

*Focus*  
*Virtual Studio*

***HotActions***

**User's Guide**

Revision as of October 6, 2010

VS 1.63

Copyright © SoftLab-NSK Ltd.

# Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
<b>2</b>	<b>CHECKING THE STUDIO AVAILABILITY .....</b>	<b>5</b>
2.1	Switching the Studio Components and Connecting External Equipment .....	5
2.2	Checking the <i>VGA</i> Settings and Launching <i>HotActions</i> .....	5
2.3	Video Settings .....	6
2.4	Audio Settings .....	11
2.5	Working with Sample Projects .....	12
<b>3</b>	<b>THE <i>HOTACTIONS</i> APPLICATION .....</b>	<b>14</b>
<b>3.1</b>	<b>The Basic Notions of <i>HotActions</i> .....</b>	<b>14</b>
3.1.1	What is a <i>Scene</i> ? .....	14
3.1.2	What is a <i>Project</i> ? .....	14
3.1.3	What is an <i>Action</i> ? .....	15
3.1.4	What is an <i>Action Library</i> ? .....	15
3.1.5	What are a <i>Hotbar</i> and a <i>Hotset</i> ? .....	16
3.1.6	What is a <i>Theme</i> ? .....	16
3.1.7	Working with Files .....	17
3.1.8	The <i>Properties</i> Dialog Box .....	18
3.1.9	The <i>Edit</i> and <i>LiveAction</i> Modes of Operation .....	18
3.1.10	Using Video Streams .....	19
<b>3.2</b>	<b>The Main Application Window .....</b>	<b>20</b>
3.2.1	Toolbars .....	20
3.2.2	The <i>Standard</i> Toolbar .....	20
3.2.3	The <i>General</i> Toolbar .....	21
3.2.4	The <i>LiveAction</i> Toolbar .....	21
3.2.5	<i>Project</i> Bar .....	22
3.2.6	Menus .....	22
<b>3.3</b>	<b>Editing <i>Actions</i> and their Properties .....</b>	<b>23</b>
3.3.1	The <i>Action</i> Tab of the <i>Properties</i> Dialog Box .....	23
3.3.2	The <i>Icon</i> Tab of the <i>Properties</i> Dialog Box .....	24
<b>3.4</b>	<b>Operations with the 3D Scene .....</b>	<b>25</b>
3.4.1	Operations with the Scene File .....	25
3.4.2	Symbols Used for the Scene Objects .....	26
3.4.3	The Scene Window Toolbar and Presentation of the Scene Objects according to Types .....	26
3.4.4	The <i>HotActions</i> Main Window Menu When Working with the Scene .....	29
3.4.5	The <i>Properties</i> Dialog Box for Different Types of the Scene Objects .....	29
3.4.6	The Function of <i>Startup Actions</i> .....	33
3.4.7	Creating <i>Startup Actions</i> .....	33
<b>3.5</b>	<b>Operations with the <i>Action Library</i> .....</b>	<b>34</b>
3.5.1	Operations with the Library File .....	34
3.5.2	Operations with <i>Actions</i> in the Library .....	35
3.5.3	The Main Window Menu When Working with an <i>Action Library</i> .....	36
<b>3.6</b>	<b>Operations with <i>Hotsets</i> and <i>HotBars</i> .....</b>	<b>37</b>
3.6.1	Operations with the <i>Hotset</i> File .....	37
3.6.2	Manipulating <i>HotBars</i> and their Buttons in the <i>Hotset</i> .....	38
3.6.3	Operations with the <i>HotBar</i> Buttons Properties .....	38

3.6.4	Using <i>Snapshots</i> for <i>HotBar</i> Buttons .....	40
3.6.5	The Context Menu and <i>Properties</i> Dialog Box when Working with a <i>HotBar</i> in the <i>Hotset</i> .....	40
3.6.6	An Alternative Document-Like Presentation of the <i>Hotset</i> .....	41
3.6.7	The Main Window Menu When Working with the <i>Hotset</i> .....	43
<b>3.7</b>	<b>Operations with the <i>Project</i></b> .....	<b>43</b>
3.7.1	Operations with the Project File.....	43
3.7.2	Working with Files and File Groups in the Project .....	45
3.7.3	The Main Window Menu When Working with the Project .....	48
<b>3.8</b>	<b>Operations with the <i>Theme</i></b> .....	<b>48</b>
3.8.1	Operations with the File of a <i>Theme</i> .....	48
3.8.2	Operations with the Theme Variables.....	49
<b>3.9</b>	<b>Operations with Text Documents</b> .....	<b>52</b>
<b>4</b>	<b>CONTROLLING THE APPLICATION MODES OF OPERATION</b> .....	<b>54</b>
<b>5</b>	<b>USING THE JOYSTICK AND MOUSE TO CONTROL OBJECTS</b> .....	<b>55</b>
5.1	Initializing the Joystick.....	55
5.2	Selecting a Manipulation Object in the <i>Edit Mode</i> .....	55
<b>6</b>	<b>USING THE <i>DEBUG OUTPUT</i> WINDOW</b> .....	<b>57</b>
<b>7</b>	<b>THE <i>RENDER OUTPUT</i> WINDOW</b> .....	<b>59</b>
7.1	The Commands of the Main Application Window Menus for Working with the <i>Render Output</i> Window .....	59
7.2	The Menu Commands of the <i>Render Output</i> Window.....	60
<b>8</b>	<b>DIALOG BOXES FOR DISPLAY SETTINGS</b> .....	<b>62</b>
<b>9</b>	<b>THE <i>OPTIONS</i> DIALOG BOX: THE VIRTUAL STUDIO GENERAL SETTINGS</b> .....	<b>64</b>
9.1	The <i>LiveAction Mode</i> Settings .....	64
9.2	The <i>Default Directories</i> Settings.....	66
9.3	The <i>Debug Output</i> Settings .....	68
9.4	The <i>Language</i> Tab .....	69
<b>10</b>	<b>TROUBLESHOOTING</b> .....	<b>71</b>
10.1	Problems with Default Directories.....	71
10.2	Problems with the <i>FD300</i> Board(s) .....	71
10.2.1	One or Several Boards are not Found in the System .....	71
10.3	Problems with Output Video .....	72
10.3.1	Absent <i>Render Output</i> Window .....	72
10.4	Technical Support Service .....	72
<b>11</b>	<b>CONNECTING VIDEO SIGNALS AND CONFIGURING HD VIRTUAL STUDIO</b> .....	<b>73</b>
11.1	The Studio Components Connection Scheme .....	73
11.2	Setting the Video Card to Work in a Configuration with Two Monitors .....	73
<b>12</b>	<b>APPENDIX</b> .....	<b>76</b>
12.1	Cable Sets for the <i>FD300</i> Boards .....	76
12.2	The Passive Breakout Box (TF782) .....	77
12.3	The Active Breakout Box (TF785 M) .....	78
12.4	Breakout Boxes for a System with Several <i>FD300</i> Boards (TF783 A, TF783 B, TF783 C).....	79

# 1 Introduction

Thank you for purchasing our product!

Your system is completely configured and ready for use. This document provides full information on the functionality of this tool for creating television production.

The *Focus* virtual studio workflow for television production can be divided into two basic stages: preliminary work and camera shooting proper in the **LiveAction** mode. Typically, the preliminary work stage of television production using the *Focus* virtual studio includes the following steps:

- Developing a scenario for the TV show or a video clip and selecting the necessary video and audio materials, design elements, etc.
- Creating and exporting a 3D scene or a set of scenes (the virtual scenery, required animation of objects and virtual cameras) via a 3D editor such as *3D Studio Max* or *Maya*. For an example of creating and exporting such a scene, see *Creating 3D Scenes. User's Guide*, it being recommended that the guide be examined on the whole beforehand.
- Creating a project containing ready 3D scenes in *HotActions*. The major portion of this document covers the *HotActions* application.
- Creating interface elements for controlling the studio in the **LiveAction** mode of *HotActions*. For example, animation of the objects or virtual cameras can be executed by pressing the assigned buttons or keys. For this purpose a special *HotActions* script language is used. *HotActions* itself is an environment of interactive control with the ability of flexible configuration. Sections starting from 3.1 cover the script language means. The script language is described in a separate document.
- Configuring input and output video and audio parameters. Video and audio settings are described in separate corresponding documents. The output video parameters, such as television standard (PAL or NTSC) and the output signal type, should be correctly adjusted by each user. If not required to change the studio operation mode in the future, usually there is no need to adjust the output video parameters again. The adjustment of the input video parameters is usually required after changes of the external equipment configuration. Chroma keying (rear projection) parameters may need to be adjusted, for example, if the illumination in the shooting studio has changed. Examples of configuring the input and output video are covered in Section 2.3.

