CHANNEL IN A BOX USER MANUAL

*(Version: **2.2**, October 2019)*

Current software versions:

AirBox Mega:	12184.0
CaptureBox Mega:	12184.0
TitleBox Mega:	12184.0
ListBox Mega:	12184.0
Clip Trimmer:	12184.0
PlayBox Live Inputs View	12184.0

This guide explains in detail all functionalities of the **_Channel in a Box_** modules.

Legal notice

The information in this manual is furnished for informational use only. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the prior written permission of PlayBox Technology UK Ltd.

The software, described in this manual, is owned by PlayBox Technology UK Ltd. It is protected by England and Wales Copyright Law, as well as by international copyright treaties, and may be used or copied only in accordance with the license agreement.

PlayBox Technology UK Ltd. provides this manual "as is" without any warranty, either express, or implied.

This publication may contain typographical errors or technical inaccuracies. While every precaution has been taken in the preparation of this document PlayBox Technology UK Ltd. assumes no responsibility for errors or omissions. Nor is any liability assumed for damages, resulting from the use of the information, contained herein. Changes are periodically made to the information herein. They will be incorporated in new versions of the manual. Please, check the **PlayBox** website regularly for User Manual updates.

PlayBox Technology UK Ltd. may introduce changes or improvements in the products, described in this manual at any time, without any special notice.

Please, address your comments or questions to:

PlayBox Technology UK Ltd
Brookmans Park Teleport
Great North Road
AL96NE Hatfield
United Kingdom

info@playboxtechnology.com

www.playboxtechnology.com

Contents

Channel in a Box User Manual

Channel in a Box User Manual

Channel in a Box User Manual logo

Preface

Dear **PlayBox** customer,

Thank you for purchasing our product! We would like to assure you that you have chosen the most cost-effective and versatile TV automation system on the market. As always, we are trying to stay close to our customers' needs, making sure they all receive adequate support and satisfaction. Your opinion about our product is an exceptionally valuable source of information for us. The ease of working with the **PlayBox** products results mainly from the suggestions and comments of our current respected customers.

This manual is structured into several sequential chapters, each aiming to ease the installation, fine tuning, and use of our products. We hope that you will enjoy working with it, and we are anxiously looking forward to receiving your feedback.

Please, send your questions, suggestions, and assistance requests to:

support@playboxtechnology.com

Style Conventions

- File names, software, documents or terms are written in *italics*
  - The data is written in the *settings.ini* file.
  - The file is located in *C:\Program Files\DMT***AirBox**
  - For further information read *Shortcuts* reference book.
  - The *VTR* is controlled via *RS-422*.
- Direct quotations from the computer screen are presented as follows:
  - **Menu Items and commands**
  - ***Tab/Page names***
  - **Column names (i.e. in a playlist or another grid)**
  - *Field names, check boxes*
  - ***Buttons***
  - Screen readings are written in [square brackets]
  - **The keyboard keys are enclosed in <> signs**
  - *Terms are defined in the Glossary at the end of the manual*
- The arrows, used in the setting procedures mean as follows:
  - ➔ A menu item follows;
  - ⇨ A page(tab) name follows;
  - → A field name, a check box name, or a value name follows.

Except for arrows, you can distinguish between the relevant menu categories also by the styles, listed above.

Introduction

MODULES

Channel in a Box consists of the following major modules:

AirBox

The most important part of a transmission automation system. It performs on-air playback and control. The module is designed for uninterrupted 24 hours / 7 days-a-week content playout from online media storage. It can also be used as a production server with interactive capabilities, such as Next, Jump, Cue, etc.

ListBox

A standalone playlist editor, designed for arranging, previewing, and trimming the available content without any hardware decoder requirements. It contains useful playlist features, such as text searching, printing, etc.

CaptureBox

Sophisticated ingest round the clock. CaptureBox controls *VTRs* via *RS-422* and automatically captures batches of scenes or programs. It also allows transferring a program to a tape precisely, by using frame-accurate timecode positioning. Needless to say, it also provides semi- or completely automated scheduled capturing, GPI interface, VU/peak audio meters, and a number of other unique tools.

TitleBox

TitleBox provides fully automated or interactive graphics overlay. It can also be synchronized to ***AirBox*** playout sequence. It allows insertion of text and dynamic graphic information in accordance to external data sources. Text information is received and rendered into high-quality graphics on-the-fly.

AIRBOX

GETTING STARTED

Quick Start

- Verify all connections and start the workstation;
- Make sure that you have suitable content available in the media folders;
- If you do not have any available, use the sample clips from the *PlayBox CD*;
- Launch *AirBox*;
- Click on the *Add* button;
- Select the files you wish to playout and click on the *Open* button;
- Click on the *PLAY* button;
- You should now see *AirBox* video on your video output.

Congratulations! You have just launched your first *AirBox* playout session!

PREAMBLE

AirBox is a universal SD/HD content playout and streaming software**.** Depending on the hardware platform and supported files in the playlist, there are different types of *AirBox* playout (playback modules).

Currently, there are 3 major modules:

Multi-format playout SD/HD

Multi-format playout allows including in the playlist files with different types of video and audio compressions, different resolution, different frame rate, etc. If the parameters of the file are different from the parameters, set for AirBox playback, then the files are transcoded in real-time (during the playback) to match to the parameters set.

Depending on the hardware installed, there are several multi-format modules (plug-ins). Check our site https:/playboxtechnology.com for the latest supported hardware platforms and for their supported features.

Pumping playout

Pumping playout allows inserting only content with MPEG1-encoded or MPEG2- encoded video into the playlist. There are three Pump plug-ins: *Data pump*, *IP pump* (LAN card based), *ASI pump* (**Deltacast ASI card** based). Usually, the working module of *AirBox* is already selected and set up when you buy a turnkey system from *PlayBox Technology UK Ltd*.

In case, you are going to select a working module alone, go to ***AirBox Menu*** -> Settings -> Modules window and select the preferred module.

The *AirBox* playout is defined in the **AirBox ➔ Settings ➔ Output** menu.

Live input

 Live Input playout allows you to configure up to 128 live presets, which are global for the machine, and therefore can be used with any client, for example *AirBox*, *CaptureBox*, and the *Multi AirBox Manager*, configuring many different live inputs. There are two major types of life sources that can be set here – fixed and dynamic. The properties of the first type are fixed, i.e., the video parameters and the number of streams cannot be changed. The same parameters of dynamic sources, on the other hand, change during playout. Check how to define your live inputs in the relevant section below.

Channel in a Box User Manual

USER INTERFACE

The main interface window is divided into several areas:

Title Bar

This is the topmost horizontal bar of the **AirBox** window. It contains abbreviations of all options, enabled on the dongle, the **AirBox** channel number, and the name of the currently executed playlist file. If the playlist has been changed, but not saved, there is an asterisk (*) after the playlist name.

Menu Bar

The **MENU BAR** is described in detail in the Menu Bar section.

Multi-purpose zone

The multi-purpose zone is situated to the left of the master counter. It consists of several tabs that allow quick navigation through the most frequently used functionalities in **AirBox.**

Use the arrows in the upper right corner to scroll the tabs.

> **TIP (!)** To show/hide tabs in this zone, right-click on it and check/un-check the ones you need.

Preview

This tab shows the currently playing video.

Channel in a Box User Manual

The Video window can be moved outside the main window by checking *External video window* from the **View** menu. If you accidentally close the window, you can view it again by checking the *Show external video window* in the **View** menu.

Some *AirBox* configurations do not provide such video overlay functionality.

In some cases you might want to disable the window, since it consumes additional resources from the system. Then you will have to uncheck *Show External Video Window* in the **View** menu.

Counters

This tab contains all counters that can be found in the **View** menu. Their names are written in the lower left corner. Right-clicking in the Clip timer or in the Block timer will invoke a context menu to switch it to count-down mode.

Please, check the [View menu section](#) below for a detailed description of the counters.

Current Playlist
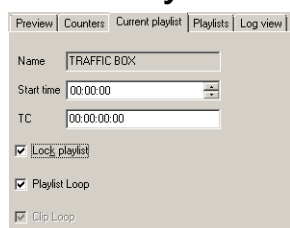
This tab contains information and simple controls about the currently loaded playlist.

Name – this is the title of the currently loaded playlist. It is different from its filename. You can change it in the dialog that appears after double-clicking in the string.

> **IMPORTANT!** When you rename the playlist, do NOT use special symbols (like '/',\' and ','). *AirBox* will NOT accept names with such symbols and it will return an error.

Start time – by default, it is set to zero. You can set a different start time for the playlist by either entering the value, or using the arrow keys.

TC – this is an offset that will be applied to the time code, which is broadcast over the network – see **Settings** menu ➔ [General](#) section.

If the ☑ *Lock playlist* box is checked, it will prevent you from editing the currently loaded playlist, as well as from loading another playlist. Use it to prevent accidental changes of the playlist during playout.

☑ *Playlist Loop* – if checked, when the playback of the last clip in the playlist is complete, playback will continue from the beginning of the playlist again.

☑ *Clip Loop* – if checked, the currently playing clip will be repeated until unchecked. When *AirBox* is in stop mode, this check will loop the first clip to be played after resuming the playout (i.e. the last played before pushing the *Stop* button).

Channel in a Box User Manual

Playlists

In this tab you can view all playlists that are found in the currently selected playlist folder (displayed in the string). To change the folder, click on the
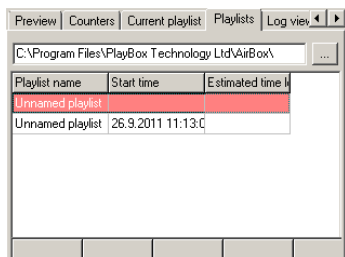
Browse [...] button to the right of the string.

TIP (!) Double-click on a playlist line to load the relevant playlist in ***AirBox***. Please, note that this will interrupt the playback!

Log view

This tab displays a list of all events that have occurred during ***AirBox*** operation.

ShotBox

This tab accommodates the live show clipboard. Here you can view and operate with the ***custom command*** buttons. It duplicates the live show clipboard window that opens from the **View** menu. Please, check the ShotBox section for details.

AirBox Playlist

AirBox works with play playlists (play order).

You can easily create a playlist manually, just by drag-n-dropping files into *AirBox* grid (or in ListBox).

It is also possible to create a playlist with any third party application software. It is necessary that this application can create playlists in a format, compatible with *AirBox*.

The *AirBox* playlist format is described in details in *AirBox SDK*. For obtaining *AirBox SDK*, please contact our *PlayBox* support team (support@playboxtechnology.com).

AirBox also provides a possibility to import playlists from simple text files (tab delimited or comma separated text files).

There are two workflows with *AirBox* playlists: ordinary playlist and daily playlist. The difference is that the daily playlist is loaded automatically at an assigned day and time, while the ordinary playlist should be loaded manually.

Daily playlists require specific naming structure in order to be played on the desired date and time automatically. The structure is: YYYY_MM_DD_HH_MM_SS, where YYYY stands for year, MM-month, DD-date, HH-hour, MM-minutes and SS-seconds.

For example, if the name of the playlist is "2011_11_20_14_00_00.ply", it will be loaded exactly on November 20th, 2011 at 14.00 h (2.00 pm).

The AirBox playlist contains video clips and events (commands), which are described in details further in this chapter.

Playlist Grid

This is the main part of the *AirBox* interface, dedicated to the playlist (play order) visualization.

When you launch *AirBox*, depending on your startup settings, you will see an empty window, and you will have to load (create) a playlist, or you will see the opened playlist.

The Grid Columns

If there is a playlist, loaded in *AirBox*, you will see information about the clips in different columns:

Start Time

This column contains each clip's starting time. During **Stop** mode, it shows the current time for the clip that will be played first when the *Play* button is pressed. The time refreshes every 2 seconds. The following clips hold positive times calculated according to the current time and the duration of the preceding clips. During **Play** mode, this column shows the actual starting time for each clip. If there has been a *Jump* command or a clip has been trimmed during **Play** mode, the starting times of the following clips will be recalculated accordingly.

If for any reason a file is removed or renamed after the playlist is loaded, it will be considered missing and hence - colored in red. **AirBox** will skip it and will jump to the next available clip. The **Start Time** column will be recalculated accordingly.

Duration

Shows the actual duration of each clip. Naturally, if a clip has been trimmed, its actual duration will be reduced. The duration of the part to be shown will be displayed in the playlist and will be used for calculation of the starting times of all following clips.

Type

Shows the type (*such as MPEG*) of the clip – this is quite useful to determine the major type of compression, since many decoders do not allow different compression types to co-exist in the same playlist.

Category

Contains category information. This information could be entered manually from the user into the playlist or could be fed from some outstanding database (for example **DataBox**).

NOTE: If category is taken from **DataBox**, you can define a field color for the clip, using **DataBox (Options**⇨**Default values**→**Color to AirBox)**.

The C*ategory* and its color can also be defined manually in **AirBox (or ListBox)**. You can fill-in the entire clip rows with their Category colors. In order to do this, go to **AirBox Settings**➜**General**⇨**Interface** and select ☑*Color entire row by category*.

Title

Shows the clip's title or filename.

Location

Shows the file location (full path).

Star

Contains data, describing the artists in the clip.

Notes

Contains additional data.

NOTE: If the Note is taken from **DataBox**, you have to specify in **DataBox (Options** ⇨ **General** →*Fields to AirBox Notes*) what exactly field to be transferred.

(!) TIP You can change the columns' order by *drag-n-dropping* them.

(!) TIP A double-click in the first column that contains the items' numbers will invoke the clip properties dialog. Double-clicking in any other column will open the **Clip Trimmer** (if it is enabled in **Settings** menu➜**General** (see the description further in this manual).

Color coding

The color-coding of the playlist's rows provides additional information to the user:

▪ **Pink row** – outlines the currently playing clip. It is visible in **PLAY** and **PAUSE** modes only.

- **Yellow row** – it is visible only while switching between clips. It shows which clip will be played next. In that moment, the yellow clip is being actually fed into the decoder circuit. This happens during the last couple of seconds before the end of the previous clip. The clip cannot be moved, trimmed or deleted anymore. Just for comparison, competitive playout solutions tend to permanently lock the clip after the currently playing one, which reduces on-air flexibility and your last-minute decision freedom.
- **Blue row** indicates the currently selected clip – all actions, i. e. *Jump*, *Move Up/Down* will be applied to it.
- **Red row** highlights those files in the playlist that are missing/invalid (not supported from the currently installed hardware). The playback skips such files and plays the next available clip.
- If the file is missing, you can set the time interval for automatic check for missing files (**Settings➔General**⇨*Missing Files*➔Check Missing files every [...] sec*)*.
- **Light cyan row** outlines the playlist events when selected.

> **NOTE:** You can change these colors to fit your personal preferences in **Settings ➔ Colors ⇨ *Playlist*** tab.

Drag-n-Drop!

One of the main features of the grid is the *drag-n-drop* functionality. It allows dragging clips within the playlist and from one grid to another (from **DataBox/ListBox** to **AirBox** or from any folder to **AirBox**). If <Ctrl> is being held while dragging a clip in the playlist, the executed operation will be **Copy**. By default, each internal (in the grid) dragging means **Move** and dragging between two windows (similar to **Windows Explorer**) is **Copy**. Clips that have already been played-back are grey and cannot be moved, but can be copied.

Pointing with the mouse cursor at a playlist's row displays a hint containing the clip's file name and IN /OUT timecode values.

Double-click over a clip invokes the *Clip Trimmer* (if it is enabled from **Settings ➔ General** ⇨ **General** → ☑ *Use clip trimmer*). If you trim a clip in the playlist, the change will not be permanent, but valid ONLY for this playlist. If you add the same clip to another playlist, it will appear un-trimmed there.

A detailed description of *Clip Trimmer* can be found further in this user's manual.

If the use of *Clip Trimmer* is not enabled, the clip's properties dialog will open.

Right-click

```
Add                    ▶
Insert                 ▶
Delete          Del
Jump            Alt+Space
Loop
Auto update
Override QC restriction

Cut             Ctrl+X
Copy            Ctrl+C
Paste           Ctrl+V

Clip Trimmer…   Alt+Enter
Properties…     Ctrl+Enter
```
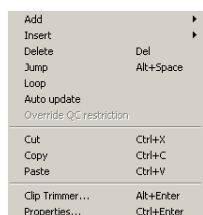
Right-mouse click over a row invokes a pop-up menu of playlist and events commands:

- **Add** – activates a drop-down list for appending various items to the end of the playlist. See their detailed description in the Edit menu section.
- **Insert** – opens a drop-down list for inserting various items before the currently selected item in the playlist. The Add and Insert drop-down lists are identical.
- **Delete –** deletes the selected playlist row(s); a dialog asks confirmation before the deletion.
- **Jump –** click it to interrupt the currently playing clip and jump to the currently selected (highlighted) item in the playlist.
- **Loop** – marks the selected clip for repeating until this option is cancelled. To remove the check, just right-click in the clip and choose *Loop* again.

> **NOTE:** This action differs from the *Clip Loop* button **#**, which loops the currently *playing* clip (not the currently *selected* one).

- **Auto update** – use it to include the currently selected clip(s) in the automatic file duration checking. It will update the duration of your clips if the files associated with them have been, or will be, changed.

> **NOTE:** You have to enable this option first in **Settings➜General** ⇨ **General** ➜☑*Auto update duration of marked files*. The file checking interval should also be set there.

- **Override QC restriction –** if you have defined a QC restriction in **Settings➜General** ⇨ **Quality Control**, and a particular file does not satisfy these restrictions, it will be marked red, just like a missing file. Thus, this menu option allows you to override the QC setting and play this particular file. The setting is applied only once during the current playout.
- **Cut/Copy/Paste –** these options are self-explanatory.
- **Clip Trimmer** – opens the *Clip Trimmer* application (if it is enabled in **Settings➜General** ⇨ *General* ➜☑*Use Clip Trimmer*. Otherwise, the *Clip Properties* dialog will open). Find detailed information about the **Clip Trimmer** in the section of the same name at the end of this manual.
- **Properties** – opens a window with clip's properties.

Clip Properties

Most of the playlist consists of clips. All metadata, connected with any clip, could be found in the *Clip Properties* window. It is opened by right-mouse-clicking over the clip, or through the **Edit Menu**.

The *Clip Properties* window has five tabs: **Source**, **General**, **Fixed-Time**, **Transition**, and **Metadata**.

All of these tabs allow you to save your settings, so that you do not have to define the settings for each file every time. In order to do that, once you have defined your desired settings, enter the **Save current settings to new preset** 🖫 button in the *Settings preset* field, and enter a name for the selected preset. Once you have saved some settings for your input, you can load them via the **Apply settings from selected preset** 🟢 button. You can also remove a preset via the **Delete selected preset** ❌ button.

Source
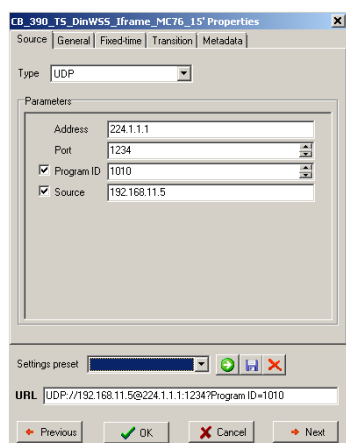
The Source tab contains a basic clip description information. The *Type* fields shows whether the input is from a **File**, a **Live**, or a **UDP** source.

If the input is a **File**, the *Parameters* field shows the file *Path*.

If it is a **Live** source, then the *Address* of the source is indicated in the field below. This Address corresponds to the *Name* of the live stream, as it is defined in the **Settings ➜ Output** ⇨ **Live Inputs** tab.

When the Type of the source is **UDP**, by default, and the user can enter the *IP Address* of the source machine and the number of the *Port*, which receives the input. There are also two additional options for this type of input:

☑*Program ID* – if you check this box, you can to choose which program to be output by **AirBox**, if the stream has more than one channel.

☑*Source* – check this box if the source broadcasts more than one stream, to be able to enter the *Source* Address of your desired stream.

<u>**General**</u>
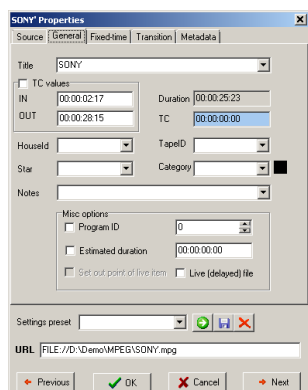
The first line in the **General** tab contains the *Title* of the selected playlist item.

The *IN*, *OUT* and *Duration* strings contain the IN point, OUT point, and clip duration data. You can change these values in order to make your clips shorter. The *Duration* is calculated automatically from the *IN* and *OUT* values. Furthermore, if you change the *Duration*, the *OUT* point will adjust accordingly.

If there is information about the time-codes of the first and last frame in the clip, you can check the ☑*TC Value* check-box and then into the *IN* and *OUT* strings you will see these time-codes.

The *TC* field is necessary when there are subtitles to be displayed together with the clip, as the time code is the reference for displaying them. If you change the *TC* value, it will be saved in the playlist for future use.

When you use subtitles, you should also enter a *TapeID* value – read more about this value in the **SubTitleBox** section further in this manual.

You can view and edit the *HouseID*, *Star*, and *Category* information in the relevant strings. The colored square next to the *Category* string represents the color, which will be displayed in the **Category** column of the playlist. You can change this color by double-clicking in the square.

Use the *Notes* string to type your notes about the clip.

NOTE: Content metadata fields, like *Category, Star, Notes,* could be filled directly into the **AirBox** playlist. In this case, the information is saved only for the current playlist. If you use the same file in the new playlist, you have to enter metadata again, unless you have saved it via the *Settings preset* field.

It is also possible to keep all metadata in **DataBox**. Then, when you create a playlist by drag and drop files from **DataBox**, the metadata will be transferred too. You just need to define in **DataBox** options the fields for transfer.

Another possibility is to use external user's database (traffic, MAM, etc.) for preparing playlist. Then you can transfer metadata from this database to playlist. The exact metadata transfer depends on the term of integration of this third party database.

Keep in mind, that if you change some metadata inside the playlist, the change will be valid only for the current playlist. The changes will NOT be transferred back to the original database**.**

Channel in a Box User Manual

☑ *ProgramID* is checked when the stream is MPTS (it could be a live MPEG2 stream coming from the network or a file). Type the *ProgramID* (for example, 12060), not the number of the program (i.e., not 1, 2 or 3).

If you are going to playback a clip while it is being captured, check the ☑*Live (delayed) file* check-box. If necessary, specify the ☑ *Estimated duration* of this live playback and then check the ☑ *Set out point of live item* box.

> **WARNING!** Enable the ☑ *Estimated duration* option ONLY when your source video is a <u>live stream</u>. Otherwise, an error will most likely occur.

Most of the properties in this window will be displayed in the playlist grid. If you save the playlist, these values will be saved as well.

> **NOTE:** Clicking *Previous*, *OK* or *Next* will apply all changes you have made to the current clip properties! If you want to discard them, please click *Cancel*.

Fixed time
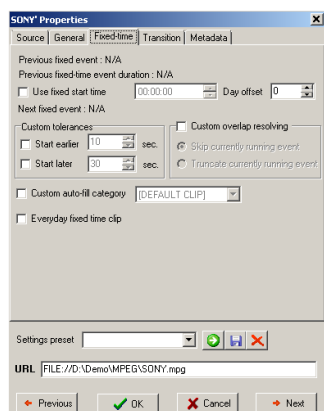
The **Fixed-time** tab allows for setting an exact start time for the particular clip.

> **IMPORTANT:** If you do not choose any of the Custom settings for the fixed-time event, the Default settings will apply. A detailed description of the default settings is available in the <u>*Fixed Time event*</u> **Settings** section further in the manual.
>
> **WARNING!** When using fixed time events, the playlist should start and end within the same calendar day (i.e., the playlist should not go beyond midnight). Otherwise, we cannot guarantee the proper operation of the fixed time logic.

When using fixed start times, you always have to account for the other fixed times in the playlist. Therefore, the first two lines in this tab contain information about the preceding fixed time event and its duration; and the fourth line prompts the start time of the following fixed event (if any). If you try to set a start time, which is in conflict with the surrounding events, the time-picker background will become red. A warning message will appear upon clicking *OK,* and the allowable start time will be prompted until you set it.

The two spin-boxes are situated between the *Previous* and *Next fixed event* rows describing the current fixed-time event. Check the ☑*Use fixed start time* box and enter the time. In the *Day offset* spin-box, enter the number of days to go prior to executing the fixed time event. Day offset zero means the date of the **AirBox** startup!

When using fixed start times, two types of time conflict might occur – overlapping of playlist items or gap in the playlist. In the first situation there is not enough time to playout all clips in the playlist before the fixed-time event starts. Therefore, you can choose between two ☑*Custom overlap resolving* options:

⊙ **Skip currently running event** – when this option is selected, **AirBox** will skip all clips that cannot be played out as a whole. Let us assume that the time remaining till the fixed start is 2 minutes and all the clips before it are longer. **AirBox** will skip them and the 2-minutes gap will be filled with the ☑*Custom auto-fill category* (see below).

⊙ **Truncate currently running event** – *AirBox* will playout as many clips before the fixed time event as possible. The currently playing clip will be truncated when the fixed time comes.

In the second situation, when there is a gap in the playlist (either because the content is not enough or you have selected to skip the currently running events) you will need a ☑ *Custom auto-fill category*. This drop-down list contains all auto-fill categories, previously prepared by you, and a [Default Clip] entry. Check the *Auto-fill settings* section to learn how to create these categories.

In case you can accept some deviation of the fixed time, check the relevant ☑*Custom tolerances* box and describe it. In this way you can ease the complex operations when it comes to a few seconds differences.

☑ *Everyday fixed time*s needed in case you loop your playlists. If you leave this check-box empty, the fixed time will be executed only during the first loop.

> **NOTE:** Please, note that you cannot use simultaneously *Day offsets* and *Everyday fixed time*.
>
> **WARNING!** DO NOT use fixed time events within a bookmark's time range!

Transition
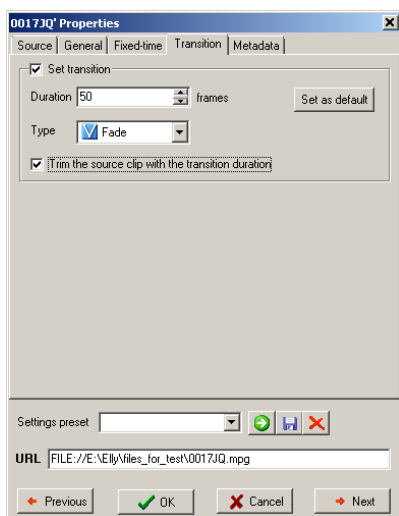
The **Transition** tab accommodates settings, related to transitions between clips in the playlist. The transition settings are applied to the beginning of the currently selected clip.

Check ☑ *Set transition* to enable the transition.

Specify the transition *Duration* (in frames) in the spin-box below.

The *Type* drop-down list contains all currently supported transitions. Choose the one you need from here.

☑ *Trim the source clip with the transition duration* – This checkbox is enabled only when you apply transition for the first time. Check this box to shorten the previous clip with the duration of the transition ensuring that enough video and audio content will be available for the transition. When this box is checked, the previous clip will be trimmed in accordance to the frames, set in the *Duration* spin-box when you press OK button.

Channel in a Box User Manual

For example, assume that your AirBox is running in PAL mode (25 frames per second), and you have set a new transition with *Duration* of 50 frames, i.e., 2 seconds, to Clip 2. Assume further that the preceding item in the playlist is Clip 1, which is 10 seconds long. Once you press OK button Clip 1 will be automatically trimmed, so that its duration is reduced by 2 seconds, to 8 seconds. Thus, when Clip 2 starts, it will begin with a transition, displaying the last 2 seconds of Clip 1.

NOTE: The clips that precede clips with applied transition in the playlist are trimmed immediately after OK button of dialog applying transition is pressed. Thus, if you want to change any transition duration during playout, you have to trim its preceding playlist item manually, by modifying the duration settings in the General tab of the Clip Properties dialog.

NOTE: Even if you have set the *Type* of transition to **None**, the clip preceding the transition will still be trimmed if the source clip with the transition duration option is checked.

If you want to keep these settings for other transitions in the playlist, press **Set as default**.

NOTE: Transitions are supported only for the new mixed playout plug-ins!!!

Metadata
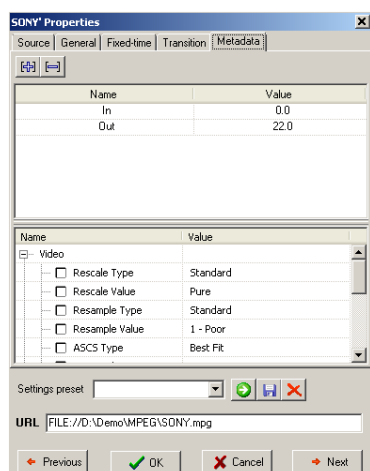
The **Metadata** tab provides room for additional information fields related to the currently selected clip.

Upper part of the window:

In the upper part of the window you can include additional metadata, related to the content of the clip. This information can be entered manually or by dragging from **DataBox**. In all cases you need to specify some metadata categories that will accommodate the relevant metadata values.

For example, the **Name** of a metadata category is [Producer] and the **Value** is [a person/ a company].Thus, if you have the producer's name in **DataBox** [Person] and map it to **AirBox** [Producer], that information will be transferred automatically when you drag clips from DataBox to the playlist grid.

You can use these fields in graphic rules, to present information that does not appear in the playlist columns.

Click on the [Metadata name] cell to enter the name that will be used as reference in the **Graphic Rules**. In the [Metadata value] cell enter the additional information that should be presented in the graphics while this clip is on air.

Please, refer to the Graphic rules section for detailed explanation about the metadata implementation.

Except for showing information about the currently playing clip, metadata categories can be used for changing the links to picture and sound objects in **TitleBox**. Thus, you can change the image, shown in a picture object, or the sound that is played in a sound object.

All you have to do here is:

1. Specify the Metadata name in the left column

For example: [new picture]

2. Specify the file path to the new media to load in the column to the right.

For example: [E:\Pictures\pic\A Dark Night.jpg]

NOTE: The file path should be local for the **TitleBox** machine!!!

Thus, you will have [new picture] [*E:\Pictures\pic\A Fantastic Voyage.jpg*] in one of the rows. Then, go to **Settings➔Graphic rules** and set the conditions and action offset time. Please, check the Graphic rules section for details.

IMPORTANT: Make sure not to use tab characters in the metadata fields! Otherwise, the saved metadata will not be interpreted correctly.

<u>Lower part of the window</u>

The lower part of the **Metadata** window is used for entering specific metadata, related to video parameters of the current clip.

The metadata is grouped in three main categories – [Video], [Mixing], and [Audio].

Video Metadata:

☑ *Rescale Type* – rescaling type for the video.

☑ *Rescale value*– if the rescale type is *Custom*, then you can add a specific rescale value in this field.

☑ *Resample Type* – resampling of the video bitrate. If the resample type is *Custom*, then you can add a specific resample value from the next field

☑ *Resample value* – if the resample type is *Custom*, then you can add a specific resample value in this field.

☑ *ASCS* (Aspect/Size conversion strategy) – this is a type of conversion of video files with a different aspect ratio than the one, set for the global playout. The values here are the same as those in the Output Video Setting dialog.

☑ *Field order* (input) – this is the field order of the input file. If there is no info about the field order in the file, or if it is wrong, you can enter it here.

☑ *Item input Aspect* – the aspect ratio of the current clip. If there is no info about the aspect ratio of the file, or if it is wrong, you can enter the value here.

Mixing

This metadata is used only if a *Shift Transition* is selected in the **Transition** property tab. Here you can define the ☑*Shift Type* and the ☑*Shift Direction* of the "Shift" transition.

Audio metadata

☑ *Language*– the language of the sound

☑ *Level* – the audio level of the current clip in dB.

> **(!) TIP:** By using audio level metadata, you can specify a different audio level for each file in the playlist.

> **For example**, you can use it to decrease only the audio level of the commercials in the playlist.

> **NOTE:** On playout, the clip metadata (if there is such) settings have a higher priority than the output settings, created in the **Output settings** dialog.

> **For example**, if the aspect ratio in the main video output is set to 4x3, and the ASCS (Aspect/size conversion strategy) is set to "*Balanced/Letterbox*". This means that if there are files with an aspect of 16x9 in the playlist, they will letterboxed at the output.

Let us assume that you have a particular file with aspect 16x9 in the playlist, which you do NOT want to be letterboxed, but rather you want it to be zoomed. Then, in the properties of this particular file, check the ☑ *ASCS* check-box and select "*Fullscreen/Zoom*" in the respective value field.

> **NOTE:** Values in *Field order* (input), as well as *Item input Aspect*metadata do not have any effect on the playout. These fields are connected to the file in the playlist and are used if the information for "*Field order*" and "*Aspect ratio*" of the file is not available in the file itself, or if it is wrong.

Events
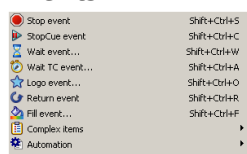
There are two types of events in relation to **AirBox**, internal and external. They are separated by a thin line –the internal events are listed above, and the external ones are below.

Internal Events

Stop event

Inserting such an event in the playlist will stop the playback automatically at reaching this point of the playlist. This function is useful for news and other interactive sessions.

StopCue event

Automatically stops the playback upon reaching this point of the playlist and shows the first frame of the next clip.

Wait event
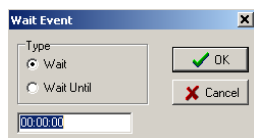
Temporarily stops the playback. A dialog box appears to specify the type of "waiting":

Wait means that playback will stop and standby for a certain period (hh:mm:ss). For example, 00:01:00 means that the playback will wait for 1 minute;

Wait Until means that playback will stop and automatically resume at a defined time (hh:mm:ss). For example, 14:00:00 means that the playback will start at 14:00:00 o'clock.

Wait TC event
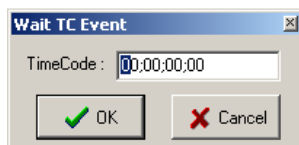
Automatically stops the playback and resumes it upon reaching the specified time code (e.g. coming from an external timecode generator). The message to the right will be displayed on the *AirBox* window and the Master counter will continue running. See the LTC reader description in the **Settings** menu ➔ **Timecode** section further in this chapter.

The TimeCode format here is HH;MM;SS;FF, where H stands for hours, M for minutes, S for seconds, and F for frames per second.

**IMPORTANT:** Depending on the standard you are using (PAL, NTSC, etc.), make sure to enter the corresponding number of frames in **Settings ➔ General ⇨ Interface ➔ Frame rate** (25 for PAL and 29.97 for NTSC).

NOTE: Be aware that when you are using PAL the timecode separators are ':' and when you are using NTSC they are ';".

Logo event

◉ *Logo On* – Starts displaying the logo, set in the preceding *Logo Preset event*. If there is no preceding *Logo Preset event,* the first logo preset from the list (in **Settings ➔ Logo…**→*Logo Presets*) will be displayed.

◉ *Logo Off*– Stops displaying the logo.

◉ *Logo preset* - Sets a logo preset, which will be displayed after initiating the *Logo On* event. The preset number corresponds to its number in the preset list in **Settings ➔ Logo…**→*Logo Presets*.

If you check the ☑*Activate now* box in the event dialog box, the logo will appear immediately. You can also activate the logo before or after the beginning of the following clip by setting an offset to it.

Return event

This event will return the playback at the position before a previously executed Jump. For example, if you have a movie in your playlist, and at a certain time you decide to jump to a block of commercials somewhere in the playlist, and then go back to the same position in the movie, you can insert a Return

event after the commercials you would like to play. Thus, the movie playback will resume after the end of the commercial break from the point before the jump.

<u>Fill event</u>

Opens the *Fill event properties* dialog. For more information, see section **Settings** menu ➔ **Auto-fill…** <u>further in this chapter</u>.

> **NOTE:** The minimum fill time is 2 seconds. In case you have gaps that are less than 2 seconds long, you can use the <u>Default Start Tolerance</u> option in **Settings** menu ➔ **Fixed time** events.
>
> **NOTE:** The Fill event is in the format HH:MM:SS, i.e., it is seconds-accurate.

<u>Complex Items</u>

Opens the *Custom Complex Event* dialog. A list of all previously defined complex events is displayed for the sake of convenience. For more information, see section **Settings** menu ➔ **Complex items…**<u>further in this chapter</u>.

External Events

> **NOTE:** The *External Events* are part of the **AirBoxPRO** option. The following section contains a description of these events:
>
> **NOTE**: In order to insert events into the playlist, you have to enable the related plug-ins in the **Settings ➔ Modules ⇨ Remote Control** section first.

<u>Acappella Switcher Output</u>
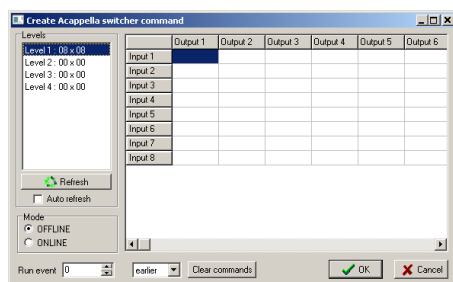
This is an event for controlling an external video/audio router. Depending on the switcher model, the currently available inputs/outputs will be displayed in a matrix grid, in which you can configure the command you need. Press **OK** when ready. You can also set an event offset via the *Run event* field, if necessary. The number you enter in the field is in milliseconds.

Concerto Switcher Output
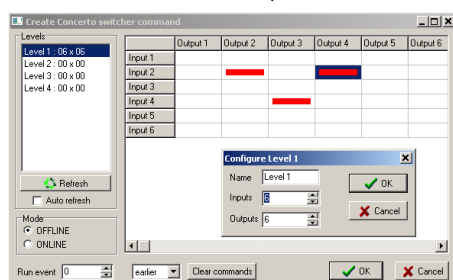
This is an event, which allows you to control Concerto series of routing switchers. When you select it, you will be able to see the dialog to the right. Double-click on the relevant *Level* to specify the number of inputs and outputs in it. After that, specify the commands in the grid (it will reflect your **Configure** settings). If you choose the **Command** mode, you will have to press **OK** to insert the event in the playlist. In **Online** mode, you can control the matrix switcher in real time. You can also set an event offset via the *Run event* field, if necessary. The number you enter in the field is in milliseconds.

DTMF output

This event inserts a line in the playlist to activate the *DTMF signal*. The following dialog box appears:

In the *DTMF Preset* field select a preset, as it is defined in the Modules -> Remote Control ->DTMF output section.

In the *Time Offset* field enter how many milliseconds *earlier* or *later* to execute this event

Elpro Switcher Output
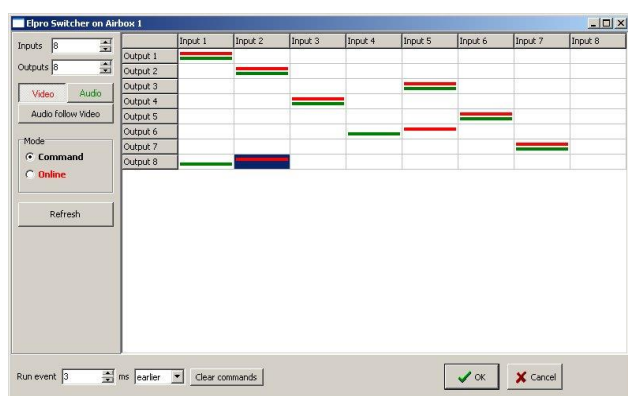
This feature activates the **Elpro Switcher output**. When you select it, the dialog below appears.

The first thing you have to do is to configure the number of inputs and outputs of the switcher from the fields in the upper left corner. Then you have to specify the desired input/output configuration. You can switch the **Video** and **Audio** separately or together. The user can work in two modes: **Command** and **Online**. In **Online** mode **Inputs** and **Outputs** can be switched in real time just by pressing the desired *In/Out* combination field in the grid. In **Command** mode you have to insert the desired *In/Out* configuration and add or insert it as an event in the playlist.

GPI Output

Inserts a line in the playlist, which activates the *GPI Output*. A dialog box appears to define the output **GPI** commands. You can have up to two commands on each COM port: GPI 1 means DTR tone (pulse on pin 4 and 5), and GPI 2 means RTS tone (pulse on pin 7 and 5).

You can select the *COM*-port from the available *COM*-ports, listed in the main window area. To configure them, go to **Settings ➔ Modules ⇨ Remote Control** tab, select the GPI Output, and press the **Configure** button.

The time for execution of a GPI event is defined via the surrounding items in the playlist. Therefore, you could correlate an event offset by using the end of the previous clip ("earlier" execution), or the beginning of the following one ("later" execution). Just fill in the number of frames in the *Run event* field and select [earlier] or [later] in the next field.

Kramer Switcher Output

Activates a *Kramer Switch Output*. A dialog box appears to define the Kramer Switcher commands:

Machine – select the number of the device, connected to the *COM port (RS-232)*. Up to 8 devices can be connected.

Input – select the number of the desired switcher input.

Run event – fill-in the number of frames for "earlier" or "later" command to the switcher (see the *GPI* section).

> **NOTE:** This switcher control plug-in works with the old control protocol. Most of the latest Kramer switchers are Protocol 2000-compliant, so you should use the "Kramer Matrix Switcher Control" module, even if your switcher is not a matrix switcher.
>
> **IMPORTANT:** The **PlayBox** modules do NOT support the Deck Control connector, supplied on the DeckLink breakout cable.

Kramer Matrix Switcher Output
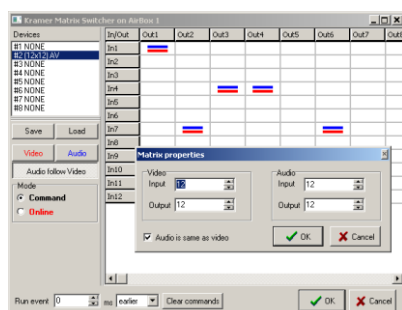
It is used for all Kramer switchers that use Protocol 2000.

This event activates the *Kramer Matrix Switcher Output*. A dialog box appears to define the Kramer Matrix Switcher commands. There are two main modes: *Command* and *Online*.

In *Online* mode the user can switch Inputs/Outputs in real time just by pressing the desired *In/Out* combination field in the grid (the right part of the window) area.

In *Command* mode the user selects the desired combination for the Matrix and after pressing *OK* the event is added to **AirBox** playlist.

The attached devices are automatically detected, but you can add more manually by double clicking a row in the list of devices (#1 NONE; #2 NONE; etc.).

Video and **Audio** can be switched separately or together. The Device list can be **Saved** and **Loaded**. You can also set time offset in milliseconds.

Leitch Matrix Switcher Output
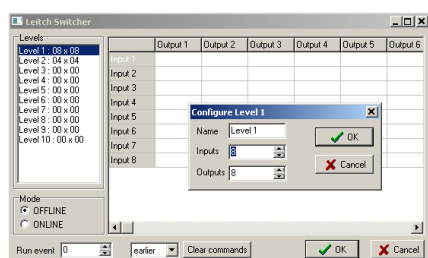
This is another optional plug-in for **AirBox**. When you choose to insert a *Leitch Matrix event*, the dialog to the right is displayed. Double-click on the relevant level to configure it (i.e. to specify the number of inputs and outputs in it) and click *OK*.

Then, specify the commands in the grid (it will reflect your *Configure* settings). If you select the *Command* mode, you will have to press OK to insert the event in the playlist. In *Online* mode you can control the matrix switcher in real time. You can also set an event offset, if necessary.

NetworkElectronics switcher
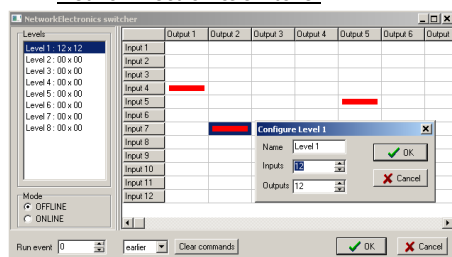
This is another optional external event for **AirBox**. It is controlled in the same way as the **Leitch Matrix** event.

Net Sender Output

The Net Sender Outputevent will send any text message via UDP. Thus, you could control IP-enabled third-party devices. All you have to do is specify the port, to which the message will be broadcast and type in the text. **AirBox** will send the message on the specified port when it reaches the event line in the playlist, observing the offset (if any).

NOTE: The maximum size per message is 512K.

Channel in a Box User Manual

Ocelot Switcher Output
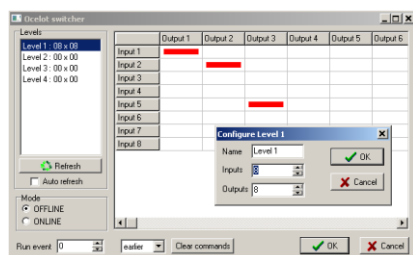
It activates the *Ocelot Switcher* output dialog, where you can define the switcher's commands.

You can schedule the device control by adding/inserting events into the **AirBox** playlist (the so-called *Offline* mode) or control it interactively (in *Online* mode).

The available matrix levels, which refer to digital video, analog audio and digital audio are listed in the *Levels* field. Right-clicking on a level invokes a pop-up menu that enables grouping it with another level, or ungrouping the currently selected pairs, or resetting all of them. Grouping is convenient for simultaneous switching of all levels in a group.

Double-clicking on a level will open a dialog, in which you can define its name, and the inputs' and outputs' numbers. Note that you cannot do this in *Online* mode. While in this mode the switcher deals only with the existing levels.

To define inputs and outputs of the selected level, mark and unmark the cross cells in the grid with a single mouse-click. Double-clicking on a column/row header invokes a dialog for changing its label. If you want to delete all current commands (at all levels), use the **Clear commands** button. Note that it works only in **Offline** mode.

If **Auto refresh** is checked, the grid will automatically reflect the matrix status every time any of the switcher's Input/Output buttons is pressed. If not checked, you can update this information manually by pressing the **Refresh** button.

When in **Offline** mode, you can specify the delay/precession of the event in milliseconds. The corresponding strings are situated in the lower left part of the window.

Quartz matrix switcher Output
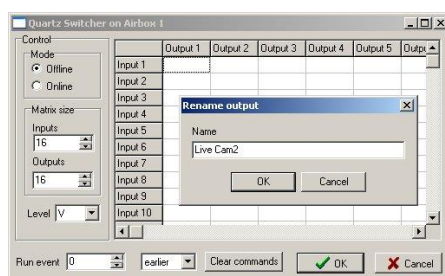
As in all other switcher events' properties, you can rename the Inputs and Outputs in the interface for easier navigation. For renaming, just double-click on the Input/Output and type the new name.

Channel in a Box User Manual

Relay GPI Switch output
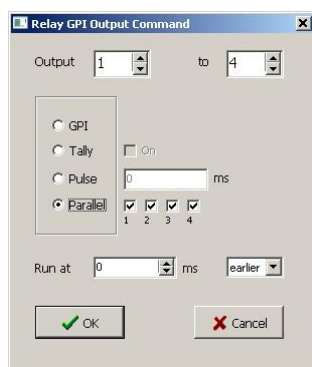
This event inserts a line in the playlist to activate the *GPI Relay Output*. The following dialog box appears:

In the *Output* field select the number of GPI output, on which the signal will be sent. Then, select what kind of a signal to send by marking the respective radio-button – **GPI, Tally, Pulse, Parallel.**

When you select **Tally,** the next *On* check-box becomes available . Check it in order to set **Tally On**.

If you select **Pulse,** a single pulse will be sent to the output. Enter the length of the pulse in milliseconds in the next field: .

If you select **Parallel,** you can send the GPI signal parallelly to several outputs. In this case, you have to select which these outputs are. In the *Output* field enter the number of the first output, and in the *To* field enter the number of the last output, which will be parallel: . Small check-boxes, which represent the outputs, will appear next to the **Parallel** radio-button. Now, by checking the respective check-box, you can select the outputs. .

In the *Run At* field enter how many milliseconds *earlier* or *later* to execute this event.

Sierra Video Switcher Output
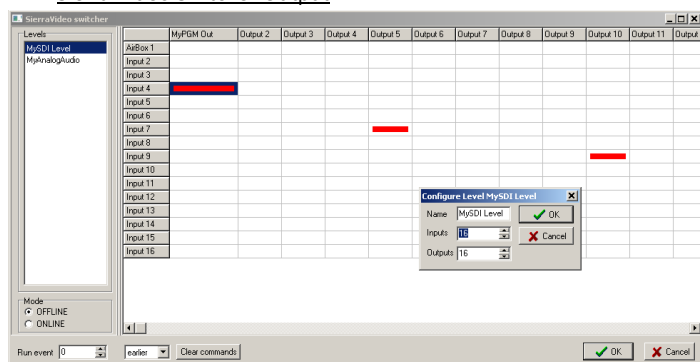
The Sierra Video Switcher Output event is another optional AirBox plugin. When you choose to insert such an event, the dialog to the right is displayed. Double-click on the relevant level to configure it (i.e., to specify the number of inputs and outputs in it) and click *OK*. Then, specify the commands in the grid (it will reflect your *Configure* settings). If you select the *Command* mode, you will have to press OK to insert the event in the playlist. In *Online* mode you can control the matrix switcher in real time. You can also set an event offset, if necessary.

Channel in a Box User Manual

Snell Switcher Output
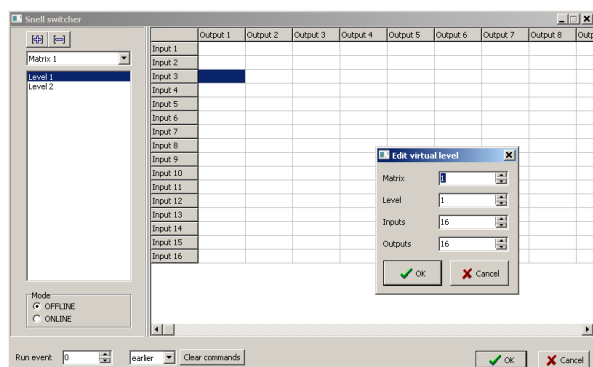
This optional plugin is used to control Snell devices that comply with the SW-P-08 protocol. It uses RS-485 cables. COM port settings may vary for different devices. The matrix is configured in the same way as the Sierra Video Switcher event's one above. The only difference is that here you can set more than one matrix, in case you have more than one device.

TitleBox NetControl Output
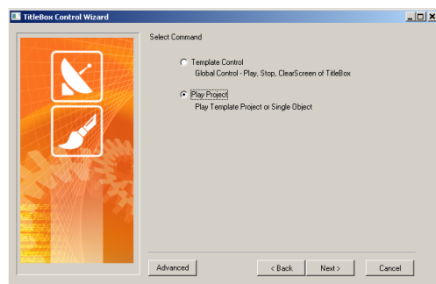
This is actually a set of events, which is used to manage objects in **TitleBox**. It could be a **TitleBox object,** which you would like to control from **AirBox**; or a **command** for controlling a currently open project in **TitleBox;** or a combination between them. For a complete description of **AirBox** with **TitleBox** integration, see *Appendix* 3

The **TitleBox** *Control Wizard* appears after selecting **Edit ➔ Add/Insert ➔ Event ➔ TitleBox NetCtrl Output…**. It will guide you through creating a **TitleBox Net Control Output** event. Follow the instructions and select the **TitleBox** project and/or object, and the desired action. You can evoke the Wizard also by right-clicking on a playlist row and selecting the relevant command from the drop-down menu (*Add/Insert Event ➔*).

If you press the **Advanced** button in the lower left corner of the *Wizard*, the *Configuration dialog box* will open. There you can define more sophisticated settings of the event (offset, duration, color, or even the text of the text objects).

Channel in a Box User Manual

Configuration dialog box:
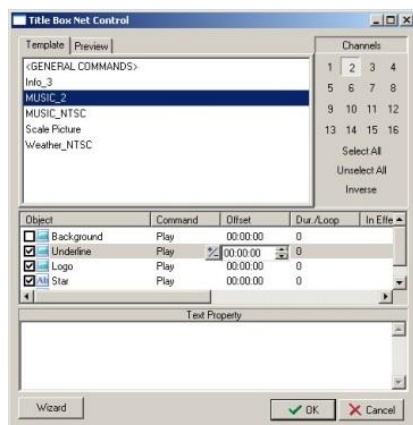
Template area: Here you can see the [General Commands] and a list of the available projects. Only the projects that have been previously saved as templates in **TitleBox** (**Network** menu ➔ *Export project as Template*) could be managed from **AirBox**.

> General Commands:

Start – starts the project in **TitleBox**.

Stop – stops ("freezes") the object(s) on the screen by ceasing the exchange of information with the graphics buffer.

Clear screen – clears the graphics frame buffer and hides all objects from the screen.

Reset – This command will stop and hide all previously displayed objects. This function would be useful if there have been other **TitleBox** events before the one you would like to insert and you do not know the display status of some object(s).

Slide show – this command triggers the Slide Controller in **TitleBox**. You can choose between **Play, Stop, Pause**, or **Next** in the drop-down list to the right. Use the spin-box next to it to specify which slide you would like to control.

> **NOTE:** The play and stop commands will be applied to the slide, whose number is specified in the spin-box!

For example: ⊙ **Slide show [Play] [2]** would mean start showing slides from slide 2.

⊙ **Slide show [Stop] [3]** would mean stop slide 3 and continue showing the other slides in the project.

If you need to stop the slide controller, please insert a Project **Stop** command.

When you select a command, it appears in the mid-window field. To activate a command, check the box in front of its name. After selecting the command, press the **OK** button. The **TitleBox** command event will appear in the **AirBox** playlist.

When you select a **TitleBox** project, its objects appear in the lower property window. You can select one (or all of them) to be controlled from **AirBox** by checking the boxes in front of the objects. Select the **Command** field (*Play, Stop* or *Update*). The [*Update*] command is used for text items that have been changed after their recent use in **AirBox**. In the fields to the right you can see the object's properties (duration, offset, effects, etc.). The *Offset* field allows you to set the time for earlier or later start of the object, according to the next video clip. After selecting the object(s), press the **OK** button. The **TitleBox** object event will appear in the **AirBox** playlist.

Channels

These correspond to the *TitleBox channel ID*, set in **TitleBox/Project ➔ Options ⇨ Network**.

Text Property area

You can type new texts of text objects in it.

IMPORTANT: In order to use **TitleBox Net Control output** event, you have to start your *TitleBox* with **Net Control** function (**TitleBox➔Network➔Net Control**), where available (*TitleBox Light Edition does not support this option*).

To control a project (its objects) from **AirBox**, you have to create that project in **TitleBox** first, and export it as a template into the *Template folder* (**TitleBox/ Network ➔ Export project as template**). The **Template folder** is created automatically during the **TitleBox** installation. If the **AirBox** module is installed on a different computer or you need to change the folder's location, then you have to start the **TitleBox Net Control Setup** (Programs ➔ PlayBox Technology Ltd. ➔ AirBox ➔ PLNetInst.exe) after the installation of **AirBox**.

Before pressing the **Finish** button in the TitleBox Net Control wizard, uncheck the ☐*Split Command* box. Thus, only one row will be inserted in the playlist.

The following paragraph contains an Example playlist, prepared with ☑ Split Command: If you want to start **Crawl1**, **Roll1,** and **Sequence** from the **"football"** project, stop all objects, and clear the screen after two clips, the playlist should look like this:

TV Content Rating

This event allows the user to show a TV rating identification, depending on the country specific regulations. Be aware that if you select to insert such an event, you will be allowed to add content ratings for the country, configured in **Settings ➔ Modules ⇨ Remote Control ➔ TV Content Rating Configurations**.
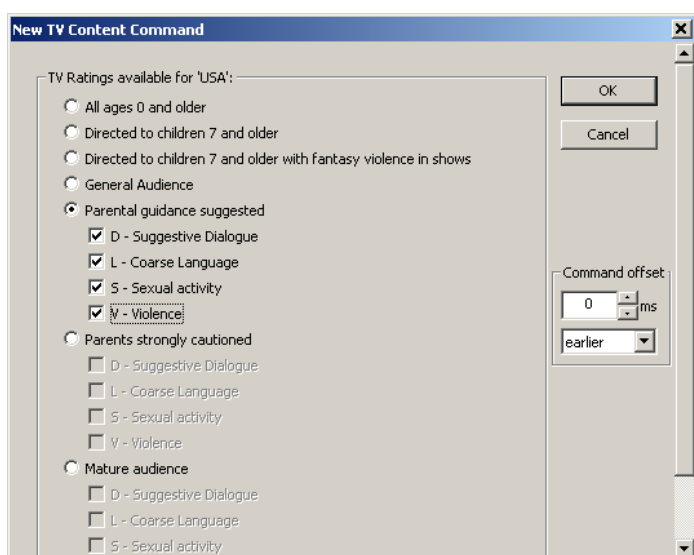
Notice the example to the left. If you have selected USA for a TV content rating country, your event settings will look like this. In addition to the country specific settings that you can choose from, you can add a command offset by entering a value in the *ms* spin-box and selecting **earlier** or **later** from the drop-down menu.

Universal Matrix Plugin

This event activates *Universal Matrix Controller (UMC)*. You can read more about the *UMC* in section <u>Universal Matrix Controller</u> further in this manual.

VideoHub Switcher Output
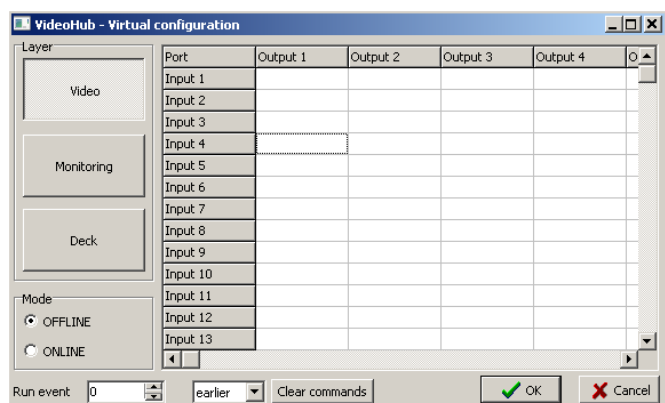
This event activates the **VideoHub Switcher**. Before adding/inserting such an event, you need to configure its settings from **Settings ➔ Modules ➩ Remote control ➔ *VideoHub Switcher Output***.

The **Configuration** dialog consists of three main Layers: **Video**, **Monitoring**, and **Deck**. Click on the respective buttons to set their individual matrices. The **Video** and the **Monitoring** layer are controlled by an *SDI Input/Output*. They share the same Input port but have different Output ports. The Deck layer, on the other hand, is controlled via serial port *RS-422*.

Other than that, the **VideoHub** matrix is operated in the same way as all other external event matrices.

VideoResizeCtrl Output
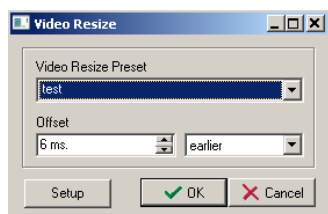
This option allows resizing the video output using pre-defined presets.

IMPORTANT: The Video Resize option is available only for **Multi-format playout** plug-ins in *AirBox*

The *Video Resize Presets* drop-down list contains all previously created presets in **Settings ➔ Modules ➩ Remote control ➔ *Video Resize Ctrl Output*** window. When you activate this function for the first time, this list will be empty.

Channel in a Box User Manual
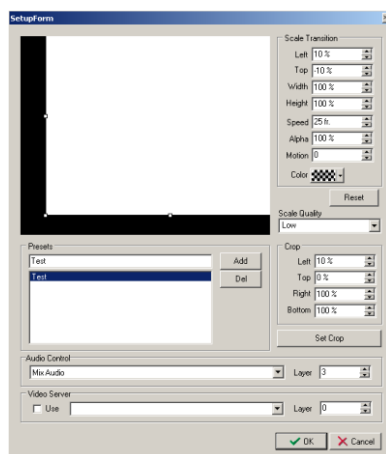
If you do not have any previously prepared presets, you can now create one from current window by pressing the **_Setup_** button. A set-up form will open:

The largest area in the **Setup Form** is designed for preset previewing. Use your mouse to squeeze and drag around the preset, or resize it using the transition spin-boxes to the right (see their description below). Write the name of the new preset in the *Preset* string and press **_Add_** to enter it in the list of available presets.

If you want to view the settings of a preset from the list, click on its name. You can change it and then press the **_Apply_** button to save the changes. If you want to remove an already existing preset, select it and press the **_Del_** button.

The **_Scale Transition_** field to the right contains tools for precise positioning and resizing of the video output. All values are preset-specific:

- **_Left_** – this percentage represents the offset from the left margin of the monitor. The video portions that remain unseen are still being rendered.
- **_Top_**–the offset percentage from the top of the monitor.
- **_Width_** – Shows the ratio between the scaled video size and the original one. Decreasing this value "shrinks" the video both to left and right.
- **_Height_**– Decreasing this percentage "shrinks" the video both to left and right.
- **_Speed_** – The time (in frames) for transition from the previous video layout to the current resize preset.
- **_Alpha_**– Use it to set transparency to the video (in percent).
- **_Motion_** – Select the transition effect for the current resize preset from the drop-down list. This is the way the video will move while resizing. The available options are: ascending effect (positive digits), descending (negative digits), and linear appearance (the zero value)
- **_Color_**– Set the color for the surrounding area of the scaled video. This part is usually used as a background for graphics insertion (i.e. in info channels).

Reset – use this button to go back to the original settings while creating/editing a preset. This button practically acts as an undo function for all unsaved changes in the **_Scale Transition_** field.

Scale quality – This is an obsolete selection box. Please, leave it to the default "Low" setting, as it is the one, which currently produces the best results.

> **NOTE:** Once you select it from the drop-down list, this setting will apply to all resize events.

Crop – You can crop the original video by percentage from *Left, Right, Top* and *Bottom*.

> **NOTE:** The crop will be executed right after you press the **_Set Crop_** button. This functionality is not preset-dependent!

Video Server

This option is used when you want to have a **Picture-in-Picture** functionality. The second picture comes from **AUX** source, as it is defined in the **Output** settings window.

Usually, the background picture is the output of *AirBox*, and the overcome picture is the resized picture, coming from the **AUX** source.

Check the ☑ *Use check-box* to allow this functionality and enter the PC name of the *AirBox* PC.

Select the *Layer* – screen layer of the resized picture.

NOTE: Screen layer should be 1 (the default value), if the resizing concerns the **AUX** source and it should be 16, if the resizing concerns the playlist.

IMPORTANT: While *Video Server* ☑ *Use* is checked and you want to resize both playlist and **AUX** source, you must first define an **AUX** source resizing event into the playlist, and then a playlist resizing event.

Audio control – this option is used in conjunction with the previous option.

If a *Video Server* is used, then you have to determine if the audio from **AUX** source (i.e. the resized picture) will be used.

There are three options - **Full** (100%); **Mute** (0%) or **Mix audio** with the audio from *AirBox* output.

Layer – the same layer number, as for the video server.

VikinX Matrix Switcher Output
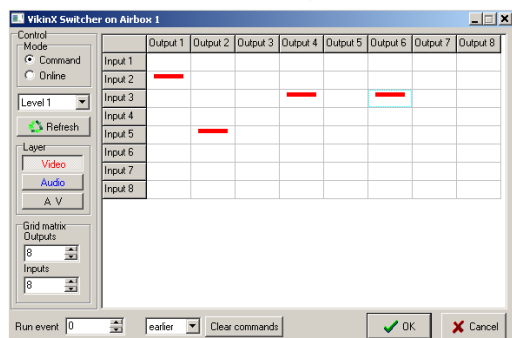
This external event controls *VikinX,* a *NetworkElectronics model switcher.* Similar to other matrix switchers, it has two operation modes: *Command* and *Online*.

While in *Command* mode, you can configure the combination of the matrix switcher and press *OK* to add the event in the playlist.

In *Online* mode, you can control the inputs/outputs in real time, just by pressing the relevant cell in the switcher grid.

You can specify the number of inputs/outputs in the grid by using the relevant buttons in the dialog. In its lower left part, specify the delay/precession of the event in milliseconds.

TIP (!) *Switcher Servers*

You can control one switcher from numerous *AirBox* machines by using the relevant *switcher server* add-on.

In the following example we will configure the Quartz matrix plug-in to send commands to the *Quartz Server* IP: Go to **Settings** menu ➔ **Modules** ⇨ *Remote* tab and select the Quartz Matrix Switcher output row.

Pressing the *Configure* button will invoke a setup dialog. You can specify the IP address in it.

Channel in a Box User Manual

The **switcher server** will receive all commands, coming from **AirBox** units in the local area network, and will control the switcher accordingly.

Currently, we can offer switcher servers for Kramer matrix switchers, VikinX matrix switchers, Elpro switchers, and Quartz matrix switchers. The setup principles are similar for all of them.

> **IMPORTANT:** When using a matrix server, you have to start it first, before running the **AirBox** channels that will send commands to it.

VTR Control plugin
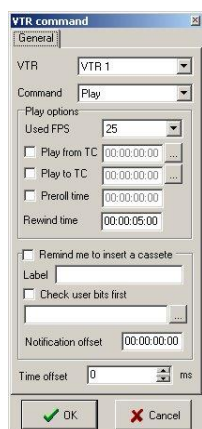
Although **AirBox** was designed as a file-based playout engine, now you can control up to 2 external VTRs through *RS422* for tape-based playout. All you have to do is insert a VTR control event in the playlist.

The VTR command dialog consists of two tabs – General and Advanced. In the **General** tab you have to specify to which VTR **AirBox** will send the command and the action to be executed upon its receipt (Play or Stop). The Play-related settings are situated below:

Select the temporal resolution of the tape in the *Used FPS* drop-down list (frames per second).

If you would like to play a tape from one point to another, check ☑ **Play from TC** and **Play to TC**, and specify the initial and the final timecode. In this case you will have to specify the maximum Rewind time (in milliseconds). Make sure not to insert another VTR control event for this particular VTR within the framework of this period (here -60000 msec.). Otherwise, a tape position conflict might occur and the earlier event will not be executed properly.

When you check the ☑ **Remind me to insert a cassette** box, you will be prompted to insert a cassette. Enter a note in the string to the right. Use the ☑ **Check user bits** checkbox to define the user bits of the cassette. You can also browse for them by pressing the ▢ button to ensure that the right cassette has been inserted. In the **Notification offset** string specify how much time before the event **AirBox** will remind for the cassette insertion.

Different VTR brands have different command reaction times. Find out how many milliseconds it takes your VTR to react to the command, sent from **AirBox** and enter this value in the *Time offset* spin-box. Once you do this, the command will be sent a little bit earlier, to guarantee seamless switching from **AirBox** to VTR playout.

> **(!) TIP:** You will need a switcher to change between the **AirBox** and the VTR playout. Therefore, a Switcher Event should always be present in the playlist before or after the VTR control event, unless you decide to switch manually. You could use the Complex events feature to create presets for later use (see the Complex Items section further in this manual).

> **NOTE:** Like all other External events, the VTR plug-in must be enabled first in the **Settings ➜ Modules ⇨ Remote Control** tab.
>
> **IMPORTANT:** Since the VTR control is executed through the PC COM port, you will need an RS232 to RS422 converter, such as Addenda or Leitch.

Channel in a Box User Manual

> **IMPORTANT:** The PlayBox modules DO NOT support the Deck Control connector, supplied on the DeckLink breakout cable!

Status Bar

The status bar is located at the lowest part of the **AirBox** window:

| Total duration: (1)09:55:00:20 | End at: (1)21:27:59:20 | Selection: 04:10:00:22 | Black Magic (1) Mixed Playback (PAL) |

The *Total Duration* field shows the total playlist duration.

The *End At* field shows the time, when the playlist will end (if not looped, of course).

If there is a **question mark (?)** after *Total Duration* and *End At*, this means that there are missing files in the playlist, and their duration is included in the total duration (i.e., the actual total length of the playlist is less than and the end-time is earlier than displayed). You can choose whether the duration of missing files should be included or excluded from the total length by checking the relevant box in **Settings ➔ General ⇨ *Missing Files* → ☑ *Show duration of missing files*.**

> **IMPORTANT:** In case there is a question mark in the *Total Length* and *End At* fields, the displayed total duration and end-time are NOT the real ones, since the missing files will be skipped during the playback.

If there is an **exclamation mark (!)** after *Total Duration* and *End At,* this means that there are missing files in the playlist, and their duration is not calculated in the total duration (i.e., the duration of the missing files is considered to be zero in the playlist).

> **TIP (!)** You can set **AirBox** to notify for the existence of missing files in the playlist by blinking the *Total Duration* and *End at* cells in red. Besides, an audio alarm can be set off through the PC sound card. See **Settings** menu description ➔ **General** section for details on how to enable these notifications for Missing files.

The *Selection* field displays the cumulated duration of the currently selected items in the playlist.

The fourth field of the **Status Bar** shows the hardware module that is currently used for playout. If it displays [None], please, check whether the appropriate module is selected in **Settings** menu ➔ **Modules** ⇨ **Output** tab.

Playback Control and Counters

00:00:09:20
DV25_001

- The **Master counter** shows the elapsed time of the current playout session. Pressing the **Stop** button resets the counter. The title of the currently playing clip is displayed below the running time code. This is quite useful in case the row played is out of the visible grid range.

> **(!) TIP:** To view the currently playing clip in the beginning of the grid, just click on the **Master counter**. This will shift the playlist items by placing the currently playing clip at the top of the list. If you want to view the clips before it, use the arrow keys on your keyboard.

00:00:00:20

- The **Clip counter**s situated under the master counter. It can operate in two modes, count-up and count-down mode. Just double-click on it to switch between them!

The playback control buttons are situated below the Clip counter:

Channel in a Box User Manual

- **Play** ▷ – starts the playback.

- **Stop** ▫ – terminates the playback.

- **Pause/Resume** ‖ – temporary interruption/resuming of the playback. The playback resumes from the same point on.

- **Next** ▷▏ – immediately stops the currently playing clip and switches to the next clip in the playlist. This happens with a smooth cut without any sound or visual artifacts.

- **Return** ↺ – this button is active only after a **Jump** has been executed. Use this button to return to the position before the last **Jump**.

- **Jump** ↻ – immediately terminates the currently playing clip and starts playing the one that is selected, regardless of its position in the grid.

- **Cue** ⚙ – interrupts the playback and pauses on the first frame of the selected clip. You can also activate it by holding down the **<Ctrl>** key and pressing the **Jump** button at the same time. Use this to prepare a playback clip while displaying its first frame on the output. Since this is a machine-and-decoder-dependent function, you may have to increase the *Cue Delay* time from the **Settings** dialog box **(Settings ➔ General ⇨ General → Cue delay**).

- **Automation** ⚙ – this button allows you to enable/disable the incoming external triggers (GPI input, DTMF tones). When Automation is OFF, the incoming commands will be ignored.

- **Reset hardware** ⚙ – resets the hardware driver. If, under any circumstances (bad *MPEG* media or bad system setup), playback is distorted so badly that normal **PLAY** and **STOP** commands cannot help, use this button to reset the hardware decoder.

> **NOTE:** This button is not visible by default! In order to show it, go to **Settings** menu ➔ **General** ⇨ **Interface** tab and remove the tick ☑*Hide reset hardware button*.
>
> **WARNING!** Pressing the ***Hardware reset*** button will interrupt the playback!

Playlist Control Buttons
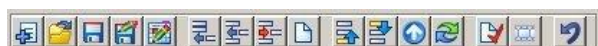
Most of these buttons actually duplicate the commands, found in the **File** menu and **Edit** menu. All buttons provide hints. Here is a description of the playlist control commands:

New playlist (open *ListBox*);

Open existing playlist;

Save playlist/**Save** playlist ***as…***

Save playlist ***as Daily***. This button will be greyed-out if you have not specified a [daily playlist folder](#) in the **Settings menu ➔ Start Up.**

Add clip at the end of the playlist/**Insert** a clip above the currently selected line.

Delete the currently selected clip from the playlist.

Clear playlist (remove all lines from the playlist).

Use these buttons to **Move** the currently selected clip(s) up (before the previous clip) or down (after the next clip) the playlist even during playback.

> **NOTE:** You cannot move a clip above the currently playing one.

Reset playlist position. If the current playlist has been started and then stopped, the clips that are already played are colored in grey and their order cannot be changed. Moreover, next time you hit the **PLAY** button, playback will start from the first black (not played yet) clip. If you hit **#**, the playlist will be reset, and the next time you hit **PLAY**, the playback will start from the very beginning of the playlist. Simply put, this is a kind of "playlist rewinding".

Randomize selection – press it to re-arrange the currently selected clips in a random manner.

This button will display the **Properties** of the currently selected line.

While a clip is selected in the playlist grid below, press this button to open the **Clip Trimmer**.

Undo up to 6 latest actions.

The Volume Slider

Probably the less noticeable part of the user interface is the *Volume slider*. It is situated between the multi-purpose zone and the playback control buttons. It is used to compensate for different sound volume in separate clips. You can control it with the [+] and [–] numeric keys on your keyboard, or just by dragging the slider. Check the box under it to mute the output sound.

This Volume Slider is connected to the slider, situated in the Volume control window. If you move the slider from here, it is moved in the Volume Control window too.

Clip Trimmer

The Clip Trimmer starts upon **AirBox** start-up. It was developed for two major purposes:

1. To enable working with the main **AirBox** interface while the Clip Trimmer is open.

2. To eliminate the possibility of interfering with the on-air playback process.

The Trimmer is described in details in the Clip Trimmer chapter further down this manual.

> **NOTE:** You can open only one Trimmer per instance!

MENU BAR
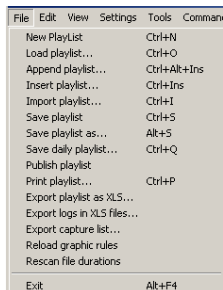
File Menu

The File menu contains all playlist file-related commands:

New Playlist

This command evokes the ListBox module – it opens a new blank playlist for editing.

Load playlist

Loads an existing playlist file (*.ply). You can also open a *Dalet Automation XML* playlists (*.xml).

> **WARNING!** Loading a playlist on the currently playing *AirBox* window will stop the playback. A warning dialog box will appear each time you try to do so.

Append playlist

This item appends another playlist to the end of the currently loaded playlist.

Insert playlist

This menu item inserts another playlist in the currently loaded playlist, above the currently selected row.

Import playlist

By using this feature you can import third-party playlists in *AirBox*.

Channel in a Box User Manual

Import a tab-delimited file
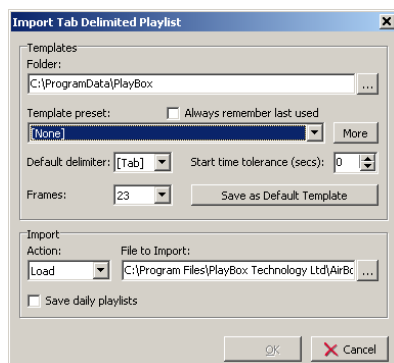
This menu item allows importing all kinds of tab-delimited text files. All you have to do is "tell" **AirBox** what the different columns of the imported file contain. The dialog to the right appears upon selecting this menu item. The **OK** button will not be active until you fill in all the strings.

The **Import Tab Delimited Playlist** operates by using playlist structure templates, previously created by you.

Use the browse button to the right of the *Folder* string to specify the location for saving your templates. You can check ☑*Always remember last used*.

The *Template preset* drop-down list contains all of the templates, saved in this folder. When you open it for the first time, there will be no templates to load and [None] will appear in the string. You will have to create them first (see next page).

The *Default delimiter* is used to select how the columns are separated in the file to be imported (tabs, semi-colons, etc.).

The *Start time tolerance* spin box is designated for usage when the playlist for import contains fixed start time entries. This means that if there are some conflicting start times that are out of the allowable tolerance, then no fixed start time events will be created during the playlist import. This value is valid both for earlier, and for later start tolerances. For example, if you enter 10 seconds here, this will be interpreted as ± 10 seconds.

The *Frames* drop-down list allows you to select the number of frames per second in the files to be imported.

In the *Action* drop-down list select whether to **Load**, **Insert**, or **Append** a playlist.

If you check the ☑ **Save daily playlists** box, the dialog will become extended, so that you can specify a folder and *Playlists start time*.

In the *File to Import* field select the file you wish to import via the browse button next to it.

You can save the imported playlist as a default template by pressing the **Save as Default Template** button.

Push the **More** button in the *Templates* field to create/modify your playlist template. The following dialog will open:

Channel in a Box User Manual
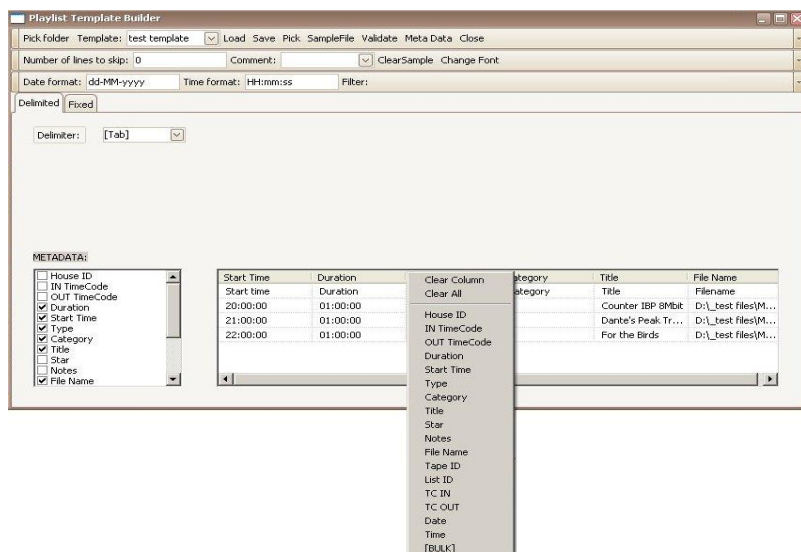
In the *Template* field, enter the template name. Use the **Sample File** button to select an example of the text file that will be imported with this template.

In the *Number of lines to be skipped* cell, enter the relevant figure (for example, -1). The skipped lines will be colored in red.

Comment– fill in a symbol to appear in the beginning of a row to be skipped from the playlist.

Go to the uppermost cell of each column, click once on it and "tell" the **Template Builder** what it contains (select from the drop-down list). Use [BULK] for the fields to be skipped (such as duration, etc.) to comply with the **AirBox** playlist structure.

When you are ready, press **Save**.

Pick– press it to use the current template for the file you are going to import.

If you already have some templates and you select one of them from the drop-down list, push the **Load** button to load it (its settings will be displayed in the grid).

Validation – press it, if you to ensure that your template is compatible with the file you are going to import (a browse dialog will open for you to print to that file).

If you need to use another sample file, push the **Clear Sample** button and then load the new sample. If you do not clear the old sample first, the new file will be appended to the end of the old one.

The third uppermost row of this dialog is dedicated to the daily playlists. There you can see the date and time format for such represented lists. In case you have to import playlists that are longer than 12 hours, use the **Filter** button to split them. Upon pressing it, the following dialog opens:

Here you can indicate some values of the clip, which will be used as a split point (for example, [House ID], [In] and [OutTC], [Category], [Title], etc.). Use the **Add** button to add these values as filtering conditions for splitting the playlists, and the **Apply** and **Remove** buttons respectively to confirm the usage of the created conditions, or to remove them.

44

Back in the first dialog, select a template preset from the drop-down list (now you should have at least one, built in the **Template builder**). Press the **browse** button to point to the *File to Import* and click **OK**.

You can set the currently selected *Templates folder* and *Template preset* as default, by pressing the **Save as Default Template** button.

Import playlists with fixed-width data fields
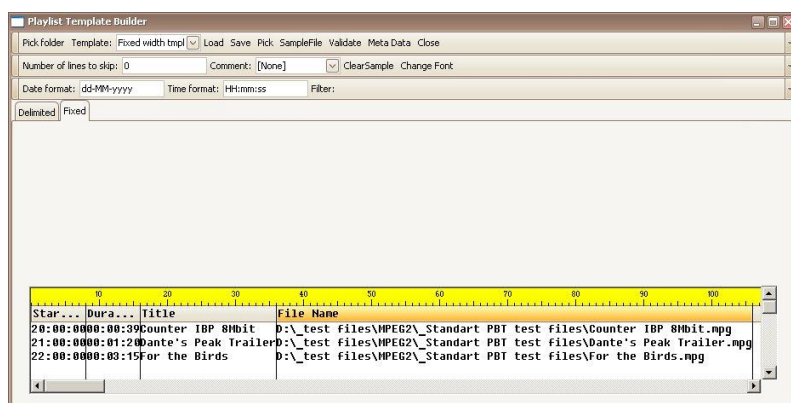
In addition to the existing schedule import engine for tab-delimited or *.csv* (Comma Separated Values) files, a new schedule type will be available for import – flat text with fixed-width data fields. For importing such playlists, go to **File Menu➔Import playlist…**. Then, click on the **More** button and select the **Fixed** tab, situated in the fourth uppermost horizontal bar to open the dialog below:

Here you can set/modify a template, which tells **AirBox** how to interpret the fixed width text file, containing the playlist. In the *Template* drop-down list, type the template name. Click on the **Sample File** button to select an example of the text file that will be imported with this template. This text file will be loaded in the data preview field, which is situated at the lower part of the **Playlist Template Builder** dialog. The actual columns in this sample text are not separated. Each of them contains a fixed number of symbols. To define the columns' width, you have to signify column breaks. For the sake of convenience, there is a symbol-related ruler.

- ▪ To create a break line, left-click in the desired position.
- ▪ To move a break line, click on it and drag it.
- ▪ To delete a break line, double-click on the line.
- ▪ To increase and decrease the width of the created grid, use the mouse scroll.
- ▪ While sliding the mouse within the data preview window, close to the pointer you can see an indicator of the symbols, passed from the start of the ruler. For example: 24 .
- ▪ After separating the columns, left-click within the bar under the ruler to specify what kind of data the relevant column contains (for example, **Title**, **Filename**, **Duration**, etc.). Click on the **Save** button to save the template, and on the **Pick** button to load it.

Save playlist

This command saves the current playlist file. The saved file contains the names of all clips arranged by their running numbers in the grid, as well as some additional attributes, like *playlist name, loop status*, etc.

Save as

This command saves the current playlist to a file. Except for our generic playlist format (*.ply*), now you can save your playlists to XML files as well.

NOTE: The XML playlist has a generic **PlayBox** format. Therefore, you cannot load any XML file as a playlist!

Save daily playlist
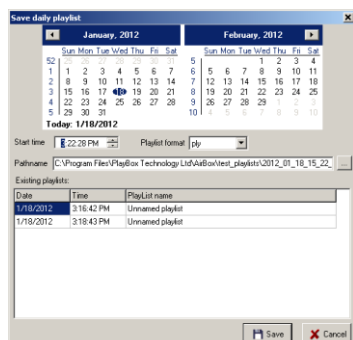
This menu item saves the current playlist in compliance with the [naming requirements](#) for daily playlists. This saving option will not be active if you have not specified a *Daily playlist folder* (see **Settings** menu ➔**Start Up…** ➔[Start Up playlist](#)).

> **WARNING!** When using fixed time events in daily playlists, the playlist should start and end within the same calendar day (i.e., the daily playlist should not go beyond midnight). Otherwise, we cannot guarantee the proper operation of the fixed time logics and the daily playlist logics.
>
> **WARNING!** We strongly DO NOT recommend using stop events and live streams without duration in daily playlists. The reason is that in case of accidental shut down, there is no way to calculate the duration of these items and thus, to determine the insertion point. So, *AirBox* will not be able to catch up with the schedule. You can still use them in daily playlists, but at your own risk.

Print playlist
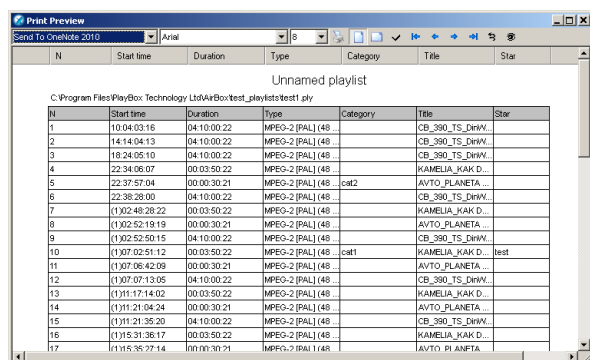
Both *AirBox* and *ListBox* provide the option of printing the current playlist. After choosing the **Print** command, a *Print-Preview* window opens and displays the current playlist**.** The columns and their order of appearance are the same, as in the grid.

In the *Print Preview* dialog, you can change the columns' size and adjust the following settings:

The drop-down list to the left contains all currently connected printers. Here you can select which printer to use.

To the right of it, you can select the *Font type* and *Font size* of the printed document

- ▪ ***Print*** – press this button to start printing the playlist.
- ▪ ***Page Orientation*** buttons– you can choose between Portrait and Landscape.
- ▪ ***Selection*** button– push it to print only the rows that have been previously selected in the grid.
- ▪ The *Name* of the printer (it may be a local or a network-connected printer).

- ▪ ***First page/Last page*** buttons– respectively go to the first and the last page of the content to be printed
- ▪ ***Go to previous page/Go to next page*** buttons – respectively visit the previous and the next page of the content to be printed
- ▪ ***Refresh Table*** button – press it to update the view.
- ▪ ***Auto-size Columns*** button – press it to automatically fit the columns' width to their content.

Export playlist as XLS

Use this option to export your playlists to comma-separated value (CSV) *.xls* files.

Export Logs in XLS files

If you need to convert the tab-delimited log files to XLS format, click on this menu item. A browse dialog will open for you to select which log to export – just select it and click **Open**. The resulting *.xls* file will be saved to the same directory.

Export capture list

If there are missing files in the playlist, invoking this command will create a list of missing clips for capturing in a *.cap* file. Exported *.cap* files are CaptureBox compatible.

