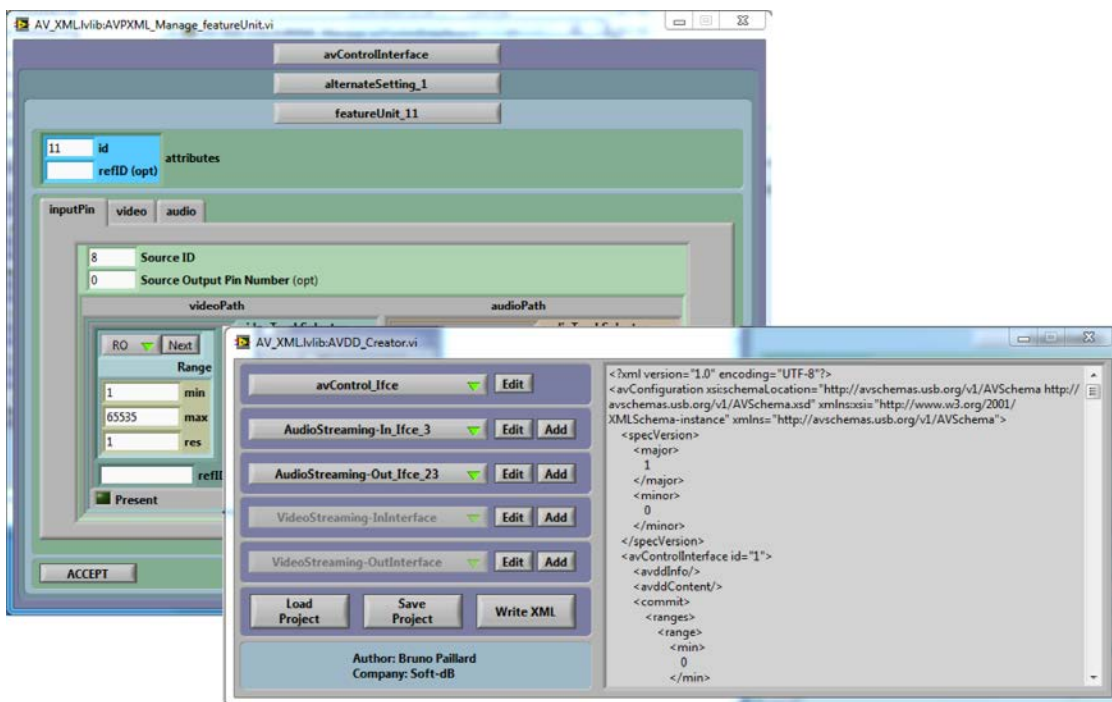


AVDD_Creator

User's Manual



December 10 2011

Bruno Paillard

1	INTRODUCTION	2
2	SUPPORTED OPERATING SYSTEMS	2
3	SOFTWARE INSTALLATION	2
4	PROJECT	3
5	NAVIGATION	3
6	DATA DISPLAY AND ENTRY	5
6.1	Hierarchical Containers	5
6.2	Disallowed Values	5
6.3	Feature Inference	5
6.4	Arrays	6
6.5	Tree-Lists	7
7	BUG REPORTING AND SUGGESTIONS	8

1 Introduction

The *AVDD_Creator* tool simplifies the generation of the *AV-Class Description Document* for inclusion into the AV device's firmware.

AVDD_Creator automates the entry of the device's features and characteristics, and creates an *AVDD* XML document that is compliant to the *AV-Schema* specification, the *AV-Core* specification and all the *AV-Format* specifications.

User-Interface features include:

- *Point & Click* graphical user interface.
- Entry of numerical values in natural scales (dB, seconds, Hz...etc.) and automatic conversion.
- Unit-ID auto-numbering and deconflicting. Upon creation of a new unit, the tool will propose an unused ID. It will not accept the same unit ID for more than one unit.
- Interface auto-numbering and deconflicting. Upon creation of a new interface, the tool will propose an unused number. It will not accept the same number for more than one interface.
- Channel number automatic sorting and elimination of duplicated entries.
- Embedded database of all legitimate formats.

2 Supported Operating Systems

The *AVDD_Creator* tool is supported on all versions of Windows from Windows XP, up and including Windows 7 at the time of this writing.

3 Software Installation

To install the tool:

- Unzip the file *AVDD_Creator_Installer_Vxxx*
- Navigate to and run *AVDD_Creator_Installer\Volume\setup.exe*

This installs the application, documentation and shortcuts in the *Start* menu.