Once the projects are ready and the input and output video is adjusted, it is possible to begin the performance of the planned TV show scenario. The shooting is performed in the **LiveAction** mode of *HotActions*. Any necessary action or adjusting parameters can be executed directly during the shooting either automatically according to the script, or interactively, under the operator's control. The **LiveAction** mode is also used during preparation for television production for the purpose of checking the availability of the created virtual scenery control interface.

Section 2 briefly describes a typical procedure of adjusting the system and gives a short compendium and information necessary to start the studio and achieve first results.

The next Sections include the reference guides for all the software components of the system, which fully describe the interfaces and functional features of the current *Focus* virtual studio version.

Some typical problems and solutions to them are discussed in Section 10. We also recommend that you regularly visit our Web and FTP sites (see Section 10.4), where information about software and documentation updates, FAQ and comments are published.

## 2 Checking the Studio Availability

This Section describes how to check the the *Focus* virtual studio operability and gives the essential principles of working with the *HotActions* application. The next Sections describe the application in detail. The video and audio settings are discussed in separate documents.

### 2.1 Switching the Studio Components and Connecting External Equipment

First of all, assemble the studio components in their correct positions. Connect the corresponding switching devices to all the input/output *FD300* boards, whose connectors are on the studio unit back panel. These may be special cable sets or *Breakout Boxes* described in Section 12.

Depending on the configuration, the studio unit may have one or more input/output *FD300* boards installed, but usually one of them only is used to output the resultant video and audio.

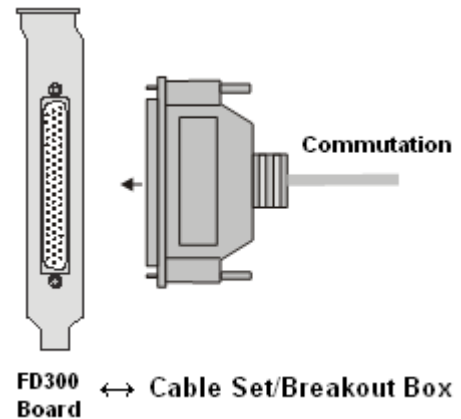
In studios that work with HD video signals, an *FD300* board is only necessary to work with audio signals (adjusting parameters). The studio resultant video image is output in such configurations through the hardware module *DVM62* (*Digital Virtual Machine*). In this case, connect the video card DVI connector with the module DVI connector, and when the computer is on, adjust the second video adapter (*NVIDIA GeForce*) operation. Adjusting such configurations is discussed in Section 11.

When working with analog video signals (component YUV or RGB, S-Video, or composite), video and audio signals are supplied from the *FD300* board through the cable set or breakout box (see Section 12, Appendix) to a TV monitor to output the resultant image. The input video sources such as video cameras and video recorders are also connected to the breakout boxes or cables.

For a studio configured to work with digital signals (SDI), the studio unit back panel must also have connectors to switch such signals.

The computer monitor, keyboard and mouse are connected in a standard way.

After making sure the connections are correct, you can switch on the computer.



### 2.2 Checking the VGA Settings and Launching *HotActions*

To work with 3D graphics, the virtual studio must have the anisotropic filtering and graphics card antialiasing modes adjusted.

Only the *NVIDIA GeForce* graphics video adapter included in the studio configuration provides the studio correct operation. To check the video card settings, open the *NVIDIA Control Panel* dialog box using the **Start** menu: **Start > Settings > Control Panel > NVIDIA Control Panel**.

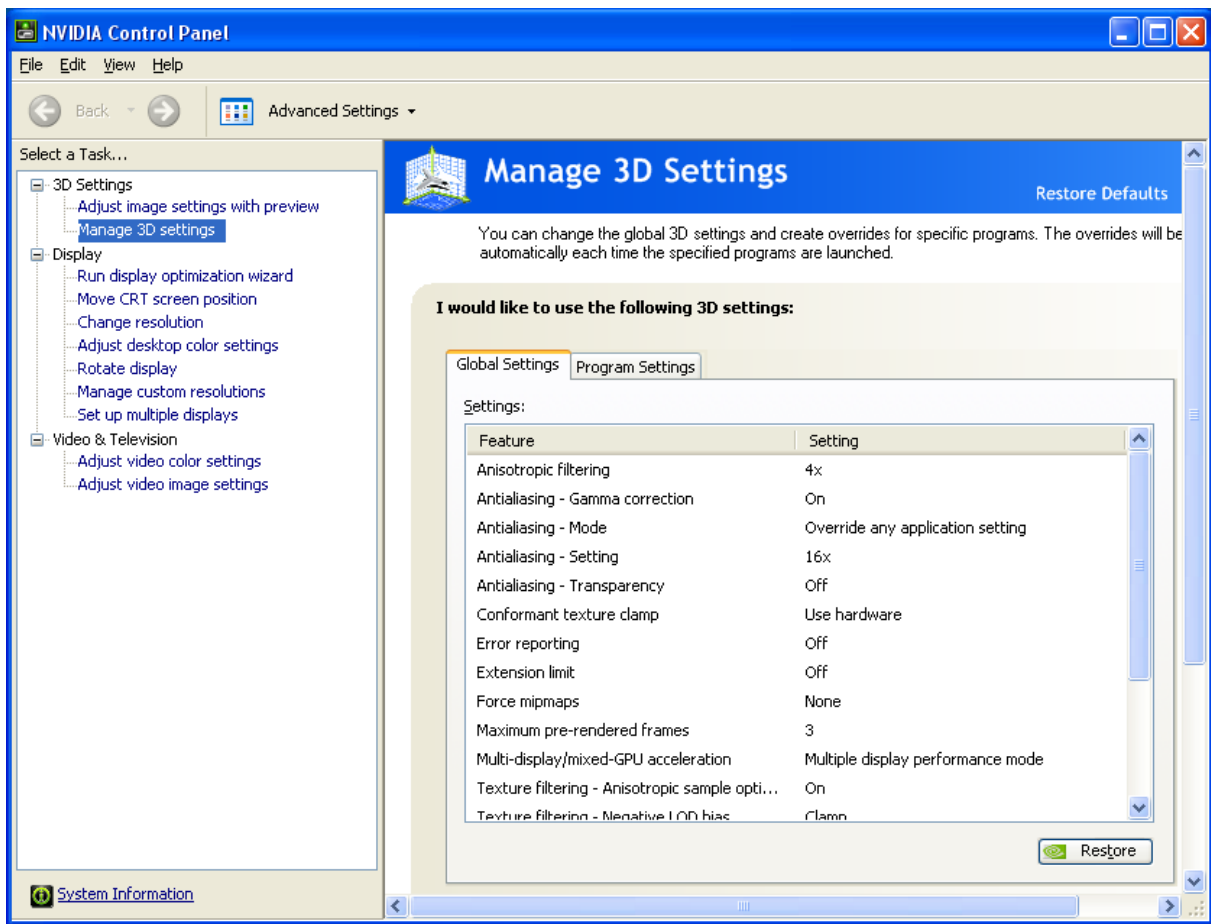


Figure 1. Setting the video card modes for working with 3D graphics.

Select **4x** from the **Anisotropic filtering** drop-down list in the **Manage 3D Settings** page of the **3D Settings** group. Select **16x** from the **Antialiasing – Setting** drop-down list. To make the mode available, previously select **Override any application setting** from the **Antialiasing – Mode** drop-down list.

After the *NVIDIA* settings are verified, proceed to setting video signals in the Virtual Studio.