Reload graphic Rules

The graphic rules are saved in a *.dat* file in the **AirBox** program directory. The graphic rules of the first **AirBox** instance (channel) are saved in the *GraphRules1.dat* file. The rules for the second **AirBox** instance are saved in a *GraphRules2.dat* file, and so on. This *.dat* file is loaded upon **AirBox** startup. If you want to change the graphic rules without stopping the program, overwrite the relevant *Graphrules#.dat* file first. Then, go to **File** menu ➔**Reload graphic rules.**

Rescan file duration

Use this command to rescan the durations of all files in the playlist.

Edit Menu

This menu contains all playlist- and clip-editing commands, available in *AirBox*. You can find the most important of them also in the drop-down menu that appears upon right-clicking in the playlist grid.

Add/Insert
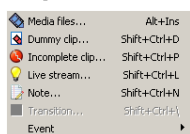
Both commands provide the same action list. The difference is in the insertion point in the playlist. While an **Add** command will add the selected event at the end of the playlist, the **Insert** command will insert the selected event above the currently selected line.

Media Files

Adds/Inserts existing clip(s) in the playlist. Browse in the dialog opened to point which clip(s) you would like to append or insert.

Dummy Clip
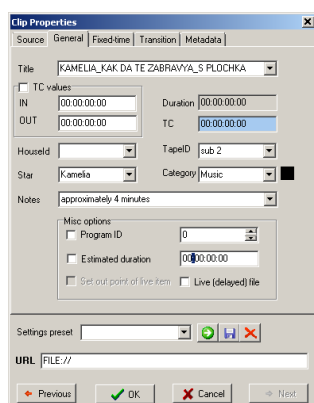
The **Dummy Clip** invokes a special dialog box for adding or inserting "virtual" clips in the playlist. You can use this option when you want to build your playlist prior to receiving a clip, whose properties are already known. *AirBox* will mark it as *Missing*, but when it becomes available, it could be used right away.

Use the **Source** tab to fill in the Location of the clip. You can read more about it in the ***Clip Properties* dialog ➔ *Source*** description.

In the **General** tab, fill-in the *Title* of the clip. The *IN* and *OUT* points, the *Duration*, and the *Timecode* are also editable. If you want you can write the *Category*, *Star,* and *Notes*. You can make the module remember each last string entry by checking the box next to its row. If you want, you can select the *File type,* or leave it as *Auto*. Read more about it in the ***Clip Properties* dialog ➔ *General*** description.

> **NOTE:** If you do NOT know the *Duration* of the clip you would like to add as a **Dummy Clip**, you can put a random number in the Duration field. When the clip is present, AirBox will automatically update its duration, as long as the following conditions are satisfied:

1. The ☑*Auto update clip's duration* option is checked in **Settings** menu ➔ **General** ⇨ **General** tab.
2. The following phrase is added in [HKEY_CURRENT_USER\Software\Digital Media Technologies Ltd.\PlayBox\2.0\AirBox 1]:"OverwriteOutpointOnAutoUpdate"=dword:00000001
3. Steps 1 and 2 are done for each **AirBox** instance, for which you would like to use that option.

If one or more of the above conditions are not fulfilled and the *Duration*, set for the **Dummy Clip** does not correspond to the actual duration of the clip, the clip will be trimmed accordingly.

Use the **Fixed-time** tab if you would like the **Dummy Clip** to start at a specific point in time. Check the ***Clip Properties* dialog ➔ *Fixed-time*** description for details on fixed-time settings.

Use the **Transition** tab to define transition settings for your dummy clip, if you wish. You can read more about it in the ***Clip Properties*dialog ➔ *Transition*** description.

Add some more info about the file in the **Metadata** tab. You can use this info later, in the graphic rules functionality. Check the ***Clip Properties* dialog ➔ *Metadata*** description for details about Metadata description.

Incomplete Clip

Just as the **Dummy Clip**, this menu item opens the Clip Properties dialog.

Usually, the incomplete clip is still being captured at the moment of playback. You need to create such an *Incomplete clip* entry in the playlist and check ☑*Live (delayed) file*. At a given time you start capturing this clip to the specified location, and once there are a few seconds of it, available on the storage, **AirBox** marks it as a valid clip and it can be played immediately. Thus, capturing and playback proceed simultaneously. If you want to predefine the file length despite its real length, just check ☑*Set out point of live item*. With this setting the file will be played exactly until the time you have specified, even if it is longer.

> **NOTE:** The TDIR option is hardware-dependent. See also Using the TDIR option in the Capture Mode section below.

Live stream

A live event could be either a UDP stream, or a Live video stream, as selected in the *Type* field of the **Source** tab. The live video stream comes from a video source, defined in **Live Input** from **Settings** menu ➔ **Output** ⇨ **Live Inputs** tab.

> **IMPORTANT**: The name is case sensitive.

If you are going to use some transition in the beginning of the event, select the **Transition** *Type* and **Transition** *Duration* in the relevant fields in the **Transition** tab.

The *URL* field (the command **forAirBox**) is generated automatically as you type in the above fields (for **UDP** or for **Live** stream). You could also enter a command directly in the *URL* field.

Channel in a Box User Manual

If you use streams (**UDP or Live**) that come from the same sources, you could create some presets via the *Preset* field at the bottom of the dialog window. Thus, the next time you need the stream, you can just insert the relevant preset instead of setting up the whole information again. Read more about the preset option in the ***Clip properties*** section above.

You can also configure the ***Fixed-time***, ***Transition***, and ***Metadata*** settings of the above events. For detailed information about these options, please read the relevant sections above.

PlayBox Live Inputs View is a tool, which shows a preview of the defined streams in **AirBox**. First you have to define streams in **AirBox**, then, after starting **LiveViewer**, you can select your desired streams.

Bookmark
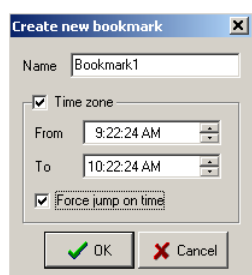
This is a special virtual item that can be used as a target point for incoming triggers (GPI input or DTMF input) or by itself. It will cause **AirBox** to **Jump** to the following item in the playlist, according to several additional conditions as described below.

The bookmark properties dialog contains a *Name* string and a *Time zone* area. When inserting bookmarks, you must enter a *Name*, and you can optionally enter a *Time zone* as well, depending on your needs.

The time zone represents a validity period of the relevant bookmark. It can act as a target for GPI or DTMF input (when [Jump to bookmark in time range] is selected as a command in the relevant setup dialog).

When ☑*Force jump on time* is checked, the bookmark will act as a fixed time event and **AirBox** will jump to it when its time is through. This can happen in two cases: if the expected trigger does not arrive until the time specified in the *To* string, or when you are using bookmarks alone in the playlist.

> For example, a jump operation will be executed to the bookmark in the screenshot at 16:36:32 o'clock, if the expected trigger has not arrived by that time.

> **WARNING!** Please, note that the *To* string will force **AirBox** to jump and execute the following items in the playlist. This might interrupt the playback order, so be careful when checking ☑*Force jump on time* in bookmarks!

There are several possible scenarios when using bookmarks and **DTMF/GPI** triggers combined:

When the **bookmark Name** is used as a reference, i.e. the DTMF/GPI command is Jump to bookmark:

- If there is a Time range, specified in the bookmark's properties and the trigger comes within this time range – **AirBox** will jump to this bookmark.

- If there is a Time range, specified in the bookmark's properties and the trigger comes outside this time range – **AirBox** will ignore the command.

- If there is no Time range, specified in the bookmark's properties, **AirBox** will jump to that bookmark unconditionally upon receiving the trigger tone/pulse.

When the **Time zone** is used as a reference, i.e. the DTMF/GPI command is **Jump to bookmark in time range**:

- If the trigger comes within the time range – **AirBox** will jump to this bookmark

- If the trigger comes outside the time range – ***AirBox*** will jump to the nearest following bookmark, according to the system clock.

WARNING! Please, make sure to set the *Time zones* correctly, since late arrival of the DTMF tone will cause ***AirBox*** to jump to the next bookmark, thus possibly skipping some of the items that are situated between the two bookmarks, as in the following example:

If a bookmark's *Time zone* is set From 15:36:32 To 16:36:32 and the DTMF tone does not arrive, ***AirBox*** could jump to the following item in the playlist at 16:36:32. If the DTMF tone arrives at 17:00:00, ***AirBox*** will jump to the next bookmark in the playlist that has a later *Time zone*, thus skipping some of the clips that are between the two bookmarks.

WARNING! DO NOT use fixed time events within a bookmark's time range!

Note

Adds/Inserts a comment line in the playlist.

Transition

This feature **Adds/Inserts** a transition into the playlist. You can read a more detailed description of ***AirBox*** events in Clip Properties ⇨ Transition.

Event

This feature **Adds/Inserts** events into the playlist. You can read more detailed descriptions of ***AirBox*** events in the section AirBox Events.

Delete Commands

Delete current from list

This item removes the currently selected clip from the playlist.

Delete unsupported clips

This command will remove all unsupported clips from the current playlist. It is used mainly when you import online media library files in your playlist. The OML files might contain *.avi* and *.wav* files. Both types are inserted in the playlist (using the Import tab delimited functionality from the **File** menu). ***AirBox*** will consider the *.wav* files as an unsupported format, so use the **Delete unsupported clips** command to cleanup your playlist.

Clear List

This command removes all clips from the playlist.

Clip Trimmer

Invokes the Clip Trimmer. It allows previewing, non-destructive trimming, and exporting parts of the currently selected clip to new files.

Properties

Clip's properties are described in details in the Clip Properties section.

Other Edit Commands

Undo

Use it to cancel up to five recent actions.

Cut

Removes the selected object(s) and keeps it (them) in the buffer-memory.

Copy

Saves a copy of the selected object(s).

Paste

Pastes the buffer content.

Randomize

This command randomizes the order of all selected clips (highlighted in the playlist). It is quite useful for music video channels.

Apply Transition

This command is used to apply a default transition to a select item (items). You can set a transition to be a default one from the **Clip Properties ⇨ Transition** tab.

Playlist loop

This command duplicates the checkbox that loops the playlist for continuous playback.

Clip loop

This button will loop the currently selected clip when *AirBox* reaches it. This functionality is not the same as the Loop checkbox in the main *AirBox* window (see the Playlist and Clip Buttons section above).

Select all

Selects the whole playlist.

Deselect

Deselects the whole playlist.

Invert selection

Inverts the selection in the playlist.

Find/Replace

This is a searching/replacing function of the playlist.

View Menu
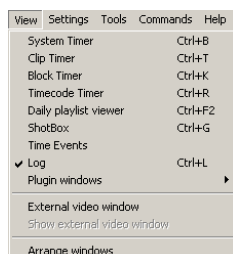
System Timer

This is a system time clock. It could be shown as a digital or analog clock. Right-click on it and select the desired appearance from the context menu. You can also set whether the date to be visible, or not.

Clip Timer

This is a clip-related timer. Its accuracy depends on the *MPEG2* decoder used. Right-clicking over it invokes a context menu, in which you can choose the timer mode (time elapsed/remaining). If you check the [✓Count down] item, the timer will show exactly how much time remains until the end of the currently playing clip. The background is black and a blue progress bar indicates what portion of the clip has already passed. If you uncheck the [✓Count down] row, the **Clip timer** will start counting up, showing the elapsed time from the beginning of the clip. The colors of the progress bar and the digits will reverse.

> **(!) TIP:** The color of the progress bar and that of the digits can be changed in **Settings➔Colors⇨Timers** by clicking in the relevant fields there.

Block Timer
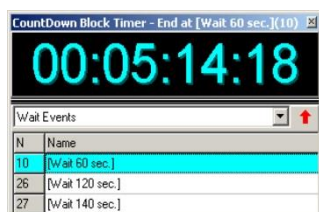
This is an *event-related countdown* timer. It shows how much time remains until the selected playlist event or until the [End of the playlist]. Choose the event type to display from the drop-down list: [Stop], [Stop Cue], [Wait], [WaitUntil], etc. If there are several uniform events in the playlist, they are shown in a pop-up list, arranged by their playlist line numbers so you can choose one of them. If you do not select an *event*, by default, the timer will show the time remaining till the end of the playlist.

A progress bar indicates how much of the time has already passed. The color of the progress bar and that of the numbers can be set from **Settings ➔ Colors ⇨ Timers.**

Channel in a Box User Manual

Timecode Timer

The timecode timer displays the timecode received from the LTC Reader plug-in, if available. The incoming LTC should be connected to the sound card of the playout server.

Daily playlist viewer

The Daily Viewer is a tool that will help you manage your daily playlists. In the area above you can view all daily playlists. The number in the brackets in front of the start time represents the number of days remaining till the start. For example, if there is a two in the brackets and then 12:00:00, this means that the playlist will start in two days, twelve hours.

When you click once on a daily playlist row, its items will be displayed in the grid below. Double-clicking on a daily playlist line will load it immediately. Please, note that this will interrupt the playback!

ShotBox

Selecting this **View** menu item will show/hide an additional window with buttons for fast switching/insertion of custom events.

All clipboard buttons can operate in two modes:

Execute mode ☆ – pressing a button while in this mode will execute the event, assigned to it immediately. In case a live stream is assigned to this button, a live stream row will be inserted in the playlist, and the playback will jump to it.

> **NOTE:** No offsets will be applied to the events while in this mode!
>
> **NOTE:** In execute mode, a complex item is treated as an events container. All events assigned to it will be executed immediately, with no offsets.

Insert mode ↩ – pressing the relevant button will insert the event assigned to it in the playlist. You can specify the point of insertion by using the ***location toggle*** button:

Insert as next → – will insert the event after the currently playing clip. If ***AirBox*** is in <u>Stop</u> or <u>Pause</u> mode, the event line will be inserted as the next line to be executed after resuming the playback.

Append at playlist end ↓ – will add the selected command to the end of the playlist.

Channel in a Box User Manual

To change the buttons' number and appearance, press the gear-wheel button . The buttons setup dialog will pop up. Define the dimensions at your will.

Right-clicking on a button in the *ShotBox* invokes a context menu, which allows you to do the following:

Assign event:

Logo event - view a detailed description of the Logo event here.

Media - view a detailed description of the insertion of media files here.

Complex event - view a detailed description of the insertion of Complex items here.

Playlist - invokes a browser dialog, through which you can assign a playlist file to be loaded upon pressing the respective button.

Jump shortcut - invokes a dialog, through which you can assign a jump command to a specified item, designated by its index number in the playlist, or a jump command to an existing bookmark in the playlist, designated by its name.

Unassign - Use this command to disable an already assigned event to the respective button.

Edit - Use this command to edit an already assigned event to the respective button.

Set title - Use this command to set a title to appear on the respective button.

Set hint - Use this command to set a hint to appear when the pointer is positioned on the respective button.

Picture:

Assign picture - this menu option opens a browser dialog, from which you can select a picture to be loaded as a logo upon pressing the respective button.

Clear picture - this command will remove the picture, selected in the assign picture option above.

Time Events
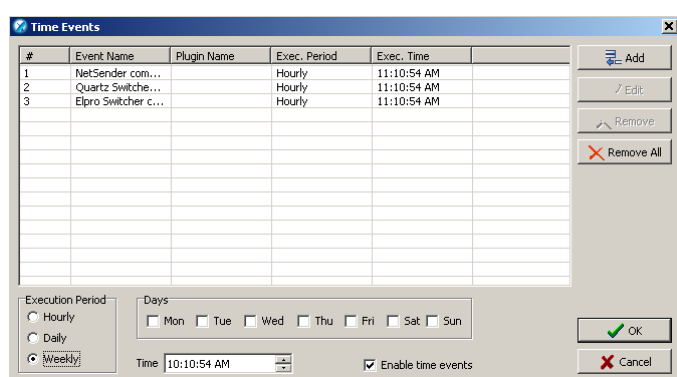
This is used for preparation of time schedules for events' execution.

In the *Time Events* window first you have to ☑ *Enable time events* by checking the relevant check-box at the bottom of the window.

Select the **Execution Period** – *Hourly*, *Daily* or *Weekly*, as well as *Time* for executing the event.

Press the *Add* button and a list of all available events will appear. Select the desired event and define it, as it is done in the **Add/Insert event** dialog.

In the example above, a *Kramer switcher* event will be executed each day at 15:00 h. and at 18:00 h. Also, a *Logo* preset 1 will be started each day at 19.00h.

If you want to edit an entry, select it and press the *Edit* button.

If you want to remove the entry, select it and press the *Remove* button.

Log

Selecting this item in the **View** menu will open a **Log View** window. It is used to show the events that have occurred during the *AirBox* operation.

(!) TIP: All windows can be freely resized and moved around the screen, staying on top of other windows.

<Ctrl>+click on a Timer window will change its appearance. Check it out!

Right-clicking on any timer window opens a drop-down menu. Select **Can Snap** in it to lock the timer position to the main window.

Channel in a Box User Manual

Plug-in windows

Select this item to display the additional windows, available with the currently used output module.

✓ Volume control window

VU&Peak Meter

Momentary Loudness

Short-Term Loudness

Loudness Meter

Volume Slider

Front Left

Front Right

Front Left

Front Right

Front Center

Back Left

LFE Subwoofer

Back Right
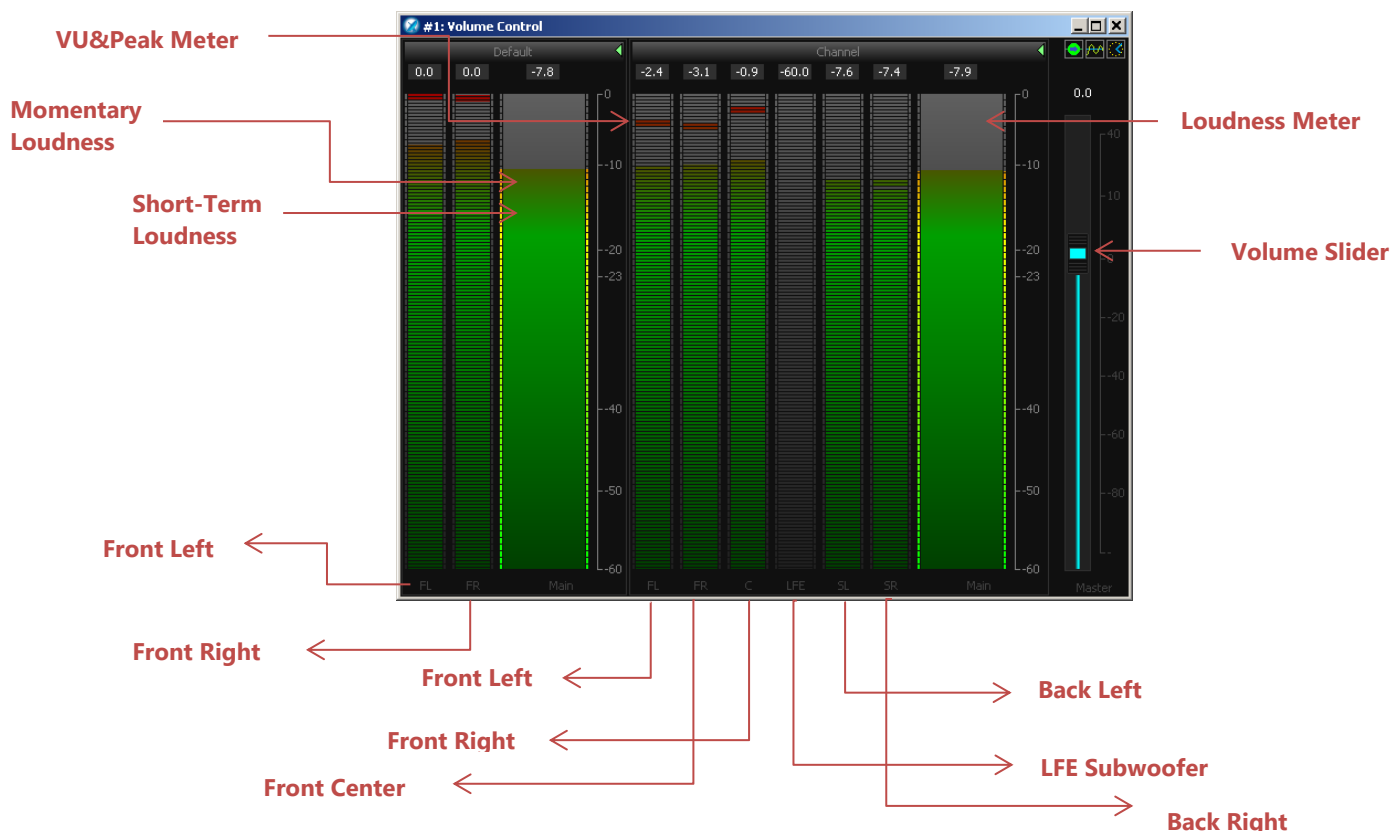
This is an additional window, which appears close to the main *AirBox* window. You can resize it by stretching its edges.

The image to the right shows the **Volume contro**l window of an *AirBox* that has two different audio configurations. The first one is for stereo, i.e., a left and a right audio channel – the [Default] section to the left. The second one is for 5.1 audio, i.e., 5 audio channels – the [Channel] section to the right. Notice that the [Default] section shows only two channels, as it represents a stereo output, while the [Channel] section shows six channels, as it is for 5.1 output.

The colored fields in the window are *volume and peak meters*. The narrower band shows the peak levels in each separate channel, and the wider one to the right shows the overall peak levels. The blue scroll to the right shows is the [Master] volume slider. This Volume slider also is visible in the main *AirBox* interface (between the multi-purpose zone and the playback control buttons).If you move the *slider* from the **Volume Control** window, it is moved in the main interface too.
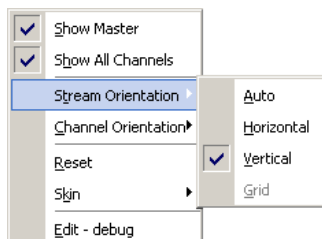
Right mouse-clicking on the *Volume Meter* area opens a context menu, like the one shown below

The **Show Master** and **Show All Channels** commands allow you to hide / unhide the master and the channel bars respectively.

In the **Stream** and **Channel Orientation** menus you can select the direction of the stream and the peak bars respectively.

Channel in a Box User Manual

The **Reset** command nullifies the peak values and the **Skin** option allows you to select a skin color for your peak bars.

The **Edit – debug** command invokes a dialog, which allows you to configure the overall appearance of your **Volume Control** window, named the **Designer** window. This window contains five tabs.
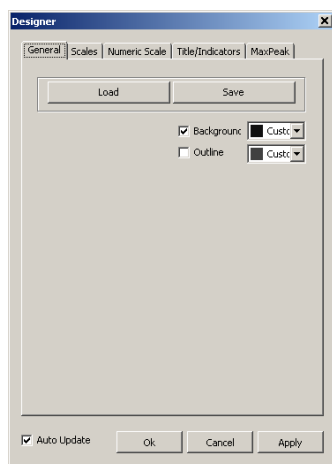
The **General** tab allows you to **Load** and **Save** custom skins for your *Volume Meter*. Here you can also set your *Background* and *Outline* colors.

Note that when you configure your desired settings, you need to press the **Apply** button in order to change the appearance of the **Volume Meter** window.

Furthermore, you can dynamically view how the appearance of the **Volume Meter** changes if you place a check in the *Auto Update* box.
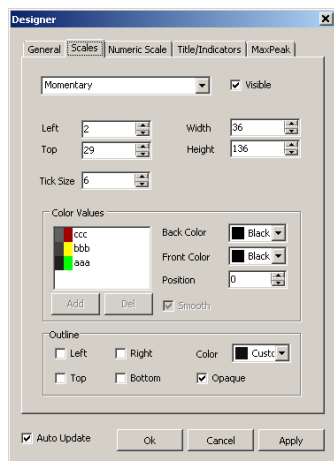
In the **Scales** tab you can show / hide the scales via the *Visible* box and set the scales type to Momentary or Short Term.

Use the *Left / Top / Width / Height / Tick Size* spin-boxes to edit the appearance of the bars.

The Color Values field allows you to edit the bar colors, depending on the peak levels. Finally, use the Outline field to set borders in your desired positions and / or colors.

Channel in a Box User Manual
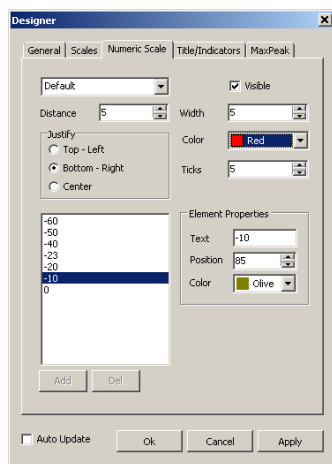
The **Numeric Scale** tab allows you to configure the appearance of the numeric scales in the **Volume Meter**. You can show / hide it via the *Visible* check-box, set its *Width*, *Distance*, *Position*, and *Color*. Furthermore, you can choose the number of Ticks (values) shown in the scale via the respective spin-box.

This tab also allows you to set the *Color*, *Position,* and *Text* of the number values.
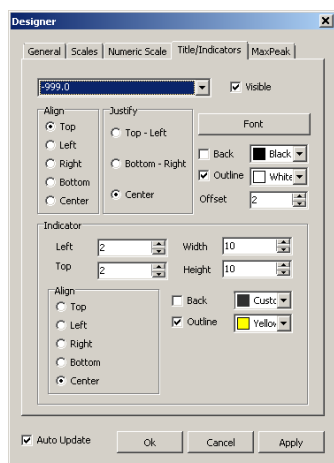
Use the **Title/Indicators** tab to configure the position and appearance of the *Titles* and *Indicators* in the **Volume Meter** window. You can set the *Alignment*, *Justification*, *Font*, and *Color* of the *Titles*. For the indicators you can change their position in the dialog, *Color*, and *Alignment*.
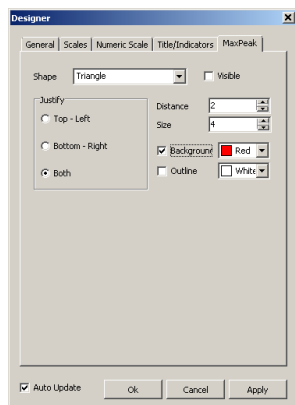
Finally, in the **MaxPeak** tab configure the *Shape*, position, and colors of the max peak indicator.

External video window

Check it to move the video window outside the main **AirBox** window. You can resize the video window by pulling its edges with the mouse pointer, or by right-clicking on it (a drop-down menu will suggest several zooming options).

Show external video window

If you have closed the external video window, you can view it again by ✔checking this line. Pressing it once again will hide the window back.

Arrange windows

Clicking on this line will align all open timers around the main **AirBox** window.

Settings Menu
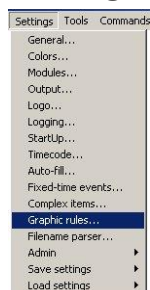
The **Settings menu** contains the most important settings for the **AirBox**.

General

This **Settings** dialog box is divided into several tabs.

<u>**General**</u>

General tab contains general functionality options:

☑*Use Clip Trimmer* – if checked, **AirBox** will open the **Clip Trimmer** upon double clicking over a clip. If this flag is not checked, the **Clip Properties** dialog will open instead.

Channel in a Box User Manual
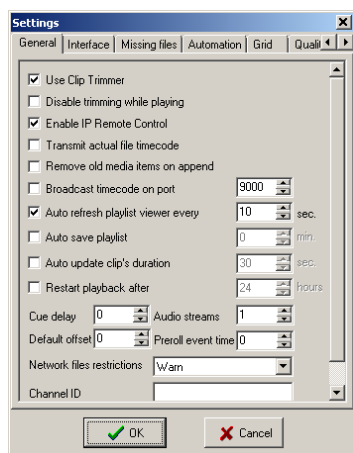
☑*Disable trimming while playing*– restricts changes in the currently playing list.

☑*Enable IP Remote Control* – enables or disables **AirBox LAN** remote control (using IP address and the Multi AirBox Manager).

☑*Transmit actual file timecode*– **AirBox** transmits the actual file timecode, embedded in the clip file. This timecode can be used by external applications, such as **SubTitle Plus** (www.subtitleplus.com).

☑*Remove old media items on append*– removes the already played files from the old playlist when appending a new one

☑*Broadcast timecode over network*– If this one is checked, **AirBox** will broadcast the playlist time code, the current clip timecode, and the relative timecode (from the beginning of the playout session) over the network (via **UDP**).

☑*Auto save playlist*– saves the playlist automatically at predefined intervals.

☑*Auto update duration of marked files* – check it to enable automatic updating of clips' duration in case of file change. If not checked, **AirBox** will "remember" the duration of the original file, and if you change this file with a longer one, it will be truncated. After you have checked this box, go to the playlist, and highlight the files you would like to update (you can use multi-selection). Right-click and choose **Auto update**. Thus, all "auto update" files will be checked in accordance to the set interval, and their durations will be updated accordingly.

☑*Restart playback after* – resets the hardware counter, which is displayed at the **Master** (green) **counter**. This option is necessary due to various platform and format limitations.

> **NOTE:** Decoder restarting may cause a few black frames in playback.

▪ *Cue delay* – this option concerns the **Cue** function (freeze at a first clip frame). It defines the delay (in milliseconds) before pausing on the first frame, since different decoders behave differently in regards to the first clip frame. This function is both decoder- and computer-dependent. You will have to find the most suitable one for your setup via testing.

> **NOTE:** This affects only some old Playout plug-ins.

▪ *Audio Streams* –depending on the hardware platform used, **AirBox** can playout more than one audio stream.
▪ *Default Offset* – this is a general offset (measured in milliseconds) that will apply to all newly-inserted external events. The positive values mean later execution of the events and the negative values mean earlier execution. This offset will appear in the **Notes** column in the playlist upon inserting an external event.

- *Pre-roll event time* – The command reaction times for external events is different, depending on the devices. Still, if you find out an acceptable compromise value that will serve your needs, you can set a general offset value to be applied in run-time to each external event you add in the playlist. Thus, you will not have to enter the same values each time you create an event. This offset will not appear in the playlist!

NOTE: Do not forget to account for the pre-roll event time when you set additional offsets in the events' properties dialogs. The offsets and the pre-roll event time are summed up in runtime.

- The next field, *Network Files Restrictions*, is designed to specify the procedure upon insertion of files that are not saved locally in the **AirBox** machine. This is a precaution option: The playback could stop upon network failure if your whole content is remote, or the playlist could end earlier – if some of your files are stored locally and the network fails, **AirBox** will label all remote files as missing, skip them, and hence, the end time of the playlist will be reduced by the duration of all missing files.

Type
MPEG1 (Remote file)
MPEG2 (Remote file)

When **Warn** is selected, the rows of all network-stored files in the playlist will become dark-green. A note [Remote file] will appear in the **Type** column.

When **Allow** is selected, you will not be notified whether your files are stored remotely or locally. Thus, you will take the chance to trust your network.

If **Reject** is selected, the rows of all network-stored files in the playlist will become red (brown when selected). A note *Rejected remote file [Remote file]* will appear in the **Type** column. These files are considered missing and will be skipped during playback. If you have left the *Show duration of missing files* flag unchecked, the playlist time will be recalculated accordingly (see the *Missing files* section below).

- *Channel ID* – enter the ID of the output Channel here
- *Default code page* – select the default text encoding

Interface
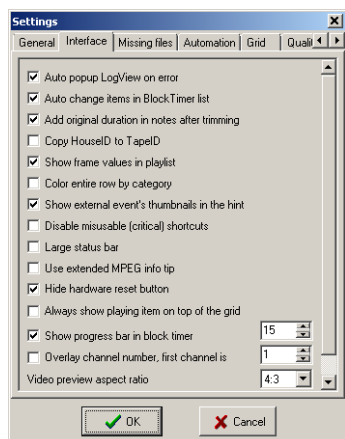
The **Interface** tab provides the following user interface options:

☑*Auto popup LogView on error* – displays the *LogView* window automatically, when an error occurs.

☑*Auto change items in Block Timer list* – the block timer will follow the subsequent events in the playlist (as selected in the block timer dialog box). If not checked, the timer will remain fixed at the selected event in the list. After this event has passed, the block timer will read negative values, since the origin point will remain fixed in the past.

☑*Add original duration in notes after trimming* – if checked, once you trim a clip, **AirBox** will automatically insert information about its original duration in the **Notes** column in the playlist. If you do not want to see this information, just leave this box unchecked.

☑*Copy HouseID to TapeID* – a convenient way to automate assigning reference information, needed for subtitles insertion.

☑*Show frame values in playlist* – check it to view the number of frames at the end of the duration value in the *Duration* column.

☑*Color entire row by category* –fills the entire row in the playlist with the *Category*-field color of the relevant clip. This option refers to categories in **DataBox**. You can change the category color in **DataBox ➜Options⇨*Default values*→***Colors to AirBox*

> **NOTE:** You will have to drag-n-drop entries from the ***DataBox*** grid into the ***AirBox*** grid to view their category color.

☑ *Show external event's thumbnails in the hint* – by default, you will be able to view thumbnails of all ***TitleBox*** templates used in the **TitleBox Net Control** events in the playlist. If you do not need them, just go to this check-box and uncheck it.

☑ *Disable misusable (critical) shortcuts* –replaces regular shortcuts of "dangerous" playback commands with alternative ones, as in the table to the left.

☑ *Large status bar* – enlarges the status bar at the bottom of the ***AirBox*** window

☑ *Use Extended MPEG info tip* – enables showing detailed file information in the hint when sliding the mouse pointer over a clip in the playlist

☑ *Hide hardware reset button* – checked by default, uncheck this box to be able to see the **Reset hardware** button under the counters in the main ***AirBox*** interface

☑ *Always show playing item on top of the grid* – scrolls the playlist grid so that the currently playing item is on top of it

☑ *Show progress bar in block timer … sec.* – defines the time before a clip's end for showing the progress bar inside the block timer (see **View ➜ Block Timer**). For instance, if a 15-second value is set, the progress bar will be visible in the last 15 seconds of the clip playout. During the rest of the time, the progress bar will not be displayed, just the counter itself

☑ *Overlay channel number, first channel is*– a big figure showing the ***AirBox*** instance will be displayed in the grid background. You have to set the number of the first channel. The numbers of all subsequently open channels will be updated automatically. You just have to check the *Overlay channel number* box in the **Settings** of each ***AirBox*** if you want to view it in the grid.

Video preview aspect ratio – select the aspect ratio for the ***Video preview*** window. This is necessary because some decoders cannot report the aspect ratio of the playing content, so this must be selected manually, in order to properly preview content.

Frame rate – this entry is necessary for estimating all kinds of time values (i.e., end time, playlist duration, etc.), while there is no accessible frame rate of a currently played file (for example, ***AirBox*** is in **Stop** mode).

<u>Missing files</u>
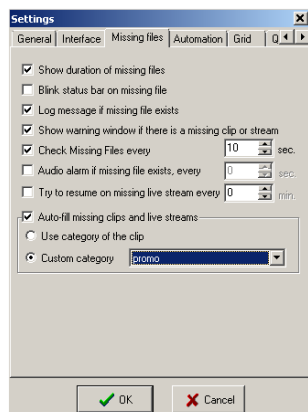
Missing files is the third tab in the **Settings** dialog box. It refers to automatic handling of missing files in the playlist.

☑ *Show duration of missing files* – if there are any missing files in the playlist, their real duration or zero-duration can be displayed. This will affect the playlist's *Total duration* and *End time* that are shown in the *Status Bar* at the bottom. If this option is checked, a question mark will appear after [Total Length] and [End at] in the *Status Bar*. If it is not checked but there are missing files, an exclamation mark will appear after [Total Length] and [End at] in the *Status Bar*.

☑ *Blink status bar on missing file* – in case there are missing files in the playlist, the *Total Length* and *End at:* cells in the status bar will become red and will start blinking.

☑ *Log message if missing file exists* – check it to include the list of missing files in the log window.

☑ *Show warning window if there is a missing clip or stream* – checking this box will ensure that a warning dialog will pop-up to prompt the operator that there is a missing clip/stream in the playlist.

☑ *Check Missing Files every …sec.* – When you prepare your playlist, you can add files to it that are not available in the online storage yet. They appear in **AirBox** as red-colored lines. Check this box to let **AirBox** auto-check whether the missing files are already available or not.

☑ *Audio alarm if missing file exists, every …sec.* - use this option to enable an audio alarm if there are missing files in the playlist. Thus, you will be alerted if there are missing files, or if you accidentally delete a clip, related to the current playlist.

> **NOTE:** The alarm is output on the PC speaker!

☑ *Try to resume on missing live stream every…min* **–** automatically checks if a missing live stream has been reconnected to **AirBox** in a period of time defined by the user

☑ *Auto-fill missing files and live streams* – **AirBox** will look for the auto-fill category, according to your preferences:

◉ **Use category of the clip** – **AirBox** will use the auto-fill category, as specified in the missing file's properties. In case there is no such category, the default auto-fill clip will be played out.

◉ **Custom category** – here you can select from your own categories, prepared in advance in **Settings** menu **➔Auto-fill** dialog.

Automation
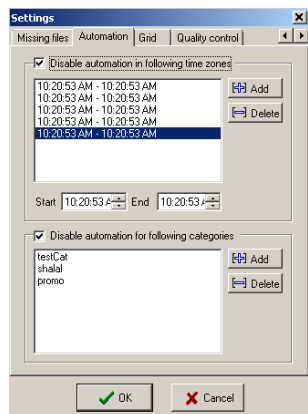
This tab allows you to manage the ***Automation*** settings that you have defined in your playlist.

☑ *Disable automation in following time zones* – here you can list the time zones to disable the execution of commands, related to external triggers. First, you have to specify the *Start* and the *End* time of a zone and then press the ***Add*** button to add it to the list. To remove a zone from the list, select it and press the ***Delete*** button above.

☑ *Disable automation for following categories* – here you can define categories, for which to disable the execution of commands, related to external triggers. The category names correspond to the *Categories,* set in **Clip Properties➔***General* tab for each playlist item.

> **WARNING!** The automation has a higher priority than the Automation graphic rules and the **Automation** event in the Edit menu! The latter will both not operate properly when using Automation, so please do not use them simultaneously!

Grid
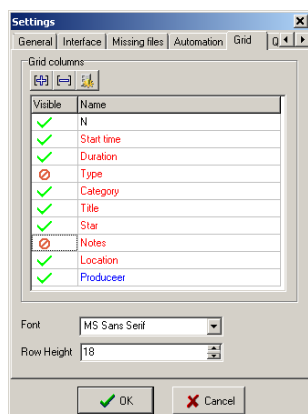
The ***Grid*** tab allows you to control the columns that appear in the playlist grid.

You can add new columns by pressing the ***Plus*** button above the list. Generally, a new column will contain a Metadata name (as it appears in the first column of the clip properties). Thus, once you enter some metadata for a clip, its value will appear in the corresponding column in the grid.

All columns are ✔**Visible** by default. Double-click in the green tick-mark to turn them ⊘ **Invisible**.

To remove a column from the list, click on the **Minus** ⬌ button.

Click 🔧 to restore the factory defaults.

Change the *Font* and the *Row Height* settings through the respective fields.

Quality control
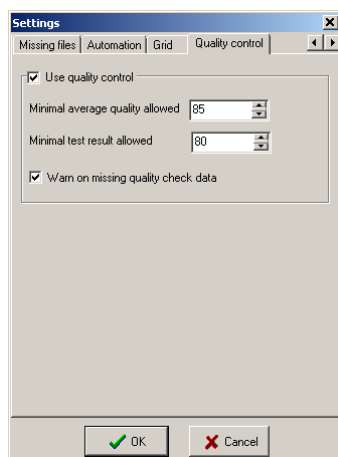
The **Quality control** tab accommodates settings, related to playlist media files, which have been passed through quality control tests. Such tests are performed by the **QCBox** tool. For more detailed information, related to **QCBox**, please, refer to the relevant chapter further in this manual.
In brief, the **QCBox** tool performs content verifications, such as loudness-related tests, checking for missing audio or video samples, etc.

The test results are defined in percentage from 0 to 100%, while 0% means that the media source has not passed the test successfully.
After that **QCBox** stores the results as metadata into a file with extension *.mtd*.
Such a file will be generated for each media file tested.

If the ☑*Use quality control* checkbox is ticked, after loading the playlist, **AirBox** will read the *.mtd* files, created by **QCBox**. According to the quality data, provided in the *.mtd* file, **AirBox** will play or skip a media file.

The *Minimal average quality allowed* spin-box is designed to indicate the average value of all quality tests performed. **For example**, if a media file is tested, using three testing plug-ins in **QCBox**, each separate plug-in will rate this file with a value from 0 -100%. If all the ratings are summed up as 250%, the average value will be equal to 250 ÷ 3 = 83 (whereas 3 is the number of plug-ins, used for tests performing).

> **NOTE:** If the *Minimal average quality allowed* of a tested media file is under the specified value in the **Settings** menu ➜**General**⇨**Quality control** tab, AirBox will not represent the file.

The *Minimal test result allowed* spin-box refers to the minimal value in percentage which is permissible per an individual test.

If you place a check in the ☑*Warn on missing quality check data,* **AirBox** will generate a warning message if there is no applicable Metadata available for a certain playlist item.

Channel in a Box User Manual

Access Restrictions

This General Settings tab allows you to add restrictions for non-administrator users, i.e., users who do NOT have the admin password, defined in the Admin section of the settings menu. The following restrictions can be added from here:

☑*Change settings* - if checked, this option restricts the user from changing any AirBox settings

☑*Change playlist* - if checked, this option restricts the user from changing the playlist

☑*Change playback status* - if checked, this option restricts the user from changing the playback status, i.e., selecting a play, stop, or a pause command

Colors

This option allows users to define the colors, used in the **AirBox** playlist for color- coding, as well as for timers.

Playlist colors:
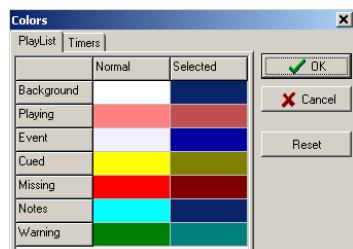
Background – playlist background.

Playing – currently played clip.

Event – event rows.

Cued – a clip prepared for playback.

Missing – missing file rows.

Note– note (comment) line.

You can define colors for *Normal* and for *Selected* status of each playlist row.

Timer colors:
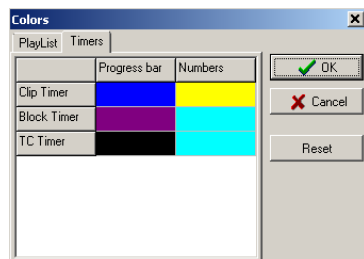
Clip timer – colors for the progress bar and the letters in the *Clip Timer*.

Block timer – colors for the progress bar and the letters in the *Block Timer*.

TC (timecode) timer – colors for the progress bar and letters in the *TC Timer*.

Clicking on the desired color box enables a combo-box for selecting a color or defining a custom one.

The *Reset* button resets the selected field to default colors.

All the screenshots in this User's Manual are made with default *AirBox* colors and all explanations are in accordance to them.

Modules

AirBox Output
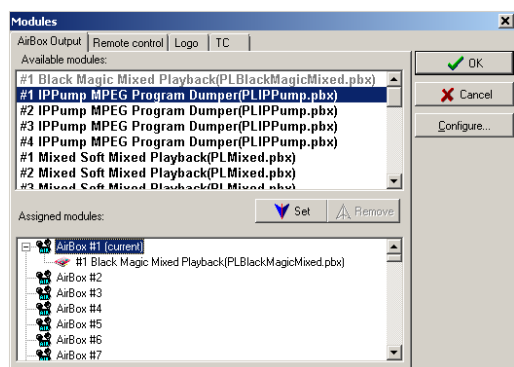
Here you can select the active playback module, depending on the installed hardware platform. Check the latest list of platforms, supported in *AirBox* in **https://playboxtechnology.com**, or write to info@playboxtechnology.com

Select the desired module from the list of available modules in the upper window, and assign it to the corresponding *AirBox* channel in the lower window by double-clicking on the module, or by pressing the *Set* button. This option allows you to start multiple *AirBox* applications on different hardware decoders at the same workstation.

NOTE: Depending on the active module, *AirBox* may recognize some files as invalid, if these files are not supported by the respective hardware used.

If there is no hardware platform installed, there are still several options:

IP Pump MPEG Program Dumper

This plug-in uses *MPEG2* source files and directly pumps the content without recompression to a network IP address, instead of a decoder. The IP Pump uses a standard UDP mechanism for data transfer. Please, see the settings menu description in **Settings ➔Output**menu.

IMPORTANT: There is no VGA Overlay preview in this mode. There is no graphics insertion in with this plug-in!

NOTE: The IP Pump is an optional plug-in and it is not a part of the standard *AirBox* functionality.

MixedSoft Playback

This is a software-based playout of files with different compressions and frame rates in the same playlist. Used mostly for demonstration purposes, it represents all features available to Hardware playback, except for the video signal output. If you have a dual-head video card with TV output, the video overlay preview can be monitored through it.

MPO Mixed Playback

Channel in a Box User Manual

This plug-in allows for simultaneous playout of different types of media (SD, HD, 25fps, 30 fps, etc.) and different outputs (SDI, Analogue, IP Streaming etc.) to run at the same time from a single AirBox playout server. Basically, with the Multi Parallel Outputs (MPO), AirBox allows broadcasters to provide parallel outputs in any combination needed to deliver content. For more detailed information on how to use the MPO Mixed Playback, please, check the relevant section <u>below</u>. If you still have any questions, related to the configuration of this **AirBox** module, contact our support team at support@playboxtechnology.com.

Mixed Stream Playback

This plug-in sends the playback stream to a network IP address or ASI output, instead of a video output. Its advantage, in comparison to the IP Pump plug-in, is that it provides the possibility of inserting graphics in the stream and there is a preview of the output video. Please, check the setup details, in the **Settings -> Output -> Streaming menu**.

> **NOTE:** The Streaming module is an option. It needs an additional license.

Remote Control

All of the available modules for remote control in **AirBox** are listed in the remote control tab, like **GPI** (General Purpose Interface), **DTMF**, video matrices, and video switchers, **TitleBox NetControl**, resizing, etc.

These modules correspond to events in the **AirBox** playlist. First, you have to enable the module from **AirBox Menu➔Settings➔Modules⇨Remote Control** and configure it. Then, you can put it as an event in the **AirBox** playlist.

Here is a list of available modules:

DTMF Input/Output plug-ins
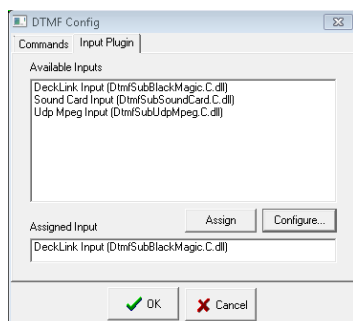
DTMF stands for **Dual-tone multi-frequency signaling. DTMF** tones are used in **AirBox** as triggering signals for some specific action in **AirBox**.

> *For example*, **DTMF** tones are used by some cable television networks to signal the local cable company/network station to insert a local advertisement or station identification.

Also, in **AirBox** you can create a **DTMF** signal and send it to the PC's sound device.

Channel in a Box User Manual
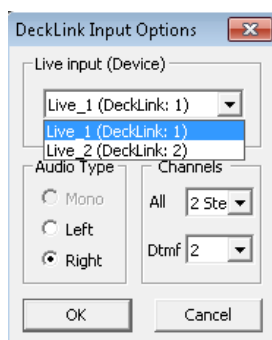
DTMF Input plug-ins

DTMF input is setup if **AirBox** receives a DTMF signal.

There are several Input sub-plugins for reading the DTMF tones, depending on the audio device, which receives the signal.

In the **Input Plug-in** tab select the receiving audio device. Currently, there are three options: **Sound Card Input**, **UDP Mpeg Input,** and **DeckLink Input**.

After selecting the DTMF Input from Modules list, press the **Configure** button to open the configure dialog. Here you can select the desired live input, as configured in the Live Inputs Editor. The device, to which the input is connected is written next to it. Select the *Audio Type* and audio *Channels* from the fields below.

Sound Card Input

This plug-in is to be used for analog audio tones, received on your PC sound card (coming from an IRD, for example). Select this line and press **Assign** to load it. Then, press **Configure** to select the sound input device and configure it:

⊙ *Mono* – select it if the incoming signal is mono;

⊙ *Left* – select it if the incoming audio signal is stereo, but you would like **AirBox** to read only the tones, coming in the left channel;

⊙ *Right* – select it if the incoming audio signal is stereo, but you would like **AirBox** to read only the tones, coming in the right channel.

UDP MPEG-2 Input
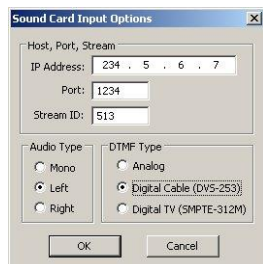
This plug-in decodes **DTMF** tones that come through **UDP** in the **MPEG2Transport Stream**.

Select the **UDP Mpeg input** from the **DTMF** plug-in list and press **Assign**. Then, press **Configure** to specify the source of the cue tones (*Host IP Address* and *Port*), and the *Stream ID*. The DTMF tones could come in three different ways:

⊙ *Analog DTMF* – the cue tones arrive as a separate audio stream in the TS. The PID of this stream should be entered in the Stream ID box.

When the *Analog* DTMF is selected, to the left you can specify the <u>source audio channel</u>.

⊙ *Digital Cable (DVS-253)* – the US implementation of digital signaling for splice points, the points are described in SI tables.

⊙ *Digital TV (SMPTE-312M)* – the European implementation of digital signaling for splice points, the points are described in SI tables.

After configuring the plug-in, click **OK** and go to the **Commands** tab. It contains all possible actions that can be executed upon receiving a certain DTMF sequence. Please, check the details in the next page.

DeckLink Input
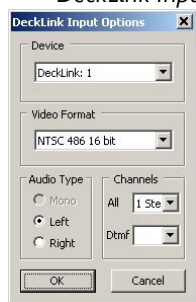
This plug-in allows for reading DTMF tones, embedded in an SDI video signal, or coming through the analog audio inputs of a *Blackmagic DeckLink board*. Select it in the plug-ins list and press **Assign**.

Then, click on **Configure** to setup the input.

If there are more than one DeckLink boards installed on your system, please choose which of them to use for capturing the cue tones.

> **NOTE:** You cannot use the same *DeckLink board* for DTMF capturing and playback at the same time!

Below you can select which audio pair contains the cue tones and whether they are coming in the ⊙ *Left* or in the ⊙ *Right* channel. Click **OK** to save your settings.

When you are ready with the plug-in setup, you can go to the **Commands** tab to create the necessary presets.

In this tab you have to specify what **AirBox** should do upon receiving a certain sequence of DTMF tones.

Enter a *Name* for the command and specify the DTMF sequence that will trigger it. Then, select what should happen upon receiving this tone sequence in the *Command* drop-down list. Most commands are self-explanatory, but there are several that need to be clarified:

Jump to / Cue to – when assigning this command, you have to specify the *Clip number,* to which it refers.

Reset – resets the master counter in the main *AirBox* window (equal to stop & play command). This event is executed between two clips and does not affect the playback.

Hardware reset– resets the hardware platform!

WARNING! This command will interrupt the playback! After resetting the hardware, *AirBox* will remain in Stop mode!

Cue – same as the *Cue* button in the main *AirBox* interface, this command will pause the playback on the first frame of the currently selected clip in the playlist.

WARNING! This command will interrupt the playback!

Jump to bookmark in time range – this command applies only to bookmarks that have some Time zone specified in their properties. *AirBox* will not care about the bookmark's names. There are two possible occasions when working with this command:

- If the cue tone arrives within the Time range, specified in the properties of any bookmark in the playlist, *AirBox* jumps to the relevant bookmark according to the current system time.

- If no bookmark in the playlist has a Time range that includes the time of arrival of the cue tone, *AirBox* ignores the command.

IMPORTANT: Please, make sure not to create bookmarks with overlapping time zones. Otherwise, *AirBox* will jump to the first bookmark in the playlist that is assigned for the relevant time zone.

Jump to bookmark – this command uses bookmarks' names for reference. You must specify the *Bookmark* name in the string below, so *AirBox* would "know" where to jump to.

There are two possible occasions when working with this command:

1) If there is a Time range specified in the bookmark's properties, and:

- the tone arrives within the time range =>*AirBox* will jump to this bookmark

- a trigger arrives outside the time range =>*AirBox* will ignore the command

Thus, you can have more than one bookmark in the playlist that is assigned for the same time range.

2) If there is NO Time range, specified in the bookmark's properties, *AirBox* will jump to that bookmark unconditionally upon receiving the trigger tone.

Please, check the Bookmarks description in the Edit menu section above for more details.

Offset – use this spin-box to postpone the execution of the specified command. The value here is in milliseconds!

NOTE: This string can contain only positive values!

Channel in a Box User Manual

> **WARNING!** All DTMF sequences (strings) must have the same length! Otherwise, make sure that longer sequences do NOT comprise some of the shorter sequences! Otherwise, we cannot guarantee the correct execution of the assigned commands.

> **TIP (!)** You can save your DTMF commands settings in a file. Thus, you can load them on other servers without having to configure the same settings numerous times.

Press the 💾 *Save* button to save the settings. To load them on another server, copy the settings file (*.dts*) to it. Then, open **Settings ➔ Modules ⇨ Remote ➔** DTMF Plug-ins configuration and press the 📂 *Load settings* button. Browse for the *.dts* file and click *Open*.

DTMF Output Plug-ins
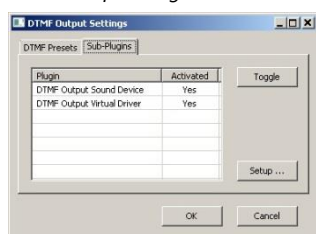
These plug-ins generate the **DTMF** tones and send them via the selected sub-plugin. Once it is configured, you can use it to add DTMF Output events in your playlist.

Select the **DTMF Output Plug-in** row and press the **Configure** button. The dialog to the right will appear. Go to the **Sub-Plugins** tab first, to select the output audio device. Press the **Toggle** button to activate it.

Currently, there are two possibilities – **DTMF Output Sound Device** and **DTMF Output Virtual Driver**:

DTMF Output Sound device
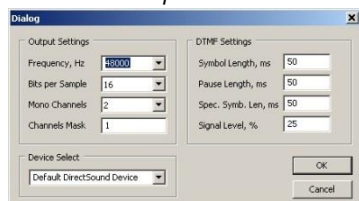
When you activate this sub-plug-in, the DTMF tone is sent to a device, defined into the plug-in configuration. Select the plug-in and press the **Setup** button. The configuration window will open.

In the *Output Settings* area you can define the parameters of the DTMF tone: *Frequency*, *Bits per sample*, number of *Mono Channels*, and *Channel Mask*. Depending on the channel Mask, the DTMF tone is sent to a specific audio channel. If Channel Mask is zero (0), the DTMF is not sent. If it is one (1), DTMF is sent to the 1st mono channel. If it is two (2), DTMF is sent to the 2nd mono channel. Note that the channel mask is a **bitmask**, where:

Stereo 1	Stereo 2
Left – 1	Left – 4
Right – 2	Right - 8
L + R – 3	L + R - 12

In the DTMF setting you can define:

Length of the standard *Symbol* in DTMF sequence – **numbers from 0 to 9 and letters A, B, C, D.**

Pause Length **–** length of the pause between symbols

Specific symbols length **–** Specific symbols are all symbols in the DTMF strings, different than numbers from **0 to 9** and **letters A, B, C, D.**

Level of the signal **–** the signal level in percent.

In the *Device select* field you can select which audio device to be used from a drop-down list. The list is created automatically, in accordance to the devices installed on your machine.

> *DTMF Output Virtual driver*

When you select this driver, the DTMF tone is sent to some third party application (which can receive tones) instead of the computer audio device.

> For example, such an application is **AirBox** itself. It can be set from **Settings ➔ Output ⇨ *Audio (General)*** tab to create DTMF tones and send them to the video output.

When you define the plug-in, you can go back to the **DTMF Presets** tab. Press the **Add** button to create a new preset. Enter the name of the preset and the DTMF string for it.

> Relay GPI Switch Input/output

This plug-in allows using of the **PlayBox GPI Relay** card.

The **PlayBox GPI Relay** card is a special hardware card designed by **PlayBox** for receiving and sending GPI signals. Each Relay supports up to four (4) GPI inputs/outputs. One or more relays could be installed on the same machine.

You can read more information about the **PlayBox Bypass Relay card** in Appendix 4 in this Manual.

The use of the **PlayBox GPI Relay** could replace the use of a COM port for receiving and sending GPI signals.

The **PlayBox GPI Relay** functionality is optional and it is not a standard part of the **AirBox** module. In order to use it, you must obtain a relevant license.

If you have a licensed **Relay GPI** option, you can define which mode (Input and/or Output) to be enabled by double-clicking on the *Enabled* field. You can configure the **Relay GPI** further by pressing the **Configure** button:

Relay GPI In
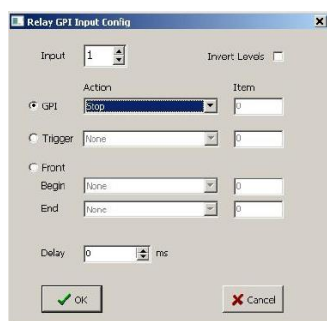
In the *Input* field select the GPI input number to be configured.

By default, the GPI signal has a High pulse level. If you want to transform the signal to a Low level, check the ☑*Invert Levels* check-box. In this way the action will be executed when the GPI relay is switched OFF, and it will be terminated when it is switched ON.

Channel in a Box User Manual

Check the ⊙ **GPI** radio-button to execute an **AirBox** action upon receiving a pulse on the Relay input. In the next *Action* field, define the **AirBox** action [Play, Stop, Pause, Resume, Next, etc.]. The next *Item* field becomes available if you have selected the [Cue to] or [Jump to] action. Then, in the *Item* field, you can specify the number of the playlist item to cue to or jump to.

If you want the same **AirBox** action to be triggered at the beginning of the GPI pulse and at the end of the GPI pulse, press the ⊙**Trigger** radio-button. If the action is [Cue to] or [Jump to], then in the *Item* field you can specify the number of the playlist item to cue to or jump to.

If you want different **AirBox** actions to be triggered at the beginning and at the end of the GPI pulse, select the ⊙**Front** radio button. In the *Begin* field, enter the **AirBox** action to be executed upon the pulse beginning. Respectively, in the *End* field, enter the **AirBox** action to be executed upon the pulse ending.

Delay – this is the delay of the **AirBox** action in milliseconds.

Relay GPI Out

The **Relay GPI Output** is performed via dedicated output events in the playlist. When the playback reaches the event, the Relay GPI trigger is activated.

In this setup window define the GPI *Output* number – the number of the GPI output, where the GPI signal will be sent.

Check the *Invert level* check-box to set the level of the pulse to be Low.

WatchDog Plug-in
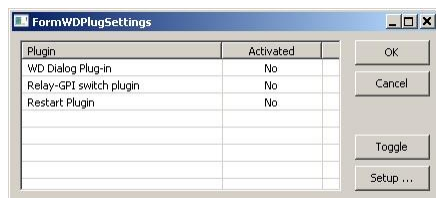
The **WatchDog** Plug-in detects if some malfunction of the application occurs. When a malfunction takes place, **WatchDog** runs "activated" Sub-Plug-ins from its sub-Plugin list

Activate the preferred sub-plugin by selecting it and pressing the **Toggle** button. When the sub-plugin is activated, you can press the **Setup** button to configure it.

Currently, there are three possible sub-plugins:

WD Dialog Plug-in.

When this sub-plugin is activated, a text message appears on the user's screen.

Relay GPI switch Plug-in.

This Sub-Plugin switches to the **PlayBox Bypass Relay card**. You can activate it and set it up if you have the **Bypass Relay card** installed on your machine.

Channel in a Box User Manual

Press the **Setup** button to enter the necessary configuration for the relay. In the *Device Select* field you will see the list of the installed **Bypass Relay cards**. Select the necessary card and press **OK**.

> **IMPORTANT:** Even if there is only one **Bypass Relay card** installed, select it in the *Device Select* field and press the **OK** button for the setting to take effect.

You can read more information about the **PlayBox Bypass Relay card** in Appendix 4 in this Manual.

Restart Plug-in

Activating this sub-plugin will restart **AirBox**.

> Elpro Switcher Output

The Configuration dialog for the **Elpro switchers** contains two radio buttons:

⊙ **Local mode** – check it if the switcher is connected to one of the PC COM ports (select it from the drop-down list below).

⊙ **Network mode** – check it if the **Elpro switcher** is connected to another **AirBox** server, somewhere in the local area network (specify the *Server* address in the string below).

> **NOTE:** In order to control an **Elpro switcher** from more than one **AirBox** instance, you need the **Elpro Switcher server**. The **Elpro Switcher server** application is an add-on, which is not included in the standard **AirBox** bundle

> Acappella Switcher Output

This configuration dialog contains only a drop-down list.

Here you have to specify the *COM port* that will interface the switcher control cable.

> Concerto Switcher Output

The configuration dialog represents the *LAN* setup for controlling Concerto switchers. Fill in the *IP address* and *Port number* of the switcher and press **OK**.

Quartz Matrix Switcher Output
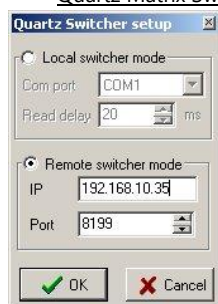
There are two possibilities for controlling the switcher:

⊙ **Local mode** – check it if the switcher is connected to one of the PC COM ports (select it from the drop-down list below).

⊙ **Remote switcher mode** – check it if the **Quartz Matrix Switcher** is connected to another **AirBox** server, somewhere in the local area network. Specify the Server *IP address* and the *Port* number and press the **OK** button.

Snell Switcher Output
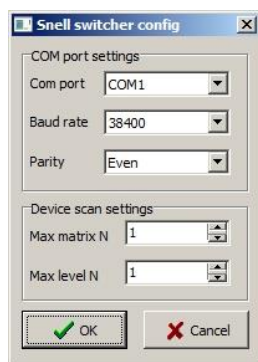
This is an external plug-in used to control Snell devices that comply with the*SW-P-08*protocol. It uses*RS-485*cables. The COM port settings may vary for different devices.

The **Snell switcher** configuration window contains the following settings:

COM port– specify which *COM port* you would like to use for connecting to the switcher. All of the available COM ports are listed in this drop-down list.

Baud rate–set the information carrying capacity of the port measured in bits/s.

Parity–this is a method used for error-detection in transmission. When parity is used with a serial port, an extra data bit will be sent with each data character. The parity bit in each character can be set to:

No Parity–this means that no parity bit will be sent at all.

Mark–this means that the parity bit is always set to the mark signal condition (logical 1) and, likewise, the **Space** parity always sends the parity bit in the space signal condition. Aside from the uncommon applications that use the 9th (parity) bit for some form of addressing or special signalling, mark or space parity is uncommon, as it adds no error detection information.

Odd parity is more common than **Even**, since it ensures that at least one state transition occurs in each character, which makes it more reliable.

The most common parity setting, however, is **No Parity**, with error detection handled by a communication protocol.

NOTE: The settings, made into the COM port settings field of the Snell switcher configuration dialog have to correspond to the ones, made into the switcher itself.

NOTE: If you have more than one device, it is necessary to specify the Max Matrix/Level value in the Device scan settings field, situated at the lower part of the dialog. Otherwise, they will not be discovered in Online mode.

<u>Network Electronics switcher output</u>
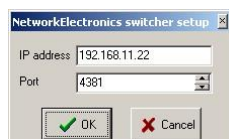
The configuration dialog represents the setup for controlling the switcher. Fill in the *IP address* and *Port* number of the switcher and press **OK**.

<u>GPI Input/Output</u>

AirBox has two GPI modes - GPI Input, where **AirBox** works as a *GPI Slave,* and GPI Output*,* where **AirBox** works as a *GPI Master*.

The GPI (General Purpose Interface) functionality is optional and it is not a standard part of the **AirBox** module. In order to use it, you must obtain a relevant license. If you have a licensed GPI option, you can define which mode to be enabled (or both of them) by double-clicking the *Enabled* field. You can configure the GPI further by pressing the **Configure** button:

GPI Input
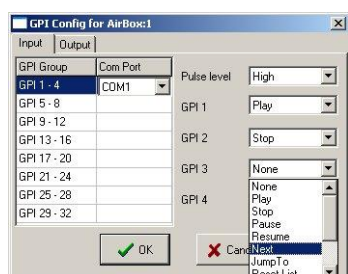
The column to the left contains a list of **GPI Groups** (four different GPI pulses per COM port). The *GPI 1* stands for the CST pulse (input on pins 8 and 5), the *GPI 2* is DSR (pin 6 and 5), the *GPI 3* is RI (pin 9 and 5), and the *GPI 4* is CD (pin 1 and 5).

Specify the COM port for each group in the **Com Port** column. In the fields to the right you can define the desired **AirBox** function (Play, Stop, Pause, Resume, Next, etc.) for the respective GPI.

If you select **Cue to** or **Jump to**, the setup window will expand for you to specify the number of the playlist item to cue or jump to.

If you select just **Cue**, the playback will pause on the first frame of the currently selected item in the playlist.

NOTE: The *GPI Input plug-in* and the *DTMF plug-in* provide identical sets of commands.

Channel in a Box User Manual
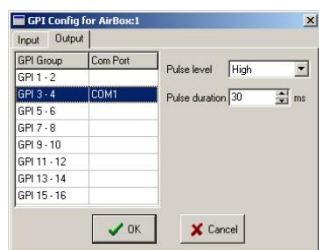
GPI Output

Here you have to specify the **GPI groups** and the respective **COM ports**, too.

The *Pulse Level* field defines the level of the trigger pulse.

The *Pulse duration* field defines the duration of the trigger pulse in milliseconds.

The GPI **Output** is performed via dedicated *GPI output events* in the playlist. When the playback reaches a *GPI event*, the GPI trigger is activated for as many milliseconds, as defined in the *Pulse duration* field. After that the playback continues. If you do not want to continue the playback, just insert a *Stop, Wait,* or *Fixed-time event* after the *GPI event.*

NOTE: You can find detailed information about the ***AirBox GPI plug-ins*** in Appendix 1, further in this manual.

IMPORTANT: The ***PlayBox*** modules DO NOT support the Deck Control connector, supplied on the DeckLink breakout cable!

Kramer Switcher Output
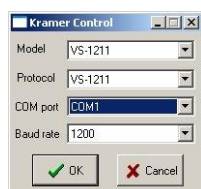
The **Kramer Switch** module allows you to control all x01 and x11 Kramer Switchers.**AirBox**controls them via the old protocol only. Protocol 2000 is NOT supported in this plug-in! The switchers may be controlled via *RS-232* and/or *RS485/422* communication ports.

The **Kramer Switch output** functionality is not a standard part of **AirBox.** If you have a license to use **Kramer Switch output**, **Enable**, and **Configure** it.

In the configuration window, you can define:

Model –the model of the Kramer Switcher used. You can select it from the pop-up list of the available ones.

Protocol – the protocol used for the respective switcher. You can select it from the pop-up list.

COM Port – the COM-port used for connecting to the switcher. You can select it from the available COM-ports.

Baud rate–the baud rate of the port.

Kramer Matrix Switcher Output

The **Kramer Matrix Switcher** module allows you to control Kramer Matrix Switchers from **AirBox**. This plug-in is also optional.
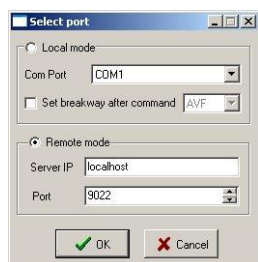
AirBox supports the entire series of Kramer matrix switchers. They provide switching between 16, 12, 10, 8, 6, or 4 video and balanced/unbalanced audio and video inputs to 16, 12, 10, 8, 6 or 4 outputs (video and audio). The matrices are controlled via *RS-232* and/or *RS485/422* communication ports. Just select the appropriate *COM Port*. For more detailed information about various Kramer matrix switchers, please, visit http://www.kramerelectronics.com

TIP (!) Use the ⊙*Remote mode* to send commands to the Kramer Matrix server. Thus, you could control one matrix from several *AirBox* channels simultaneously.

NOTE: This plug-in also supports all Kramer switchers that are protocol 2000 compatible, such as *Kramer VS-1201xl, VS-1001xl, VS-801xl, VS-601xl, VS-401xl* and *VS-1211, VS-1011, VS-811, VS-611, VS-411*. They provide switching between 12, 10, 8, 6 or 4 video and balanced/unbalanced audio inputs to one output (video and audio).

Please, check your switcher manual for instructions on how to change the control protocol, or contact support@playboxtechnology.com .

IMPORTANT: *PlayBox* modules DO NOT support the Deck Control connector, supplied on the DeckLink breakout cable!

Leitch Matrix Switcher Output
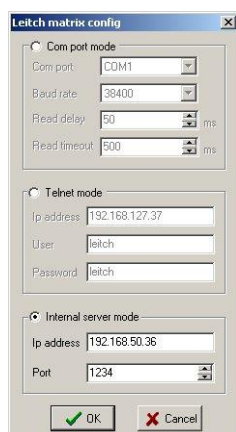
AirBox can control Leitch Matrix Switchers either through the computer COM port or using the IP address of the matrix (via TCP/IP protocol). The configuration window is divided into three areas – one per each mode.

For the ⊙ *COM port mode* you should set:

COM Port – the *COM-port* you will use for connecting to the switcher. All available *COM-ports* are listed in the drop-down list.

Baud rate– set the information carrying capacity of the port, measured in bits

Read delay–the delay between the command given to the matrix switcher and its execution. By default, it is 50 milliseconds, and it is not recommended to change this value.

Read timeout – the time, for which the switcher reports status.

If you choose to control the matrix via the TCP/IP protocol, check ⊙ **Telnet mode,** and enter the relevant *IP Address* at the bottom of the configuration dialog.

Check the factory-provided *User* name and *Password,* and type them in the relevant strings.

> **TIP (!)** The default user/password for the *Leitch Integrator* is Leitch/Leitch. For the Panacea series these are Leitch/Leitch Admin respectively.

The ⊙ **Internal server mode** is used to send commands to the Leitch Matrix Server. The Matrix Server is an add-on application, which provides possibilities to control one matrix switcher from several **AirBox** channels simultaneously. In the *IP address* field type the address of the machine, where the Switcher server is. Below, enter the Port you will use to connect the switcher.

Net Sender Output

This is a plug-in that allows you to send any text message via UDP. The parameters of this plug-in are set during event insertion. Please, check the relevant **External events➔Net Sender Output** section for further details.

Ocelot Switcher Output
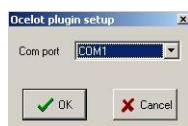
Another type of switchers, supported by **AirBox,** are the Ocelot switchers. The only thing you can set in their configuration window is the *COM port* used for connecting to the switcher. You will find the available *COM ports* in the drop-down list.

> **IMPORTANT:** The *PlayBox* modules DO NOT support the Deck Control connector supplied on the DeckLink breakout cable!

TitleBox Net Control output
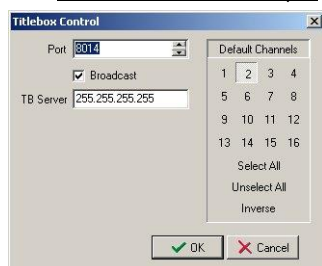
This function allows managing the **TitleBox** objects from **AirBox**. This is done by inserting external events (**TitleBox NetControl Output** events) in the playlist.

Press the **Configure** button to configure further the **TitleBox** control. You can set the connection port for this UDP communication (it is set to 8012 by default). If there is more than one **TitleBox**, controlled through **AirBox** in your network, set the **TitleBox** channel, to which **AirBox** commands should refer.

If you check *Broadcast*, the commands will be sent throughout the entire network. If unchecked, the commands will be sent only to the specified **TitleBox** Server.

For a detailed step-by-step guide on how to integrate **TitleBox** with **AirBox,** see *APPENDIX 3 – Integration of AirBox with TitleBox .*

Channel in a Box User Manual

TV Content Rating

This plug-in allows you to set a TV content rating identification to your output, which is done by adding / inserting such events to your playlist.

After you enable the plug-in from the **Enabled** column, press the *Configure* button and select the country, which rating system you would like to apply to your output. Depending on your configurations here, you will be able to add a specific external event in your playlist.

Universal Matrix Plug-in
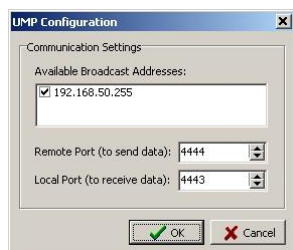
This plug-in is related to the use of the **Universal Matrix Controller (UMC).** You can read more about it in the **Universal Matrix Controller** section, further in this manual.

The **UMC** is a **server** application, installed on a separate server(s). It is an intermediary in the communication between **AirBox** and the video matrices in the local network. It communicates with all type of matrices known from **AirBox** - Kramer Switcher, Elpro Switcher, Acappella, Concerto, etc.

In the near future, the **Universal Matrix Plug-in** will replace all currently existing matrix plug-ins in **AirBox.**

Communication settings:

▪ *Broadcast addresses* – The **UMC** automatically scans the network and locates the available broadcast addresses. You can enable or disable broadcasting to a specific address by checking/un-checking the respective checkbox.

▪ *Ports* – The default port for sending data to the clients is **4443**. The default port, listened to by **UMC** is **4444**.

UDP Switcher
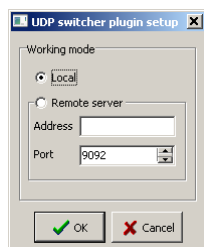
The **UDP Switcher Plugin** is used to set-up the connection to **UDP switcher**.

The UDP switcher is a server application, which runs separately. It communicates with its clients via a **UDP Switcher Plugin** (**UMP**).

In the configuration dialog, select *Local*, if the server application is running on the same machine.

Select *Remote Server* if the server application is running on a remote machine. For *Remote Server* you have to enter the *IP Address* of the machine, where the *UDPSwitcher.exe* is installed, and the port number.

Channel in a Box User Manual

> **IMPORTANT:** The port number must be 9092, as it is by default!

VideoHub Switcher Output
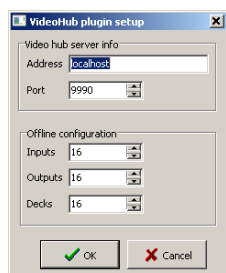
This is an external plug-in, used for performing control over VideoHub broadcast routers, produced by Blackmagic Design Pty. Ltd.

The configuration dialog of the *VideoHub control plug-in* is divided in two fields:

In the *VideoHub server info field*, specify the *Address* and *Port* of the router.

The *Offline configuration* field is used in case the matrix is not physically available. *For example*, when you use **ListBox,** and you would like to create an event for controlling the switcher, you have to create a virtual configuration of the switcher. Based on it, the relevant command will be created. In the *Offline configuration* field, specify how many *Inputs*, *Outputs,* and *RS-422 Serial control ports (Decks)* the router has. When ready, press **OK**.

Video Resize Control Output

This is an event that ensures automated scaling of the video output. Press the **Configure** button to open the **Set-Up** form, and create your video resize presets.

For more information, check the *AirBox* **Edit** menu ➔**Add** ➔ **Events** ➔*Video Resize Control Output* section above in this chapter.

VikinX Matrix Switcher Output

Pressing the **Configure** button while the *VikinX Switcher* row is selected will evoke a setup dialog, where you can define the following:

COM Port – Use the drop-down list of available COM-ports to select the one that will be connected to the switcher.

Read delay–the delay between the command, given to the switcher, and its response. By default, it is 50 milliseconds, and it is not recommended to change this value.

Once you have set these configurations, you will be able to add VikinX Switcher events in your playlist.

VTR Control Plug-in
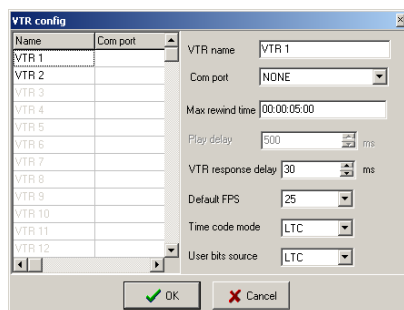
You can use it to control up to two VTRs for combined file-based/tape-based playout. Enable it and press **Configure**. A dialog will open for you to make the necessary setting. First, enter the **VTR Name** and the **COM port**, to which it is connected. Then, specify the maximum rewind time. Please, note that in the playlist you do not have to insert VTR events that are overlapping in relation to the maximum rewind time!

Below you can make some VTR-specific fine-tuning: *Play delay* and *VTR response delay*.

Further, set the default frame rate of the output in the *Default FPS* field. Finally, specify the *Time code mode* and the *User bits source*.

For details on how to insert VTR control events in the playlist, please see the relevant section above.

SierraVideo Switcher Output
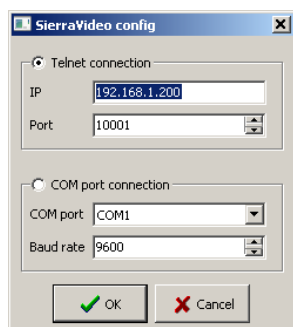
Pressing the **Configure** button while the **SierraVideo Switcher Output** row is selected will evoke a setup dialog, where you can setup the connection to this switcher. You can select from two types of connections:

⊙ **Telnet connection** – if you select this type of connection, you need to fill in the *IP* address and the *Port* number in the respective fields, in order to connect to the switcher.

⊙ **COM port connection** – if you select this type of connection, use the drop-down list of available *COM ports* to select the one that will be connected to the switcher, and fill in the *Baud rate* in the respective field.

Channel in a Box User Manual
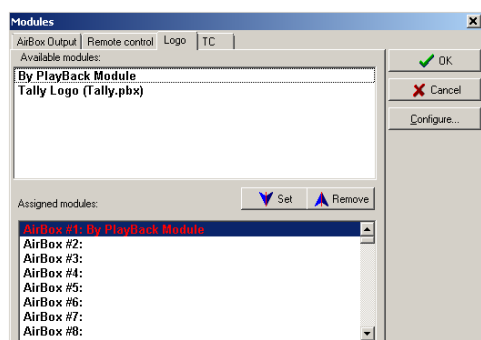
Logo

This option allows displaying a logo over the clips, played in **AirBox.** You may select a logo source from the list and assign it to the appropriate **AirBox** channel.

By Payback Module

This is an internal logo generator that keys the selected logo over the output video. This is the option to be used in most cases.

Pressing the ***Configure*** button while the ***Logo*** tab is selected will open the *Logo configuration* dialog box, where you can define *Logo Presets* to be used later in **AirBox** (see the **Logo** section of the **Settings** menu description, further in this manual).

Tally Logo
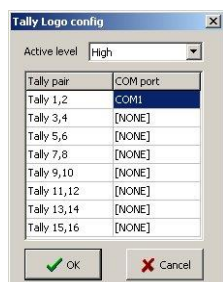
This plug-in will raise or lower the output voltage on a specified COM port when playback reaches a logo event line. This is designed to control external logo generators that support GPI input.

As only two pins on a COM port can be used for the GPI output, you can have only two different logo commands per port. Thus, you will be able to switch between Logo 1 and Logo 2 on COM 1; Logo 3 and 4 will be assigned to COM 2, and so on.

The pulse level is defined in the *Active level* drop-down menu. It is used to control the logo ON and logo OFF commands. For example, if the active level is [High], a Logo ON event turns on the output voltage, and a Logo OFF event turns it off.

> **IMPORTANT:** No logo will be displayed in the **AirBox** output!

Press ***Configure*** while the ***Tally logo*** is selected to setup the **COM port** for outputting the GPI pulse.

Channel in a Box User Manual

TC Generator
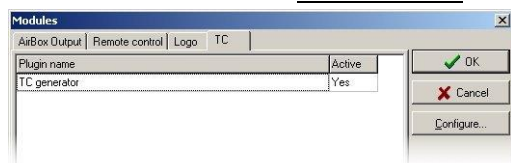
This plug-in allows for outputting LTC on the **AirBox** server sound card. Thus, you can synchronize external devices to the time code coming from **AirBox**.

To enable this plug-in, double-click in the **Active** column to the right.

AirBox can output time code from most of its counters. You have to press **Configure** to select it. The available *Counters* are listed in the lower-right corner of the **LTC Generator** dialog. You can select the counting mode (for counters that support this feature) in the *TC direction* drop-down list.
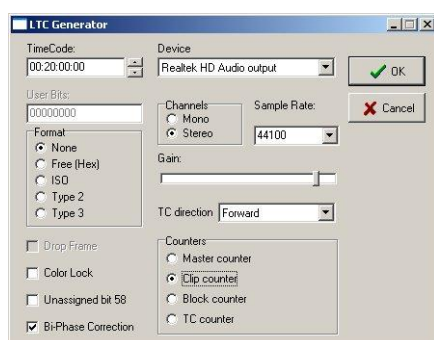
Above you can select the output *Device* and set the output channel type and sampling rate.

The left-hand side of this dialog contains some basic LTC-related settings:

Time Code – this string shows the currently running timecode.

User Bits – Depending on the selected *User Bits* format (see below), you can preview and edit the output user bits here.

Format – these options are User-bits-related and reflect the most widely-used standards.

Output

This window contains information about the settings of the currently active plug-in and allows its fine-tuning. The **Output Setup** dialogs may vary depending on the decoders' models. Usually, these settings are defined at the factory, so you do not have to adjust them, unless advised by our support team.

As explained in the Preamble section above, there are two major playout modules, supported by **AirBox**.

The **AirBox** playout is defined in the **AirBox ➔Settings ➔Output** menu, described below.

Multiformat playout

Depending on the supported hardware for video/stream output, the following Multi-format plug-ins are available: **Blackmagic DeckLink plug-in, DELTACAST HD plug-in, streaming plug-in** (LAN/ASI card based) and **Software plug-in** (VGA card based).

The software module is used mainly for demonstration or test purposes.

The settings for multi-format playout are the same for all of the supported hardware. There are just small differences, which will be described separately.

You can open the output-settings window from the **AirBox Menu ➔Settings➔Output**.

Channel in a Box User Manual

Common settings

The common settings are valid for all of the supported hardware.

Video tab

Here you can define the parameters of the AirBox video output. The settings will affect all files in the playlist!

- *Video Standard* - specify PAL or NTSC video standard
- *Field Order* – corrects the order of the field.
- *Flip Image* – check this box to flip the video vertically.

NOTE: *Flip Image* will affect ALL files in the playlist!

- Aspect Ratio – the aspect ratio of the video output.

- Aspect/Size Conversion Strategy – it is used to define how the output video will look like if the output aspect ratio is different from the aspect ratio of the files in the playlist.

For example, if the output aspect is set to **4:3**, but the clips in the playlist are with a ratio of **16:9**.

The following settings are available:

Best fit – will keep the real size of the image. Two horizontal black bars will be added above and under the video to fill-up the remaining part of the screen. The horizontal size of the video is preserved.

Balanced Letterbox – the horizontal black bars will be smaller. Small parts at the two sides of the video will be left out of the screen.

Full screen/Zoom – the vertical size of the video will be preserved. Video will be cut equally to the left and to the right.

Stretch – using this setting, there will be no black bars on the screen, but a little change in the form of the shapes.

Custom – it is used for custom settings. When you select it, the drop-down box at the right side of the *Aspect/Size* will become active, as well as the *Conversion Strategy* box. The latter allows you to adjust what portion of the image and what portion of the black bars will be visible in percentage.

For example, if you select 50 %, the result would be like the **Best Fit** Aspect/Size ratio. At 100% the result will be like **Full screen/Zoom,** and at 0% – like **Letterbox**.

- Rescaling – In case the source file has a different resolution than the output resolution, there might be some Rescaling of the video.

⊙ **No Rescaling** – if selected, there will not be any rescaling.

⊙ **Crop** – if the source file has a higher resolution than the output resolution, only the middle part of the video will be visible. If the input video has a lower resolution, the remaining part of the output picture is filled with a black color.

⊙ **Standard** – this setting corresponds to the Low one from the *Custom* list.

⊙ **Custom** – if pressed, the user has to choose the type of quality from the drop-down list near the relevant radio-button.

NOTE: Working with higher video quality requires more CPU power.

- Resampling –In case the source video has a different frame rate than the output one, the following *Resampling* options are provided:

⊙ **No Resampling** – there will be no resampling.

⊙ **Repeat/Skip** – for example, if the output video standard is set to NTSC, but you have to play a file, which belongs to the PAL standard, some frames will be repeated, in order to fill up the frame number to 30. By analogy, some frames could be skipped.

⊙ **Standard** – corresponds to the **9-Very High** quality from the *Custom* list.

⊙ **Custom** – if selected, you have to select an item by your choice from the drop-down list near this radio-button

Audio (General) *Tab*
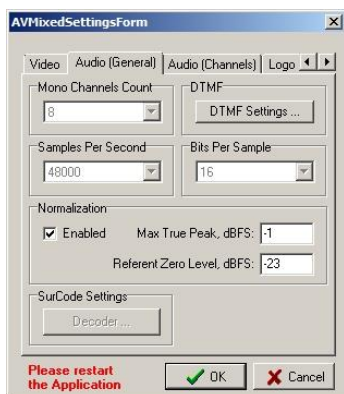
Here you can define some settings for the audio output.

Channel in a Box User Manual
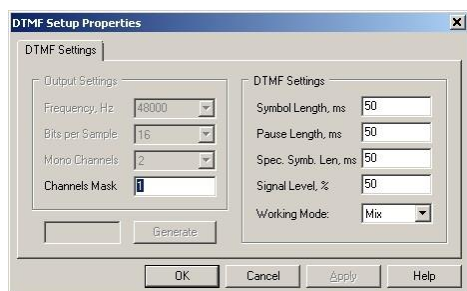
DTMF settings.

Here you can define an output DTMF signal to be generated by the **AirBox** application. Press the **DTMF settings** button to open the **DTMF set-up window**.

