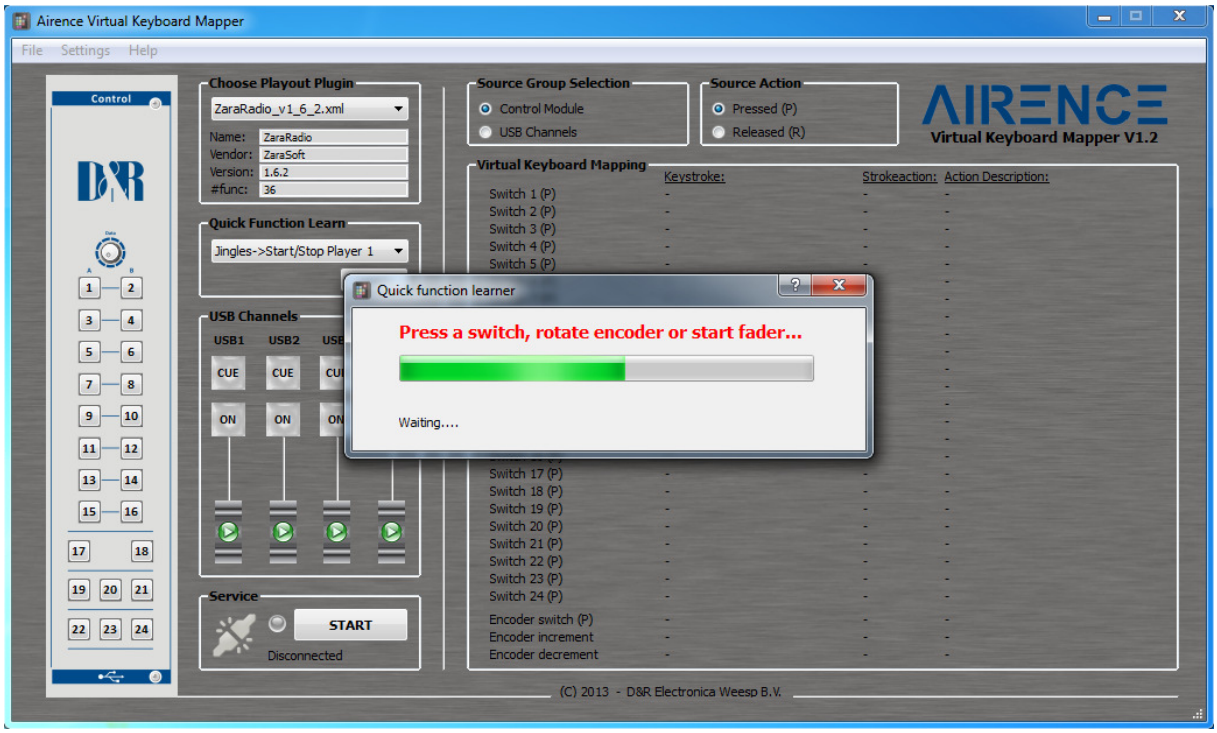

Figure 3. Waiting for a control signal trigger

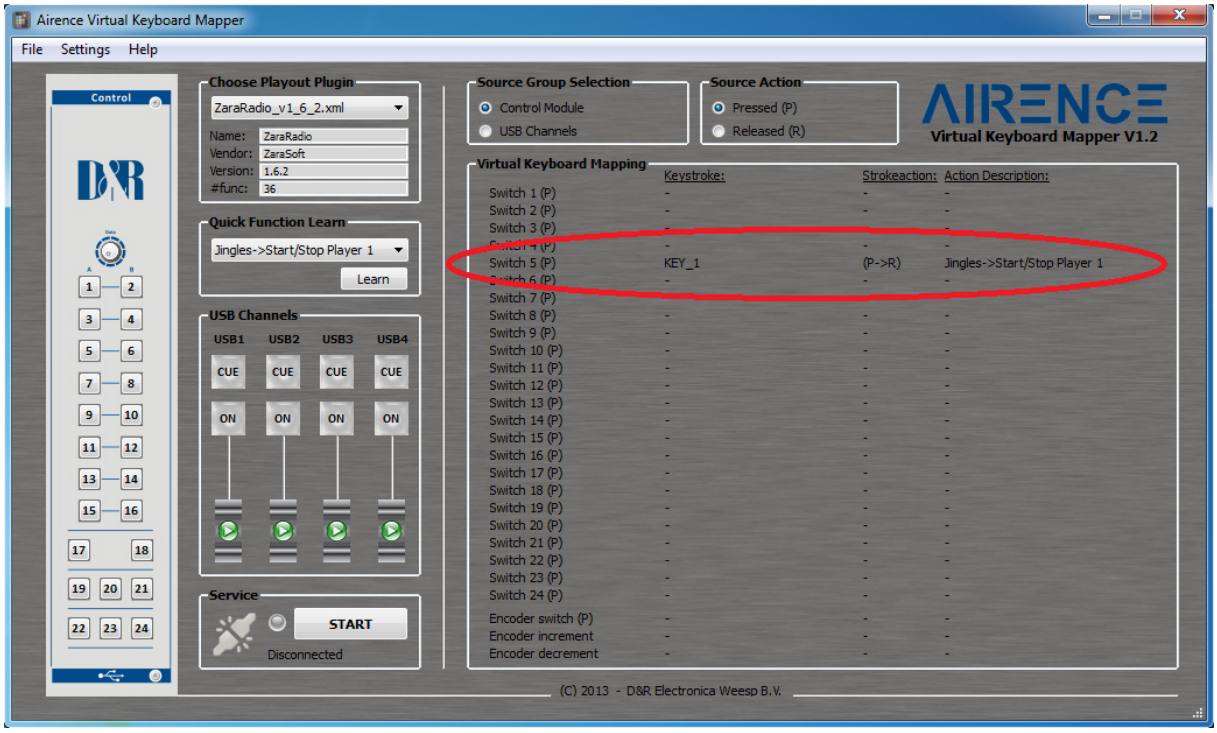

Figure 4. The function is mapped

4.2 Manually Function mapping

In order to manually map a function to a control signal one needs to press one of the available control signal buttons in the application. Figure 5 shows these buttons in the green outlined area.

The '*Keystroke Configuration Editor'* window will show up to select a function for this control signal.