4 Project

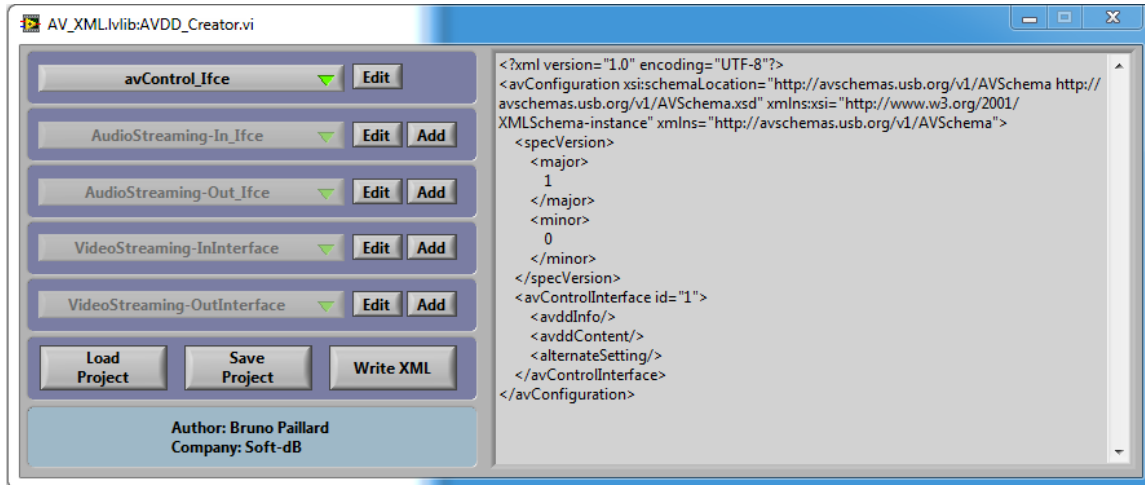


Figure 1

The project contains all the data relative to one configuration of the device.

Figure 1 shows the main application panel. Buttons on the left pane control the creation and edition of the device's interfaces. The right pane displays the corresponding *AVDD* document, in XML.

- **Save Project** Saves an internal database containing the entire device data entered so far. The data is saved in a file with a proprietary format and the *.avd* extension.
- **Load Project** Restores a device database from a previously saved *.avd* file.
- **Write XML** Creates the XML file corresponding to the device data.
- **Edit** Edits the interface selected with the menu ring to the left of the button.
- **Add** Creates a new interface of the type indicated by the menu ring to the left of the button.

5 Navigation

The device description is organized as a large tree, containing all the data relative to one configuration of the device. The user-interface allows the developer to navigate and edit that tree

At levels in the hierarchy where it is possible to do so, *Add* and *Edit* buttons allow the developer to create further branches, or navigate and edit the ones that already exist.

At every branch, a front panel is presented with features and controls relative to that branch, and possibly *Add* and *Edit* buttons to navigate to descriptions, further down the *AVDD* hierarchy.

For instance Figure 2 shows the panel corresponding to the *avControlInterface*. At the bottom of the panel, the *alternateSetting* menu allows the developer to add alternate settings of that *avControlInterface*, or edit existing ones.

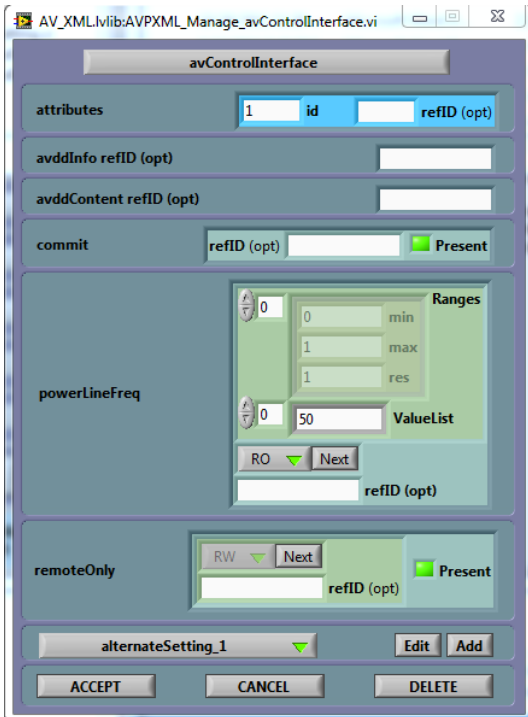


Figure 2

Pressing the *Edit* button shows *alternateSetting* No 1 (Figure 3), which itself branches-off to *inputTerminals*, *outputTerminals*, *mixerUnits*...etc.