- ▪ *Output settings:* Here you can define the parameters of the DTMF tone – *Frequency*, *Bits per sample*, number of *Mono Channels*, and *Channels Mask*.

Depending on the *Channels Mask*, the DTMF tone is sent to a specific audio channel. If *Channels Mask* is zero (0), the DTMF is not sent. If it is one (1), DTMF is sent to the 1st mono channel. If it is two (2), DTMF is sent to the 2nd mono channel.

Length of the standard *Symbol* in DTMF sequence **–numbers from 0 to 9 and letters A, B, C, D**

Pause Length **–** length of the pause between symbols

Specific Symbols Length **–** Specific symbols are all symbols in the DTMF strings, different from numbers from **0 to 9,** and letters **A, B, C, D**

Signal Level % **–** the signal level in percent

Working Mode **–** here you can set the DTMF to either be <u>Mixed</u> with the output audio, or to <u>Replace</u> the output audio

The Normalization field

This new functionality is developed in accordance with *EBU Rec. R 128*. It is designed for our clients to benefit from the following:

- ▪ Performing monitoring and control over the perceived loudness of audio sources.

- ▪ Reducing the loudness differences between programmes. This is applicable when the nature and content of audio matherial(s) changes frequently. *For example*, such a problem most commonly occurs at the breaks for advertising, when there is a jump between audio levels.

AirBox performs the audio normalization by using a loudness metadata that describes how loud each programme is. The <u>loudness</u> metadata is provided by the **QCBox** tool.

QCBox performs quality control tests of the media content. After that it stores the test results as metadata into a file with extension **.mtd*. Such an **.mtd* file is generated separately for each media file tested.

Based on the metadata, provided by **QCBox** for each media file, **AirBox** will adjust the audio levels of all media sources separately.

IMPORTANT: *AirBox* does not process the audio by applying any permanent modification of the source content (such as compression, peak limits, EQ). It just changes the audio during playout by adjusting the audio levels.

A basic explanation of the workflow of the entire process is described below:

- ✓ Loading the playlist

✓   Accessing the *.mtd* files, which correspond to the content and are created by the **QCBox** tool for reference

✓   Reading the Programme Loudness values, described for each *.mtd* file. Programme Loudness refers to the <u>average</u> loudness of a programme.

Based on that, **AirBox** performs a correction of the audio, in order to reach the referent -23/-24 LUFS. This value is configurable through the output settings, provided into this window, as follows:

Tick the ☑ *Enabled* check-box in order to activate the normalizing process.
When normalizing, the operations, described below will follow:

▶   <u>Performing a level shifting</u>

    AirBox does not perform any processing of the source sound internally. It normalizes the loudness of the whole programme (file-based playout channel) by the means of level adjusting. **For example,** if there is a source file with average programme loudness of **-32 dBFS**, it will be increased with **9 dB** in order to reach the referent **-23 dBFS,** specified in the **AirBox** output adjustments.

▶   <u>Keeping   track</u>
    of the *Maximum True Peak level* of the audio. If the maximum peak level exceeds **zero dB**, there will be a clipping introduced. Therefore, if the maximum true peak level exceeds the value, specified in the **AirBox**Output Settings, **AirBox** assumes that the source is not prepared in a good manner by the sound engineer. In such a case **AirBox** will not reproduce the relevant media source.

Let us *follow the latter example* with the source media file, having average programme loudness of **-32 dBFS**. If the maximum true peak level of this file is **-8 dB**, when increased with **9 dB** the peak will jump to **1** (above zero). As mentioned above, exceeding the zero may introduce clipping or other audible distortions, and exceeding the Max TPL value, specified for the **AirBox** output settings (it is recommended for this value to be **-1 dB**), will cause skipping of the source's playback.

A value of **-23 dBFS** corresponds to the *Referent Zero Level* (by default). This is a target level, according to which the programme loudness levels should be normalized, as recommended in *EBU Tech. Rec. R 128*. There is a tolerance of **±1.0 LU,** which exists for programmes, where an exact normalization is not achievable in practice.

For the *Max True Peak dBFS* string you should indicate the maximum TPL allowed. According to *EBU R 128,* it is **-1dBTP** (measured with a **True Peak** meter).

For more detailed information, please refer to the chapter, dedicated to **QCBox** and *Example 1* in *Appendix 7*further in this manual.

Channel in a Box User Manual

Audio (Channels) Tab
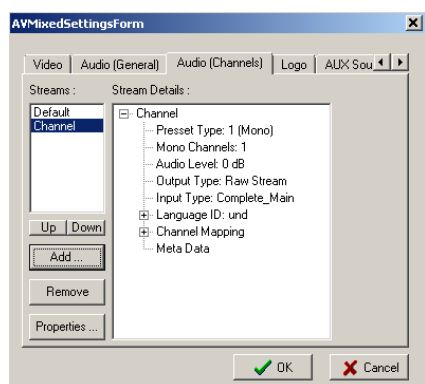
It contains the following settings:

- **Streams** (audio tracks)

The list of streams is visible in the **Streams** window. When you select a stream from the list, you can see the details for this audio stream in the next **Stream Details** window.

To define/remove/edit a stream, press the ***Add/Del/Prop*** button respectively at the right side of the window.

To change the order of the streams in the stream list, use the ***Up*** and ***Down*** buttons.

When you press the ***Add*** button, the set-up window will open.
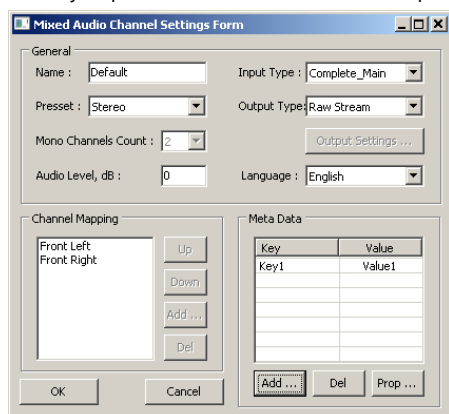
- Enter the name of the new stream in the *Name* field.

- In the *Preset* field you can select a preset for a channel mapping from a drop-down list. Here you can see a list of the most popular channel mappings.

For example, if you select **Stereo**, the channel mapping will contain a [Front Left] and a [Front Right] channel. If you select ***3 front LFE***, there will be: [Front Left], [Front Right], [Front Center], and [LFE Subwoofer].

- In *Mono Channels Count* enter the number of mono-channels, included in this stream.

Channel in a Box User Manual

NOTE: If you have selected some preset from the *Preset* list, then the number of the mono channels appears automatically, and you cannot change it.
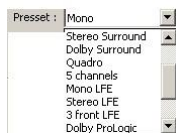
If you select the *Preset* to be **Unknown**, you can enter the number of mono channels manually, and you can define your own channel mapping into the *Channel Mapping* area.

▪ *Channel Mapping*
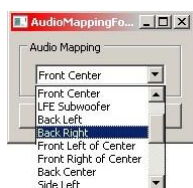
By default, when you select some number of mono channels, in the *Channel Mapping* field, you will see the most popular mapping types for them.

NOTE: There is an option, in which the Channel Mapping could be disabled. If you would like to disable these settings, please, contact our support team at support@playboxtechnology.com.

If you select 2 mono channels, the *Mapping* will contain a [Front Left] and a [Front Right] channel.

If you select 4 mono-channels, the Mapping will be: [Front Left], [Front Right], [Front Center], and [LFE Subwoofer].

You can delete one or more of these mono-channels by pressing the **Del** button to the right of the *Channel Mapping* area. Then you can add a new channel for audio mapping, by pressing the **Add** button.

To change the channels' order, just select one of them and move it up or down in the list, by pressing the **Up** or **Down** button.

NOTE: If you have selected some preset from *Preset* list, then you cannot edit the *Channel Mapping*

IMPORTANT: In case the available in-file mono-channels are different in count than the ones, selected for the output (for example, you have 2 mono-channels in file, but define 4 mono-channels for the output), the re-mapping is made by the software up to the most-popular re-mapping rules. If you want to make your own re-mapping, please, ask our **PlayBox** support team (support@playboxtechnology.com) for assistance.

▪ *Audio Level in dB* – Enter a value here for changing the output audio level of the current stream. The default value is zero (0), which means that there is no change. If you enter one (1), the output audio level will increase by one (1) dB. If you enter minus one (-1), the output audio level will decrease by one (1) dB.

▪ In the *Input Type* field, you can select from a drop-down list a type for the sound in this audio stream, like "dialogue", "music and effects", "commentary", "voice over", etc.

▪ In the *Output type* field select the output encoding method – *PCM* or *Dolby Digital*.

Channel in a Box User Manual

NOTE: Currently, *Dolby Digital* output is supported with a *Deltacast* card only.

> ▪ ***Output settings*** – This button is activated when the *Output type* is *Dolby Digital*. It provides a set-up window of the *Minnetonka SurCode for Dolby Digital*. Its detailed description is available further in the section <u>Settings for Deltacast HD Mixed Output</u>.

> ▪ The language of a sound could be selected from the *Language* field.

> ▪ *Meta Data* – this is a list of additional audio metadata for user-defined audio streams. *For example,* default volume, copyrights, etc.

You can **Add**, **Delete**, or **Edit** metadata by selecting it and pressing the ***Add/Del*** or ***Prop*** button respectively.

When you are ready, press the ***OK*** button at the bottom of the ***Mixed Audio Channel Settings*** window. The newly defined audio stream will appear in the stream list.

Please, check <u>*Examples 2 – 19*</u> in <u>*Appendix 5*</u> further in this manual for a more thorough understanding of the functionalities of this *AirBox* feature.

IMPORTANT: If you have defined more output audio streams than you have in your file, then, by default, the extra streams will be empty. It is possible to dublicate the first output audio stream into the extra streams. For further instructions and help, please, contact our ***PlayBox*** support team (<u>support@playboxtechnology.com</u>).

<u>Logo Tab</u>
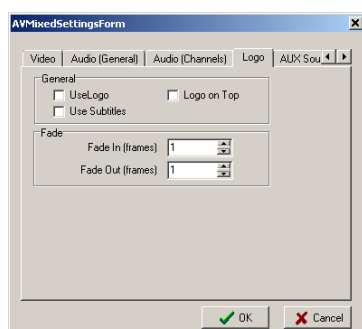
This tab contains two fields – *General* and *Fade*.

The first one contains the following check-boxes:

☑ *Use Logo* – if not checked, the user will not be able to use logo.

☑ *Use Subtitles* – check it to enable the usage of subtitles.

☑ *Logo on top* – you can use it to avoid eventual overlapping of the logo and the subtitles.

To adjust the duration of the *Fade In* and *Fade Out* effects when showing/hiding the logo, use the relevant spin-boxes, situated in the *Fade* field.

AUX Source Tab
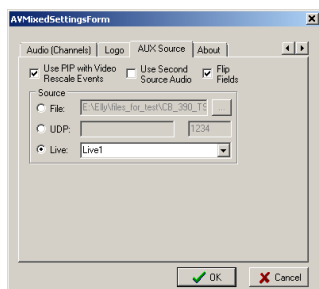
There are settings for using an additional video source. There are three possible AUX *Sources*:

⊙ Previously prepared **File** – you need to enter its location here;

⊙ **UDP** stream – enter the IP address and port;

⊙ **Live** video – a video signal, coming from a **Live Input**, as it is defined in the **Live Input** settings window. Here you have to enter the name of the live source, exactly like it is entered in the **Live input** window.

The additional video source is usually used for creating Picture-in-picture (PiP).

To allow the PiP functionality, you have to check the ☑*Use PIP with Video Rescale Events* box. The video files from the playlist will be treated as a background video. The live source video will be resized in accordance to the settings, made within the *Video Resize Control event* (for more information, please, check the Video Resize Control Output description).

NOTE: Keep in mind that when creating the resize control event, you have to specify an offset of at least 1 second (1000ms).

Check the ☑ *Use Second Source Audio* box if you want the output audio to be the one, coming from the AUX source.

IMPORTANT: It is recommended for the live stream video standard and fields order to be similar to the playlist input video.

NOTE: This plug-in supports the transition between clips in the playlist. Please, check the clip properties description for details.

Channel in a Box User Manual

<u>Settings for Deltacast HD Mixed Output</u>

Video settings

The Deltacast playout module supports the creation of *Genlock*. **GenLock** (short for **gen**erator **lock**) is a common technique, where the video output of one source, or a specific reference signal, is used to synchronize other television picture sources together.

Press the **Genlock** button in the video settings window to open the **GenLock** settings dialog.
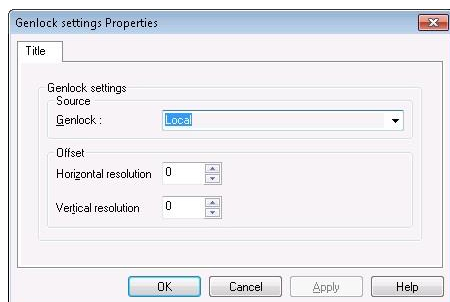
Here you can define a *Source Genlock* and *Offset* of *Horizontal* and *Vertical resolution*.

The *Genlock sources* could be **Local** (generated internally from the Deltacast HD card) or **Blackburst** (coming from an external source, connected to the black burst port of the card).

The *Offset* is defined in pixels.

Audio settings

The Deltacast playout module supports **Dolby Digital** audio output type.

The Dolby Digital functionality is being provided by **SurCode for Dolby Digital**. **SurCode for Dolby Digital** and is manufactured under a license from **Dolby Laboratories**. **SurCode** is a trademark of **Minnetonka Audio Software**.

Select a Dolby Digital as an Audio Output type in the audio-stream settings window, and then press **Output settings** button to open the **SurCode for Dolby Digital** window.

Below, there is a description of **SurCode** windows as per **SurCode** user's guides.

Main settings

Channel in a Box User Manual

PLAYBOX
TECHNOLOGY
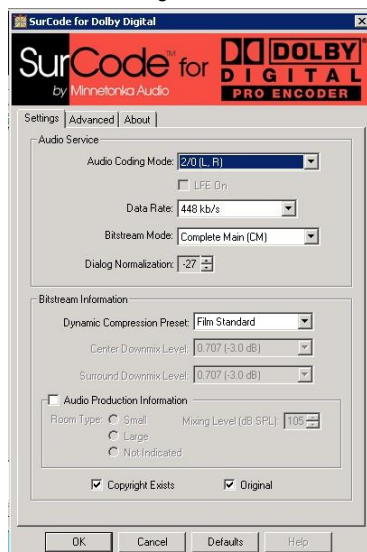
- ▶ Audio Services

    - **Audio Coding Mode –** from this pull-down window, you can select the input mode - mono, stereo, or surround input;

    - **Data Rate -**it indicates/sets the sound file sample rate.

  - **Bitstream Mode**

The Bitstream Mode parameter indicates the type of audio service that the bitstream conveys. Complete Main (CM) is the normal mode of operation and contains a complete audio program including dialog, music, and effects. The CM and ME Main Services can be further enhanced by means of Associated Services.

The Bitstream Modes and audio service types are:

Main Service: Complete Main (CM)

Main Service: Music and Effects (ME)

Associated Service: Visually-Impaired (VI)

Associated Service: Hearing-Impaired (HI)

Associated Service: Dialog (D)

Associated Service: Commentary (C)

Associated Service: Emergency (E)

Associated Service: Voice Over (VO) / Karaoke

  - **Dialog Normalization (*dialnorm)***

The Dialog Normalization (*dialnorm*) value indicates how far the average dialog level of the encoded program is below digital 100% full scale (0 dBFS). Valid settings are *1 dB to -31 dB*. This parameter determines the audio reproduction level and affects other parameters and decoder operation.

- ▶ **Bitstream Information**

The parameters in this group directly relate to the Dolby Digital Bitstream

Information (BSI) fields.

  - **Center Downmix Level**

The Center Downmix Level parameter indicates the nominal Lo/Ro downmix level of the Center channel with respect to the Left and Right channels. This parameter setting does not affect Lt/Rt downmixes.

This parameter *appears in the bitstream* only when three front channels are in use, i.e., only when the Audio Coding Mode is set to 3/0, 3/1, or 3/2

  - **Surround Downmix Level**

The Surround Downmix Level parameter indicates the nominal Lo/Ro downmix level of the Surround channel(s) with respect to the Left and Right channels (consistent with the ITU BR specification). This parameter setting does not affect Lt/Rt downmixes.

This parameter *appears in the bitstream* only when a Surround channel is in use, i.e., only when the Audio Coding Mode is set to 2/1, 2/2, 3/1, or 3/2. It is recommended that the parameter be user-adjustable only when one of these modes has been selected.

  - **Audio Production Information Exists**

Channel in a Box User Manual

The Audio Production Information Exists flag indicates whether the Mixing Level and Room Type parameters explained below exist within the Dolby Digital bitstream.

- **Room Type**

The Room Type informational parameter indicates the type and calibration of the mixing room used for the final audio mixing session. The Room Type value is not normally used within the Dolby Digital decoder but can be used by other elements in the audio system. This parameter appears in the bitstream only when the Audio Production Information Exists parameter is set to 1, or yes.

Valid values for Room Type**: *Large room; Small room; Not indicated*

- **Mixing Level**

The Mixing Level informational parameter indicates the absolute Sound Pressure Level (SPL) of the audio program as heard by the original mixing engineer. This information makes it possible to replay the program at exactly the same loudness, or at a known difference in loudness. By knowing how much lower a program is played at home, for example, it is now possible to apply the correct degree of *loudness compensation*.

The value for Mixing Level represents the theoretical loudness of a full-scale (0 dBFS) tone in one channel.

- **Copyright Exists**

The Copyright Bit informational parameter sets the value of a single bit within the Dolby Digital bitstream. If this bit has a value of 1 (the check-box is checked), the information in the bitstream is indicated as protected by copyright. If it has a value of 0 (the check-box is not checked), the information is not copyright protected.

- **Original Bitstream**

The Original Bitstream informational parameter sets the value of a single bit within the Dolby Digital bitstream. This bit has a value of 1 (the check-box is checked) if the bitstream is an original. If it is a copy of an original bitstream, it has a value of 0 (the check-box is not checked)

Advanced settings
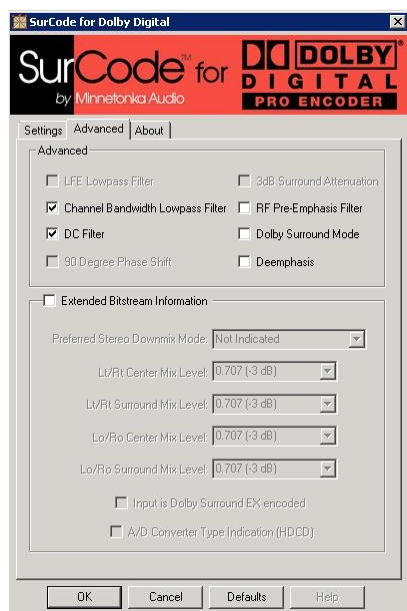
- **LFE Lowpass Filter**

The LFE Lowpass Filter parameter can be used to activate a 120 Hz low-pass filter applied to the LFE input channel. If the digital signal fed to the LFE input does not contain information above 120 Hz, this filter can be disabled. This parameter is user adjustable only when the LFE channel is enabled.

- **Channel Bandwidth Lowpass Filter**

The Channel Bandwidth Lowpass Filter parameter can be used to activate a low-pass filter with a cut-off near the specified audio bandwidth that is applied to the main input channels. If the digital signal fed to the main input channels does not contain information above the specified audio bandwidth, this filter can be disabled.

- **DC Highpass Filter**

This parameter can be used to activate the DC Highpass filter for all input channels. The DC Highpass filter should always be enabled unless the encoding engineer is absolutely sure that there is no DC in the input audio.

- **Surround Channel 90-Degree Phase-Shift**

The Surround Channel 90-Degree Phase-Shift feature is useful for generating multichannel Dolby Digital bitstreams that can be downmixed in an external two-channel decoder to create a true Dolby Surround compatible output. This parameter is user-adjustable only when Surround channels are present in the bitstream, i.e., only when Audio Coding Mode is set to 2/1, 2/2, 3/1, or 3/2. The 90-Degree Phase-Shift parameter should always be left enabled except under specific conditions. These include, but are not necessarily limited to, system calibration, encoding of certain test signals, and in the *extremely rare* case when the discrete playback of highly coherent program material may be compromised.

- **Surround Channel 3 dB Attenuation**

The Surround Channel 3 dB Attenuation function is useful for applying a 3 dB attenuation to the Surround channels of a multichannel soundtrack created in a room with film style calibration, when encoding it for consumer home theater playback. Cinema soundtrack Surround channels are mixed +3 dB relative to the front channels in order to account for cinema calibration standards. Home theater Surround channel gains are calibrated differently, and so a -3 dB adjustment to the Surround tracks is necessary. This parameter is user-adjustable only when Surround channels are present in the bitstream, i.e., only when Audio Coding Mode is set to 2/1, 2/2, 3/1, or 3/2.

- **RF Overmodulation Protection (RF Pre-emphasis Filter)**

The RF Overmodulation Protection parameter determines whether or not an RF preemphasis filter is used in the overload protection algorithm to prevent RF overmodulation in set-top box decoders. It is primarily used for broadcast applications.

- Dolby Surround Mode

- **Digital De-emphasis**

Dolby Digital encoders can allow activation of digital de-emphasis applied to the linear PCM input signals whenever it is detected that the input has been preemphasized. Detection is typically achieved by monitoring the pre-emphasis flags within the channel status data of the incoming digital audio signal (e.g., AES/EBU or S/PDIF). Since the value of this parameter depends on some other parameter(s) or condition(s), it does not require explicit user control and can be adjusted automatically by the encoder.

- Extended Bitstream information (AC3 metadata)

About more information, please refer to the SurCode for Dolby Digital documentation.

Settings for MPO Mixed Playback

When you have selected the MPO Mixed Playback in **Settings➜Modules⇨*AirBox Output***, an additional tab, named **Outputs**, will appear in the ***AirBox*Settings** menu **➜Output** dialog. This tab allows you to add, remove, and edit different outputs.

> **NOTE:** The hardware devices are pre-configured to be viewed by ***AirBox*** by our staff duging installation. This is done by an application, called ***Device Manager***.
>
> **NOTE:** The resources, which are reserved by the ***Device Manager***, are visible here.
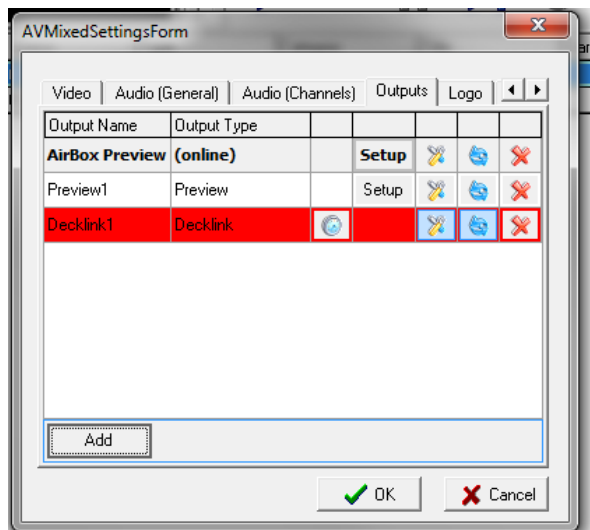
As shown in the image to the left, the **Outputs** tab allows you to view a list of already configured outputs. The **Output Name** column shows you the individual names you have given to each separate output. In the **Output Type** column you can see the type of that particular preset.

The **Clock** column (third one from the left) allows you to select the clock of which hardware device to be used. Note that if you only have one hardware device configured, it will be used by default. If you set the clock of non-hardware output to be used, AirBox will simply use the system timer. Be aware that in this case an additional timebase correction may be necessary. In the example to the left the clock of the Decklink platform is selected.
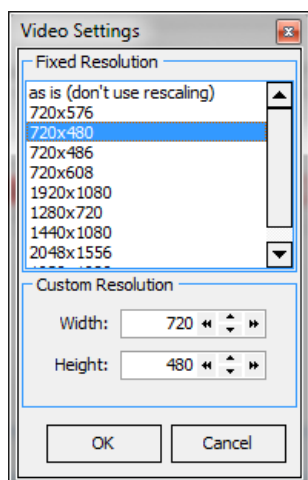
The **Setup** column allows you to invoke the setup dialog of each separate output. To do that, simply double click on the corresponding [Setup] row and a dialog will appear, depending on the type of output.

The next column, fifth from the left, invokes a dialog like the one shown to the right. Here you can set the resolution of the output.

The sixth column from the left is used for refreshing the output, and the last one – for removing it.

99

Channel in a Box User Manual
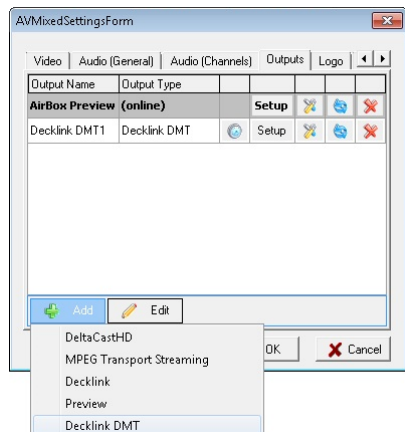
To add a new output preset, simply press the Add button at the bottom. You can select from one of the following types of outputs: **Deltacast HD**, **MPEG Transport Streaming**, **Decklink**, **Preview,** or **Decklink DMT**. Depending on your selection, a different setup dialog will appear.

Adding DeltaCastHD or Decklink
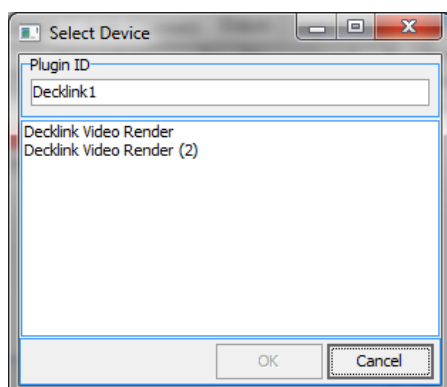
If you select **Decklink** or **Deltacast HD**, a dialog, as the one above will appear. Here you can simply enter the name of your output, and choose an existing Decklink platform from the list. Once you are done, press **OK**.

Adding MPEG Transport Streaming
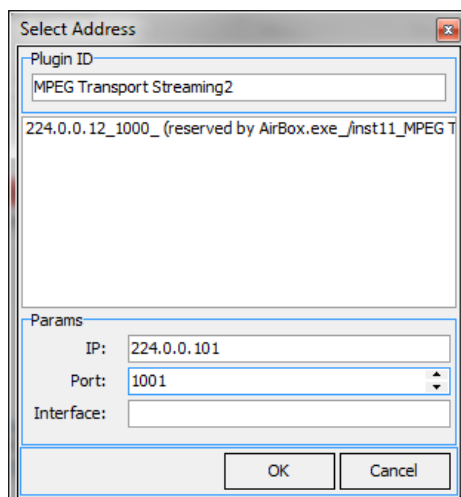
Channel in a Box User Manual

If you add an **MPEG transport streaming** output, a dialog like the one above will appear. Enter the *IP* address, *Port* number, and *Interface* name of the respective **AirBox**, to which you wish to send the transport stream.

Adding Preview
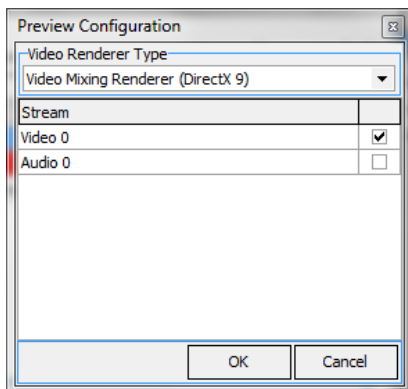
If you select a **Preview** output, a dialog as the one to the right will appear. Select the Video Renderer Type from the drop-down list and place a check next to the streams you wish to output (Video / Audio).

> **NOTE:** If there are Closed Captions in the content that is played, you will be able to see them in the preview window.

Pressing the **Plugin Setup** for a predefined **MPEG Transport Stream** in the **Outputs** tab invokes a dialog, like the one below.
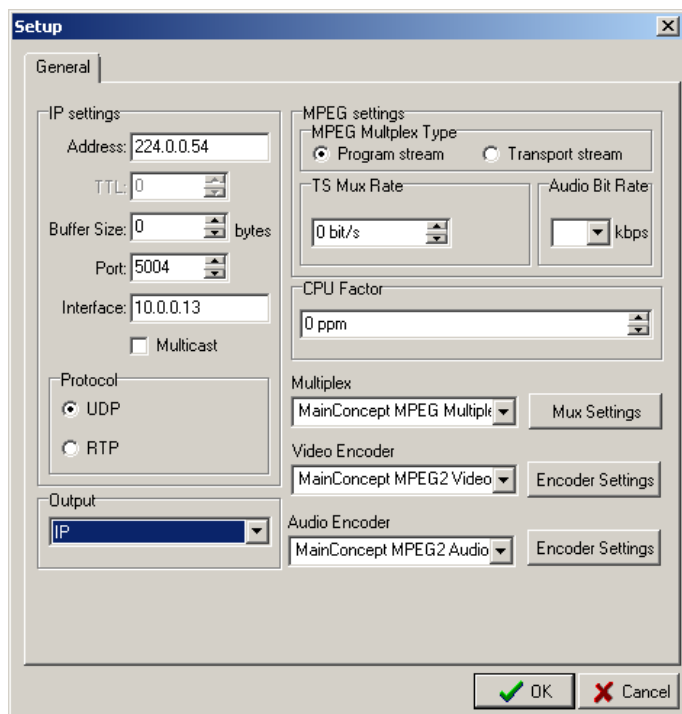
- In the *IP settings* field configure the following:

PLAYBOX
TECHNOLOGY

- ▪ *Address* – enter the IP address of the remote machine to receive the stream. If you check ☑*Multicast*, i.e. streaming to multiple machines, you will have to enter a special multicast address in the *Address* field (the multicast addresses are in the range between 224.0.0.0 and 239.0.0.0).
- ▪ *TTL* stands for time-to-live for the Internet packets. This spin-box is active only if the ☑*Multicast* box is checked. The higher the *TTL* value, the longer the packets will "live" and pass through more network routers. By default the TTL is set to "1," which is enough for local networks.
- ▪ *Buffer size* – in this spin-box you can specify the size of the IP packets to be sent along the network. This size should be a multiple of 188 bytes, which is the size of an MPEG-2 TS packet. We recommend a 1316 bytes buffer size, which will comply with the standard network MTU.
- ▪ *Port* - define which communication port will be used for the streaming. By default it is 1234.
- ▪ *Interface* - if there is more than one network card in the PC, you need to specify which one should be used for the streaming in this string. If you leave it empty, **AirBox** will stream through all of the available interfaces.
- ▪ *Multicast* – check it to stream to all PCs in the network.
- ▪ *Protocol* – the network protocol.
- ▪ In the *Output* drop-down menu select whether the output to be sent over ASI or over IP
- - In the *MPEG settings* field configure the following:

If you click on ⊙ **Program stream**, the protocol will be switched automatically to ⊙ **UDP**. With ⊙ **Transport stream** you can choose between a ⊙ **UDP** and an ⊙ **RTP** protocol.

The MPEG stream settings are situated at the right side of the window. There are two options for the *Multiplex Type*: Program or Transport stream.

When ⊙ **Transport stream** is selected, the *TS Mux Rate* spin-box becomes active, so you could either select ☑ *Automatic,* or fix the Mux rate manually. The *Audio Bit Rate* is adjustable for both stream types. It is 192kbit/s.

If you select ⊙ **Program stream**, the network protocol will be switched automatically to ⊙ **UDP**.

With ⊙ **Transport stream** you can choose between UDP and RTP.

CPU time factor – similar to IP Pump, in IP Streamer we use the CPU clock to generate the PCRs of the Transport Stream. Enter the value from the stream analyzer in this spin box to compensate the inaccuracy of the CPU clock frequency.

Pressing the **Multiplexer Settings** button will open the MainConcept Multiplexer© properties dialog. You can view the parameters of the output MPEG-2 TS stream in it.

The Video Encoder Settings and the Audio Encoder Settings are described below.

Channel in a Box User Manual

Adding Decklink DMT

If you select **Decklink DMT**, a dialog will appear, where you can simply enter the name of your output and choose an existing Decklink platform from the list. Once you are done, press **OK**.

NOTE: DMT Decklink Output only works with driver 9.7.2 or later.

Pressing the **Plugin Setup** for a predefined **Decklink DMT** in the **Outputs** tab invokes a dialog, like the one below:
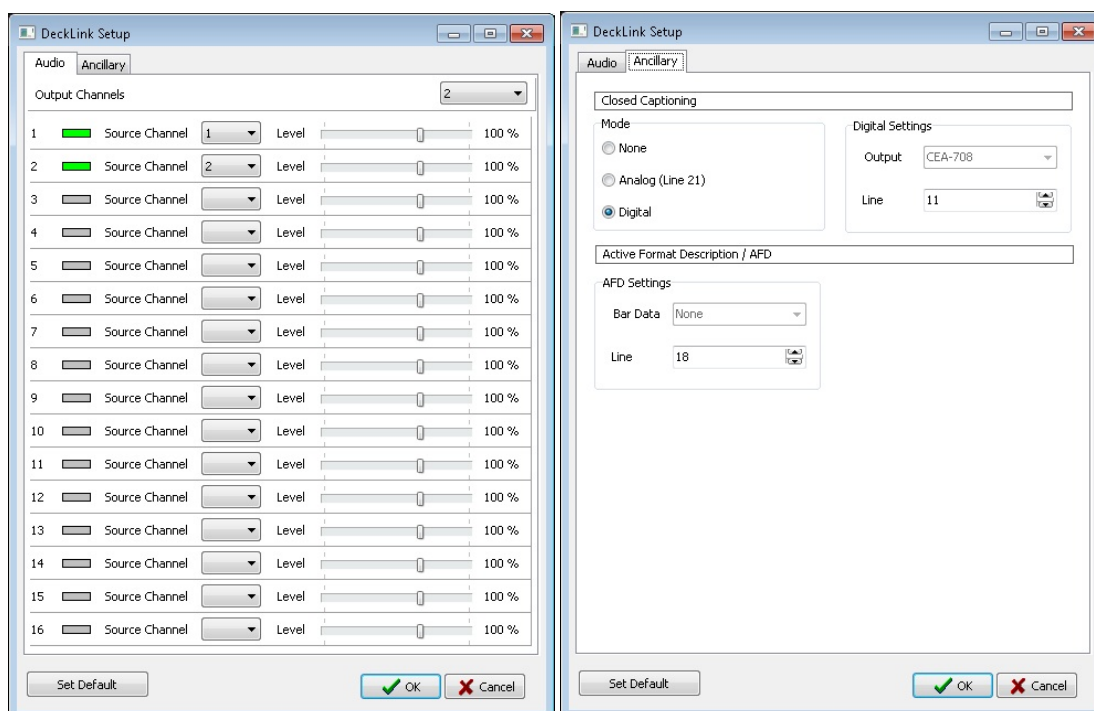
This dialog contains two tabs – **Audio** and **Ancillary**.

In the **Audio** tab you can configure the number of output audio channels to be used. This is done via the drop-down list in the top right corner of the dialog. Up to 16 audio channels are available here.

Once you configure the number of output audio channels, additional options for each channel are available in the list of channels. For each *Source Channel* in the list you can configure the corresponding output channel, as well as the volume *Level*.

Use the **Set Default** button at the bottom to return to the default configurations of the **Audio** tab.

In the **Ancillary** tab you can configure closed captions and AFD settings.

In the Closed Captioning field select the closed captioning mode:

⊙ **None** no closed captions will be output

⊙ **Analog (line 21)** analog closed captions will be output on line 21

103

Channel in a Box User Manual

⊙ **Digital** – digital closed captions will be output. Notice that you can configure the digital closed captions in the next field – *Digital Settings*. Select the *Output* type from the drop-down list, both CEA 608 and 708 are supported. Use the Line scroller to select the line, on which you would like closed captions to be output. By default, line 11 is used here.

In the Active Format Description / AFD field you can configure the *Bar Data* type, as well as the output *Line* to be used.

NOTE: Please, note that for the correct usage of the AFD feature you have to set the following output configurations in **Settings ➜ Output ⇨ Video** tab:
– *Aspect Ratio* = Auto
– *Aspect/Size Conversion Strategy* = Stretch

<u>Settings for Streaming Mixed Output</u>
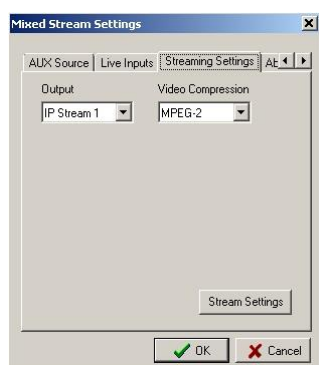
The streaming plug-in sends the playback stream to a network IP address or an **ASI** (Asynchronous Serial Interface) output, instead to a video output board.

Currently, the supported streams are **ASI** (Asynchronous Serial Interface) stream and UDP stream (IP stream). For using ASI stream, you have to have a dedicated ASI card installed (currently the supported ASI card is Deltacast ASI).

For the steaming output, into the output settings window, there is an additional settings tab – **Streaming settings.**

As it is possible to run more than one *AirBox* instance (channel) on the same PC, you have to define the *AirBox* output channel number in the *Output* field (IP Stream # or ASI Stream #).

Then, for the run *AirBox* channel, define the output **Video Compression** type. It is possible to have *MPEG-2* or *AVC/H.264* video compression.

There are different streaming settings for *MPEG-2* and *AVC/H.264*.

Channel in a Box User Manual

MPEG-2 streaming settings
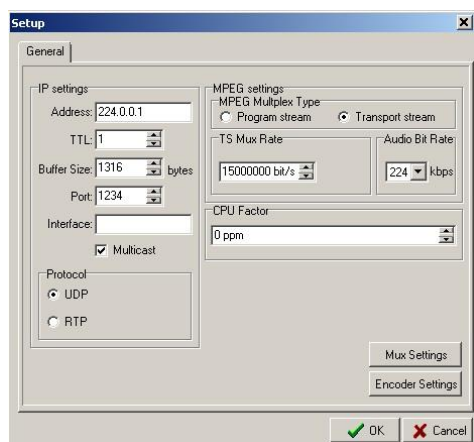
If you select MPEG-2 stream, the following setting window opens:

IP settings:

- *Address* – enter the IP address of the remote machine to receive the stream. If you check ☑*Multicast*, i.e. streaming to multiple machines, you will have to enter a special multicast address in the *Address* field (the multicast addresses are in the range between 224.0.0.0 and 239.0.0.0).
- *TTL* stands for time-to-live for the Internet packets. This spin-box is active only if the ☑*Multicast* box is checked. The higher the *TTL* value, the longer the packets will "live" and pass through more network routers. By default the TTL is set to "1," which is enough for local networks.
- *Buffer size* – in this spin-box you can specify the size of the IP packets to be sent along the network. This size should be a multiple of 188 bytes, which is the size of an MPEG-2 TS packet. We recommend a 1316 bytes buffer size, which will comply with the standard network MTU.
- *Port* - define which communication port will be used for the streaming. By default it is 1234.
- *Interface* - if there is more than one network card in the PC, you need to specify which one should be used for the streaming in this string. If you leave it empty, **AirBox** will stream through all of the available interfaces.
- *Multicast* – check it to stream to all PCs in the network.
- *Protocol* – the network protocol.

If you click on ⊙ *Program stream*, the protocol will be switched automatically to ⊙ *UDP*. With ⊙ *Transport stream* you can choose between UDP and RTP.

MPEG settings:

The MPEG stream settings are situated at the right side of the window. There are two options for the *Multiplex Type*: Program or Transport stream.

When ⊙ **Transport stream** is selected, the *TS Mux Rate* spin-box becomes active, so you could either select ☑ *Automatic,* or fix the Mux rate manually. The *Audio Bit Rate* is adjustable for both stream types. It is 192kbit/s.

If you select ⊙ **Program stream**, the network protocol will be switched automatically to ⊙ *UDP*.

With ⊙ **Transport stream** you can choose between UDP and RTP.

CPU time factor – similar to IP Pump, in IP Streamer we use the CPU clock to generate the PCRs of the Transport Stream. Enter the value from the stream analyzer in this spin box to compensate the inaccuracy of the CPU clock frequency.

Pressing the **Mux Settings** button will open the dedicated setup dialog of installed Multiplexor module. Please, do not change the default settings here, before consulting with our support team.

Pressing the **Encoder Settings** button will open the setup dialog of the video codec properties dialog.
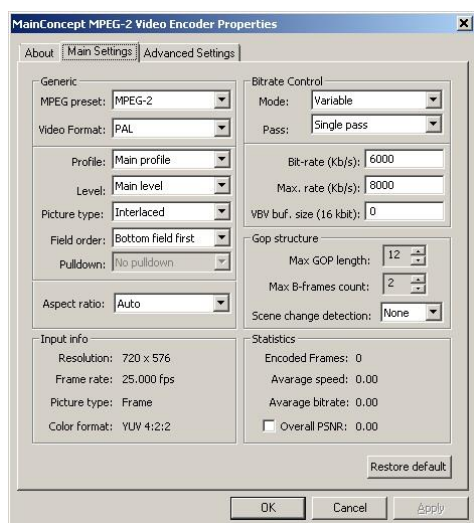
You can set some encoding video parameters here, like video bitrate, Video Profile, GOP structure, etc.

For reference, here is a brief description of the options for the MainConcept© video encoding.

In the *MPEG Preset* drop-down list all of the available MPEG video compression types in the Main Concept encoder are listed.

The *Video Format* drop-down list provides a list of video formats – PAL, NTSC, SECAM, etc.

Below you can select the encoding Profile, Level, Picture type.

Further down you can specify the Field Order (bottom or top first) and the Pull-down mode. Pull-down is applied for conversion when the source bitrate and the output bitrate are different.

The Bit Rate Control field is situated to the right. There you can specify the bit rate Mode: *Constant, Variable,* or *Constant Quantization*. The specific bitrate values strings can be managed from the subfield below.

The *Pass* drop-down list is used to specify the multi-pass encoding mode:

> Simple – encoding without gathering statistics

> Analyze – encoding and gathering statistics for next pass

> Encode – encoding using the gathered statistics and updating it.

In the GOP structure area, you can specify:

Max GOP length – the maximum length of any Group of Pictures. Larger numbers frequently provide better compression. Smaller numbers provide better error recovery and better access to the frames for editing.

Max B-frames count – the maximum number of *B frames* in a GOP. Some decoding situations, such as video conferencing, may require "no B-frames" for providing low communication delay.

Scene change detection – enables/disables the scene change detection.

The ☑ *Overall PSNR* check-box is situated in the lower right corner of the *Statistics* filed. It enables/disables the measuring the Peak Signal to Noise Ratio. This requires some CPU resources, so check it only if you need these statistics.

The **Advanced Settings** page contains more sophisticated options that should not be changed, unless advised by our support team!

AVC/H.264 streaming settings
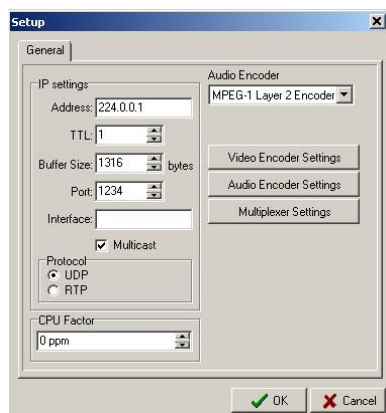
- The IP settings are the same as the Network settings in the MPEG2 Streaming plug-in, described above.

- Pressing the **Video Encoder Settings** button will open the video codec properties dialog.

You can check the description of the MainConcept©, video codec settings for reference.

- Pressing the **Audio Encoder Setting** button will open the audio codec properties dialog.
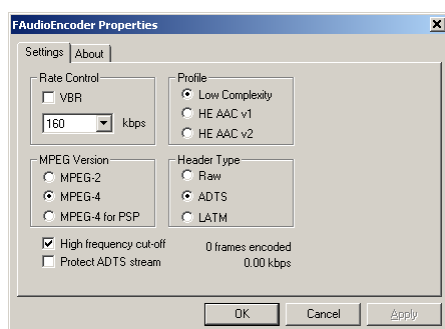
The MainConcept AAC Encoder© properties dialog is described below.

The *Bitrate Control* specifies the average output audio bitrate; the *Profile* sets the object type; and the *Header type* sets the output bit stream format (raw or with ADTS headers).

Check ☑ *High frequency cut-off* to reduce the encoded data.

- Pressing the **Multiplexer Settings** button will open the MainConcept Multiplexer© properties dialog. You can view the parameters of the output MPEG-2 TS stream in it.

Pump playout

DataPump MPEG Program Dumper

This Plug-in is used for dumping the playlist into a single MPEG2 file.

Pressing the **Play** button will open a browse dialog for saving the resulting file.

There are no setup options for this plug-in.

Channel in a Box User Manual

> **NOTE:** This feature works faster than the real-time one. For example, a 15-minute playlist compiles as a single file for less than 10 minutes. The speed depends on the *HDD* performance and the input/output stream configuration.

IP Pump MPEG Program Dumper
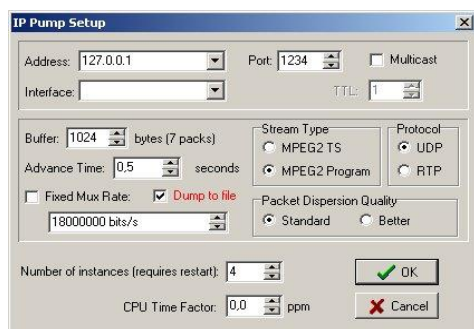
It is an optional playback plug-in that streams the content that is being played out to a network IP address. The IP Pump uses a standard UDP mechanism for data transfer.

In the *Address* field, enter the IP address of the remote machine to receive the stream. If you check *Multicast*, i.e. streaming to multiple machines, you will have to enter a special multicast address in the *Address* field. Usually its first three digits are in between 224 and 239.

If you have more than one network connection, specify which one should be used for the streaming in the **Interface** string. If you leave it empty, **AirBox** will stream through all of the available interfaces.
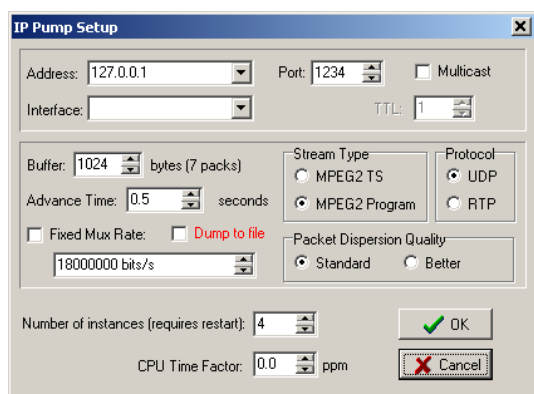
In the *Port* field, define which communication port will be used for the streaming. By default, it is 1234.

TTL stands for time-to-live for the Internet packets. This spin-box is active only if the ☑*Multicast* box is checked. The higher the *TTL* value, the longer the packets will "live" and pass through more routers. By default, it is "1," which is enough for local networks.

In the *Buffer Size* field you can specify the size of the IP packets to be sent along the network. It should be a multiple of 188 bytes, which is the size of an MPEG-2 TS packet.

The *Advance time* value represents the maximum time for buffering before initiating the streaming.

Use the ☑ *Dump to file* checkbox for recording the playlist to a file.

☑ *Fixed Mux Rate*: This check forces the plug-in to output a stream with fixed bit rate. This is done through zero packets, also known as "stuffing packets". It is applicable for transport stream output only!

> **NOTE:** The mux rate must be set to at least 10% higher than the highest content bitrate.

Protocol – now RTP is supported along the UDP protocol. RTP is considered a more professional choice because of the time stamping in the packets' headers. It is applicable only for transport streams!

Packet Dispersion Quality- as the name states, this forces the plug-in to output better dispersed packets. This means that the jitter will be much lower, thus producing a more "professional" stream. ⊙*Better* should comply with the standard – MPEG-2 Real-Time Interface Specification (ISO/IEC 13818-9 [65]).

If you have multiple channel licenses, specify their number in the spin-box in the bottom of this dialog. Thus, you will be able to see the same number of IP pump plug-ins in **Settings➔Modules⇨Output**. By default, there are four IP pump plug-ins there.

CPU Time Factor – in IP Pump we use the CPU clock to generate the PCRs of the Transport Stream. Enter the value from the stream analyzer here to compensate for the inaccuracy of the CPU clock frequency.

> **WARNING!** Consult your network administrator prior to changing the network-related settings.
>
> **NOTE:** There is no overlay preview with this plug-in!

Logo
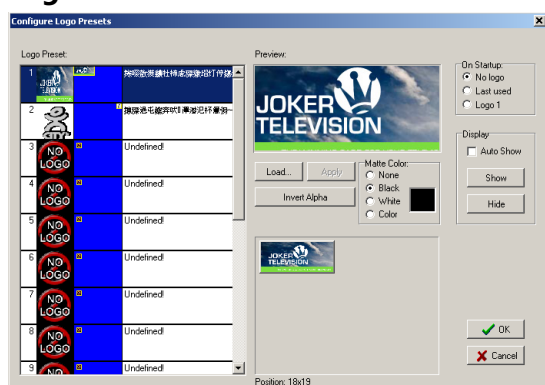
The **Logo settings** dialog is used to configure logo images for displaying in **AirBox** (their position on the screen, transparency, etc.).

The *Logo Preset* field contains a list of logo presets available for use.

To load a new preset, press the **Load** button, and select a file. It will appear in the *Preview* area. If you approve it, press the **Apply** button to save the logo file in the preset list.

Invert alpha – inverts the alpha key of the logo (if the original picture file has an alpha).

Matte Color – this is the background color, which should be mixed with transparent colors, and thus, removed from the logo image.

Position – illustrates the logo position on the screen.

On Startup – these options are used to define the logo appearance upon pushing the **Start** button in **AirBox**:

⊙ **No Logo** –no logo appears when **AirBox** is started.

⊙ **Last used** –If there are no Logo preset events in the playlist, the last used logo will appear when **AirBox** is started.

Channel in a Box User Manual

⊙ ***Logo 1*** – If there are no Logo preset events in the playlist, Logo Preset #1 (from the list) will appear when ***AirBox*** is started.

☑ *Auto show* box – check it if you want to display the logo while you are editing it.

NOTE: All changes will be shown on the screen immediately!

Show button – starts displaying the logo.

Hide button– hides the logo from the screen.

NOTE: The picture format depends on the decoder. The 32-bit RGBA files are most preferable, but TGA, JPG and PSD files are widely supported too.

IMPORTANT: You will not have fade transition when displaying Logo and Subtitles simultaneously on the DeckLink platforms!

Please, refer to the [SubTitleBox section](#) for more details on the subtitling options in ***AirBox***.

Logging

![Log Options dialog box]

AirBox generates two types of log files: System logs and an As-Run log. A *Logs* directory is created upon ***AirBox*** installation. It contains a *System* subfolder by default. *Errors, System events, Events,* and *Event type* are logged in the *System log,* regardless of your preferences.

In the ***Log Options*** dialog box you can select what playlist entry information will be included in the logs, if they should be created on a *Daily* basis or not, and you can create log-management settings.

If ☑*Enable logging* is checked, an *As-Run* log file will be created. Otherwise, ***AirBox*** will not generate *As-Run* logs. When you first check this box, browse for a folder, in which to save your *As-Run* logs. If you do not select a folder, a subfolder *As-Run* will be created in the *AirBox\Logs* directory. Your *As-Run* log files will be saved there.

Delete system logs older than [30] *days* – this option will help you manage your system log files by deleting the ones older than the number of days set in the spin-box. You can choose between 14 and 100 days.

Delete as-run logs older than ⌞30⌟ *days* – this option acts in the same way as the one above but for the *As-Run* logs.

You can add info fields in the log by checking the relevant check-boxes in the list. Use the red arrows to change the position of the currently selected item in the *Log* file.

Some third-party applications require an exact number and content of the columns in the log file. Therefore, we have created several empty fields that can be inserted in the log to comply with such specific requirements.

Below the list of info fields, there are several selectable options:

☑ *Insert date in first line of file* is used in daily logs, if you want to view the date in the first line of the log as well. When unchecked, the date is contained only in the filename, and the log itself contains only time entries.

☑ *Column headers* represent a line in the log that displays the log configuration under it. Thus, any time you change the log configuration (through the check-boxes and red the arrows) and click **OK**, a new column header line will appear in the logs.

☑ *Log Frames* will display the number of frames at the end of all time-containing values.

☑ *As-run log per playlist* – checking this box will create a separate as-run log each time a new playlist is loaded in **AirBox**.

☑ *Log metadata* – all metadata fields will appear as additional tab-delimited entries in the row of the relevant clip.

You can add new columns to the log by pressing the plus sign button. This allows for logging certain Metadata fields from the clip properties. All you have to do is specify the name of the Metadata you want to log (as it appears in the first column of the Metadata tab).

You can also set the periodicity of log files – Daily log or aggregated files.

If ☑ *Daily log* is selected, a new *As-Run Log* file will be created each day. The *System logs* are always daily, despite this setting. The *Day start at* field determines when to start the new log. This setting is necessary, in case you wish to log 24-hour periods that start at your wish (for example, at 02:00:00 o'clock, instead of 00:00:00)

The *Log* file is generated in simple Tab-delimited text format. It can be imported to various applications, text editors, etc.

> **TIP (!)** You can easily export the logs to MS Excel format from **File** menu ➔ **Export logs in XLS files…**.

> **NOTE:** Regardless of the above settings, **AirBox** will generate System logs that are saved in the program directory. The minimum period for keeping the system logs is 14 days (see above).

Channel in a Box User Manual
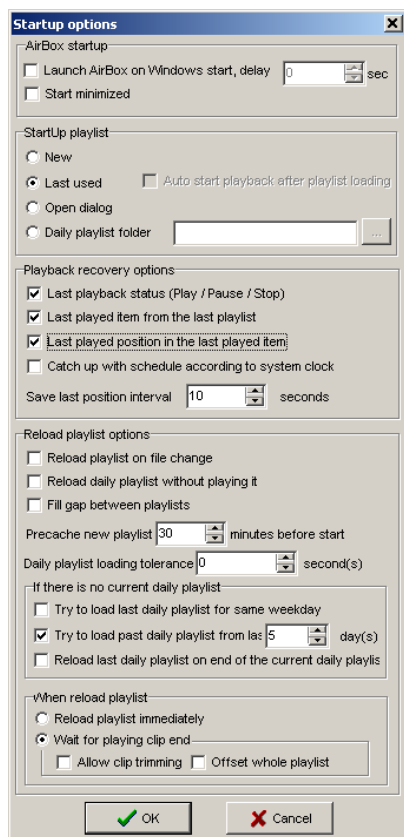
StartUp

Here you can find useful options for setting the behavior of **AirBox** on module start-up: *StartUp Playlist*, various *Playback recovery options*, how playlists will be reloaded and two system options.

AirBox startup

At the top of the **StartUp** window, you can choose between two system options:

☑ *Launch AirBox on Windows start, delay* [30] *sec.* – you might need some delay, as the system might take some time to initialize the installed decoder. If **AirBox** starts before that, it will not be able to perform normally, as the decoder will be "missing". Increase the delay to ensure that **AirBox** will not start before hardware initialization.

☑*Start minimized* – if you place a check in this box, the **AirBox** will be automatically minimized on startup.

StartUp playlist

In this section you can choose a playlist to be loaded upon **AirBox** startup.

- ⊙ **New** loads a blank playlist;

- Select the ⊙ **Last used** option and the next time you start **AirBox** it will load the last played list. In addition, you can check ☑*Auto start playback after loading* the last used playlist.

- ⊙ **Open dialog** invokes a query to choose a playlist from the last used playlist location.

- ⊙ ***Daily playlist folder*** gives an opportunity for some simple scheduling of playlists. Select a folder, in which you will place the daily playlists by filling-in its path, or just browse for it with the browse button. There you will save the playlists you want to schedule. They require a specific naming structure in order to be played on the desired date and time. An example name is "2003_11_20_14_00_00.ply," where 2003 stands for the year, 11_20 is for MM_DD (month_day), and 14_00_00 is HH_MM_SS (hour_minutes_seconds). Only files with such a naming structure will be played automatically when ***AirBox*** is running in this mode. Now you can use the [Save daily playlist](#) feature in the ***AirBox*** **File** menu to achieve this naming automatically.

> **IMPORTANT:** If you are using the ⊙***Daily playlist folder*** option make sure that you have defined a folder, where the daily playlist folder is situated!
>
> **NOTE:** For more thorough understanding of the AirBox options, described above, please, check *[Examples 20 – 28](#)* in *[Appendix 5](#)* further in this manual.

Playback recovery options

Here you can specify what ***AirBox*** should do in case of accidental restart /abnormal termination of the module. The following options are available if ***AirBox*** is not running in daily playlist mode:

☑ *Last playback status (Play/Pause/Stop)* – if checked, it resumes the last playback status of ***AirBox,*** depending on the status, in which it was before stopping.

☑ *Last played item from the last playlist* – if checked, it resumes the last item from the playlist that was played by ***AirBox*** before stopping.

☑ *Last played position in the last played item*– functions as the two previous options but it also remembers the position of the last item to resume status exactly at it is. The interval for saving the last position is changeable through the spin-box below.

☑ *Catch up with schedule according to system clock* – this option allows playlist recovery after a failure and synchronizes it with the system clock.

When you set this option, you should bear in mind that the default start time for all playlists in ***AirBox*** is 00:00:00 hours. Therefore, if you have started your playlist manually, after a power failure, ***AirBox*** will resume playback from that point in the playlist, which corresponds to the initial time (00:00:00) plus the current system clock time. *For example,* you start your playlist manually at 9:00 o'clock; and at 9:30 a power failure that persists till 10:00 occurs. After the power supply is recovered at 10:00 o'clock, ***AirBox*** will resume playback from the point in the playlist that is 10:00 hours away from its beginning (considering 00:00:00 as an initial start time). In short, ***AirBox*** will "forward" the playlist to the point of power supply recovery, taking 00:00:00 as initial time.

In order to overcome this, you should compile your playlists to start with [wait events](#). The wait events' values should be set in hours from 00:00:00 to the desired start time of your playlist, i.e., for the above example, your playlist should start with a "wait for 9 hours" event.

Thus, in case of power failure, ***AirBox*** will account for the initial time, set by you, and will resume playback from the point that corresponds to the power recovery time. Let us consider the example above. With this setting, at 10:00 o'clock ***AirBox*** will start from the point, which corresponds to one hour after the beginning of the playlist (set to 9:00 o'clock through a wait event).

> **NOTE:** Playback recovery options will not be applied when ***AirBox*** is running in daily playlist mode! In daily playlist mode, ***AirBox*** will always try to catch-up with the schedule according to the playlist start time and the current system time.

Reload playlist

In this section you can adjust settings, related to the way ***AirBox*** handles playlist reloading.

☑ *Reload playlist on file change* – reloads the current playlist automatically, if in the meantime it has been changed and saved by another user.

☑ *Reload daily playlist without playing it* – in certain occasions playlists should not start playing when reloaded.

Channel in a Box User Manual

> **WARNING!** When this box is checked, **AirBox** will not calculate the insertion point when reloading playlist on file change, either!

☑ *Fill gap between playlists* – if there is a gap between the current playlist and the playlist after it, AirBox will use the <u>auto-fill</u> clip to compensate for this gap if this box is checked

Pre-cache new playlist `30` *minutes before start* – specify how many minutes before the playlist starts **AirBox** should cache it.

> **IMPORTANT!** The pre-cache time value has to be AT LEAST equal to the duration of the last file in the old playlist plus the time of the gap between the two playlists.
>
> **NOTE:** For long playlists, we recommend entering higher pre-cache time values.

Daily Playlist loading tolerance `1` *second(s)* – specify the daily playlist loading tolerance in seconds. Loading a playlist always takes a few moments, depending on its length. **AirBox** calculates the exact point to start playback after the playlist is loaded. Thus, if there is an event in the beginning of the playlist, it might be skipped (as the event's duration is zero, its time could be considered to have passed after playlist loading). Therefore, you should set some loading tolerance to ensure the correct execution of the first event in the playlist (if any). Usually, one second should be enough, but it depends on the playlist length.

> **WARNING!** Always enter a positive value in this spin-box. **AirBox** will not load a daily playlist at an earlier time!

If you work in Daily playlist mode, it is important to clarify what will happen in the case *If there is no current daily playlist*. **AirBox** offers the following possibilities:

☑ *Try to load last daily playlist for same weekday* – **AirBox** will search the daily playlist folder for old playlists for the same day of the week (i.e., Fridays), and will load the most recent one.

☑ *Try to load past daily playlist from last* `5` *days* – **AirBox** will search the daily playlist folder for old playlists in accordance to your settings and will play the most recent one (i.e., yesterday's, if available).

> **NOTE:** If there are three playlists from the previous day (i.e. 10:00, 12:00, and 18:00), **AirBox** will load the latest one (i.e., 18:00), regardless of the current time (i.e. even if now it is 9:00 o'clock).

☑ *Reload last daily playlist on and of the current daily playlist* – **AirBox** will load the currently playing daily playlist again, as soon as it ends.

If you have checked ☑*Reload playlist on file change,* it is important to define when the reloading will happen.

When reloading playlist

In the field below you can set two additional rules on playlist reloading – reload the playlist immediately, or reload the playlist after the end of the currently played clip.

⊙ *Reload playlist immediately:*

Depending on the playlist type –<u>Regular</u> or <u>Daily</u>, the behavior of **AirBox** is different:

Regular (not Daily) playlist operation:

There are two possible situations for reloading the playlist:

Channel in a Box User Manual

1). If the currently playing clip (its List ID) is available in the newly edited playlist, **AirBox** will reload the new playlist and will start playing it from the next item (next List ID). Thus, **AirBox** will jump to the next item in the playlist.

2). If the currently playing clip (List ID) is removed from the newly edited playlist, then **AirBox** will reload the new playlist and will start playing it from the beginning.

Daily playlist operation

AirBox will remove all items from the current playlist <u>after</u> the currently played item. Instead of them, **AirBox** will load the items from the newly edited playlist.

> **IMPORTANT:** In daily playlist you can edit only those items, which come after the currently playing item!
>
> **IMPORTANT:** You can save the changed playlist and reload it only if there is no pre-cashed file or playlist. A pre-cashed file is a file, locked for playout.

⊙ *Wait for playing clip end*

> **NOTE:** This option is available only for **Regular (not for Daily)** playlists.

1). If the currently playing clip (its List ID) is available in the newly edited playlist, **AirBox** will finish playing it. Then, it will load the new playlist, and will start playing the next clip accordingly.

2). If the currently playing clip (List ID) is removed from the newly edited playlist, **AirBox** will reload the new playlist and will start playing it from the beginning.

When selecting this radio button, you have two additional options:

> ☑ *Allow clip trimming* – check it if you would like the currently playing clip to be trimmed upon playlist reloading
>
> ☑ *Offset whole playlist* – check it if you would like the whole playlist to be offset upon playlist reloading

> **NOTE:** For more thorough understanding of the AirBox options, described above, please, check *Examples 29 – 40* in *Appendix 5* further in this manual.

Timecode
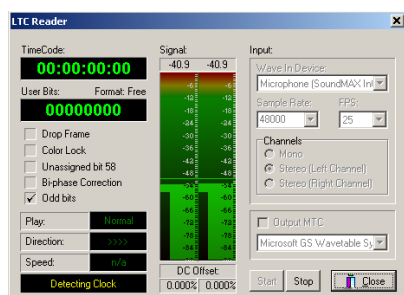
If you are licensed to use the **LTC reader** module and you have a *DirectSound* compatible sound card, you may use it to read LTC timecode from your transport device. The setup of this module is quite simple. Select your sound card device from the list. Select *Sample rate* and *Channels* and **Start** the module. The current timecode is shown in the *TC Timer*.

Channel in a Box User Manual

The timecode, fed into the LTC reader, will be used when inserting Wait TC events in the playlist. Thus, you can save the playout status of **AirBox** to an external TC generator.

> **WARNING!** Do not type any values in the *Sample Rate* string! Select these values only from the *Sample Rate* drop-down list!

Auto-fill
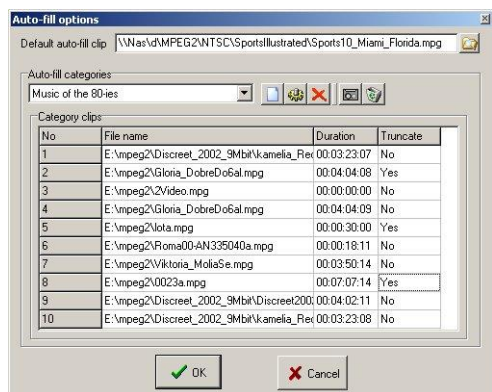
In the **Auto-fill options** dialog you can specify a default auto-fill clip, and create your own auto-fill categories, containing certain clips. These settings are necessary for the automatic filling of gaps that may appear in your playlist when using Fixed-time events.

On the top of this window you can see the *Default auto-fill clip* string.

Browse for it by pressing the folder button to the right.

The *Auto-fill categories* field is situated below. The drop-down list contains all previously prepared categories (hence, it will be empty when you open it for the first time). To create a new category, press the white sheet button to the right, and enter the name of the new *Category*. Click **OK**. You can rename your categories later by pressing the gear-wheel button.
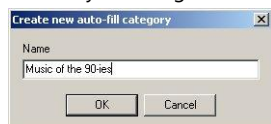
Now you have to add clips to your category. Press the tape button to open a browse dialog to locate the clip.

Let us have a look at the clips' description below. The first three columns need no explanation, but the last one is very important. The **Truncate** column contains info on whether the respective clip can be truncated, or not (you can change [YES]/[NO] by double-clicking in the relevant cell). This is important because if you have a 2 minutes gap for example, and all of the clips in the auto-fill category are longer than 2 minutes and cannot be truncated, there will be a conflict. As a result, none of the clips will be played out and the default auto-fill will be executed instead.

> **NOTE:** When playing auto-fill categories, **AirBox** will first look for clips that can be played out as a whole, then for clips that can be truncated. Thus, if you have a two-minute gap and the auto-fill category contains the following:

Clip#1 with **Duration** 3:00, **Truncate**=[YES]

Clip#2 with **Duration** 2:05, **Truncate**=[NO]

Clip#3 with **Duration** 1:50, **Truncate**=[YES]/[NO],

Channel in a Box User Manual

AirBox will playout Clip#3 first (since it can be played out to its end), and then Clip#1 (since it can be truncated). This comes to illustrate that the order of the clips in the category can be changed automatically, depending on the gap size.

To delete a clip from the current category, select it and press the delete selected clip ![x] button. If you want to remove the entire category, press the delete category ![icon] button.

> **NOTE:** If the user does NOT define any auto-fill clip, AirBox will use its own default auto-fill clip, which is simply an empty video, showing a black screen.

Fixed-time event

When there is more than one fixed-time event in the playlist, they might be in conflict with each other (this can happen if you insert/append a playlist containing fixed-time events to the current playlist, which also contains fixed-time events).

The first section in this dialog is designed for setting the rules for automated resolving of such conflicts between consecutive fixed-time events:

Default start time conflicts resolving
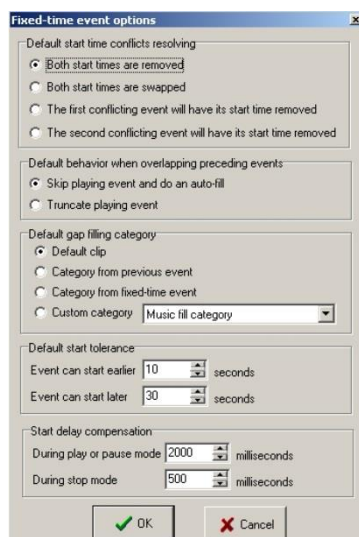
Let us assume that there are two fixed-time events already inserted in the playlist, but the one that is programmed for later playback (hereafter called **the second**) has an earlier start time than the preceding fixed-time item (hereafter called **the first**). In other words, **the first** has a later start time than **the second**. This dialog provides four possible actions if there is such a case in the playlist:

⊙ **Both start times are removed** – if you check this option, the start times of **the first and the second** fixed-time events will be removed and they will be played-back as ordered in the playlist, without executing the defined start times.

⊙ **Both start times are swapped**– check this and the items' start times will be exchanged. As a result, **the first** item will be played-back first, at the start time, set for **the second** item. **The second** item will be played-back after that, starting at the time, set for **the first** item.

⊙ **The first conflicting event will have its start time removed** – this will remove the fixed start time of **the first** item and it will be played-back in its turn, as an ordinary clip in the playlist. **The second** item will be played **after the first** one and will preserve its fixed start time.

⊙ **The second conflicting event will have its start time removed** – this will remove the start time of **the second** item and it will be played back in its turn, as an ordinary clip in the playlist (after the first item). **The first** item will be played back as fixed.

> **NOTE:** The playlist is protected against engendering conflicting situations when inserting <u>fixed-time events</u>. You will not be allowed to set a fixed start time for an item, if there is an overlapping fixed-time already inserted in the playlist. If you try to enter a conflicting start time, the *Use fixed start time* field will become red, and pressing **OK** will invoke a warning dialog.

Conflicting situations could also be caused by applying offset values to these items.

Default behavior when overlapping preceding events

When using fixed-time events in the playlist, one can always come across some conflicts with the "ordinary content" (i.e., without fixed start times). There can be two major conflict types:

1.) The content to be played until the fixed start time is more than the time allowed (*for example*, there are 15 minutes of clips and only 10 minutes until the fixed time) – **so there is overlapping**;

2.) There is not enough content to be shown until the fixed start time (say there are 5 more minutes of clips in the playlist before the fixed time row, and the fixed time itself will go on-air after 15 minutes) – **a gap is formed**.

It is up to you to precise the rule for automatic settling of this conflict.

The default behavior when overlapping preceding events could be one of the following:

⊙ **Skip playing event and do an auto-fill** – all preceding clips that cannot be played as a whole (i.e., from the beginning until the end) before the beginning of the fixed-time clip will be skipped. A gap will appear in the playlist as a result of this skipping. This gap will be filled in accordance with the auto-fill rules, set by you (see the Default gap-filling category below).

Let us assume that there are five minutes remaining until the fixed time and you have two clips to go (before the fixed time), both of which are 3 minutes long. In this case **AirBox** will not be able to play the whole second clip, so this clip will be skipped, and an auto-fill event (category or default clip) will be played-back instead.

⊙ **Truncate playing event** – **AirBox** will playout as much of the preceding clip as possible, and when the time comes, it will cut to the fixed-time event (see also default start tolerance below).

Default gap-filling category

Default gap-filling category is the third section of the ***Fixed-time event options*** dialog and it contains several options for automatic filling of gaps in the playlist:

⊙ **Default clip** – The default auto-fill clip must be an MPEG2 (if you are using an MPEG2 plug-in) or DV (if you are using a DV plug-in) file. Most often it is a program logo, animation, etc. The default auto-fill clip will be played-back each time there is a gap in your playlist and you have not specified anything else to be done in order to fill it (you have not set any auto-fill category, the clips from this category are missing for some reason, etc.).

In brief, the default auto-fill clip is always there in case something goes wrong with the other filling content.

You can specify this file in the **Settings menu➜Auto-fill** dialog.

⊙ **Category from previous event** – sets the auto-fill category of the last played fixed-time event in the playlist (if such an event exists).

⊙ **Category from fixed-time event** – uses the category from the current fixed-time event (this corresponds to the auto-fill setting in the ***Clip Properties dialog*** of the fixed-time item).

⊙ **Custom category** – a sequence of clips created by you in the ***Auto-fill options*** dialog, which will be played-back if a gap appears.

<u>Default start tolerance</u>

In case you could accept some later or earlier start of your fixed-time events, specify the exact parameters here. Thus, if the time until the end of the previous clip is within the allowable *later start*, it will be played out without being truncated. Again, if it finishes earlier, within the allowable *earlier period*, the fixed time clip will start playing without the need to fill-in a several seconds' gap. This is NOT a fixed offset!

Example: Let us assume that there are two items in the playlist – an "ordinary" clip, followed by a fixed-time clip. The first one finishes at 11.00.20, and the second one must start at 11.00.00. This is a time-conflict situation. However, if the fixed-time has set a later start tolerance of 30 seconds for example, *AirBox* will play the first item until its end and the fixed-time will start at 11.00.20.

To recapitulate, later tolerances allow playback of the previous item to the end, and earlier tolerances allow avoiding short gaps (if possible).

<u>Start delay compensation</u>

It allows for setting offsets to a fixed-time event to compensate the time needed for reaction of the hardware. The offsets may be:

- *During play or pause mode-* tries to compensate for the time (in milliseconds), necessary for executing commands like **Jump**, **Next**, and **Resume**.

- *During stop mode* – tries to compensate for the time, necessary for executing the **Play** command.

NOTE: These settings are hardware-specific, so their exact values have to be determined experimentally.

WARNING! When using fixed-time events, the playlist should start and end within the same calendar day (i.e., the playlist should not go beyond midnight). Otherwise, we cannot guarantee the proper operation of the fixed-time logic and that of the daily playlist logic.

Complex Items
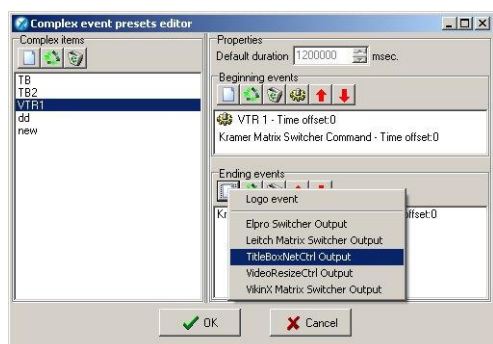
Two main ideas stand behind the complex events:

1.) Avoiding insertion of numerous rows in the playlist when there are several external events that have to be executed at the same time (or one after another):

For example, when you need to run a program off-tape, you will have to insert a VTR event, plus two switcher events, one for cutting from *AirBox* video to VTR output, and the second one for performing the opposite. All of these can be part of a complex event, so basically, you have a single complex event, named "VTR1," for example, which would take care of sending signals to the matrix and to the VTR, when necessary;

Channel in a Box User Manual

2.) Enabling creation of such complex events' presets, so the user will not have to insert the same external events one by one each time he needs them. Instead, he could pick up one of the available presets from the list he has created.

For example, if you have a matrix switcher, which needs to route IN5 to OUT2 in order to get *AirBox* on-air, and it needs to route IN2 to OUT2 in order to get live feed pass-through, you could predefine 2 complex events with custom names, such as "To AirBox" and "To Live". These complex events will contain the matrix switcher event with its IN/OUT configuration, as defined. Then you just need to insert the complex event by name.

So basically, a complex event represents a set of external events with their adjustments and offsets.

The *Complex event presets editor* window is divided in two sections:

The *Complex items* field to the left contains a list of available presets and several buttons for preset management.

To create a new preset, press the ***Create new complex item*** button and enter its name in the *New Complex Item* dialog. You can edit this name later, by pressing the ***Edit*** button. To delete a Complex event from the list, press the ***Bin*** button.

The *Properties* field in the right contains the "essence" of the complex event - a sequence of *Beginning events* and *Ending events*.

You can add beginning or ending events by pressing the ***Add new external event*** button in the Beginning/Ending events section respectively. Pushing these buttons invokes a drop-down list of all enabled external events.

NOTE: To have any external event available in the list, make sure it is enabled in *AirBox* **Settings** menu ➔ **Modules** ⇨ **Remote control** tab.

Each time an event is being added, its settings dialog opens for you to make the relevant adjustments (they can be edited further on) and specify its offset (if necessary). The events are executed in accordance with their offsets and regardless of their order in the Beginning/Ending events list.

NOTE: The beginning events can have earlier and/or later offsets, while the ending events can have only earlier offsets.

(!) TIP: If you need to make some fine-tuning of a certain external event when inserting a *Complex item* in the playlist, select it and press the ***Gear-wheel*** button above it. Thus, each time you insert this complex item in the playlist, the settings dialog of this event (i.e., a switcher configuration dialog) will open for you to make the final touches. These changes will not be saved in the original complex event; they will be valid only for the current insertion.

(!) TIP: If you want to change the events' order, select an event from the list and press the red ***Up/Down*** arrows to move it.

NOTE: If you have set the Default duration of the complex event, its execution will result in holding the *AirBox* playback during the event (as in a wait event). Besides, the offsets of the included external events will be calculated in relation to the complex event's start and stop points. Please, refer to the following *example*:

Let us assume that we have created a complex event with duration of 15 seconds. It includes 2 beginning events:

Event A with 3 seconds **earlier** offset, and

Event B with 2 seconds **later** offset,

As well as an Ending event (C) with 5 seconds **earlier** offset.

The Complex event, described above, will be executed as follows:

AirBox plays the clip, situated before the complex event in the playlist. 3 seconds before its end, Event A will be executed. 3 seconds later this clip will end, and a 15-seconds hold of the playback will start (the complex event duration has started). 2 seconds later, Event B will be executed. Event C will be executed after 8 seconds (5 seconds before the end of the complex event).

The next clip in the playlist will start playing 5 seconds later (the 15-second duration of the complex event has expired).

NOTE: When the complex event duration is zero, the included events' offsets will be calculated in relation to the surrounding clips. The **AirBox** playlist and playback will not be paused. Please, refer to the following *example*:

Let us assume that we have created the same complex event, but with zero duration: **AirBox** is playing the clip situated before the complex event in the playlist. 5 seconds before its end, Event C will be executed. Two seconds later (3 seconds before the clip's end), Event A will be executed. Three seconds later the clip ends and the next clip in the playlists starts playing. Two seconds after its beginning, Event B will be executed.

Graphic Rules
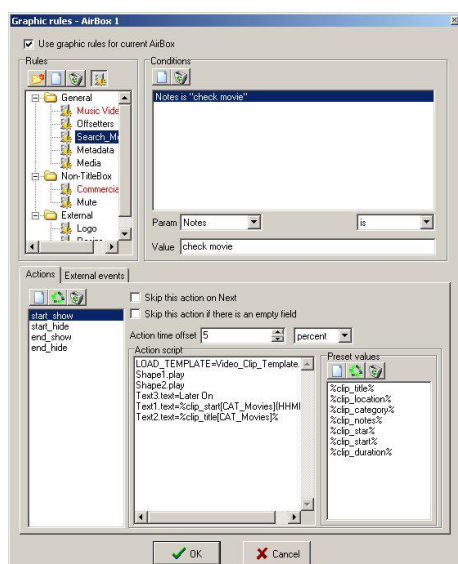
The information contained in the **AirBox** playlist can be displayed automatically over the video output through the **TitleBox Net Control** functionality. The main application of this feature is for music channels, to display the performers' names and song titles. .

First of all, you have to create a **TitleBox** template, with the objects that will display the information, contained in the playlist. In the example below we have created TB_Music.tmpl. It contains a background picture (we called this object Background); a text object for the performer's name (called Star), and a text object for the song title (called Title).

Once the template is created and available in the templates folder (see TitleBox NetControl description [above](#)), go to **AirBox Settings** menu ➔ **Graphic rules**.

The rules for extracting playlist information and displaying it over the video are set here.

The Graphic rules dialog contains three fields: *Rules*, *Conditions,* and *Actions*.

The *Rules* field contains a list of all available rule presets.

PLAY BOX
TECHNOLOGY

Rules are organized in Groups. You can create new groups by clicking on the **Folder** ![icon] icon in the *Rules* zone. By default, you will see a **[General]** group with an **[Example]** Rule in it.

The *Conditions* field contains the list of all necessary conditions that will enable executing the relevant actions below as per the currently selected Rule. In case you use Metadata (from the clip properties dialog) as a condition parameter, type its name in the cell to the right and then specify the operator.

In the *Actions* field there are three lists and a couple of check-boxes. The list to the left contains all actions, related to the currently selected rule. The *Action Script* list represents the commands that will be sent to **TitleBox** during the currently selected Action.

The *Preset values* list contains the variables that can be added to the script and are used most often.

For better understanding of the **Graphic rules**, please, follow the example below.

To create a new rule, select the group folder, to which it should belong, and press the **New rule** ![icon] button in the *Rules* field. Enter the name of your template and press **OK**.
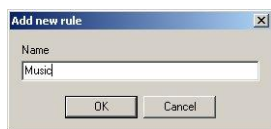
Then, go to the *Conditions* field to the right and set the conditions for executing **Actions** (we will set them later). In the example above, we have set the *Parameter* to **[Category]**, the *Operator* to **[is]**, and the *Value* to **[Music]**. Thus, when the condition **Category is Music** is fulfilled, the **Actions** below will be executed.

To add a new action, press the **New Action** ![icon] button in the **Actions** tab, and enter its name (**[Clip_Start_Show]** in our example), then press **OK**. This action will show the selected information from the play upon clip start.

In the *Action time offset* spin-box, specify the desired offset for execution. It can be in percentage from the clip duration or in milliseconds (select this in the spin-box to the right)

Finally, carefully describe the action in the *Action script* field:

[LOAD_TEMPLATE=TB_Music.tmpl] – This line describes which template will be used to show the text from the playlist.

[Background.play] – a **Play** Command for the picture object **Background**.

[Title.text=%clip_title%] – This line contains the name of the **TitleBox** object **[Title]** and a description of the text that it should contain (as you see, in this case we will extract information from the **clip_title** column of the relevant playlist entry).

You can drag-n-drop the *Preset Values* from the list to the right.

[Star.text=%clip_star%] – similar to the above, here we have described which cell contains the information to be displayed in the **[Star]** text object.

AirBox sends out this command to **TitleBox** through the net control option and the clip title and the performer name are displayed.

Now we have to send out another command to hide the graphics. We will call it **Clip_Start_Hide**:

[LOAD_TEMPLATE=TB_Music.tmpl] – point out which template is addressed.

[Background.stop] – A **Stop** command for the **Background** object.

[Title.stop] – A **Stop** command for the **Title** text object.

[Star.stop] – A **Stop** command for the **Star** text object

> **WARNING!** All entries in the *Action script* field are case-sensitive, so enter them **exactly** as they appear in the *TitleBox* template project, including the name of the template itself.

In the example in the screenshot above, we have set four actions to be executed when the Category is Music:

Clip_Start_Show – to show the clip title and performer when 5% of the clip duration has passed.

Clip_Start_Hide – to hide the title and the performer when 20% of the clip have passed.

Clip_End_Show – same commands as in **Clip_Start_Show**, but with 80% *Action time offset* to show the title and the performer at the end of the clip.

Clip_End_Hide – same commands as in **Clip_Start_Hide**, but with 95% *Action time offset*, to hide the graphics when 95 percent of the clip has passed.

You can add more variables from the Metadata tab in the clip's properties dialog.

The preset value format for the metadata is **[%metadata_MetadataName%]**, where **MetadataName** is the same as in the *Metadata* tab of the clip properties dialog (in the first column).Thus, if the metadata name is Producer, the preset value script will be **[%metadata_Producer%]**. Now you can create an Action script that will send this information to *TitleBox* (for example, **[Producer.txt=%metadata_Producer%]**).

Metadata categories can also be used for changing the file links in picture and sound objects in *TitleBox*. Thus, you could control which picture(s)/sound(s) should appear over each video clip.

Let us assume that in one of the rows in the *Metadata* tab we have typed **[new picture] [E:\Pictures\pic\A Fantastic Voyage.jpg]**.

With this example action script, *AirBox* will send a command to *TitleBox* to:

1. **Load** the project called **[template.tmpl]**;

2. **Play** the object called [picture] and replace the currently loaded picture file with the one, specified in the *Metadata* tab of the currently playing clip, under metadata category **[new picture]**:

[LOAD_TEMPLATE=template.tmpl]

[picture.MEDIA=%METADATA_new picture%]

Where:

[picture.] is the name of the object that we want to control

[MEDIA] means that this command contains information about a new media file to be loaded in this object.

[%METADATA_new picture%] – This is the place, from which *AirBox* should "read" the new file path for the media file to be loaded. Again, the format is **%METADATA_MetadataName%**, where **MetadataName** is the same as in the *Metadata* tab of the clip properties dialog (in the first column).

The same action script applies to sound objects:

[LOAD_TEMPLATE=template.tmpl]

[sound.MEDIA=%METADATA_new sound%]

Thus, *TitleBox* will **load** project **[template.tmpl]**, **play** the object called **[sound]**, and **replace** the currently loaded file with the one, specified in the *Metadata* tab of the currently playing clip, under metadata category [new sound].

> **NOTE:** The Metadata name is case sensitive! You must type it exactly as it appears in the file properties dialog!

The same action script applies to directX objects:

diectX.MEDIA=%METADATA_value%,

where **[directX]** is the name of the object that we want to control;

[MEDIA] means that this command contains information about a new media file to be loaded in this object.

[%METADATA_value%] – this is a name of **Metadata** which contains the path to the clip.

or

diectX.MEDIA=%clip_location%,

where **[%clip_location%]** is a clip location taken from "location" property of the clip in the playlist.

You can display information about the **title of an upcoming clip** in the playlist. The preset value descriptor has the following format: **%clip_title[+n]%**. Afterwards, create a script to send this information to *TitleBox*: **Next.text=%clip_title[+1]%**. Where Next is the name of the text object in *TitleBox* and **[+1]** is the index off-setter (to show the title of the following clip).

Moreover, a bookmark can be used as a reference for presenting information about an item in the playlist. Let us assume that the **Bookmark** name is **Test1** and you want to display information about the title of the following clip. Then the descriptor value should be **%clip_title[BM_Test1]%.**

To display the **start time of an upcoming clip**, type **%clip_start[+n]{HHMMSS}%**. The command, sent to *TitleBox* could be **Start.text=%clip_start[+3]{HHMM}%**.