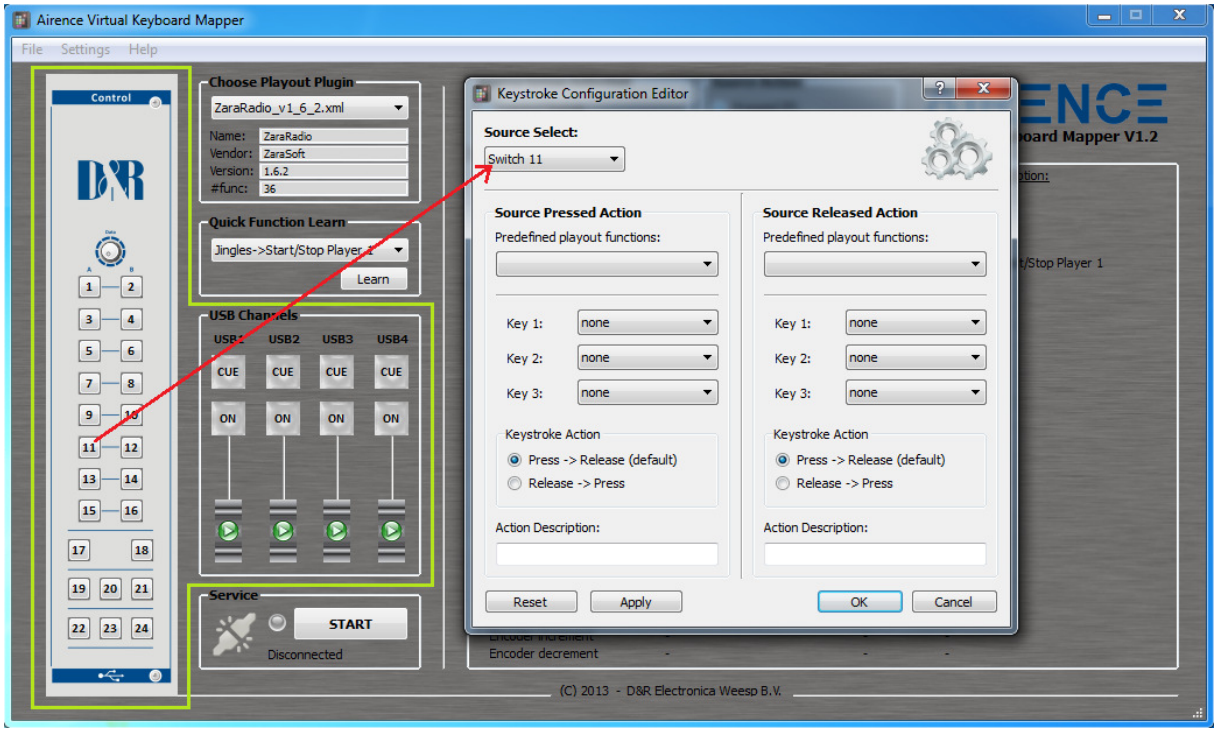

Figure 5. Control signal sources

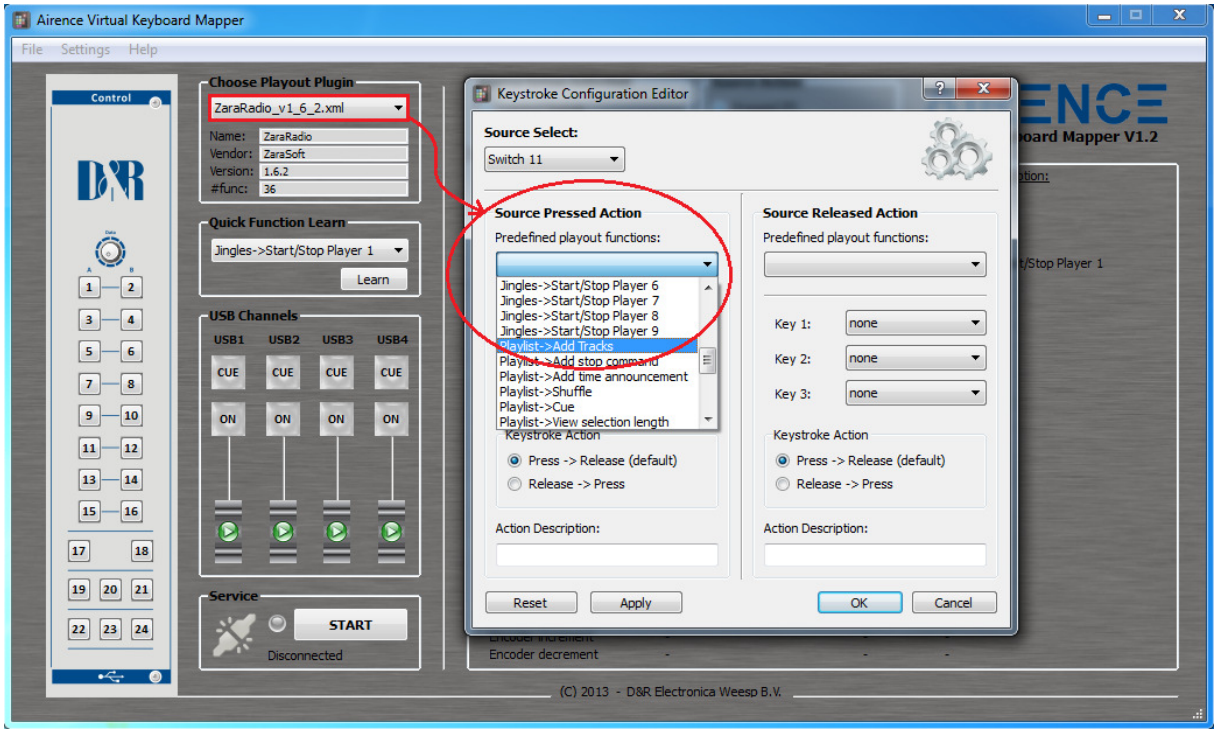

Figure 6. Available functions from the plugin

Figure 6 shows how you can select a function from the list. The list contains all the functions which are related to the selected plugin. If no plugin is selected before, the list will be empty and nothing can be selected.

Once a function is selected the fields below will be filled with the information extracted from that function. The keystroke, keystroke action, and the Action description fields are filled in automatically. One can adjust these parameters if necessary, and finally press OK to confirm the mapping.