## 2.3 Video Settings

The video settings are discussed in detail in a special document on setting video and audio (*Video and Sound Settings*). Only the basic adjustment steps are discussed here.

When working in the Studio with analog or digital (SDI) video signals, previously determine some parameters of theirs in the *FDConfiguration* application (Figure 2). The application is discussed in *FD300 configuration. User's Guide*.

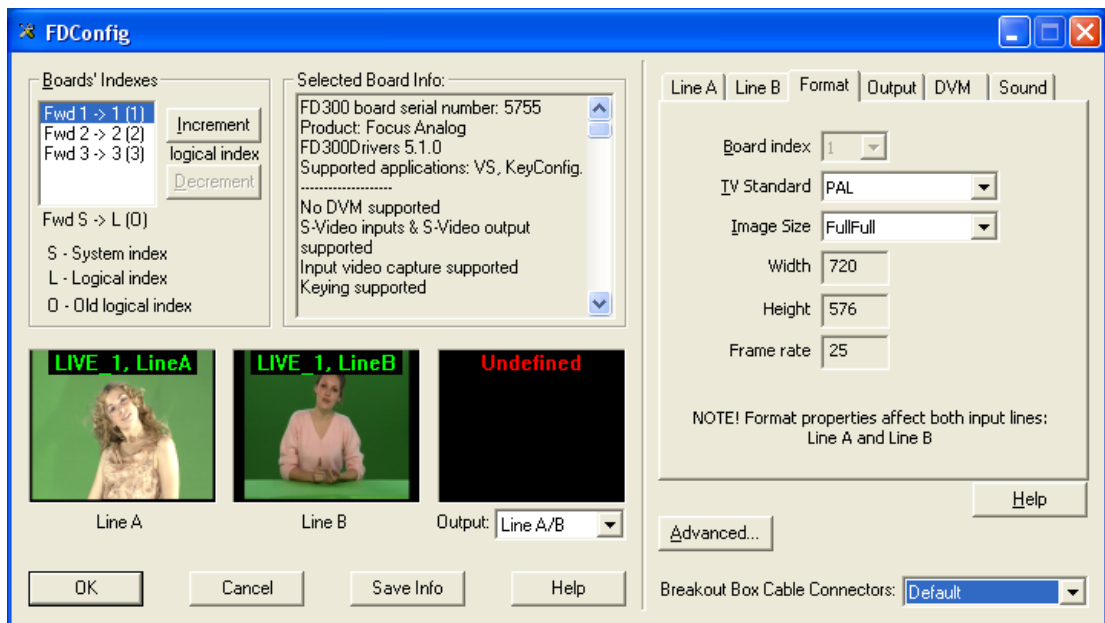


Figure 2. Presetting video signal parameters in the FDConfiguration application

For all the FD300 boards indicated under **Boards'Indexes** as **Fwd 1**, **Fwd2** and so on, select a TV standard for the operational signals supplied to the board (PAL, NTSC, or other) from the **TV Standard** drop-down list on the **Format** tab.

Click the **Advanced...** button (Figure 2). The *Advanced Settings* dialog box opens (Figure 3). For each board with connected video sources, enable the **A/V Capture Filter** option under **Use Direct Show filters**. Since operation with video signals in the Virtual Studio is performed through *Direct Show* filters, it is necessary that their operation be enabled in the settings of the FD300 boards to which video signals are supplied.

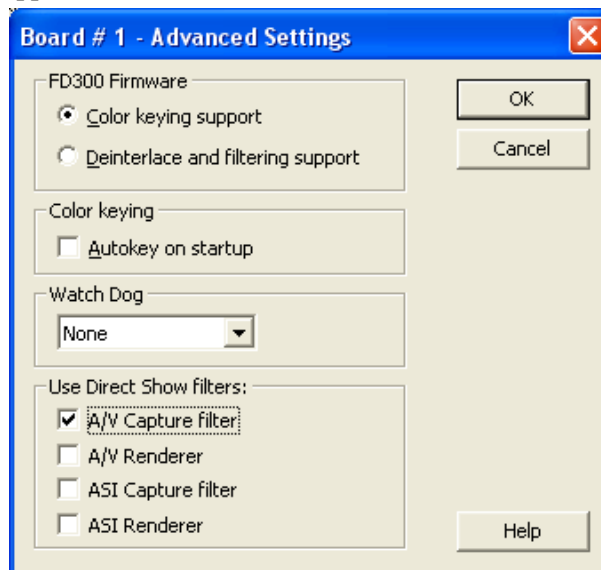


Figure 3. Enabling work with video signals on the FD300 board through *Direct Show* filters.

To output the video signal of the resultant image, the FD300 board having the largest logical index **L** is used in the Studio. Usually it matches the system index **S**. For this board, select an analog video signal type from the **Analog Video Output** drop-down list on the **Output** tab. Usually, it is Y/C+CVBS (S-Video signal type) or YUV+CVBS (component). The default is ColorBar+CVBS: color-bar test pattern output.

The TV format of the Studio output video signal is determined by the output board format selected from the **TV Standard** drop-down list on the **Format** tab.

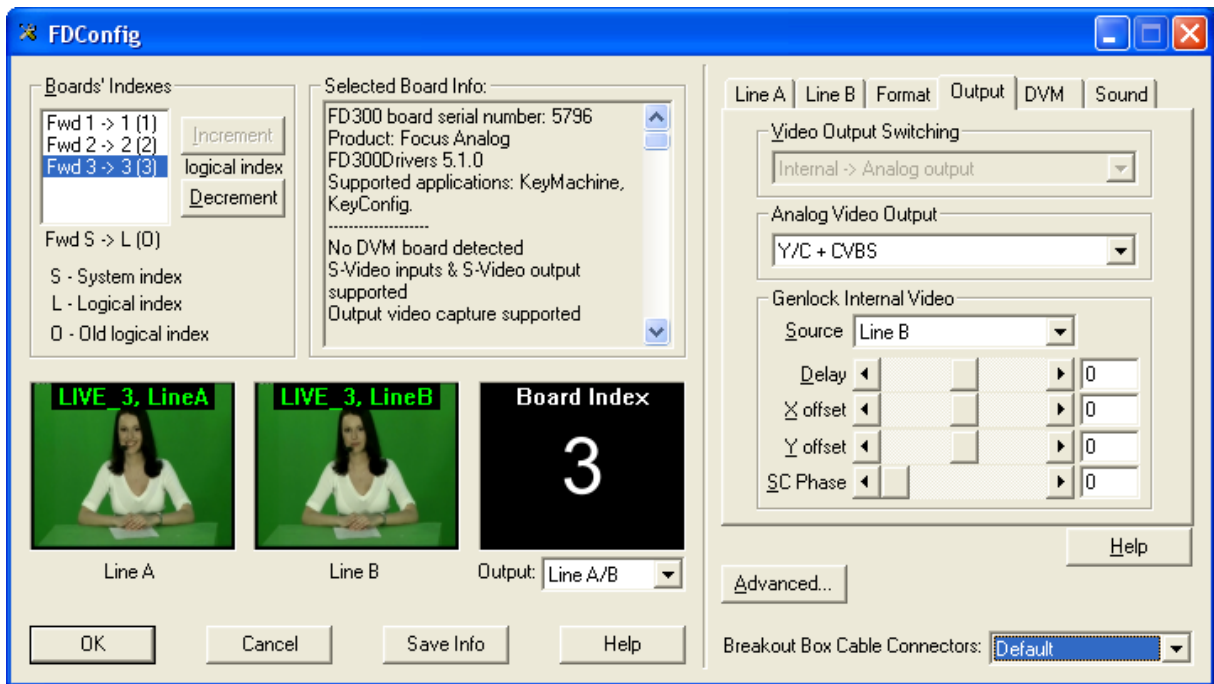



Figure 4. Setting the Studio output video signal in the *FDConfiguration* application.