> **NOTE:** If you are using daily playlists, the number **[+n]** for the next file could correspond to some file in the next day's playlist.

> *For example,* if you have: **%clip_title[+100]%**, and the current playlist ends after 90 lines, *AirBox* will continue to count the items from the next day's playlist until it reaches 100 (i.e., 10 more items). The number of playlists ahead, in which *AirBox* will continue searching, is five (5), by default.

If you want to change this number, please, consult with our *PlayBox* support team (support@playboxtechnology.com) for instructions.

To show the start time and the title of an up-coming clip, create the following command:

Next.text=%clip_start[+4]{HHMM}% - %clip_title[+4]%, where Next is the name of the text object in *TitleBox*, **[+4]** is the off-setter, {HHMM} is the time format.

The time format can also be {HH} or {HHMMSS} or {HHMMSSFF}.

As in NTSC mode, the start time appears as a drop-frame timecode in the playlist, it would be more convenient to display the start time according to the system time on the PC. There are two options for the system time script: {T} – for short system time (according to the Regional settings) and {TT} – for long system time (according to the Regional settings). Thus, instead of **Next.text=%clip_start{HHMM}%**, you should type **Next.text=%clip_start{T}%.**

Now you can show information about upcoming clips that belong to a certain category. The script format is **[CAT_Category Name[±index]]**.

Title.text=%clip_title[CAT_Movies]% – %clip_start[CAT_Movies]%.

In the example above, the content to be displayed in TitleBox object Title is: the title of the next clip down the playlist that belongs to Category Movies, then a dash, and then the start time of this clip.

Text1.text=%clip_title[CAT_Movies+1]% – %clip_start[CAT_Movies+1]%

In the example above, the content to be displayed in *TitleBox* object Text1 is the title of the second clip down the playlist that belongs to Category Movies, then a dash, and then the start time of this clip.

> **IMPORTANT:** There should be no plus or minus signs in the category name!

Instead of **Category**, you could use the **Star** as an index modifier to show information about upcoming clips in the playlist. Thus, the scrip will look like this:

Text1.text=%clip_start[STAR_Madonna]% - %clip_title[STAR_Madonna]% - to show the stat time and title of the next clip in the playlist that has [Madonna] assigned as **Star**.

   OR

Text1.text=%clip_start[STAR_Madonna+1]% - %clip_title[STAR_Madonna+1]% - to show the title of the second clip down the playlist that has [Madonna] assigned in the **Star** column.

> **IMPORTANT:** Again, there should be no plus or minus signs in the Star name!

Besides **Category** and **Star**, you could use a clips' *Metadata* as an index modifier to display information about upcoming events. In such cases, the script should look like this:

Text1.text=%clip_start[METADATA_Show]% - %clip_title[METADATA_Show]%. Thus, *AirBox* will display information about the next clip in the playlist that has Metadata called Show in its properties (in the first column of the Metadata tab).

   OR

Text1.text=%clip_start[METADATA_Show+1]% - %clip_title[METADATA_Show+1]%. Thus, you will display information about the second clip down the playlist that has Metadata Show in its properties. Replacing "+1" with "+2" will display information about the third clip down the playlist that has Metadata Show in its properties, and so on.

> **NOTE:** Even if there is no value for this metadata (in the second column of the *Metadata* tab), it will be considered valid and *AirBox* will display information about that clip.
>
> **IMPORTANT:** There should be no plus or minus signs in the Metadata name!

Special actions in the *Graphic Rules* can control events, different from graphics events.

Thus, you can control the logo insertion for example. The commands can be assigned as separate actions, or as lines in other graphic rules actions.

The beginning of such an Action script is marked with an exclamation mark {!}. The script must end with an exclamation mark too {!}.

Inside the script, there can be three types of commands: **LOGO_OFF** (to stop showing the logo), **LOGO_ON** (to show the last used logo), or **SHOW_LOGO_PRESET_1...16** (to show one of the 16 logo presets as specified in **Settings** menu ➔Logo).

Here is an example of a command to stop the logo: **{!LOGO_OFF!}**

IMPORTANT: The logo script is case-sensitive, i.e., it must be typed in capital locks!

Another couple of actions can switch the incoming triggers ON and OFF. Thus, the received GPI In pulses, DTMF tones, or Time Code will be ignored, and the commands assigned to them will not be executed during a certain clip.

These actions are called AUTOMATION actions. The possible commands are:

{!AUTOMATION_OFF!} – to disable the incoming GPI/DTMF triggers.

{!AUTOMATION_ON!} – to enable the incoming triggers.

WARNING! Do not use automation rules simultaneously with skip zones (**Settings** menu ➔**General**⇨Skip zones) to avoid conflicting logics!

TIP! If you need to switch the **AUTOMATION ON/OFF** manually, use the ***Automation*** button in the main *AirBox* window. If you need to switch **OFF** the **Audio output** of certain clips, you can use the following Action script: **{!MUTE_ON!}**. Type **{!MUTE_OFF!}** in the *Action* script to switch the audio back **ON**.

(!) TIP: If you are using the **Next** command in *AirBox*, all graphic rules that would have been executed in normal playback, will be executed at once. This might cause an unpleasant flickering of your graphics. To avoid this, check ☑*Skip this action on Next*.

(!) TIP: If some of the information that should be displayed in the graphics object is missing from the playlist, you can skip sending the whole command by checking ☑*Skip this action if there is an empty field*.

(!) TIP: To enable/disable a ***Graphic rule*** or an entire Rules group, select it and press the ***Enable/disable*** 🖼 button. If you want to disable the use of all graphic rules, just un-check ☐*Use graphic rules for current AirBox* at the top of the ***Graphic rules*** window.

(!) TIP: You can rename a group or a rule by clicking on it while it is selected. To move rules from one group to another, just drag-n-drop them.

If you need to assign some events to be executed instead of the above actions, you can do that in the ***External events*** tab.

Like in other set-up dialogs, you can choose the event type after pressing the ***New event*** 🖼 button.

The event settings dialog varies according to the event type you select.

Please, check the events' descriptions in the Edit menu section above.

NOTE: the list of available events will contain only the plug-ins that have been enabled in **Settings** menu ➔**Modules**⇨***Remote Control***.


Please, note that by default, all events will be executed simultaneously. If you want to execute some of them later, please, specify the required positive offsets.

NOTE: Negative offsets will not be executed!

IMPORTANT: Please, make sure not to overlap later offsets with other clips' graphics rules!

NOTE: To view a list of the commands, used in Graphic Rules, please, refer to Appendix 2 <u>below</u>

Filename Parser
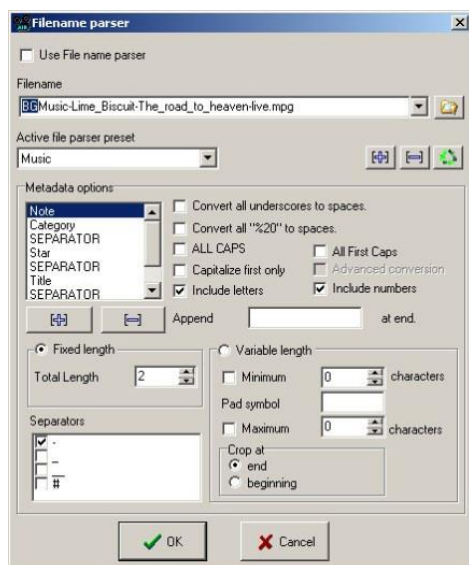
This module will allow **AirBox** to Parse names of files and get information from them. It provides an easy way to use information, included in the file name, without using a database. You can make the most of this feature if you have a standard naming structure for your clips.

In this setting dialog box, you have to "tell" **AirBox** how you name files, so it will "know" what certain parts of your filenames mean. The dialog box is divided into three zones – the uppermost zone provides options to write/browse for a sample filename and to set general "parsing rules" – presets; in the middle zone you should "explain" the naming structure by including metadata fields; and in the lower zone you can specify the properties of each metadata field.

For clarification, please, have a look at the following example:

Let us pick a sample filename, like **BGMusic-Lime_Biscuit-The_road_to_heaven-live.mpg.** Enter the latter name in the *Filename* field. You could also browse for existing files or choose from the drop-down list of filenames after pushing the arrow button to the right of the *Filename* field. The drop-down list contains all filenames of the currently loaded playlist.

This is a possible way to name music files – **create a preset** (let us call it **Music**):

Push the **Plus** button, situated to the right of the *Active file parser preset* field ░░░░░. A dialog will prompt you to name the new preset. You can rename it later on by pushing the **Recycle** 🔄 button, or delete it by pushing the **Minus** 🔲 button.
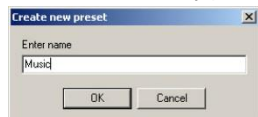
   Once you have entered the preset name, you can start "explaining" the naming rules. Let us go back to the sample filename – you can see that its structure contains, in order of appearance, an abbreviation (**BG**), a category name (**Music**), a separator (**-**), a name of a performer/star (**Lime_Biscuit**), another separator (**-**), a title (**The_road_to_heaven**), one more separator (**-**), and a note (**live**), followed, of course, by the file format. This is what you have to "tell" **AirBox**. Here is how:

In the Metadata options area, use the Plus ⊞ and Minus ⊟ buttons to add/remove parser fields. You can choose from the available types of fields. In our example, you have to enter one by one: **Note** (it will stand for the abbreviation **BG**), **Category**, **SEPARATOR**, **Star**, **SEPARATOR**, **Title**, **SEPARATOR**, and **Note**. If you want to skip some information in the filename, select **NONE** in the relevant position. Do not forget to set which symbols are regarded to as separators (check the relevant boxes in the lower left corner). Here, exclude the lower dash from the separators list, as it represents the space within the separate fields.
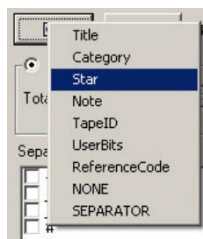
To the right of the *Metadata* fields list you can see a number of checkboxes that provide some conversion options:

☑ *Convert all underscores to spaces* - in our case **Lime_Biscuit** will become **Lime Biscuit**.

☑ *Convert all "%20" to spaces* is not applicable in our case, but otherwise you can use it for downloaded files (their names often happen to contain "%20" instead of spaces)

☑ *ALL CAPS* will turn all letters in the filename to upper case.

☑ *All first caps* – capitalizes the first letter of each word in the relevant metadata field

☑ *Capitalize first only* – capitalizes only the first letter of the relevant metadata field

☑ *Include letters* and☑*Include numbers* are checked by default. If you uncheck some of them, the filename parser will ignore the relevant characters (i.e. will not include them in the field).

You have to set your preferences for each metadata field separately (select it by clicking on it).

Finally, you have to specify the length of each metadata field.

This is not a problem if you choose to name your files with fixed length for each field – ⊙check the *Fixed length* flag and specify the number of characters using the arrows. The corresponding characters in the *Filename* field will be highlighted in blue, so that you can see your setting.

However, setting variable lengths is a bit more complicated. ⊙ Check the *Variable length* flag in the lower right field and then specify minimum and/or maximum characters to be included in the relevant metadata field. If you have set a *Minimum* value, but the relevant metadata field contains fewer characters, you will need some *Pad symbol* to fill-in the gab.

If you have set a *Maximum* value, but the relevant metadata field contains more characters, you will have to insert a *NONE* field before the *SEPARATOR* field, thus, telling **AirBox** to ignore the remaining symbols to the separator.

Back to our example, **Lime_Biscuit**contains 12 symbols. If we set a minimum value of15 symbols and a Pad symbol (*) for the [Star] field, the Filename Parser will display **Lime Biscuit***** in the playlist grid. If we set a maximum of 9 symbols for the [Star] field, the *Metadata* fields list should contain "… [Star]*,* [NONE]*,* [SEPARATOR]…," instead of "… [Star], [SEPARATOR]…". You can change the positions of the metadata fields by drag-n-dropping them.

Append … at end field gives an opportunity to add characters to the end of a metadata field.

Some fields, like [Title], [Category], etc., are displayed directly in the **AirBox** grid.

Others, like [Tape ID] for example, may provide information to **SubTitle Plus** (www.subtitleplus.com) or **_SubTitleBox_** (see below) for proper display of the corresponding subtitles. When you use the filename parser to display subtitles, you have to create a preset with naming structure [Tape ID] and [SEPARATOR] and the fields' length should be set to *Variable*. In the clip's properties dialog fill in the same *Tape ID* as that in the subtitle file.

Channel in a Box User Manual

NOTE: In order to use the features of the *Filename parser* module, you must set it first, and then add files to the playlist. THE MODULE CANNOT PARSE AN ALREADY LOADED PLAYLIST, since it already contains all the metadata for the relevant clips, included in it.

WARNING! If, in the newly-loaded playlist, there is any information in the fields that is also used by the *Filename Parser* preset, this information will be overwritten!

Admin

A new menu item, the **Admin…** aims at improving the security of *AirBox*. The idea for password protection of some settings that might be crucial to playback performance is already a fact.

The first time you enter this menu item, you will have to go to the *Change password* row. Clicking on it will open a dialog box, containing three cells – *Old password*, *New password,* and *Confirm password*. As this is the first time you enter this menu item, leave the first cell empty, then write your password twice – in the second and in the third cells (your password can contain up to 256 symbols). Click **OK**.

WARNING! Make sure not to forget the password!

Now you are already logged on. In order to restrict the access of all other users to the **Settings** menu, you just have to log off. Almost all menu items will become inaccessible – at any attempt to enter, a password will be required.

The only exceptions are **Colors** (as they will not affect the playback); *Enable* SubTitleBox (but not *Configure*); and Mirror mode (*Full mode*, *Idle mode*, *Change dongles*, also without configuration options access).

If you decide not to use password protection any more, just go to **Change password**, and enter the old password in the relevant field. Leave the *New password* and *Confirm password* fields empty and click **OK.**

Save/Load Settings

If you need to configure several playout servers in the same way, you can use these two options to copy settings and apply them to other machines. Thus, you will avoid configuring all your machines one by one.

You can export the settings of all *AirBox* channels on the machine, or you can only copy those of the currently open channel.

To export the settings of the current *AirBox* instance, go to **Settings** menu ➔**Save config**, and select the **AirBox #...** line. The resulting file has extension *.pb1*, which means that it contains settings for only one *AirBox* channel.

WARNING! Please, make sure to name the channel-setting file after the instance name, so that you know for sure which channel's settings are contained in it. Later, when you try to load the *.pb1* file on another system, *AirBox* will not be able to distinguish which channel's settings you load; therefore you need to recognize this from the filename.

To export the settings of all *AirBox* channels, select All *AirBox* instances. The resulting file has extension *.pb0*. Later, when you load it on another machine, it will affect all *AirBox* channels available there

Live Inputs

Opens the Live Inputs editor.
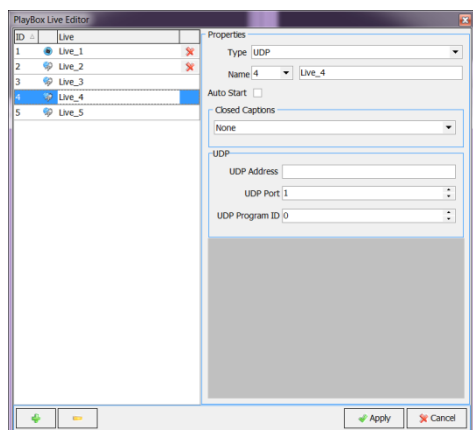
When you are using **AirBox** with a live input device, you have different configurations in the **Live Inputs** tab.

The **Live Inputs** dialog allows you to add as many live inputs as you wish and edit their settings.

The **Live Inputs** dialog contains a list of all available live inputs. They are designated by a Live name and an ID number. Clicking on the *Details* field to the right contains information about each separate input. In order to see the specifications about a given input, just select it from the list.

When you select a Live Input, the right area of the **PlayBox Live Editor** will show you its configuration properties. Two different types of inputs can be configured: CaptureDevice and UDP. Note that you can change the type of each separate input by selecting the desired option from the corresponding Type drop-down list.

The **CaptureDevice** properties are shown to the right. Set its ID number and *Name* from the corresponding field and ☑Enable / ☐Disable the ☑*Auto Start* option, which allows the input to be automatically initiated once it is set.
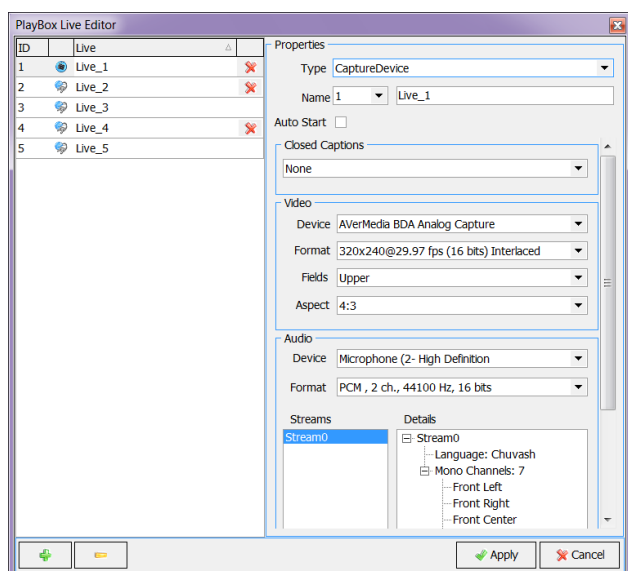
If you would like to capture closed captions, select the *Closed Captions* type to be captured from the drop-down menu – **Analog** or **Digital**. If you select **None**, closed captions will not be captured.

Further down the dialog select the video *Device*, used for this input, its *Format*, *Fields* order, and *Aspect* ratio.

Channel in a Box User Manual

The *Audio* field also allows you to select a *Device* and *Format*. Furthermore, here you can add a number of streams, depending on your needs. Simply press the **Add** button to insert a new stream or the **Edit** button to change the settings of an already existing audio stream. The **Remove** button will delete the selected stream.
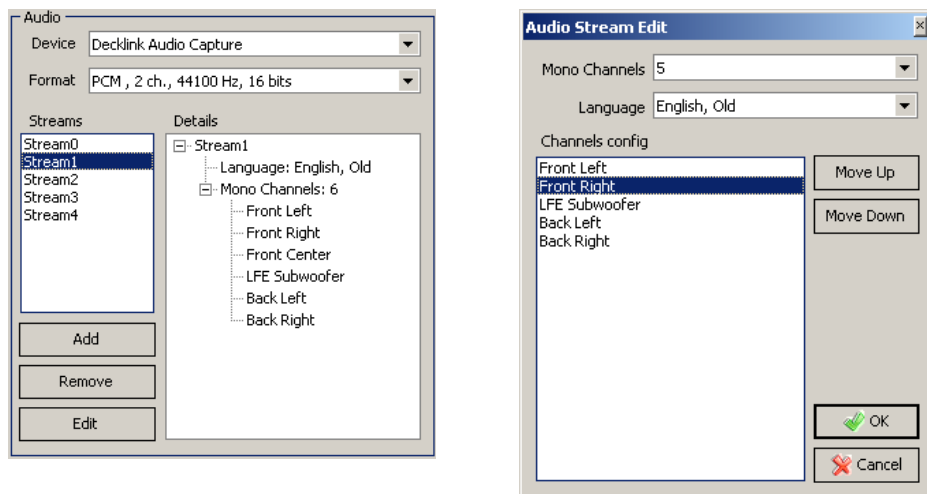
Pressing the **Add** / **Edit** button invokes the dialog to the right. Here you can select the number of *Mono Channels* and enter the language of the input.

If you select **UDP** for an input type, a dialog as one to the left will appear. Here you should set the *UDP Address*, *Port*, and *Program ID*.

The live inputs, configured in **AirBox**, are visible in the Multi AirBox Manager and in the PlayBox Live Inputs View.
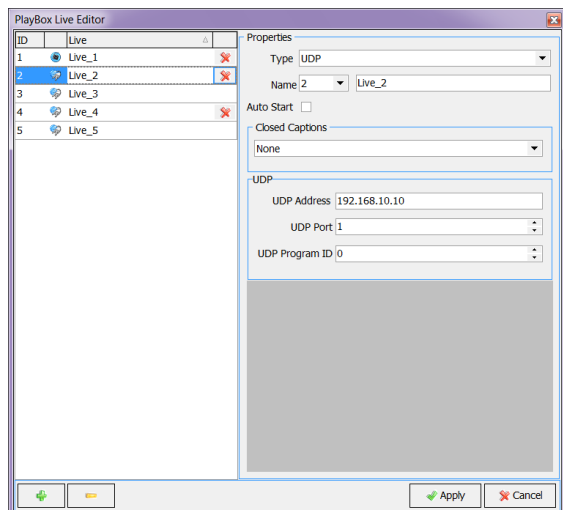
Tools Menu

This menu was created to accommodate some useful tools in **AirBox**.

Channel in a Box User Manual
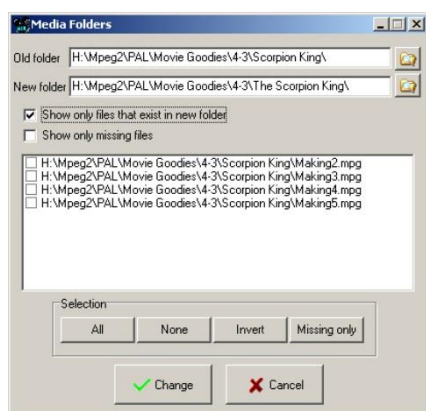
Change file path…

This module is intended for relocating file paths. If you have built a playlist and, for some reason, the file locations have been changed, here you can quickly relocate file paths. This can be done easily. Just fill in the *Old folder* field with the original location, and then fill in the *New folder* field with the new location. You may browse for path with the ![icon] button. There are two view options:

- If you check ☑*Show only files that exist in new folder*, only the files that are available in the *New folder* will be displayed in the list below;

- ☑*Show only missing files* displays only the files from the playlist, which are *Missing*.

The *Selection* modes are self-explanatory and give opportunity for fast selecting/deselecting of all items, inverse, and missing only selection.

When the **Change** button is pressed, the file paths are relocated, and this is reflected in the playlist immediately.

SubTitleBox
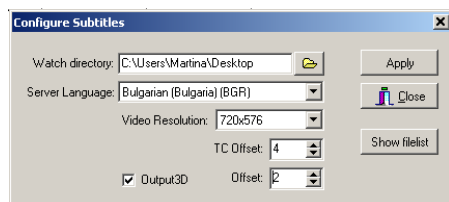
SubTitleBox is an optional plug-in for the **AirBox** module. It enables using subtitles together with your running clips. **SubTitleBox** does not provide any editing options; it just shows your subtitles synchronized with your clips. The supported subtitle formats are the ones, created by our own subtitling software **Subtitle Plus** – www.subtitleplus.com (*.sub*); Screen Subtitling/Win2020 (*.pac*); and EBU t3264 (*.stl*). It is best to use **SubTitleBox** with **Subtitle Plus** native files. They contain the *TapeID* (the reference to the corresponding media files), so **SubTitleBox** could "know" when to load and display the subtitles for each clip.

Enable the plug-in and configure it. The only thing you need to specify in the **Configure…** dialog box is the folder, where your subtitles are located (*Watch directory*), and the language you are using. Press the **Show filelist** button to view all the subtitle files, contained in the *Watch directory*:
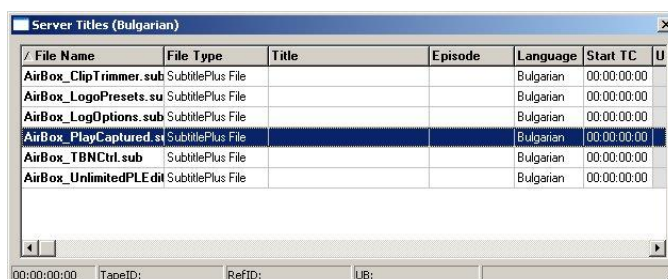
SubTitleBox will take care of broadcasting your subtitles in accordance to the playlist, loaded in *AirBox*.

> **NOTE:** Make sure the **Filename parser** is set to a preset with naming structure [Tape ID] and [SEPARATOR] before you load the playlist. Check the ⊙ ***Variable length*** radio button. In the clip's properties dialog, fill in the same *Tape ID* as of the subtitle file. Otherwise, *AirBox* will not display the subtitles.

Use the *TC Offset* spin-box to apply an offset to the subtitles displayed. This offset is measured in frames. You can also use negative numbers here.

Check the *Output3D* box if you want to display 3D subtitles. The Offset spin-box here is used for depth measuring. If you set a positive number, the subtitles will appear deeper in the screen and vice versa.

Mirror Mode
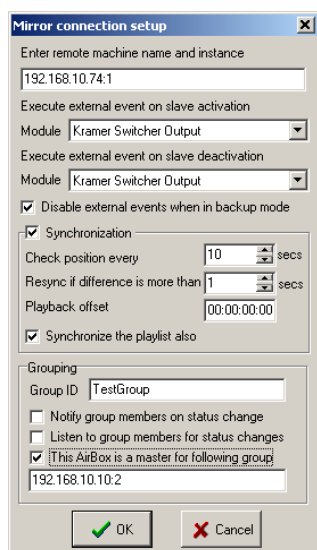
The **Mirror mode** is another optional plug-in, available for *AirBox*. The Mirror mode provides options for full redundancy of your main playout unit against system failures, such as power loss, broken motherboard or RAM, etc. In order to use this option, you must have two licenses for *AirBox*, running on two different workstations with a network connection between them. For the sake of convenience, these are called *Master* and *Slave* hereafter. They communicate via TCP/IP protocol.

<u>**Failure actions:**</u>

There are two possibilities to back up your *AirBox* in case an unrecoverable failure of the Master *AirBox* occurs (the master *AirBox* server is down for a long period, faulty motherboard, CPU, RAM, etc.):

   1. When there are two fully functional *AirBoxes* on both machines, plus one *AirBox backup* license on your *Slave* machine: In this scenario, you just have to uncheck the Mirror mode on the *Slave* machine and continue working with the full *AirBox* functionality.

   2. When there is one fully functional *AirBox* installed on the *Master* machine and one *AirBox backup* (with limited functionality) on the *Slave* machine: In this scenario you will have to attach the full *AirBox* dongle to the *SlaveAirBox* server and use the **Change Dongles** menu command in order to operate the full functionality on the *Slave* machine. Obviously, the master machine does not need a dongle while faulty.

> **NOTE:** When *AirBox backup* is a stand-alone license, *AirBox* will start in **Mirror Mode** automatically. No full functionality available!

To set the *Master*, start **AirBox** and check ☑ *Enable IP remote control* in **Settings ➔ General ⇨ _General_**. Otherwise, the *Slave* machine will not be able to connect to the *Master*.

To set the *Slave*, go to **Tools ➔ Mirror Mode ➔ Configure…,** and do the following settings in the **_Mirror connection setup_** window:

In the topmost string, *Enter remote machine name and instance*, write the Remote machine IP address or name.

The BackUp can execute some external event upon connecting/disconnecting to the *Master*. This is useful, for example, for auto switching a video switcher to another input/output when the *Master* unit fails, or when you switch back to it.

Execute external event on slave activation – this event will be executed when the current **AirBox** is switched to BackUp mode. For example, if the *Master* has been down for some time and now it is running OK, so you want to start the main playback from it again.

Execute External event on slave deactivation – this event will be executed when the *Master* fails and the current *BackUp* takes over the playback.

NOTE: For switcher control upon connecting/disconnecting, you need the PRO option, enabled on your dongle.

☑ *Disable external events when in backup mode* – Check this box if you want to stop external events' control when the backup AirBox is initiated.

The lower half of this window concerns the synchronization between the *Slave* machine and the *Master* machine. Do not forget to check the ☑ *Synchronization* box, if you need it.

Set the frequency of position enquiries to be sent by the *Slave* to the *Master* in the *Check position every…sec* cell (the least allowable is 10).

Below you can set the maximum allowable difference (in seconds) between the *Master* and the *Slave* playback. If the difference goes beyond this value, the *Slave* machine will have to resynchronize to the *Master*.

Playback offset value compensates the delay that may occur due to some additional factors (e.g., the network communication delay or backup playback reaction time).

Check ☑ *Synchronize the playlist also* if you want to check the playlists at each resynchronization. This is a "double insurance," in case some playlist data is lost during the IP communication.

It is possible to **group** several *Slave* machines to operate together. The *Group*-related settings are situated at the bottom of the **Setup** dialog.

The purpose of Grouping is described in the example below:

Let us assume that we have three **_Master AirBox_** channels. Their outputs are connected to one external device (let us call it **MasterOut**). The output of **MasterOut** sends the three signals together (as a Multiplexer would do).

There are three **_Slave AirBox_** machines, listening to the three *Master* machines. The outputs of the *Slaves* are connected to another external device, similar to that of the *Masters* (we will call this device **SlaveOut**).

The outputs of **MasterOut** and **SlaveOut** are connected to a switcher.

In case any of the masters fails (for example **_AirBox2_**); *Slave 2* will take over the playout but the **MasterOut** device will be outputting only the signals coming from **_AirBox1_** and **_AirBox3_**. The signal of *Slave 2* (that substitutes **_AirBox 2_**) will be output on the **SlaveOut** device. In order to have all signals output on one device, we have to start the playout on *Slave 1* and *Slave 3* and switch from **MasterOut** to **SlaveOut**.

Therefore, we have implemented the option for grouping *Slave* machines. Thus, in case one of the slaves in the group starts playing, it will "tell the others to start playing too, and the switcher will be switched automatically from **MasterOut** to **SlaveOut**. Now you have the three signals output together again.

Back in the **Mirror connection** dialog, there are three fields to setup *Grouping*:

Group ID – fill in the name of the group of *Slaves*. This name should be the same in all *Slave* machines that belong to this group, so you have to fill it in each *Slave*'s **Mirror connection** dialog.

☑ *Notify group members on status change* – enable this so the current *Slave* will report its status to all other *Slaves* in the group. If you do not want this *Slave* to control the whole group, leave this box unchecked.

☑ *Listen to group members for status changes* – check it if you want the current *Slave* to start/stop playing when any *Slave* in the group starts/stops. If you do not want this *Slave* to be affected by the status of other *Slaves* in the group, leave this box unchecked.

☑ *This AirBox is a master for following group* - check it to define a current **AirBox** as a master of the group.

The Mirror mode has two major sub-modes: **Full mode** and **Idle Mode**.

Full Mode:

The *Slave* and the *Master**AirBox*** are always playing the same content simultaneously.

Advantage: The *Slave* channel always runs in perfect synchronization and you can switch over to it any time.

Disadvantage: This will double the network traffic, since the two **AirBox** servers will be transferring data at the same time.

Idle Mode:

The *Slave**AirBox*** is remains idle and listens to the *Master**AirBox*** activity. If the *Master**AirBox*** stops responding, the *Slave**AirBox*** will start playing immediately from the same point, at which the *Master**AirBox*** was last.

Advantage: No additional network traffic overhead.

Disadvantage: It might take up to a second to start the playback process from the point it failed. The information about all previously executed external events will not be preserved. Thus, if the *Master* was running in video scale mode before the failure, the *Slave* will precede outputting full-screen video.

In both modes, whenever you change anything in the *Master**AirBox*** playlist, the *BackUp**AirBox*** does the same automatically.

Regardless of the mode, you will always use content files that are stored either on local or on network storage.

The easiest approach is to use content ONLY from a redundant network-attached storage (NAS) or SAN. This means that the *Slave**AirBox*** will use the same file path as the *Master**AirBox*** to locate the content files.

If you are going to use local content files, or a mixed approach, then you should install **SafeBox** (see the ***SafeBox*** section for details) on the *Slave**AirBox*** machine. Its task will be to automatically replicate (copy) the new-coming content from the *Master**AirBox***'s local storage to the local storage of the *BackUp**AirBox***.

The **Change dongles** option allows you to change the dongle (WIBU-key) of a currently playing unit without interrupting the work of the **AirBox**. Thus, if your *Master* machine is down and you need the full functionality on your *Slave* machine, you will be able to change the dongle for the latter without interrupting its playback. After you have inserted the new dongle, press **Tools➔Mirror Mode➔ Change Dongle,** and the new dongle will be recognized.

> **IMPORTANT!** The *BackUp* machine should run the same software version as the *Master* machine!

Channel in a Box User Manual

Playlist Checker
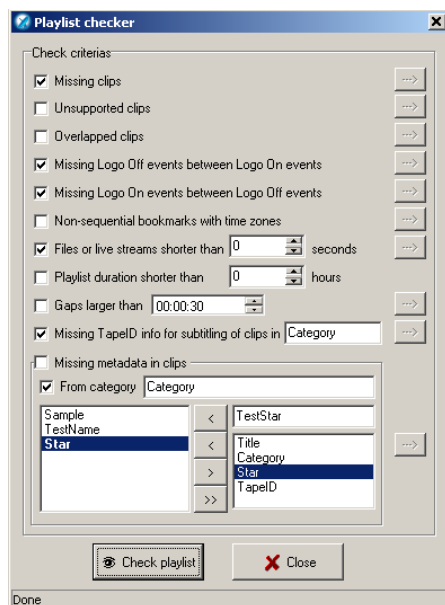
This tool checks the playlist for inconsistencies. Enable the criteria you need by checking the box in front of them.

Below you will find description of some of the checking criteria:

☑ *Missing clips* – checks for missing clips in the playlist

☑ *Unsupported clips* – checks for clips with an unsupported format in the playlist

☑ *Overlapped clips* – this check is related to <u>fixed-start time</u> clips that overlap previous clips in the playlist.

☑ *Missing Logo Off events between Logo On events* – checks if there is a Logo Off event between the Logo On events

☑ *Missing Logo On events between Logo Off events* – checks if there is a Logo On event between the Logo Off events

☑ *Non-sequential bookmarks with time zones:*

Bookmarks with time zones should be situated in sequential order, i.e., following the normal time flow. This means that bookmarks with earlier time zones should be situated up in the playlist, while bookmarks with later time zones should be placed down the playlist. This will ensure their correct execution.

☑ *Gaps larger than…* – when using fixed-start times and there is not enough content to be played before them, <u>gaps</u> are formed in the playlist.

☑ *Missing TapeID info for subtitling of clips in* [Category name] – <u>Subtitles</u> are shown based on the *TapeID* of the clip and the currently running time code. If a *Tape ID* is missing from a clip's properties, no subtitles will be displayed over it. Therefore, it is important to make sure that all *TapeID*s are in place. Just specify the category of clips that require subtitling and check if all of them have *TapeID*s in their properties.

☑ *Missing Metadata in Clips* – this field contains settings for verifying the availability of all metadata, usually needed for <u>Graphic rules</u>.

Metadata is described in the clip properties. Each metadata has a name and value, where the name represents a category, and the value represents the information about this category. Please, check the <u>Metadata description</u> above. The playlist checker uses the metadata names as reference and checks the presence of the corresponding metadata values.

☑ *From category* [Category name] – here you have to specify the category of the clips you want to check, like clips that belong to category [Music]

The list of metadata fields to be checked is situated to the left below. You can pick them from the list of presets to the right: Just select the Metadata name you need and press the [<] button to add it.

If the Metadata name is not available in the list of presets, create a custom name in the string above it, and press the [<] button to add it to the left.

To remove a metadata name from the check-list, select it from the list to the left, and press the [>] button.

To clear the whole check-list, press [>>].

If the *Check playlist* button finds an inconsistence, the relevant criterion turns red. The arrow to the right of it becomes active. Press it do view the list of inconsistencies, sorted by their position in the playlist.

Media Folders
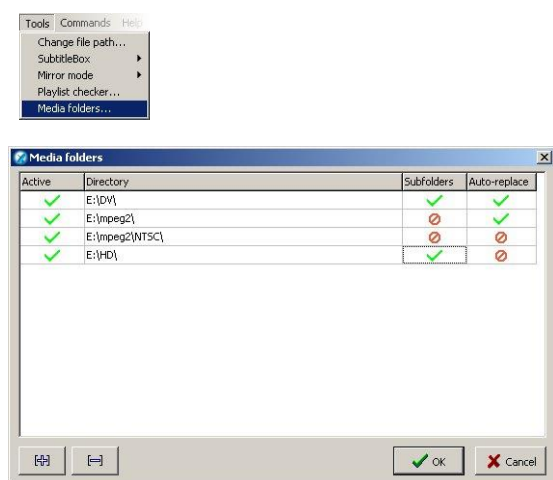
This tool was developed to allow for specifying watch folders that might contain media files. Thus, if there are missing files in a playlist, these folders will be automatically searched upon playlist loading.

NOTE: Media folders do not operate in run-time! You need to reload the playlist in order to check the watch directories.

To add a new watch directory, click on the plus sign and browse for it. As soon as it is inserted in the list, you can adjust its settings:

Double-click in the **Active** column to enable watching the folder, described in the line to the right.

If you want *AirBox* to parse the sub-directories of a selected watch folder, double-click on the **Subfolders** column to turn the closed sign into a green tick-mark.

Auto-replace – this function will update the file paths of the missing files upon saving the playlist. If not enabled, the new location of files will not be saved in the playlist.

Commands Menu

This menu contains commands, related to playback and logo presets. The available commands are:

Playback commands

This menu duplicates the playback control buttons, situated under the master counter (***Play/Stop/Pause/Next/Jump/Return***). The commands are executed just by clicking on the appropriate field. For the user's convenience, a relevant shortcut is written to the right of each command.

Logo presets

Here you can trigger logo presets by clicking on them. The preset numbers correspond to those in the **Settings** menu ➜ **Logo** dialog.

Jump to: This is just a shortcut – <Ctrl+Shift+#>. By using it, you can jump to a clip at your will. Just press and hold down <Ctrl+Shift>, enter the desired clip's number, and then release the <Ctrl+Shift>.

Help Menu

This menu contains useful information about the ***AirBox*** module and the possibilities for getting technical support from us.

PlayBox Help

Opens the ***AirBox*** context-sensitive help.

PlayBox Doctor

This module gives the opportunity to generate easy-to-complete problem reports. It is integrated in each ***PlayBox*** module. It can gather almost all the information, needed for the ***PlayBox*** support team in order to provide you with the prompt answers, without too many questions about your system configuration.

The Basic User's manual contains a detailed description of the PlayBox Doctor Report and other functionalities. If you do not have the Basic manual, you can download it from our website –

About…

Displays the **About** box of the ***AirBox*** module. It contains useful information, such as: module version, WIBU Box number, mode, registration, etc. The name of the currently selected platform is displayed at the bottom.

LISTBOX

GETTING STARTED

ListBox is dedicated to creating and editing **playlists**. It allows preparing playlists in advance for the **AirBox** module and playing them on-air later.

ListBox does not need any special workstation or platform.

Quick Start

- Launch **ListBox**;
- Right-click on the grid;
- Select **Add ➔ Media file;**
- Browse for the files you want to insert in the playlist and click on the **Open** button.
- Go to **File** menu ➔ **Save playlist as** and browse for the location to save your playlist to.
- Type some name for the playlist and press **Save**.

Congratulations! You have just created your first **PlayBox** playlist!

USER INTERFACE

The interface of **ListBox** is identical with the **AirBox** interface. It is designed like that for user's convenience. There are some differences, which are described in this section.

Playlist Control Buttons

These buttons provide access to the most commonly used playlist functions. All commands from the Toolbar can also be found in the Menu Bar.

Some particular buttons on the Toolbar will appear enabled or disabled, depending on the selection made.

The toolbar contains the following command buttons, shown in that order: **New** , **Open** , **Save**/**Save as** , **Save as daily** , **Append/Insert Element** , **Delete Selection** , **Clear Playlist** , **Move up/Move Down** , **Reset playlist** , **Randomize** , **Properties** , **Trim** , and **Undo** . Their functions are described below in the relevant menu sections.

Playback control and counters

The **Master Counter** shows the elapsed time of the current playout session. Pressing the **Stop** button resets the counter.

```
00:00:08:17
```

The **Clip counter** is situated under the master counter. It can operate in two modes: count-up and count-down mode. Just double-click on it to switch between them!

```
00:00:08:17
```

The playback control buttons are situated under the Clip counter:

- **Play** ▷ – starts the playback.

- **Stop** □ – terminates the playback.

- **Pause/Resume** ▯▯ – temporary interruption and resuming of the playback. The playback resumes from the same point on.

- **Next** ▷▏ – will stop the currently playing clip immediately and will switch to the next clip in playlist. This happens by a smooth cut without any sound or visual artifacts.

- **Return** ↻ is active only after a **Jump** has been executed. Use this button to return to the position before the last **Jump**.

- **Jump** ↻ – terminates the currently playing clip immediately and starts playing the selected one, regardless of its position in the grid.

- **Cue** 🕸 – This button will interrupt the playback, and will pause at the first frame of the selected clip. You can also activate it by holding down the **<Ctrl>** key and pressing the **Jump** button at the same time. Use this to prepare a clip for playback, while showing its first frame on the output. Since this is a machine-and-decoder-dependent function, you might have to increase the *Cue Delay* time in the **Settings** dialog box (**Settings ➔ General ⇨ General → Cue delay**).

- **Fix overlapping** 🗓 – press it to fix the overlapping issues in the playlist. The fixed time flags will be removed and the duration of the overlapping files will be truncated.

- **Fixed playlist start mode** 📑 – press it to see the start time of the clips in the playlist in relation to the start time of the playlist, which is defined in the **Current playlist** tab. Press the button again to see the start times of the clips in the playlist in relation the current system time.

Playlist Grid

This area is dedicated to playlist visualization. It looks the same as the *AirBox* *Playlist Grid*, but the grid headers are colored, so you can easily distinguish between *AirBox* and *ListBox.* The order of grid columns can be changed by drag-n-dropping them to the left or to the right.

You can control the columns to appear and their order in the grid from **Settings** menu ➔ **General** ➔ **Grid**.

Grid Features:

- The *Color Coding* in the grid provides additional information for the users:

  Dark Blue bar shows the currently selected clip –all actions, i.e., Move Up/Down are applied to it.

  Red text rows contain events. You can insert the same events as in *AirBox*.

  Red-colored rows represent clips, which cannot be found at the specified file location. A red minus appears in front of the clip position number.

- Drag-n-Drop – It allows dragging playlist items within the grid or from one grid to another (from *DataBox* or from/to *AirBox)*.

If you hold down the **<Ctrl>** key while dragging, the executed operation is **Copy.** If you just drag-n-drop, the operation is **Move**.

- **Multi-selection** – allows simultaneous manipulation of many clips– move, randomize, etc. Clips can be added to the selection by holding the **<Shift>** or **<Ctrl>** key. The **<Shift>** key selects from-to, while the **<Ctrl>** adds a single clip to the selection.

- **Double-clicking over a clip** invokes the **Clip Trimmer**. A detailed description of the **Clip Trimmer** can be found further in the manual in its corresponding chapter. Double-clicking over a missing clip invokes its properties dialog. If you want to view the properties of an existing clip, right-click in its row and select **Clip properties...** from the context menu.

- **Right-clicking** in the grid opens a context menu. It contains commands from the **Edit menu** and the **Playlist menu** that are described further in the manual.

- ▪ *Columns description:*

- **Start Time** – shows the start time of each clip.

- **Duration** – shows the actual duration of each clip. If a clip has been trimmed, its new actual duration is displayed in this column.

- **Type**– shows the type of the clip (*MPEG, or AVI DV*).

- **Category**– contains category information, fed by **DataBox**. The background is colored with the predefined category color.

- **File Name**– contains information about the file-paths and names. If you need to change the path of a certain file in **List Box**, just press **<F8>** to open the Browse dialog.

If a file is missing after the playlist is loaded in **AirBox**, its line will be skipped and the playout will continue with the next available clip.

- **Title**, **ClipID**, **Star**– contain data, fed by **DataBox**, describing the clip name, clip ID, and the performing artists. You can edit these fields manually in **ListBox**, in the clip properties dialog.

- **Notes** – displays trimming notification and data from the fields, defined in **DataBox➔Options⇨General➔***Fields to AirBox notes*. You can edit these notes in the clip properties dialog in **ListBox**.

> **(!) TIP:** You can choose the columns to be shown in the grid. Go to **Settings menu ➔General ➔ Grid** and check the ones you would like to see. If you would like to change the columns' order, rearrange them in this dialog, or simply drag their headers, while working in the playlist.

Status Bar

| Total duration: 00:46:58 (!) | Loop at: 15:08:55 (!) | Selection: 00:09:42 (!) | Mixed Soft Playback | |

The Status bar is located in the lowest part of the **ListBox** window.

Its first cell – *Total Length* – shows the playlist duration. If it is longer than 24 hours, the number of days will be displayed in brackets in front. The figure in parenthesis in the end shows the number of rows in the list. The Second field – *End at/Loop at* –shows when the playlist will end. If the playlist is longer than 24 hours, the number of days will be displayed in brackets in front. The third cell – *Selection Length* – represents the duration of the currently highlighted lines in the grid. The fourth cell displays the currently active plug-in.

> **NOTE:** **ListBox** is always running on the Software Mixed plug-in. As this plug-in supports virtually all file formats, please go to **Settings menu ➔ Modules ➔ Output,** and select the plug-in you are planning to use in **AirBox**. This will allow for correct file verification in **ListBox**.

MENU BAR

The *Menu Bar* is situated in the upper left end of the window and contains the same menus as **AirBox**. However, some of the options are disabled, as they are not relevant in playlist preparation. The following paragraphs contain description of the differences that appear in **ListBox**. Please, check the **AirBox** File menusection above for detailed information about the common settings.

Channel in a Box User Manual

File Menu

Export Capture list

If there are missing files in the playlist, you can create a list for **CaptureBox** to ingest them. Select **Export** capture list from the **File** menu and browse for the location to save the playlist to. The resulting capture lists (*.cap*) will be named after the currently loaded playlist.
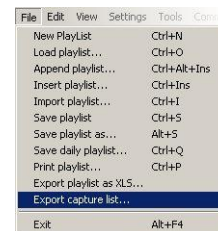
Export Logs to XLS

This File menu item appears in **AirBox**, but it is not visible in **ListBox**.

Reload Graphic Rules

This File menu item appears in **AirBox**, but it is not visible in **ListBox**.

Edit Menu

This menu contains commands, related to playlist editing. As it is identical to the **AirBox** Edit menu, please, check the relevant section above for details.

View menu

This menu is identical to the **AirBox** View menu. The only difference is that in **ListBox** you cannot view external preview window, while in **AirBox** you can.

Settings menu

As the **Settings** menu contains many playback-related options, here you will find most of the differences between the two modules.

General

In the **General** tab of this options dialog, there are several fields that are not accessible in **ListBox**:

☑ *Restart playback after* – this setting is always enabled.

☐ *Enable remote IP control* – This functionality is related to remote control of the playback. As **ListBox** is intended for playlist preparation, it is not needed here.

TC transmission options – as they are related to on-air subtitling, they are not needed in **ListBox**.

All the other settings in this dialog are the same as in **AirBox ➜ Settings** menu ➜ **General** ⇨ **General**.

Interface

In the **Interface** tab, there are two options that are always enabled and inaccessible:

Modules

AirBox Output tab – here you can find a list of presets that will be used for file verification while building your playlists in **ListBox**. All you have to do is select the platform and file format you intend to use for playout in **AirBox,** and double-click to assign it.

In the **Remote** tab you can enable the events plug-ins you intend to use in **AirBox** and configure them. If **AirBox** and **ListBox** are running on the same server, these settings will be shared between them. Please, check the **AirBox** section for external events description.

The **Logo** tab and the **TC** tab are not relevant in **ListBox**, thus, they are disabled.

Tools menu

The **Tools** menu is identical to the **AirBox** Tools menu. Please, check the relevant section above for details.

Commands menu

The commands menu contains some playback-related and logo-related commands. As they are identical to the **AirBox** commands, please, check the relevant section above for details.

Help menu

PlayBox Help

This menu item opens the **ListBox** context-sensitive help. It is still under development.

PlayBox Doctor…

Open it to start a tool that automatically collects data about your system and current setting, in order to provide you with proper support. All you have to do is fill-in the mandatory fields and send the resulting PlayBox Doctor report to us. See detailed description of the PlayBox Doctor features in the Basic PlayBox manual.

About…

Clicking on this menu item displays the **About** box of the **ListBox** module. It contains useful information about the module version, WIBU Box number, license type, registration, etc.

CAPTUREBOX

GETTING STARTED

Quick Start

1. Connect the video source signal to **CaptureBox** video input;

2. Launch **CaptureBox**;

3. Select the media folder you wish to capture to;

4. Fill in the *Tape ID* or *Channel* field;

5. Type the file name you want to capture the content to;

6. Click on the **Manual Capture** button;

7. When required, stop the capturing by pressing the **Abort** button.

Congratulations! You have just captured your first **CaptureBox** clip!

CAPTURE SETTINGS

Depending on the plug-in used you have different capture settings. To select the desired plug-in, press the Device Select button in the main user interface. The following dialog appears:
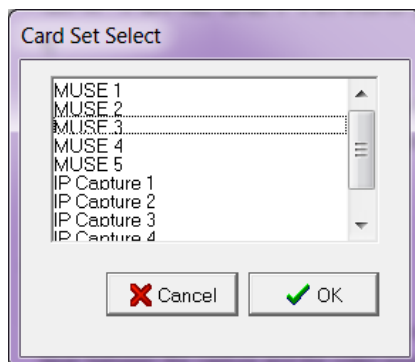
Select the plug-in you would like to use and press OK. To view a detailed description of the MUSE and the IP Capture plug-ins, please, refer to the respective sections below.

MUSE

The MUSE capture plugin stands for Media Universal Source Engine and it provides an additional layer between the capture and the video source, thus allowing the use of the same video source by many clients.
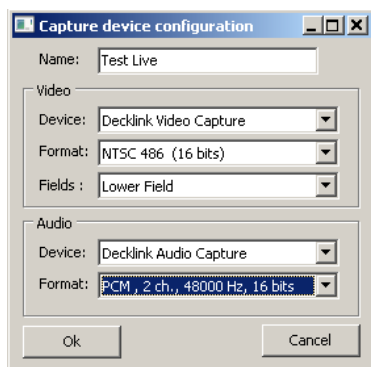
For example, the same LIVE IN video source can be used simultaneously by CaptureBox and *AirBox*. The number of applications, which can use the same live input is limited by the machine resources.

You can make the preferred capture settings for MUSE input in this dialog:

The *Input* tab allows you to select an already preset **Live input** from the drop-down menu, or add a new input. Pressing the **Edit** button invokes the *Live Inputs Editor*, which is the same as the one in *AirBox* and is described in the relevant section above.
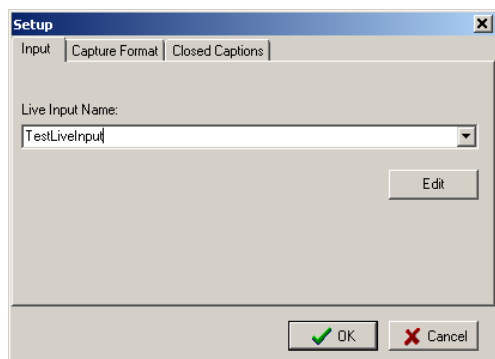
In the *Capture Format* tab you can select your *Video, Audio Encoders* and *Muxer* from the corresponding drop-down menus. You can also write a negative number here if you want to remove frames from the input.
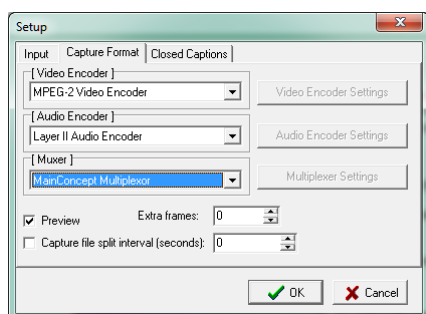
Use the ☑ *Capture file split interval* box if you would like you capture filed to be split in pre-defined seconds intervals.

If you select the *MainConcept Multiplexor* Muxer, the **Video**, **Audio**, and **Multiplexer** settings will be powered by MainConcept©. You also have another option here - the **Avi Multiplexor.**

IMPORTANT! The *PCM Audio Encoder* ONLY works with the *DVCPROHD Video Encoder* and the *Avi Multiplexor*. All other video encoders, except for the *DVCPROHD* one, MUST be used with the *MainConcept Multiplexor*.

The **Closed Captions** tab allows you to manage Closed Captions handling when capturing. Choose one of the three options:
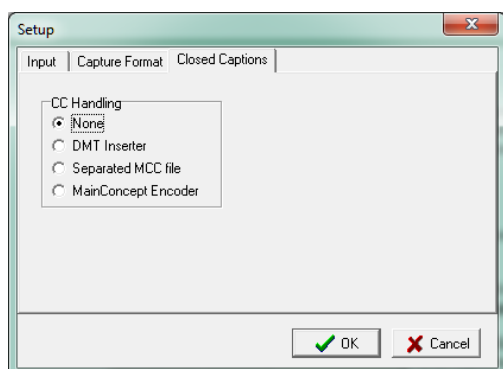
⊙ **None** – Select this option if you would like to disable the Closed Captions option.

⊙ **DMT Inserter** – Select this option if you would like the Closed Captions to be burned in the output image.

⊙ **Separated MCC file** – When this option is selected, the Closed Caption will be saved to a file with an *.mcc* extension in the **CaptureBox** folder.

⊙ **MainConcept Encoder** – This option also burns the Closed Captions in the image. However, it uses the MainConcept encoder for this purpose, instead of the **PlayBox** one.

Video Settings

Depending on the selected *Video Encoder*, **MPEG-2 Video Encoder** or **H264 Video Encoder**, pressing the **Video Encoder Settings** button will invoke different dialogs.

MPEG-2 Video Encoder
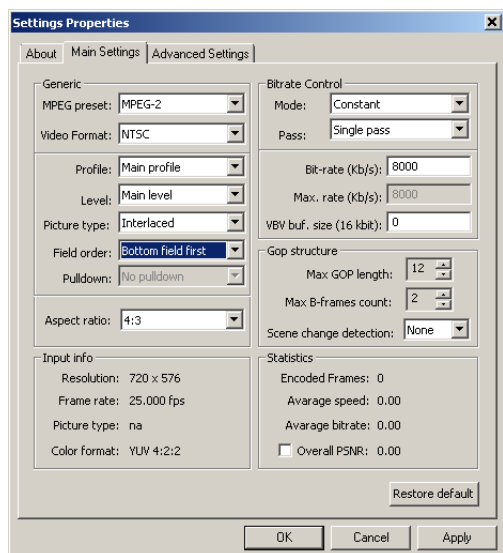
The **Main Settings** tab allows you to configure the video encoder. You can select the settings of the *MPEG preset* and the *Video Format* in the corresponding fields. Further down in the *Generic* area, you are able to define some additional settings of your video input, like *Profile, Level, Picture type, Field order,* and *Aspect ratio*. If you select **Progressive** *Picture type*, you can also set a *Pulldown* mode to convert the number of frames.

The *Bitrate Control* area allows you to modify the input video encoding *Mode* and the type of encoding from the *Pass* field. The encoding type can be **Single pass**, **Multi-Pass Analyze**, and **Multi-Pass Encode**.

Depending on the *Mode*, you have different options for the bitrate speed. For **Constant** mode you can set the Bit-rate speed in kb/s from the corresponding field and for **Variable** mode you can set the minimum *Bit-rate* and the *Maximum rate* in the respective fields. Also, you can set the *VBV buffer size* for the latter two modes. For **CQ_adaptive** and **CQ_strict** mode you can enter the number of I, P, and B frames.

> **NOTE:** The *VBV buffer size* should be set to 112. If you want to enter a different value, please consult our support team at support@playboxtechnology.com.

In the *GOP structure* area you can set the *Maximum GOP length* and the *Maximum B-frames count*, Depending on your needs, enter the following values:

- For I-frame set the Max GOP length to '1' and the Max B-frames count to '0'
- For PAL set the Max GOP length to '12' and the Max B-frames count to '2'
- For NTSC set the Max GOP length to '15' and the Max B-frames count to '3'

In this field you can also set the *Scene change detection* mode.

The bottom fields, *Input info* and *Statistics*, show the configurations of the input, as well as statistics about the encoding speed, bitrate, and the number of encoded frames. If you place a check on the ☑*Overall PSNR* box, you will be able to see also the peak signal-to-noise ratio of your video input.

If you press the **Restore default** button, your settings will be changed back to the default ones.

The **Advanced Settings** tab allows you to change some additional settings. In general, these settings should not be changed, unless advised by our support team. If this is the case, in order to change a certain setting from the **Parameter** column, simply double-click on the corresponding **Value** input, and enter the desired value.
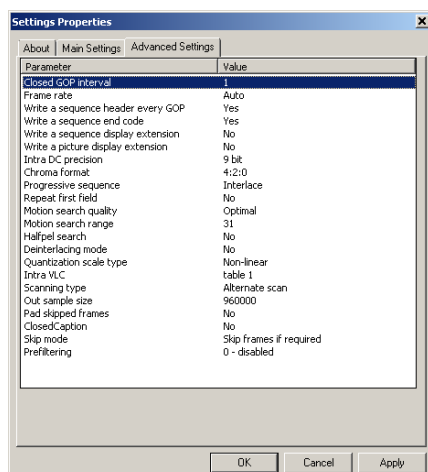
> **IMPORTANT:** Please, do NOT change these settings before consulting with our support team.

MXF Capture Configurations

To configure the proper settings for MXF capture, please follow PRECISELY the steps below:

1. In the **Capture Format** tab of the **Setup** dialog select the following: **MPEG-2 Video Encoder** from the *Video Encoder* drop-down list, **PCM Audio Encoder** from the *Audio Encoder* drop-down list and **MainConcept MXF Multiplexer** from the *Muxer* drop-down list.
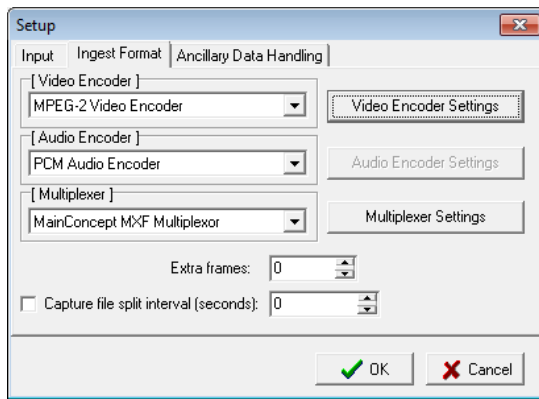
2.   Press the **Video Encoder Settings** button.

3.   In the dialog that appears make sure that **MPEG-2** is selected for *MPEG preset* and **422 profile** is selected for *Profile*:
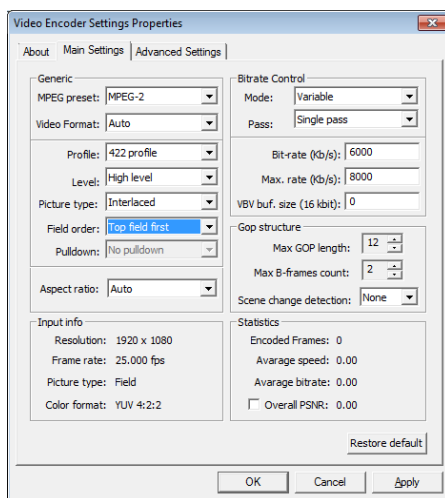
4.   Go to the **Advanced Settings** tab and enter the desired configurations, depending on your preference. Please, refer to the given settings below for SD or HD. When you are ready with the configurations press Apply.

5.   Go back to the **Main Settings** tab and enter the appropriate configurations for SD or HD, depending on your preference. Press **Apply** when you are ready.

6.   Press **OK**. Your MXF capture configurations are now saved successfully!

Settings for SD

Channel in a Box User Manual

Please, refer to the following screenshots to view the SD settings for MXF capture:

Video Encoder Settings Properties

Tabs: About | Main Settings | Advanced Settings

Generic

MPEG preset:	MPEG D10_30
Video Format:	Auto
Profile:	422 profile
Level:	Main level
Picture type:	Interlaced
Field order:	Top field first
Pulldown:	No pulldown
Aspect ratio:	Auto

Bitrate Control

Mode:	Constant
Pass:	Single pass
Bit-rate (Kb/s):	30000
Max. rate (Kb/s):	30000
VBV buf. size (16 kbit):	576

Gop structure

Max GOP length:	1
Max B-frames count:	0
Scene change detection:	None

Input info

- Resolution: 720 x 608
- Frame rate: 25.000 fps
- Picture type: Field
- Color format: YUV 4:2:2

Statistics

- Encoded Frames: 0
- Avarage speed: 0.00
- Avarage bitrate: 0.00
- Overall PSNR: 0.00

Restore default

[OK] [Cancel] [Apply]

Video Encoder Settings Properties

Tabs: About | Main Settings | Advanced Settings

Parameter	Value
Closed GOP interval	1
Frame rate	25.0 fps
Write a sequence header every GOP	Yes
Write a sequence end code	Yes
Write a sequence display extension	Yes
Write a picture display extension	No
Intra DC precision	10 bit
Chroma format	4:2:2
Progressive sequence	Interlace
Repeat first field	No
Motion search quality	Optimal
Motion search range	31
Halfpel search	Yes
Deinterlacing mode	No
Quantization scale type	Non-linear
Intra VLC	table 1
Scanning type	Zig-Zag scan
Out sample size	960000
Pad skipped frames	No
ClosedCaption	No

[OK] [Cancel] [Apply]

Channel in a Box User Manual

Settings for HD

Please, refer to the following screenshots to view the HD settings for MXF capture:
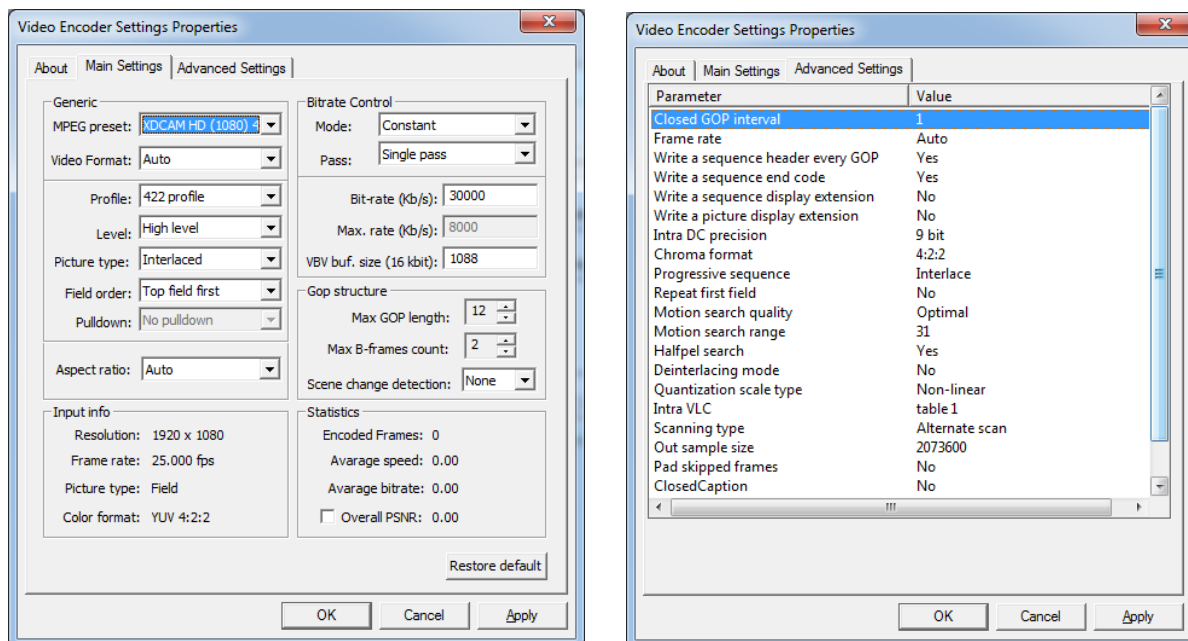
H264 Video Encoder

The **Main Settings** tab allows you to configure the video encoder. You can select the settings of the *AVC preset*. Each Preset is a predefined group of encoding settings, designed to facilitate the user. You can also specify the *Profile*, *Level*, and *Performance* in the corresponding fields. Further down in the *Generic* area, you are able to define some additional settings of your video input, like *Frame type* and *Slice count*.
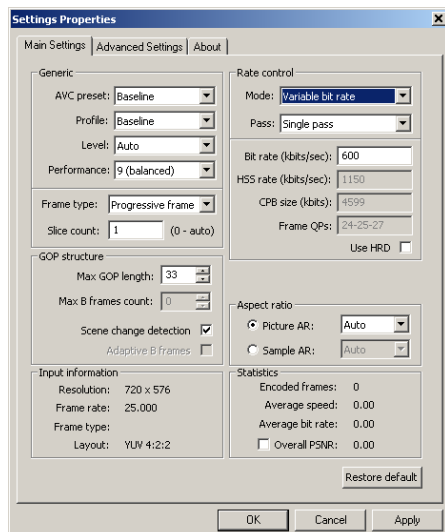
The *Rate Control* area allows you to modify the input video encoding *Mode* and the type of encoding from the *Pass* field. You have three options here:

- **Single pass** – encoding without gathering statistics
- **Multi-Pass Analyze**– encoding and gathering statistics for next pass
- **Multi-Pass Encode**. – encoding using the gathered statistics and updating it

Depending on the *Mode*, you can set different values. For **Constant** and **Variable bit rate** mode you can set the minimum *Bit-rate* speed in Kbits/sec, and for **Constant quantizer** and **Target quality** mode you can enter different values for the *Frame QPs* in the respective field. If you place a

check in the ☑ *Use HRD* box the program will optimize the buffering mechanism with a hypothetical reference decoder, so that the video bit stream will not suffer from buffer overflow or underflow.

Once the Use *HRD box* is checked, you will be able to set the *CPB size* (the size of coded picture buffer in Kbits). If your *Rate control Mode* is **Variable bit rate** or **Target quality**, you can also enter a value for the *HSS rate* (the hypothetical stream scheduler rate (bits/sec) of the encoded video elementary stream).

Once the Use *HRD box* is checked, you will be able to set the *CPB size* (the size of coded picture buffer in Kbits). If your *Rate control Mode* is **Variable bit rate** or **Target quality**, you can also enter a value for the *HSS rate* (the hypothetical stream scheduler rate (bits/sec) of the encoded video elementary stream).

> **NOTE:** In *Use HRD* mode it is advisable that the *Rate control Mode* is set to **Constant bit rate** and the *CPB size* is equal to three times the Bit rate.

In the *GOP structure* area you can set the *Maximum GOP length* and the *Maximum B-frames count*, Depending on your needs, enter the following values:

- For I-frame set the Max GOP length to '1' and the Max B-frames count to '0'
- For PAL set the Max GOP length to '12' and the Max B-frames count to '2'
- For NTSC set the Max GOP length to '15' and the Max B-frames count to '3'

Depending on your preferences, you can also check ☑*Scene change detection* and ☑*Adaptive B-frames*, if you have entered a value in the *Max B frames count* spin-box.

The *Aspect ratio* field allows you to set a ⊙ **Picture AR** or a ⊙ **Sample AR** mode and select a ratio from the corresponding drop-down menu.

⊙ **Picture AR** fixes the aspect ratio of the whole picture;

⊙ **Sample AR** fixes the aspect ratio of pixels in the output picture.

The bottom fields, *Input info* and *Statistics*, show the configurations of the input, as well as statistics about the encoding speed, bitrate, and the number of encoded frames. If you place a check on the ☑*Overall PSNR* box, you will be able to see also the peak signal-to-noise ratio of your video input.

If you press the **Restore default** button, your settings will be changed back to the default ones.

The **Advanced Settings** tab allows you to change some additional settings. If you want to change a certain setting from the **Parameter** column, simply
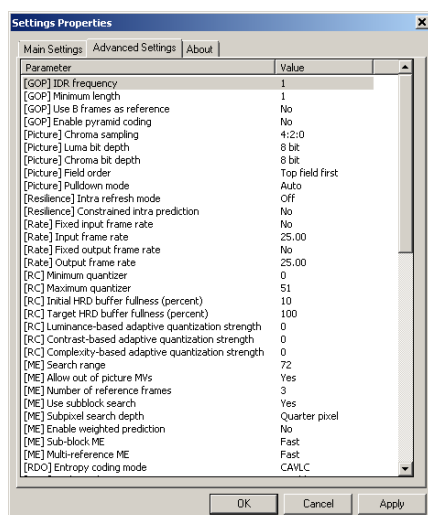
double-click on the corresponding **Value** input, and enter the desired value.

Channel in a Box User Manual

DMT QSV H264 Video Encoder

This video encoder only works with a dedicated video processor. For more information please contact our support at support@playboxtechnology.com.

Please, refer to the image below to view the correct video configurations for **DMT QSV H264 Video Encoder**:
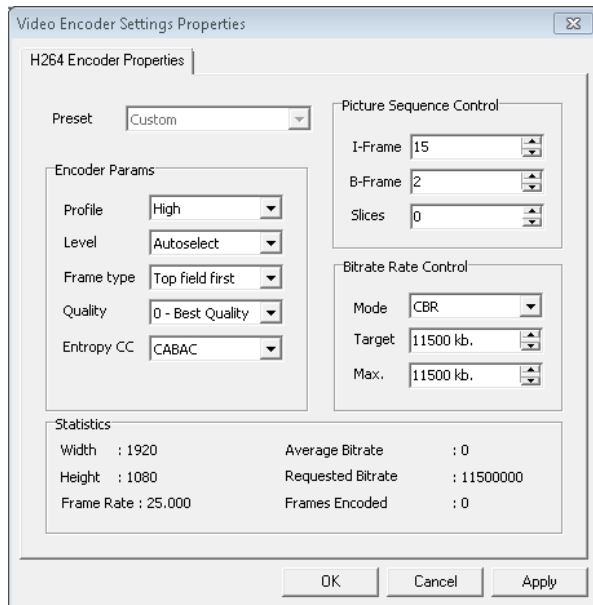
DMT AMD H264 Video Encoder

This video encoder only works with a dedicated video card. For more information please contact our support at support@playboxtechnology.com.

DVCPROHD Video Encoder

This video encoder only works with the **PCM Audio Encoder** and the **Avi Multiplexor.** For more information please contact our support at support@playboxtechnology.com.

Please, refer to the image below to view the correct video configurations for **DVCPROHD Video Encoder**:

Channel in a Box User Manual
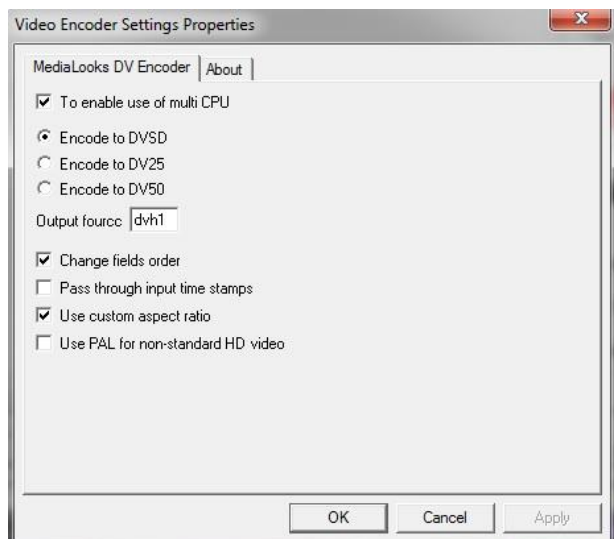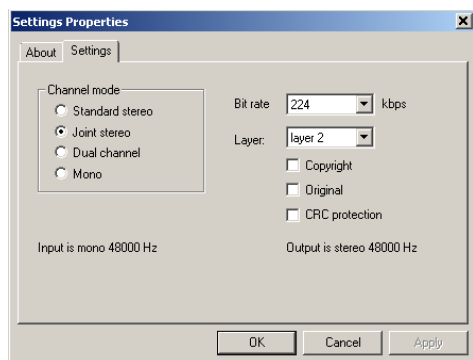
Audio Settings

Pressing the **Audio Encoder Settings** button in the **Capture Format** tab of the **Setup** dialog will invoke the following dialog:

Here you can define the audio *Channel mode* from the corresponding radio buttons. Also, you can set the *Bit rate* in *kbps* and select Audio **layer 1** or **2** for the output. In addition, you can check if the output should be ☑*Copyrighted*, ☑*Original*, or ☑*CRC protected*.

NOTE: The advised settings here are 224 kbps *Bit rate* **layer 2**.

Multiplexer Settings

The third button, **Multiplexer Settings,** opens the following dialog: Here you can view the parameters of the output MPEG-2 TS stream.
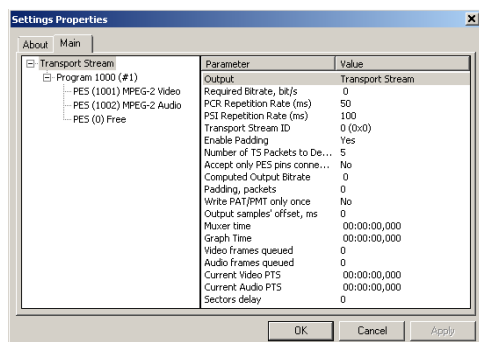
The left field of this dialog gives you an overall schema of your input. Depending on the selected row to the left, you will have different output in the left area of the Settings Properties window, which shows you the configuration of the particular input. You can manage these configurations by double-clicking on the respective entry in the **Value** column.

> **IMPORTANT:** Please, do NOT change these settings before consulting with our support team.

IP Capture

This plug-in is designed for capturing of MPEG2 Transport Streams coming from the network. The resulting file format is MPEG2 TS (no re-encoding takes place).

In the *Destination Address* field enter the IP address of the desired media to be captured and select the *Port* to be used. In case the *Destination Address* is a multicast address, i.e., it receives streams from multiple addresses, you also need to enter the IP address of the specific stream you wish to capture. This is done in the *Source Address* field. Finally, use the *Network Interface Address* field to select the desired network card to be used.
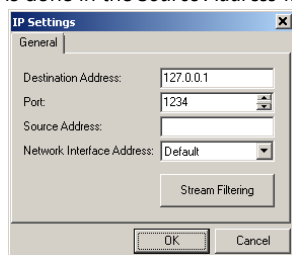
In the setup dialog, enter the IP address of the sending machine and the port at which it streams. If the incoming stream is MPEG2 Transport steam, push the **Stream Filtering** button to select which streams should be left in the captured file.
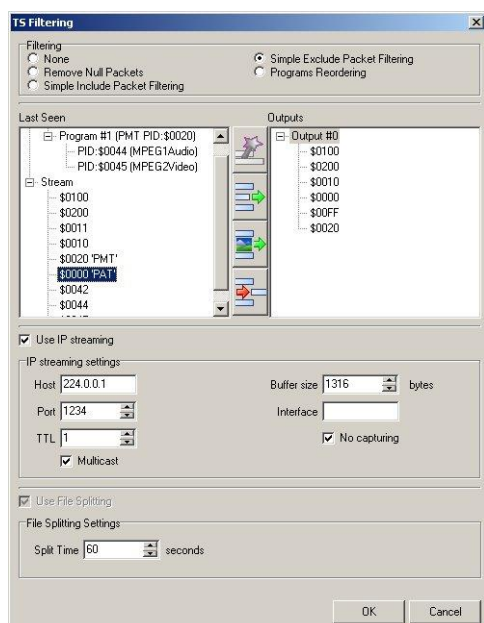
In the middle of the Filtering dialog, you can find a list of all Programs and Streams that were present in the incoming stream at the time of pressing the **SetUp** button.

To the right of it, there are several buttons that are activated, depending on the selected *Filtering* method. You can select it in the area above:

None – there will be no filtering. **CaptureBox** will capture the incoming stream as is.

Remove Null Packets – as some interfaces need constant bitrates to operate properly, Null packets are included to stuff-up the gap between the real bit rate and the required bit rate. These packets do not carry any information and can be removed in order to reduce the bit rate of the captured files, thus saving storage space.

Simple Include Packet Filtering – Check this radio-button and select which streams to be included in the Output:

Use the **Add** button to add the currently selected line in the left to the Outputs list in the right.

Use the **Add Custom** button to type manually the PID you would like to add to the Outputs list.

If you want to remove an already added stream, select it in the Outputs list and press the **Delete** button.

Simple Exclude Packet Filtering – Check it and select the PIDs you do not want to include in the Output file/stream. Use the **Add/Add Custom** button to create a list of PIDs to be excluded from the output.

Programs Reordering – in this mode, you can create several outputs by pressing the **Create Output** button. Then, you can assign the programs to go to each output: Press the Create output button as many times as necessary. Then, select the output in the list to the right and click on the program line to the left. Press **Add** to assign it to the relevant output. Then, check ☑*Use IP streaming* if you want to send the selected output to the network; or leave it unchecked if you only want to capture the stream to an MPEG2 Transport Stream file. When there is more than one output, the endings of the resulting files' names will indicate the number of the output (zero-based).

Check ☑*Use IP streaming* to send the incoming signal to the network. In the IP streaming settings area, specify the *Host* IP address and the *Port* to which you would like to send the stream. Check *Multicast* if you need to send the stream to numerous machines.

In the *TTL* spin-box, specify the number of switchers the stream can pass (Time to Live).

To the right, modify the buffer size depending on your needs. Usually we recommend setting this size to 1316 (7 UPD packets of 188 bytes).

If you have more than one LAN cards in the machine, specify which one should be used for the streaming. Otherwise, the stream will be output through all network connections.

Check ☑*No capturing* if you do not want to save the incoming stream to a file. If you leave it unchecked, **CaptureBox** will write the stream to an MPEG2 TS file.

☑*Use File splitting* is active when **CaptureBox** writes the captured stream to a file (i.e. ☐*No capturing* is not checked). This functionality allows you produce chunks of files based on predefined periods. You can adjust the period (in seconds) in the *Split Time* box below.

USER INTERFACE
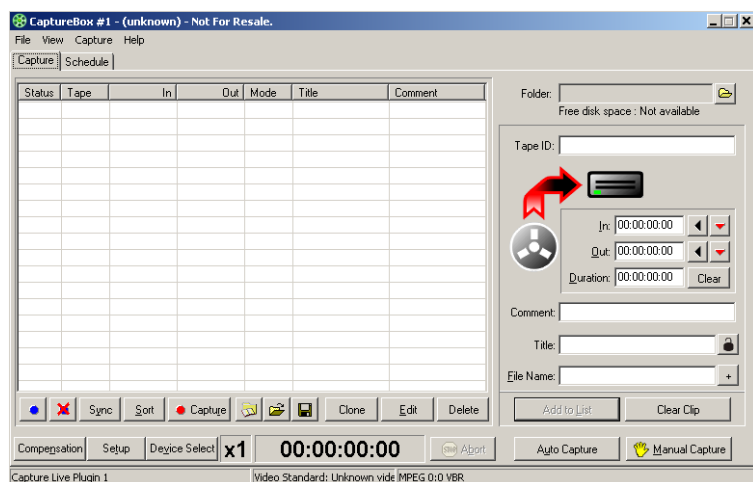
Capture Mode

When the **Capture Page** is active, the module is in "record" mode. In this mode, you can choose a batch of scenes from tapes, collect them in the batch capture list or record every single scene manually from a *VTR*.
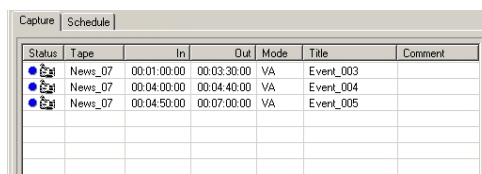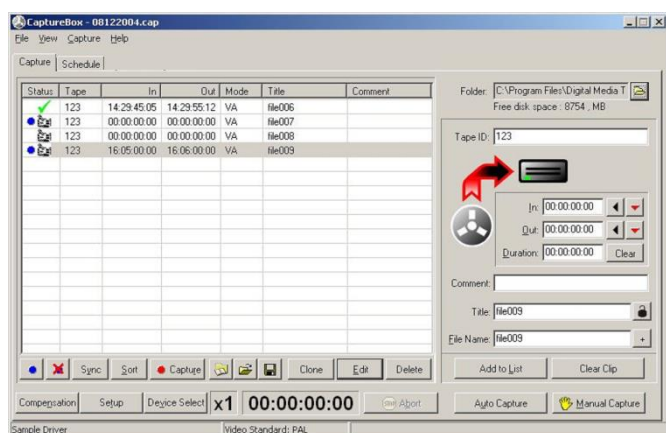
Batch Grid

The automated batch capture grid occupies a large part of the window. Many people refer to this function as "batch capturing". You can define a list of scenes with their **start** and **end** timecode values. Then activate batch capturing. **CaptureBox** captures the desired scenes automatically from the corresponding tapes. The operator should only change the tapes when prompted.

Grid Columns:

- Status column shows the current state of each clip, as follows:

  - If the clip has been successfully captured, a green mark ✓ appears.

  - If the clip is captured, but there is no option for *VTR* control, a yellow hand appears, and the timecode values in the **In** and **Out** columns are zeroes.

  - If not the whole clip has been captured, or there has been a problem during the capturing process, a red mark × appears.

  - If the clip is not captured yet, a "camera" appears.

  - If the clip is included in the next **capture** session list, a blue dot ● appears. The dot can be removed by clicking that field. This will exclude the clip from the next capturing session.

  - If the clip is not included in the next **capture** session, the blue dot is missing. The dot can be added by clicking that field. Thus, you will include the clip in the next capturing session.

  - Tape column shows the ID of the tape, from which the particular clip will be captured.

  - The **In** column shows the initial timecode, when clip capturing will start.

  - The **Out** column shows the timecode, when clip capturing will stop. If you click on the grey bar named **Out**, it will be renamed to **Duration** and the column will show the clips' durations.

  - Mode column shows what will be captured for the particular clip – video (V), audio (A) or both (VA). Currently only VA is supported.

  - The **Clip Name** column shows the clip names. If you click the grey bar named **Clip Name**, it will change to **File Name** and the column will display the destination full path where the clip will be stored – hard disk name, folder and file.

  - Comments column shows the comments (if any) for each clip. You can enter your comments in the relevant string of the clip data field to the right.

Grid buttons:

▪ The **Blue dot** button includes the selected clip in the next capture session.

▪ The **Red X** button excludes the selected clip from the next capture session.

▪ The **Sync** button synchronizes the list. All clips with "not captured" status are included in the next capture session.

▪ The **Sort** button sorts the list by Tape ID and then by Start Timecode. Thus, the batch capture process is simplified and optimized.

- The**Capture** button starts an automated capture session. During this session, all clips, marked with a blue dot will be captured to the hard drive.

- **Delimited List Import** button enables loading all types of tab-delimited text files into the batch grid. You will have to create templates to "tell" **CaptureBox** what is the structure of your tab-delimited file, i.e. what information does each column contain.

- Pressing the **Delimited List Import** button will open a dialog for you to specify the template to use when loading your file:
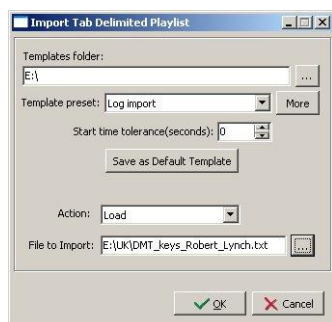
The *Template preset* drop-down list contains all the templates stored in the **Template Folder** (see the **Template Builder** description below).

Select the *Action* you would like to execute from the drop-down list. You can either Insert, Append or Load the file. The latter action will delete all previously loaded entries in the batch grid.

Finally, browse for the file you would like to import using the selected template and click **OK**.

NOTE: The **OK** button will not be active until you fill in all the strings in this dialog.

At opening this dialog for the first time, you will have to **create a template** first. First, select the **Template Folder** – all the templates you create will be stored there until you change it.

Push the **More** button to open the **Template Builder**:

Type the template name in the *Template:* string. If the selected templates folder already contains some template files, they will be listed in the drop-down list.

Push the **Sample File** button to open an example file for your template.

If there are some rows in the beginning of the file that you would like to skip, enter their number in the *Number of lines to skip* string. The skipped lines will be colored in red.

If there is a symbol in the beginning of each row in the file that you would like to skip, select it from the *Comment:* drop-down list. Then, specify the Delimiter from the drop-down list.

Now that you have set the basic rules, you will have to "explain" the **Template Builder** what information each column contains: Go to a column's header and click in it. Then select one metadata category from the drop-down list to assign it to the relevant column. Once assigned, this category will be checked

in the *METADATA* list to the left. You can un-assign a category either through un-checking it in the METADATA list or by selecting [Clear] from the drop-down list.

Select [Bulk] if you wish to skip a column.

When the preset is ready, press the ***Save*** button to store it in the Templates folder.

Press the ***Validate*** button to check if the current template matches a specific file.

To load the currently selected template in the *Template preset* string, press the ***Pick*** button.

If you already have some templates and you select one of them from the drop-down list, push the ***Load*** button to load it (its settings will be displayed in the grid).

Back in the **Import Tab Delimited Playlist** dialog, you can set a default template by pushing the ***Save As Default Template*** button.

NOTE: Do not use the *Start time tolerance* spin-box – it is related to ***AirBox*** only.

- The ***Folder*** button allows loading a list, preliminary prepared in ***CaptureBox*** with clips ready for capturing (**.cap* file). The name of the current list is written in the title bar of the module, right after ***CaptureBox***. If the list has been changed and not saved, an asterisk (*) appears after its name.

- The ***Diskette*** button saves the current capture list to a file, which can be used later.

- The ***Clone*** button "clones" the selected clip. Its data (title, file name, in/out point, duration) are copied in the right-hand panel for use in the next entry of the batch capture list. This functionality could save time for entering almost the same data for each row. Just change the different points and there it is!

- The ***Edit*** button allows changes in the description of a clip. The clip data are loaded into the Data Fields and you can edit them. During editing, the Edit button transforms to ***Cancel*** and the ***Add to List*** button transforms to ***Apply.*** By pressing the ***Apply*** button, your changes are applied in the list.