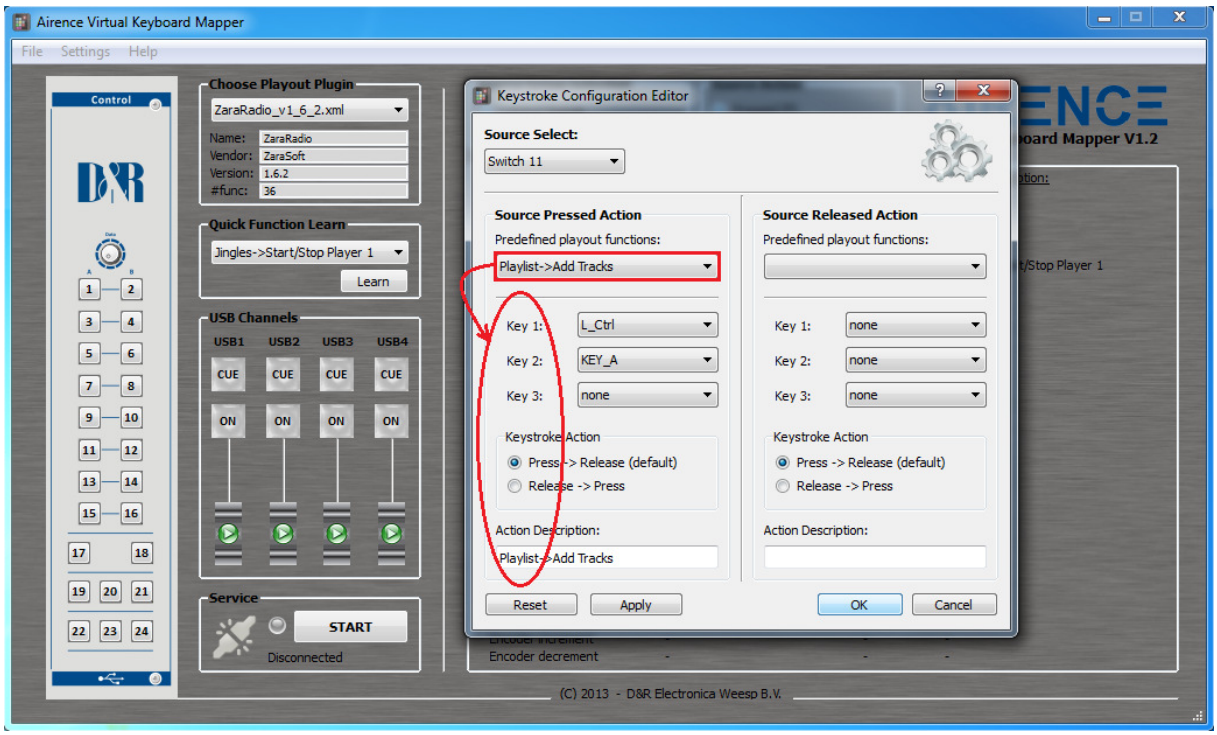

Figure 7. Selecting a function

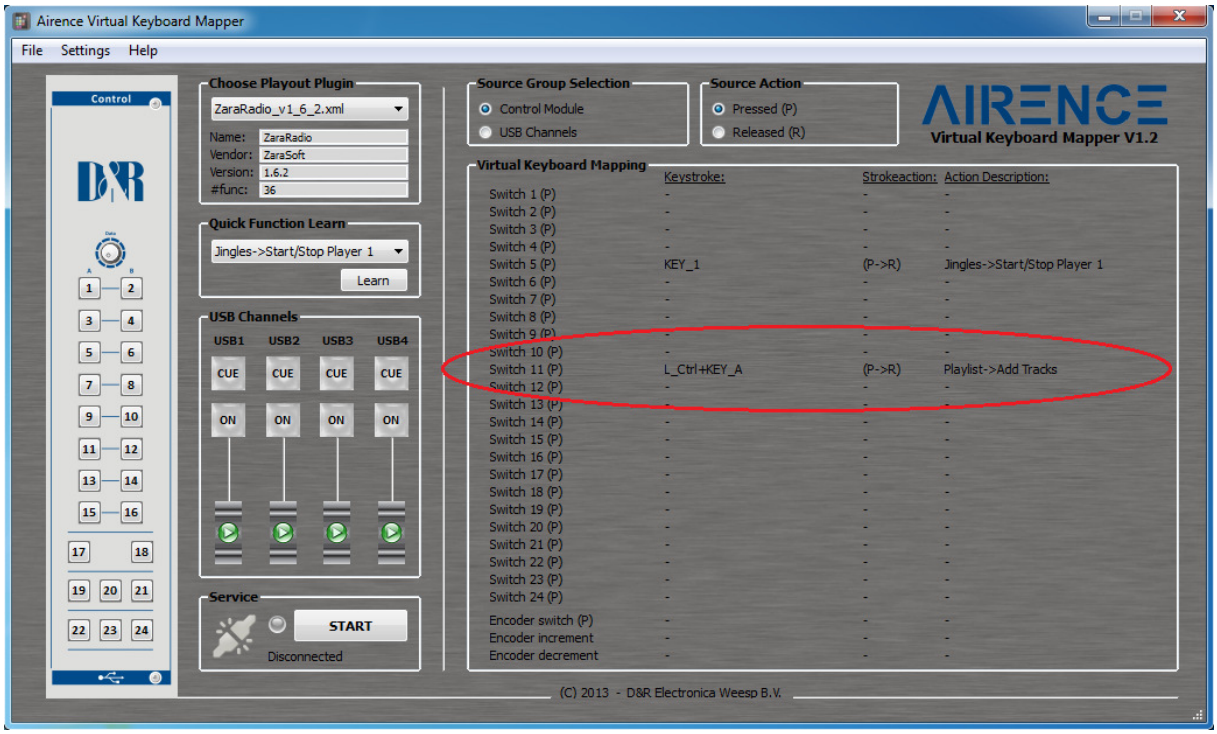

Figure 8. The function is mapped

5 Create your own plugin

An playout plugin is a set of function descriptions of a specific software application (playout software) which are mapped to keystrokes. A plugin is represented as a standard XML-file which can be seen in the figure below. There is a plugin folder *PLUGINS* created in the installation directory during setup where the plugins are located or has to be placed when creating one. Present plugins in this folder are loaded when *Virtual Keyboard Mapper* is starting up.

The advantage of using plugins will be the ease of mapping a control signal to a function. One don't needs to worry about the underlying keystroke of a relevant function since selecting a function from the *Keystroke Configuration Editor* will fill in the corresponding keystroke automatically. These keystrokes are fetched from the plugin file.