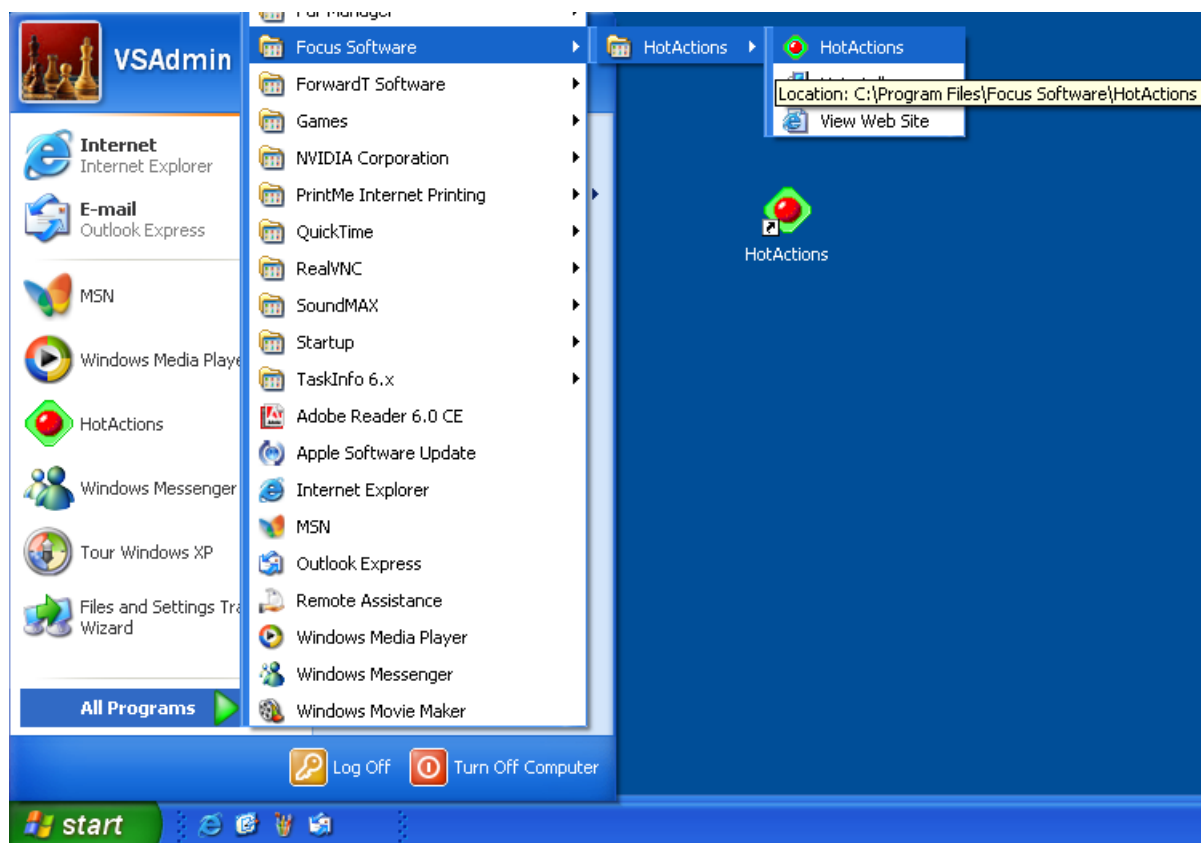
If the Studio video signal output board is to operate as the *Genlock* option (synchronization of the output signal to the external source), select the board line to which the synchronization signal is supplied from the **Source** drop-down list under **Genlock Internal Video**: Line A or Line B. The indices of the video signals connected to the board are determined in the **Input** drop-down list on the corresponding tabs **Line A** or **Line B** of the *FDConfiguration* application.

Absent the requirement of synchronization of the output video signal to the external video source, select for the output board the mode of its internal synchronization **Master** from the **Source** drop-down list under **Genlock Internal Video**.

After presetting parameters of the FD300 boards for working with video signals, proceed to mastering the main application of the Virtual Studio - *HotActions*. To launch the application, use the **Start** menu (Figure 5) or double-click the **HotActions**  icon on the *Windows* desktop.



Before launching the application in an HD studio (Section 11), plug the HASP dongle into the computer USB port (to have support for the HASP dongle in the operating system, use the HASPUserSetup.exe installer). Otherwise a warning message to have the protection device plugged in is displayed when launching the application.





**Figure 5. Launching the main application of the Virtual Studio.**

The *HotActions* application is represented by the same-name window to control the studio and the *Render Output* window (Section 7). If the *Render Output* window is not displayed on the screen, you can open it by using the **Render Window** command from the **View** menu (Section 3.2.6). The **Center Render Window** command from the **Window** of the *HotActions* main window locates the *Render Output* window in the center of the main window.

Setting the input video signals in the *HotActions* application is performed on the **Input Video Options** tab of the *Video Configure* dialog box. To open the dialog box, click the **Live Video Configure**  button of the *General* toolbar (Section 3.2.3) or  button of the *Live Action* toolbar (Section 3.2.4), or use the **Video Configure...** command from the **Tools** menu, or press F9 on the keyboard.

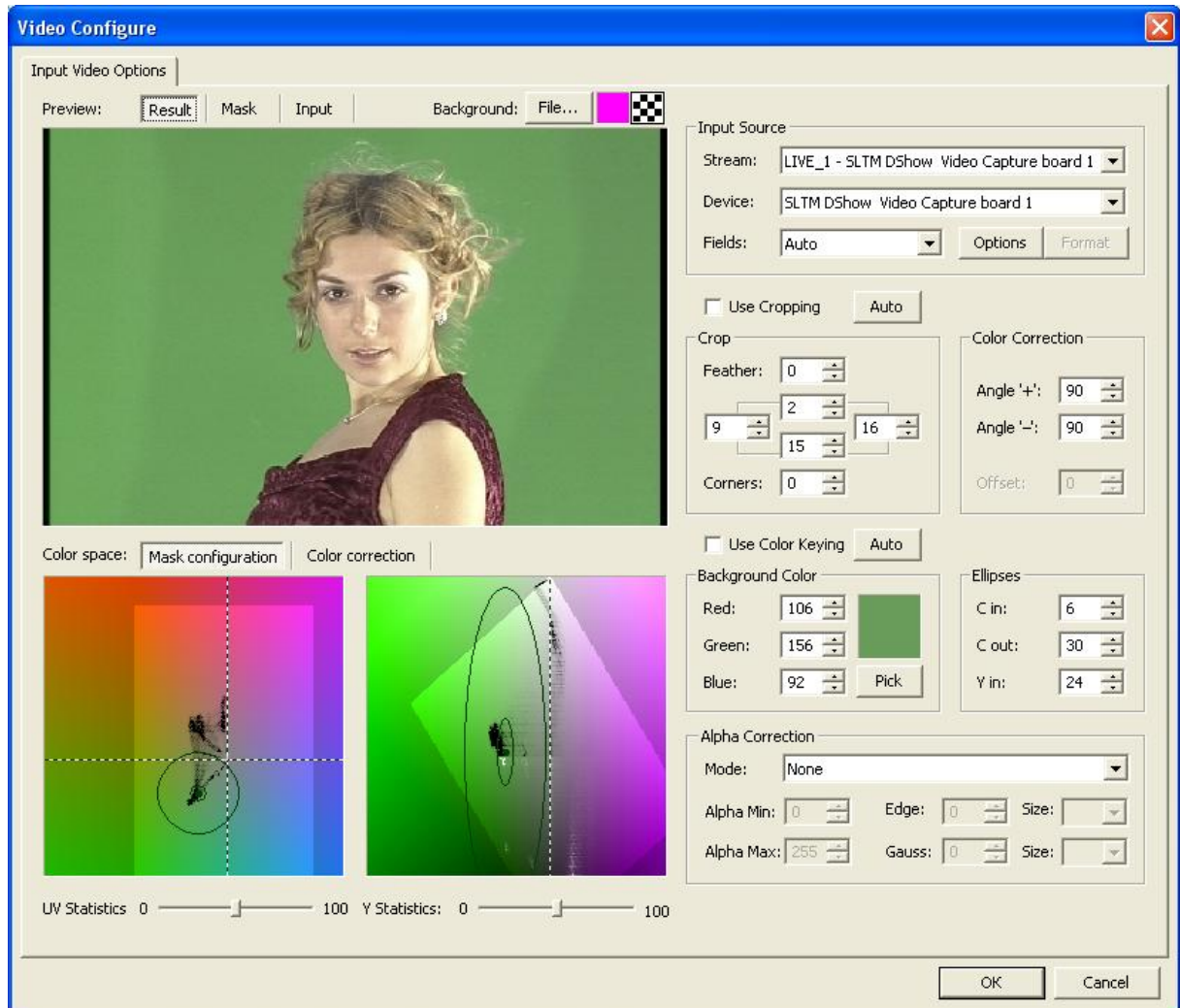


Figure 6. Dialog box for setting input video signals in the *HotActions* application.