You can also edit a clip by double-clicking it.

- The ***Delete*** button removes the selected clip from the list.

- The ***Compensation*** button– invokes a dialog box for defining the capturing delay compensation (in frames), when you capture from video recorder. On some stations you must manually compensate for some delays when capturing to different formats (DV, MPEG2 AVI, MPG).

- The ***Setup*** button – shows a window for setting up Inputs/Outputs of device, capturing format and specific settings.

NOTE: The setup dialog box may vary, according to the platform type. See the Capture Settings section above for description of some platforms' setting dialog boxes.

- The ***Device select*** button– allows choosing the capture device or software simulation of capturing (**Sample Driver**).

- The ***Abort*** button– it activates during the capture preparing only. Press it to stop the capture.

- The *Speed* field x1 shows the speed of capturing.

- The *Timecode* field– shows the timecode during capturing.

Channel in a Box User Manual

Clip Data fields

This field is designated for clip description. It is not possible to include the clip in the batch capture list if you have not specified the clip location on the hard drive, the tape ID, the initial and the final timecode, as well as the clip name.

- *Folder* – this field describes the folder in which files will be captured. Pressing the **browse** button next to it opens a browse dialog box where you can specify a hard disk and a folder for storing the captured clip. Under the field you can see information about the free disk space at the selected disk.

- *Tape ID* – In this field you must type the ID of the source tape from which will be captured the footage. This is very important if you work with more than one tape or the tape timecode is not continuous. Later, during the batch capture session, you will be asked for tapes by their IDs.

- *In* and *Out* fields specify the initial and the final timecode respectively. If you choose to fill-in the information manually, you could use either of the following separators: **colon (:), semi-colon (;), dot (.)** or **comma (,)**. Of course, you do not need to enter the leading zeros in any field.

For example, if you enter 1.2.3 this will be translated to 00:01:02:03.

If you don't enter any disjunctive symbols in the timecode, this will be interpreted as a number of frames. For example, if you enter "100", this will be interpreted as 4 seconds (00:00:04:00).

- *Duration* – Its value is automatically calculated by subtracting *In* from *Out* values. It is possible to type a value only in the *In* field and define *Duration*. The value of *Out* field will be calculated automatically.

To the right of *In, Out* and *Duration* fields are situated buttons that function as follows:

- Pressing the **black arrow**, pointing left, will insert the current timecode from the *VTR* in the corresponding field

- Pressing the **red arrow**, pointing down, will rewind the tape exactly to the timecode, written in the corresponding field

- Pressing the **Clear** button clears all values in the *In, Out* and *Duration* fields.

In the *Comment* field you could enter a description or a comment, concerning the particular scene sequence. Your comment will be displayed in the relevant Grid column.

Title represents the name of the clip. If left empty, the field will be automatically filled-in with the corresponding *File name*.

- **Lock** button is used for locking the *Title* to the *File name*, i.e. any changes in the clip name will affect the file name and vice versa.

- *File Name* stands for the name under which the captured clip will be saved. If left empty, the field will be automatically filled-in with the *Title*.

For your convenience, the filename is automatically increased by pressing the **Plus** button or **Add to List** button. If the last clip name was [Capture00*1*], the next filename would be [Capture00*2*], and so on. If the filename does not end with a number, but with a letter, the letter will change in alphabetical order, i.e. if the last clip name was [Sofi*a*], the new filename will be [Sofi*b*], then [Sofi*c*] and so on. Of course, there is an option to enter a new name manually.

- **Plus** button [+] increases the *File name*.

- **Add to List** button - transfers the clip data into the batch capture list (on the left) and most of the clip fields are cleared except *Tape ID* and *Folder*. The *File name* increases.

  - **Clear Clip** button - clears all clip data.

Capturing

[Auto Capture] [🖐 Manual Capture]

There are three methods to start capturing:

- **Manual Capture**–this mode can be applied only if *Tape ID, File Name* and *Folder* fields are filled-in. If the timecode field *In* is empty, pressing this button will open a window for manual start of the capturing. If the *In* field contains timecode, **CaptureBox** will start counting down for the same amount of time before commencing the capture. Press **Finish** button to stop the capture.

- **Auto Capture** (automated single capture) – this mode can be started only if the *Tape ID, In, Out, File Name* and *Folder* fields are filled-in. When capturing is finished, the clip data is automatically moved into the batch list and marked as captured. There is no need of re-capturing, except when a blue dot is set in front of it. To stop the capturing manually, press the **Abort** button. Clip data will be moved to the batch list, but the clip will be marked as not fully captured.

- The Batch Capture is in fact automated capturing of a series of scenes. Activate it with the **Capture** button, which is situated under the clip list. All the clips from the list that are marked with a blue dot will be captured in ascending sequence of the time codes and tape IDs. During this process, no special attendance is necessary– one should only take care of changing the tapes when prompted.

> **TIP (!)** You can adjust the preview window size by right-clicking in the preview window.

> **IMPORTANT:** The PlayBox modules DO NOT support the Deck Control connector, supplied on the DeckLink breakout cable!

Using the Time Delay – Instant Replay option (TDIR)

TDIR functionality is available only for *.mpg files, i.e. for plug-ins, where *.mpg files are supported.

To use the TDIR option, launch **AirBox** and right-click over the grid. Choose **Add/Insert Incomplete clip** from the context menu. You can also **Add/Insert Incomplete clip**s from the **Edit** menu.

In the Clip Properties dialog, browse and point the location where the clip will be captured. Type the Filename and copy it (**<Ctrl** + **C>**), you will need it later in **CaptureBox**. Start the playback.

Next, run **CaptureBox** and prepare for capturing - make the necessary settings and fill in the *Clip Data fields*. Use **<Ctrl** + **V>** to paste the filename you copied from **AirBox**.

There should be at least 5 seconds between the capture start and the start of the incomplete file's playback.

Here is an example of setting this function. In it, a clip that is still being captured will be played back in **AirBox** for 20 seconds:

1. Open **AirBox**

2. In the **Settings** menu➔**Settings** dialog, set *Check Missing* every [3] seconds and click **OK**.

3. Load a playlist; right-click and select **Insert/Incomplete clip**.

4. In the **Clip Properties** dialog, type in the title and the filename.

5. Use **<Ctrl** + **C>** to copy the file name, you will need it later.

6. Browse for the location to which the clip will be captured. Then set the *Duration* to [20] seconds. In the bottom of the Clip Properties dialog, check ☑*Live (delayed) file* and clip and set duration ().

7. Push Play. The incomplete file is marked Missing as it still does not exist.

8. Open **CaptureBox** and specify the Folder you will capture to,

9. the clip name and the filename (Use **Ctrl** + **V** to paste from **AirBox**)

10. Start manual capturing.

11. A few seconds later the incomplete clip will become available, but it will still be marked as missing as it was inserted before it was created.

12. It will be played for 20 seconds as specified in its properties.

13. When **AirBox** starts playing the next clip, you can go to **CaptureBox** and stop capturing (if needed).

(!) TIP: You could fully automate this process using the **CaptureBox** scheduler. Prepare your schedule-capturing list and insert the incomplete (still missing) clips in the **AirBox** playlist accordingly. Thus, if you have set the correct timing, **CaptureBox** will start capturing and **AirBox** will start playing back the relevant clips.

WARNING! You must have SCSI or very fast SATA storage for the TDIR option. Otherwise capturing will most likely fail.

RS-422 Controller

Each *VTR* with *RS-422***SONY** protocol can be used in **CaptureBox**. Switching the active window between the main and this one is done by pressing the **<Insert>**key. It is possible to attach this window to the main one by moving it close to any of its edges. Thus, the *VTR* control panel will move together with the main window. Here is a short description of the *VTR* control panel:

▪ The *VTR* brand is displayed in the window's caption. If there is no connection, the text [Not Connected] will appear.

▪ Display Indicators:

- **TCR** has three possible states – **TCR** (normally read timecode), **T*R** (corrected timecode), **TCG** (generated timecode, red).

- **RecInh** a record-protected tape is inserted in the *VTR*.

- **Local** indicates that the *VTR* is in local control mode and the remote control is not possible.

- **Tape** indicates that there is a tape in the *VTR*.

- **EOT** (End-Of-Tape) alarms that the tape is about to end or that the tape end has actually been reached.

- Indicator **>>** or **<<** - shows the tape roll direction.

- **Preroll** indicator informs that the *VTR* is executing a preroll command.

- **Servo** indicator shows that playback is running stable.

- **CueUp** indicator – shows that the timecode positioning command has been successfully completed.

▪ *Tape/AutoEE* and *Full EE* – depending on *VTR*'s model and its settings, a combination of these keys opens the **E-E** circuitry.

▪ **Config** button – opens a configuration dialog box, where you can define:
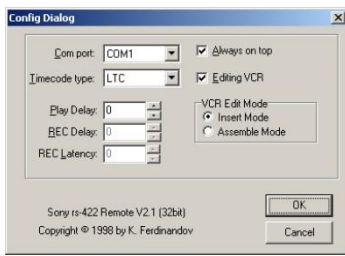
- ▪ *Com port* – the port to which the recorder is connected.

- ▪ *Timecode type* – the type of the timecode: LTC, VITC, etc.

- ▪ *Play Delay* – the delay (in frames) of the video recorder start towards the capturing start. This setting refers to the **Capture mode**.

- ▪ *REC Delay* – the delay (in frames) of the clip start towards the video recording start. This setting refers to the **Print-to-tape mode**.

- ▪ *REC Latency* – the frame offset of the records beginning toward the clips *IN* point. This setting refers to the **Print-to-tape mode.**

NOTE: The *REC Delay* and *REC Latency* fields are accessible only when the ☑*Editing VCR* box is checked.

- ▪ *Always on top* **–** check it, if you want to seetheRS422 window always on top.

- ▪ *Editing VCR*– it should be checked, if you use an editing video recorder. If the box is checked, you can set the *REC delay* and *REC latency* fields and the *Video recorder edit mode.*

- ▪ *Video recorder edit mode* – you can choose between *Insert* and *Assemble.*

- ▪ *RecInh* box– allows or forbids the **REC** button in order to protect the tape from accidental mistakes.

- ▪ The *Shuttle* slider – allows shuttling within the particular *VTR* capabilities. If the *VTR* allows it, you could achieve variable speed by shifting this button left or right.

- ▪ The *Jog* slider allows frame-accurate positioning.

IMPORTANT: The **PlayBox** modules DO NOT support the Deck Control connector supplied on the DeckLink breakout cable!

Schedule
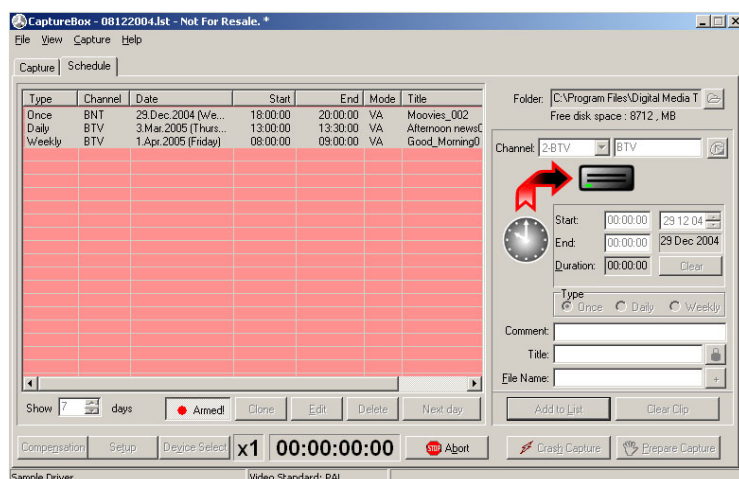
When the **Schedule** page is active, the module is in "schedule" mode. In this mode, you can define a capturing schedule and start capturing later.

The **Schedule** mode interface is similar to the **Capture** mode interface. There are several differences in Clip description and Grid buttons.

Schedule Grid

The grid occupies a large part of the window. In it, you can create a list of scenes that will be captured after activating the scheduling. This feature is used mainly for capturing TV and satellite signals, but of course, you can capture *VTR* signals as well.

Grid Columns:

Type – shows the schedule type: daily, weekly, etc.

Channel – the TV channel which will be captured.

Date – the starting date of each capture item.

Start – shows the initial time, when clip capturing will start.

End – shows the time, when clip capturing will end. If you click on the grey bar named **End**, it will change to **Duration** and the column will display clip durations.

Mode –shows what will be captured for the particular clip – video (V), audio (A) or both (VA).

Clip Name – contains clip names. If you click the grey bar called **Clip Name** it will change to **File Name** and the column will display the destination pull path where the clip will be stored – hard disk name, folder and file.

Comments –shows the comments for each clip.

Grid buttons:

Show days field– here you can define how many days in advance (counting from today) you will see in the schedule list.

Press the ***Arm*** button after creating the schedule list. This will activate the schedule capturing mode. The capturing will automatically start and stop, according to the "Start time" and "End time" values entered in the list.

Clone "clones" the selected clip. Its data (type, channel, in/out point, duration) is copied for use in the next clip to be included in the schedule list.

Edit button changes the description of the clip. The clip data are loaded into the Data Fields and you can change them. During editing, the ***Edit*** button transforms to ***Cancel*** and ***Add to List*** button transforms to ***Apply.*** The changes you've made will be applied in the list by pressing the ***Apply*** button. You can also edit a clip in the list by double-clicking it.

Delete button removes the selected clip from the list.

If you select a grid entry and press the **Tomorrow** button a new entry to the list will be created. It will have the same data as that of the selected entry, but its starting day will be on the following day. In short, this button performs a kind of "tomorrow cloning".

Clip Data Fields

The *Folder* field contains information about the file location of the captured clip. Pressing the browse button next to it opens a browse dialog box, where you can specify a hard disk and a folder for storing the clip.

Under the field, you can see information about the free disk space on the selected hard drive.

The Channel field contains information about the TV channel which will be captured. You can select it from the list of available channels in the left string or create it in the field next to it.

Pressing the satellite 🔴 button, opens a dialog box in which you can create the list of up to 16 channels. You can add a new channel by double clicking in an empty line and typing the channel name.

   Start time/End time - specify the initial and the final time respectively.
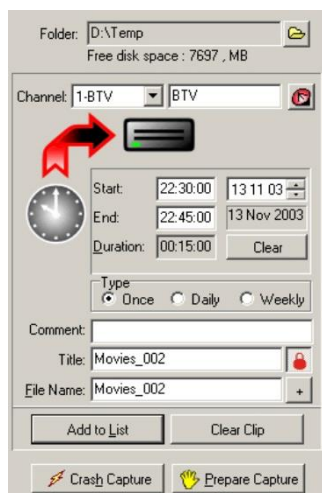
> **NOTE:** These must be in 24-hours' time format! AM/PM is not supported!

   Duration – Its value is automatically calculated by subtracting *Start* from *End* values.

   Date **–** the capture starting date.

   Type **–** defines the frequency of capturing – once, every day, or every week.

   Comment - contains a description or a comment, concerning the particular scene sequence.

   The *Clip Name* field contains the name of the clip you're going to capture. If you leave the field empty, it will be automatically filled-in with the corresponding file name.

   The **Lock** button 🔒 is used for locking the *Clip name* to the *File name*. Any changes you make in either field will automatically occur in the other.

   In the *File Name* field you can enter a name for the captured file. If the field is left empty, it will be automatically filled with the *Clip name*.

   Add to List button transfers the clip data into the schedule capture list.

   Clear Clip button clears all clip data.

   Crash Capture button begins capturing of the available channel immediately and without confirmation, irrespective of the entered schedule list.

Prepare Capture – opens a dialog box asking for confirmation to begin capturing of the available channel instantly, independent of the entered schedule list.

MENU BAR

File Menu

Open

This command opens a previously created Capture list (*.cap) or Schedule list (*.lst).

Save

Press it to save the current Capture list (*.cap) or Schedule list (*.lst).

Kramer Config

In **Schedule mode**, you can list up to 16 channels to be captured. Use a Kramer device to switch between them automatically. You can also switch the channels manually via the *Manual Channel Switch*. In the *Machine* field, enter the number of the Kramer switcher used (there could be up to 8 switchers connected to the PC).

If you press the ***Advanced*** button, the following dialog will open for you to specify the interface settings:
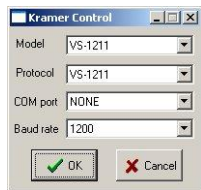
Select your Kramer switcher model from the drop-down *Model* list.

In the *Protocol* field, specify what protocol it uses.

Assign the *COM port* for the switcher using the drop-down list of available COM ports.

Set the *Baud rate* according to the Kramer switcher manual.

Channel in a Box User Manual

GPI Config
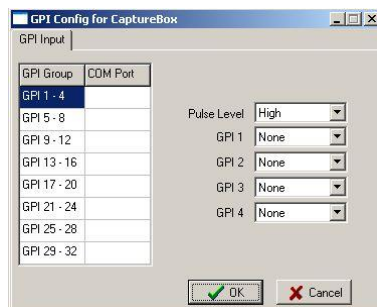
Capturing can be initiated or stopped from an external *GPI* device. This option is valid for manual capture only. You can connect up to 8 *GPI*-devices. Each of them can send up to 4 *GPI* commands. Each device must be associated with an available *COM* port on the computer.

To assign a device, select it from the list to the left and specify to which COM port you will connect it.

In the *Pulse Level* field to the right, you can specify the type of the trigger pulse. Below, set the GPI commands from the relevant drop-down lists.

The pins involved in **CaptureBox** GPI are the same as of **AirBox** GPI. For more information about the *GPI* interface and pins involved, look up in **Appendix 1** further in this manual.

DTMF Config

The DTMF reader allows the user to slave **CaptureBox** to external DTMF tones that arrive on the sound card of the PC. Currently it works only in Capture mode and the supported commands are Start and Finish.

A two-tab dialog will open on selecting this menu item.

In the **Plug-in** tab, specify the device to which is connected the DTMF carrying cable.
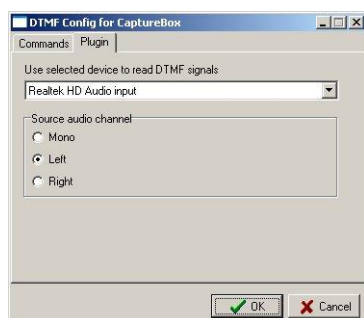
In the *Source audio channel* field below, select which channel is carrying the DMTF tones.

In the **Commands** tab, create a list of commands to be executed at receiving certain combination of tones.
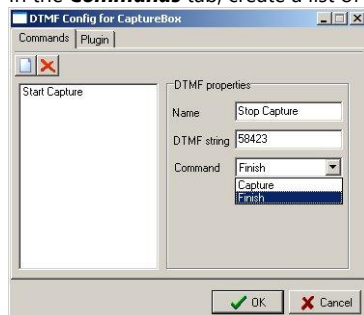
First enter the command's Name in the string above. Then, specify the sequence of tones that will trigger this command. Finally, select the Command in the drop-down list.

Press the **Save Preset** button in the upper left to save the command parameters.

Repeat the procedure to add commands to the list in the right.

To delete a command, select it and press the ❌ ***Delete*** button.

Thus, each time this tone sequence appears on the PC sound card, ***CaptureBox*** will execute the selected command.

> **NOTE:** The DTMF reader operates only in Capture mode! It will not execute commands while ***CaptureBox*** is running in Schedule mode!

Preferences

The preferences dialog box consists of three pages:

In the ***General*** page you can check the current video standard which depends on the selected plug-in.
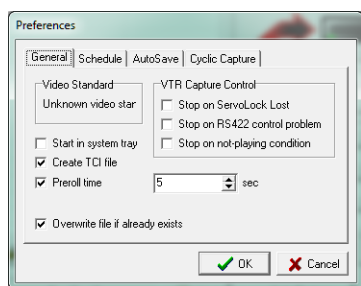
The video standard can be changed in the specific set up form of the plug-in (if the relevant plug-in supports this functionality). The current video standard is also displayed in the bottom of the main ***CaptureBox*** window. If you want ***CaptureBox*** to start minimized, check ☑*Start in system tray.*

TCI files are automatically generated during capturing. They contain information about the tape ID, In and Out time code on the tape, as well as clip duration and notes on the captured files. Most of this information can be imported in ***DataBox*** for automated creation of new records in the database. Please check the TCI Import section in ***DataBox➜Options*** description.

The time codes in *.tci files can be read by the Clip trimmer for subtitling purposes. If you do not need them, uncheck ☑*Create TCI files*, which is checked by default.

☑ *Stop on ServoLock Lost* – this option is designed to prevent bad video capturing. If for some reason, the VTR reports lost servo lock ***CaptureBox*** will stop capturing and will send a Stop command to the VTR. Thus, you will be able to see the exact position of the bad tape.

☑ *Stop on RS422 control problem*– if this box is checked ***CaptureBox*** will stop ingesting in case the RS422 is disconnected. This is especially useful in batch capturing, since the file duration is kept.

☑ *Stop on not-playing condition*– if this box is checked ***CaptureBox*** will stop ingesting if the VTR is not in playing state, i.e., it is stopped, paused, rewinding/forwarding, or is in a go-to state

☑ *Preroll time in seconds*–if checked, this box allows you to set a custom preroll time via the spin-box below.

☑ *Overwrite file if already exists* – if you check this box, if a file with the same name as the currently captured file exists, it will be automatically overwritten

In the ***Schedule*** page, you can define Naming Type for the captured files. They are AUTOMATICALLY formed using the original filename entered in the File Name field (while in Schedule mode) plus some kind of index to distinguish them from one another:
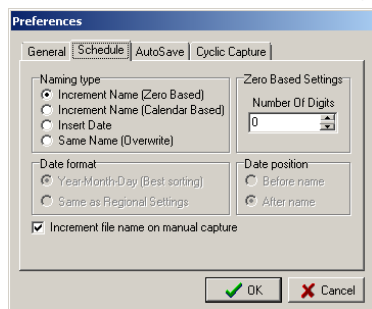
⊙ *Increment name (Zero Based)* –the consequent file names will be formed by adding increasing numbers to the original file name. The number in the original file name is always zero; therefore, this naming type is called "Zero-based". In the Zero Based Settings field, you can specify the number of digits to be displayed.

⊙ *Increment name (Calendar Based)* –the names of consequent files will be formed by adding the subsequent calendar number of the capturing day to the original file name (this number may vary from 1 to 365/366 – in leap years).

⊙ *Insert date* – inserts the capturing date in the filename. If you choose this, the Date position (choose position before or after the file name) and Date format fields will become active.

⊙ *Same name (Overwrite)* – Thus, each time **CaptureBox** starts schedule capturing, it will write the data to the same file.

Below, you can specify the date format and position.

> **NOTE:** File name incrementing is valid only for Schedule mode, for more than one capturing session (daily or weekly repetition).

The **AutoSave** page allows enabling automatic storing of capture lists. In it, you can also define the auto-save period in minutes by checking the ☑*Enable List AutoSave* box and entering the period in minutes.
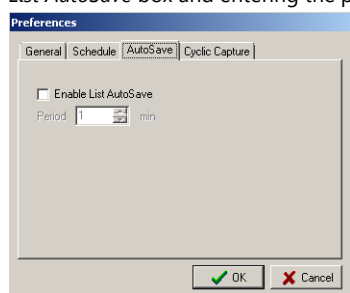
The **Cyclic Capture** page allows the user to define a number of cyclic files, which will be used for capturing. Thus, if you check the ☑*Enable Cyclic FileName Capture* box and enter, for example, [20] in the *Cyclic Files Count* spin-box, **CaptureBox** will only use 20 file to store the captured video. When it reaches the 20th file, it will go back to the first and start overwriting data.
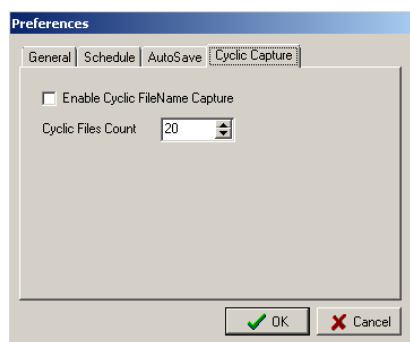
RS-422 Config

This command shows the volume mixer of the available audio device.

Audio Mixer is used in Windws XP when is used MATROX DIGISUITE - it open Windows Audio Mixer. We don`t use Windows XP and Matrox board so it is good to remove Audio Mixer menu from CaptureBox menu.

Exit

Click it to close **CaptureBox.**

View Menu

Use this menu to open some additional windows to the main **CaptureBox** window:

Preview Window

You can use it to monitor the video currently present on the encoder input.

Sony transport

This command shows the *RS422* controller window. The RS422 controller is described profoundly above, in the User Interface section.

FireWire Transport

This command shows the *FireWire (IEEE – 1394)* controller windows.

Volume Meter

Shows the *volume & peak meter* for the currently captured audio. The Volume & peak Meter is described in more details in the **AirBox** chapter above.

Timer

This command displays the system time window.

CPU Monitor

Capturing on some hardware platforms is quite CPU-intense. To prevent poor encoding and frame-dropping, **CaptureBox** has an automatic protection which will stop the capture if the CPU usage goes above 85%. This monitor will help you predict such possible situations and set your PC prior to starting the capture.

A drop-down menu activates on right-clicking in this window. In it, you can select which CPU to view (if there is more than one). Besides, you can view all CPUs simultaneously, or an average value of the CPUs' usage.

> **(!) TIP:** You can arrange all windows together by pressing **<Shift+F12>**

Capture Menu

Compensation

It invokes a dialog box for adjusting the capturing delay compensation (in frames). The capturing delay is hardware specific, so the values in this dialog are determined after the method test and mistake.

Setup

It opens a capture setting dialog box. This command duplicates the **SetUp** button situated under the batch capture grid. The dialog box is different, according to the encoder used. See the Capture Settings section to view the description of some encoders' setting dialogs.

Device select

It opens a dialog box for choosing the capture device.

Capture

It is active only when you work in Capture mode (the Capture tab is selected). It provides three options that duplicate the relevant buttons under the clip data fields:

- **Batch Capture** starts the batch capturing (following a predefined list)

- **Auto Capture** starts auto capturing – the program will control the VTR

- **Manual Capture** starts recording the currently available video source. A dialog box will appear prompting for your confirmation.

Schedule

It is active, only when you are working in *Schedule mode*.

- **Arm** – activates the schedule.

- **Prepare** – activates manual capturing from the currently available video source. A dialog box will appear prompting for your confirmation.

- **Crash** – starts capturing immediately from the currently available source, without asking any further confirmation.

TITLEBOX

GETTING STARTED

The **TitleBox (TB)** module is an on-air graphics manager. You can create different static or dynamic objects in **TitleBox**, such as rolls, crawls, still pictures, clocks, etc., and save them in projects. The projects could be used directly for broadcasting or as customized templates.

In **TitleBox** you can also start objects from previously created project(s) at different times, thanks to its **Scheduler.**

TitleBox works together with the **AirBox** playout. Separate objects in **TitleBox** can be started or stopped via **AirBox**, by inserting *TitleBox Net control events* in an **AirBox** playlist.

For more information about an **AirBoxTBNetCTrl** event, please see AirBox Settings->Modules->RemoteControl->TitleBox NetControl chapter.

Quick Start

1. Launch **TitleBox**;
2. Click the **Crawl** button in the object palette in the left;
3. Draw a rectangle in the **Preview Area**;
4. Type a text in the **Properties** dialog box and press **OK**;
5. Press the two **Play** buttons – one in the bottom of the window and one in the third row of the taskbar.
6. Congratulations! You have just created your first **TitleBox** crawl!

Manage TitleBox Instances

You can use more than one **TitleBox** instance if you are licensed to. The number of **TitleBox** instances you can use depends on the number of licenses you have. You can create shortcuts for the different instance numbers. This is done in the following way:

Assume that you are allowed to use three different **TitleBox** instances – **TB instance #1**, **TB instance #2**, and **TB instance #3** and that you want to add a desktop shortcut for each separate instance. You have to do the following:
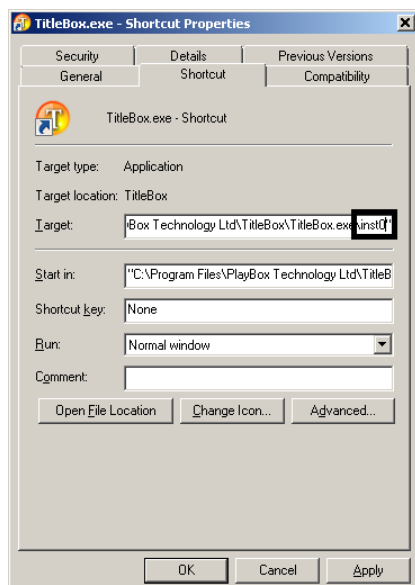
1. Create three **TitleBox** shortcuts on your desktop.
2. Right-click on the first shortcut and in the *Target* field of the shortcut add "\inst0" at the end of the line, as shown in the image to the right. This will associate this **TitleBox** shortcut to **TB Instance#1**.
3. Click OK and repeat the above step for the other two shortcuts. Enter "\inst1" and "\inst2" respectively for **TB Instance #2** and **TB Instance #3**.

For **TB Instance #[n]**, you need to enter "\inst[n-1]" in the *Target* field of the Shortcut Properties.

USER INTERFACE
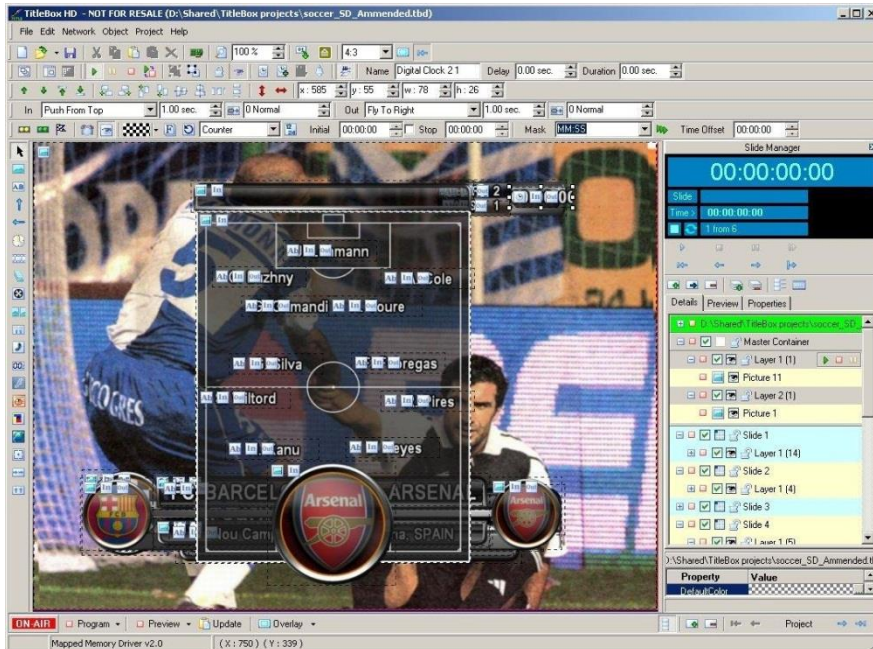
Work Area

TitleBox interface is very user friendly and it can be used easily in a live environment. The main part of the user interface is the *Work area*. It shows the objects, their positions, types, and status.

Objects can be moved and resized in this area. If an object is too complex and is not rendered yet, a yellow/black sign, saying **[Rendering]** will appear on the screen. This sign will automatically disappear when the rendering is complete.

The little icons in the upper-left corners of the objects show their type, lock status, and transitions. You can enable/disable the icons from **Project** menu ➔ **Options** → **General** → ☑ *Show Objects Icons*.

The color bar at the bottom of the work area displays the current mode of operation. By default, the preview mode is outlined in blue. You can change this color in the **Project** menu ➔ **Options** → **General** tab.

Channel in a Box User Manual

Toolbars

The **Toolbar** is designed to facilitate the project management, the individual object control, as well as the object ordering and alignment:
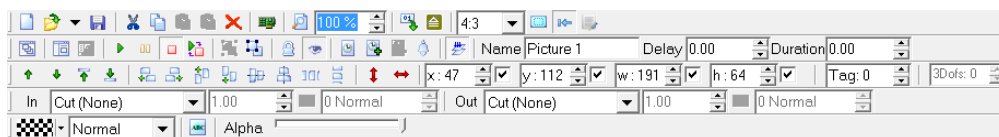
The **New, Open,** and **Save** buttons correspond to the relevant commands in the **File Menu**. The **Cut**, **Copy**, **Paste**, and **Delete** buttons correspond to the relevant commands in the **Edit Menu.**

Paste as Copy/ In Slide will create a new instance of the object that is copied in the clipboard. Unlike the simple **Paste** that creates a new object, **Paste as Copy** does not create objects. All changes applied to the originally copied object will be applied to all its instances.

The **Select Display device** button opens the list of available hardware drivers, from which you can select the one to work with. See also the corresponding section in the Project Menu description.

> **TIP (!)** If you want to change the driver before its initialization, hold down <**Ctrl + Shift**> during **TitleBox** startup to open the **Driver select** dialog.

The **Zoom** spin-box allows resizing the work area, so you could see the whole project at a glance. It is very useful for viewing HD projects.

This button opens the **Data Source Manager** window. It contains all supported data providing plug-ins. You can use them to "translate" different types of data and display the information they contain in some of the **TitleBox** objects. Please, check the Data Source Manager section further below.

This button opens the **Task Manager** window. Please, check the Task Manager description in the related section further below.

When the **Output aspect ratio** is changed, **TitleBox** will automatically change the size of the objects according to the selected aspect. All of the aspect changes concerning one particular project are listed as presets in the **Display resolutions** dialog. To open it, press the **display resolution** button. A dialog will appear. It provides possibilities for fast switching between the different presets. Pressing the button or double-clicking one of the rows will initiate an aspect change. To delete a preset, use the button.
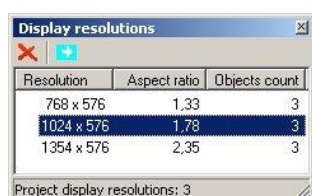
The **External aspect control** button is pressed by default. It is intended for automated switching of the aspect ratio according to an external source. Such a source can be WSS present in the incoming video, for example.

The **As-run log ON/OFF** button allows you to enable/disable the **TitleBox As-Run Log** application. The **As-Run Log** is configured in **Project menu ➔Options⇨As-Run Log**.

The **Objects list** button shows a window with a list of all objects, available in the current **TitleBox** project.

The **Properties buttons** are object-related. They allow for viewing or changing the properties of the currently selected object.

There are two types of properties: **Standard Properties**, that provide the standard object options, and **Property Tools**, which provide an additional set of properties. The **Property Tools** are active only for texts, rolls, and crawls.

> **IMPORTANT:** The properties can be interactively changed at any moment, even when the object is running on-air.

Channel in a Box User Manual

The **Play**, **Pause**, **Stop**, and **Toggle Play/Stop** buttons ▶ ⏸ ⏹ 🔁 control the play status of any particular object. They correspond to the relevant commands in the **Object Menu.**

The **Group** and **Ungroup** buttons 🔲🔲 are intended for grouping and ungrouping a selection of objects.

The **Lock** and **Visible** buttons 🔒 👁 define the object status. It could be locked for moving and resizing or not. It could also be visible in the preview area or not.

The **Schedule** buttons 🕐 🕐 🕐 concern project scheduling. The scheduling function allows you to start each object in a project at a specific time/day of week, with specific parameters. You can read more about scheduling in the Scheduler section.

The **Schedule window** button 🕐 opens the scheduler.

The **Add to Schedule** button 🕐 adds the selected object to the schedule.

The **Remove from Schedule** button 🕐 removes the selected object from the schedule list.

The **Show Events List** button 🕐 will open a list of all events, assigned to the currently selected object. See also the Task Manager description.

The **Mix Draw Mode** button 🔀 defines the draw mode of the object. If the button is pushed, two or more overlapping objects will be blended.

> **(!) TIP:** When in **Mix object** mode, you can switch between the overlapping objects using the *<Alt>* + *click* combination. The name of the currently selected object will appear in the *Name* field and will be highlighted in the Object list window. You can activate this window from within the **Object** menu.

The *Name* field contains the name of the object. The default names are [Type]_[#], i.e. [Crawl 1], [Roll 3], etc. You can change them at your will. The names are most important when you use the **Objects list** window, where only names and properties are displayed. In order to change the object name, just select the object, and then enter a new name into the *Name* field.

In the *Delay* field you can specify a delay for each individual object. A three-second delay means that the object would start three seconds after you have pressed the **Show object** ▶ button.

In the *Duration* field you can specify the duration for an object, if needed. The duration determines how long the object will be displayed after pressing the **Show object** ▶ button.

The **Order** buttons ↑ ↓ ↟ ↡ allow defining the objects' order in case they overlap. Overlapping dynamic objects is not desirable.

The **Alignment** buttons 🔲🔲🔲🔲🔲🔲🔲 - allow for aligning objects to each other. These buttons work when more than one object is selected.

The **Lock buttons** ↕ ↔ are used for locking the horizontal or vertical (or both) sizes of the object.

The **X** and **Y** numeric fields stand for X and Y positions of the selected object in the 2D space.

The **W** and **H** numeric fields stand for Width and Height of the selected object.

> **NOTE:** You cannot change some objects' size during their running on air.

The *Tag* Tag: 0 spin-box is used for additional object control. By default, an object's tag is set to [0]. However, you can enter another value here. The idea of the tag is that you can simultaneously operate with many objects that share the same tag number. This could be done via the script task and the program script task.

The *3D Offset* spin-box allows you to set an offset to the selected object in terms of depth. If you set a positive number, the object will appear deeper in the screen and vice versa.

Channel in a Box User Manual

The transitions toolbar will help you in setting your objects' *In* and *Out* transition effects:
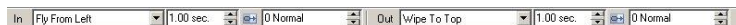
First, select the transition effect from the *In* and *Out* drop-down list. The available types are [Cut(None)], [Cross Fade], [Fly] and [Wipe]. They are not active for **Analog clock** objects.

In the spin-boxes next to the *In/ Out* fields set the duration of this transition (by default it is 1.00 second).

NOTE: Increasing the figures in these fields will decrease the speed of the transition.

Finally, define the type of motion during the transition (ascending, descending, or constant). By default, the motion type is set to [0 Normal]. This means that the transition will proceed at constant speed.

For example, if you have set a [Fly From Left] effect and duration of 2 seconds, your object will move at a constant speed from left to right till it reaches the desired position within two seconds. If you set the motion type to [(1 to 5) Ascending], the object will start in a slower motion and its speed will accelerate during the 2 seconds of transition. In descending mode, the speed of the transition will decelerate.

The **Full screen transition** buttons are situated next to the transition duration spin-boxes. You might need this functionality in case there are several grouped objects on the screen and these objects have the same transition effect and duration. If you trigger or stop such objects simultaneously, they will move in relation to each-other, because they have to cover different distances on the screen for the same period. Using this button will make objects move together, but not in relation to each-other, during the transitions. The transition of all grouped objects will look as if the entire screen is moving, not the separate objects.

The lowest toolbar contains object-specific settings that change depending on the currently selected object.

If no object is selected, this toolbar allows for changing the work area and the grid.

Push the **Token** button if you need to be more precise while drawing the objects in the work area.

You can change the colour of the drawing canvas from the *Work Space* drop-down palette. To change the colour of the surrounding area, use the *Back* palette.

The ☑*View grid* box is checked by default. If you do not want to view the grid dots, uncheck it.

TIP (!) You can adjust the distance between the dots in the **Project** menu ➔ **Options** ⇨ *General* tab.

If the ☑*Snap to Grid* box is checked, all objects in the project will be aligned to the nearest grid point.

If you want to load a picture as a background in the work area, you can browse for it after pressing the button.

NOTE: Currently, it is not possible to unload the background file, unless you close *TitleBox*. You can just change it.

TIP (!) You can take a snapshot of the current output (fill and key) from *TitleBox* by pressing **<Ctrl+F12>** keys on your keyboard. Then, you can paste the shot into a picture editor and save it to a file.

Object Palette

The **Object palette** contains buttons for all supported graphics objects. Click on the object you need and draw a rectangle in the work area to create it Press this button to select object(s) by dragging a rectangle around them.

Create a Still picture object.

Create a Text Template object (with background).

Create a Roll object (vertically running text).

Create a Crawl object (horizontally running text).

Create an Analogue Clock object (with custom background and clock hands).

Create an Animation object (a sequence of 32-bit .*tga* files, or animated .*gif* files).

Create an Animation File.

Insert Direct Show Media Source.

Insert a Banner object.

Create a Chat note object.

Create a sound object (any DirectShow-supported sound formats – .*mp3*, .*wav*).

Create a Digital Clock object (with custom background and font).

Insert a flash object.

Insert a Power Point presentation.

Create a Primary shape – squares, ovals, triangles with outline offsets.

Create a browser object.

Create a Screen capture object.

Create a chat line

Create a chat roll

Create 3D picture

Create 3D text

Insert 3D Mesh object

Channel in a Box User Manual
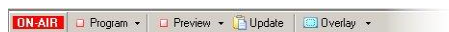
System Bar

The system bar is situated at the bottom of the interface window. It contains buttons for hardware control:

[ON AIR] – playout notification filed. It blinks red when **TitleBox** is on air (i.e., the Program driver is playing).

Program – this is the button that controls the program output driver. Pressing it will start/stop the graphics frame buffer. If it is not running, no graphics will appear in the output. Pressing the arrow to the right of it opens a drop-down list that contains several items:

Stop/Play– stop or play the program output driver.

Clear will erase the graphics frame buffer, so there will be no remnants from any previously loaded projects.

Program Driver SetUp – opens the settings options of the currently selected Program driver.

Driver manager opens the list of available drivers, so you could choose the one to use for outputting the graphics or a driver to preview it.

Preview – this is the button for controlling the selected preview driver. While the preview driver is playing, the bar under the work area will turn blue. A text message in it will notify you that the Preview driver is running.

The preview mode allows you to make changes in your project, and view them in a preview window without actually applying them on the output.

When the preview driver is stopped, all changes you make in the project will be applied in real time to the output.

Pressing the arrow to the right of the **Preview** button opens a drop-down menu. It contains the same items as the ones of the program driver.

> **NOTE:** In preview mode you cannot create a Flash object! You can just edit already existing objects!
>
> **NOTE:** Some objects might not appear properly in the simple preview window!

Update - While in preview mode, press this button to apply the changes to the program output.

Overlay – It is not in use for the current **TitleBox** version.

Status Bar

The status bar displays information about the currently selected driver and the X/Y position of the pointer in the work area:

\\192.168.60.218\Soft\mp2\CaptureBox\CB_help.chm

Channel in a Box User Manual

MENU BAR

File Menu

This menu contains commands related to the project's file:

New – allows creating a new project.

Open – opens an existing project (*.tbd*).

Reopen – shows a list of the ten recently open projects.

Merge – merges the current project with another one.

> **NOTE:** Global layer links might not operate properly because the Master container of the second project will appear as a simple slide in the merged project. You will have to drag its layers to the merged Master Container and set the layer links again.

Save – saves the current project to the open file (*.tbd*).

Save as – saves the current project to a specified file (*.tbd*).

Edit Menu

The Edit menu contains object-related commands:

Undo – use it to undo **Move**, **Size**, and **Create**. Please, note that there is no undo for deleting objects.

Cut – removes the selected object(s) and keeps them in the buffer-memory.

Copy – saves a copy of the selected object in the buffer-memory.

Paste – pastes the buffer content in the project.

Delete – deletes the selected object.

Select All – selects all objects in the *Preview* area.

Network Menu

This menu concerns the connection between *AirBox* and *TitleBox*.

> **NOTE:** This feature is available for the *TitleBox* full version only. It is not available in the *TitleBox* light version.

For detailed step-by-step guide on how to integrate *TitleBox* with *AirBox* look up in APPENDIX 3 – Integration of AirBox with TitleBox

Export Project as Template – exports the currently open project as a template. It will be saved in the specified Template folder. You have to specify this folder during *TitleBox* Net Control activation (*see Appendix 3 for details*).

> **IMPORTANT:** The **project must be exported as a template,** so that it could be controlled through *AirBox.*

Net control – shows the connection status between *AirBox* and *TitleBox*.
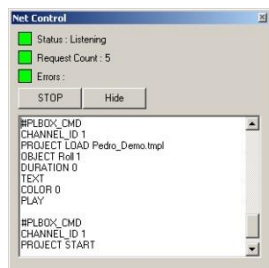
> **IMPORTANT:** In order to be able to control *TitleBox* through **Net Control output** events from *AirBox*, you have to switch *TitleBox* in **Net Control** mode (*TitleBox* **Network** menu➜**Net Control**).

If you want to control a *TitleBox* project (and its separate objects) from *AirBox*, you have to export it as a template (*TitleBox Network* menu➜**Export project as template**). The template folder will open automatically, so you do not have to browse for it. Just type a name and press *Save*.

Object Menu

This menu contains object-related commands:

> **Play**- shows and runs the selected object on the graphics frame buffer.

> **Pause**- freezes the object but it remains displayed on the graphics frame buffer.

> **Stop** – hides the object from the graphics frame buffer.

> **Add to Scheduler** – Adds the selected object to the Schedule.

> **Remove From Scheduler** – Removes the selected object from the Schedule.

> **Scheduler Properties** – Shows the properties of the schedule

> **Order** – This function is active when there are overlapped objects. You can move the selected object under or over the others.

> **Alignments** – This function is active when more than one object is selected. It allows for aligning the objects to each other.

> **Assign Task -** this function invokes the *Task Manager* for the selected object. More detailed information you can read in section *Task Manager*, further in that manual.

> **View Events** – select it to view all tasks, assigned for the currently selected object. This menu item corresponds to the *View Events List* button in the toolbar .

> **Object list** – shows a window with a list of all objects available in the current *TitleBox* project (see its description further in this chapter).

> **Property** – opens the *Properties* window of the currently selected object.

> **Property Tools** – opens additional properties windows for objects (text, rolls, and crawls).

The commands from this menu (except for Assign Task) are also displayed in the second uppermost row of the toolbar.

Project Menu

This menu contains project-related commands. Some of them (**Play**, **Pause**, etc.) are also operated through buttons, situated along the bottom of the window.

Channel in a Box User Manual

Project playout commands

Play– starts the driver of the graphics frame buffer. If this button or command is not triggered, you will not be able to show any graphics on the screen.

Stop – hides all objects from the screen but they remain loaded in the graphics frame buffer.

Clear Buffer– clears the graphics frame buffer. This command helps avoiding accidental showing of remnants of an old project when loading a new one.

Scheduler – Opens the scheduler window for setting up the scheduled graphics insertion.

Mix Objects – This command is a duplicate of the *Mix Draw Mode* button. It will blend the overlapping objects in the project.

Driver Select
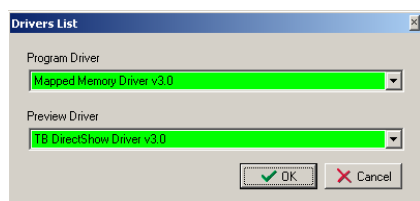
Click on it to view the list of available hardware drivers and select the one to work with. If there are no hardware devices installed, the list will contain the following lines:

[Preview Output Driver]- It is used for previewing the **TitleBox** project in a specially designed software preview window.

[Mapped Memory Driver] – This driver is used, when **AirBox** and **TitleBox** are being used in the same playout system. It allows keying the graphics over the video played in **AirBox**.

[TB DirectShow Driver] – This driver is used when there is a **Blackmagic DeckLink** card installed on the machine.

You have to choose the *Program Driver* and the *Preview Driver* separately.

Please, check the **PlayBox** web site (https://www.playboxtechnology./com) or contact **PlayBox** support team (support@playboxtechnology.com) for currently supported boards.

Driver Setup

Press it to change the output settings. The available options depend on the currently used driver.

Mapped Memory Driver settings
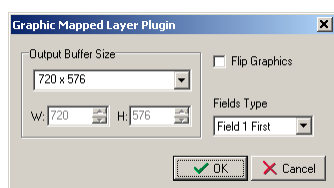
Output Buffer Size – select the output resolution from the drop-down list. If you select a Custom resolution, then the next *Width* and *Height* fields will become available for enterenig your resolution values.

☑*Flip*– if you check this box, the output graphics will be flipped at 90 degrees.

Fields type – select the field order from the drop-down list.

Press the *OK* button to save the changes.

Channel in a Box User Manual

TB DirectShow Driver settings

If **TitleBox** works as a standalone application, you have to select [TB DirectShow] driver. The following windows will appear:
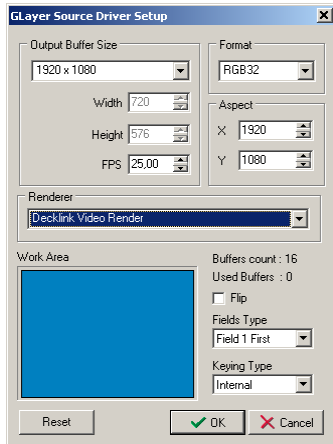
Output Buffer Size – this is the output **TitleBox** resolution.

If you select a *Custom* resolution in *Output Buffer Size*, then the next **Width** and **Height** fields became available for enterenig your resolution values.

In *Aspect X/Y* fields enter the aspect for the display resolution.

FPS stands for *Frames per second* setting.

Format is a video formats - RGB, UYVY, etc.

Renderer – this drop-down list shows all available direct show renderers on this machine. Select the **DeckLink video renderer** here.

Work Area – it is not in use in the current version.

☑*Flip*– check the check-box, in order to flip the output graphics at 90 degrees.

Fields Type – select the field order here from the drop-down list.

Keying Type–here you can select the type of the keying (or no keying). Some support either "Internal Key" or "cc" only. Other DeckLink cards support both keying types.

Internal keying means that the video input, keying, and resulting video output, are all performed internally within the one DeckLink workstation.

External keyingis used when you want your alpha channel key, and the video graphic fill, to output through separate DeckLink SDI outputs for use with an external video mixer.

Please, refer to the DeckLink documentation for more information about supported keying.

<u>**Preview Output Driver settings**</u>

This driver is used only for preview. It is not meant to be used as an output driver.
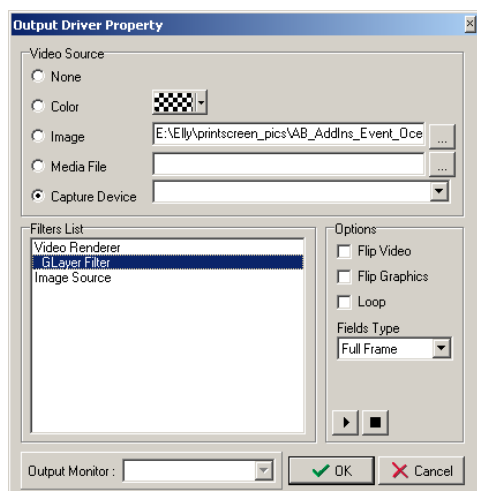
In the *Video Source* field you can select the background, on which **TitleBox** graphics will be overlaid. Just press on the radio button that corresponds to your needs. For **Image** and **Media File** you have to use the browse button and enter the file-path of your desired file, and for **Capture Device** you should select the device you are using from the drop-down menu.

From the *Filters List* field select the appropriate filter.

The *Output Monitor* field becomes available if you have more than one monitors, connected to your machine. Then you can select which monitor to show the preview window.

The *Options* area allows you to flip your video or graphics by 90 degrees by checking the relevant *Flip Video* and *Flip Graphics* boxes. If you check the ☑*Loop* box your Project will be looped.

The *Fields Type* drop-down list allows you to select whether to preview the **Full Frame**, **Field 1 First**, or **Field 2 First**.

Finally, the **Play** and **Stop** ▶ ∎ buttons at the bottom allow you to initiate or terminate the preview of your Project.

Plug-ins

This item shows the list of available external plug-ins:

<u>**File link**</u>

Opens a list of all objects in the project and their links, if any, to external files. Any changes in those files are applied in real time. The file links apply to the text and picture objects only!

<u>**Task Manager**</u>

Select this menu item to create a list of tasks, related to objects in the current project. Check the detailed description in the **Task Manager** section.

<u>**GPI Manager**</u>

In the GPI manager, you can view all GPI events and the objects, to which they have been assigned. To specify GPI events, you have to run **Task Manager** and make the necessary definitions there.

<u>**Active Event (Tally)**</u>

This function keeps you informed whether there are any graphics showing on the screen or not. If there is at least one object that is being played, a high pulse is sent to the specified *COM* port. When there are no objects played, the pulse goes low. Thus, any external GPI device can be activated when there are no objects played out in **TitleBox** (i.e., a sound alarm to let the operator know that there are graphics displayed). After choosing this menu item, you

can select the desired *COM* port or [None] if you do not want to send out any pulses. Furthermore, you can specify the signal type (RTS or DTR). RTS will send a pulse to pin 7, and DTR – to pin 4. Pin 5 is the ground. For more details, please, refer to the GPI pin out description in <u>Appendix 1</u>.

DataSource Manager

This plug-in opens the **Data Source Manager**. It contains all supported data providing plug-ins. You can use them to "translate" different types of data and display the information that they contain in some of the **TitleBox** objects. Please, check the **Data Source Manager** description <u>further below</u>.

Options

Press it to open the **Project Options** window. It contains the following pages:

General Options:
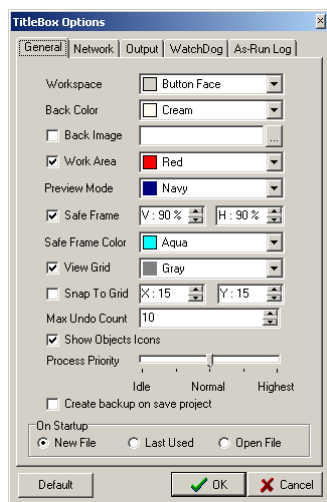
These options allow for defining the colors, the safe area, and the grid of the *Work area:*

     Workspace – set a color for the area that surrounds the workspace.

     Back Color – set a background color for the workspace.

     ☑ *Back image* – you can select an image for background via the browser ⬚ button.

     ☑ *Work Area* – set a color for the borderline of the work area.

     Preview Mode - set a background color for the workspace, when it is in preview mode.

     ☑ *Safe Frame* – check it to view the safe area in the workspace. You can adjust its size by using the [H] and [V] percentages to the right.

     Safe Frame color – choose the color for the safe area border.

     ☑ *View Grid/ Grid color* – check the box to see the grid and select its color. The grid is very useful when you resize your objects.

     ☑ *Snap to Grid* – check this box to align the objects in the project to the grid. Use the [X] and [Y] spin-boxes to define the distance between two neighboring points in the grid.

     Max Undo Count – specify the number of latest actions to be saved in the undo history. Please, note that the higher number here means higher memory usage.

> **NOTE:** You cannot undo removing objects!

     ☑ *Show Objects Icons* – check this box to enable showing the objects icons in the upper left corner of each object in the work area.

     Process Priority – this slider will affect all other programs running on the system. Please, do not change the default setting, unless advised so.

☑ *Create backup on save project* – check this box to if you want **TitleBox** to create a backup file of your project every time you save it. The backup file will be saved in the same folder, in which your current project is saved, and it will be with extension **.backup*

On Startup – here you can select the start mode of **TitleBox.**

Network Options
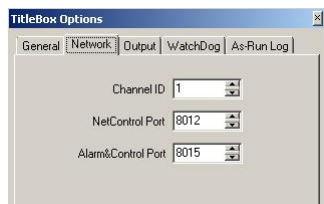
These options are used for interaction between **TitleBox** and **AirBox**.

TitleBox Channel ID – **TitleBox**'s channel ID; it corresponds to **AirBox -> Settings -> Modules -> Remote control** [TitleBox Net Control] → **Configure**[Channels].

NetControl Port – select a network port for receiving commands. This is the port, at which **TitleBox** receives commands from third-party applications (like **AirBox Net Control** commands)

Alarm&Control Port – select a network port for communication with **PlayBox** applications (like **AlarmBox** or **TitleBox Text Client**).

Output Options
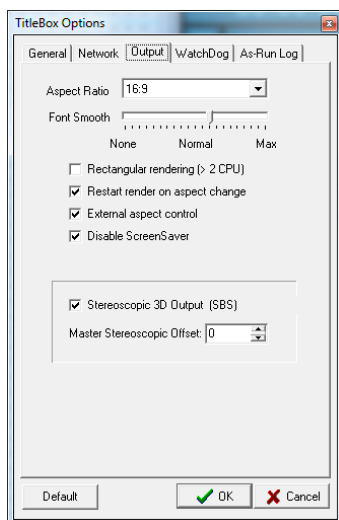
Here you can set the aspect ratio of the output. The default setting is [4:3]. The slider below determines the smoothness of the font characters. The greater the value, the smoother the font is.

☑ *Rectangle rendering* is recommended for more powerful systems. It changes the algorithm of graphics rendering for optimizing the performance. Do not use it on single CPU machines, as it increases the CPU load!

IMPORTANT! This option with is NOT recommended with animation objects and Direct Show Media objects!

☑ *Restart render on aspect change* – check this box to restart the rendering when the aspect ratio is changed.

☑ *External aspect control* - it is intended for automated switching of the aspect ratio, according to an external source. Such a source can be WSS present in incoming video, for example.

NOTE: The ☑*External aspect control* check-box is related to the **External aspect control** button on the **TitleBox** toolbar.

☑ *Disable ScreenSaver* – disables the screensaver on the machine, on which **TitleBox** is in use.

☑ *Stereoscopic 3D Output (SBS)* – enables the **TitleBox** stereo 3D output. Use the *Master Stereoscopic Offset* spin-box to assign the overall offset of the 3D output. The offset here measures the depth of the objects. If you set a positive number, the objects will appear deeper in the screen and vice versa.

NOTE: You can assign object-specific 3D offset values to each object by using the <u>corresponding option</u> in the main **TitleBox** toolbar.

WatchDog

The **WatchDog** plug-in is used in **TitleBox** systems working on pass-through mode. Its purpose is to switch the by-pass relay card in case of **TitleBox** crash, in order to bypass a video signal.

The WatchDog works with a dedicated **PlayBox Bypass relay** card. You can read more about this card in <u>Appendix 4</u> in this manual.

You need to have an installed **PlayBox Bypass** relay card in order to configure the WatchDog plug-in.

When you press the **Configure WatchDog** button, the following window opens.
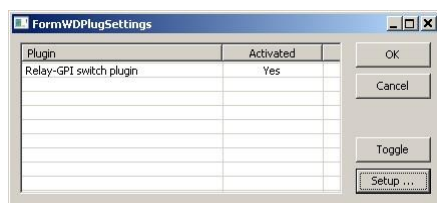
Activate the plug-in first by double-clicking on the entry under the **Activated** column next to the corresponding **Plugin**, so that the [No] text in it turns to [Yes]. Then press the **Setup** button to open the set-up window:

In the *Device Select* field you will see the list of the installed **Bypass Relay** cards. Select the card you need and press **OK**.

IMPORTANT: Even if there is only one Bypass Relay card installed, select it from the *Device Select* field and press the **OK** button, for the setting to take effect.

As-Run Log

Use this tab to enable and configure *TitleBox* logging. Be aware that an as run log file is configured and saved per *TitleBox* instance and it is not larger than 2 GB.

The first three check-boxes allow you to do the following:

☑ *As-Run Log Enabled* – check this box to enable logging. You can also enable/disable the <u>As-Run Log</u> via the corresponding <u>button</u> in the *TitleBox*
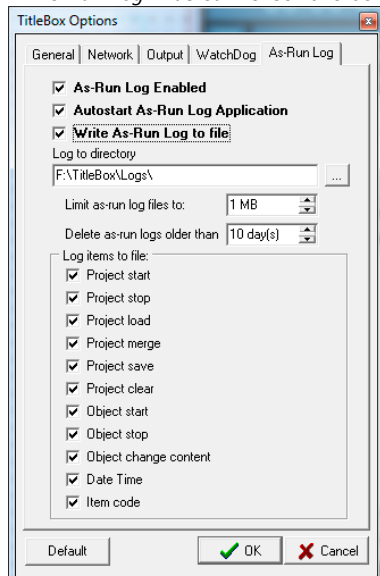
toolbar.

☑ *Autostart As-Run Log Application*– check this box if you want the As-Run Log to start automatically when *TitleBox* is initiated.

☑ *Write As-Run Log to file*– check this box if you want to save your As-Run Log to file. Notice that once you enable this option, some additional options appear in the dialog.

In the *Log to directory* field enter the folder, where you want your log file to be saved.

The name of the logged file will be in the following format:

TBRL_YYYY-MM-DD_HH-MM-SS_#n, where:

YYYY-MM-DD_HH-MM-SS is the date and time when the first command in the file has been sent to the *As-Run Log* and #n is the respective *TitleBox* instance number.

For example, TBRL_2012-06-19_15-23-01_#1.log is a log file, in which the first command has been sent on June 19th, 2012 at 15:23:01 o'clock from *TitleBox* Instance #1 to the *As-Run Log*.

The *Limit as-run log files to* spin-box allows you to set a limitation for the log file size in Megabytes. The minimum value that you can enter here is 1 MB, and the maximum is 2000 MB.

Be aware that the log files are saved per instance and per day, meaning that when the current date changes a new log file, for the next date, will be opened by the *As-Run Log* for writing. New log files are created also if the limit, set in the abovementioned spin-box is reached. The last case when the *As-Run Log* will create a new log file is if it is manually shut down by the user and reactivated by *TitleBox*.

Use the *Delete as-run logs older than* spin-box to automatically delete log files that are older than the configured number of days. The minimum number of days here is 1 day and the maximum is 180 days.

In the *Log items to file* area place a check next to the events, which you want to appear in your log file. The options here are self-explanatory. If you press the *Default* button only the options, which are checked in the picture above will be selected.

NOTE: These configurations refer only to the log file that is saved. The actual **As-Run Log** application displays all types of events.

Help Menu

This module gives the opportunity to generate easy-to-complete problem reports. It is integrated in each **PlayBox** module. It can gather almost all the information, needed for the **PlayBox** support team in order to provide you with the prompt answers, without too many questions about your system configuration.

The Basic User's manual contains a detailed description of the PlayBox Doctor Report and other functionalities. If you do not have the Basic manual, you can download it from our website.

Displays the **About** box of the **AirBox** module. It contains useful information, such as: module version, WIBU Box number, mode, registration, etc. The name of the currently selected platform is displayed at the bottom.

Help Index

Opens the **TitleBox** context-sensitive help. You can also invoke it by pressing <Shift + F1> key combination.

CREATING OBJECTS

Step By Step

All objects are created in a similar way:

1. Select the appropriate object button from the *Object Palette.*

2. Draw a rectangle in the *Preview Area,* where you wish to place it.

3. A **Properties** dialog box, allowing fine-tuning of non-text objects (Still picture, Analog clock, Animation, Digital Clock, DirectShow media, Banner, etc.), will appear.

4. Pressing **OK** inserts the created object in the project.

You can edit the properties of text-containing objects either by using the special toolbars that appear when double-clicking on the relevant object, or when clicking on the **Properties** buttons ▣ ▣ . The unique properties of all types of objects in **TitleBox** are described in detail in their relevant sections in the Object Properties below.

Editing Objects

You can edit objects either by invoking the object properties dialog box (for non-text objects), or by using the special toolbars (for text objects). Do this by double-clicking on the object, or by clicking on the **Properties** buttons.

Animations and **Clocks** CANNOT be resized in **TitleBox**, whereas **Pictures** and **Text templates** with graphics background can.

Crawls, Rolls and **Text Templates** without graphics background can be resized even on-air. You can resize all dimensions of Text objects, Crawl's width and text size, and Roll's height and text size. However, resizing a Crawl's height and a Roll's width should be done while the objects are stopped.

Deleting Objects

An object can be deleted by selecting it and clicking on the **Delete** button on the **Main Toolbar**. It can also be deleted by using the **<Delete>** key on your keyboard.

All objects can be deleted by clicking on the **New Project** button and clicking on the **No** button in the dialog that asks you if you want to save your changes.

Channel in a Box User Manual

Object List

The **Object list** is intended for fast switching and reviewing of objects and their properties, such as **Left** and **Top** side positions, **Duration** and **Delay**. Here you can easily **Group/Ungroup** objects, change their **Order** (z-order), and control their playout and visibility status by pressing the relevant buttons on the toolbar.
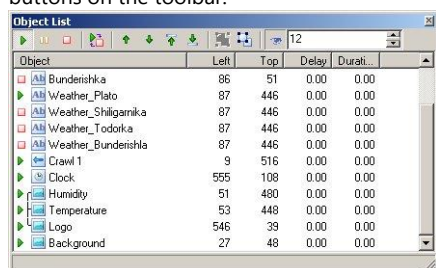
Object Properties
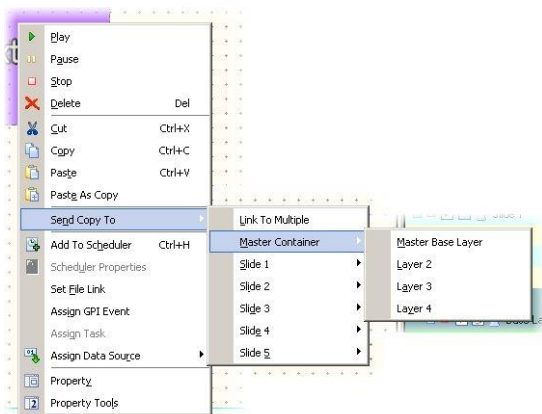
The **Object Properties** dialog boxes are different, depending on the object type. All buttons have specific pictures and provide hints when you slide the mouse pointer over them.

Right-clicking on any object invokes a menu, like the one shown in the screenshot to the left.

The upper half of the menu contains the most common status and editing commands.

The **Send Copy To** command allows you to easily create copies of the currently selected object of all slides and layers in your project.

The **Add To Scheduler** item will open the **TitleBox** Scheduler, where you can set the scheduling for the selected object.

If the object has already been added to the Scheduler, use the **Scheduler Properties** item to view the settings of all objects in the current project.

Set File Link – select this item to invoke a browse dialog to locate the text file (*.txt* or *.rtf*), to which you would like to link your object. Thus, your object's content will be updated every time the linked text file is saved. An additional window will open to display all objects and their file links (if any) in the current project. You can set/change the file links from within the **Links** dialog. Just press the browse button to the far right of the relevant row to open a new browse dialog.

If you right-click on the selected row, the following dialog will open:
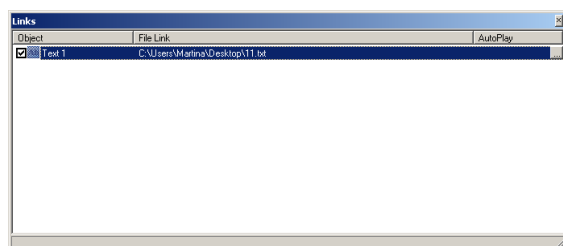
If you do not want to have a link anymore, just uncheck the **Active** row. When you create a new link, **Active** is checked automatically.

Channel in a Box User Manual

Pressing **Set File Link** will open a new browse dialog for selecting a text file for the link.
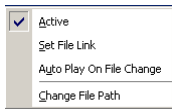
AutoPlay On file Change is used for starting the object automatically, if there is a change in the linked text file.

Change File Path allows for changing the file path to the linked file. It is used in case the linked file is moved to another location.

(!) TIP: When a **Roll**, **Crawl,** or **Text** object is linked to a text file (*.txt* or *.rtf*), you can insert a still picture in the text – the image will be displayed among the characters, according to the position of its script in the text. Write the following command in the text file:

<BITMAP>[file path of the image file]</BITMAP>

Make sure that BITMAP is written in capital letters. Here is an example:

Example 1 – Inserting a Still Picture in a Text File
1) Save your image. Let us assume the file name is "logo.bmp" located on D:\
2) Enter the text in a file, for example "Hello, this is a test project".
3) Continue writing the following: <BITMAP>D:\logo.jpg</BITMAP>.
Thus, your text file will be:
Hello, this is a test project <BITMAP>D:\logo.jpg</BITMAP>
It could also be:
Hello, this <BITMAP>D:\logo.jpg</BITMAP> is a test projector Hello, this is a <BITMAP>D:\logo.jpg</BITMAP> test project
Every time you edit the text and save the changes, the text on the output will be refreshed.

If you want to change the picture, change the file name and location part in the script (here: D:\logo.jpg).
If you want to insert an animated GIF file in the text, write <MOVIE> instead of <BITMAP>. For example:
Hello, this is a test project <MOVIE>D:\smilie.gif</MOVIE>

NOTE: There is no spell-checker implemented in *TitleBox* text objects (roll, crawl and text template). You can use some external application for spell-checking and then just copy and paste the text into *TitleBox* object.

Still Picture Properties
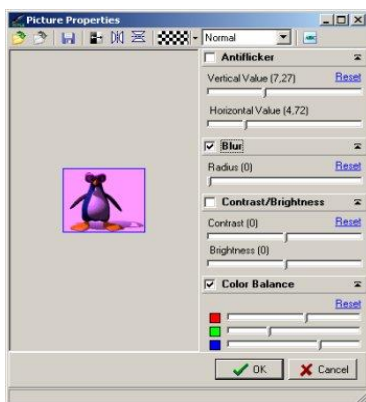
The **Picture Properties** dialog box looks like the one to the left.

The **Toolbar** allows you to *Open* and *Save* the image, *Load* and *Invert* the mask (the alpha channel), and *F* the image horizontally and/or vertically. If your picture does not have an alpha channel, you can import one separately, by using the *Open mask* button.

The *Draw Alpha Only* button provides a new, interesting option. Push this button to create a Picture object over other objects in the project. Thus, you will overlay its alpha over all underlying objects. Try to make effects this way!

At the right part of the window you can adjust the object's *Contrast*, *Brightness* and change its *Color Balance*. You can choose to *Blur* the object and set the blur radius by using the slider below.

The *Anti-flicker* option is designed for smoothing the high-contrast computer graphics when overlaying it over video. Change the *Vertical Value* to prevent flickering of the graphics' edges.

Text Object Properties

The Text objects have three groups of settings options:

Text and Background Properties

You can edit the common text attributes (such as font selection, size, color, blur, shadow, etc.) and background attributes from another object-specific toolbar:

This toolbar appears in the last row of the **TitleBox** toolbar when you double-click on a Text template object. Another way to invoke it is by pressing the Property button (in the second row of the toolbar) while a Text object is selected (single click).

Write the new texts directly in the object!

Pressing invokes the **FontDialogEx** dialog, which allows you to manage the font of the selected text. Here you can find all the formatting options, as known from other windows-based editing applications:
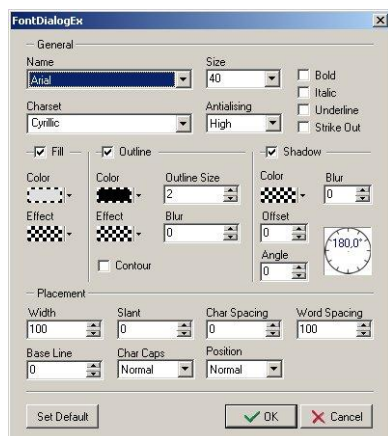
The *Set Default* at the bottom button restores the **TitleBox** native settings.

> **NOTE: *TitleBox*** supports all true type fonts.

Pressing invokes the **Paragraph**-formatting dialog box, where you can set all paragraph-formatting options.

The *Script State* and *Script Control* areas are related to the use of Arabic language and Arabic (Arabic-Indian) digits in the project.

The **Set Default** button at the bottom restores the **TitleBox** native settings.

The [image] button in the toolbar allows you to change the color of the background.

Press [image] to view a transparency background in the preview area while editing.

The following fields in the toolbar provide general text-formatting options (font, font size, bold, italics, underlined, text alignment, font color).

The [image] button inserts a still picture. Press it to open a browser dialog for selecting a picture image to be inserted inside the text into the text object.

The [image] button inserts OLE compatible objects.
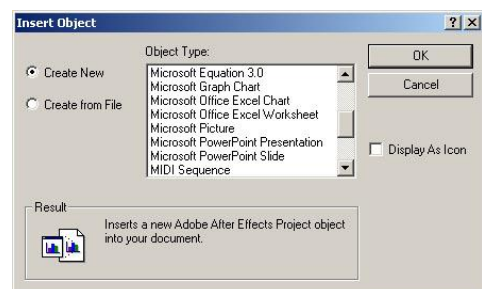
When you press it, the following dialog will appear:

If you choose ⊙ **Create New**, the relevant application opens, and you will be able to create the desired object. Any changes in the relevant **OLE** object will be visible in **TitleBox**, when saved.
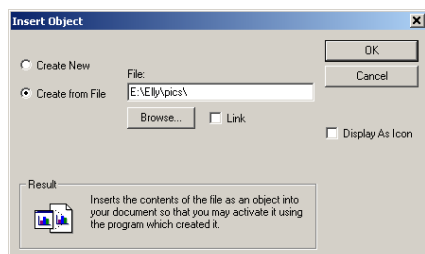
If you choose ⊙ **Create from File**, you will be prompted to browse for it. You can create a link, or display it as an icon through checking the relevant box.

The drop-down list for vertical alignment is used to manage the position of the inserted OLE objects into the text object.

Channel in a Box User Manual

Background properties

If you click once on a text object, the following object-specific toolbar appears in the last line of the *TitleBox* toolbar:

The button is related to the objects (OLE object or pictures), inserted into the text. Press this button to open the list of all objects that are already inserted into the text objects. The insertion of the objects is described in details in the previous section.
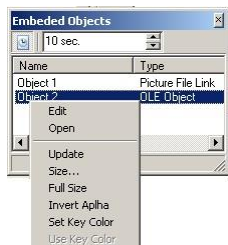
You can set an auto refresh period for the object. Press the clock button in the upper left corner to activate this function and set the refreshing period in seconds by using the arrows. Thus, if you update the original file, it will be refreshed automatically in your *TitleBox* project. In addition, right-mouse-clicking on an object in the list will invoke a context menu, containing several useful options:

Update – it is valid for OLE objects only;

Size... – you can set a custom size to your inserted item;

Full size – display the original size of the object;

Invert Alpha – inverts an object's alpha channel;

Update from file – it is valid for picture objects only.

If you want to open an image as a background, push the *Open image* button . Use the drop-down list to the right to fix its layout in the object (Normal, Stretch, Tile or Center).

Delete the background image by pushing the button.

Text Object Property Tools
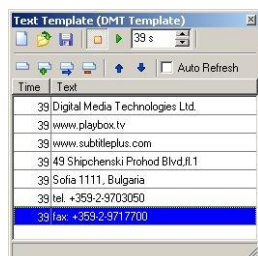
The property dialog box for text objects allows for displaying different texts consequently, for specified periods in seconds. The different texts are shown as a list of texts (separate lines) in the *Text Template* window – press the button to view it (or right-mouse-click over the object and select **Property Tools**).

You can prepare texts in advance or create them online. Use the buttons in this dialog to do the following:

New button – open a new text template.

Open button – You can open a previously prepared text file, by using the **Open** button. Each paragraph in the text appears as a separate line in the *Text Template* window.

Save button – saves the entered text as a file.

Stop button - stops displaying the text lines in the preview/output window.

Play button – starts displaying the text lines in the preview/output window.

Global Time spin-box – defines the frequency of changing the text lines. This is actually a "duration," applied for each new line added in the text template. If you need to specify a different duration for a particular line, use the spin-box in the text input dialog.

Edit item button – opens a dialog box for editing the selected text line.

(!) TIP: You can change the text by right-clicking on a text string as well.

Add item button – adds a text line. A dialog box opens for typing the text in it.

Insert item button – inserts a text line. A dialog box opens for typing the text in it.

Delete item button – deletes the selected text line.

Moving up – moves the selected text line up.

Moving down - moves the selected text line down.

☑*Auto refresh* box – automatically displays the changes made in the currently loaded text file, even during its play-out.

⬆ Roll/ ⬅ Crawl Properties

There are three groups of properties for these objects, similar to the text template: common text attributes; continuity mode and queue options; and dynamic speed properties. The first two are controlled through object-specific toolbars, while the third one can be set in the specially designated dialog box.

Continuity and Queue mode

When a crawl/roll object is selected (single click) the following string appears in the last row of the toolbar: .

The ***gear-wheel*** button opens the ***Embedded Objects*** list. This is the one already described for text objects. Please, check above in the Manual.

Speed *field* - controls the speed of dynamic objects, such as animations, crawls, and rolls. Its value can be positive (right-to-left movement) or negative (left-to-right movement). If zero, the object is frozen.

NOTE: You can change the speed interactively at any moment, even when the object is running on-air.

If the value in the ***Loop*** field is [0], the object will be displayed endlessly.

The next three buttons refer to the object's background continuity mode. The first one means that the background will be displayed only while the text is running. The second one will "glue" the texts one after another, without any space in between. The third button will display the background continuously.

The last two buttons in this toolbar concern the Queuing functionality. After pressing one of these buttons, ***TitleBox*** will "remember" the changes for background colors and text colors and it will play them one after another, i.e., you will be able to make a queue of color changes (loops). If none of the above buttons is pressed, ***TitleBox*** will show only the latest change. You can choose how to switch between changes (loops) while in play mode:

Use if you want to display the subsequent loops one by one, i.e., to leave some space between them (empty or with background color, depending on your settings – see the previous paragraph). In addition, use if you do NOT want any space between the successive loops in the object. ***TitleBox*** will generate one loop out of all loops in the queue.

Text and Background Properties

You can edit the common text attributes (such as font selection, size, color, blur, shadow, etc.) and background attributes from another object-specific toolbar:

Channel in a Box User Manual

It appears in the last row of the **TitleBox** toolbar when you double-click a Roll/Crawl object. Another way to invoke it is by pressing the Property button (in the second row of the toolbar) while a Crawl/Roll object is selected.

Write the new texts directly in the object!

> **NOTE:** There is no possibility to insert images as background but you can still insert images and OLE objects in the Roll/Crawl objects.

Pressing ⊞ invokes the **FontDialogEx** dialog, which allows you to manage the font of the selected text. Here you can find all the formatting options, as known from other windows-based editing applications:
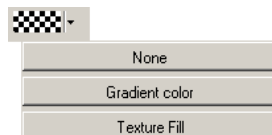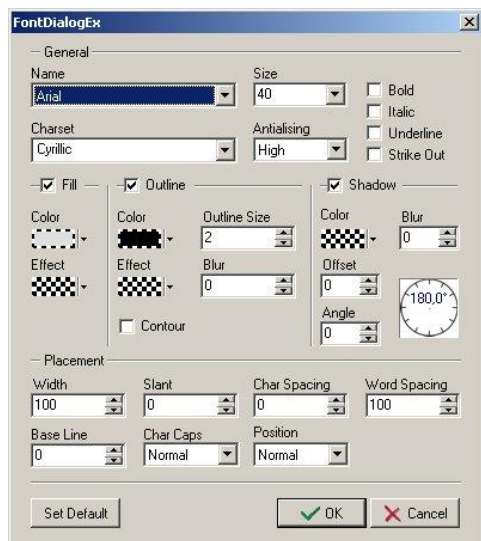
Use the **Fill Effect** button and its additional buttons to modify the colors for your text. You can select **None** for color, **Gradient color**, or **Texture Fill**.

If you click on **Gradient color** from the drop-down list, you can select from 255 levels of graded transparency and a vast variety of colors.

In the **Gradient properties** dialog, fix the desired settings:
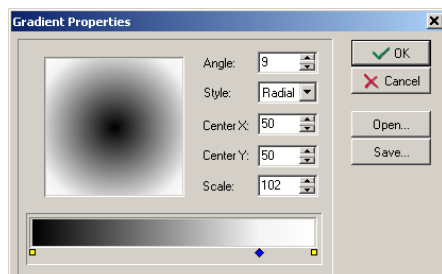
- Choose a Style for the gradient from the drop-down list.
- In the square field to the left drag the cursor to change the gradient positioning.
- In the rectangle field below, define the gradient colors and their initial points.
- Place the plus-sign cursor in the desired position and click. A black triangle will appear in that position to mark the currently selected point. All other marks will become white.
- Double-click on the black triangle to invoke the **Color setting** dialog.

194

- Modify the color for this color-change point at your will and click **OK**.

Back in the **Gradient properties** dialog, you can change the position of the point by dragging it. Also, you can add as many color-change points to the gradient, as you like. Finally, click **OK**.

Pressing [P] invokes the **Paragraph**-formatting dialog box, where you can set all paragraph-formatting options:
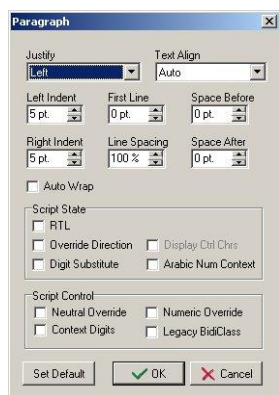
- The two buttons to the right concern the background. Press [icon] to view a transparency background during editing in the preview area.

- Text-formatting buttons - font, font size, bold, italics, underlined, text alignment, font color.

- Object links buttons.

- Push the [icon] button to insert a still picture object link.
- Push the [icon] button to insert OLE compatible objects. Use the drop-down list to fix their position within the Roll/Crawl object.
- The Text Template Properties section above contains a detailed description on how to import **OLE**-compatible.

(!) TIP: When a **Roll**, **Crawl** or **Text** object is linked to a text file (**.txt* or **.rtf*), you can insert a still picture in the text – the image will be displayed among the characters, according to the position of its script in the text. Write the following command in the text file:

<BITMAP>[file path of the image file]</BITMAP>

Make sure that BITMAP is written in capital letters. Please, check Example 1 <u>above</u>.
1) Save your image. Let us assume the file name is "logo.bmp" located on D:\
2) Enter the text in a file, for example "Hello, this is a test project".
3) Continue writing the following: <BITMAP>D:\logo.jpg</BITMAP>.
Thus, your text file will be:
Hello, this is a test project <BITMAP>D:\logo.jpg</BITMAP>
It could also be:
Hello, this <BITMAP>D:\logo.jpg</BITMAP> is a test project
or
Hello, this is a <BITMAP>D:\logo.jpg</BITMAP> test project
Every time you edit the text and save the changes, the text on the output will be refreshed.
If you want to change the picture, change the file name and location part in the script (here it is **D:\logo.jpg**).
If you want to insert an animated *.**gif** file in the text, write <MOVIE> instead of <BITMAP>. For example:
Hello, this is a test project <MOVIE>D:\smilie.gif</MOVIE>

NOTE: There is no spell-checker implemented in the **TitleBox** text objects (roll, crawl, and text template). You can use some external application for spell-checking and then just copy and paste the text into the **TitleBox** object.

Dynamic Speed Properties

Pushing the Property tools button [image] while a Roll/Crawl object is selected will open the Dynamic properties dialog box:
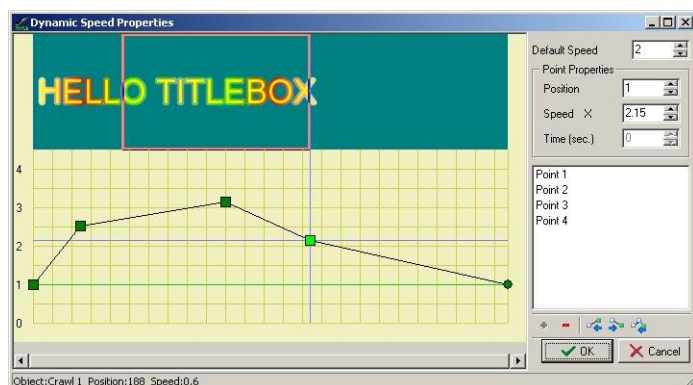
This property dialog box allows you to specify different speeds of the **Roll**'s and **Crawl'**s movement.

The movement is represented graphically and you can define the speed of each point of the graphics. The horizontal axis of the graphics represents the position of the **Crawl/Roll** on the screen. The vertical axis represents the speed multiplier (0; 1; 2; etc.) of the default speed, which is set in the main screen (see the **Toolbar** section above). The zero value means *0x*default speed, i.e., the object does not move; one means *1x*default speed, i.e., the object moves with the default speed; two means *2x*default speed, i.e., the object moves twice as fast as the default speed, etc.

On the top of the graphics, you can see the object's **(Roll/Crawl)** text. By moving the mouse pointer over the grid (the blue lines) or by using the arrow keys, you can select the position in the text, where you would like to change the speed. The text section, which will be displayed at the selected "speed change" point, is enclosed in a frame.

By default, the first point is situated at the beginning of the graphics. A new point is added by pressing the ***Add*** [icon] button or by double-clicking in the yellow-squared area.

When you select a position to change its speed, a green point will appear in the grid and its properties (*Speed* and *Position*) will be displayed in the *Point Properties* area to the right. The position's coordinates are also displayed in the status bar.

If you set a speed of '0' for any position, then you will have to define a delay period. This is the period (in seconds), during which the object will remain stopped. The wait-time appears in a red square under the zero-point.

All points are shown in the *Point list* to the right of the graphics. Their names are [Point #], where the # stands for the sequential number of the point.

The ***Align buttons*** [icons] allow for aligning the selected point toward the previous, the next or the first point in the graphics.

The *Default speed* field shows the default speed, as it is defined in the main ***TitleBox*** window (see the **Toolbar** section above).

In the *Point Properties* area, the following properties of the selected point are shown: *Speed, Position,* and *Time* (for zero-speed points only).

The *Point list* shows the list of all "speed change" points in the object.

The ***Add*** button [icon] adds a new "speed change" point in the object graphics.

The ***Delete*** button [icon] deletes the selected "speed change" point in the object graphics.