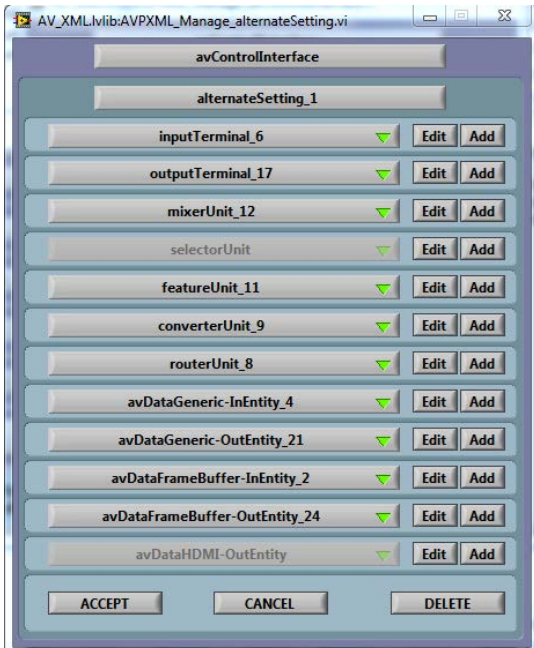


Figure 3

At the top of every panel is a set of title bars indicating the hierarchy to which that panel belongs. For instance Figure 3 shows the constitution of *alternateSetting* No 1 of the *avControlInterface*.

At the bottom of every panel is a set of three buttons:

- **Accept** Accepts all the data entered at this level, and all of its branches further down. This action closes the window and goes back to the window one level up in the hierarchy.
- **Cancel** Cancels all the modifications made at this level, and all of its branches further down. It reverts to the data that existed prior to any modifications made from that level down. This action closes the window and goes back to the window one level up in the hierarchy.
- **Delete** Deletes the whole branch, including all of its branches further down. For instance pressing *Delete* in Figure 3 would delete the whole *alternateSetting* No 1. This action closes the window and goes back to the window one level up in the hierarchy.

6 Data Display and Entry

The data is displayed and entered in a natural way, using numerical and ASCII field boxes, as well as menu rings and buttons.

6.1 Hierarchical Containers

The data is normally displayed using hierarchical containers. For example in Figure 4 the *min* field (1) is part of the *Range* container (2), which is part of the *videoTrackSelector* (3), which is part of the *videoPath* (4), which is part of the *inputPin* description (5).

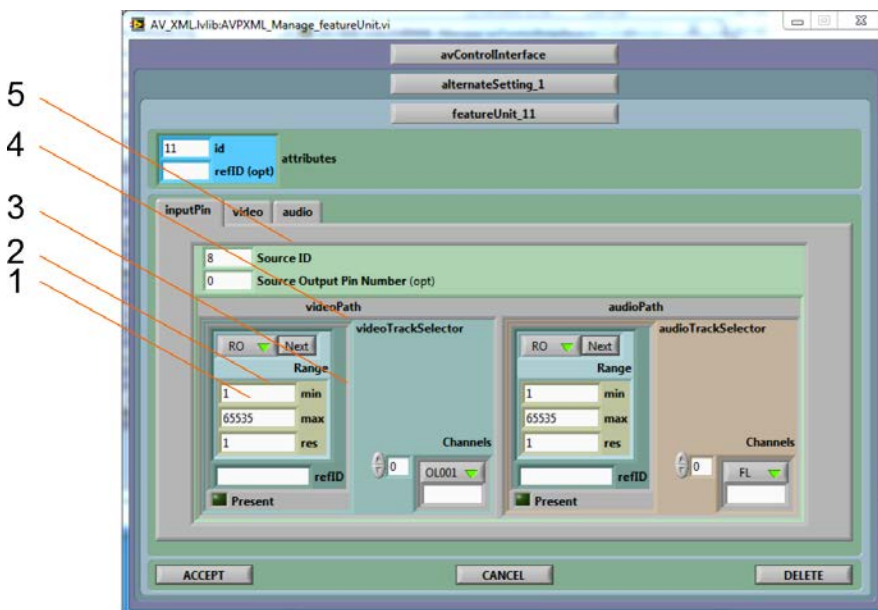


Figure 4

6.2 Disallowed Values

When attempting to enter data in a field, if certain values or settings are disallowed, the field will immediately revert to the prior value. This is the case for instance when trying to enter a unit-ID that is already assigned to another unit.