From the **Stream** drop-down list under **Input Source**, select a video stream (LIVE\_1, LIVE\_2, and so on) to show the video signal connected to the video source input board. Assign an input device – one of the board channels (*FD300* or *Aja XENA*) from the **Device** drop-down list to the selected video stream. Since each board has two input channels, there are two options in the drop-down list for each board: *SLTM DShow Video Capture board 1* and *SLTM DShow Video Line B Capture board 1* – the *FD300* input channels, or *AJA Capture Filter board 0 channel 1* and *AJA Capture Filter board 0 channel 2* for *Aja XENA*.

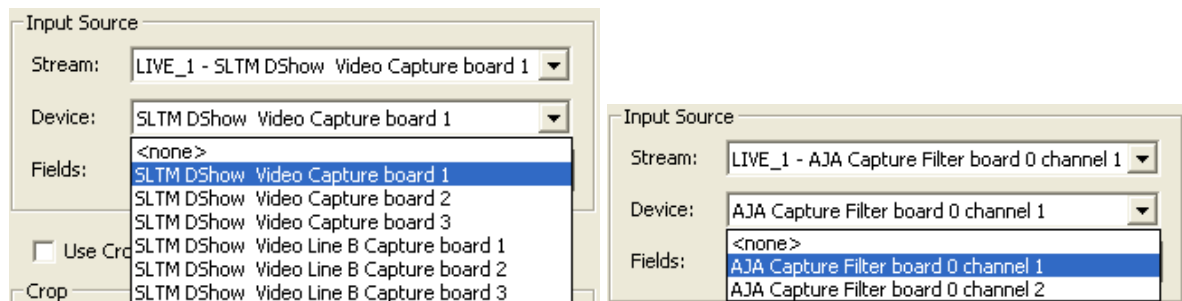


Figure 7. Drop-down list of the connected video signal input devices.

There are only the input channels for the *Aja XENA* boards (HD stream input devices) in the drop-down list.

When an input channel for the video stream is selected, you can change the field order for the video signal (when necessary) in the **Fields** drop-down list. Or you can change the video signal. To do so,

click the **Options** button. The **Properties** dialog box opens. Select another video signal index from the **Input** drop-down list on the **Input** tab.

Use the **Crop** area to set the frame cropping values when displaying – the nondisplayed offsets: top, bottom, left and right offsets respectively. The **Feather** parameter value determines the smoothness of transition from nondisplayed parts of the frame to displayed. The **Corners** parameter value determines the corner cropping. Clicking **Auto** enables the cropping offsets self-adjustment. Checking  **Use Cropping** enables the area settings; unchecking the checkbox disables all the frame croppings.

To set the input video signal chroma key (rear projection) parameters, check the  **Use Color Keying** check box. When the background quality is good, lighting being enough, automatically setting this area parameters usually gives good results.

The dialog box parameters are described in detail in *Video and Sound Settings*.

Setting the resultant image video signal output is performed in the **Render Options** dialog box (Figure 8). The dialog box is described in detail in Section 8. To open the dialog box, press F11 or select the **Render** command in the **Options** menu of the *Render Output* window. When rendering video image by the *FD300* board, select from the **Output Device** drop-down list under **General** the *FD300* board with the number matching the logical index **L** specified in the *FDConfiguration* application (Figure 4).

When working with HD video signals (Section 11), select the output *Display2: DVM62* (as the video signal output device) from the **Output Device** drop-down list in the **Render Options** dialog box.

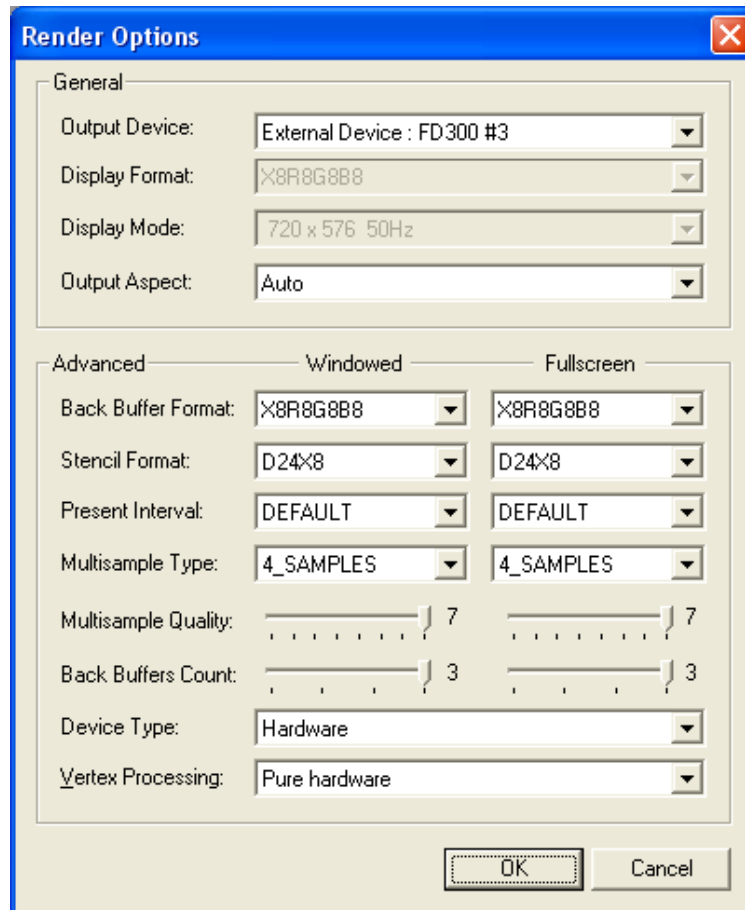
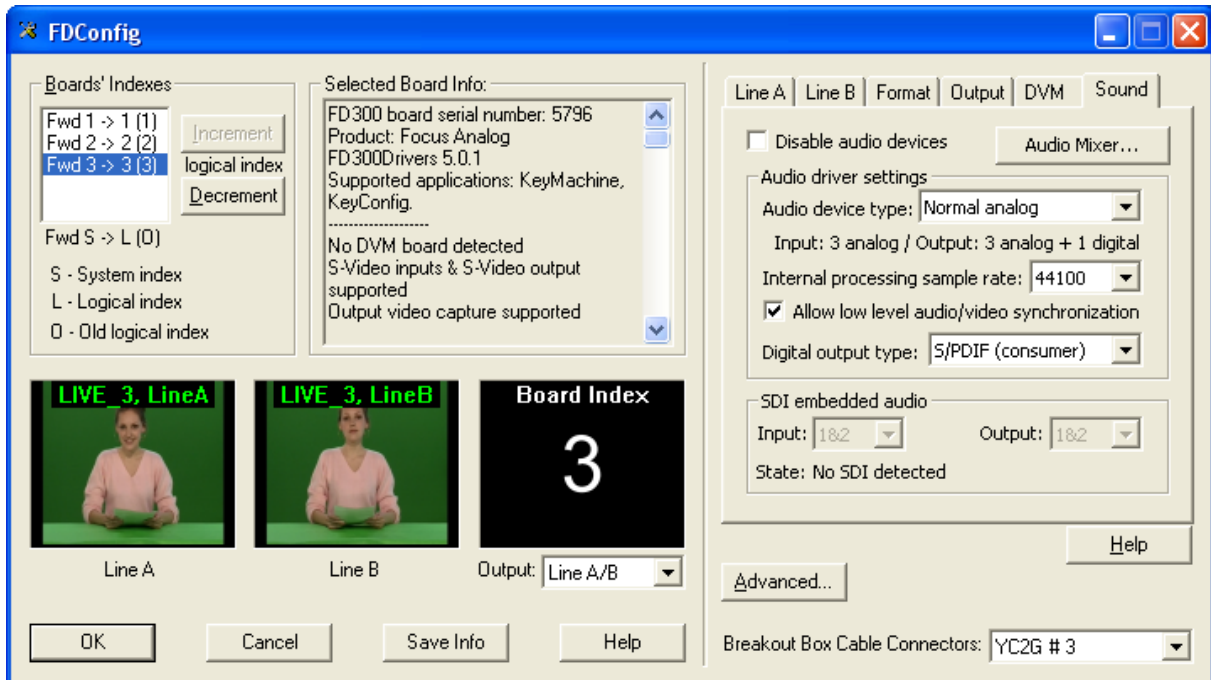


Figure 8. The Studio resultant video image output settings dialog box.