Channel in a Box User Manual

⏱ Analog Clock Properties
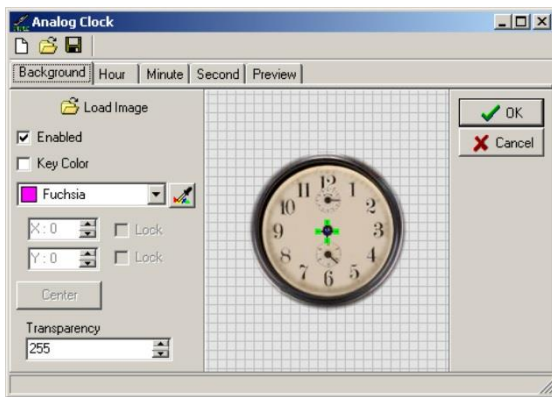
The **Analog Clock** properties dialog box looks like the one above.

The **New** button opens an empty clock property object.

The **Open** button loads a previously created clock object (***.clc**).

The **Save** button saves the current clock image into a file (***.clc**).

There are different pages for each clock layer – background (clock plate), hours, minutes, and second hands, as well as a preview page of the overall clock layout.

All pages have an identical structure: a *settings area* and a *preview area*.

> *Settings:*

Load Image – loads the relevant image (for the background, hour, minute, or second hands).

☑/☐ *Enable* – enables/prohibits displaying the relevant element.

☑/☐ *Key Color*– key color for the image. If the image does not have a mask, you can select the key color.

☑/☐ *Lock position* – locks the [X/Y] position of the image.

Transparency– sets the image transparency

> *Preview area*– It is used for previewing the corresponding clock element.

(!) **TIP:** To achieve a satisfactory result, use a picture-editing application to create four square images with equal dimensions. Save them in separate files – one for each element of the clock (background, hour hand, minute hand, and second hand). Be sure to place the hands' ends on the exact centers of the relevant images. Keep in mind that the clock object will have the same size as the image in the file. It cannot be resized!

Channel in a Box User Manual

Animation Properties

The *Sequence Property* dialog box looks like this:
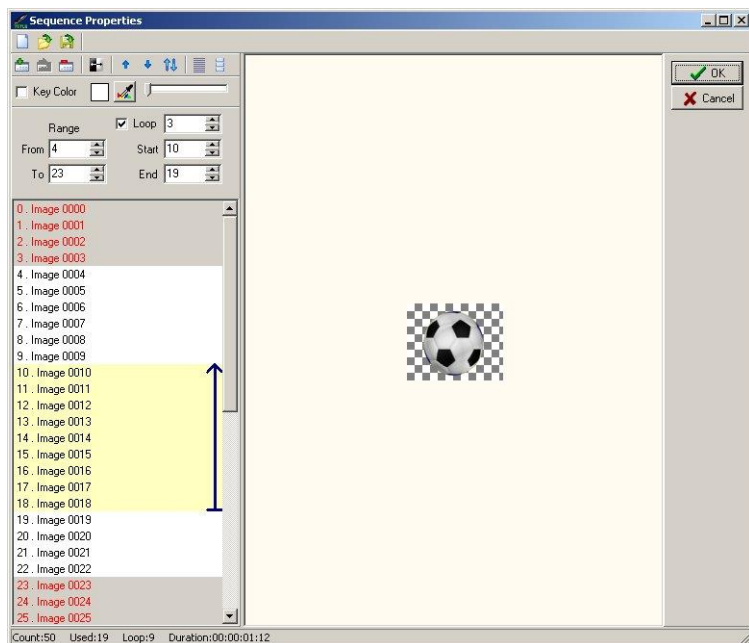
The *New* button allows for creating a new sequence.

The *Open* button loads a file (*.seq*) or a sequence of files (*.tga*) for an animated logo:

Besides, you can load animated *.gif* files. If you use the *Add* button, *TitleBox* will load only the first picture of the file.

The *Open* button, *TitleBox* will load the entire range of *.gif* images.

> **NOTE:** We do NOT recommend using *.gif* sequences for high quality applications, since they have only 256 indexed colors and do NOT have 8-bit transparency (just one color can be either entirely transparent, or entirely solid).

If you want to load only a part of the files, uncheck the ☑*Load All* box and enter either the number of the first or the last file of the sequence, or both, by checking the ☑*Load From* and ☑*Load To* boxes and filling the relevant spin-boxes. Use the blue arrows to the right to enter the number of the currently selected file in the relevant field. If you are loading an interlaced animation, specify the first field – ⊙*A Field First* or ⊙*B Field First*.

Click *Open*. The sequence will be displayed in the animation property window. The currently selected file from the sequence will be shown in the preview area to the right.

The *Export* button saves the current sequence as a file (*.seq*).

The *Add* button adds a new file to the sequence.

Channel in a Box User Manual

The **Delete** button deletes a selected file from the sequence.

The **Insert** button inserts a file into the sequence.

The **Invert alpha** button inverts the alpha channel of the selected file.

The **Move Up/Down** buttons move the selected file up/down the list.

The **Reverse** button reverses the files' order

The **View** buttons are used for changing the sequence files' view to **list** or **thumbnail** mode.

If the animation files do NOT have an alpha channel, you can select a key color by using the **Key color** tool:

The *Range* and the☑ *Loop*-related spin-boxes are situated below the key-color setting kit. Select the *Range* of frames that will be used in the sequence – enter the values you wish in the *From* and *To* fields. The names of the frames that are out of this range will become red, and their background – grey. The background of the working range will remain white. If you want to loop between two frames in this working range, place a check in the ☑*Loop* box, and fill in the *Start* and *End* fields, as well as the number of loops. The frames, included in the loop will be highlighted in pale yellow. A blue arrow to the right of them will mark the final and the initial frame of the loop.
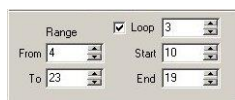
After adding your animation to the preview area, an additional toolbar will become active. In order to see it, click on the animation object:
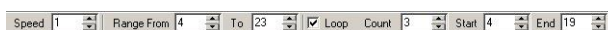
Select the desired *Speed* and enter the number of times you want to ☑ *Loop* the animation.

This taskbar duplicates the range and loop settings in the **Properties** dialog box. You can change those settings here as well. Just choose a *Range* of animation frames to be displayed, or check/uncheck the ☑ *Loop* option (it enables leader-loop-trailer functionality). When you check ☑ it, select the range of frames, within which you want to loop. If this option is selected, the animation will start from the beginning. Run to the *End* frame and loop between *Start* and *End* frames. If you want the animation to run regularly again, simply uncheck ☑ the *Loop* box.

NOTE: The *From* and *Start* fields represent the first frame that will be shown, and the *To* and *End* fields represent the first frame that will NOT be shown. Thus, the difference between the *To* and the *From* (and between the *End* and *Start*) values will be equal to the number of frames that will be shown in your sequence. These numbers will be displayed in the status bar of the sequence's properties dialog box. *Count* stands for the total number of frames in the list, *Used* stands for the number of frames in the working *Range*, and *Loop* stands for the number of frames that will be looped.

IMPORTANT! You must have enough uninterrupted free RAM to load a TGA sequence in **TitleBox**. The minimum free RAM needed for loading a TGA animation is calculated with the following formula: (Animation_Width multiplied by Animation_Height, multiplied by 4) multiplied by the Animation_Frame_Count.

Example 2 – Minimum Free RAM for Animation Sequence

To save or open a project, which contains such an animation, you will have to multiply the needed RAM by 2.

Thus, the minimum free uninterrupted RAM for saving/opening a 100-frame animation with 300x200 frame size will be:

(300 * 200 * 4) * 100 * 2 = 48,000,000 bytes = ~45.77 MB.

Channel in a Box User Manual

> **TIP (!)** In order to reduce these requirements, you could export your longer animations to sequence files by using the **Create Animation File** object (see the section below).

> **NOTE:** Fielded animations are NOT supported! Use frame-based animations with each field in a separate file. Then, use speed 2 for proper speed and field interpretation.
>
> **NOTE**: The Animation/Sequence is limited to 32-bit TGA.

Create Animation File
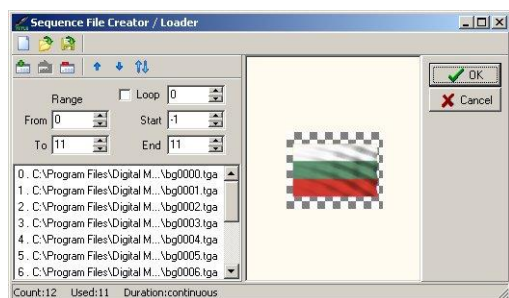
.This object was created to avoid the real-time rendering of sequences, thus, reducing the system resources' load during playout. We recommend using it for large-sized animations with numerous frames. The resulting file format is *.seq2*.

The properties of this object appear in the dialog above. It is similar to the *Animation properties* dialog and most of the options are the same.

To load an already existing sequence file, click on the **Open** sequence button.

To create a new sequence file, click on the **Add Images** button and browse for them. In the **Open sequence** dialog, select the images to load and click **Open**.

You can use the arrows to change the image order.

In the *Range* spin-boxes, specify which images should be included in the sequence. All of them are included by default.

Once you have loaded all the frames you want to include in the sequence, press the **Export Sequence** button, and save the file.

> **NOTE:** Fielded animations are NOT supported! Use frame-based animations with each field in a separate file. Then, use speed of '2' for proper speed and field interpretation.
>
> **NOTE**: The Animation/Sequence is limited to 32-bit TGA.

Direct Show Media Properties

TitleBox allows for inserting all kinds of Direct-Show compatible media.

First, you have to specify the source media:

The ⊙ **File** string contains the file path or URL to play in the object. Press the **Load** button to the right to browse for it.
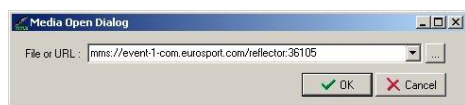
> **NOTE:** In the dialog you should specify the URL of a live network stream, not a link to a file.

The ⊙ **Capture Device** drop-down list contains all direct-show compatible capture devices installed in the PC (like FireWire camera, DeckLink card, etc.). Select one of them.

⊙ **Custom (Support)** – this is an advanced option. Click on it to select the filters to use in this object on your own. You can add filters to the graph by clicking on the *Add* button. A list of all available filters will open for you to select from. To view the properties of a filter in the graph, select its line in the list and press *Properties*.

Example 3 – Direct Show Media Capture Device

If there is a separate DeckLink card installed, *TitleBox* could use it as a capture device. It will be visible as DeckLink Video Capture (2) in the capture list, as in the screenshot bellow:
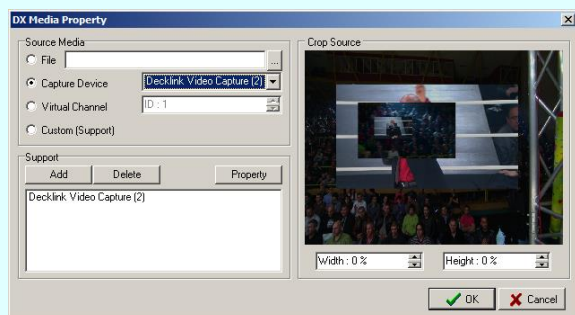
You can crop the image [Width] and [Height] by using the relevant spin-boxes under the preview window.

When a direct show object is selected in the **TitleBox** preview area, an object-specific toolbar appears under the standard ones:

The **Properties** button opens a list of all filters used in the current graph.

Check ☑ *Alpha* to use the video's alpha channel. You can adjust it by using the percentage spin-box to the right.

In this toolbar, there are two drop-down lists: one for selecting the display *Mode*, [Stretch] or [Center], and another one for the scaling quality. If the [Pure (Fast)] quality is selected, the CPU usage is lower.

Enter the *Loop* number to repeat the video as many times, as you want. *Loop* = [0] means that the video will be endlessly repeated; *Loop* = [1] means that the video will be played once; *Loop* = [2] means that the video will be played twice, and so on.

Any loop number different from zero represents how many times the object will be looped.

☑*Keep Aspect* – check this box to preserve the aspect ratio when resizing the object.

The next two spin-boxes control the image's[X] and [Y] offset in relation to the object's center. Use them while in [Center] mode, to move the video vertically or horizontally within the object boundaries.

When in [Center] mode, you can use alpha matte to fill-in the space between the edges of the video and the object's borders. You can select its *Color* from the ▦▦▦▾ palette to the left of the ☑*Use Alpha* check-box.

The two sound controls are situated at the end of the toolbar. The *Audio Delay* spin-box allows for adjusting the A/V sync of the object. To the right of it you can find the *Volume* control.

NOTE: The audio of your DirectShow objects will be output on the Default Audio device, set in Windows **Control Panel➔Sound and Audio Devices⇨Audio**.

WARNING! As the playout of direct show media objects is carried out by third-party filters, we cannot guarantee the A/V sync of these objects.

▦ Banner Properties

Pushing the **_Insert Banner_** ▦ button invokes a dialog box for you to create a list of picture files. They will be displayed as a slideshow in this object:

Use the respective buttons ▦▦▦ | ▦ | ↑ ↓ ↕ to ***add, insert,*** and ***delete*** pictures from the list; ***invert their alpha***, move them ***up*** and ***down***, as well as ***reverse*** their order.

When you select a picture from the gallery to the left, it is displayed in the preview window to the right. The buttons above it are relevant for the currently selected picture. The ▦▦▦▾ button allows you to choose a background color for it. Specify how long it will be displayed (in seconds), the transition pattern and duration, as well as the picture's layout (from the drop-down list in the upper right corner).

If you would like to set the same parameters for all pictures, included in the banner, simply press the ***Set As Default*** ⬇ button.

Thus, the settings you have already made will affect all newly inserted images in this object.
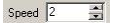
When you check the ☑*Watch Directory* option, a browser dialog will open for you to specify a folder (watch folder) that contains picture files. Opening the folder will activate a kind of slideshow in the banner and the images from the watch folder will be shown in the banner object in a random order.

> **NOTE:** Keep in mind that all files, used in the banner objects are kept into the project. In this way, the size of the project is increased and it is possible to overload it. To avoid such overloading, we recommend using the *Watch Directory* option.

Chat note objects

Another object from the object palette is the **Chat note** object . This kind of object can be used together with some third-party applications, like SMS, chat, etc. It creates a text file. You can insert a file link (**Project ➔ Plugins ➔** File Link) in this text file. The chat note will be updated every time the relevant file is saved. The object's properties are controlled in the same way as the other objects (see the Text object properties section).

A **Chat Note** can be also treated as a **Textobject** but when you enter a text in it, the text is always shown as a new line coming from the bottom, and the old text is rolling to the top. The information entered is not saved into the project.

When you click on this object once, an additional object-specific toolbar will appear below the standard ones. Specify the desired speed for changing the text lines in the chat note in seconds via the *Speed* spin-box: Speed 2

When you double-click on the object, a window will appear. The properties are the same as for the Text object's Text and Background properties.

> **IMPORTANT**: When the object is linked to *.txt* file, then the font formatting is taken from **TitleBox** font properties. When the object is linked to an*.rtf* file on the other hand, then the font formatting is taken from the *.rtf* file itself.

Sound objects

The sound objects are actually links to DirectShow-compatible sound files. After drawing the object rectangle in the work area, a browse dialog opens for you to locate the sound file. When a sound object is selected, the following toolbar appears under the standard toolbars in **TitleBox**:

Loop 0   Filename : 09 - Secret.mp3

Here you can specify a number of loops for this sound and change its appearance in the work area from the color palette.

The sound filename is displayed to the right. If you want to change the sound or the file path, double-click on the object to invoke the browse dialog.

> **(!) TIP:** You could apply transition effects to your sound objects via the *In* and *Out* transition toolbar, situated above the object-specific toolbar. It contains two drop-down lists of transition effects and two spin-boxes for the *In* and *Out* transition duration.

Select [Fade] from the *In* drop-down list and enter some figure in the spin-box next to it. Thus, the sound volume will increase gradually according to the time set – the longer the time, the slower the increase.

Select [Fade] from the *Out* drop-down list and enter a figure in the spin-box to the right of it. Thus, the sound volume will decrease gradually at the end of the sound file.

> **WARNING!** Be careful when moving **TitleBox** projects containing sound files! If the sound file is not available at the new location (its file path is not the same as the one saved in the project), the sound object will not be executed!
>
> **NOTE:** The sound objects are quite heavy. Inserting more than three sound objects per project could slow down the rendering of **TitleBox** objects.
>
> **NOTE:** This object's sound will be output through the Default Audio Device (set in Windows **Control Panel➔Sounds and Audio Devices**⇨*Audio* tab).

🕐 Digital Clock Properties

The default **_Digital Clock_** object looks like the one to the right:

It displays the current system time. You can change it by using the _Time Offset_ spin-box (_see the bottom of this page_).

A text-formatting toolbar appears upon double-clicking on the digital clock object:

Use this toolbar to edit the digital clock's common text attributes (such as font, size, color, blur, shadow, etc.) and background attributes. This toolbar has already been described in the Text template properties section above.

If you click once on your **_Digital Clock_** item, the toolbar below will appear below the standard **_TitleBox_** toolbars
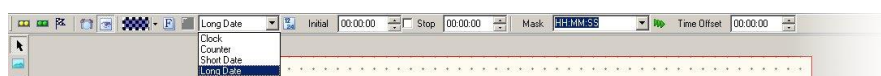
The **_Digital clock_** has four modes of operation. It can run as a [clock] or a [counter], or it can display [Long] or [Short Date].

While in [clock] mode, the object runs as an ordinary digital clock. You can choose its time format through the **_12/24 hours clock_** button. 🔲. The twelve-hour time format is active when the button is pressed, and the twenty-four hours format – when it is not pressed.

The background color is changeable through the **_Palette_** 🔲 button.

Select a mask for the clock in the _Mask_ drop-down menu. Update the mask by using the green arrows button 🔲 next to it. If you have changed the mask, you will have to press this button to show it on the display.

In [counter] mode some additional options are available. You can select the counter to be countdown by pressing the **_Countdown_** 🔲 button. If you push this button, the counter will count from an _Initial_ time to a _Stop_ time or until the ☑_Stop_ box is checked. If not pushed, the counter will count up. After selecting the counter type and entering the _Initial_ and _Stop_ times, as well as a _Mask_, you are ready to run your **_Counter_**. Run your project and show your object by using the well-known **_Play_** 🔲 button. Then, **_Prepare_** 🔲 your _counter_ and **_Start_** 🔲 it.

After pressing the **_Prepare_** button your _counter_ sets to _Initial_ time and waits for a **_Start_** command. **_Stop_** the counter with the 🔲 button. Pushing the **_Intermediate_** 🔲 button will "freeze" the _counter_. During freeze, the _counter_ will be running in the background but you will see a still frame, displaying the time of the counter from the moment, when the **_Intermediate_** button has been pressed. When you push the **_Intermediate_** button once more, the **_counter_** will start showing the current time again.

Two buttons are active for both the [clock] and [counter] modes – **_Properties_** 🔲 and **_Visible_** 🔲. Pressing **_Properties_** opens the font formatting dialog. The **_Visible_** button determines if the _Clock/Counter_ will be visible on the monitor, or not. It does not stop the _clock/counter_ but it just hides/shows it.

Time Offset – This spin-box allows for creating digital clocks for different time zones.

In [Date] mode the object will display the system date instead of the system time. There are two date formats: [Long Date] and [Short Date]. Their appearance depends on the regional settings of your **_TitleBox_** server. You can change them in Windows **Control Panel➔Regional and Language Options➔** _Regional Options_. After pressing the **_Customize_** button, go to the **_Date_** tab. There you can check your current system date settings and change them at your will.

Channel in a Box User Manual
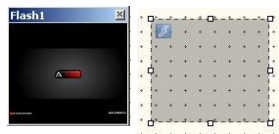
Flash objects

Flash objects, similar to sound objects, are actually links to flash files. To create a flash object, press the flash button in the **Object palette** and draw a rectangle. Two windows will open as soon as you release the mouse button: a standard browse dialog to locate the flash file and an interactive properties window. The name of the Flash object is displayed in the caption of the properties window.

When a flash object is selected in the work area, the following toolbar appears underneath the standard *TitleBox* toolbars:

It provides the following options:

Press the *Pointer* button to show your mouse pointer on the output. Thus, the operator's actions will be shown on the monitor.

Use the *Invert* button to change the color of the pointer that is displayed on the output.

Select the mode, in which your flash should be displayed from the first drop-down list to the left. It contains all standard flash settings. Make your choice in accordance to the system capabilities of your PC.

Use the second drop-down list to set the quality of your flash image. Practically, this is an anti-aliasing setting of the flash object.

You could select a color from the last drop-down list to the right and make it transparent by checking the ☑*Use Key Color* box.

The last check-box to the right is related to aspect-ratio incompatibility. If your flash has a 4:3 aspect ratio and the output (in **Project** menu ➜**Options**⇨*Output*) is set to some other aspect ratio (such as 16:9, for example), checking this box will ensure the correct displaying of your flashes in the working area.

NOTE: You must have a Flash Player installed on your PC in order to play these objects.

WARNING! Do NOT close the preview window of the flash object and do NOT minimize the *TitleBox* window.

Power Point Objects
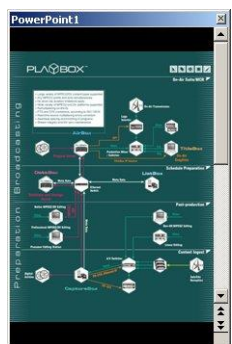
TitleBox allows you to insert Power Point presentations in your projects. Just press the *PowerPoint Presentation* button from the object palette and draw a rectangle in the work area. Then browse for the file location. You can change the file later by double-clicking in the Power Point object or by pushing the *Properties* button in the toolbar.

Channel in a Box User Manual

If you right-click on a PowerPoint object or push the **Properties 2** 12 button, the PowerPoint properties window will open. It is interactive and allows you to control your slide shows.

The name of the currently selected object is written in the window caption.

WARNING! Do NOT minimize the **TitleBox** window while displaying a power point presentation – this will hide it from the output! Do NOT close the preview window of the object.

IMPORTANT: You must have a full PowerPoint version installed on the **TitleBox** PC in order to run these objects. Only PowerPoint 2003 is supported.

■ Primary Shapes
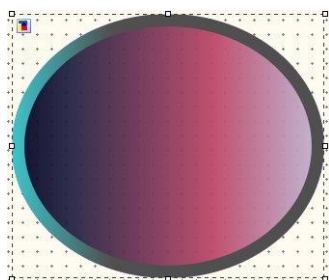
You can create your own mattes for the text objects directly in the **TitleBox** work area. They are true-color with 256-level transparency.

Press the **Create Shape** button in the object palette and draw a rectangle in the work area. By default, an oval shape will appear in it, and the object-specific toolbar will appear under the standard ones:
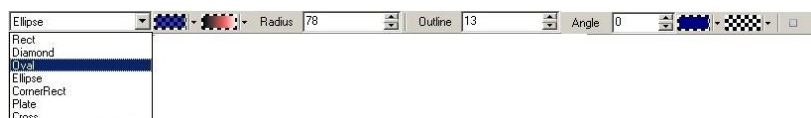
The toolbar provides a wide variety of editing options. The drop-down list of available shapes is situated to the far left. The color palette next to it is designated for the background color. You can select it from the available ones, or create it on your own.

Further to the right there is another palette for the fill effects. You can choose between **None, Gradient color,** or **Texture Fill**.

The fill and outline effects are the same as in text objects. Check the Roll/Crawl [properties section](#) above.

NOTE: The background color serves as an alpha channel to the fill-effect. The background color (and transparency) will be mixed with the fill-color and its transparency.

Use the **Radius** spin-box to enlarge your objects or to make them smaller.

The next spin-box determines the **Outline** width. The color palettes next to it are for the background color of the outline and its fill effects respectively. Similar to the shape background and fill, these will be mixed as well.

The **Angle value** is in degrees and it is valid for **polygon** and **star** shapes only.

The **Square** button □ in the far right of this toolbar fixes the aspect ratio of the shape object. It will change your object to fit a square and will keep this shape when you resize it.

WARNING! You will NOT be able to revert to your previous shape after checking the *Square* check box.

Channel in a Box User Manual

Browser Properties

The browser object uses Internet Explorer to display web pages as graphics on the screen.

As soon as you release the mouse key after drawing the object's rectangle, the Open page dialog will appear in the work area. Type in the *URL* you would like to browse, or paste it from the clipboard.

Once you click **OK**, the object-specific toolbar will open under the standard toolbars:

In it, you can type another *URL* to be loaded in the browser object:
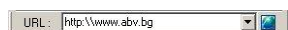
The **Properties 2** button will invoke the Browser Open Dialog. In it, you can see the currently loaded web page. If the object's dimensions are smaller than the page itself, you can use the scrolls to select which part of the page to be shown on screen.
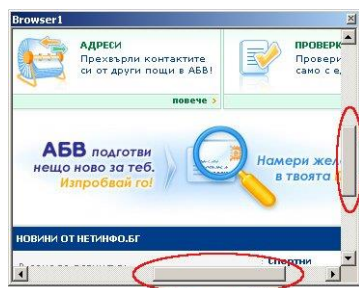
IMPORTANT: If a link in the webpage is set to open in a new window, you will not be able to open it in the same browser object. There is no practical way to grasp such a window into the same object.

WARNING! Do NOT minimize *TitleBox* while using browser objects! This will make the objects disappear off the screen!

WARNING! Do NOT close the preview window, as it might freeze the source web site.

Screen Capture object

This object will allow you show some parts of your desktop on the screen.

The window above will appear as soon as you create the screen capture object. Its size is the same as of the rectangle you drew in the work area. Drag it to the area you would like to show on the screen.

In the upper right corner, there is a two-sided arrow button. Use it to minimize/maximize the main *TitleBox* window but leave the Screen Capture window ON. Please, note that if you use the minimize/maximize buttons in the main *TitleBox* window; the screen capture dialog will be affected as well.

In the lower left corner, you can find two other buttons that control the mouse pointer. The first one is for showing/hiding it from the screen. The second button will invert the cursor color. Thus, if the background of your desktop is light, you can turn the cursor black, so it will be easier seen on the screen.

When a screen capture object is selected in the work area, the object-specific tool bar will appear under the regular toolbars:

The first two buttons in it duplicate the ones for cursor control in the lower left corner of the Screen Capture window. To the right is the ☑*Antialiasing* check box. Use it to make the sharp edges smoother, thus avoiding possible aliasing effects on the output.

> **NOTE:** It is NOT possible to capture overlay video in this object.

Chat Line

Chat line objects appear in the work area as the one below.

Chat lines, like Chat notes, are developed to display text messages, coming from SMS service applications. Chat line objects should be linked to text files, where third-party applications save the newly-received messages. Unlike chat notes, the messages in chat lines are moving constantly, even if there are no new messages arriving in the linked text file. When a new message appears there, it will be displayed during the next queue rotation.
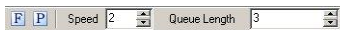
The Chat line toolbar contains the following settings:

F - use this button to format the chat line font.

P - paragraph formatting allows you to change the alignment, spacing, word wrap, and other properties of the currently selected paragraph.

You can change the speed of the text through the *Speed* spin-box: the higher value, the faster will move the text. If you want to change the direction, type a minus [-] in front the speed value.

The *Queue Length* spin-box is used to define the number of messages to be displayed in the chat line. By default, it is three '3', which means that in the chat line you can see only three messages at a time. When a new message appears in the linked file, it will be displayed at first place in the chat-line object. The oldest message will disappear, so the number of the messages in the chat-line will remain three (3).

Chat Roll

The function of this object is the same, as it is with the Chat Line and Chat Notes (displaying messages coming from third-party applications to a linked text file). The only difference is that it displays the saved text as Roll ticker.

To create a Chat roll object, Press the button in the object palette. Then, right-click on the object to set the File link.

The object-specific toolbar is identical to the Chat lines. Please, check the Chat line section above for details.

Channel in a Box User Manual

SCHEDULER

Schedule Commands

Object Settings

Object List
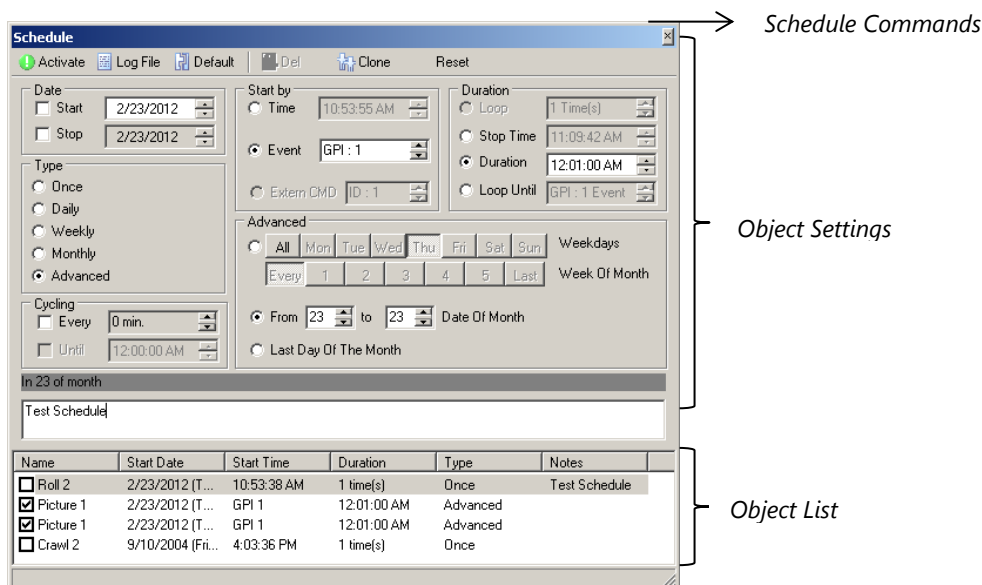
The **Schedule mode** allows you to create schedules for playing the objects in your project.

To add an object to the **Schedule**, press the **Add to Schedule** button from the **TitleBox** Toolbar. Note that this button appears on the toolbar after you have selected an object from the *Work Area*. The **Schedule** window will open. It consists of *Schedule commands*, *Object settings*, and an *Object list*.

Scheduler Commands

Activate– press it to activate the schedule for the current project.

Log File – press it, if you want to create a log file. A browse window will open for defining a log file.

Default– sets the default settings. You can create your own default settings as well.

Del– deletes a selected object from the *Object list*.

Clone– "clones" the settings of the selected object. A new line for the same object will appear in the *Object list*. Then you can modify its settings. This feature is useful, when you want to define a different behavior of the object in the different days or hours, for example.

Reset– resets the Schedule.

Object Settings

The available settings for scheduling objects are situated here, in the main part of the **Schedule** dialog.

You should select an object from the list below to make the settings for it.

Date – defines the **Start** and/or **Stop** date. By default, the **Start** day is today. The **Stop** date is not mandatory.

Start by – defines the starting trigger. It could be a specific time, an event (for example *GPI*) or CMD.

Duration – defines the duration of the object appearance through

Loop – the number of loops. This field is active only for dynamic objects.

Stop Time – the stop time within the relevant day.

Duration – duration within the relevant day.

> **NOTE:** This duration is different than the object's duration, set in the project. You have to be very careful when you define different scheduling and object duration.

If the object duration is shorter, the object will disappear after its end, even if its schedule duration is not finished.

If the scheduling duration is shorter, the object will disappear after its end, even its own duration is longer.

Loop Until – defines a stop event for the object's occurrence in the output (e.g., GPI trigger).

Type– defines the frequency of the object occurrence in the output – ⊙*Once*, ⊙*Every day*, ⊙*Every week*, ⊙*Every month*, or in a more sophisticated pattern, ⊙*Advanced*.

Cycling – defines the period for repetition of the object occurrence in the output in minutes. It is possible to define the termination of the object's occurrence within the day –☑*Until* spin-box. Note that the latter spin-box is only active if you have defined a cycling period in the ☑ *Every* spin-box.

Advanced – to get access to these settings, you have to click on the ⊙ ***Advanced*** button in the *Type* field. Here you can define a specific day of the week or a date for showing the object.

Objects List

The *Objects list* occupies the lower part of the ***Scheduler*** window. All the scheduled objects and their settings are displayed in it.

In front of each object in the list, there is a check-box. It is used for ☑activating/☐ deactivating the scheduler settings of the particular object.

Right-clicking on an item from the *Objects list* opens a context menu for copying and pasting the object settings. You can also use the following key combinations:

<Ctrl + Alt + C> - Copy

<Ctrl + Alt + V> - Paste

Please, refer to the screenshot below:
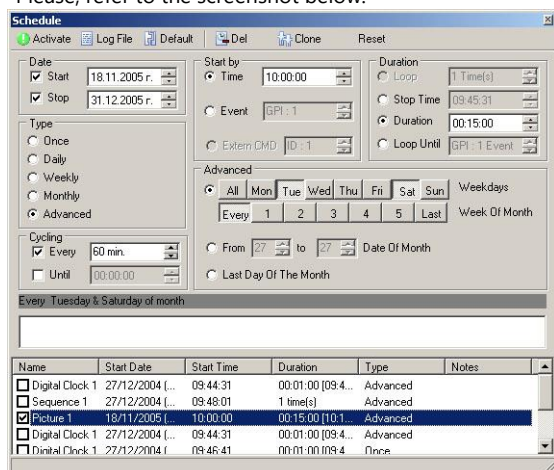
As it is visible in the *Date* field, [Picture1] starts at 10 o'clock on November 18th, and it is put on schedule until December, 31st. Furthermore, in the *Duration* field it is set to be displayed for 15 minutes every hour. Finally, [Picture1] is also being displayed on Tuesdays and Saturdays, as set in the *Advanced* area.

SLIDE MANAGER

In the past, you could create your ***TitleBox*** projects in a single layout only. If you had to open a new layout, you had to load a new project. It was like having a single sheet of paper per project. With the ***Slide Manager*** your project turns into a sketchpad, allowing you to organize your projects in a multi-

Channel in a Box User Manual

slide layout. Each sheet of this pad is called a **Slide**. In each **Slide** you can have numerous groups of objects, called **Layers**. Finally, in each **Layer** you can have as many objects as your system can handle.

You can now control your slides/layers/objects manually, or play the slides consecutively (or simultaneously) by using the Slide Controller.

The **Slide Manager** is locked to the right of the work area. You can open and close it by using the **Slide Manager** button, situated in the lower right corner of the **TitleBox** interface. You can also drag the **Slide Manager** out of the main **TitleBox** window.

Slide control buttons

These buttons are situated in the lower right corner of the **TitleBox** interface. They provide simple **slide control** options. The numbers in the middle represent information about the currently selected slide (i.e., slide 2 of 3 in the screenshot above).

The main **Slide Manager** window consists of three logical parts – control buttons at the top, slides' thumbnails and a project structure, situated in the predominant part in the middle (further divided into three tabs), and a properties area at the bottom, where you can find a detailed description of the currently selected element in the list above.

Project control buttons

The first three buttons at the top of the Slide Manager window are used for controlling Slides. You can **Add**, **Insert**, and **Delete** slides by using them.

> **NOTE:** The difference between **Adding** slides and **Inserting** slides is that when you add a slide, it appears at the bottom of the slides list, while when you insert a slide, it appears above the currently selected slide.

The following two buttons are used to control Layers. You can **Add** or **Delete** layers.

While in the **Details** tab, you can show/hide slides' thumbnails by pushing the **Thumbnails** button.

In case you need to move a slide or an object, just drag-n-drop it to the new place where you would like it to be.

If you want to show/hide/pause a slide, a layer, or an object manually, slide the mouse pointer to the end of its row. The **Play/Stop/Pause** buttons will appear. They are related only to the currently selected line. The elements' hierarchy will be observed, i.e., when you push **Play** for a Slide, all Layers and objects that belong to it will be played (if not disabled). If you push **Play** for a Layer, only the objects in this layer will be played. Respectively, if you push **Play** for an object, **TitleBox** will show only this object.

If a Slide is in **Stop** mode, it will not prevent you from playing its Layers and objects.

Pressing the **Shortcut mode** button will enable the shortcuts functionality. Thus, if you have assigned some shortcuts to slides in your project, you will be able to control these slides using the relevant shortcuts. Please, check the Preview section below for details.

Project Preview area

The project preview area consists of three tabs as described below.

Details

In this tab you can view the hierarchical structure of your project. It is divided in two parts – project tree and properties.

Project tree
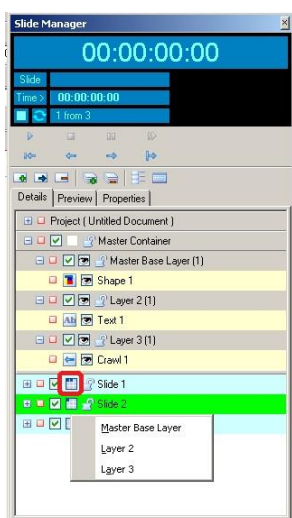
The project tree part is further divided in two levels – Level 1 that contains the Project line and the Master container and Level 2 that contains all "regular" slides. Right-clicking anywhere in the project tree invokes a context menu, which is described in the dedicated section.

Level 1

The **Project line** is situated on top. It displays the filename of the current project. The **_line control buttons_** here have a slightly different functionality as compared to the other lines below. These buttons control the Slide Controller that will play all slides in the project one after the other (see the Slide control section for details). Under the project line, you will find a list of all unassigned objects (if any). These objects are present in the object but do not belong to any of its slides. Click on the plus sign in the beginning of the project line to view them. You might need to show/hide such objects manually, without affecting the rest of the project.

The **Master Container** line is situated right under the project line. In it, you can create layers to be assigned later to the "regular" slides below. By using these master layers, you will avoid inserting the same objects in numerous slides.

For example, if you need a logo shown in Slides 1, 3, 7 below, just create a Layer in the Master container. Then, go to Slide1, click (left mouse button click) the _**Assign Layer**_ button and select the logo-containing layer from the drop-down list. Repeat the procedure for Slides 3 and 7. Thus, each time _**TitleBox**_ displays Slide 1, 3 or 7, it will also display the logo-containing layer from the Master Container.
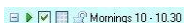
Another way to do this is just drag-n-drop the Layer from the Mater Container to the relevant slide.

Level 2

All slides are listed in this part of the _**Slide Manager**_ window. You can create as many Slides and Layers as your computer can handle. You MUST have at least one Layer per slide.

Each element is displayed in a separate line. Each line starts with a plus sign. You can click in it to expand the relevant level.

Next to the plus sign, you can see a play status notification. Further on there are several controls that vary within the different levels. You can find a detailed description of each level below.

In a **Slide line**, you will see Enable/Disable check box, _**Assign Layer**_ button, _**Lock/Unlock**_ button _(it is under development now)_, and the Slide name (check the Slide Line Properties section below on how to change Slide names).

If you have assigned some layer(s) from the Master Container to a slide, their number will be reflected in the Assign Layer button. Thus, if you have assigned one layer, a small dark-blue square will appear in the button. If you assign two layers to the slide, there will be two dark-blue squares in the Assign Layer button, and so on.

In a **Layer line** there is no ***Assign Layer*** button as this contradicts the hierarchy. However, there is a ***Visible/Invisible*** button. This button will hide the currently selected layer from the work area, so its objects will not impede you while creating another layer in the same slide. After the Layer name, there is always a figure in brackets. It represents the number of objects contained in this layer.

In an **Object line**, you can see the status of the object (play / pause / stop), its visibility status, an icon, distinguishing its type, and the name of the object.

> **NOTE:** The object visibility concerns only the status of the object within the project and it does NOT affect the output.

In case you need to move a slide or an object, just drag-n-drop it to the new place, where you would like it to be.

Right-Click
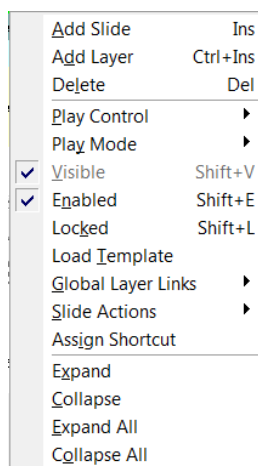
The first three commands are the same as the project control buttons:

Add Slide – use this command to add a slide at the bottom of the project tree

Add Layer – use this command to add a new layer at the bottom of the currently selected slide

Delete – use this command to remove the currently selected item from the project tree

The **Play Control** menu allows you to **Play**, **Stop** or **Pause** the currently selected item.

Use the **Play Mode** menu to add a play mode to the currently selected slide. Play modes are described in the Properties section.

You can also enable/disable the following properties:

Visible – this option is only applicable for layers, it shows/hides the currently selected layer.

Enabled–use it to enable/disable the slide/layer.

Locked – use it to lock/unlock the currently selected slide/layer for editing.

Load Template – use this command to load an already created ***TitleBox*** template in the currently selected slide. Select your template from the browser that appears.

Use the **Master Base Layer** command from the **Global Layer Links** menu to link the currently selected slide to the master layer. In this way the slide will use the design of the master layer as a base.

Use the **Slide Actions** menu to assign an action to the currently selected slide:

None – no action will be assigned

Stop Playing – the slideshow will stop playing when it reaches this slide

Jump To... – select an existing slide. Whenever the slideshow reaches this slide, it will jump to the slide you select.

Wait Key Press – the slide will not be executed until you press a key.

If you select the **Assign Shortcut** command the following dialog will be invoked:

In the *Shortcut* field enter a key combination by pressing the desired buttons on the keyboard.

In the *Command* field select the desired command to be executed by the shortcut – ***Toggle Play/Stop*** or ***Play From Slide***.

The last four commands relate to the appearance of the project tree:

Expand – expands the currently selected slide/layer

Collapse – collapses the currently selected slide/layer

Expand All – expands the whole project tree

Collapse All – collapses the whole project tree to level one

Properties

In the area under the project tree, you can see the property's window for the currently selected line (project, slide, layer, or object).

- The properties of the **Project Line** contain some default settings that will affect all newly created slides:

Default color– this color will be applied in the preview thumbnails of your slides. You might need it in case there are some white characters in a text object that would be impossible to see on a white background.

Default slide duration– the time for showing the slides when playing them with the Slide Controller;

Default Play mode– the way of showing the slide on the screen:
Cross Play– if the previous slide has an out transition, the following slide will appear as soon as the out transition starts.
Stop Previous– if the previous slide has an out transition, the following slide will "wait" for this transition to end and only then will appear on the screen.

Add – the previous Slide will remain on the screen and the following will be overlaid on top of it. Currently, there is a known limitation, related to this type of play mode – once you pile up a slide on top of another one, the first slide will remain on the screen, even if the second one is stopped. Thus, if you want to stop the first slide, you will have to do this manually.

Clear Previous – the previous slide will be removed immediately off the screen, its out transitions (if any) will not be executed.

Loop – specifies how many times the Slide Controller should play all slides in the project. The default value [zero] means endless loop – the Slide Controller will play all slides one by one repeatedly.

- If a **Slide Line** is selected, you will see the following properties:

Slide 2	
Property	**Value**
BackColor	
Duration	10
Enabled	True
Lock	False
PlayMode	pmCrossPlay
ViewName	Slide 2

Duration – this is the time (in seconds) for showing the selected slide with the Slide Controller. If you decrease this value to zero, the duration will be set to [Auto]. Auto duration is applicable when there are objects with different duration in the slide. In such a case, the Slide Controller will show the slide until the longest duration is through, then will go to the next slide in the list.

TIP (!) Use it for objects like crawl/roll and animation when they are looped.

Enabled – it is equal to the ☑*Enable* check box in the slide's row.

Lock – reflects the padlock status in the slide's row.

Play mode– how *TitleBox* should treat the previous slide on starting the currently selected one. Please, check the project properties above for details.

View Name– click in this line to change the slide name at your will.

- If a **Layer Line** is selected, you will see:

Enabled – shows the status of the ☑*Enable* check box of the currently selected layer.

Lock – reflects the padlock status in the layer's row

Visible – shows the status of the Visible/Invisible button.

- When an **object** is selected, you will see below all its properties that can be adjusted in the main *TitleBox* interface. These properties are different for the different objects and have already been described in the Object Properties section.

Preview
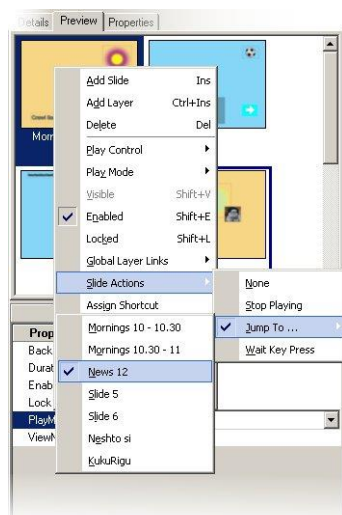
In this tab you can see thumbnails of all Slides in the current project.

Double-clicking in a slide's thumbnail will load it in the work area for editing. If you select an object in the work area, its properties will be displayed in the Details area below.

Right-clicking in a slide will open a context menu. It contains basic slide controls.

The **Play Mode** represents the way of showing the slide on the screen in relation to the previously played slide. Please, check the project properties section above for further details.

Sliding the mouse over the **Global Layer Links** line will open a drop-down list of all Layers available in the Master container.

The **Slide Actions** list contains four commands related to the Slide Controller. They will be executed when the duration of the current slide is over.

None is the default slide action, no command will be sent to the Slide Controller when the slide duration is over.

Stop Playing – when the duration of the currently selected slide is over, it will send a stop command to the Slide Controller. A small sign will appear in the lower right corner of the slide thumbnail to notify the operator about the assigned action.

A Jump To... action will make the Slide Controller loop between the current slide and the specified Jump To... slide.

Wait Key Press is a pause command sent to the Slide Controller. Playback of slides will resume at any key stroke. The following sign will appear in the thumbnail to notify the operator.

Selecting **Assign Shortcut** will open a shortcut-defining dialog. To define a shortcut, click in the *Shortcut* string and press the keys you would like to assign.

Below, select the command to be executed at pressing these keys:

⊙**Toggle Play/Stop** – to show/hide the slide off the screen;

⊙**Play From Slide** – to start displaying all following slides in a row (via the Slide Controller).

To clear the shortcut, press the **X** button to the right of it.

All assigned shortcuts will appear in the slide's thumbnail:

 The blue background means the assigned command is Play From.

 The green background means the assigned command is Toggle Play/Stop.

 The red background means that there is an assigned shortcut to this slide but the Shortcut mode button is not pressed.

The assigned shortcuts will be active only when the **Shortcut mode** button is pushed.

NOTE: In shortcut mode, the newly assigned shortcuts have higher priority than the default ones. Thus, while in shortcut mode, the default shortcuts in *TitleBox* will operate normally unless duplicated with newly assigned shortcuts. For example, the default command for **<Ctrl + H>** is Add to schedule, but if you assign **<Ctrl> + <H>**to be Toggle Play/Stop of a slide, the shortcut will act as Toggle Play/Stop.

Channel in a Box User Manual

Properties

In this tab, you can see the detailed properties of the line that is currently selected in the Details tab.

Slide Controller

The slide controller will play your slides consecutively, with duration at your will. If you want to rearrange the order of showing the slides, you will have to drag-n-drop them in the **Details** tab. If you want to skip a slide, just disable it by un-checking the ☑*Enable* checkbox in its row. .

The Slide Controller window contains four rows and eight buttons. The Controller counter is situated in the top row. It counts the time since the slide controller was started.

Below, you can see the name of the currently playing slide.

Further down is the Slide counter. By default, it shows the elapsed time since the start of the current slide. If you want to see how much time remains to the end of the slide duration, click in the Time> cell. It will turn into <Time and will start counting down.

The last row has three parts – play status, loop value and the number of the currently playing slide.

At the bottom of the Slide Controller are situated the playout control buttons. They provide hints and hotkeys:

Button	Command	Shortcut
	Play	**F5**
	Stop	**F6**
	Pause	**F7**
	Resume	**F7**
	Go to First Slide	**Ctrl + Home**
	Go to Previous	**Ctrl + PageUp**
	Go to Next	**Ctrl + PageDown**
	Skip Next Slide	**Ctrl + Left arrow**

By default, the Slide Controller is a separate window that appears outside the Slide Manager. If you want to lock it to the Slide Manager window, right-click in the counter and select Dock in Manager.

Channel in a Box User Manual

DATA SOURCE MANAGER

The Data Source Manager allows users connect to different data providers through the selected plug-ins.

This engine allows connecting different properties of a **TitleBox** object to different data sources or providers, like RSS channels, *.html files, *.xml files or third party data bases. Besides, the same data source(s) can be related to different objects in **TitleBox**.

The dialog below opens after pressing the Data Source Manager Button in the main **TitleBox** interface.
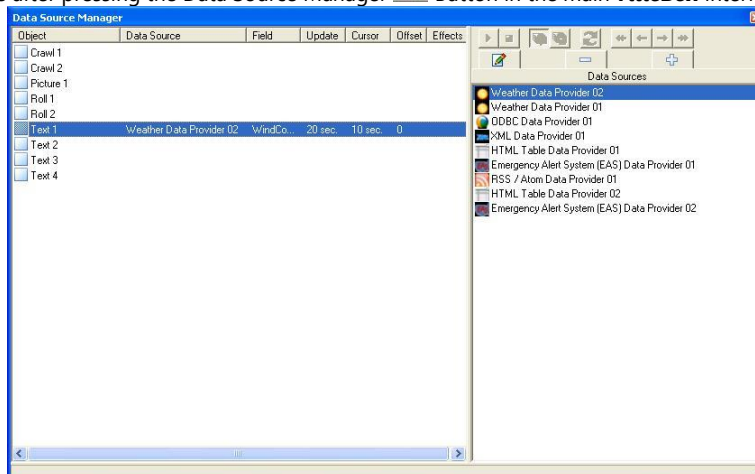
> The left part of this dialog contains a list of all objects in the current project and already assigned to them data source instances.

To add a new data source instance, press the **Add** button.

The list of currently available data provider plug-ins will open. In it, you can choose the data source to be used as a new instance. Select its line and press **Config**. The configuration dialogs vary according to the selected plug-in.

You can read more detailed description of each plug-in definition further in that manual.

When you configure the plug-in, and press **OK** button, into the main screen a new data provider appears in the list.

Select the needed data provider and drag and drop it over the **TitleBox** object in the left.

Then a **DataDistributor Properties** window will open. Here you have to define the way of data source appearing.

Into the header line of the Property window is visible the name of the selected data provider.

Into **Update** area you have to specify how often to update the source information. Select **Automatically**, to update the data automatically on source change. Automatically option is available only for EAS data provider and File link data provider.

Select **Every xx sec**, to update the data periodically at every xx sec. By default, the update is at every 60 seconds.

The term "**cursor**" further in that section, is used to define the cursor position into the data source.

Into **Cursor scroll** area, you can specify how to move the cursor to the next row, if the data source has many rows: Manually, or at Every xx seconds. If you select Manually, you can start the data provider manually from main data provider menu

TIP (!) If you want at every object start to show the first line from the data source, leave **Cursor scroll** to be Manual and check **Start cursor scroll on any Object Play**.

In **Cursor control** area, there are the following settings:
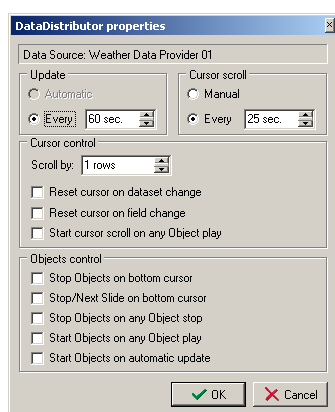
Scroll by X rows, where **X** is a number of rows to scroll the cursor. If **X** =1, the cursor will scroll each row. If **X** =2, the cursor will scroll each second row (i.e. the cursor will skip one row), etc.

Reset cursor on dataset change – check this, if you want to move the cursor to the initial position, when the data in Master column of data provider is changed.

Reset cursor on field change- check this, if you want to move the cursor to the initial position, if any field into data provider changes.

Start cursor scroll on any Object Play – check this, to start the cursor scrolling from the first position in data source, any time, when the object is started.

In **Objects control** area there are the next settings:

Stop Objects on bottom cursor – check it, if you want to stop the object(s) connected to the data provider, when the cursor is at the end of the data in data source.

Stop/Next Slide on bottom cursor - check it, if you want to stop the object and go to the next Slide, when the cursor is at the end of the data in data source.

Stop Objects on any Object stop – check this, to stop all objects connected to the same data provider, if any of these objects is stopped. It is useful, when the project is control remotely.

Start Objects on any Object play check this, to start all objects connected to the same data provider, if any of them is started.

Start Objects on automatic update - check this, to start all objects connected to the same data provider, on automatic update of the data source. It is relevant, if Automatic update is checked. Automatically is available only for EAS data provider and File link data provider

Press **OK** button to save **Source Link** settings.

After that, an **Object Link Properties** window will appear. Here you have to define how to link the data from the data source to the **TitleBox** object.

In the header of the window, you can see the name of the **TitleBox** object to which data provider is assigned and the name of the data provider.
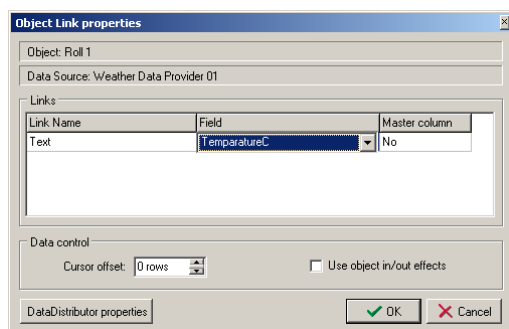
Next is a **Links** area, where are three main columns:

Link Name – this is the type of the selected object (*Text, Sound*, etc.).

Field – here you can see the list of all available fields from the data-source. Select from drop-down list one of them, which you want to show into the **TitleBox** object.

> **NOTE:** You can assign a data-source field to a **TitleBox** object, only if they are from the same type – text, sound, etc.

Master column – you can select one of the data provider's columns to be a **Master** column. Enter YES in Master column field, if the selected field is from this master column. Set a column to be a **Maste**r, if you want to reset the cursor for all columns, on data change in this particular column,

In **Data control** area you can set a **Cursor offset** – this is the offset upon the cursor for displaying the data into this particular **TitleBox** object.

> Example:
> Let us show the data from data source in a table 3x3 (3 rows and 3 column) in *TitleBox*. You have to draw 9 objects in *TitleBox* project, each connected to a separate field from data provider. Into Data Source Link Properties->Cursor control->Scroll by, enter 3(scroll should be equal to the numbers of table rows). Into Object Link Properties ->Cursor offset, cursor offset should be 0 for the first row; 1 for the second row and 2 for the third row.
> If you want to use in/out effects, you have to check the relative check-box.
> Press the OK button to save the settings.

Weather Data Provider

.This data provider displays information coming from weather forecast sources. For now, the only available provider is YAHOO! weather. In this way online weather information will be shown in the objects.

To adjust the *Weather Data Provider*, click the button in the main **TitleBox** interface. The *Data Source Manager* dialog will open. The left part of it contains a list of all objects in the current project. To add a new *Weather Data Provider* instance, press the **ADD** button. A list of all currently available data source plug-ins will open. Select the *Weather Data Provider* and after that press the **ADD** button. A dialog will appear for specifying a station:
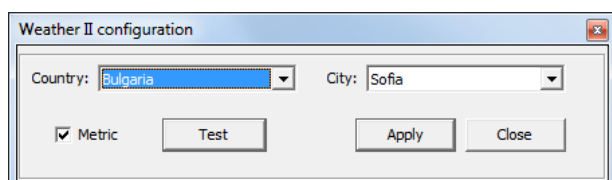
Use the *Country* and *City* drop-down lists to pick your desired location. Keep the ☑*Metric* box checked if you want the temperature to be displayed in Celsius. Uncheck it for Fahrenheit. Press the **Test** button to see if the connection to the station is successful. Press **Apply** to save your configurations and then press **Close**.

After pressing **OK**, a *Weather* [image: Weather Data Provider icon] appear in the list to the right part of the *Data Source Manager* dialog. For displaying information, using this instance, select the [blank] line and drag it to the object desired.

When you release the mouse button, a fine-tune dialog will open so you can select what kind of information will be shown in the object (as Temperature, Postal Code, Current city humidity, etc.).

This plug-in will display the weather information according to the settings you have made. *For example*, if you intend to display information about *Sofia* town and you have chosen *Temperature C* from the *Field* drop-down list, the Weather Data Provider will show the current temperature within the object (have a look at the example to the right).
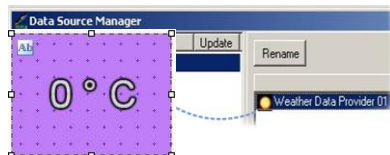
FileLink Data Provider

Select this plug-in if you intend to link certain objects in **TitleBox** to text documents.
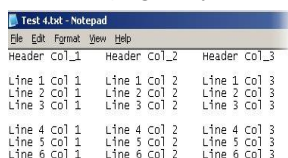
Earlier in this Manual it was described a simple file link. It is assigned to a text object and shows the whole text document, as it is.

Unlike *simple File Link*, the *FileLink Data Provider* allows the user to modify how the text from the text document will be displayed on screen.

Example:
By using *FileLink Data Provider*, one text document could be interpreted as a table. It is applicable for texts, where the data is organized in columns and rows (like in the example Test4.txt file in the screenshot below). The user can define in what kind of order to display the rows and columns.
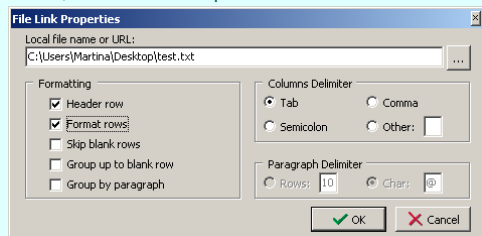
Select the *File Link Data Provider* and press *Config*. Browse for the file you need, select it and press *Open*.

The dialog above will appear, so you could define Formatting of the linked file.

Check the ☑ *Header row* check box, if the first row of the linked document is a column header and should not be visible on the screen.
Check the ☑ *Format rows* check box if the file consists column delimiters (like coma, semicolon, etc.). Checking this check – box will activate Columns Delimiter radio-buttons.
Check the ☑ *Skip Blank rows* check box, when there are empty rows in the linked file; they will be skipped on data scrolling in the text object.
Check the ☑ *Group up to blank row* check box, if you want to show into the text object, a group of data separated by empty row.
Check the ☑ *Group by paragraph* check box, if you want to show into the text object, a group of data separated by some Paragraph Delimiter. Checking this check – box will activate Paragraph Delimiter radio-buttons.
Columns Delimiter is a symbol, which is used for separating the data by columns in the text file. You can choose between Tabulation (Tab), Comma (,) and Semicolon (;) as delimiter, or you can define your own delimiting symbol in field *Other*. In our example, the column separator is Tab.
Paragraph Delimiter is a symbol, which is used for separating the data by paragraphs in the text file. You can select a certain number of rows as paragraph separation or you can define a symbol, which to be used as paragraph separator. Enter the symbol in Char field.
If you want to separate the source data by paragraphs, but there are not delimiter symbols inside your file or the numbers of rows per each paragraph is different, then you can use the PlayBox Delimited Inserter.

PlayBox Delimited Inserter (TextConvert.exe) is an additional application, which helps the users to separate the source data file at different paragraphs. It is installed through the main **TitleBox** installation. You can start the application from **TitleBox** main folder (by default: C:\Program Files\PlayBox Technology Ltd\TitleBox).
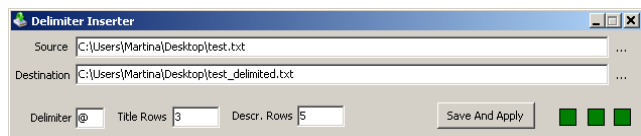
Into the **Source** field, browse the location and name of the source file.

Into *Destination* field browse the location and name of the resulting file.

Select the type of the delimiter symbol in *Delimiter* field.

It is possible to have alternating paragraphs with different length (number of rows). Then you can select the number of rows for the first paragraph in TitleRows field and the number of fields for the second paragraph in Descr. Rows.

Example:
Let the source file contains news data, organized into two paragraphs: in the first one there are the titles of the news and in the second one, there are the full texts of the news. Then the "Title Rows" number is different than the "Description"rows number.
When you are ready, press the Save And Apply button. The new file will be created in destination folder. Depending on status of this proccess, the next three squares will flash:
First square is Running status : Red=Stop; Green=Running;
Second square is Error status: Red=Error; Green=OK;
Third square is Converting status: Yellow=Converting; Green=Waiting for file change.
If the source file is updated periodically, keep the Delimiter inserter opened. It will create a new file at each saving of the source file. Converting square will show the status.
Back into the Data Source Manager window; drag the FileLink data source from the right to a text object in the left.
When you release the mouse button, a link fine-tune dialog will open so you could select which column from the file should be linked to that text object.

NOTE: Currently with File Link Data Provider, you can link your text object only to *txt files!

ODBC Data Provider

This plug-in allows connecting to ODBC-compatible database formats and displaying the information they contain on the screen.

Choose the ODBC Data Provider in the list of available plug-ins and press **Config**.

The ODBC Connection Manager will open.

Press to create a new connection. The following dialog will open for you to specify the Connection Alias and view the Connection String.

Press **Build** to configure the connection. The window below will open:

The ODBC configuration dialog contains four tabs:

In the **Provider** tab, select the data base type you want to connect to.

When you click **Next**, the **Connection** tab opens, so you could configure the connection to the selected database.

The **Connection** tab looks differently depending on the Provider you have chosen. When you are done entering the connection details, presses **Test Connection** to check if it works correctly.
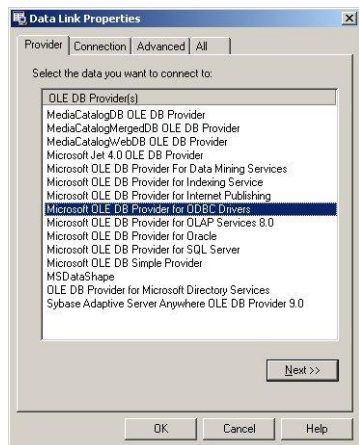
After configuring the connection, you can open the **Advanced** tab for more configuration options, which differ depending on the selected Provider.

Finally, you could view all settings for the current connection in the **All** tab. Click **OK** and **Close** in all open windows until you return to the Data Source Manager. In it, double-click on the newly-created ODBC Data Provider instance. A fine-tune dialog will open for you to configure the data Query. Press **Execute Command** to check if the connection is working correctly.

Press **Apply** and **Close**.

To assign the ODBC Data Provider instance to an object, drag it from the list in the right to the relevant object in the left. Close the Data Source Manager window.

XML Data Provider

Choosing this plug-in allows you to insert data into **TitleBox** object from an *.xml file.

Choose it from the list of available plug-ins and press **Add** button. The following window will open:

Into **XML** field, enter the location of the *.xml file.

Into **XSL** field enter the location of the related *.xsl (*.xslt) file.

An XSL (eXtensible Stylesheet Language) file is needed for defining XML document transformation and presentation. Since the XML language does not use predefined tags, it is necessary to provide the application with information on how to interpret the XML document.

In **TitleBox**, the XSL file is needed to transform the XML file to coma separated text table.

> **NOTE:** Currently, under Microsoft Windows 7 you can use only Msxml4.0 (with update msxml4-KB973685). There is no update from Microsoft Windows 7 for Msxml6.0.

If you have a header line, check the check-box **Treat first row as header.**

Press **More** button to see the source *.xml file, the related *.xls file and the result of the transformation. In order to see the **Result**, first press **play** button.
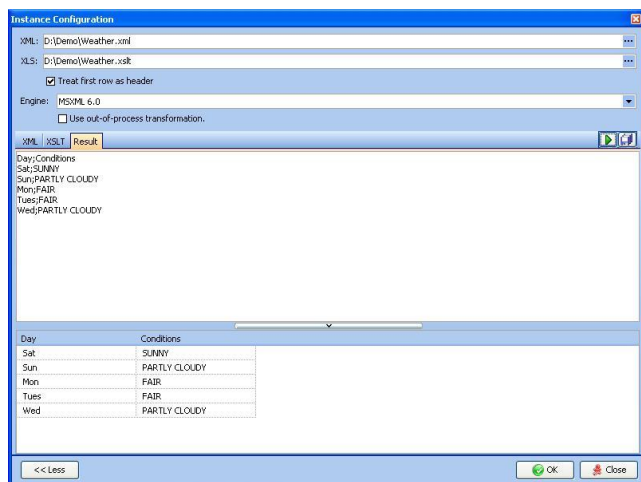
NOTE: If *.xml files used with XML data provider are web based files, it is needed to download these files locally on your PC first. For downloading, you can use a dedicated application (*Downloader.exe*) included in **TitleBox** installation.

You can find *Downloader.exe* in **TitleBox** folder. Start the application and the following window will open:
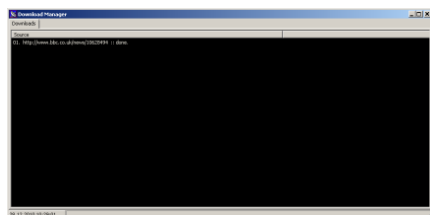
To add a new source for downloading, right mouse click on the black area and the set-up dialog will appear. Here you have to enter the URL of the source and the location for saving the file. If access to the source needs some Username and Password, enter them in the relevant fields.
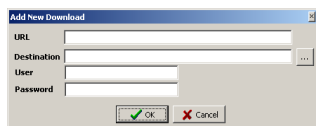
RSS Data Providers

Choosing this plug-in allows you to connect to RSS feeds from the Internet.

RSS (Really Simple Syndication) is an XML-based format for sharing and distributing Web content, such as news headlines for example. Currently many web sites provide RSS feeds.

There are 2 types of RSS data providers in **TitleBox.** The main difference between them is that *Data Provider I* shows RSS feed line by line into the linked object, while *Data Provider II* shows all lines of RSS source together into the linked object.

RSS/Atom Data Provider I

Choose it from the list of available plug-ins and press **Add** button. The following window will open.
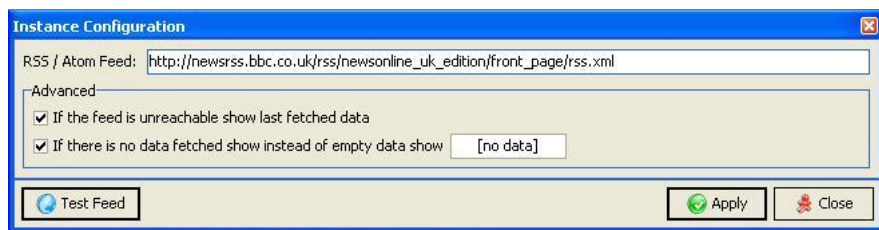
In **RSS/Atom Feed** field enter the URL of the needed RSS feed.

To check the connection, press *Test Feed* button. You will receive a message if the connection is successful.

There are two advanced options available for choosing the proper behavior if the RSS feed is not reachable or if there is no data.

Check the first check-box, into *Advanced* area if the RSS feed is not reachable, but you want to see the old data in the relative *TitleBox* object;

Check the second check-box, if there is no any data in RSS feed, but you want to have some text into related *TitleBox* object (like "No data available" or "please excuse us...", etc.).

Press *Apply* button to accept the settings.

Press *Close* button to close the window without changes.

RSS/Atom Data Provider II

As mentioned before, *Data Provider II* shows all lines of RSS source together into the linked object. The different lines are separated by some delimiter.

Choose Data Provider II from the list of available plug-ins and press *Add* button. The following window will open.
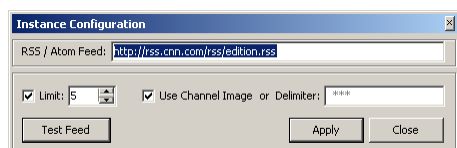
In **RSS/Atom Feed** field enter the URL of the RSS feed.

Check **Limit** check-box if you want to limit the number of displayed lines and enter the number of lines into the next field. By default, Limit check-box is checked and the number of lines is five.

Then enter the **Delimiter** sign. If the source channel provides the Channel image, ⬡ you can select **Use Channel Image** check-box. If it is checked, but there is no channel image, then the *AirBox* channel image will be shown by default. If you want to use your own picture as delimiter, you can use bitmap or movie tags into the field. Inserting of the bitmap or movie tags is the same like for text objects. See here for description.

NOTE: RSS data providers works with Internet Explorer only. You need Internet Explorer v.8.0 or higher.

NOTE: It is possible to see RSS data in *TitleBox* differently ordered than it is visible in the RSS source site. It is because some RSS sources (news channels for example) give opportunity to user to order the visible into browser data differently - by date or by title, etc., while *TitleBox* data-provider extracts the data from the original *.xml file of the RSS and there is no possibility to re-arrange their order.

TIP (!) You can use downloader.exe, described in XML data provider section for downloading RSS feed first. It is useful in case the link to the RSS breaks**.**

HTML Table Data Provider

Choosing this plug-in allows you to insert data into *TitleBox* object from an *.html file.

Choose it from the list of available plug-ins and press **Add** button. The following window will open.
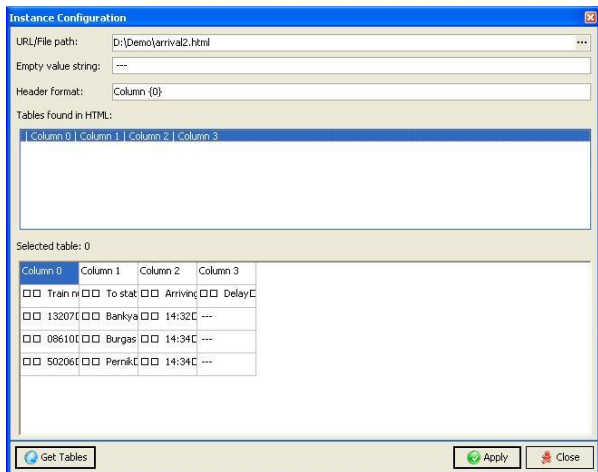
URL/File path – enter here the location of your *.html table

Empty value string – enter here the string, which you want to be visible if there are no data into the table.

Header format – this is the format of the table header. By default, it is "Column {0}"

Into **Tables found in HTML** area, you will see the list of tables found into the source *.html file. In order to see a table, you have to select a line and press **Get Table** button in the bottom of the window.

Press **Apply** button, to accept the settings.

Press **Close** button, to close the window without saving the changes.

EAS (Emergency Alert System) Data Provider

The **Emergency Alert System (EAS)** is a national warning system in the United Stated of America put into place in 1997. The EAS requires broadcasters, cable television systems, wireless cable systems, satellite digital audio radio service (SDARS) providers, and direct broadcast satellite (DBS) providers to provide the communications capability to the President to address the American public during a national emergency.

The system also may be used by state and local authorities to deliver important emergency information, such as AMBER alerts and weather information targeted to specific areas.

Each State and several territories have their own EAS plan.

The EAS regulations and standards are governed by the Public Safety and Homeland Security Bureau of the FCC (Federal Communications Commission).

All EAS equipment must be FCC certified for use.

TitleBox is connected to EAS decoder unit via LAN connection. The messages coming from EAS unit are connected to **TitleBox** objects via EAS data provider.

EAS data provider is started as service. It could be assigned to a text and to an audio object.

When you select EAS data provider from a list of data providers, the following setup win window opens:

Channel in a Box User Manual
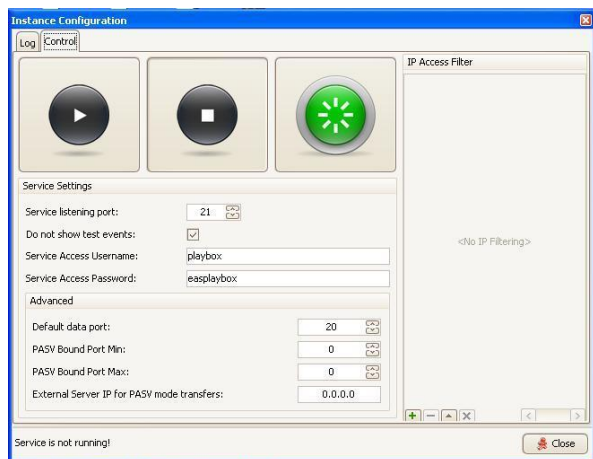
Here, you have to enter the IP address of the EAS unit, as well as some advanced details of the connection. If there is connection username and password defined, you have to enter them also.

When you are ready, press the **Play** button, in order to start the service.

NOTE: It is very important to set the proper time zone to the system and the system clock to be accurate, in order the service to work correctly.

IMPORTANT: *TitleBox* works withdecoders, produced of Digital Alert Systems, LLC with FCC ID: R8VDASDEC-1EN.

TASK MANAGER

Task Manager is an instrument for creating specific **tasks** in *TitleBox*.

A **task** is an action, which is executed in *TitleBox* (like play object, stop object, etc.). The **Task** usually is performed, when happens a specific condition. The condition (trigger) on which **Task** is executed is defined by user.

The condition (trigger), on which the **task** is performed, is called **event**. An **event** could be the object's status (like play, stop, etc.) or a command (like incoming GPI signal; *TitleBox* internal command, etc.).

First you have to create a **task**, and then you can assign this **task** to an **event**.

Suppose that the user wants to start *Object1* when *Object2* stops. Then the **task,** which have to be created is "*Start Object1*" and the condition, i.e. the **event** is "*OnStop Object2*".
Another example is, if the user wants to start *Object1*, when a *GPI* signal comes. Then the **task** is "*Start Object1*" and the **event** is incoming the *GPI* signal.
Another example: To generate a *GPI* signal when *Object1* starts, then the **task** is "*GPI*" and the **event** is "*OnStart Object1*".

Tasks

For creating **tasks**, you can start the **Task Manager** from **Project** menu ➔**Plugins**➔**Task Manager.**

The window will open also if right mouse click over an object in *TitleBox* working area and select **Task Manager** from the list.

227

Channel in a Box User Manual
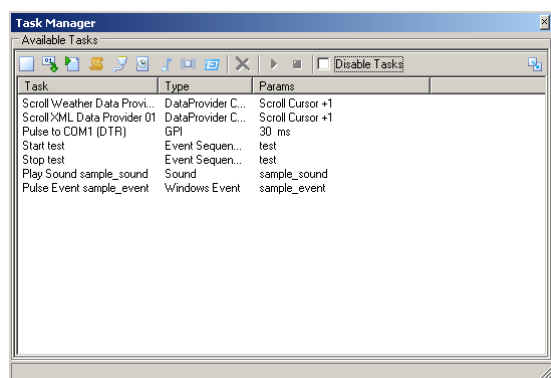
All **tasks** are visible into the **Task** list.

To create a new **task**, press the related icon from the menu bar.

To delete a **task**, select it and press ![x] delete button from the menu bar.

The **tasks** could be executed manually (for test purposes). Press ![play] play button to start a **task** and stop ![stop] button to stop a **task**.

The available **tasks** are:

🔲 Object Control

It provides options for creating object-related **tasks**: *Play* Object; *Stop Object; Toggle Play/Stop; Pause object; Move Object.* Select one of these options and then select an object from a drop-down list at the right of the window. In this drop-down list are listed all objects existing in the current project. Press **OK** and you will see your **task** in the **task** list.
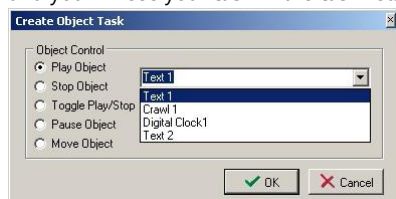
If you select the ⊙ **Move Object** control, an additional window will open where to enter the moving options.

Select ⊙ **Move to** and enter **X** and **Y** values, to move the object to a position with exactly these **X** and **Y** coordinates.

Select ⊙ **Increment position** and enter **X** and **Y** values, to increase the existing coordinates of the object with these values.

Repeat the procedure till you have all necessary **task**s related to objects.

DataProvider Control.

Here you can create **tasks** related to data providers.

At the first field in the window, you have to select a data provider from a list of all existing data providers. Then you can choose an activity for this data provider, like ⊙ *Stop*, ⊙ *Start*, ⊙ *Update* data provider.

If there are more than one row in the data provider source (like in RSS data provider or in ODBC data provider), then you can select also ⊙ *Cursor scroll* –move cursor to next row; ⊙ *Cursor top* – move cursor to the first row and ⊙ *Cursor bottom* – move cursor to the last row.

If ⊙ *Cursor scroll* is selected, an additional field is visible. There you can enter the number of rows to scroll cursor.

Command Script task

This command is used, if you want to create a **task** based on *TitleBox* internal commands. You can refer to the *TitleBox* API to learn more about *TitleBox* internal commands. In the Command Script dialog you can press <Ctrl+F1> to invoke the API help and <Ctrl+Space> to invoke the list of commands. To exit from the two, press <Esc>.

For example, you can create a *Script* for loading a new project in *TitleBox*:
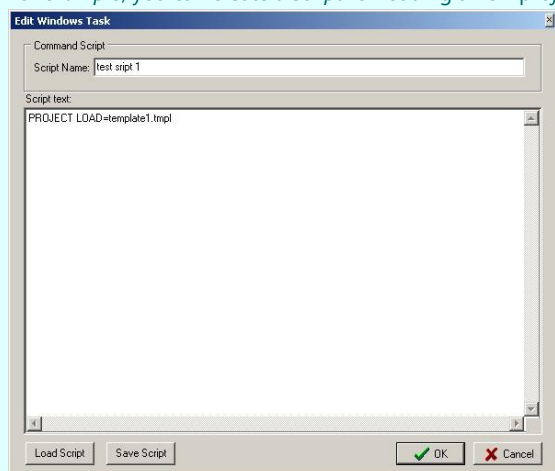
"PROJECT LOAD=template1.tmpl"

Further you can assign this **task** to any of the existing **events.**
Enter the name of the **task** in the **Script Name** field. Be aware that if you save your script with a certain name, recognized by *TitleBox*, it will be executed automatically under given conditions. Check the *Auto-executed Scripts* function, described below.
Enter the *TitleBox* command into **Script text** field.
Press **OK** button to create the **task.**
If you want to save the script from created **task** in a file, press the **Save Script** button
If you want to use an existing script, press **Load Script** button and select the script file.

IMPORTANT: In order to use the Script task properly, you need to be familiar with the TitleBox API commands, which are listed and explained in the **TitleBox API User Manual**. To obtain the latter, please, contact our support team at support@playboxtechnology.com.

Program Script Task

The *Program Script Task* is an extended feature in *TitleBox*, which not only gives you the opportunity to write program scripts for all the tasks that can be done in *TitleBox* by writing a code for them but it also allows you to make your projects even more functional and user friendly. Note that in order to take full advantage of this *TitleBox* option, you need to be an advanced user of the program and you must have at least a basic programming knowledge.

This *TitleBox* feature supports four different language syntaxes:

- Pascal script
- C++ script
- Visual Basic script
- Java script

TIP (!): Considering the *TitleBox* functionality, it is advisable that you use **Pascal** or **C++** script, instead of **Visual Basic** or **Java** script. Since you always have to declare the type of a variable in **Pascal** and **C++**, the latter are more thorough in terms of avoiding mistakes. In **Visual Basic** and **Java** script, on the other hand, new variables are not defined in terms of their type. They simply adopt the variant type and can accept all kinds of variables, no matter if they are string, integer, etc., throughout your script, which inevitably becomes a prerequisite for errors.

Please, be aware that *TitleBox*'s *Program Script Task* does NOT support all the functionalities that each of the programming languages, listed above offers. All the **classes**, **functions**, **types**, and **variables** that this interface supports, are listed in the tree view, situated in the right area of the *Create Program Script* window.

IMPORTANT! *PlayBox Technology*, does NOT offer support for third party scripts, developed with the **Program Script Task**. This function should be used at the sole responsibility of the corresponding third party developer.

Channel in a Box User Manual

User Interface

Work Area

Pressing the **Program Script** button in the **Task Manager** window invokes the following user interface:
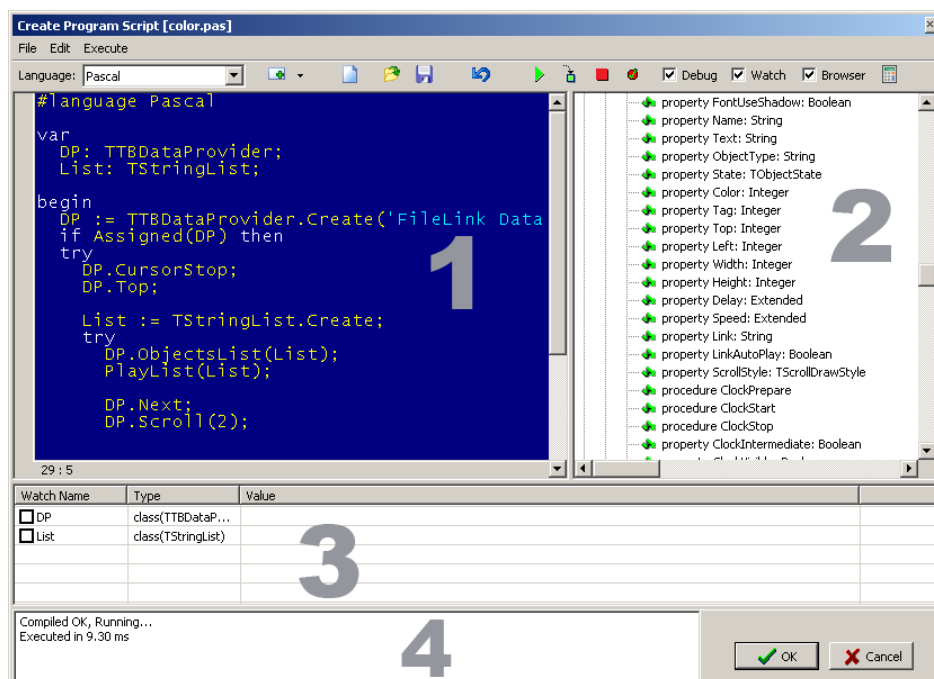
1. Program View
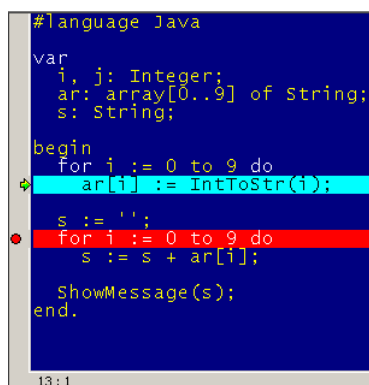
This is the area, marked with number '1' in the picture above. This is the place, where the user enters their script. Depending on the programming language, selected from the *Language* drop-down menu above it, the script will automatically load a language identifier in the beginning of the script window, which is in the following format: [#language Pascal/C++/Basic/Java], depending on the language selected.

Note that the grey area to the left shows certain events, added manually in the script, like **Execute next step**, and **Toggle breakpoint**, as explained in the Toolbar section below.

Also, the grey area at the bottom of the program script view shows the position of the cursor, [13:1] in this example. This information is useful when debugging, since the errors are indicated in accordance to these position coordinates in the *Debug* view.

2. Browser View

This is the area, marked with number '2' in the picture above. This view shows you a list of all the [Classes], [Functions], [Types], and [Variables], available for you to include in your program script. You can expand/shrink the list of each of the above items by pressing the plus/minus sign next to it, or by double-clicking on them.

The Tree View supports **drag-n-drop** and **double-click** functionalities, as explained below:

– *Drag-n-Drop*

If you drag-n-drop an item, designated with 📗 or 📖 in the program script, only the class type will be implemented in the script. For example, if you drag-n-drop the [TWinControl = class(TControl)] to the script, only [TWinControl] will be implemented in the script view.

If you drag-n-drop an item, designated with 🔧 or *fx* in the program script, the whole line will be implemented in the script. Thus, if you drag-n-drop the [property Cursor: Integer], the whole [property Cursor: Integer] line will be placed in the script view.

– *Double-Click*

Besides drag-n-dropping, you can also double-click on the items from the Tree View. If you double-click on a 📗 or a 📖, the list will correspondingly expand or shrink.

If you double-click on an item, designated with 🔧 or *fx* on the other hand, it will appear in your script as a name and brackets next to it with the appropriate number of places for you to fill in, separated by commas. For example, if you double-click on the function [DirectoryList(const Path: String; List: TStrings)], the following will be implemented in the program script: [DirectoryList(,)].

The *Browser* area is only visible while the ☑*Browser* box in the toolbar is checked.

3. Watch View

This is the area, marked with number '3' in the picture above. It allows you to have an additional control over your program via "watching" how your variables behave in your script.

The watch area consists of three columns:

Watch Name – this column shows the name of the item/variable that is being watched

Type – this column shows the type of the corresponding item/variable (like integer, string, float, etc.)

Value – this column shows the current value of the corresponding item/variable. Note that if the script has been executed up to a certain line only, the entry will represent the value of the item until this step

Right-clicking on any line in this area will invoke the following context menu:
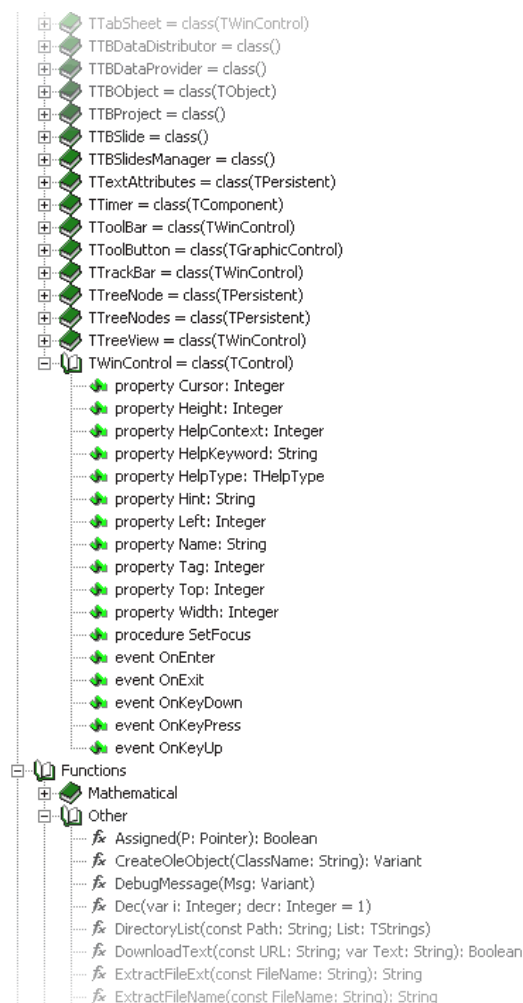
If you select **Add** or **Edit**, the ***Watch Properties*** dialog will open. Another way for invoking this dialog is by pressing <Ctrl> + <F7>. In the *Expression* field enter the name of the item you wish to watch. Select the type of the variable by pressing one of the radio buttons below. If you choose ⊙***Auto***, the type of the variable will be obtained automatically from the script, if possible. If you wish to transform the result in another value type, you can select a type, different from the default type of your variable, as long as the two types are compatible.
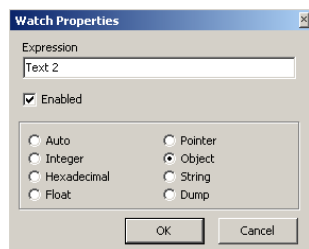
Checking the ☑*Enabled* box will add a check-box to the corresponding row in the *Watch List*. The idea of this check-box is to give the user the ability to control which variables should be "watched," and which should not be "watched" at a time. Thus, if an item is not checked☐, its value in the **Value** column will not be changed until the box next to it is checked ☑back.