6.3 Feature Inference

Whenever possible the presence or absence of a certain control or element in the *AVDD* is inferred from the data entered. This is the case in Figure 2 where the mere presence of values in the *valueList* causes the *powerLineFreq* control to be present in the *avControlInterface*.

However in some cases it is not possible to infer the presence of a control from the data itself. In such cases an explicit *Present* check-box is shown in the corresponding container, and must be activated for

the corresponding control to be present in the AVDD document. Figure 2 illustrates this situation where the *avControlInterface* has a *remoteOnly* control, but no *commit* control.

6.4 Arrays

In some cases data is entered as a list or as an array of values or settings. Such data is displayed using an *array control*. An *array control* can be a simple list of channels, as is shown in Figure 5 (1), or it can be an array of more complex data, itself possibly containing arrays. In Figure 6 (1) the *audioModes* array contains a *Channels* array, indicating that every defined mode has an associated list of channels.

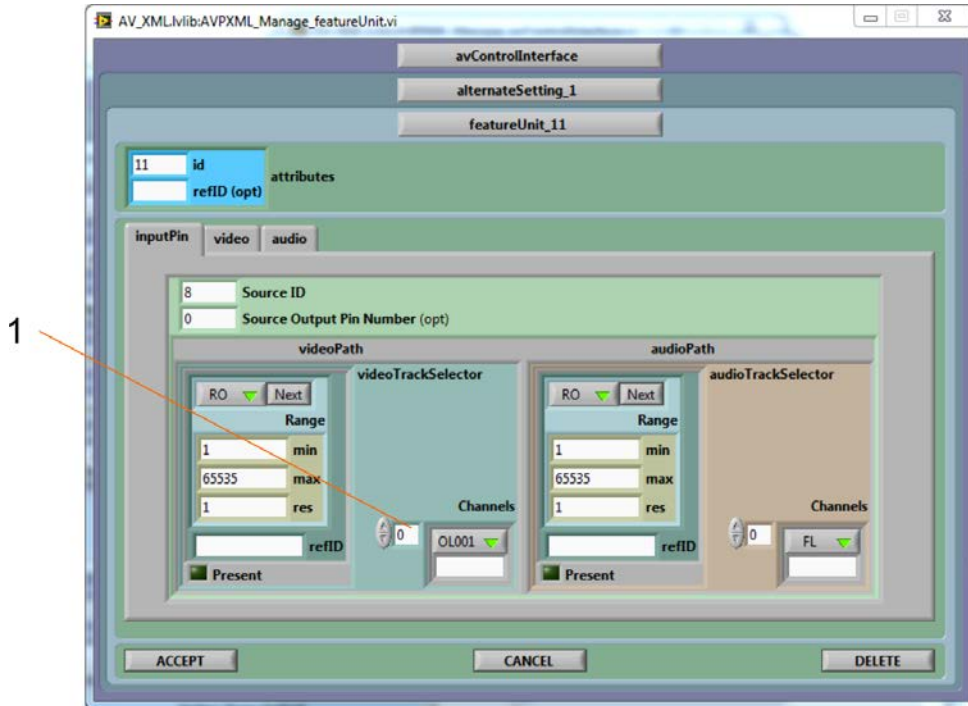


Figure 5

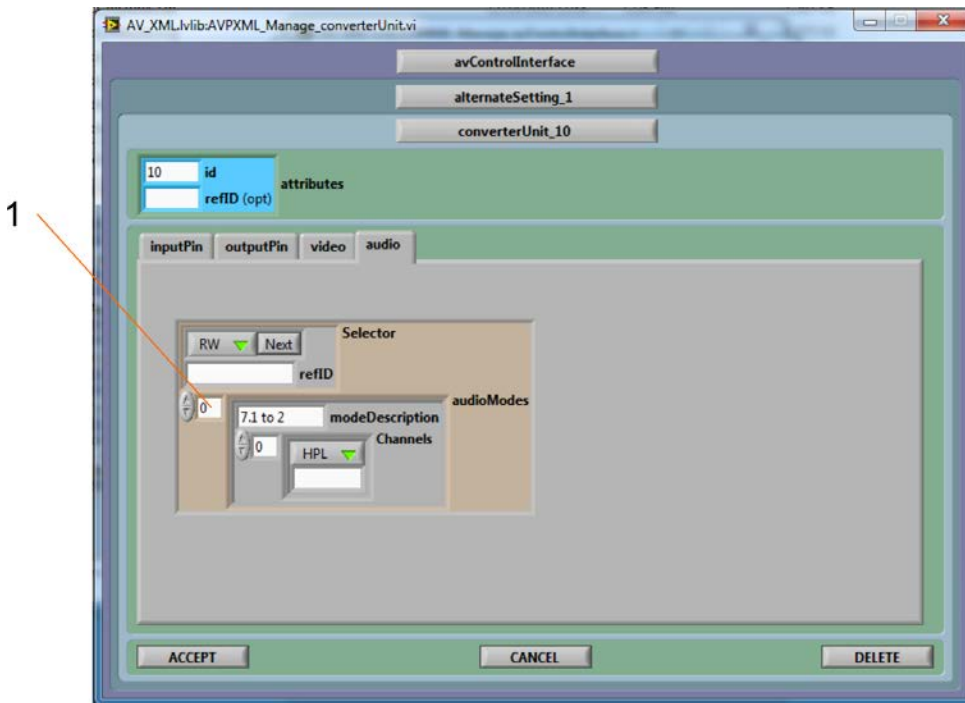


Figure 6

Every array has an index field at its top-left corner. This zero-based index indicates the element that is being viewed. To view another element, simply change the value in the index field, either by writing directly into the index field, or by using the up/down arrows at the left of the field.

When an array element does not exist yet, its data field is greyed-out. To populate an element, simply enter data into the greyed-out data field, and it will become effective.