## 2.4 Audio Settings

External audio equipment is connected to the *FD300* board connectors via a special cable set (Section 12.1) or breakout box (Sections 12.2, 12.3, 12.4). All the audio played in *HotActions*, e.g. various WAV or MP3 files started by the script commands are output via the sound card which is set in the operating system as default. To combine the audio stream with audio from a microphone or other sources connected to the switching devices, select one of the *FD300* audio channels as the output

device in the operating system settings (**Start -> Settings -> Control Panel -> Sounds and Audio Devices -> Audio, Default device** drop-down list under **Sound playback**). Usually it is an output device of the output board with the largest logical index **L** in the *FD300 Configuration* application (Figure 4). The **Disable audio devices** option on the **Sound** tab in the *FDConfiguration* application must be previously unchecked.



**Figure 9.** The tab for setting sound on the output board in the *FDConfiguration* application.

The required mode of sound mixing in the Studio (the *HotActions* application) is set in the **Sound Configure** dialog box. To open the dialog box, click  or , or use the **Sound Configure** command from the **Tools...** menu (see also *Sound Configure. User's Guide*).

## 2.5 Working with Sample Projects

When all required input and output video settings are adjusted, you can proceed to further acquaintance with the Studio capabilities.

This Section covers a sequence of operations when working with the virtual studio project files.

Select the **Open Project...** command in the **File** menu of the *HotActions* main window. The *Open* dialog box opens. Select one of the sample projects. Most often the sample projects folder is **D:\Focus\VS\_Samples**.

After the first run, the project name is added to the context menu of the **Recent Projects** command. You can use the menu to reopen the project.


Let us select a *Sport* project located on **D:\Focus\VS\_Samples\Sport\Sport.vsp**.



When loading a project, the arrangement of windows corresponding to the documents attached to the project is restored.

The list of the project documents sorted into groups is displayed in the project window named in this instance *Sport.vsp* (Figure 10). For more details about projects, see Section 3.1.2.

The information about the open scene: the number of mesh objects, polygons, textures, and video adapter memory value occupied by them is displayed in the *Debug Output* window (Section 6) when loading a project.

The window titled *Sport.hot* is used to directly control the scene. It is the *Hotset* containing *HotBars* – floating toolbars. Creating such toolbars and working with them are discussed in more detail in Sections 3.1.5 and 3.6.

To check the the *Focus* virtual studio availability, click the **Init All**  button initializing a scene. If the previous steps are performed correctly, presently the *Render Output* window displays the scene with a video image signal coming to the input (Figure 6).

Click the **Start LiveAction**  button. The button is located on the second from the left toolbar of the main window. The button is duplicated on a special toolbar (initially hidden) with large buttons which is shown and used when switching to the **LiveAction** mode. When the application operates in the mode, the **Start LiveAction** button changes its appearance to .

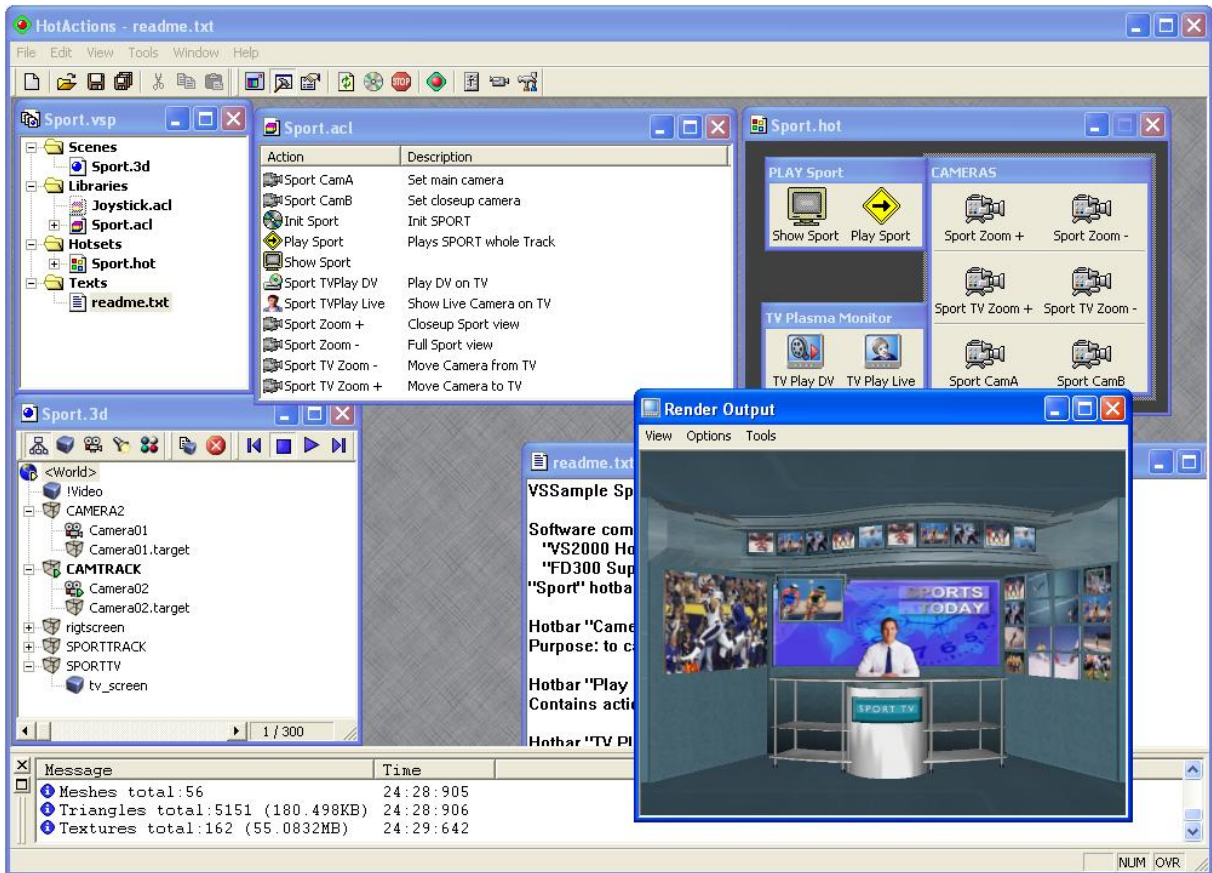



Figure 10. An example of the Studio operation: the *Sport* project loaded

In the **LiveAction** mode, the application switches to *Fullscreen* mode by default, and all the windows, except for the *Hotset* windows, are hidden or minimized. The settings concerning the **LiveAction** mode are discussed in Section 9.1.

Clicking the **Play Sport** button on the *Hotset* floating toolbar starts the scene animation playback (Figure 10). The names of the buttons executing analogous operations in other projects are usually formed according to the formula: **Play** + project name. The buttons of another toolbar of the project (**CAMERAS**) are designed to control cameras.

Clicking the  button executes switching from the **LiveAction** mode back to the editing mode. Note that in the editing mode, clicking a button of any toolbar only selects it but not operates. To test the operation of a button without switching to the **LiveAction** mode, click it with the **Alt** key pressed.

After the first test loading and a sample play back, you can explore the Studio operation in more detail reading the descriptions in the following Sections.

## 3 The *HotActions* Application

The *HotActions* application included in the *Focus* virtual studio software package is designed to create and configure an ergonomic environment for interactively controlling 3D graphics scenery along with video streams in real-time operating mode. Working with the application, the user is presented with the *HotActions* main window, which provides media for controlling the virtual studio, and the *Render Output* window displaying the created video images.