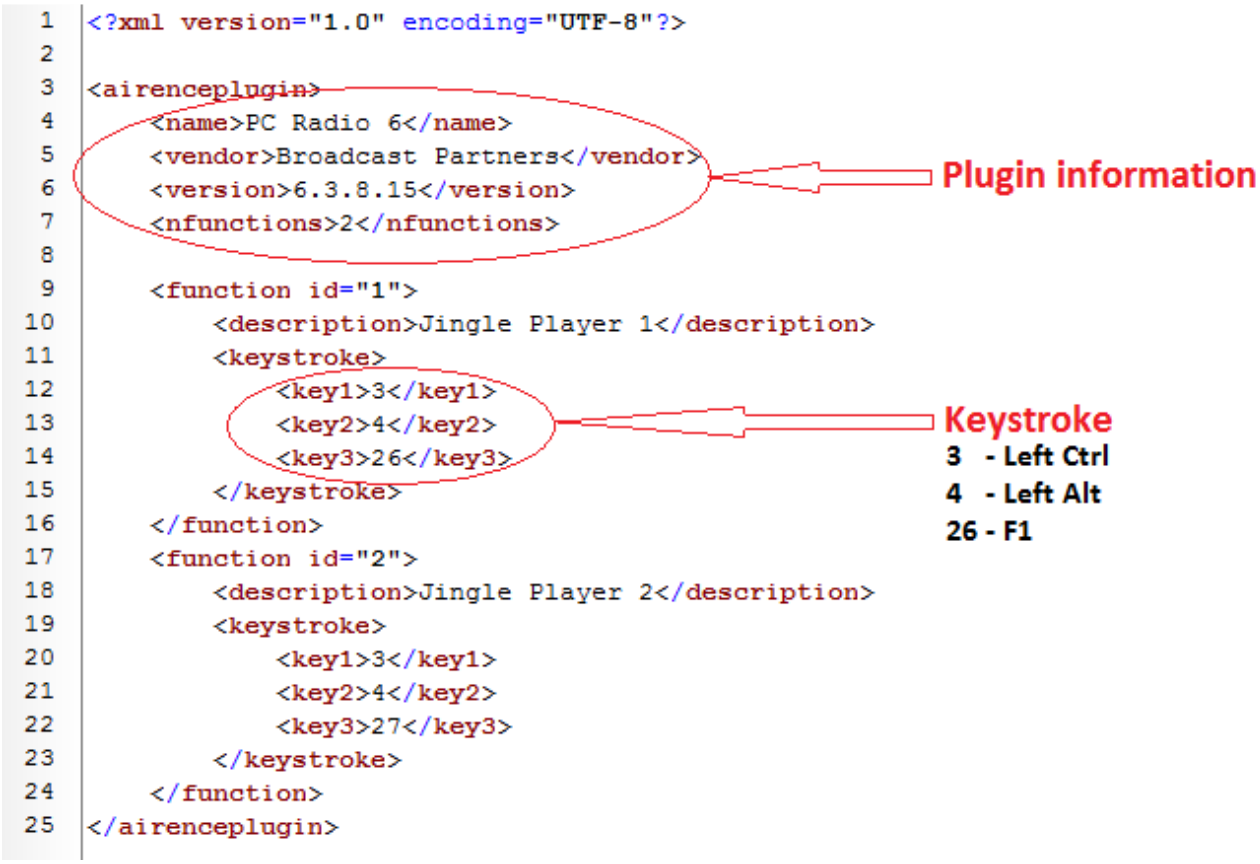

Figure 9. Example of playout plugin file

5.1 Writing plugin XML file

A plugin is specific for each playout software. If the used playout software is not available as a plugin one needs to create the plugin yourself. Since the plugins are represented in the XML format (which is readable for machine as well as human) it is not that hard to create one. It is advisable to use another plugin as template and adjust that file. In the above figure a basic example plugin file is shown with two functions. If more functions are needed simple copy and paste a function block (line 9 – 16).

The plugin starts always with a block of plugin information (name, vendor, version, nfunctions). The nfunctions element contains the number of functions which are included in the plugin. After the plugin information the functions are listed beneath each other. In the description field the name of the function needs to be filled in. This name will be used later on in the *Keystroke Configuration Editor* to choose a function from the list.

Finally, there is a keystroke element which contains three other elements *key1*, *key2*, and *key3* respectively. These last three elements contain together the keystroke to perform when the function needs to be executed. A keystroke can consist of maximum three keys. Each key element can be filled with a ID key value between 0 and 91. The ID value represent a key on the keyboard which can be found in the table in the figure below.

ID:	Name:	ID:	Name:	ID:	Name:	ID:	Name:
0	none	23	Delete	46	Key_8	69	Key_S
1	Esc	24	Page Up	47	Key_9	70	Key_T
2	Tab	25	Page Down	48	-	71	Key_U
3	Left Ctrl	26	F1	49	+	72	Key_V
4	Left Alt	27	F2	50	Backspace	73	Key_W
5	Left Shift	28	F3	51	Key_A	74	Key_X
6	Left Win	29	F4	52	Key_B	75	Key_Y
7	Right Ctrl	30	F5	53	Key_C	76	Key_Z
8	Right Alt	31	F6	54	Key_D	77	Numpad 0
9	Right Shift	32	F7	55	Key_E	78	Numpad 1
10	Right Win	33	F8	56	Key_F	79	Numpad 2
11	Space	34	F9	57	Key_G	80	Numpad 3
12	Enter	35	F10	58	Key_H	81	Numpad 4
13	Arrow Left	36	F11	59	Key_I	82	Numpad 5
14	Arrow Right	37	F12	60	Key_J	83	Numpad 6
15	Arrow Up	38	Key_0	61	Key_K	84	Numpad 7
16	Arrow Down	39	Key_1	62	Key_L	85	Numpad 8
17	Scroll Lock	40	Key_2	63	Key_M	86	Numpad 9
18	Print Screen	41	Key_3	64	Key_N	87	Numpad /
19	Insert	42	Key_4	65	Key_O	88	Numpad *
20	Pause/Break	43	Key_5	66	Key_P	89	Numpad -
21	Home	44	Key_6	67	Key_Q	90	Numpad +
22	End	45	Key_7	68	Key_R	91	Numpad .

Figure 10. Keyboard keys vs. ID