To enter the first element into an initially empty array, set the index to 0 (the first element) and enter data into the data field. Then for the next elements, increase the index and enter the next data fields one by one.

When entering data at an index number that is further than the last existing element of the array (for instance when entering element number 3 into an array that is initially empty), the empty elements ahead of the element being entered are automatically populated with a default value for that type of array. In such a case it is advisable to examine the elements that have been automatically populated, to make sure the default value is what is intended.

To insert or delete an array element, set the index to the desired element, right-click on the data field and select the desired operation.

6.5 Tree-Lists

In some cases formats are defined as part of a hierarchical list or “tree list”. This is the case in Figure 7. In the left pane the *videoType11* format is *vcidFULL*, part of the *vsidRGB888* group. In the right pane the *videoFrameID* selected is *vfid1920x1080P*, part of the *foid2D* group. In such a case one and only one “leaf” format must be selected in order for the *AVDD* to be properly formed. Groups can be expanded or compressed using the plus or minus sign to the left of the group name.

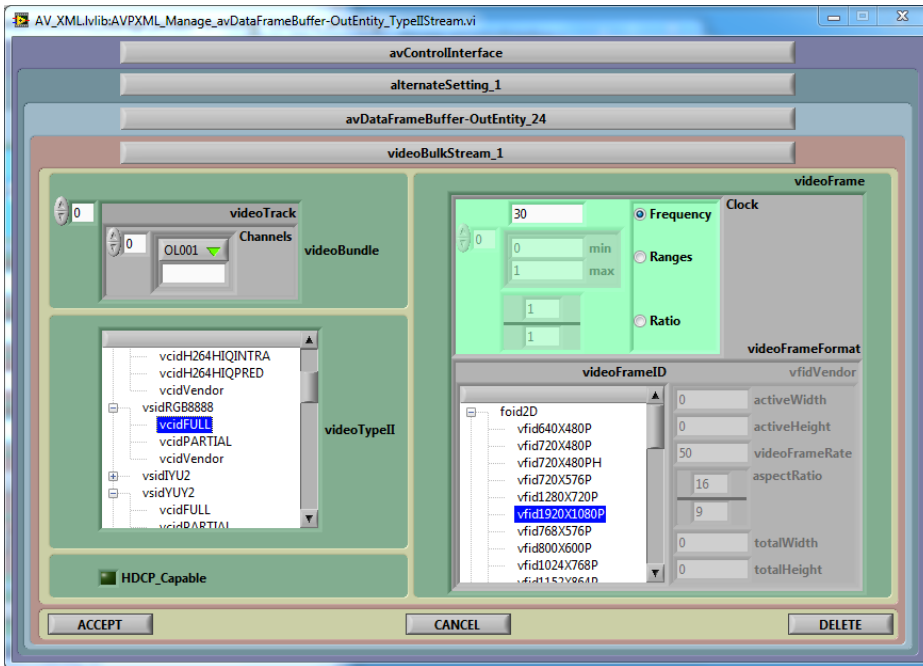


Figure 7

7 Bug Reporting and Suggestions

Send bug reports, comments or suggestions to:

b.paillard@softdb.com or bruno.paillard@convergenceinstruments.com

Please include a *reply-to* email address.