As was mentioned in the Introduction, the workflow when working with *HotActions* may be divided into three stages:

1. Creating commands for the modification and animation of 3D scene objects (changing their positions, textures, switching between virtual cameras, etc.), creating the corresponding interface elements for controlling studio in the **LiveAction** mode.
2. Preliminarily configuring the input and output video and audio signals in the **Render Options** dialog box (Figure 8), the **Input Video Options** tab of the **Video Configure** dialog box (Figure 6), the **Sound** tab of the **FDConfig** dialog box (Figure 9) of the application settings, directly from the *HotActions* program, this subject being covered in a separate document (*Video and Sound Settings*).
3. The shooting proper in the **LiveAction** mode using the interface elements created for performing the script.

### 3.1 The Basic Notions of *HotActions*

To present the general idea of how to work with the application, we describe the basic means of working in it: *Scene*, *Project*, *Action*, *Actions Library*, *Hotset*, *Hotbar*, *Theme*, *LiveAction*.

#### 3.1.1 What is a *Scene*?

A 3D scene is a set of virtual scenery (3D objects with or without animation, virtual cameras, light sources, etc.) created in a 3D editor such as *3D Studio Max* or *Maya*. A scene will be exported to the virtual studio and used there, which implies compliance with some requirements and upholding certain restrictions. That is why is recommended to previously examine *Creating 3D Scenes. User's Guide*. The document also contains a full-length example of exporting a scene and creating a project for it in the *HotActions* application.

The result of the export procedure is a **\*.3d** file (a scene document), which can be used in the *HotActions* application (see Section 3.4).

Rendering a scene into the initial (basic) state, loading, initializing it and naming its requisite components is performed via the dedicated initializing *Actions* (see Section 3.4.7).

#### 3.1.2 What is a *Project*?

General working with the studio in the *HotActions* application is organized by creating *Projects*.

A project file (\*.vsp) contains references to all the files (documents) used for work. It also contains information about the condition of files (open, hidden, or closed), windows and toolbars arrangement, and other settings, for example, the paths to the files included in the project.

Only one project can be open at the same time in the application. Closing the project window closes all the windows connected with it after a confirmation request to save files.

A window of any project contains *Standard Groups* – separate folders for each standard type of files, with which *HotActions* works. When files are added to a project, they are automatically saved in the appropriate folder/group. The user can also create *Custom Groups*, i.e., groups that are not assigned to any certain file type. They can contain any files used by the *HotActions* project, or, more precisely put, references to them united on some basis.

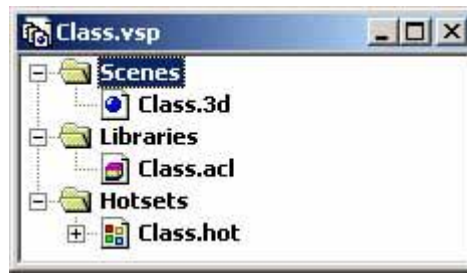


Figure 11. An example of an open project window

Context menus appearing at a right click on a group name or a file name in the group allow to work with groups and their contents. A detailed description of controlling a project see in Section 3.7.

### 3.1.3 What is an Action?

After a scene is created in a 3D editor, such as *3D Studio Max* or *Maya*, and exported (the procedures are described in detail in a special document), it can be used in the *HotActions* projects (see Section 3.4).

*Actions* with control commands are universal, easily modified and transported tools of interaction with a scene.

*Action* is one or several text commands. The result of its execution is some modification in a virtual scene. The concept of *Action* is basic in *HotActions*. *Actions*, in particular, include:

- loading of scenes files, textures, sound, etc.;
- switching of virtual cameras;
- start/stop of one or several objects animation playback;
- start/stop of the sound playback;
- change of the scene elements properties (for example, visibility, scale, position, color, etc.).

One *Action* can invoke other *Actions* in sequence or simultaneously, which makes scene sequences.

An *Action* execution is initiated by the user (the operator of the 3D scenery) with a click on a *Hotbar* button (see Section 3.1.5) or pressing a *Hotkey* on the keyboard in the process of interactively controlling the scenery (*LiveAction*).

The content of an *Action* is edited in the *Properties* dialog box (Figure 17). The execution of the *Action* commands can be tested from that very dialog box. For more detail about editing and debugging *Actions*, see Section 3.3. The regulations of creating *Actions* and the special language are described in *Using the Script Commands. User's Guide*. The initializing (*Startup*) *Actions* are a special group. They are sets of commands automatically executed when switching to the *LiveAction* mode. The task of *Startup Actions* is rendering the scene to the initial (basic) condition, loading and naming the necessary components. For information about creating *Startup Actions* see Section 3.4.7.

*Actions* are saved on the disk as text documents (*Actions Libraries*, see Section 3.1.4).

### 3.1.4 What is an Action Library?

An *Action Library* is a set (library) made up of several *Actions*, incorporated in one document (a file with the \*.acl extension).

As a rule, *Actions* are united in libraries by a common attribute, for example, they are assigned to the same 3D scene of a particular project. Thereby, there is only one *Action Library* in the majority of projects.

In most cases an *Action Library* is created by the scene designer/ animator, as they know the scene structure best of all, and/or by the program director (see Section 3.5 for information about working with an *Action Library*).

Since every *Action* is an executed script of commands, you can assign a certain button on the floating toolbar to any *Action*. To perform the operation in a new project, it is necessary to create at least one *Hotbar*.

### 3.1.5 What are a *Hotbar* and a *Hotset*?

*Hotbars* are button panels associated with a *Hotset* and directly used in the *LiveAction* mode as the basic interface for the scene control.

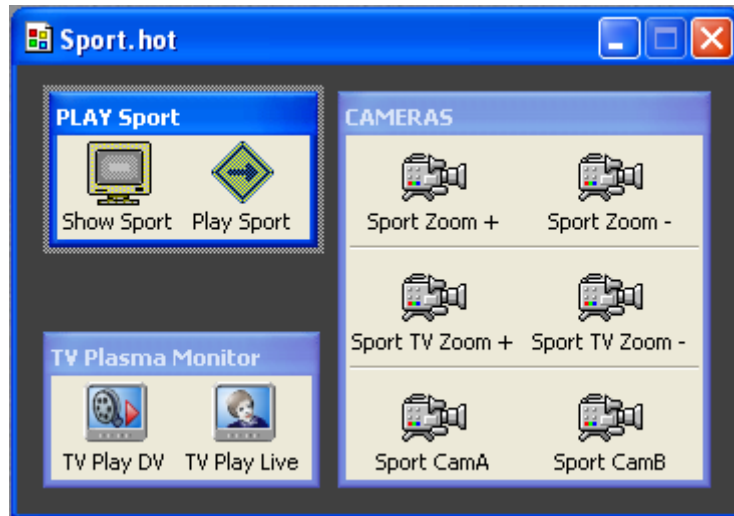


Figure 12. *Hotbars* in a *Hotset*: the basic tools in *HotActions*

*Hotset* is a set of *Hotbars*. A *Hotset* is handy in that all the *Hotbars* belonging to it can be hidden, shown, or moved simultaneously.

Each button of a *Hotbar* has a corresponding reference to a certain *Action*. By default their names are identical. It is possible to add buttons to any *Hotbar* by dragging and dropping *Actions* from *Action Libraries* directly onto the *Hotbar*. If an *Action* is simply put on an empty place of a *Hotset*, a new *Hotbar* is created. Working with *Hotsets* and *Hotbars* is discussed on the whole in Section 3.6, and Section 3.6.2 describes operations with the buttons of a *Hotbar*.

Operations with *Actions* in a library are discussed in Section 3.5.2. For details on editing *Actions* themselves, see Section 3.3.

### 3.1.6 What is a *Theme*?


*Theme* is a specific document type presenting a description of the project variables (parameters) used in its *Actions*. That allows the designer to vary the appearance and behavior of the project scenes by changing the values of the parameters forming the *Theme*.