The **Delete** and the **Clear** commands from the context menu are self-explanatory.

The *Watch* area is only visible when the ☑*Watch* box in the toolbar is checked.

4. Debug View

This area shows all the syntax errors that appear in the script while debugging. Note that logical errors will not be traced by the script engine.

The *Debug*mode is only active when the ☑*Debug* box in the toolbar is checked.

> **NOTE:** The ***TitleBox*** script engine allows you to check whether or not you are in *Debug* mode via a special variable, named [DEBUG], which is of type Boolean. When [DEBUG = True], this means that you are in *Debug* mode, and when [DEBUG = False], the Debug mode is off, so the script is executed without interruptions. Please, check *Example 13* in *Appendix 7* below.

Toolbar:
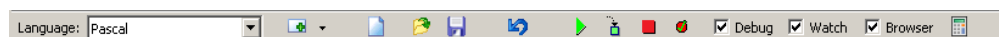
Use the drop-down menu in the toolbar for selecting the programming *Language*, in which you are going to write your script. Your options are ***Pascal***, ***C++***, ***Basic***, and ***Java***.

Several buttons and check-boxes are situated after the drop-down menu. Their functions are as follows:

 - This is the ***Insert*** button. It is used for inserting **Modules**, **Objects**, **Slides**, **Tasks,** and **Data Providers**. When you press the black arrow next to it, a context menu will appear:
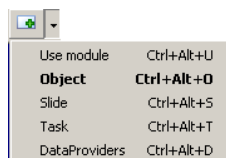
Here you can select the type of the item you would like to insert in your script. Notice that there is a combination of keys written next to each item. Thus, instead of clicking on the respective item from the ***Insert*** drop-down menu, you can simply enter the corresponding key combination. Furthermore, note that in the screenshot above the second line [Object] is marked with a bold font. This means that when you press only the ***Insert*** button instead of the arrow next to it, an **Object** will be added to the script. If, next time you insert an item you select [Slide] from the context menu, its line will be bolded and a **Slide** will be added after clicking only the button instead of the arrow.

If you select the **Use module** option from the context menu above, you can insert any script, situated in *TitleBox*'s Library, regardless of the language, in which it is written. This option is very useful if you need to use the same function in different scripts. Please, check *Example 15* in *Appendix 7* [below](#) to see how modules are inserted in different script syntaxes.

 – Use the **New** button to clear the *Program View* and start working on a new script.

 – Use the **Open Script in editor** button to load an already saved script.

 – Use the **Save Script to file** button to save your script.

 – Use the **Undo changes** button to undo your latest changes. You can also press the <Ctrl>+<z> keys instead of this button. Up to 60 steps can be reversed.

 – Use the **Run Script** button to debug your entire script. You can also press the <F9> key instead of this button.

 – Use the **Execute next step** button to only execute the line after the one, where the cursor is situated. You can also press the <F8> keys instead of this button.

```
#language Pascal

var
  i, j: Integer;
  ar: array[0..9] of String;
  s: String;

begin
  for i := 0 to 9 do
    ar[i] := IntToStr(i);

  s := '';
  for i := 0 to 9 do
    s := s + ar[i];

  ShowMessage(s);
end.
```

In the example above, please notice the light-blue line with a green arrow in the beginning. After a step is executed via the button, described above, this step is marked like that.

 – Use the **Stop execution** button to stop debugging your script. You can also press the <Ctrl>+<F2> keys instead of this button.

 – Use the **Toggle breakpoint** button to insert a breakpoint in the script. You can also press the <F5> key instead of this button. When a breakpoint is added, the next time you start executing your script, it will pause executing at that breakpoint.

```
#language Pascal

var
  i, j: Integer;

begin
  j := 1;
  i := 0;
  while i < 10 do
  begin
    j := j + 1;
    Inc(i);
  end;

  ShowMessage(j);
end.
```

In the example above, please notice the red lines with a red circle in the beginning. After a breakpoint is included, the line that corresponds to the breakpoint will be marked like that.

Also, be aware that you can add a breakpoint also by clicking in the grey area to the left of the script view. If you want to remove that breakpoint, simply click on it again.

☑ *Debug* – If checked, the script will be debugged when executed. Otherwise, only the final result will be shown and no messages will appear in the *Debug* area at the bottom.

☑ *Watch* – If checked, the *Watch* area will be visible.

☑ *Browser* – If checked, the *Browser* area will be visible.

🔲 - The **Evaluate expression** button invokes the dialog, shown below. In the *Expression* field you can enter different mathematical expressions, in which you can include variables from your script. After pressing <Enter>, the program will show you the result of the expression in the *Result* field. If you have entered a variable and/or other symbol that cannot be recognized by the script engine, it will return an error message in the latter field.
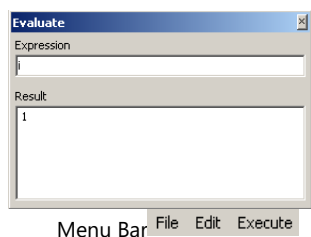
File

The File menu consists of an **Open**, **Save**, and an **Exit** command, which are all self-explanatory. Be aware that your script can be automatically executed if you **Save** it under a name, recognized by *TitleBox*. Check the **Auto-executed Scripts** function, described below.

Edit
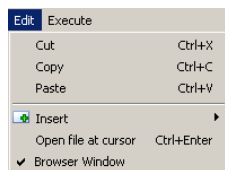
The edit menu consists of several commands. The first three, **Cut**, **Copy**, and **Paste,** are self-explanatory.

The **Insert** command is the same as the insert button, explained above in the Toolbar section.

The **Open file at cursor** command simply allows the user to load a script at the current location of the cursor. As visible in the screenshot above, this command can also be invoked with the <Ctrl>+<Enter> key combination.

The **Browser Window** command is the same as the ☑*Browser* checkbox in the Toolbar, explained above.

Execute
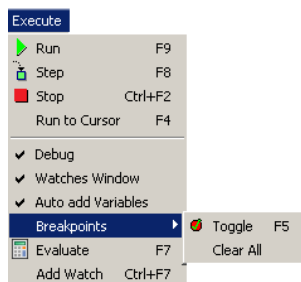
The Execute menu is very similar to the Toolbar, explained above. However, it offers some additional commands, explained below. Also, it shows the equivalent keyboard buttons that could be used for the commands.

The **Run to Cursor** command is not included in the toolbar. When selected, this command will execute the script to the point, where the cursor is currently situated.

This menu also offers an additional function for the **Toggle** breakpoints. If you go to **Breakpoints**, a context menu will be opened, from which you can either add a breakpoint (**Toggle** command), or remove all breakpoints, via the **Clear All** commands.

If checked, the **Auto add Variables** option will automatically add all variables used in the script to the *Watch View* at the bottom of the **Create Program Script** dialog.

<u>**Creating TitleBox Items**</u>

Take a look at the Browser View in the Create Program Script dialog. When you expand the classes section, a large list of classes appears. The classes are arranged in alphabetical order. Most of them are self-explanatory and refer to common programming options and functions. Several of them classes, however, were created strictly for *TitleBox*:

TTBObject
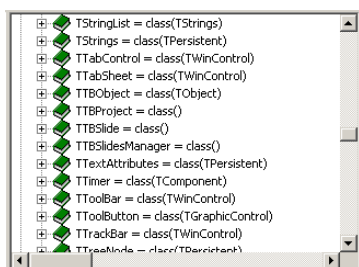
This class operates all objects in *TitleBox*, Text objects, Pictures, Rolls and Crawls, 3D objects, etc. Be aware that the possible properties for all *TitleBox* objects are listed here. Some of them are strictly related to a particular type of object. For example, you cannot apply [ClockPrepare] to a Text object. If you try to apply a property or method that the object in question does not support, the script engine will simply ignore it.

There are two ways, in which you can create an object in *TitleBox*: with the **Create** and with the **CreateNew** command.

The **Create** command is used when an object has already created in the *Work Area* via the *Object Palette*. Thus, when you **Create** an object, you simply link your script command with that object by using the object name. If you use this option for creating objects, instead of typing in the name of the particular object, you can simply press <Ctrl> + <Alt> + <O>, which will invoke a dialog with a list of all objects in the project, like the one shown below. To select an object to link to, double-click on its name in the list.
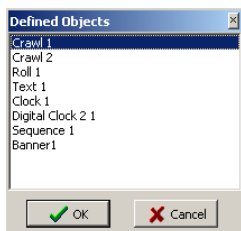
The **CreateNew** command, on the other hand, is used when the object you want to create does not already exist in your project. When you create it, you can define its X and Y parameters, as well as its width and height. If you do not define any parameters, it will be placed at the (0,0) coordinate in the grid.

A description of all properties, procedures, and functions that the [TTBObject] class supports is available in *Appendix 6* <u>below</u>. Furthermore, you can check *Example 1*, *Example 2*, *Example 5*, and *Example 10* in *Appendix 7* below to see how it works in practice.

NOTE: If you try to create a new object with the name of an existing one, be aware that *TitleBox* will automatically add '1' to the end of the name, so that it does not coincide with the already existing name.

IMPORTANT: You should always set your objects free in your script via the **Free** command after you stop working with them. This is done to prevent your memory from overloading. Be aware that the **Free** command does not delete an objects. It simply breaks the link between the object and the program script.

NOTE: If you want to remove an object both from the script, and from the grid, you should use the **Delete** command.

> **TIP (!)** Place your **Free** command after the **Try–Finally** command combination to make sure that the **Free** command is executed.
>
> **TIP (!)** If you are using the [TTBObject] class to update text formatting in a certain text object, do NOT forget to use the **UpdateParams** command to apply the new formatting to the text. Check *Example 5* in *Appendix 7* <u>below</u>.

TTBSlide

This class is used in the same way as the [TTBObject] class. You can create new slides (**CreateNew** command) and link to already created slides (**Create** command), and you need to set your Slide free after you stop working with it (**Free** command).

A description of all properties, procedures, and functions that the [TTBSlide] class supports is available in *Appendix 9*<u>below</u>. Furthermore, you can check *Example 4*, *Example 5* and *Example 9* in *Appendix 10* below to see how it works in practice.

> **NOTE:** If you attempt to create a new slide with the name of an already existing one, *TitleBox* will return an error in the *Debug* view.
>
> **NOTE:** If you use the **Delete** command to delete a slide, only the slide will be removed and not the objects, contained in it. They will simply be moved to the 'zero' slide, which contains all objects that are not assigned to a particular slide.

TTBSlidesManager

The [TTBSlidesManager] class is used for operating with the slides manager. Like the [TTBProject] and [TTBStorage] class, the user does not create, nor free this class, as it stores only one variable, [Slides], which is created upon *TitleBox* initialisation.

A description of all properties, procedures, and functions that the [TTBSlidesManager] class supports is available in *Appendix 7*<u>below</u>. Furthermore, you can check *Example 9* in *Appendix 7* below to see how it works in practice.

TTBDataProvider

The [TTBDataProvider] class is used for operating with the data providers that are already set in the <u>Data Source Manager</u>. Be aware that when you **Create** such a [TTBDataProvider] variable and link it to a certain provider, the variable in the script will control all the objects that are linked to that particular Data Provider. Furthermore, you need to set the [TTBDataProvider] variable **Free** after using it. This class is often used with <u>Transform Events</u>.

A description of all properties, procedures, and functions that the [TTBDataProvider] class supports is available in *Appendix 6* <u>below</u>. Furthermore, you can check *Example 6*, *Example 7*, *Example 8*, and *Example 12* in *Appendix 7* below to see how it works in practice.

TTBDataDistributor

The [TTBDataDistributor] class is used for operating with the data distributor properties that are already defined for each Data Provider in the <u>Data Source Manager</u>. This variable is NOT set free after operation, since it is freed together with its corresponding Data Provider.

A description of all properties, procedures, and functions that the [TTBDataDistributor] class supports is available in *Appendix 6* <u>below</u>. Furthermore, you can check *Example 7* in *Appendix 7* below to see how it works in practice.

TTBProject

This class is used for controlling your whole Project. As it stores only one variable, [Project], which is created upon *TitleBox* initialisation, you do not create, nor do you set free a [TTBProject] class.

A description of all properties, procedures, and functions that the [TTBProject] class supports is available in *Appendix 9* below. Furthermore, you can check *Example 10* in *Appendix 7* below to see how it works in practice.

TTBStorage

The [TTBStorage] class is used for storing specific values to be used in different scripts and / or projects within one **TitleBox** session. It allows the user to save up to 100 values and it is extremely useful when you need to use the same values in more than one script or even a whole **TitleBox** project. As it stores only one variable, [Storage], which is created upon **TitleBox** initialisation, you do not create, nor do you set free a [TTBStorage] class.

A description of all properties, procedures, and functions that the [TTBStorage] class supports is available in *Appendix 6* <u>below</u>. Furthermore, you can see how it works in practice in *Example 14* in *Appendix 7* <u>below</u>.

<u>Additional Tips and Notes</u>

There are two additional classes, which are not **TitleBox** specific but are commonly used in the **TitleBox** scripts for various operations and functions. They are the [TRichEdit] and the [TBitmap32] class.

The [TRichEdit] class is often used with all **TitleBox** text objects, especially with procedure TTBObject.TextAssign(RichEdit: TrichEdit), described in *Appendix 6* <u>below</u>. It allows you to change all kinds of formatting in a given text selection.

The [TBitmap32] class is used for creating and controlling Bitmap32 images. It is often associated with procedure TTBObject.ImageAssign(Bmp32: TBitmap32), described in *Appendix 6* <u>below</u>. A Bitmap32 image will not be shown in the **TitleBox** work area, unless you paste it in a certain object in the grid. These variables need to be set **Free** after using in the script.

A description of all properties, procedures, and functions that the [TBitmap32] class supports is available in *Appendix 6* <u>below</u>.

For your convenience, check the table below for the major functions of the classes, specifically created for **TitleBox**. It also shows whether or not they have to be set Free in the script after operation:

Class	Function	Free
TTBObject	To be linked to or create all types of object in **TitleBox**.	Yes
TTBSlide	To be linked to or create slides in your project.	Yes
TTBSlidesManager	To be linked to the <u>Slides Manager</u> in your project.	No
TTBDataProvider	To be linked to all Data Providers set in the <u>Data Source Manager</u>.	Yes
TTBDataDistributor	To be linked to all Data Distributors, set in the <u>Data Source Manager</u>.	No
TTBProject	To be linked to your current project in **TitleBox**.	No
TTBStorage	To save certain variables to be used in different scripts and / or projects within one **TitleBox** session.	No

TIP (!): If you are not certain whether you need to **Free** a certain variable or not, you can check the list of functions, procedures, and properties, related to its class in the <u>Browser View</u>. If [procedure: Free] exists there, then you have to free that particular variable. Otherwise, the variable does not need to be set free.

Working with More than One TitleBox Item Simultaneously

If you want to operate with more than one *TitleBox* object or slide at a time, the script engine allows you to group your desired items via the List commands. The Program Script Task has two types of list commands – for grouping of objects and for managing already grouped objects.

Create Lists of Items

The program script engine allows you to operate with several items together via the List variables. In order to do that, first, you need to define a set of objects or slides to belong to a certain list. For this purpose, you can use the available List commands, which are situated within the TTBProject and TTBSlidesManager classes:

ObjectsList – This command is situated in [Classes → TTBProject] in the *Browser Area*. It lists all the objects in the project and you can choose which objects you wish to keep in the list, with which you are going to operate.

TagsList – This command is also situated in [Classes → TTBProject] in the *Browser Area*. It lists all objects that share the same tag. You can read more about Tags in TitleBox in the relevant section above.

TasksList – This command is also situated in [Classes → TTBProject] in the *Browser Area*. It lists all tasks in your project.

SlidesList – This command is situated in [Classes → TTBSlidesManager] in the *Browser Area*.It lists all slides in your project.

DataProvidersList – This command is situated in [Classes → TTBProject] in the *Browser Area*. It lists all the objects in the project and you can choose which objects you wish to keep in the list, with which you are going to operate.

Manage Lists of Items

Once you have grouped your desired items via one of the **List** commands, described above, you can work with these groups within your script. There are three **List** functions, available in the script engine, [PlayList], [StopList], and [ToggleList]. All of them are situated in [Functions → TitleBox].

Please, check *Example 3*, *Example 6*, *Example 7*, and *Example 9* in *Appendix 7* below to see how to use the *TitleBox* **List** commands in practice.

Error and Exception Messages

For the purposes of avoiding memory load and endless scripts, two error checks are implemented in the *TitleBox Program Script*:

Object Free Check

If some of the objects, used in the script have not been set free with the **Free** command, the *Program Script* will show an error message, like the one in the image below, listing all the variables that have not been freed during the script execution.
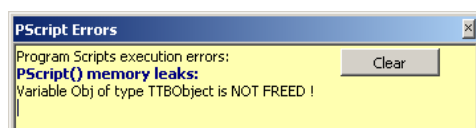
Execution Time Check

If the execution of a certain script takes more than 1 - 2 seconds, the *Program Script* will terminate the script execution and return an error message, like the one below, stating at how many seconds the execution has been stopped.
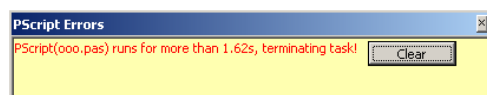
Exception Messages

Sometimes it is very useful to work with Exception Messages and the **Try – Except / Try – Catch** block, if you are working with a certain procedure in your script, which does not work in some exceptional cases. You can check *Example 11* in *Appendix 7* below to see how it works in practice.

GPI Task

This **task** is designed for creating **GPI** pulses.
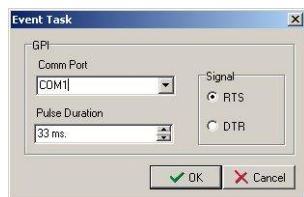
Into the GPI **task** window, specify the GPI *Signal* type (RTS or DTR) and to which COM port it should be sent. Specify also the Pulse duration in milliseconds.

Press **OK** to add this **task** to the **Task Manager** list.

The GPI events window can be started from **Project->Plugins-GPI manager** as well.

In the GPI manager, you can view all GPI events and the objects to which they have been assigned.
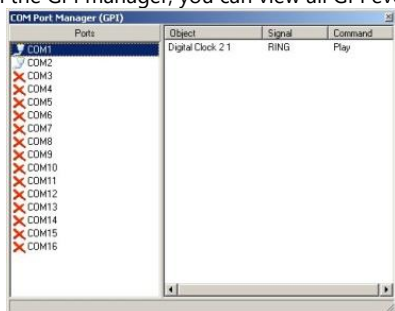
Event sequencer Task

This **task** is designed for generating a time sequence for executing another **task.**

An *example* of such sequence is, if you want to start a Digital Clock, then to start a picture and move it horizontally on screen over a specific period of time and at finally to start another task. You will have the following sequence of tasks:

1. **Task** *"Play Digital Clock 1"*
2. **Task** *"Play Picture 1"* (to be started 5 sec. after the beginning of the sequence)
3.**Task** *"Move Picture1"* (to be started 10 times at every 5 seconds)
4. **Task** *"Start timer1"*

In this example, you have to fill the setup window like it is shown in the following screenshot:
Into Task Name, enter the name of your time sequence (sequence 1).
Into OnStart field, select the name of the first task in the sequence (Play Digital Clock 1). Select the name from the drop-down list. If you don't want to have task on sequence start, uncheck relative Enable check-box.
Into Delay before Loop field, enter the interval of delay in milliseconds (5000ms=5sec.). This is the delay after the sequence start and before the start of the loop task. Into OnEndDelay field enter the task which will be executed with this delay (PlayPicture1). Select the name of the task from the drop-down list with existing tasks. If you don't want to have delayed task, uncheck relative Enable check-box.
Into OnLoop field, enter the task which will be executed on every loop (Move Picture 1). Into Loop Interval field enter the interval for repeating the task

(5sec=5000ms). Into Loops Count field enter how many times the task will be repeated (10).
If Loops Count is zero (0), the looping will be infinite, until task is running.
Into OnStop field, enter the task which will be executed at the end of the sequence (Start timer1). This task will start when the looping is finished or when a Stop sequence command is send.
When you are ready with definition of the sequence, press the OK button to save it. You will see two tasks in the task list, created automatically: Start sequence name and Stop sequence name. The command Stop sequence name usually is used to stop a sequence when the Loop Interval is zero (infinite).

Play Sound

It is designed for creating sound **tasks**. Pushing the icon will open the **Play Sound** dialog and all you have to do is to browse for the sound you need. You can open all Direct-Show compatible sound files (*.wav, *.mp3, etc.*). You can preview all sound **tasks** in the **Task Manager** using the ***Play*** and ***Stop*** buttons.

Windows Named Event Task

This **task** is related to Windows Named Event Objects. For more information about them, you can refer to Windows documentation.

The **Event Name task** was developed to synchronize the external applications to ***TitleBox***. Thus, if there is a third party application that can accept certain event names, ***TitleBox*** can trigger events in these applications.

Each system event should have a Name. You can type it in the ***Event Name*** string. Below, you have to choose the ***Action*** and the type of ***Initialization***.

Create Windows Task
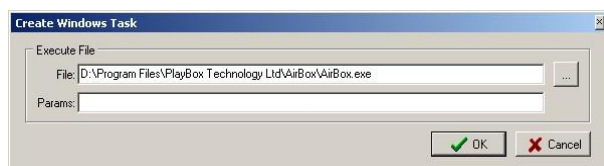
This **task** is related to executing of the external applications.

Into field **File** enter the executable file name (including the full file's path). You can start *.exe *.bat or *.cmd files.

Into **Params** field, you can enter some parameters for starting the application.

For example, if in the **File** is entered ***AirBox***, into **Parameter** you can enter the name of the playlist which you want to start or the ***AirBox*** instance number.

If there is a file association existing, you can enter into the File field, only the file name of the document to start the associated application.

For example, if you have a file association for ***Windows Media Player***, into field **File** you can enter only the location and name of the *.mp3 file.

Auto-executed Scripts

TitleBox provides an additional way to assign scripts to certain events. If you want a certain script, created either via the <u>Command Script Task</u>, or by the <u>Program Script Task</u> to be executed when you **load**, **unload**, **start**, or **stop** your project, simply save the corresponding script with one of the following names:

- "**OnProjectLoad**" –this script will be executed right after a project is loaded;

- "**OnProjectUnload**" –this script will be executed right before unloading a project;

- "**OnProjectStart**" –this script will be executed right before starting a project;

- "**OnProjectStop**" –this script will be executed before stopping project.

> **NOTE:** If you have more than one script, bearing one of the above listed names, *TitleBox* will only execute the first one from the list, i.e., the oldest one.

Input Events
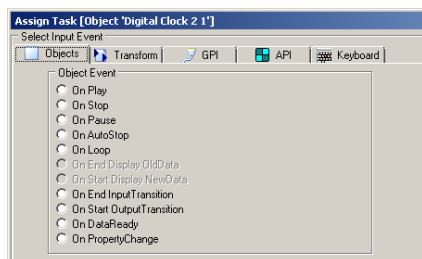
The **events** are triggers, on which the **tasks** are performed.

The **Input Event** dialog opens when you select an object, right mouse click and select **Task Manager** from menu.

The input events are grouped in four major groups: Object related events, GPI events, API events, Keyboard events.

Objects events

The available object's **events** are: Play, Stop, Pause, AutoStop, Loop, End display old data, Start display new data, End Input transition, Start Output transition, Data Ready, Property Change.

AutoStop – this event concerns rolls, crawl, and animation objects. This event is generated when a number of loops for the object is set and the last loop is executed.

End display old data and Start display new data events are connected to changing of the content into the rolls, crawls and animations. Because of the nature of these objects, it is possible to have both – old content and new content on screen. The End display old data event happens when the old data stops to be visible on the screen. And the Start display new data event happens when new data become to appear on the screen.

Data Ready – this event is related to changing the data into an object. When the new data is ready to be displayed, than Data Ready event is happening.

Transform events
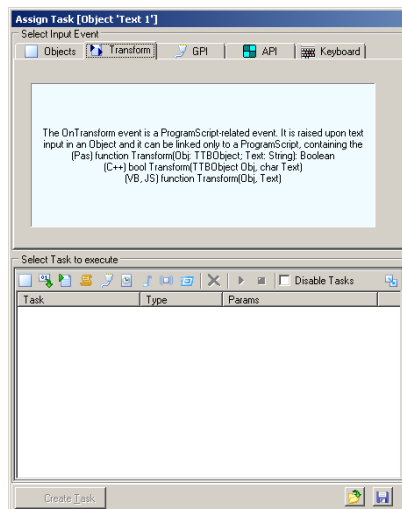
As explained in the image above, these events are used in combination with the Program Script Task. In other words, the only way to use such an event is to link it to a certain **Program Script** by entering the following function in the script:

For **Pascal** script: <**functionTransform(Obj: TTBObject; Text: String); Boolean**>;

For **C++** script: <**bool Transform(TTBObject Obj, char Text)**>;

For **Java** and **Visual Basic** script: <**function Transform(Obj, Text)**>;

The **Transform** event is used for transforming the information that is fed by a Data Provider, Net Control, or File Link to **TitleBox**. You can link such an event to all types of objects in **TitleBox**. Thus, you could change the format of the text that is provided by the data source, erase some parts of it, add new text to it, or even assign a picture to be displayed instead of the text that is being fed.

In order to create a **Transform** event for a certain object, two things need to happen. First, you need to save a program script, which contains the abovementioned transform function with a reference to that particular object in the Program Script Task. Then, you need to click on that object, press the **Task** 📤 button on the toolbar, and go to the **Transform** tab. Select the script that contains the corresponding **Transform** function from the list in the *Select Task to execute* area, and press the Create Task button.

Notice that the **Transform** function is of type **Boolean**. This means that you should implement a **True** or a **False** value to be returned by your script to the particular object, containing the **Transform** event. If a **False** value is returned to the object, it will ignore the **Transform** event and output the information, as provided by the **Data Provider** linked to it. Otherwise, if the object receives a **True** value from the **Transform** function, it will show the output as it is provided by the latter.

Please, check *Example 12* in *Appendix 7* below to see how the **Transform** function works in practice.

IMPORTANT: The **Transform** event only executes the **Transform** function, shown above, NOT the whole script. Thus, you do NOT need to save a separate script file, containing the transform function only. You can simply add the desired transform function to any program script that you use for your project.

NOTE: If the script that you assign to the **Transform** event does NOT contain a **Transform** function, **TitleBox** will simply ignore this event. Rather, it will display the corresponding **Data Provider** entry exactly as it is being fed to the object, linked to it.

NOTE: If there are any modal windows (e.g., Show Message dialogs) within the **Transform** function, **TitleBox** will ignore them. This precaution is introduced for the purpose of preventing the **Transform** function execution from postponing.

GPI events
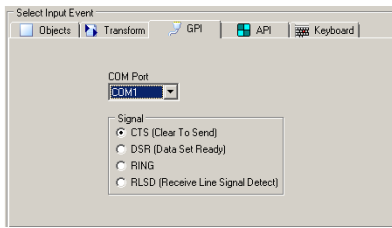
This feature in *TitleBox* allows controlling objects through receiving certain signals on the PC COM ports. In order to specify the "meaning" of each signal to each COM port, select the Com port and then select the type of signal.

API events

These events are connected to usage of *TitleBox API* (Application Program interface).

In *TitleBox API*, there is a command "EVENT xxx", where xxx is a number equal or higher than 1001. In **Task Manager->API events**, you can define a specific number which to be used further in such Event command.

When to the API event (*for example: number=1001*) is assigned a task and *TitleBox* receives a command "EVENT xxx" (*xxx=1001*), then the task assigned to this API event will be executed

KeyBoard events
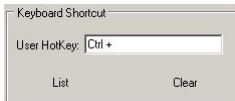
Keyboard events are user definable keyboard shortcuts for some action.

These events are connected to some **task**, so when the keyboard combination is pressed, the assigned **task** is executed.

Press the **List** button to see all created keyboard shortcuts.

Press the **Clear** button to delete all created keyboard shortcuts.

Assigning a task to an event
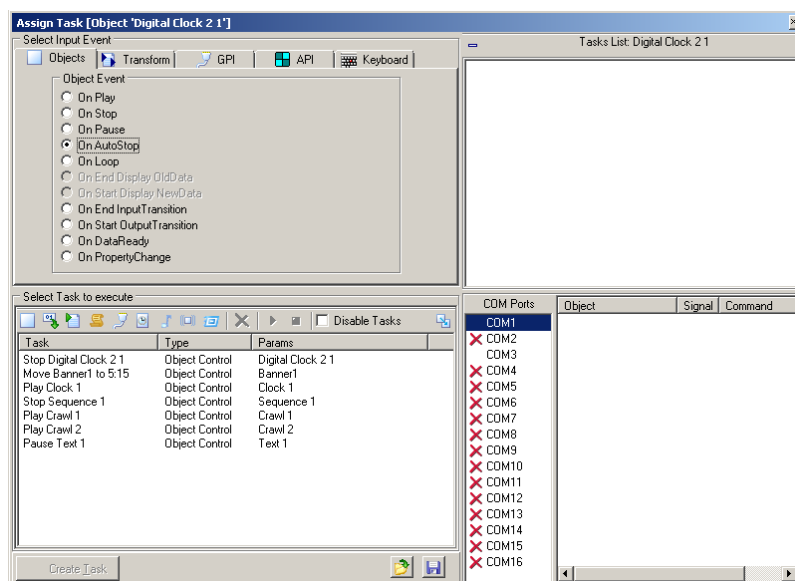
To assign a **task** to an **event**, you have to open the **Assign Task** dialog. It is available after right mouse clicking on an object in *TitleBox* project and selecting **Assign Task** from menu.

You will see the list of already created tasks. If there are no tasks, first create them into **Task** area of the window.

Select an **event** from the available events and then select a **task** from **Task** list. Press the **Create Task** button in the bottom of the **Assign task** window

In the right side of the **Assign Task** window, you can see the list of all **tasks** related to the currently selected object.

If you want to see the **tasks** related to another object, just select this object and press the **task list** button from the main *TitleBox* window.

To see the **tasks**, which are not related to any object (keyboard shortcuts), unselect all objects in *TitleBox* and press the **task list** button.

There is a specific definition, if you want to control the whole *TitleBox* project on incoming GPI signal.

To invoke the global **Assign GPI event** window, unselect all objects in *TitleBox* project, right mouse click and select Assign GPI event from menu.

Select the PC COM port and the type of the GPI signal which will be received (CTS, DSR, RING or RLSD).
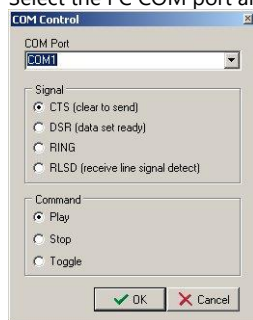
In the lower part of the screen, select the Command which *TitleBox* have to execute when the GPI signal is received. There are three possible commands: Play, Stop and Toggle.

In the *example* from screenshot above, when on port COM1 a CTS signal is received, *TitleBox* will start to play the current project

AS-RUN LOG

The As-Run Log is an external application, displaying log information for all **TitleBox** instances, which are running locally. It can be enabled/disabled from the dedicated button in the main **TitleBox** toolbar. Also, it is configured per instance in **Project** menu ➔ **Options** ⇨ **As-Run Log** tab.

User Interface

The interface of the **As-Run Log** is very simple and user-friendly.
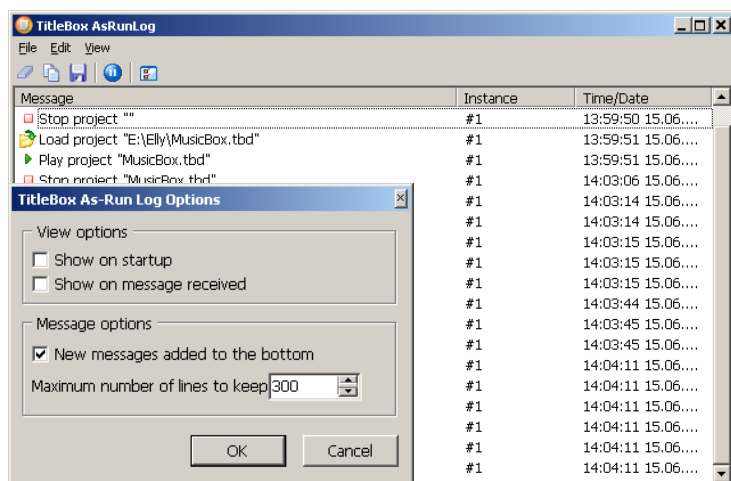
Grid

The largest area in the **As-Run Log** is dedicated to displaying messages about all the events that take place in the local **TitleBox** instances, which have enabled logging. Each **Message** contains information about the **Instance**, to which it refers, as well as the **Time/Date** of the event.

Toolbar

The toolbar allows you to perform the following simple commands:

🖊 **Clear window**– press this button to clear the log window

📋 **Copy selected lines**–press it to copy the currently marked lines from the log. Use the <Shift> and <Ctrl> keys to select lines

💾 **Save to file**–press this button to save the current log messages to file

⏸ **Pause messages**–press this button to temporary stop logging messages

🗗 **Options...**- press this button to invoke the Options dialog, which looks like the image below:

Use the check-boxes to configure the following:

☑ *Show on startup*–check this box if you want to view the **As-Run Log** when you initiate **TitleBox**

☑ *Show on message received*–check this box if you want the log to be displayed every time it receives a message from **TitleBox**

☑ *New messages added to the bottom* –check this box if you want the log messages to be displayed at the bottom of the field as opposed to the top

In the *Maximum number of lines to keep* specify the number of lines you wish to keep visible in the ***As-Run Log*** interface. You can enter a maximum of 1,000,000 lines. If you enter 0, then the log will display all messages without clearing any lines.

Menu Bar

The ***As-Run Log*** menu bar contains the following menus:

File Menu

File	
⏸ Pause Messages	Ctrl+P
🗏 Options…	Ctrl+O
⭕ Shutdown	Ctrl+Alt+X
🗗 Hide Window	Alt+F4
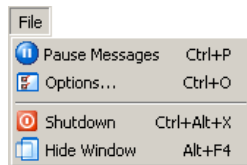
The first two commands are the same as the ones in the toolbar – the **Pause Messages** command is used for temporary stopping the log messages and the **Options** command invokes the options dialog, described above.

You can also exit from the ***As-Run Log*** via the **Shutdown** command or just minimize it with the **Hide Window** command.

Edit Menu

Edit	
🗐 Copy Selected Lines	Ctrl+C
✏ Clear Window	Ctrl+L
💾 Save to File…	Ctrl+S
Select All	Ctrl+A
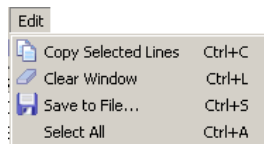
Commands here also repeat the commands from the toolbar. You can copy the lines selected in the log, clear the log window, or save the messages to a file. Also, you can select all lines via the **Select All** command.

View Menu

View	
✔ Show Toolbar	Ctrl+B
Stay on Top	Ctrl+T
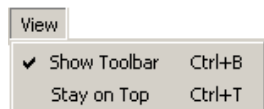
The **View** menu allows you to show and hide the toolbar from the user interface. Also, you can check the **Stay on Top** command so that the ***As-Run Log*** is always displayed on top of the other windows.

CLIP TRIMMER

The **Clip Trimmer** is used for previewing and trimming clips in **ListBox, DataBox** and **AirBox** modules. You can preview clips even during the on-air playback in **AirBox**.

Trimmer allows marking IN and Out points for video playout and then the player (**AirBox**) plays the video exactly between these points. This way no real cutting of the file is performed.

USER INTERFACE
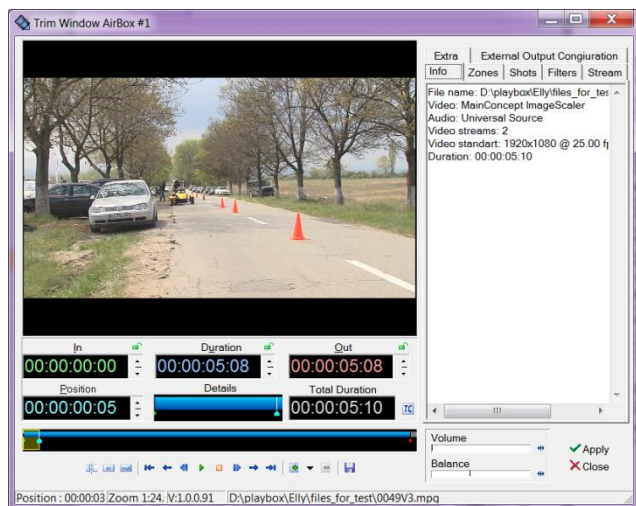
It combines a *VGA* preview screen and tools for editing the *In* and *Out* points of *MPEG* clips. .

Clips are loaded into the **Trim Window** by double-clicking on them in **ListBox, DataBox** or **AirBox** modules.

The trimmer window has a number of keyboard shortcuts that are listed in the shortcuts leaflet in the PlayBox customer's package.

Preview Screen

The Preview screen occupies the largest part of the Trim Window. It shows the video clip which is being trimmed.

Pressing the right mouse button over the **Preview screen** invokes a context menu with the most frequently used commands in the **Trimmer**.

Trimmer Control

Once a clip is loaded, you can navigate and edit it, using the **Clip Trimmer** controls.

The following boxes are situated under the preview window:

In box – displays the In point timecode

*Out*box –displays the Out point timecode

Duration box – displays the clip duration. It is equal to the difference between *In* and *Out* points.

After entering the In and Out timecodes, the Duration value will be calculated automatically. Moreover, after entering the In and Duration values, the Out point timecode will be calculated automatically. It is possible to lock the In, Out or Duration value (but not all the three of them, of course) by clicking on the lock icon to the right of the box. Thus, you can protect their values from changing (by mistake, for example).

To enter values in the In, Out or Duration box, you can use the arrows beside them or type numbers in the boxes.

Position box contains the timecode of the currently selected point in the clip. If you type another timecode in this box and press **<Enter>**, the marker will move to the relevant position.

Total Duration box – displays the original, untrimmed duration of the clip.

Details box – it shows the part of the clip, enclosed in the zoom frame.

The *Trim bar* visualizes the clip length and the clip markers.

The *Zoom frame* is a yellow square in the trim bar. It defines what part of the clip is being shown in the Detail View. It is quite useful for long clips. You can move or resize the zoom frame, by mouse dragging. You can define a new Zoom area by drawing a rectangle through dragging the mouse while holding its right button.

SPLIT button - splits the clip and thus defines separate sections in the clip. After splitting the clip, each new clip section appears as a separate row in the play list. You can play them separately, change their order, insert other clips between them, etc.

You can define *In/Out* points for each section of the split clip. The *In, Out* and *Duration* boxes display information about the selected section. Multiple In/Out definitions are used for skipping some parts of the clip during its playback. Thus, you can exclude existing commercials, titles, etc. from the original clip.

You can create separate sections also by pressing repeatedly the **In/Out** buttons. Pressing the **In** button after an existing *Out*point, creates a new section after the existing one. Pressing the **Out** button before an existing *In* point, creates a new section before the existing one.

IN button – marks the *In* point for start of playback in **AirBox**. The *IN* point marker is colored green and by default it is at the beginning of the clip.

OUT button – marks the *Out* point for end of playback in **AirBox**. The *OUT* point marker is colored red and by default it is at the end of the clip.

You can set an *In/Out* point, by dragging the *IN/OUT* marker on the trim bar or by entering timecode directly into the *IN/OUT* box.

The *IN/OUT* markers show the frame before/after (i.e. if the marker is on the *Out* point, you actually see the next frame, but not the last frame of the trimmed part.).

NOTE: For MPEG files, the trimming is *GOP* accurate! *IN* points can be positioned only on I-frames, while OUT points can be positioned on I- or P-frames. Depending on the stream's *GOP* size, this may lead to a slight inaccuracy, limited to half a *GOP* for the *IN* points and a couple of frames for the OUT points. If the stream's *GOP* size is one (I-frames only), then trimming will be frame accurate.

NOTE: Even if you specify In and Out points in a particular timecode, some playout plug-ins (for ex. IPPUMP) will correct them to nearest I-frame for IN point or I- or P-frames for OUT point. Thus outputted duration could be shorter then specified.

Play button starts playing the clip. Then it transforms to pause/resume.

If you want to play only the trimmed part (i.e. between the *IN* and *OUT* points), press **<Shift>**+ **Play**.

Stop button stops the playback and "rewinds" the clip to its beginning.

Pause /**Resume** button. It stops and resumes the clip playback from the same timecode position.

Button - skips one frame forward

Button ![icon] - reverts to one frame backward

Button ![icon] - fast forward (if it is possible)

Button ![icon] - rewind (if possible)

Button ![icon] - moves to the next mark point (in/out/bookmark)

Button ![icon] - moves to the previous mark point (in/out/bookmark)

Add Zone button ![icon] - marks the beginning of a zone. A yellow point appears in the trim bar.

Delete Zone button ![icon] - deletes the selected zone point.

Open button will load a clip in the Trimmer window.

Export button ![icon] - opens a dialog box for exporting the trimmed clip as a file.

If you have divided the clip into several sections, they will appear in the **Export list** as different parts. Select which of them to export by checking the checkboxes in front the relevant section. Besides, you can change their order using the up and down arrows.

If you choose to export several parts at a time, you have to specify the way they should be exported: as separate files *(⦿Export selection(s) as multiple files)* or as a single file *(⦿ Export selections as a single file)*.

Pressing the **Export** button in this dialog opens another dialog box for you to specify the file name and store location and to begin the export.

To delete a clip section, select it and press **<Ctrl>+<D>**.

To reset the original clip length, press **<Ctrl>+<BkSp>**.

The *Throttle* has three levels – Low, Normal and High. Each of them "tells" the CPU what priority to give to the export process.

The higher the priority, the slower the other simultaneously running processes on the machine, and the quicker the export made.

IMPORTANT! Export works only for MPEG2 files.

NOTE: Export is not frame accurate. It is *GOP* accurate! *IN* points can be positioned only on I-frames, while OUT points can be positioned on I- or P-frames. Depending on the stream's *GOP* size, this may lead to a slight inaccuracy, limited to half a *GOP* for the *IN* points and a couple of frames for the OUT points. If the stream's *GOP* size is one (I-frames only), then trimming will be frame accurate.

Volume Control

You can change the audio volume and audio balance of the clip, using the relevant sliders in the **Clip Trimmer**. The new values will not take effect in **AirBox** playback. These sliders are used for convenience during the trimming process.
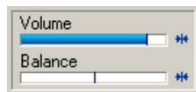
Volume box – shows the volume level of the clip playback. There is a reset button beside the slider to restore the default value.

Balance box – shows the clip playback L-R balance. There is a reset button beside the slider to restore the default value.

Status Bar

It is located along the lowest part of **Trimmer** window.
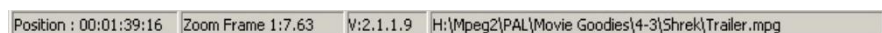
- The first field [Position] – shows the current timecode, when dragging over the trim bar.

- The second field [Zoom Frame] – shows the scale of the Zoom zone.

- The third field [Version] – displays the current **Trimmer** version.

- The last field shows the clip's filename and location.

TRIMMER PAGES

There are six pages on the right side of the **Trimmer** Window: **Info**, **Zones**, **Shots**, **Filters**, **Stream** and **Extra**.

Video Clip Info
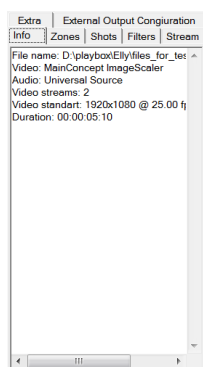
This page shows important system stream information, extracted from the file. It contains:

- Filename
- Video compression type
- Video stream information: number of streams; image width and height in pixels; video frame rate; video bitrate.
- Audio stream information: number of streams; audio compression type; audio sampling rate; audio channels; audio bitrate.
- Type of *MPEG* packages
- Type of *PES* (packetized elementary stream)
- Mux rate (the multiplexing rate in bps)
- Duration of the clip in HH:MM:SS.ms (hours: minutes: seconds. milliseconds)

Clip Zones
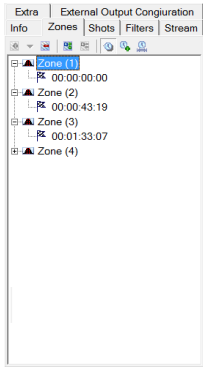
This page is used for registering a set of timecodes (points) in the clip. Thus, you can mark an initial or final timecode of an important zone in the clip. For example, in each movie there are predefined commercial break points that might be used later for automated commercial slot insertion. Those breakpoints can be defined in ***Clip Trimmer*** as single split points or zones (blank/stills) that will be skipped, and a commercial slot will be inserted instead.

To add a **Zone**, set the cursor at the appropriate point and press the ***Add Zone*** button in the page or in the main window. A yellow point appears in the trim bar, showing the beginning of the zone. You can define different zone types, pressing the black arrow of the ***Add Zone*** button. It opens a button menu:

   - **Bookmark** – creates a single split point at the current play head position. You can change the Zone name (Bookmark by default) and timecode manually.

   - **Simple skip zone** – creates a two-point zone. The first point is the current play head position; and the second is 5 seconds later, by default. You can change the Zone name and timecodes manually.

   - **Advanced skip zone** – creates a four-point zone. The first point is set at the current play-head position. By default the second point is 1 second later, the third point is 5 seconds later and the fourth point is 6 seconds later. You can change the Zone name (Advanced Zone by default) and timecodes manually.

To add a next marker in the **Zone**, press the ***Add Zone marker*** button. It will add a marker 10 seconds after the last marker in that zone, by default.

To delete a marker from the **Zone**, select it and press the ***Delete Zone marker*** button. It will delete all markers, except the first one, which can be deleted only by deleting the whole zone.

To delete a **Zone**, select it and press the ***Delete Zone*** button from the page or from the main window.

By pressing the Clock buttons, you can choose a display mode of the markers:

   - **Time** – shows exact timecodes in HH:MM:SS:FF format

   - **Offset value** – shows the interval between the current marker and the first one.

   - **Duration** – shows the time difference between two consecutive zone markers.

Clip Shots
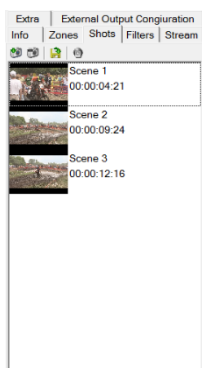
Use this page to extract shots from the video clip.

To create a shot, stop the play-head at the desired frame and press the **Add** button. The shot, as a thumbnail and its time-code are displayed in the tab. You can change the shot's name (Scene No., by default) manually.

To delete a shot, select it and press the **Delete** button.

To extract a shot to a file, select it and press the **Export** button. A browse window appears to select the file name, type (***.jpg** or ***.bmp**) and location.

To set a shot as thumbnail of a file, press the **Thumbnail** button.

Clip Streams

This feature is applicable in case the file contains more than one video and/or audio stream. Here you can choose which of them to view/listen during playback.

The check boxes ☑ in front the relevant streams show their playback status. The checked streams will be loaded in the playlist after clicking ✔ OK and will be played in their turn.

The green ticks ✔ ✔ serve for previewing the desired streams – if you click on the single tick, you will preview only the selected stream. If you click on the double tick, you will preview all the streams simultaneously. If you preview several audio streams, they will be mixed-up. If the streams are video, the preview window will split according to the number of streams (up to 32).

Use the blue arrows ↑ ↓ to move streams up and down the list.

Extra
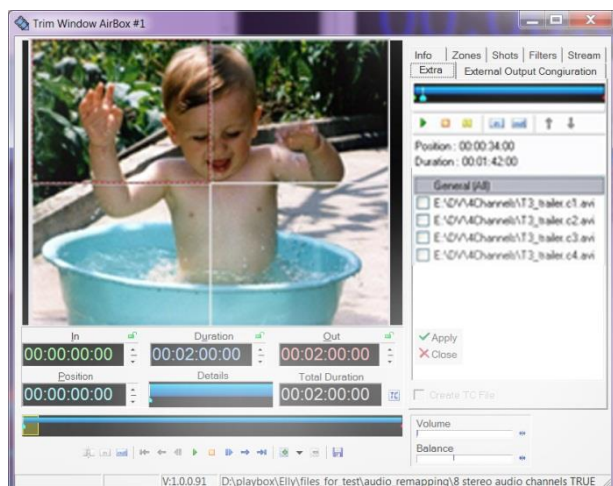
This "extra" will help you in managing multiple monitor video wall systems of up to 16 monitors. The files that are usually used for video walls end in *.c#.mpg or *.c#.avi (for example *.c1.mpg or *c3.avi). Just load the first file in the playlist. *AirBox* will detect the others automatically.

The trim bar and the buttons above the list of files are designed for control of the entire bunch. The trim bar and the buttons under the preview window are used for control of the currently selected monitor (the one in the red dotted-line frame).

This feature will help you in synchronizing the video wall and will allow you to control each monitor separately. You can also rearrange the display positions of files by drag-and-dropping the relevant squares in the preview window (i.e. file #1 can be displayed on monitor#3, etc.) or by using the blue arrows above the file list. The files in the preview window are arranged horizontally, in up to 4 rows of up to 4 files each. This means that if there are 16 files in the bunch (the most possible), the first row will contain files #1 to #4, the second one will contain files #5 to 8#, etc.

The ☑Create TC file check box is situated at the bottom of the file list. If you check this box, a TC file will be created automatically, saving you settings, such as *In* and *Out* points, file order, etc. Thus you will be able to use your settings again the next time you open the relevant file.

Clip Filters
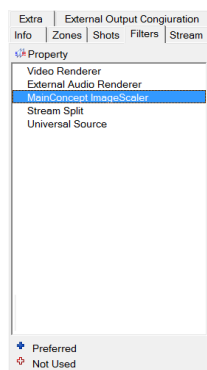
This tab provides information about the filters that participate in playing the current clip and an opportunity for changing the filters' settings.

The Windows Registry contains a key [*HKEY_CURRENT_USER\Software\Digital Media Technologies Ltd.\PlayBox\2.0\DXPresets*] in which you could create your own string values, containing groups of preferred filters (presets). All filters of the currently selected preset are displayed in *Filters* page.

There is a legend at the bottom of the page:

The ⊞ sign marks all filters from the preset.

The ⊞ sign marks filters that are included in the preset, but do not participate in executing the current particular file.

There is a *Property* button right under the tab name. Pushing it will open the property page (if any) of the selected filter. You could change the filter's setting there.

External Output Configuration
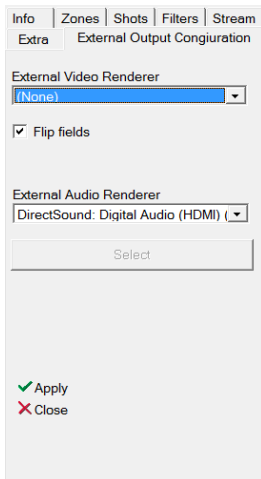
This tab allows you to select an *External Video Renderer*, as well as an *External Audio Renderer* from the corresponding drop-down lists. For the video renderer you are able to flip the fields of the output video by checking the ☑*Flip fields* box. In case the external audio renderer is multi-audio, then the first 8 audio channels are output in SDI. Once you are done selecting your external video and audio renderers, press the ***Select*** button.

PlayBox Live Inputs View

PlayBox Live Inputs View is an additional application which is installed together with *AirBox*.

GETTING STARTED

Installation

Where to install it

PlayBox Live Inputs View is usually installed under folder **C:\Program Files\PlayBox Technology Ltd\UDP Switcher**

The *PlayBox Live Inputs View* icon will appear in the **Toolbar** .

USER INTERFACE

The main purpose of the *PlayBox Live Inputs View* is to preview the video, which is defined as live event in *AirBox*.
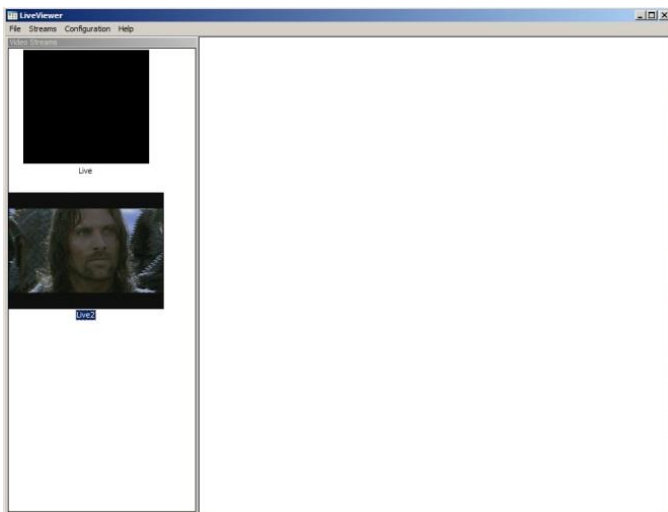
In order to see preview of the video source, first you have to set some video sources in *AirBox* from the *AirBox* menu ➔ **Settings** ⇨ *Live Inputs* tab.

IMPORTANT! You have to restart *AirBox* after setting up the video sources.

Now you are ready to open *PlayBox Live Inputs View*. Go to the **Streams** menu ➔**Add / Remove stream** and notice that a *Preview List* with already defined live video sources can be found there. It looks like the image to below:
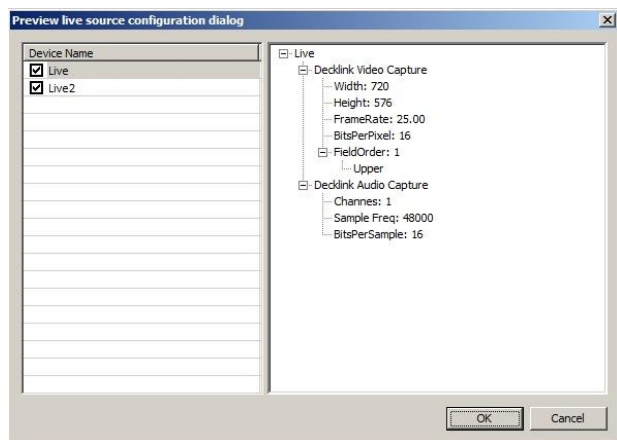
In the *Device Name* field check the respective source names of the sources you would like to preview in the *PlayBox Live Inputs View* and press **OK** button.

To remove a video source from a list, just un-check the device name from the list.

Into the left part of the main screen you will see small preview windows of all checked sources. By clicking on some of the small windows, you will see the preview of the respective source into the big screen to the right, as shown in the screenshot below.
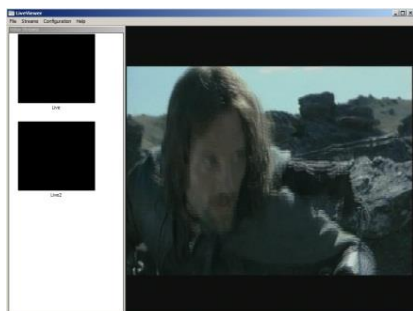
The position of the view-list of video streams could be changed by selecting the desired position from **Streams** menu➔**View orientation**. The **Video Renderer** context menu allows the user to select a video renderer to be used from the list of available presets. Here you can also enable / disable the audio of the configured stream via the **Audio Enabled** command.

Pressing **Configurations** menu ➔**Customize shortcuts** invokes the dialog, shown below:
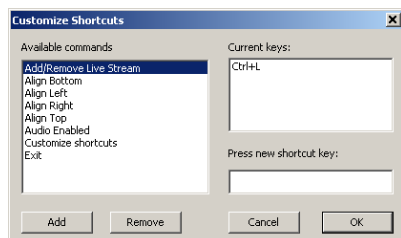
Here you can set a list of user's shortcuts for some main commands, like **Add/Remove streams**; **Alignment**, **Audio Enabling**, etc.

To the left, you will see the list of *Available commands*, and to the right the relevant shortcut will appear in the *Current keys* field. If you want to change the key combination for a certain command, select it from the *Available commands* list, go to the *Press new shortcut key* field, press the desired keyboard

combination, and notice that it appears in the respective field. When you are done with the configuration of your shortcut keys press the **Add** button. The new key combination will be assigned to the respective command. To exit the dialog, press **OK**.

APPENDIX 1 - PlayBox GPI

GPI IN AIRBOX

GPI (General Purpose Interface) is implemented in **AirBox, TitleBox and CaptureBox** as a set of triggers, associated to certain pins on the standard *PC RS-232* Serial Ports (*COM1, COM2*, etc.). In order to function, the *COM* port should be correctly installed in the Windows environment (See Device Manager → Ports).

GPI-IN can be used to slave **AirBox, TitleBox** and **CaptureBox** to triggers from external devices or simple contact switches, 4 triggers per COM port.

GPI-OUT can be used to slave external equipment to events taking place in **AirBox** or **TitleBox**, 2 triggers per COM port.

GPI PINOUT

Each *COM* port accommodates 4 **GPI-IN** pairs and 2 **GPI-OUT** pairs, but not at the same time. A particular *COM* port can be assigned as either **IN, OUT** or **Unused**.

GPI-IN pairs are located at output pins *DTR&RTS* and input pins *DSR, CTS, RI, CD*. You can use any of the output pins for supplying voltage to the GPI circuit.

GPI-OUT pairs are located at output pins *DTR&RTS*, the first GPI trigger generates a pulse on *DTR*, and the second *GPI* trigger generates a pulse on *RTS*.

GPI-IN IMPLEMENTATION

The simplest triggering device would be a pair of wires running from the *COM* port *GPI* pair, soldered to a momentary contact switch. This switch can be either **Push Button Normally Open** (PBNO) or **Push Button Normally Closed** (PBNC). The trigger type is configured in *AirBox GPI Settings Panel* as **High** or **Low** pulse. Many external devices like switchers or mixers have dedicated *GPI* connectors or screw terminals with a description of what *GPI* trigger type (low or high pulse) was implemented. Depending on the trigger setup, a trigger can be a temporary closing or opening of the *GPI* circuit. High pulse means the *GPI* trigger is activated when the circuit is temporary closed. Low pulse means the *GPI* trigger is activated when the circuit is temporary opened.

According to the number of installed *COM* ports, **AirBox** supports up to 32 **GPI-IN** triggers (up to 8 *COM* ports with 4 **GPI-IN** triggers per port).

GPI-OUT IMPLEMENTATION

In order to control external devices by *GPI*, a simple 12V contact relay should be used. It should be connected to the corresponding *COM* port pin pair (4-6 or 7-8) and it should close or open the *GPI* circuit of the external device. Depending on the **AirBox** setup, the *GPI* trigger will generate a pulse (low or high, user-defined) to the corresponding output pin for a short user-defined period (pulse duration).

According to the number of installed *COM* ports, **AirBox** supports up to 16 **GPI-OUT** triggers (up to 8 *COM* ports with 2 **GPI-OUT** triggers per port).

AIRBOX AS A GPI SLAVE

A wide variety of **AirBox** actions can be associated to a *GPI* trigger:

- o   Start playback
- o   Pause/Resume playback
- o   Stop playback
- o   Jump to next clip
- o   Jump to specific clip (predefined playlist index)

o   Jump to specific location in the clip/playlist (predefined timecode)

o   Playlist reset (during stop mode only)

o   Hardware reset (terminates the playback!)

o   Turn logo on

o   Turn logo off

o   Cue the selected clip

o   Cue to specific clip

o   Jump to bookmark in time range

o   Jump to bookmark name

AIRBOX AS A GPI MASTER

AirBox can activate a *GPI* trigger though specially designated GPI Output event. Please check the following page for GPI Output reference.

AIRBOX GPI SETTINGS PANEL

All **GPI-IN** triggers can be assigned to a specific *AirBox* action. From *AirBox Settings* ➔ *Modules⇨Remote Control*, select "GPI Input" **enabled**, to allow the **GPI-IN** trigger. Press the **Configure** button, to invoke a table for setting the *COM* ports, GPI groups and the available actions for them.

All **GPI-OUT** triggers can be activated by specific *AirBox* event: From *Settings* ➔ *Modules⇨Remote Control*, select "GPI Output" **enabled**, to allow the **GPI OUT** trigger. After that you can Add/Insert GPI Output event in the playlist, by right mouse clicking.

RS232 9-PIN D-SUB PINOUT REFERENCE

Pin	Name	Description	Direction
1	CD	Carrier Detect	In
2	RXD	Receive Data	In
3	TXD	Transmit Data	Out
4	DTR	Data Terminal Ready	Out
5	GND	System Ground	-
6	DSR	Data Set Ready	In
7	RTS	Request to Send	Out
8	CTS	Clear to Send	In
9	RI	Ring Indicator	In

GPI INPUT REFERENCE

(Triggers that control *AirBox, TitleBox*, and *CaptureBox*)

GPI Input	Name	Contact Pins
GPI 1	CST	8 + 4
GPI 2	DSR	6 + 4
GPI 3	RI	9 + 4
GPI 4	CD	1 + 4

GPI OUTPUT REFERENCE

(Pulses sent out from *AirBox* and *TitleBox*)

GPI Output	Name	Contact Pins
GPI 1	DTR	4 + 5
GPI 2	RTS	7 + 5

APPENDIX 2 – Graphic Rules' Commands, Used for Communication between AirBox and TitleBox

In the table below you can find a list of commands, used in the AirBox Graphic Rules option.

> **IMPORTANT:** All of the commands used in Graphic Rules are CASE SENSITIVE!

LOAD_TEMPLATE=TB_ProjectName.tmpl	Points out which template (created from a TitleBox project) is addressed.
ObjectName.play	Sends a play command for the corresponding TitleBox object, designated by its name.
ObjectName.stop	Sends a stop command for the corresponding TitleBox object, designated by its name.
ObjectName.text	Used for adding text to the corresponding TitleBox object, designated by its name.
ObjectName.media	Used for adding media to the corresponding TitleBox object, designated by its name.
PlaylistColumn.text=%clip_PlaylistColumn%	This line contains the name of the AirBox playlist column [PlaylistColumn] object [Title] and a description of the text that it should contain.
	For example, the line [Title.text=%clip_title%] contains the name of the TitleBox object [Title] and a description of the text that it should contain. I.e., in this case, we will extract information from the Title column of the relevant playlist entry.
%clip_title%	Use this line to show the text from the corresponding title field in the AirBox playlist.
%clip_location%	Use this line to show the text from the corresponding location field in the AirBox playlist.
%clip_category%	Use this line to show the text from the corresponding categofy field in the AirBox playlist.
%clip_notes%	Use this line to show the text from the corresponding notes field in the AirBox playlist.
%clip_star%	Use this line to show the text from the corresponding star field in the AirBox playlist.
%clip_start%	Use this line to show the text from the corresponding start time field in the AirBox playlist.
%clip_duration%	Use this line to show the text from the corresponding duration field in the AirBox playlist.
%clip_start_date%	Use this line to show the text from the corresponding start date field in the AirBox playlist.
%metadata_MetadataName%	Use this line to show the text from the corresponding metadata, designated by its name.

%clip_title[+n]%	Use this line to display information about the title of an upcoming clip in the playlist, where [+n] is the off-setter.
%clip_title[BM_Test1]%	Use this line if you want to use a bookmark for presenting information about an item in the playlist. Here Test1 is the Bookmark name and [+n] is the off-setter.
%clip_start[+n]{HHMMSSFF}%	Shows the start time and the title of an up-coming clip. Here Next is the name of the text object in TitleBox, [+n] is the off-setter, and {HHMMSSFF} is the time format to be used. Be aware that the time format can also be {HH} or {HHMMSS}.
	In case you are in NTSC mode, the start time is displayed in accordance to the PC system time. Thus, instead of {HHMMSSFF}, here you should use {T} or {TT}, where:
	{T} stands for short system time
	{TT} stands for long system time
%clip_title[CAT_CategoryName+n]%	Use this line to display the title of the next clip (if n=0) down the playlist that belongs to the category, designated by its name. You can also use [+n] as an off-setter.
%clip_start[CAT_CategoryName+n]%	Use this line to display the start time of the next clip (if n=0) down the playlist that belongs to the category, designated by its name. You can also use [+n] as an off-setter.
{!LOGO_ON!} / {!LOGO_OFF!}	Use this command to show / hide the logo from the screen.
SHOW_LOGO_PRESET_n	Use this command to display logo preset number [n], where [n] is between 1 and 16 and stands for the respective logo Preset, as configured in Settings ➔ Logo.
{!AUTOMATION_ON!} / {!AUTOMATION_OFF!}	Enables / disables the incoming GPI / DTMF triggers.
{!MUTE_ON!} / {!MUTE_OFF!}	Switches the audio on / off.

APPENDIX 3 – Integration of AirBox with TitleBox

TitleBox settings:

1) Run **Programs > PlayBox Technology Ltd. > TitleBox > PLNetInst.exe** and select a folder for your *TitleBox* templates.

2) Create your *TitleBox* projects.

3) Export them via **Network > Export project as template**, giving them respective names.

4) *.tmpl (template) files are exported in the templates folder.

5) Go to **Project\Options**, look at *Network* tab. Remember the TitleBox channel ID and Port values. Confirm any changes.

6) Go to **Network > Net control**. Run it.

That's all for *TitleBox*. Now it stays in standby mode and executes the commands coming from *AirBox*. Please note that it is not possible to edit objects in *TitleBox* while in this mode.

AirBox settings:

1) Go to **Settings ➜ Modules**, then to *Remote control* tab. Enable TitleBoxNetCtrl Output setting with **Yes**. Click Configure.

Enter same Channel ID and Port values as in *TitleBox* settings. Confirm.

 2) Go to **Events ➜ Add/Insert event ➜** TitleBoxNetCtrl Output

3) There are two modes for event insertion - 'Wizard' or 'Advanced' (selectable through the ***Advanced*** button). You are recommended to choose the 'Wizard' mode for now. The functions are self-explanatory, but since this module is still under development, some of them are not functioning as desired...

4) In 'Wizard' mode you can choose between Template Control and Play Project - the first one is for global *TitleBox* control commands; the second one is for project/objects control commands.

5) If you have entered Play Project mode, further you can select your project by list - all exported template projects should appear in this list. Select a whole project or some objects from it.

6) Click ***Finish***. That's it!

Run *AirBox*. When the time for a *TitleBox* event approaches, respective commands are being sent to *TitleBox* and it runs the appropriate objects.

APPENDIX 4 – PlayBox GPI board and Bypass Relay board
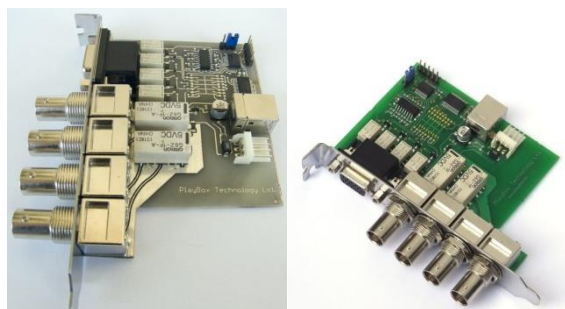
The PlayBox GPI Relay board has **4 GPI Inputs, 4 GPI Outputs with Relays, 1 Bypass relay (up to 2.6GHz)**

The following table is showing the pinout of the 15pin connector

Pin No	Signal Name	Remarks
1	NC	Not Connected
2	GPI OUT 3	Contact pin 2-7
3	GPI OUT 2	Contact pin 3-8
4	GPI OUT 1	Contact pin 4-9
5	GPI OUT 0	Contact pin 5-10
6	GND	Common Ground
7	GPI OUT 3	Contact pin 2-7
8	GPI OUT 2	Contact pin 3-8
9	GPI OUT 1	Contact pin 4-9
10	GPI OUT 0	Contact pin 5-10
11	+5V	5V DC From the board
12	GPI IN 0	Active - Short to +5V
13	GPI IN 1	Active - Short to +5V
14	GPI IN 2	Active - Short to +5V
15	GPI IN 3	Active - Short to +5V
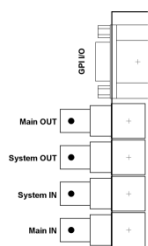
Main IN is the signal which is coming from outside the system

Main OUT is the signal which is going outside the system

System IN is the Bypassed system input

System OUT is the Bypassed system Output

All GPI inputs are using optocouplers.

All GPI outputs are relays. Both side of the contact are available on the 15pin connector.

Connected BNCs in bypass mode   Connected BNCs in active mode
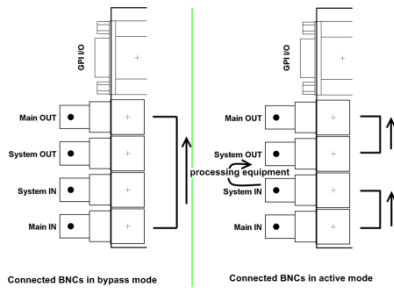
Technical Specification

Board size: 100x90mm

Bypass Relay:

-Tested Resolutions: PAL, NTSC, 720p50/60, 1080i50/60/59.94

-Contact switch time: 10ms

GPI Output:

-Maximum switching current: 0.7A

-Maximum switching Voltage: 220V AC

-Maximum switching capacity: 40W

-Minimum switching voltage: 250uV

-Resistive load: min 100 000 operations

-Contact switch time: 10ms

GPI Input:

-Maximum voltage to the input: 12V

Board Power:

-Using Floppy type connector from the machine

-Board Maximum Current: without using the +5V VCC on the connector 300mA

Board control: Thru USB (using internal connector is highly recommended)

APPENDIX 5 – AirBox and ListBox Examples

Example 1.   Volume Normalization:
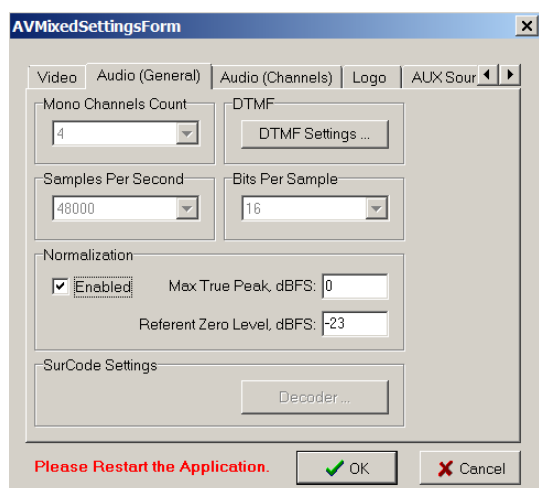
Suppose that you have enabled volume normalization and you have set the Referent Zero Level to -23 dbFS. Suppose further, that you have no metadata for the input file, and that the Volume Level at the output is above the referent level. Then, when you play the file, there will be no normalization, as there is no metadata file.

If you set metadata for your input file, the Volume at the output will be kept to the Referent Level, and it will be the same during the entire playout of the file.

Example 2.   No Language Descriptions:

Suppose that you have PCM to PCM audio with 8PCM Stereo audio stream at the output with languages, set to Undetermined in the Language field. However, suppose further, that the number of input channels is 16, so it is equal to the number of output channels, and that there is no metadata defined for the input audio stream file. Thus, your Mixed Audio Channel Settings Form will look like this:
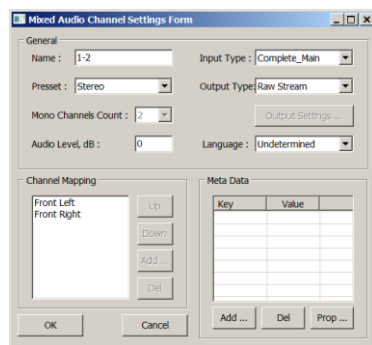
When you play the input file, the input audio streams will be sequentially mapped to the output streams.

Example 3. Remapping Strategy with Defined Languages:

Suppose that you have PCM to PCM audio with 7 PCM Stereo audio streams at the output and an input file with 16 channels, so the number of output channels is less than the number of input channels. However, you have defined languages for your output and an input file with language metadata, like this:

Output	Input
Channel 1-2 French	Stream 1 (1-2) German
Channel 3-4 Bulgarian	Stream 2 (3-4) No language
Channel 5-6 German	Stream 3 (5-6) Bulgarian
Channel 11-12 Bulgarian	Stream 4 (7-8) French
Channel 13-14 Spanish	Stream 5 (9-10) Russian
	Stream 6 (11-12) No language
	Stream 7 (13-14) No language
	Stream 8 (15-16) No language
Result after input file is played	
French goes to French → Stream 4 to 1-2	Stream 2 to 13-14
Bulgarian to the 2 Bulgarian streams → Stream 3 to 3-4 AND to 11-12	Undetermined output gets the available non-mapped input with language → Stream 5 to 7-8
German to German → Stream 1 to 5-6	All other unassigned streams will be mapped in sequential order: → Stream 6 to 9-10
Spanish has no matching input stream, so it gets the 1st in order without a defined language	→ Stream 7 to 15-16 → Stream 8 is discarded

Example 4. Remapping Strategy with no Language Description at the Input:

Suppose that you have have PCM to PCM audio with 7 PCM Stereo audio streams at the output and an input file with 16 channels, so the number of output channels is less than the number of input channels. Suppose further that the output languages are defined in the same way as in *Example 3* above, but the input file has no languages defined. As a result, when you play the input file, the input streams will be mapped to the output streams in sequential order, regardless of the language definitions for the output channels.

Example 5. Remapping Strategy with no Language Description at the Output:

Suppose that you have have PCM to PCM audio with 7 PCM Stereo audio streams at the output and an input file with 16 channels, so the number of output channels is less than the number of input channels. Suppose further that the output languages are undetermined and the input file has the same set of languages, as the one in *Example 3* above. As a result, when you play the input file, the input streams will be mapped to the output streams in sequential order, regardless of the language definitions of the input channels.

Example 6. No Language Descriptions and Disabled Mapping:

Suppose that you have no defined languages for both the output and the input audio, and your mapping is disabled. Suppose further, that the number of input audio streams is 3, and the number of output audio streams is 8, all of which are Stereo. Thus, the input channels are less than the output channels. The result after you play the input file will be that your input audio streams will go to the first three output audio streams in sequential order and the other five audio output streams will play silence.

Example 7. Input Language Descriptions and Disabled Mapping:

Suppose that you have defined 8 Stereo output streams with Undetermined languages and 3 input streams, of which only the third has a defined language, which is English. Suppose further, that your mapping is disabled. Since the input stream with English assigned has a higher priority than the streams with no language specified, when you play the input file, the 3rd stream of the input file will go to the 1st stream of the output. The other two streams of the input will go to streams 2 and 3 of the output in sequential order. The remaining 5 audio output streams will play silence.

Example 8. Output Language Descriptions and Disabled Mapping:

Suppose that you have an output with 8 Stereo streams, the 4th of which has English assigned, and you have an input file with 3 streams, all of which have Undetermined language metadata. Suppose further, that your mapping is disabled. Since the output stream with language assigned has a higher priority than the streams with no language specified, when you play the input file, the 1st input stream will go to the 4th output stream, and the other two input streams will be mapped sequentially to the 1st and 2nd output streams. The remaining 5 output streams will play silence.

Example 9. No Language Descriptions and Enabled Mapping:

Suppose that you have 8 Sterei output streams and 3 input streams, all of which have Undetermined language metadata. Suppose further, that your mapping is enabled. When you play the file, the input audio streams will go sequentiallyl to the first three output audio streams in the order they are defined. Furthermore, the audio of the 1st output stream (in English) will also go to the remaining 5 output audio streams, for which there is no input audio.

Example 10. Input Language Descriptions and Enabled Mapping:

Suppose that you have an input file with 3 streams, the third of which is set to English, and the other two are undetermined. Suppose further, that your output has 8 Stereo streams, and that your mapping is enabled. Since the input stream with language assigned has a higher priority than the streams with no language specified, when you play the file, the 3rd input stream will go to the 1st output stream, and the other two input streams will go to the 2nd and 3rd output streams in sequential order. Furthermore, the audio of the 1st output stream (in English) will be copied to the remaining 5 output audio streams, for which there is no input audio.

Example 11. Output Language Descriptions and Enabled Mapping:

Suppose that you have an input file with 3 streams, all of which have Undetermined language metadata. Suppose further, that you have defined 8 Stereo output streams , the 4th of which has English assigned, and that your mapping is enabled. Thus, when you play the file, 1st input stream will go to the 4th output stream, since it has a defined language and has a higher priority than the other streams. The other two input streams will go to the 1st and 2nd output streams in sequential order, and the audio of the 1st output stream (2nd input stream) will go to the remaining 5 output audio streams, for which there is no input audio.

Example 12. Down Convert 5.1 to 2.0 with no Language Descriptions:

Suppose that your input file has the following audio streams:

Stream 1 - 5.1

Stream 2 - 2.0

Stream 3 - 5.1

Suppose further that none of the above streams have languages assigned, and that there are 8 PCM Stereo streams, defined at the output, which also have no languages assigned. When you play the file, the 2nd input stream will go to the 1st output stream with no language, and the 1st and the 3rd input streams will go correspondingly to the 2nd and 3rd output streams. The rest of the output streams will either play silence, or will duplicate the 1st output channel, depending on your registry configurations.

NOTE: If you want to make a special registry configuration, related to Example 11, please, contact our support team at support@playboxtechnology.com.

Example 13.   Down Convert 5.1 to 2.0 with Language Descriptions:

Suppose that you have the same input file, as the one in Example 11 above but it has English assigned for Stream 3. Suppose further, that the output has 8 PCM Stereo streams and English is assigned to the 3rd and the 4th. When you play the file, the 3rd input Stream (in English) will be down converted to Stereo 2.0, and will go to the 2nd output. Stream 2 from the input will go to the 1st output stream, and the 1st input stream will be down converted to Stereo 2.0, and will go to the 3rd output stream.

Example 14.   Down Convert 5.1 to 2.0 with a Different Count of Language Descriptions:

Suppose that you have the following input:

Stream 1 - 5.1 → no language

Stream 2 - 2.0 → English

Stream 3 - 5.1 → English

Suppose further that your output has 8 PCM Stereo 2.0 streams with English assigned to the 2nd output stream. Since input Stream 2 matches output stream 2 both by language and by type, it will go to it. Input Stream 3, which is in English, has a higher priority than the input stream with no language. Thus, it will be down converted to Stereo 2.0 and will go to the 1st output stream without a language assigned. Finally, Stream 1 from the input will also be down converted to Stereo 2.0, and it will go to the 3rd output stream.

Example 15.   Up Convert 2.0 to 5.1 with no Language Descriptions:

Suppose that you have the following output and input with no languages assigned:

Output	Input
Stream 1 (Channels 1 – 6) – 5.1	Stream 1 - 2.0
Stream 2 (Channels 7 – 12) – 5.1	Stream 2 - 5.1
Stream 3 (Channels 13 – 14) – 2.0	Stream 3 - 2.0
Stream 4 (Channels 15 – 16) – 2.0	

When you play the file, you will have the following result:

Input Stream 2 will go to the 1st output stream (Channels 1 – 6)

Input Stream 1 will go to the 3rd output stream (Channels 13-14)

Input Stream 3 will go to the 4th output stream (Channels 15-16)

The remaining 2nd output stream (Channels 7 – 12) will either play silence, or it will duplicate the 1st output stream (Channels 1 – 6)

> NOTE: If you want to add a special configuration, related to the example above, please contact our support team at support@playboxtechnology.com.

Example 16. Up Convert 2.0 to 5.1 with Language Descriptions:

Suppose that you have the following output and input:

Output	Input
Stream 1 (Channels 1 – 6) – 5.1 → no language	Stream 1 - 2.0 → no language
Stream 2 (Channels 7 – 12) – 5.1 → English	Stream 2 - 5.1 → no language
Stream 3 (Channels 13 – 14) – 2.0 → no language	Stream 3 - 2.0 → English
Stream 4 (Channels 15 – 16) – 2.0 → no language	

When you play the file, you will have the following result:

Input Stream 3 will be up converted to Stereo 5.1 and will go to Output Stream 2 (Channels 7 – 12)

Input Stream 1 will go to Output Stream 3 (Channels 13 – 14)

Input Stream 3 will go to Output Stream 4 (channels 15 – 16)

The remaining Output Stream 1 (Channels 1 – 6) will play silence.
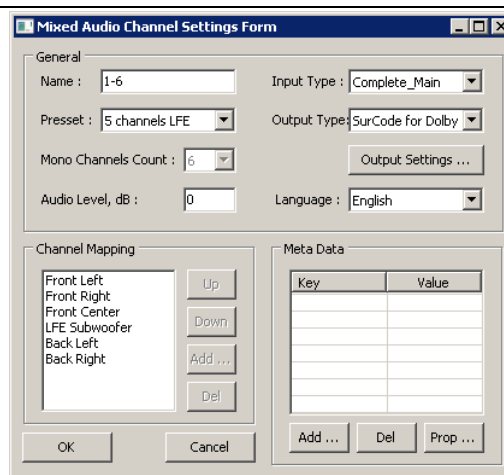
Example 17. AC3 Audio with no Language Descriptions:

Suppose that your input file is in AC3 audio, and that you have the following output audio settings:

Stream 1 (Channels 1-6) – Default

Stream 2 (Channels 7-12) – preset 5 channels LFE and output type SurCode for Dolby

Stream 3 (Channels 13-14) – PCM Stereo

Stream 4 (Channels 15-16) – PCM Stereo

Suppose further, that you have no languages assigned for the output, nor for the input streams.

When you play the file, you will have the following result:

Input Stream 1 will go to Output Stream 1 (Channels 1-6)

Input Stream 2 will go to Output Stream 2 (Channels 7-12)

Input Stream 3 will be converted to PCM Stereo and it will go to Output Stream 3 (Channels 13-14)

Input Stream 4 will be converted to PCM Stereo and it will go to Output Stream 4 (Channels 15-16)

Example 18.   AC3 Audio with Language Descriptions:

Suppose that your input file is in AC3 audio and it has English assigned for its 2nd Stream, as well as French for its 1st Stream. Suppose further, that you have the following output audio settings:

Stream 1 (Channels 1-6) – Default

Stream 2 (Channels 7-12) – preset 5 channels LFE, output type SurCode for Dolby, and English assigned

Stream 3 (Channels 13-14) – PCM Stereo and French assigned

Stream 4 (Channels 15-16) – PCM Stereo

When you play the file, you will have the following result:

Input Stream 1 will be converted to PCM Stereo, and it will go to Ouptut Stream 3 (Channels 13-14)

Input Stream 2 will go to Output Stream 2 (Channels 7-12)

Input Stream 3 will go to Output Stream 1 (Channels 1-6)

Input Stream 4 will be converted to PCM Stereo, and it will go to Output Stream 4 (Channels 15-16)

Example 19.   PCM Input to AC3 Output:

Suppose that you have an input file with 8 PCM Stereo streams, and that you have an AC3 output. Suppose further, that none of the input streams, nor the output streams, have languages assigned.

When you play the file, you will have the following result:

Input Stream 1 will go to Output Stream 3 (Channels 13-14)

Input Stream 2 will go to Output Stream 4 (Channels 15-16)

Input Stream 3 will be up converted to 5.1, and it will go to Output Stream 1 (Channels 1-6)

Input Stream 4 will be up converted to 5.1 and it will go to Stream 2 (Channels 7-12)

Example 20.   Creating a Regular Playlist:

Open *AirBox* and go to **File ➔ New Playlist.** Notice that *ListBox* is opened. In ListBox go to Settings – Modules – AirBox Output and double-click on the output plugin that matches the one, set for AirBox. Save your settings and notice that the ListBox output is configured according to the AirBox output plugin setup.

The next step is to add items to the playlist by drag-n-dropping. Select a media folder, containing the files you want to play and place them in the playlist area of the ListBox main window. Save your playlist from File – Save with your desired name. Close ListBox.

Example 21.   Creating a Daily Playlist:

Press the Create New Playlist ⬚ button in AirBox. Notice that ListBox is opened. Add your desired media files via drag-n-dropping them from the respective folder to the playlist grid. Go to File – Save daily playlist. The Daily Playlist dialog will open. Set your start time and select the folder, where the daily playlist will be saved. Notice that the name of the playlist is created automatically. Press Save and verify that the specified folder contains a file with the name displayed in the Save Daily Playlist form.

Example 22.   Load a Playlist Manually:

Open AirBox and go to File – Load Playlist. Select the playlist, already created in Example 20 above and press Open. Verify that the properties of the items in the playlist, i.e., duration, type, and metadata are correct. Select the Playlist tab from the multi-purpose zone and double-click on a playlist from the list.

Start a new playlist by pressing the Create New Playlist ⬚ button in AirBox or via the <Ctrl + N> key combination. Notice that ListBox is opened. There go to Settings – Modules – AirBox Output and double-click on NONE. Save your settings.

Add your desired media files via drag-n-dropping them in the playlist grid, save your playlist and close ListBox. Notice that all the files in the playlist have a duration of [00:00:00:00], the items are marked in red and the Type column contains the text [Unsupported media].

Go to AirBox and click on the Load Playlist from File ⬚ button. Open the playlist that you have just saved in ListBox. The newly loaded files will have a duration of [00:00:00:00]. Go to Settings – General – check Auto update duration of marked files. Set the time for auto update to 1 minute and save your settings. Select some of the items in the playlist, right-click and select Auto update. The auto update symbol will appear in the first column of the selected items. In about a minute the duration of the selected files for auto update will change and become greater than [00:00:00:00].

Example 23. Automatically Load a New Playlist on StartUp:

Open AirBox and go to Settings – StartUp. Verify that the New radio button is selected and save your settings. Restart AirBox and notice that when you restart AirBox an empty playlist will be displayed.

Example 24. Automatically Load the Last Used Playlist:

Open AirBox and load a playlist. Go to Settings – Startup, select the Last used radio button and place a check in the Auto start playback after playlist loading box. Save your settings and restart AirBox. Notice that the last used playlist is loaded and the playout starts immediately.

Example 25. Trigger the Open Playlist Dialog at Startup:

Open AirBox and go to Settings – Startup. Select the Open dialog radio button and save your settings. Restart AirBox and notice that the open playlist dialog appears automatically upon startup. Select your desired playlist and press Open.

Example 26. Automatically Load an Existing Daily Playlist on StartUp:

Open AirBox and create several daily playlists (as shown in Example 21 above). Create one playlist for the current day, which starts 5 minutes after the current time, one for the next day, and one for the previous day. Save your playlists in the default folder, suggested by AirBox / ListBox. Go to AirBox Settings – StartUp and select the Daily playlist folder radio button. Verify that the Reload playlist immediately option is selected. Save your settings and restart AirBox. Notice that the daily playlist for the current day is loaded and the playout starts when the start time occurs.

Example 27. Automatically Load an Existing Daily Playlist on StartUp:

Open AirBox and create two different daily playlists for the current day. Set the start time of the first one to be 5 minutes after the current time and that for the second one to be 10 minutes after current time. Go to AirBox Settings – Startup and in the field next to the Daily playlist folder option fill in the folder path, where your daily playlist are saved. Save your settings and restart AirBox. Notice that the daily playlist with start time closer to the current time (i.e., the one set for 5 minutes after the current time) is loaded and the playout starts when the start time occurs.

Example 28. No Current Daily Playlist is Available:

Open AirBox and move all daily playlists for the day to a different folder. Then load a 24-hour playlist and save it as a daily playlist with a date, corresponding to 1 week before the current date and a start time of [00:00:01:00]. In AirBox Settings – Startup dialog verify that none of the boxes from the If there is no current daily playlists field is checked. Restart AirBox and notice that no playlist is loaded.

Revisit AirBox Settings – Startup dialog and check the Try to load last daily playlist for same weekday box and save your settings. Restart AirBox and notice that the playlist, saved for last week is loaded.

Another way to load the latter daily playlist automatically is to check the Try to load past daily playlist from last box and enter 10 days in the spin-box in AirBox Settings – Startup dialog. However, if you enter less than 7 days in this spin-box, the daily playlist that is 1 week old will not be played automatically.

Example 29. Reload a Regular Playlist on File Change in AirBox:

Let us assume that you are using two AirBox instances, **AB#1** and **AB#2** and that the second instance is set to **Soft Mixed output plugin**.

Assume further that **AB #1** has the following configurations in Settings – Startup dialog:

- The New radio button is selected in the StartUp playlist field.
- The Reload playlist on file change box is checked and the Reload daily playlist without playing box it is NOT checked in the Reload playlist options field.
- The Reload playlist immediately radio button is selected in the When reload playlist field.

Imagine that we load the same **regular** playlist in both AirBox instances and start their playout. If we **delete** the currently played clip in **AB#2** and save the playlist, in **AB#1** the playlist will be reloaded without the deleted clip and the playout will start from the beginning of the playlist.

Furthermore, imagine that we insert a few media files after the currently played clip in **AB#2** and save the modified playlist. As a result, the playlist in **AB#1** will be reloaded, and the playout will continue with the next clip.

Example 30.   Reload a Regular Playlist on File Change in ListBox:

Let us assume that you are using AirBox together with ListBox. Assume further that you start ListBox, load the playlist currently played in AirBox, delete a few items before the currently played clip, and save the playlist. As a result, the playlist will be reloaded in AirBox, and the playout will continue with the next clip.

Now imagine that the Wait for playing clip end radio button is selected following in AirBox Settings – Startup and that you reset the playlist with the Reset button and press Play in AirBox. Furthermore, assume that you modify the playlist in ListBox and save it while AirBox is playing. As a result, the playlist in AirBox will be reloaded after the end of the currently playing clip and the playout will continue from the next clip.

Finally, suppose that you delete the currently playing clip in ListBox and save the playlist. As a result, the playlist in AirBox will be reloaded after the end of the currently playing clip, and the playout will start from the beginning of the playlist.

Example 31.   Reload a Daily Playlist Immediately on File Change in ListBox:

Open AirBox and create a daily playlist with start time a few minutes later than the current time. Go to Settings – Startup and select the Daily playlist folder radio button. Verify that the Reload on file change option is checked and select the Reload playlist immediately radio button. Set the Pre-cache new playlist to 30 minutes before start, save your settings, and restart AirBox. The playout of the daily playlist should start when its start time occurs. When this happens, open ListBox, insert a few media files after the currently played clip, and delete a few others with start time after the currently played clip. Save your playlist and notice that the daily playlist in AirBox is reloaded and the playout is resumed according to the current time.

Example 32.   Reload a Daily Playlist on File Change in ListBox While a New Playlist is Pre-cached:

Open AirBox and create a daily playlist with start time a few minutes later than the current time. Go to Settings – Startup and select the Daily playlist folder radio button. Verify that the Reload on file change option is checked and select the Reload playlist immediately radio button. Set the Pre-cache new playlist to 30 minutes before start, save your settings, and restart AirBox. Open the log view and monitor it until a message like this appears: [17:10:22:000 - Precaching playlist: D:\playlists*.ply 17:10:22:129 - Playlist D:\playlists*.ply is cached and ready for cue]. Next, open ListBox, modify the currently playing daily playlist and save it. Notice that the changed playlist is NOT reloaded and a warning message is displayed.

Example 33.   Reload a Daily Playlist on File Change in ListBox Without Playing it:

Open AirBox and go to Settings – Startup. Check the Reload daily playlist without playing it box, save your settings and restart AirBox. Next, open ListBox, modify the playlist and save it. Notice that in AirBox the playlist is reloaded and the playout is stopped.

Example 34.   Reload a Daily Playlist on File Change in ListBox Using the Wait for Playing Clip End Option:

Open AirBox and create a daily playlist with start time equal to a few minutes after the current time. While the playout is stopped, go to Settings – Startup and select the Wait for playing clip end radio button. Save your settings and restart AirBox. Next, open ListBox, modify and save the same playlist by deleting and / or adding items. Notice that in AirBox the playlist is reloaded after the end of the currently played clip and the playout is resumed according to the current time.

Example 35.   Overlapping Between Daily Playlists:

Suppose that you have two daily playlists in AirBox, set for the current day. Furthermore, suppose that in AirBox Settings – Startup dialog the Daily playlist folder radio button, as well as the Reload playlist immediately radio buttons are selected.

Now let us assume that an overlapping exists between the current and the new playlist. In other words, the first playlist is with short duration and a start time a few minutes after the current time, and the second playlist is with start time that overlaps the first one. Start playing the first daily playlist and notice how when the start time of the second playlist occurs the currently played clip is trimmed and the second playlist is immediately loaded and played.

Example 36.   Gap Between Daily Playlists:

Suppose that you have two daily playlists in AirBox, set for the current day, of which the first one starts a few minutes after the current time, and the second one starts a few minutes after the end of the first one. Furthermore, suppose that in AirBox Settings – Startup dialog the Daily playlist folder radio button, as well as the Reload playlist immediately radio buttons are selected.

Now let us assume that there exists a gap between the current and the new daily playlist and the Fill gap between playlists box is checked in AirBox Settings – Startup dialog. Start the playout in AirBox and notice that after the end of the first playlist the corresponding auto-fill clip is played, and the playout of the second playlists begins when its start time occurs.

Note that if the Fill gap between playlists box is not checked and you perform the playout, described above, when the first playlist ends, the AirBox playout will be stopped and the new playlist will be in Cue mode.

Example 37.   Wait for Playing Clip End Option:

Open AirBox and go to Settings – Startup. Select the Wait for playing clip end radio button and save your settings. Suppose that you have two overlapping daily playlists, as in Example 35 above. Open AirBox and go to Settings – Startup. As a result, when you start the playout and the time for starting the second playlist occurs, the currently playing clip from the first daily playlist is played until the end ant all following clips are removed.

Suppose further that you have the above setting but you have defined a Daily Playlist Tolerance in AirBox Settings – Startup. Thus, when the start time of the second playlist occurs two things might happen:

If the start time of the second playlist is closer to the current clip end time, i.e., the time difference is shorter or equal to the tolerance, defined in the Startup Settings, the start time of the second playlist will be updated by adding the time difference.

If, however, the start time of the second playlist is closer to the current clip start time, i.e., the time difference is lower or equal to the tolerance, defined in the Startup Settings, the start time of the next clip will be updated by subtracting the time difference and the overlapping clip will be removed.

Furthermore, if the above tolerance subtractions are not enough to compensate for the time difference, the clip trimming will be taken into account.

Example 38.   Wait for Playing Clip End Option with Allow Clip Trimming:

Open AirBox and go to Settings – Startup. Select the Wait for playing clip end radio button, check the Allow Clip Trimming box and save your settings. Suppose that you have two overlapping daily playlists, as in Example 35 above and that you have defined a Daily playlist loading tolerance. Suppose further that when you start the playout and the time for starting the second playlist approaches it appears so that the end time of the currently playing clip is closer to the end time of the first clip from the second playlist. In other words, the end time of the first clip of the second playlists occurs sooner than the end of the currently playing clip. Now let us assume that the Daily playlist loading tolerance appears to be greater or equal to the above time difference. As a result, when you restart AirBox, the start time of the second clip in the new playlist will be updated by subtracting the time difference and the overlapping and the initial clip of the second playlist will be removed.

Example 39.   Wait for Playing Clip End Option with Offset the Whole Playlist:

Open AirBox and go to Settings – Startup. Select the Wait for playing clip end radio button, check the Allow Clip Trimming box and save your settings. Suppose that you have two overlapping daily playlists, as in Example 35. Suppose further that you have set a value for the Daily playlist loading tolerance and that the delay of the new playlist is within this value. As a result, when you start the playout and the start time of the second playlist approaches, the new playlist will be moved with the time difference between the two playlists.

Example 40.   Wait for Playing Clip End Option with no Clip Trimming, nor Offset:

Open AirBox and go to Settings – Startup. Select the Wait for playing clip end radio button. Suppose that you have two overlapping daily playlists, as in Example 35. Suppose further that you have set a value for the Daily playlist loading tolerance, which is less than the difference between the playlists. As a result, when you start the playout and the start time of the second playlist approaches, the overlapping clip from the new playlist will be removed and

the gap between the current clip end and the start time of the next clip in the playlist will be filled with an auto-fill clip if the Fill gap between playlists option is checked. Otherwise, the playout of the second playlist will be Cued.

APPENDIX 6 – TitleBox specific Class Properties, Functions, and Procedures Explained

TTBOBJECT (INHERITOR OF TOBJECT)

TTBObject Property/Procedure/Function	Description	Objects That Use It
function Create(const ObjName: String): Constructor	Links to an existing object in the project	All
function CreateNew(const ObjType, ObjName: String): Constructor	Creates a new object in the project and links to it	All
procedure Free	Frees a TTBobject variable	All
procedure Delete	Deletes currently selected object and sets it free	All
procedure Play	Runs currently selected object	All
procedure Stop(AutoStop: Boolean = False)	Stops currently selected object	All
procedure Pause	Pauses object's execution	All
procedure Toggle	Switches object's status from Play to Stop and vice versa	All
procedure UpdateParams	Updates the text when formatting properties are changed through the script	Text, Roll, Crawl, Chat objects
function MoveToSlide(const SlideName: String): Boolean	Moves the object to an existing slide	All
procedure Move(IncX, IncY: Integer; Frames: Integer = 0)	Moves the selected object by <IncX> and <IncY> in the respective directions. <IncX> and <IncY> may be positive or negative values. The movement is relative to object's current position.	All

Channel in a Box User Manual

TTBObject Property/Procedure/Function	Description	Objects That Use It
	Optionally a number of frames may be specified to define the duration of the change.	
procedure MoveTo(PosX, PosY: Integer; Frames: Integer = 0)	Same as procedure Move command except that <PosX> and <PosY> define the new position as absolute values.	All
procedure SizeTo(Width, Height: Integer; Frames: Integer = 0)	Sets object's size according to the specified values.\n\nIf some of the values are 0, the respective property of the object is not changed.	All
procedure Rect(Left, Top, Right, Bottom: Integer)	Sets object's position and size with one command	All
procedure RectTo(Left, Top, Width, Height: Integer; Frames: Integer = 0)	Defines a new position of the object as absolute values and sets the object's size according to the specified values. If some of the values are 0, the respective property of the object is not changed.	All
procedure TopOne	Moves the currently selected object with 1 layer to top (Z-order)	All
procedure BackOne	Moves the currently selected object with 1 layer to bottom	All
procedure ToTop	Moves the currently selected object to top	All
procedure ToBack	Moves the currently selected object to bottom	All
procedure Loop(Start: Integer; Stop: Integer = -1)	Defines the number of loops to run. If you set '0' the object will be looped endlessly.	Roll, Crawl, Animation File, Animation Sequence, Banner, Sound, Digital Clock objects
procedure TextAssign(RichEdit: TRichEdit)	Assigns an object of type TRichEdit, i.e., places its text with its formatting in the object at matter	Text, Roll, Crawl, Chat objects
procedure TextSelect(StartPos: Integer = -1; EndPos: Integer = -1)	Use this command to change the parameters of a part of the text. Selects part of the text starting at [StartPos] and ending at [EndPos], where the first letter of the text is indexed with '1'. Use a '-1' value for selecting the text from the beginning, until the end, or both, as follows: • If [StartPos] = -1 and [EndPos] has a positive integer value, the text will be selected from the beginning to the letter with index, equal to the [EndPos] value. • If [StartPos] has a positive integer value and [EndPos] = -1, the text will be selected from the letter with index, equal to the [StartPos] value until the end • If [StartPos] = -1 and [EndPos] = -1 the whole text will be selected	Text, Roll, Crawl, Chat objects

Channel in a Box User Manual

TTBObject Property/Procedure/Function	Description	Objects That Use It
property AutoWrap: Boolean	Enables / Disables the auto wrapping of the selected text	Text, Roll, Crawl, Chat objects
property FontSize: Integer	Sets the font size for the selected text.	Text, Roll, Crawl, Chat objects
property FontName: String	Sets the font for the selected text.	Text, Roll, Crawl, Chat objects
property FontShadowSize: Integer	Sets the font shadow size to the selected text.	Text, Roll, Crawl, Chat objects
property FontStyle: TFontStyles	Sets a variable of type TFontStyle with value fsBold, fsItalic, fsStrikeout, or fsUnderline to the selected text.	Text, Roll, Crawl, Chat objects
property FontShadowAngle: Integer	Sets the font shadow angle to the selected text.	Text, Roll, Crawl, Chat objects
property FontShadowOffset: Integer	Sets the font shadow offset to the selected text.	Text, Roll, Crawl, Chat objects
property FontShadowColor: Integer	Sets the font shadow color to the selected text.	Text, Roll, Crawl, Chat objects
property FontEdgeSize: Integer	Sets the font edge size to the selected text.	Text, Roll, Crawl, Chat objects
property FontEdgeColor: Integer	Sets the font edge color to the selected text.	Text, Roll, Crawl, Chat objects
property FontLineSpace: Integer	Sets the line space of the font to the selected text.	Text, Roll, Crawl, Chat objects
property FontColor: Integer	Sets the font color to the selected text.	Text, Roll, Crawl, Chat objects
property FontJustify: Integer	Justifies the font horizontally in accordance to the following values: '0' for Center '1' for Left '2' for Right	Text, Roll, Crawl, Chat objects
property FontAlignment: Integer	Vertically aligns the font in accordance to the following values: '0' for Bottom '1' for Center '2' for Right	Text, Roll, Crawl, Chat objects
property FontUseFill: Boolean	Disables / Enables fill of the text	Text, Roll, Crawl, Chat objects

TTBObject Property/Procedure/Function	Description	Objects That Use It
property FontUseOutline: Boolean	Disables / Enables outline of the text	Text, Roll, Crawl, Chat objects
property FontUseShadow: Boolean	Disables / Enables shadow of the text	Text, Roll, Crawl, Chat objects
property Name: String	Sets or gets the name of the object	All
property Text: String	Sets or gets text data of the selected object. ANSI, UTF8 and RTF are supported.	Text, Roll, Crawl, Chat objects
property ObjectType: String	Returns the type of an object	All
property State: TObjectState	Returns the property state: autostop, pause, play, or stop	All
property Color: Integer	Sets or gets the color of the object	All
property Tag: Integer	Sets or gets the tag number of the object	All
property Top: Integer	Sets or gets the Y coordinate of an object in terms of pixels	All
property Left: Integer	Sets or gets the X coordinate of an object in terms of pixels	All
property Width: Integer	Sets or gets the width of an object in pixels	All
property Height: Integer	Sets or gets the height of an object in pixels	All
property Delay: Extended	Sets or gets a delay play time for an object in seconds	All
property Speed: Extended	Sets or gets the speed of an object. If you set a speed of '0' the object will not move. If you set a negative speed value you will reverse the movement of the object.	Rolls, Crawls, Animation Files, Animation Sequences, Mesh Files, Chat Notes/Lines/Rolls
property Link: String	Links the specified file to a TEXT or GRAPHICS object. The file's content is used as a data source for the object.	Text, Roll, Crawl objects
property LinkAutoPlay: Boolean	Disables / Enables Auto play of the object linked to a file whenever the data in the file changes.	Text, Roll, Crawl objects
property ScrollStyle: TScrollDrawStyle	Defines the scroll style of the object by setting a TScrollDrawStyle variable to it, which can have the following values:	Rolls, Crawls, Animation Files, Animation Sequences, Mesh Files, Chat objects

TTBObject Property/Procedure/Function	Description	Objects That Use It
	[ssColorBlank] – the background of the current text will remain until the text disappears from the screen. The new text's background will appear as soon as the text is displayed. [ssEmpty] – there will be background only under the displayed text. As soon as the text ends, the background will disappear [ssSigned] – no space between the backgrounds of two successive texts	
procedure ClockPrepare	Sets the initial clock time and waits for start command.	Digital Clock Objects
procedure ClockStart	Starts the clock.	Digital Clock Objects
procedure ClockStop	Stops the clock.	Digital Clock Objects
property ClockIntermediate: Boolean	Disables / Enables clock freeze.	Digital Clock Objects
Property ClockVisible: Boolean	Shows / hides a clock object	Analogue/Digital Clock Objects
property ClockCountdown: Boolean	Valid only in counter mode. When enabled the counter counts down. If disabled, the counter counts up.	Digital Clock Objects
property ClockStyle: Boolean	Defines the mode of operation for the Clock object. If True – object is in Clock mode and runs as an ordinary digital clock If False – object is in Counter mode counts from initial to stop time if the option ClockUseStopTime is on.	Digital Clock Objects
property ClockOffset: Extended	Used to create clocks for different time zones. Specifies a time offset related to the current time. The format is: <hh:mm:ss>	Analogue/Digital Clock Objects
property ClockInitTime: Extended	Sets the clock initial time in format <hh:mm:ss>	Digital Clock Objects
property ClockStopTime: Extended	Sets the clock stop time in format <hh:mm:ss>	Digital Clock Objects
property ClockUseStopTime: Boolean	Disables / Enables the use of the clock stop time.	Digital Clock Objects

TTBObject Property/Procedure/Function	Description	Objects That Use It
property ClockMask: String	Sets a mask for the format the time will be displayed in. The following options are available: HH:MM, HH:MM:SS, HH:MM:SS:MS, HH:MM:SS:FR, MM:SS, or SS	Digital Clock Objects
procedure InEffect(Effect: TEffectType; Duration: Extended = 1.0; Motion: Integer = 0; FullScreen: Boolean = True)	Sets an IN effect for the selected object	All
procedure OutEffect(Effect: TEffectType; Duration: Extended = 1.0; Motion: Integer = 0; FullScreen: Boolean = True)	Sets an OUT effect for the selected object	All
procedure LoadImage(const FileName: String)	Loads the specified file in the selected object	Text, Picture Objects
procedureImageAssign(Bmp32: TBitmap32)	Assigns an already defined TBitmap32 image to an object	Text, Picture Objects
property ImageStyle: TStretchDrawStyle	Defines image display style from the following options: '1' for Stretch '2' for Tile '3' for Center	Text, Picture Objects
property ImageAlpha: Integer	Defines image transparency 0 to 255, where 00 is full transparency, and 255 is solid color	Text, Picture Objects
property ImageFlip: Boolean	Disables / Enables image flip	Text, Picture Objects
procedure LoadMedia(const FileName: String)	Loads the specified media file in the selected object.	DirectShow Media Source Objects
property MediaPosX: Integer	Defines the X position of the video in the selected object	DirectShow Media Source Objects
property MediaPosY: Integer	Defines the Y position of the video in the selected object	DirectShow Media Source Objects
property MediaAlpha: Integer	Defines the transparency for the selected object from 0 to 255, where 00 is full transparency, and 255 is solid color	DirectShow Media Source Objects

TTBObject Property/Procedure/Function	Description	Objects That Use It
property MediaStyle: Integer	Sets a style to the Media object from two options: '0' for Stretch '1' for Center	DirectShow Media Source Objects
property MediaColor: Integer	Defines the background color in case the object is in <Center> mode and media does not fit in the object (there is some empty space left).	DirectShow Media Source Objects
property MediaQuality: Integer	Sets video quality from the following options (note that '1' is the recommended value): '0'for Pure '1' for Low quality '2'for Medium quality '3' for High quality '4'for Highest quality	DirectShow Media Source Objects
property MediaUseOwnAlpha: Boolean	Disables/ Enables the use of own ALPHA	DirectShow Media Source Objects
property MediaUseAlpha: Boolean	Disables / Enables the use of ALPHA	DirectShow Media Source Objects
property MediaKeepAspect: Boolean	Disables / Enables the <Keep aspect> option	DirectShow Media Source Objects
property MediaAudioVolume: Integer	Sets the volume for the media in decibels from 0 to 10,000, where 0 is the highest volume, and 10,000 is mute	DirectShow Media Source Objects
property MediaAudioDelay: Integer	Sets the delay value for the audio related to the video in milliseconds. This command is used to synchronize the audio with the video in the DXMedia object.	DirectShow Media Source Objects
procedure Event(EventNo: Integer)	Triggers user defined event specified by its number.	All
property OnPlay: String	Sets or gets a user defined task to an object to be executed on Play	All
property OnStop: String	Sets or gets a user defined task to an object to be executed on Stop	All
property OnPause: String	Sets or gets a user defined task to an object to be executed on Pause	All

TTBObject Property/Procedure/Function	Description	Objects That Use It
property OnAutoStop: String	Sets or gets a user defined task to an object to be executed on AutoStop	All
property OnLoop: String	Sets or gets a user defined task to an object to be executed on Loop	All
property OnEndInputTransition: String	Sets or gets a user defined task to an object to be executed on End Input Transition	All
property OnStartOutputTransition: String	Sets or gets a user defined task to an object to be executed on Start Output Transition	All
property OnDataReady: String	Sets or gets a user defined task to an object to be executed on Data Ready	All
property OnPropertychange: String	Sets or gets a user defined task to an object to be executed on Property Change	All
property OnTransform: String	Sets or gets a user defined task to an object to be executed on Transform	All

TTBSLIDE

TTBSlide Property/Procedure/Function	Description	Used By
function Create(const Name: String): TTBSlide: Constructor	Links to an existing slide in the project	Slides
function CreateNew(const Name: String): TTBSlide: Constructor	Creates a new slide in the project	Slides
procedure Free	Frees a TTBSlide object	Slides
procedure Delete	Removes currently selected slide and sets it free. All objects from the slide are moved to the Unassigned Slide.	Slides
procedure ObjectList(List: TStrings)	Get the list of all objects currently contained in the Slide	Slides

procedure Play	Plays the specified slide (Jump to)	Slides
procedure Stop	Stops the specified slide	Slides
property Index: Integer	Sets or gets the index number of a slide	Slides
property Name: String	Sets or gets the name of a slide	Slides
property BackColor: Integer	Sets or gets the background color of a slide	Slides
property Duration: Extended	Sets or gets the duration of a slide in seconds	Slides
property Enabled: Boolean	Disables / Enables a slide	Slides
property Lock: Boolean	Locks / Unlocks a slide	Slides
property PlayMode: TPlayMode	Sets or gets a property of type TPlayMode to a slide, which can have the following values: pmAdd,pmClear,pmCrossPlay, pmStopAll, and pmStopPrevious	Slides
property State: TObjectState	Gets a property of type TObjectState, which can have the following values: osAutoStop,osPause,osPlay, and osStop	Slides

TTBSLIDESMANAGER

TTBSlidesManager Property/Procedure/Function	Description	Used By
procedure SlidesList(List: TStrings)	Lists the names of all slides in a variable of type TStrings	Slides Manager
function Count: Integer	Returns the number of the slides in Slides Manager	Slides Manager
procedure Play	Plays the slideshow	Slides Manager
procedure Stop	Stops the slideshow	Slides Manager
procedure Pause	Pauses the slideshow	Slides Manager
procedure Reset	Initializes all buffers used by the slideshow	Slides Manager

TTBSlidesManager Property/Procedure/Function	Description	Used By
procedure Prev	Plays the previous slide in the Slides Manager	Slides Manager
procedure Next	Plays the next slide in the Slides Manager	Slides Manager
procedure Delete(Index: Integer)	Removes the slide, designated by index number from the Slides Manager. All objects from the slide are moved to the Unassigned Slide.	Slides Manager
procedure Move(OldIndex, NewIndex: Integer)	Moves the currently selected slide to a new position by shifting the indexes of the other slides from the Slides Manager.	Slides Manager
index property Slide(p0: String): Integer	Returns the index number of a Slide, distinguished by its name	Slides Manager
index property Items(p0: Integer): String	Returns the name of an item (Slide), distinguished by its index number	Slides Manager

TTBDATAPROVIDER

TTBDataProvider Property/Procedure/Function	Description	Used By
function Create(const Name: String): TTBDataProvider: Constructor	Links an object to a configured Data Provider	Data Providers
procedure Free	Frees a Data Provider	Data Providers
procedure Delete	Removes the Data Provider and sets it free	Data Providers
procedure ObjectsList(List: TStrings)	Lists the names of all objects, linked to a given Data Provider in a variable of type TStrings	Data Providers
procedure Play	Plays all objects, linked to a Data Provider	Data Providers
procedure Stop	Stops all objects, linked to a Data Provider	Data Providers
procedure CursorStart	Starts the cursor in a Data Provider	Data Providers
procedure CursorStop	Stops the cursor in a Data Provider	Data Providers

TTBDataProvider Property/Procedure/Function	Description	Used By
procedure Top	Moves the cursor to the top line in the Data Provider	Data Providers
procedure Bottom	Moves the cursor to the bottom line in the Data Provider	Data Providers
procedure Prev	Moves the cursor to the previous line in the Data Provider	Data Providers
procedure Next	Moves the cursor to the next line in the Data Provider	Data Providers
procedure Scroll(Lines: Integer = 1)	Moves the cursor with a given number of lines forward.	Data Providers
procedure Update	Refreshes the information, sent from the Data Provider	Data Providers
procedure ColumnsList(List: TStrings)	Lists the names of the columns from the Data Provider to a variable of type TStrings	Data Providers
function LinkObject(Obj: TTBObject; DataColumn: String; Offset: Integer = 0; UseEffects: Boolean = False): Boolean	Sets the properties of a link between an object and a Data Provider as in the Object Link Properties Dialog	Data Providers
procedure DeleteObjectLink(Obj: TTBObject)	Removes a link between a Data Provider and an object	Data Providers
function GetValue(Col, Row: Integer; var Data: Variant; var DataType: TDPDataType): Boolean	Gets the value of a particular entry in a Data Provider, designated by its row and column number	Data Providers
function GetColumn(Col: Integer; var DataType: TDPDataType): Variant	Gets the whole column from a Data Provider	Data Providers
property Distributor: TTBDataDistributor	Sets a Data Distributor to a Data Provider	Data Providers
property Name: String	Sets the name of a Data Distributor	Data Providers
property State: TObjectState	Sets a variable of type TObjectState, which can have the following values: osAutoStop, osPause ,osPlay, and osStop	Data Providers
property ColCount: Integer	Sets the number of columns in a particular Data Provider to an integer variable	Data Providers
property RowCount: Integer	Sets the number of rows in a particular Data Provider to an integer variable	Data Providers

TTBDataProvider Property/Procedure/Function	Description	Used By
index property Columns(p0: Integer): String	Sets the name of a column, distinguished by its number, to a string variable	Data Providers

TTBDATADISTRIBUTOR

TTBDataDistributor Property/Procedure/Function	Description	Used By
property Update: Integer	Sets how often to update the source information from a Data Provider in seconds. If you enter '0', the updating will be Automatic	Data Distributors
property Scroll: Integer	Sets the time in seconds, by which to scroll the cursor in the Data Provider. If you enter '0' the scrolling will be Manual	Data Distributors
property ScrollBy: Integer	Sets the number of rows, by which the cursor scrolls within a Data Provider	Data Distributors
property ResetOnDatasetChange: Boolean	Disables / Enables the Reset On Dataset Change property for the scroll options of the Data Distributor	Data Distributors
property ResetOnFieldChange: Boolean	Disables / Enables the Reset On Field Change property for the scroll options of the Data Distributor	Data Distributors
property StartCursorOnPlay: Boolean	Disables / Enables the Start Cursor On Play property for the scroll options of the Data Distributor	Data Distributors
property NextSlideOnBottom: Boolean	Disables / Enables the Next Slide On Bottom property for the scroll options of the Data Distributor	Data Distributors
property StopObjectsOnBottom: Boolean	Disables / Enables the Stop Objects On Bottom property for the scroll options of the Data Distributor	Data Distributors
property StopObjectsOnAnyStop: Boolean	Disables / Enables the Stop Objects On Any Stops property for the scroll options of the Data Distributor	Data Distributors
property StartObjectsOnAnyPlay: Boolean	Disables / Enables the Start Objects On Any Play property for the scroll options of the Data Distributor	Data Distributors
property StartObjectsOnUpdate: Boolean	Disables / Enables the Start Objects On Update property for the scroll options of the Data Distributor	Data Distributors

TTBPROJECT

TTBProject Property/Procedure/Function	Description	Used By
procedure New	Opens a new project	Projects
procedure Start	Starts the project	Projects
procedure Stop	Stops the project	Projects
procedure Clear	Clears the project	Projects
procedure Close	Closes TitleBox	Projects
function Load(const TemplateName: String): Boolean	Loads a project in TitleBox	Projects
function Merge(const TemplateName: String): Boolean	Merges a new project to the existing one	Projects
procedure SaveAs(const TemplateName: String)	Saves a project under a given name	Projects
procedure NetStart	Switches to Net Control mode of operation	Projects
procedure NetStop	Switches off Net Control mode of operation	Projects
procedure StopAll	Stops all objects in the project	Projects
procedure MoveAll(IncX, IncY: Integer; ExcludeList: String = ")	Moves all objects in the project. The ExcludeList variable is optional. If you enter a value for it, the objects, listed there will not be moved.	Projects
procedure ObjectsList(List: TStrings)	Gets the list of all objects currently loaded in TitleBox.	Projects
procedure TagsList(Tag: Integer; List: TStrings)	Gets the list of all objects with a given tag. If you enter a negative value, a list of all existing tags will be returned.	Projects
procedure TasksList(List: TStrings)	Gets the list of all tasks currently set in TitleBox.	Projects
procedure DataProvidersList(List: TStrings)	Gets the list of all Data Providers currently set in TitleBox.	Projects

TTBProject Property/Procedure/Function	Description	Used By
property State: TObjectState	Gets a variable of type TObjectState, which can have the following values: osPlay, and osStop	Projects
property StateStr: String	Gets the current state of the project in a string variable	Projects
property Modified: Boolean	Returns a true / false value, designating if the project has been modified since the last time it has been saved or opened	Projects
property AspectRatio: Extended	Gets the Aspect Ratio, in which the project is set	Projects
property ResolutionX: Integer	Gets the X resolution, in which the project is set	Projects
property ResolutionY: Integer	Gets the Y resolution, in which the project is set	Projects
property Version: String	Gets the last TitleBox version, in which the project has been saved	Projects

TITLEBOX FUNCTIONS

Function	Description
Execute(const CommandScript: String)	Executes a script, written for the Command Script
LoadScript(const FileName: String): Boolean	Loads any script, written for TitleBox. If you use the name of an existing script, it will be overwritten by the new script
SetScript(const Name, Script: String)	Saves a script with a given name. If you use the name of an existing script, it will be overwritten by the new script
RunTask(const Task: String)	Runs a task, saved in the Task Manager, designated by its name
TBVersion: String	Returns the version number of the currently opened TitleBox
Pause(Objects: array of String)	Pauses an array of objects, designated by their names
PauseList(List: TStrings)	Pauses a list of objects
Play(Objects: array of String)	Plays an array of objects, designated by their names
PlayList(List: TStrings)	Plays a list of objects
Stop(Objects: array of String)	Stops an array of objects, designated by their names
StopList(List: TStrings)	Stops a list of objects
Toggle(Objects: array of String)	Toggles an array of objects, designated by their names
ToggleList(List: TStrings)	Toggles a list of objects

TBITMAP32 (INHERITOR OF TOBJECT)

TBitmap32 Property/Procedure/Function	Description
function Create: Constructor	Creates a TBitmap32 item
procedure Free	Frees a TBitmap32 item
function ClassName: String	Returns the class name of an item in a String variable
procedure Assign(Bmp32: TBitmap32)	Assigns a Bmp32 object to a TBitmap32 item
procedure AssignFrom(Bmp: TBitmap; Flip: Boolean = True)	Assigns a Bmp32 object to the current TBitmap32 item. The image can be flipped.
procedure AssignTo(Bmp: TBitmap; Flip: Boolean = True)	Assigns the current TBitmap32 item to another Bmp32 object. The image can be flipped.
procedure Draw(X, Y: Integer; Bmp32: TBitmap32; Mix: Boolean = False; Flip: Boolean = False)	Draws a TBitmap32 item in given X and Y coordinates. The image can be flipped. It can also be mixed with the current image if it has a defined alpha value.
procedure DrawFrom(X, Y: Integer; Bmp: TBitmap; Flip: Boolean = False)	Copies a Bmp32 object in the current item in given X and Y coordinates. The image can be flipped.
procedure DrawTo(X, Y: Integer; Bmp: TBitmap; Mix: Boolean = False; Flip: Boolean = False)	Copies the current item in given X and Y coordinates to another Bmp32 object. The image can be flipped. It can also be mixed with the current image if it has a defined alpha value.
procedure Fill(Color:TRGBA)	Fills a TBitmap32 item with a given color
procedure FillRect(Left, Top, Right, Bottom: Integer; Color: TRGBA; Mix: Boolean = False)	Creates a Rectangle by giving it coordinates and fills it with a given color. The rectangle can be mixed with the current image if it has an alpha value.
procedure LoadFromFile(const FileName: String)	Loads a TBitmap32 item from a file
procedure LoadFromStream(const S: TStream)	Loads a TBitmap32 item from a stream
procedure SaveToFile(const FileName: String)	Saves a TBitmap32 item to a file
procedure SaveToStream(const S: TStream)	Saves a TBitmap32 item to a stream
procedure SetSize(Width, Height: Integer; PixelFormat: TPixelFormat = pf32bit)	Sets width, height, and pixel format, defined by a TPixelFormat class, to a TBitmap32 item

procedure SetTransparent(Color: TRGBA; Alpha, NoAlpha: Integer)	Searches in an object for a color by its RGB value and sets an alpha value to it in [Alpha], as well as an alpha value for the remaining part of the object, i.e., the one that does not contain the searched color, in [NoAlpha]
property Alpha: Integer	Defines image transparency 0 to 255, where 00 is full transparency, and 255 is solid color
property Canvas: TCanvas	Return the canvas object of class TCanvas for a TBitmap32 item
property Force32bit: Boolean	Formats an image to 32-bit
property Height: Integer	Gets/ sets the height of a TBitmap32 item
property Pixel(p0: Integer; p1: Integer): TRGBA	Gets / sets the pixel value at coordinates X, Y
property PixelFormat: TPixelFormat	Sets a pixel format, defined by a TPixelFormat class to a TBitmap32 item
property Pixels: Variant	Returns the address of the pixel buffer
property ScanLine(p0: Integer): Pointer	Returns the address for the first pixel in Line
property Size: Integer	Returns the pixel size of a TBitmap32 item
property Width: Integer	Gets / Sets the width of a TBitmap32 item

TTBSTORAGE

TTBStorage Property/Procedure/Function	Description
procedure Clear	Clears the TTBStorage class, i.e., removes all variables saved in it
function Count: Integer	Returns the number of variables saved TTBStorage
index property Items(p0: Integer): Variant	Gets / sets an item in TTBStorage, distinguished by its index

For a detailed description of all other classes, supported by the *TitleBox Program Script Engine*, please, visit:
http://docwiki.embarcadero.com/VCL/en/Classes
http://docwiki.embarcadero.com/VCL/en/Controls
http://docwiki.embarcadero.com/VCL/en/Graphics

APPENDIX 7 – TitleBox Program Script Examples

Below you can find simple examples, presenting the basic *TitleBox Program Script* classes and functions, as well as some basic programming rules and organization to be used when programming with this new *TitleBox* feature. Each example is shown in all the four language syntaxes, supported by the *TitleBox Script Engine* and has a short explanation.

Example 1

Pascal	C++

```
var

  Obj: TTBObject;


begin

  Project.Start;


  Obj := TTBObject.Create('Text 1');

  if Assigned(Obj) then

  try

    Obj.Play;

  finally

    Obj.Free;

  end;


  Obj := TTBObject.CreateNew('PICTURE', 'Pic1');

  if Assigned(Obj) then

  try

    Obj.Left := 10;

    Obj.Top := 10;

    Obj.Width := 200;

    Obj.Height := 200;

    Obj.Play;
```

```
{

  Project.Start;


  TTBObject Obj = new TTBObject("Text 1");

  if(Obj)

  try

  {

    Obj.Play;

  }

  finally

  {

    Obj.Free;

  }


  Obj = TTBObject.CreateNew("PICTURE", "Pic1");

  if(Obj)

  try

  {

    Obj.Left = 10;

    Obj.Top = 10;

    Obj.Width = 200;

    Obj.Height = 200;
```

```
  finally                                          Obj.Play;

    Obj.Free;                                    }

  end;                                          finally

end.                                            {

                                                  delete Obj;

                                                }

                                              }
```

Basic	Java

```
Project.Start                                Project.Start;


Obj = new TTBObject("Text 1")                var Obj = new TTBObject("Text 1")

if Assigned(Obj) then                        if(Obj)

  try                                          try

    Obj.Play                                     Obj.Play

  finally                                      finally

    Obj.Free                                     Obj.Free;

  end try

end if                                        Obj = TTBObject.CreateNew("PICTURE", "Pic1")

                                              if(Obj)

                                              try

Obj = TTBObject.CreateNew("PICTURE", "Pic1")  {

if Assigned(Obj) then                           Obj.Left = 10

  try                                           Obj.Top = 10

    Obj.Left = 10                               Obj.Width = 200

    Obj.Top = 10                                Obj.Height = 200

    Obj.Width = 200                             Obj.Play

    Obj.Height = 200                          }

    Obj.Play                                  finally

  finally                                     delete Obj;

delete Obj
```

295

```
  end try

 end if


 end
```

Example 1 Explained:

1. Declaration of an object variable, named **Obj** of type **TTBObject** (in **Pascal** and **C++** only)
2. Starting the Project
3. Linking the **Obj** variable to an existing object, named "**Text 1**"
4. Checking if the **Obj** variable is actually linked to a "**Text 1**" object with an **if** statement
5. **Obj** variable is played and freed within a **try-finally** block
6. A new picture object is created in the work area and linked to the **Obj** variable
7. Check if the picture object is actually assigned for the **Obj** variable
8. The new picture object is given **X** and **Y** coordinates, as well as a **width** and **height** values in pixels and it is played and freed in a **try-finally** block

Example 2

Pascal	C++

```pascal
var

  Obj: TTBObject;

  St: TObjectState;


begin

  Obj := TTBObject.Create('Text 1');

  if Assigned(Obj) then

  try

    St := Obj.State;


    if St = osPlay then

      Obj.Stop

    else

      Obj.Play;


  finally

    Obj.Free;

  end;

end.
```

```cpp
{

  TTBObject Obj = TTBObject.Create("Text 1");

  if (Assigned(Obj))

    try

    {

      TObjectState St = Obj.State;


      if (St == osPlay)

        Obj.Stop;

      else

        Obj.Play;

    }

    finally

    {

      Obj.Free;

    }

}
```

Basic	Java

```
Obj = TTBObject.Create("Text 1")

if Assigned(Obj) then

  try

    St = Obj.State

    if St = osPlay then Obj.Stop

    if St = osStop then Obj.Play

  finally

    Obj.Free

  end try

end if


end
```

```
var Obj = TTBObject.Create("Text 1");

if (Assigned(Obj))

  try

  {

    var St = Obj.State


    if (St == osPlay)

      Obj.Stop

    else

      Obj.Play;

  }

  finally

    Obj.Free;
```

Example 2 Explained:

1. Declaration of a variable of type **TTBObject**, named **Obj** and a variable of type **TOBjectState**, named **St** (in **Pascal** and **C++** only)
2. Linking the **Obj** variable to an existing object, named "**Text 1**"
3. Check if the **Obj** variable is actually linked to a "**Text 1**" object with an **if** statement
4. Setting a value to the St variable, equal to the state of the Obj variable in a **try-finally** block
5. Introducing an **if** statement, which checks if the "**Text 1**" is in state play. If yes, the **Obj** variable is stopped. If no, the **Obj** variable is played.
6. The **Obj** variable is freed.

Example 3

Pascal	C++
| ```pascal
var
  List: TStringList;
  I: Integer;
  Obj: TTBObject;
  S1, S2: String;

begin
  List := TStringList.Create;
  try
    Project.ObjectsList(List);

    for I := 0 to List.Count - 1 do
    begin
      Obj := TTBObject.Create(List[I]);
      if Assigned(Obj) then
      try
        S1 := Obj.Name;
        if Obj.State = osPlay then
          S2 := 'Play'
        else
          S2 := 'Stop';
      finally
        Obj.Free;
      end;

      ShowMessage(Format('%s: State = %s', [S1, S2]));
    end;
  finally
``` | ```cpp
{
  TStringList List = new TStringList;
  try
  {
    Project.ObjectsList(List);

    for(int i = 0; i < List.Count; i++)
    {
      TTBObject Obj = TTBObject.Create(List[I]);
      if (Assigned(Obj))
      try
      {
        char S1 = Obj.Name;
        char S2;
        if (Obj.State == osPlay)
          S2 = "Play";
        else
          S2 = "Stop";
      }
      finally
      {
        Obj.Free;
      }

      ShowMessage(Format("%s: State = %s", [S1, S2]));
    }
  }
  finally
``` |

```
    List.Free;

  end;

end.
```

Basic

```
List = new TStringList

try

  Project.ObjectsList(List)


  for i = 0 to List.Count - 1

    Obj = TTBObject.Create(List[I])

    if Assigned(Obj) then

    try

      S1 = Obj.Name

      if Obj.State = osPlay then

        S2 = "Play"

      else

        S2 = "Stop"

      end if

    finally

      Obj.Free

    end try

    ShowMessage(Format("%s: State = %s", [S1, S2]))

    end if

  next ' for i

finally

  List.Free

end try


end
```

```
{

  List.Free;

 }

}
```

Java

```
var List = new TStringList

try

{

  Project.ObjectsList(List);


  for(var i = 0; i < List.Count; i++)

  {

    var Obj = TTBObject.Create(List[I]);

    if (Assigned(Obj))

    try

    {

      var S1 = Obj.Name;

      var S2;

      if (Obj.State == osPlay)

        S2 = "Play"

      else

        S2 = "Stop";

    }

    finally

      Obj.Free;


    ShowMessage(Format("%s: State = %s", [S1, S2]));

  }

}

finally
```

	List.Free;

Example 3 Explained:

1. Declaration of a variable named **List** of type **TStringList**, an integer variable named **I**, a variable named **Obj** of type **TTBObject**, and two string variables, named **S1** and **S2** (in **Pascal** and **C++** only).
2. Creating the **List** variable
3. Assigning the names of all the objects from the project in the **List** variable
4. Initialization of a **try-finally** block
5. Initialization of a **for** loop, walking through the whole list of names.
6. Linking the **Obj** variable to the corresponding object from the **List** variable (with number **I**).
7. Checking if an object is actually assigned to the **Obj** variable with an **if** statement
8. Assigning a value to the **S1** variable, equal to the current name of the **Obj** variable, i.e., the name of the object with number **I** in the **List**.
9. Checking the state of the **Obj** variable (play or stop) with an **if** block
10. Assigning the state of the **Obj** variable to the **S2** variable (play or stop)
11. The **Obj** variable is freed.
12. Showing the current values of **S1** and **S2** within the loop in a Message Dialog
13. Finishing the loop and setting the **List** variable free

Example 4

Pascal	C++
| ```
var

  Slide: TTBSlide;

  List: TStringList;

  I: Integer;

  S: String;


begin

  List := TStringList.Create;

  try

    Slide := TTBSlide.Create('Slide 1');

    if Assigned(Slide) then

      try

        Slide.Play;

        Slide.ObjectsList(List);

        S := 'Objects in Slide "' + Slide.Name + '": ';

        for I := 0 to List.Count - 1 do
``` | ```
{

  TStringList List = new TStringList;

  try

  {

    TTBSlide Slide = TTBSlide.Create("Slide 1");

    if (Assigned(Slide))

      try

      {

        Slide.Play;

        Slide.ObjectsList(List);

        char S = "Objects in Slide " + Slide.Name + ": ";

        for(int i = 0; i < List.Count; i++)

          S = S + List[I] + ", ";

        ShowMessage(S);

      }

      finally
``` |

Left column:

```
      S := S + List[I] + ', ';

    ShowMessage(S);

  finally

    Slide.Free;

  end;

finally

  List.Free;

end;

end.
```

Basic

```
List = new TStringList

try

  Slide = TTBSlide.Create("Slide 1")

  if Assigned(Slide) then

  try

    Slide.Play

    Slide.ObjectsList(List)

    S = "Objects in Slide " + Slide.Name + ": "

    for i = 0 to List.Count - 1

      S = S + List[I] + ", "

    next ' for i

    ShowMessage(S)

  finally

    Slide.Free

  end try

  end if

finally

  List.Free

end try
```

Right column:

```
    {

      Slide.Free;

    }

  }

  finally

  {

    List.Free;

  }

}
```

Java

```
var List = new TStringList

try

{

  var Slide = TTBSlide.Create("Slide 1")

  if (Assigned(Slide))

  try

  {

    Slide.Play

    Slide.ObjectsList(List)

    var S = "Objects in Slide " + Slide.Name + ": "

    for(var i = 0; i < List.Count; i++)

      S = S + List[I] + ", "

    ShowMessage(S)

  }

  finally

    Slide.Free;

}

finally

  List.Free;
```

end	

Example 4 Explained:

1. Declaration of a variable, named **Slide** with a type **TTBSlide**, a variable named **List** of type **TStringList**, an integer variable, named **I**, and a string variable, named **S** (in **Pascal** and **C++** only)
2. Creating the **List** variable
3. Creating the **Slide** variable by linking it to "**Slide 1**" in a **try-finally** block
4. Checking if a slide is actually assigned to the **Slide** variable
5. Playing the **Slide**
6. Assigning the objects' names from **Slide** to the **List** variable
7. Assigning text to the **S** variable, including "Objects in Slide" and the name of the **Slide**, assigned to the **Slide** variable
8. Adding the name of each object from the **List** variable to the **S** variable in a **for** loop
9. Showing the current value of the **S** variable in a Message Dialog
10. The **Slide** variable is freed
11. The **List** variable is freed

Channel in a Box User Manual

Example 5

Pascal	C++

```
var

  Obj: TTBObject;


begin

  Obj := TTBObject.Create('Text 1');

  if Assigned(Obj) then

  try

    Obj.TextSelect(1, 3);

    Obj.FontSize := 50;

    Obj.FontStyle := [fsBold, fsItalic];

    Obj.FontName := 'Arial';

    Obj.UpdateParams;

  finally

    Obj.Free;

  end;

end.
```

```
{

  TTBObject Obj = TTBObject.Create("Text 1");

  if (Assigned(Obj))

  try

  {

    Obj.TextSelect(1, 3);

    Obj.FontSize = 50;

    Obj.FontStyle = [fsBold, fsItalic];

    Obj.FontName = "Arial";

    Obj.UpdateParams;

  }

  finally

  {

    Obj.Free;

  }

}
```

Basic	Java

```
Obj = TTBObject.Create("Text 1")

if Assigned(Obj) then

try

  Obj.TextSelect(1, 3)

  Obj.FontSize = 50

  Obj.FontStyle = [fsBold, fsItalic]

  Obj.FontName = "Arial"

  Obj.UpdateParams

finally

  Obj.Free

end try
```

```
var Obj = TTBObject.Create("Text 1")

if (Assigned(Obj))

try

{

  Obj.TextSelect(1, 3)

  Obj.FontSize = 50

  Obj.FontStyle = [fsBold, fsItalic]

  Obj.FontName = "Arial"

  Obj.UpdateParams

}

finally
```

```
end if


end
```

```
Obj.Free
```

Example 5 Explained:

1. Declaring a variable named **Obj** of type **TTBObject**
2. Linking the **Obj** variable to an already created text object, named "**Text 1**"
3. Checking if an object is actually assigned to the **Obj** variable
4. Initialising a **try-finally** block
5. Selecting the first three symbols from the text object and setting their size, font and font style
6. Updating the parameters of the **Obj** variable so that the new text formatting is applied
7. The **Obj** variable is freed

Example 6

Pascal	C++
| ```pascal
var

  DP: TTBDataProvider;

  List: TStringList;


begin

  DP := TTBDataProvider.Create('FileLink Data Provider Text04');

  if Assigned(DP) then

  try

    DP.CursorStop;

    DP.Top;


    List := TStringList.Create;

    try

      DP.ObjectsList(List);

      PlayList(List);


      DP.Next;

      DP.Scroll(2);


      StopList(List);

    finally

      List.Free;

    end;

  finally

    DP.Free;

  end;

end.
``` | ```cpp
{

  TTBDataProvider DP = TTBDataProvider.Create("FileLink Data Provider Text04");

  if (Assigned(DP))

  try

  {

    DP.CursorStop;

    DP.Top;


    TStringList List = new TStringList;

    try

    {

      DP.ObjectsList(List);

      PlayList(List);


      DP.Next;

      DP.Scroll(2);


      StopList(List);

    }

    finally

    {

      List.Free;

    }

  }

  finally

  {

    DP.Free;

  }
``` |

Basic	Java
	`}`
`DP = TTBDataProvider.Create("FileLink Data Provider Text04")`	`var DP = TTBDataProvider.Create("FileLink Data Provider Text04")`
`if Assigned(DP) then`	`if (Assigned(DP))`
`try`	`try`
	`{`
`  DP.CursorStop`	`  DP.CursorStop`
`  DP.Top`	`  DP.Top`
`  List = new TStringList`	`  var List = new TStringList`
`  try`	`  try`
	`  {`
`    DP.ObjectsList(List)`	`    DP.ObjectsList(List)`
`    PlayList(List)`	`    PlayList(List)`
`    DP.Next`	`    DP.Next`
`    DP.Scroll(2)`	`    DP.Scroll(2)`
`    StopList(List)`	`    StopList(List)`
	`  }`
`  finally`	`  finally`
`    List.Free`	`    List.Free;`
`  end try`	`  }`
`finally`	`finally`
`  DP.Free`	`  DP.Free;`
`end try`	
`end if`	
`end`	

Example 6 Explained:

1. Declaration of a variable, named **DP** of type **TTBDataProvider** and a variable, named **List** of type **TStringList** (in **Pascal** and **C++** only). The Data Provider input is already linked to file "**Text 4.txt**"shown [below](#).
2. Checking if the **DP** variable is actually linked to the existing data provider
3. Initialising a **try-finally** block

4.   Stopping the cursor scrolling within the linked text file
5.   Moving the cursor to the top line of the linked text file
6.   Creating the **List** variable
7.   Initialising another **try-finally** block
8.   Returning a list of all objects, linked to the Data Provider in the **List** variable
9.   Playing all the objects in the **List** variable
10. Scrolling the cursor to the next line in the Data Provider
11. Scrolling the cursor with two lines in the Data Provider
12. Stopping all the objects in the **List** variable
13. The **List** and **DP** variables are freed

Example 7

Pascal	C++

```pascal
var

  L: TStringList;

  O: TTBObject;

  DP: TTBDataProvider;


begin

  DP := TTBDataProvider.Create('FileLink Data Provider Text04');

  if Assigned(DP) then

  try

    L := TStringList.Create;

    try

      DP.ColumnsList(L);


      O := TTBObject.Create('Text 1');

      try

        DP.LinkObject(O, L[1]);

      finally

        O.Free;

      end;

    finally

      L.Free;

    end;
```

```cpp
{

  TTBDataProvider DP = TTBDataProvider.Create("FileLink Data Provider Text04");

  if (Assigned(DP))

  try

  {

    TStringList L = new TStringList;

    try

    {

      DP.ColumnsList(L);


      TTBObject O = TTBObject.Create("Text 1");

      try

      {

        DP.LinkObject(O, L[1]);

      }

      finally

      {

        O.Free;

      }

    }

    finally

    {
```

```
  DP.Distributor.Scroll := 2; // sec

  DP.Distributor.ScrollBy := 1; // row

  DP.CursorStart;

  DP.Play;

 finally

  DP.Free;

 end;

 Project.Start;

end.
```

```
  L.Free;

 }


  DP.Distributor.Scroll = 2; // sec

  DP.Distributor.ScrollBy = 1; // row

  DP.CursorStart;

  DP.Play;

 }

 finally

 {

  DP.Free;

 }


 Project.Start;

}
```

Basic	Java

```
DP = TTBDataProvider.Create("FileLink Data Provider Text04")

if Assigned(DP) then

try

  List = new TStringList

  try

   DP.ColumnsList(List)


   O = TTBObject.Create("Text 1")

   try

    DP.LinkObject(O, List[1])

   finally

    O.Free

   end try

 finally
```

```
var DP = TTBDataProvider.Create("FileLink Data Provider Text04")

if (Assigned(DP))

try

{

  var L = new TStringList

  try

  {

   DP.ColumnsList(L);


   var O = TTBObject.Create("Text 1")

   try

    DP.LinkObject(O, L[1])

   finally

    O.Free;
```

```
    List.Free

  end try


  DP.Distributor.Scroll = 2 'sec

  DP.Distributor.ScrollBy = 1 'row

  DP.CursorStart

  DP.Play

finally

  DP.Free

end try


Project.Start

end if


end
```

```
  }

  finally

    L.Free;


    DP.Distributor.Scroll = 2 // sec

    DP.Distributor.ScrollBy = 1 // row

    DP.CursorStart

    DP.Play

  }

  finally

    DP.Free;


  Project.Start
```

Example 7 Explained:

1. Declaration of a variable, named **L** of type **TStringList**, a variable, named **O** of type **TTBObject**, and a variable, named **DP** of type **TTBDataProvider** (in **Pascal** and **C++** only). The Data Provider input is already linked to file "**Text 4.txt**"shown below.
2. Checking if **DP** variable is actually linked to the existing data provider
3. Initialising a **try-finally** block
4. Creating the **L** variable
5. Adding the names of the columns in the Data Provider to the **L** variable
6. Linking the **O** variable to an existing text object, named "**Text 1**"
7. Initialising another **try-finally** block
8. Setting the link between the data provider and the text object to show data from column with index '1' from the Data Provider, i.e., the second column
9. Setting the **O** variable and the L variable free
10. Setting the Data Distributor for the Data Provider to scroll every 2 seconds
11. Setting the Data Distributor for the Data Provider to scroll by 1 line
12. Starting the cursor movement in the source file
13. Playing all the objects, linked to the Data Provider
14. The **DP** variable is freed
15. Starting the project

Example 8

Pascal	C++

```
var
    DP: TTBDataProvider;
    I, Rows, Cols: Integer;
    ColumnValues: Variant;
    DT: TDPDataType;
    S: String;

begin
    DP := TTBDataProvider.Create('FileLink Data Provider Text04');
    if Assigned(DP) then
    try
        Rows := DP.RowCount;
        Cols := DP.ColCount;

        S := '';

        ColumnValues := DP.GetColumn(0, DT);
        if DT = dtText then
        for I := 0 to Rows - 1 do
            S := S + VarArrayElement(ColumnValues, I) + #13#10;

        ShowMessage(S);

        ColumnValues := null;
    finally
        DP.Free;
    end;
end.
```

```
{
    TTBDataProvider DP = TTBDataProvider.Create("FileLink Data Provider Text04");
    if (Assigned(DP))
    try
    {
        int Rows = DP.RowCount;
        int Cols = DP.ColCount;
        char S = "";
        TDPDataType DT;

        variant ColumnValues = DP.GetColumn(0, DT);
        if (DT == dtText)
        {
            for(int I = 0; I < Rows; I++)
                S = S + VarArrayElement(ColumnValues, I) + #13#10;
        }
        ShowMessage(S);

        ColumnValues = null;
    }
    finally
    {
        DP.Free;
    }
}
```

Basic

DP = TTBDataProvider.Create("FileLink Data Provider Text04")

Java

var DP = TTBDataProvider.Create("FileLink Data Provider Text04");

```
if Assigned(DP) then

try

  Rows = DP.RowCount

  Cols = DP.ColCount

  S = ""

  ColumnValues = DP.GetColumn(0, DT)

  if DT = dtText then

    for I = 0 to Rows - 1

      S = S + VarArrayElement(ColumnValues, I) + #13#10

    next

  end if

  ShowMessage(S)


  ColumnValues = null

finally

  DP.Free

end try

end if


end
```

```
if (Assigned(DP))

try

{

  var Rows = DP.RowCount

  var Cols = DP.ColCount

  var S = ""

  var ColumnValues = DP.GetColumn(0, DT)

  if (DT == dtText)

  {

    for(I = 0; I < Rows; I++)

      S = S + VarArrayElement(ColumnValues, I) + #13#10;

  }

  ShowMessage(S)


  ColumnValues = null;

}

finally

  DP.Free;
```

Example 8 Explained:

1. Declaration of a variable, named **DP** of type **TTBDataProvider**, three integer variables, named **I**, **Rows**, and **Cols**, a variant variable, named **ColumnValues**, a variable, named **DT** of type **TDPDataType**, and a **String** variable, named **S** (in **Pascal** and **C++** only). The Data Provider input is already linked to file "**Text 4.txt**"shown [below](#).
2. Checking if **DP** variable is actually linked to the existing data provider
3. Initialising a **try-finally** block
4. Returning the number of rows in the Data Provider in the **Rows** variable
5. Returning the number of columns in the Data Provider in the **Cols** variable
6. Setting a value of ' ' to the string variable
7. Setting the values from column 1, i.e., the column with index 0 to the **ColumnValues** variable
8. Checking if the data type in the abovementioned column is of type **dtText**, i.e., text data
9. Initialising a for loop, covering all the rows in the column and adding their values to the **S** variable one by one
10. Showing the value of **S** in a message dialog
11. Setting a null value to the **ColumnValues** variable to free memory
12. The **DP** variable is freed

Example 9

Pascal	C++

```pascal
var

  Slide: TTBSlide;

  List: TStringList;


begin

  List := TStringList.Create;

  try

    Slides.SlidesList(List);


    Slide := TTBSlide.Create(List[0]);

    if Assigned(Slide) then

    try

      Slide.Play;

      Slides.Move(Slide.Index, List.Count);

    finally

      Slide.Free;

    end;

  finally

    List.Free;

  end;

end.
```

```cpp
{

  TStringList List = new TStringList;

  try

  {

    Slides.SlidesList(List);


    TTBSlide Slide = TTBSlide.Create(List[0]);

    if (Assigned(Slide))

    try

    {

      Slide.Play;

      Slides.Move(Slide.Index, List.Count);

    }

    finally

    {

      Slide.Free;

    }

  }

  finally

  {

    List.Free;

  }

}
```

Basic	Java

```basic
List = new TStringList

try

  Slides.SlidesList(List)
```

```java
var List = new TStringList

try

{

  Slides.SlidesList(List)
```

```
Slide = TTBSlide.Create(List[0])

if Assigned(Slide) then

try

  Slide.Play

  Slides.Move(Slide.Index, List.Count)

finally

  Slide.Free

end try

end if

finally

  List.Free

end try


end
```

```
var Slide = TTBSlide.Create(List[0]);

if (Assigned(Slide))

try

{

    Slide.Play;

    Slides.Move(Slide.Index, List.Count);

}

finally

  Slide.Free;

}

finally

  List.Free;
```

Example 9 Explained:

1. Declaration of a variable, named **Slide** of type **TTBSlide** and a variable, named **List** of type **TStringList** (in **Pascal** and **C++** only)
2. Creating the **List** variable.
3. Initialising a **try-finally** block
4. Adding the names of the current slides in the project to the **Slides** variable
5. Linking the **Slide** variable to the first slide, i.e. the slide with an index of '0'
6. Checking if a slide is actually linked to the **Slide** variable
7. Initialising another **try-finally** block
8. Playing the **Slide**
9. Changing the index of the slide to the index of the last slide in the **SlidesList**, i.e., moving it at the end of the slide list
10. The **Slide** and the **List** variables are freed

Example 10

Pascal	C++
| ```pascal
var

  Obj: TTBObject;


begin

  Project.New;


  Obj := TTBObject.CreateNew('TEXT', 'Text 1');

  if Assigned(Obj) then

  try

    Obj.Left := 20;

    Obj.Top := 30;

    Obj.Width := 300;

    Obj.Height := 100;

    Obj.Text := 'Test Object';

    Obj.Play;

  finally

    Obj.Free;

  end;


  Project.Start;

end.
``` | ```cpp
{

  Project.New;


  TTBObject Obj = TTBObject.CreateNew("TEXT", "Text 1");

  if (Assigned(Obj))

  try

  {

    Obj.Left = 20;

    Obj.Top = 30;

    Obj.Width = 300;

    Obj.Height = 100;

    Obj.Text = "Test Object";

    Obj.Play;

  }

  finally

  {

    Obj.Free;

  }


  Project.Start;

}
``` |
| **Basic** | **Java** |
| ```basic
Project.New


Obj = TTBObject.CreateNew("TEXT", "Text 1")

if Assigned(Obj) then

try

  Obj.Left = 20
``` | ```java
Project.New;


var Obj = TTBObject.CreateNew("TEXT", "Text 1")

if (Assigned(Obj))

try

{
``` |

```
  Obj.Top = 30

  Obj.Width = 300

  Obj.Height = 100

  Obj.Text = "Test Object"

  Obj.Play

finally

  Obj.Free

end try

end if


Project.Start


end
```

```
  Obj.Left = 20

  Obj.Top = 30

  Obj.Width = 300

  Obj.Height = 100

  Obj.Text = "Test Object"

  Obj.Play

}

finally

  Obj.Free;


Project.Start;

```

Example 10 Explained:

1. Declaration of a variable, named **Obj** of type **TTBObject** (in **Pascal** and **C++** only)
2. Opening a new project.
3. Creating a new text object within the project, named "**Text 1**" and linking it to the **Obj** variable
4. Checking if an object is actually linked to the **Obj** variable
5. Initialising a **try-finally** block
6. Assigning **X** and **Y** coordinates to the text object, as well as width and height
7. Entering the text "**Test Object**" in the object
8. Playing the object
9. Setting the **Obj** variable free
10. Starting the project

Example 11

Pascal	C++
| ```pascal
var

  I: Integer;


begin

  I := 0;

  try

    I := 5 div I;

  except

    I := -1;

    ShowMessage(ExceptionMessage);

  end;

  ShowMessage(I);

end.
``` | ```cpp
{

  int I = 0;

  try

  { I = 5 % I; }

  catch

  {

    I = -1;

    ShowMessage(ExceptionMessage);

  }


  ShowMessage(I);

}
``` |

Basic	Java
| ```basic
dim I = 0

  try

    I = 5 / I

  catch

    I = -1

    ShowMessage(ExceptionMessage)

  end try


  ShowMessage(I)
``` | ```java
var I = 0

  try

    I = 5 / I

  Catch

  {

    I = -1

    ShowMessage(ExceptionMessage)

  }


  ShowMessage(I)
``` |

Example 11 Explained:

1. Declaration of an integer variable, named **I** (in **Pascal** and **C++** only)
2. Setting a value of '0' to the **I** variable
3. Initialising a **try-except / try-catch** block
4. Attempting to divide '5' by the **I** variable

5. Setting a value of '-1' to the I variable in case there is an error between the "**try**" and "**except**" statements and shows a message with a description of the type of error that has occurred. In this case the step will not be executed, since I is equal to '0' and you cannot divide by '0'.
6. Showing the value of **I** in a message dialog

Example 12

Pascal	C++

function Transform(Obj: TTBObject; Text: String): Boolean;
 	
var DP: TTBDataProvider;
 	
begin
 	
  if Text <> '' then
 	
    Result := False
 	
  else
 	
  begin
 	
    DP := TTBDataProvider.Create('FileLink Data Provider Text04');
 	
    DP.Next;
 	
    DP.Free;
 	
    Result := True;
 	
  end;
 	
end;
 	
begin
 	
end.
	bool Transform(TTBObject Obj, char Text)
	
	{
	
	  if (Text != "")
	
	    return(False);
	
	  else
	
	  {
	
	    TTBDataProvider DP = TTBDataProvider.Create("FileLink Data Provider Text04");
	
	    DP.Next;
	
	    DP.Free;
	
	    return(True);
	
	  }
	
	}
	
	{
	
	}

Basic	**Java**

function Transform(Obj, Text)
 	
  if Text <> "" then
 	
    return False
 	
  else
 	
    DP = TTBDataProvider.Create("FileLink Data Provider Text04")
 	
    DP.Next
 	
    DP.Free
 	
    return True
 	
  end if
 	
end function
	function Transform(Obj, Text);
	
	{
	
	  if (Text != "")
	
	    return(False)
	
	  else
	
	  {
	
	    var DP = TTBDataProvider.Create("FileLink Data Provider Text04")
	
	    DP.Next
	
	    DP.Free
	
	    return(True);
	
	  }

end	}
	{
	}

Example 12 Explained:

1. Declaration of a **Transform Function**
2. Declaration of a variable, named **DP** of type **TTBDataProvider**(in **Pascal** and **C++** only)
3. Checking if the Text in the object, to which the transform function is linked is different from ' ', i.e., if there is an empty line.
4. If an empty line is not found, returning a **False** result to the **Transform Function**, i.e., the object will display information as it is fed by the Data Provider
5. If text is not different from ' ', i.e., an empty line is found, the **DP** variable is linked to the *.txt file Text4, shown <u>below</u>.
   The cursor is moved to the next line in the **Text 4** file
6. The **DP** variable is set free
7. A **True** result is returned to the **Transform Function**, i.e., the object has already shown the proper data

Example 13

Pascal	C++
begin	{
if DEBUG then	if(DEBUG)
ShowMessage('Debug!')	ShowMessage("Debug!");
else	else
ShowMessage('Executing!');	ShowMessage("Executing!");
end.	}

Basic	Java
if DEBUG then	if(DEBUG)
ShowMessage("Debug!")	ShowMessage("Debug!")
else	else
ShowMessage("Executing!")	ShowMessage("Executing!");
end if	
end	

Example 13 Explained:

1. Checking if TitleBox is in debug mode
2. If yes, showing a message with text "Debug!"

Channel in a Box User Manual

3. If no, showing a message with text "Executing!"

Example 14

Pascal	C++

``pascal
var
 	
  S: String;
 	
begin
 	
  S := 'Stored Value';
 	
  if Storage[0] <> 500 then
 	
  begin
 	
    Storage[0] := 500;
 	
    Storage[1] := S;
 	
  end;
 	
  ShowMessage(Storage.Count);
 	
  if Storage.Count > 10 then
 	
    Storage.Clear;
 	
end.
``
	``cpp
	{
	
	  char S = "Stored Value";
	
	  if(Storage[0] != 500)
	
	  {
	
	    Storage[0] = 500;
	
	    Storage[1] = S;
	
	  }
	
	  ShowMessage(Storage.Count);
	
	  if(Storage.Count > 10)
	
	    Storage.Clear;
	
	}
	``

Basic	**Java**

``basic
dim S = "Stored Value"
 	
  if Storage[0] <> 500 then
 	
    Storage[0] = 500
 	
    Storage[1] = S
 	
  end if
 	
  ShowMessage(Storage.Count)
 	
  if Storage.Count > 10 then
 	
    Storage.Clear
 	
  end if
 	
end
``
	``java
	var S = "Stored Value"
	
	  if(Storage[0] != 500)
	
	  {
	
	    Storage[0] = 500
	
	    Storage[1] = S
	
	  }
	
	  ShowMessage(Storage.Count)
	
	  if(Storage.Count > 10)
	
	    Storage.Clear
	``

Example 14 Explained:

1. Declaration of a string variable, named **S** (in **Pascal** and **C++** only)
2. Setting a value, equal to "**Stored Value**" to **S**.

3. Initialization of an **if** statement for checking whether the first item in the **Storage** is different from 500. If it is different, assigning a value of 500 to the first item in the **Storage** and a value, equal to **S** to the second item in the **Storage**.
4. Ending the **if** statement.
5. Showing the number of items stored in a Message dialog.
6. Clearing the **Storage** if the number of items, stored is greater than 10.

Example 15

Pascal	C++

uses 'Unit.pas', 'Test.cpp', 'String.vb', 'Script.js';
 	
begin
 	
  ...
 	
end.
	#include "Unit.pas", "Test.cpp", "String.vb", "Script.js"
	
	{
	
	  ...
	
	}

Basic	**Java**

imports "Unit.pas", "Test.cpp", "String.vb", "Script.js"
 	
  ...
 	
end
	import "Unit.pas", "Test.cpp", "String.vb", "Script.js"
	
	{
	
	  ...
	
	}

Example 15 Explained:

1. Assigning the following saved modules to be used in the script: **"Unit.pas"**, **"Test.cpp"**, **"String.vb"**, and **"Script.js"**. Notice that the four files are written in different script syntaxes.
2. Initialising and ending the main part of the script.

Test 4.txt

Header Col_1	Header Col_2	Header Col_3
Line 1 Col 1	Line 1 Col 2	Line 1 Col 3
Line 2 Col 1	Line 2 Col 2	Line 2 Col 3
Line 3 Col 1	Line 3 Col 2	Line 3 Col 3
Line 4 Col 1	Line 4 Col 2	Line 4 Col 3
Line 5 Col 1	Line 5 Col 2	Line 5 Col 3
Line 6 Col 1	Line 6 Col 2	Line 6 Col 3
Line 7 Col 1	Line 7 Col 2	Line 7 Col 3
Line 8 Col 1	Line 8 Col 2	Line 8 Col 3
Line 9 Col 1	Line 9 Col 2	Line 9 Col 3
Line 10 Col 1	Line 10 Col 2	Line 10 Col 3
Line 11 Col 1	Line 11 Col 2	Line 11 Col 3
Line 12 Col 1	Line 12 Col 2	Line 12 Col 3
Line 13 Col 1	Line 13 Col 2	Line 13 Col 3
Line 14 Col 1	Line 14 Col 2	Line 14 Col 3
Line 15 Col 1	Line 15 Col 2	Line 15 Col 3
Line 16 Col 1	Line 16 Col 2	Line 16 Col 3
Line 17 Col 1	Line 17 Col 2	Line 17 Col 3
Line 18 Col 1	Line 18 Col 2	Line 18 Col 3
Line 19 Col 1	Line 19 Col 2	Line 19 Col 3
Line 20 Col 1	Line 20 Col 2	Line 20 Col 3
Line 21 Col 1	Line 21 Col 2	Line 21 Col 3
Line 22 Col 1	Line 22 Col 2	Line 22 Col 3
Line 23 Col 1	Line 23 Col 2	Line 23 Col 3
Line 24 Col 1	Line 24 Col 2	Line 24 Col 3
Line 25 Col 1	Line 25 Col 2	Line 25 Col 3
Line 26 Col 1	Line 26 Col 2	Line 26 Col 3
Line 27 Col 1	Line 27 Col 2	Line 27 Col 3
Line 28 Col 1	Line 28 Col 2	Line 28 Col 3
Line 29 Col 1	Line 29 Col 2	Line 29 Col 3
Line 30 Col 1	Line 30 Col 2	Line 30 Col 3

APPENDIX 8 – Closed Captions in PlayBox Modules

The text below concerns the management of 608 and 708 CC files, as well as the operation of the DMT 608/708 Decoder.

First, when you work with Closed Captions, you need to keep in mind the following issues:

- Most existing files, as well as all files that are captured with **CaptureBox** have only 608 CC, meaning that they need to be transcoded to 708 CC in some way.
- Files that only have 708 CC will not produce 608. Transcoding from 708 CC to 608 CC is usually impossible. Thus, one can rarely find files that only have 708 CC but such files will not contain 608 CC. Files need to have BOTH 608 and 708 CC, meaning that probably ALL of the incoming content is going to have 608 CC. So, having a 608 to 708 CC transcoder and ingesting 608 CC only will be transparent to the files and it will make no difference if the file is kept in its original 708 CC format, or if it is transcoded.
- In case the original 708 CC is lost and only the transcoded 608 CC version is kept, the lost data will probably contain only bells and whistles, which could exist in 708 (in the form of fancier fonts), but the content of the CC file in terms of text, position of lines, colors, and so on, will be the same.

Afterwards, the following technical aspects need to be taken under consideration:

- Full CC Ingestion (608 and 708) from the original source – via SD-SDI or HD-SDI streaming
- Support of DVD, ATSC and SCTE-20/21 CCs in MPEG files (PS, SPTS, MPTS).
  Note: Only ATSC supports 708 CC
- Support of DVD, ATSC, SCTE-20/21 in H.264 files (via MainConcept©)
  Note: MainConcept© has a lot of bugs and many of the files may present errors in the abovementioned formats
- Support of DVD, ATSC, SCTE-20/21 in live streams (MPEG by DMT, H.264 by MainConcept© – same as above)
- Support of 708 CC in live digital inputs (SD-SDI, HD-SDI) - SMPTE-334-2
- All of the above will lead to a pass-through of the available CCs. Sources that only contain 608 CC, will also be transcoded to 708 CC.

APPENDIX 9 – Windows Configurations for Using Non-English Object Names in AirBox – TitleBox Net Control Mode

The following instruction is applicable for users who work with non-English languages for TitleBox objects and use TitleBox together with AirBox. Execute the steps, provided below and you will be able to use TitleBox objects in your target language.

NOTE: If you are using TitleBox and AirBox on different machines, make sure to make the following configurations on both machines.

1. Go to the **Windows Start Menu → Control Panel → Region and Language → Location** tab.

2. From the *Current location* drop-down list select the corresponding location, relative to the language you are using.
3. Go to the **Administrative** tab of the same dialog:

4. Click **Change system locale...** A new dialog will be invoked:

5. From the dialog, displayed above, select the language you are using.

APPENDIX 10 – AIRBOX KEYBOARD SHORTCUTS

PLAYLIST MANAGEMENT

Function	Shortcut 1	Shortcut 2
New playlist	Ctrl+N	
Load playlist	Ctrl+O	
Save playlist	Ctrl+S	Alt+V
Save daily playlist	Ctrl+Q	
Save playlist As	Alt+S	
Add media files	Alt+Ins	Alt+A
Insert media files	Ins	Alt+I
Add dummy clip	Shift+Ctrl+D	
Insert dummy clip	Ctrl+Alt+D	
Add live clip	Shift+Ctrl+V	
Insert live clip	Ctrl+Alt+V	
Edit media/event	Alt+Enter	Double click
Properties	Ctrl+Enter	
Delete current from playlist	Del	Alt+Del

Clear playlist	Ctrl+Del	Alt+C
Playlist loop	@	Shift+2
Insert playlist	Ctrl+I	
Append playlist	Ctrl+A	
Reset playlist	#	
Full Mirror Mode	Ctrl+M	
Idle Mirror Mode	Ctrl+E	
Insert Stop event	Ctrl+Alt+S	
Insert Stop Cue event	Ctrl+Alt+C	
Insert Wait event	Ctrl+Alt+W	
Insert Logo ON	Ctrl+Alt+L	
Insert Logo Preset	Ctrl+Alt+P	
Insert Logo Off	Ctrl+Alt+O	
Insert GPI event	Ctrl+Alt+G	
Insert Kramer Switch output	Ctrl+Alt+K	
Insert Kramer Matrix output	Ctrl+Alt+R	
Insert TitleBoxNetCtrl output	Ctrl+Alt+T	
Insert Note event	Ctrl+Alt+N	

Channel in a Box User Manual

Insert Return Event	Backspace	Alt + R
Add Stop event	Shift+Ctrl+S	
Add Stop Cue event	Shift+Ctrl+C	
Add Wait event	Shift+Ctrl+W	
Add Logo Off	Shift+Ctrl+O	
Add GPI event	Shift+Ctrl+G	
Add KramerSwitch output	Shift+Ctrl+K	
Add Kramer Matrix output	Shift+Ctrl+R	
Add TitleBoxNetCtrl output	Shift+Ctrl+T	
Add Note event	Shift+Ctrl+N	
Activate logo preset	Ctrl+ Num	
Jump to	Ctrl+Shift+Num	
Show Big timer	Ctrl+B	
Show Clip timer	Ctrl+T	
Show Block timer	Ctrl+K	
Show Log	Ctrl+L	
Show logo panel	Ctrl+G	
Show TC timer	Ctrl+D	

Channel in a Box User Manual

PLAYBACK CONTROL

Function	Shortcut 1	Shortcut 2
Play	Space	Alt+P
Scheduled Play	Alt+Y	
Stop	Double Esc	
Pause/Resume	Space	
Next	Ctrl+Space	Alt+N
Jump	Alt+Space	Alt+J
Reset hardware	Ctrl+Backspace	
Cue	Ctrl+Alt+Space	<F12>

VOLUME CONTROL

Function	Shortcut 1	Shortcut 2
Volume Up (fast)	Num +	
Volume Down (fast)	Num -	
Volume Up (precise)	Shift+Num +	
Volume Down (precise)	Shift+Num -	

GRID CONTROL (WHEN SELECTED)

Function	Shortcut 1	Shortcut 2
Page scrolling	PageUp/PageDown	
Jump to grid beginning	Home	
Jump to grid end	End	
Jump to line Num	Ctrl+Shift+Num	
Clip move (order change)	Shift+Up/down	Drag&drop
Clip copy	Ctrl+Drag&drop	
Scroll to playing clip	Click on the green counter	

APPENDIX 11 – LISTBOX KEYBOARD SHORTCUTS

GENERAL

Function	Shortcut
New playlist	Ctrl+N
Load playlist	Ctrl+O
Save playlist	Ctrl+S
Save playlist As	Alt+S
Print playlist	Ctrl+S
Append a clip at playlist end	Alt+Ins
Insert a clip before selected clip	Ins
Remove a clip or clips	Del
Remove a clip without confirmation	Shift+Del
Clear All	Alt+Del
Clear All without confirmation	Shift+Alt+Del
Edit MPEG Media	Alt+Enter / Double-click

Find text	Shift+Ctrl+F

Undo/Redo last action	Ctrl+Z / Ctrl+Y
Insert Stop event	Ctrl+Alt+S
Insert Stop Cue event	Ctrl+Alt+C
Insert Wait event	Ctrl+Alt+W
Insert Wait Until event	Ctrl+Alt+U
Insert Wait Time Code event	Ctrl+Alt+A
Insert Logo event	Ctrl+Alt+L
Insert GPI event	Ctrl+Alt+G
Insert KramerSwitcher event	Ctrl+Alt+K
Insert KramerMatrix output	Ctrl+Alt+R
Insert TitleBoxNetCtrl output	Ctrl+Alt+T
Add Stop event	Shift+Ctrl+S
Add Stop Cue event	Shift+Ctrl+C
Add Wait event	Shift+Ctrl+W
Add Wait Until event	Shift+Ctrl+U
Add Wait Time Code event	Shift+Ctrl+A
Add Logo event	Shift+Ctrl+L
Add GPI event	Shift+Ctrl+G

Channel in a Box User Manual

Add KramerSwitcher event	Shift+Ctrl+K
Add KramerMatrix output	Shift+Ctrl+R
Add TitleBoxNetCtrl output	Shift+Ctrl+T
Insert Note event	Ctrl+Alt+N
Add Note event	Shift+Ctrl+N

GRID CONTROL

Function	Shortcut 1
Page scrolling	PageUp/PageDown
Jump to grid beginning	Home
Jump to grid end	End

SELECTION MANAGEMENT

Function	Shortcut 1
Selection bar movement	Up/Down arrows/ Click
Select multiple sequential clips	Shift+Up/Down arrows
Select more clips out of the sequence	Ctrl+Click
Select All	Ctrl+A
Invert selection	Shift+Ctrl+I

Cut selection	Ctrl+X
Copy selection	Ctrl+C
Paste selection	Ctrl+V
Move multiple selection	Alt+Drag-n-drop
Duplicate selection	Ctrl+Drag-n-drop
Move to playlist beginning	Alt+Home
Move selection 1 line up	Alt+Up
Move selection 1 line down	Alt+Down
Move selection to playlist end	Alt+End
Shift clips into selection range	Shift+Ctrl+H
Reverse clips order	Shift+Ctrl+V
Randomize clips order	Shift+Ctrl+R

APPENDIX 12 – CAPTUREBOX KEYBOARD SHORTCUTS

Function	Shortcut
Setup	Alt+T
Compensation	Alt+N
In field	Alt+I
Out field	Alt+O
Duration field	Alt+D
File Name field	Alt+F
Add to List	Alt+L
Sync	Alt+Y
Sort	Alt+S
Capture	Alt+R
Edit	Alt+E
Finish	Alt+F
Auto Capture	Alt+U
Manual Capture	Alt+M

Channel in a Box User Manual

RS-422 Controller

Deactivation	Ins
Play	Up arrow
Still	Down arrow
Stop	Spacebar
FF	Right arrow
REW	Left arrow
Seek FF	Hold right arrow
Seek REW	Hold left arrow
Frame Advance	Ctrl+ right arrow
Frame Reverse	Ctrl+ left arrow

APPENDIX 13 – TITLEBOX KEYBOARD SHORTCUTS

Function	Shortcut
Open project	Ctrl+O
Save project	Ctrl+S
Exit TitleBox	Alt+F4
Undo	Ctrl+Z
Cut selection	Ctrl+X
Copy selection	Ctrl+C
Paste selection	Ctrl+V
Delete	Del
Select All	Ctrl+A
Add to Schedule	Ctrl+H
Copy schedule settings	Ctrl+Alt+C
Paste schedule settings	Ctrl+Alt+V

PROPERTY TOOL

Add point	A
Align to the previous point	Alt+left arrow
Align to the next point	Alt +right arrow
Align to the first point	Alt +down arrow
Move to the next point	Ctrl+ left arrow
Move to the previous point	Ctrl+ right arrow
Movement into the object graphics	Left/right/up/down arr.

APPENDIX 14 – CLIP TRIMMER KEYBOARD SHORTCUTS

Function	Shortcut
PLAY / PAUSE / RESUME	Space
Play Selection	Shift + Space
Close Trimmer	Esc
Mark In	I
Mark Out	O
Get Scene / Annotation	Ctrl+S
Export Frame as file	Ctrl+E
Add Zone	Ctrl+Z
Add Marker in Zone	Ctrl+M
Split	R
Reset	Ctrl+BkSp
Delete clip section	Ctrl+D
Lock/Unlock In point	Ctrl+Alt+I
Lock/Unlock Out point	Ctrl+Alt+O
Lock/Unlock Duration	Ctrl+Alt+U

Go to In Point	Ctrl+I
Go to Previous Mark Point	Ctrl+Left
Go to Out Point	Ctrl+O
Go to Next Mark Point	Ctrl+Right
Go to Beginning	Home
Go to End	End
Go to Next Zone	Ctrl+Alt+Right
Go to Previous Zone	Ctrl+Alt+Left
Select Next Clip section	Alt+Right
Select Previous Clip section	Alt+Left
Go 1 frame earlier	Left
Go 1 second earlier	Shift+Left / Up
Go 10 seconds earlier	PageUp
Go 1 frame later	Right
Go 1 second later	Shift+Right / Down
Go 10 seconds later	PageDn
Goto	Type numbers and press Enter

Index

Channel in a Box User Manual

PlayBox Technology UK Ltd
Brookmans Park Teleport
Great North Road
AL96NE Hatfield
United Kingdom

info@playboxtechnology.com

www.playboxtechnology.com