The values of the parameters specified in a *Theme* are substituted when launching the *Action* in the following way: if a variable name is found in the *Action* (the **Variable Name** field in the **Variable** tab, Figure 59), it is replaced with its value (the **Value** field in the **Variable** tab, Figure 59). Thus, after loading a *Theme*, the new values of the variables are used when starting the *Action*.

It is possible to change a *Theme* in the *LiveAction* mode (see Section 3.1.9).

There are two basic types of *Themes*: **Standard** and **Custom**. In the **Theme** drop-down list of the **Project** toolbar (see Section 3.2.5) the names of **Standard** themes start with an asterisk (\*). The principal difference between the **Standard** and **Custom** themes is that in the *LiveAction* mode the user can edit **Custom** themes (change their variable values), while the **Standard** themes are available only for reading (loading). The user can also create a **Custom** theme on the basis of a **Standard** theme. Both types of themes are always saved in the corresponding folders of the project folder.

All the project themes are available to select in the *LiveAction* mode (see Section 3.1.9).

 A project can have only one theme loaded. If it is not specified explicitly, the application loads the default theme (**Default** in the **Theme** drop-down list, see Section 3.2.5), which is the last selected (saved) theme.

For more details on working with *Themes*, see Section 3.8.



### 3.1.7 Working with Files

*HotActions* works with the following file types:

- projects (\*.vsp files);
- scenes (\*.3d, \*.dxf, \*.wrl files);
- *Action libraries* (\*.acl files);
- *Hotsets* (\*.hot files);
- text documents (\*.txt, \*.ini, \*.inf files).

To work with all the files, it is necessary to create a **Project** in the *HotActions* application. A project file contains references to files of the other types and allows manipulating them, for example, open them simultaneously or individually, hide, etc. (see Section 3.7.2).

Usually all the files included in a project are opened right when opening the project. The existing projects are opened from the **File** menu of the main application window with the **Open Project...** command. After a project was opened for the first time, its name appears in the local menu of the **Recent Project** command and now the project can be opened by selecting its name in this local menu. A context menu of a required group allows adding an existing file to the current project.

A window of any project has separate folders for each type of files – these are *Standard Groups*. It is also possible to create *Custom Groups* which are not assigned to any particular file type. Details about groups and operations with files in a project are described in Section 3.7.2.

To open a file not included in the project, select the **Open...** command in the **File** menu. In the *Open* dialog box, select the file name in the required folder, having selected the file type from the drop-down list beforehand (Figure 13).

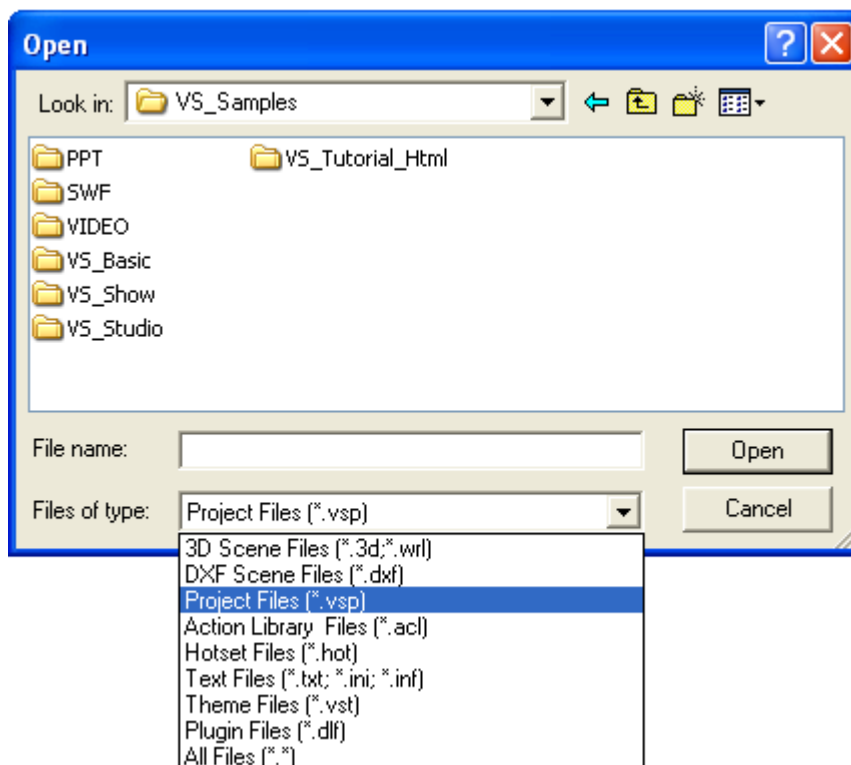

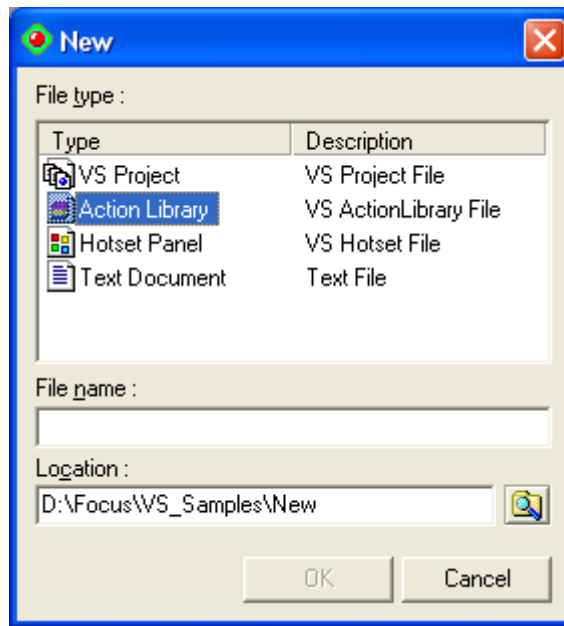


Figure 13. The *Open* dialog box



After that, it is possible to open files in an easier way, with a left click on a filename in the **Recent Files/File** local menu. When opening a file, the application always displays a request to add it to the project (if there is a project loaded).

New files are created in the dialog box opened by the **New** command in the **File** menu (Figure 14), by pressing **Ctrl+N** keys, or by clicking the  button on the window toolbar. In the **New** dialog box, select a file type in the **File Type** field, specify a new project name and a folder for saving it.



**Figure 14.** The dialog box for creating a new file

If an opened file has been changed, and the changes are not saved, an asterisk (\*) appears in its title bar after the filename. The symbol is removed after the first saving of the current file state.


To save the changes in a file of any type, use the **Save** command of the **File** menu when the window with the file is active, or the equivalent  button on the toolbar. To save the file under a different name, use the **Save As...** command. Changes in all the open files (except the scene files) can be saved with the  button.

Details about using each file type are described in the corresponding Sections of this Guide.

### 3.1.8 The *Properties* Dialog Box

The *Properties* dialog box is used to display all the available information about various scene objects, elements of studio control, project components, etc.

The dialog box contents vary depending on the selected object.

The *Properties* dialog box is opened in the same way for all the objects in the studio: by clicking the **Alt+Enter** keys or the  button on the main window toolbar, or from the object context menu. When the dialog box is opened, the button on the toolbar is in the pushed position. A second click on the button hides the *Properties* dialog box. There is no need to close the dialog box every time, it is enough to just switch the focus between the required elements to have their properties displayed.

By default the *Properties* dialog box is displayed docked to the right border of the main window (*Docking View*) and can be rendered “floating” and back with a double click on its title bar.

### 3.1.9 The *Edit* and *LiveAction* Modes of Operation


The *HotActions* application operates in two basic modes: *Edit* and *LiveAction*.


Right after the application is started, it is in the *Edit* mode, in which the user performs all the operations connected with preparing the environment for the *LiveAction* mode. The operations are: opening and closing files, creating *Actions Libraries*, *Hotsets*, *Hotbars*, *Themes*, setting various options, etc. In this mode, the user is presented with two windows: the main window and *Render Output* window (see Section 7).

When switching to the *LiveAction* mode, the application displays a confirmation request to save changes in the modified files of the project.

The *LiveAction* mode is used for the interactive control of 3D scenery (e.g., when creating TV broadcasts). By default, both the application windows switch to the *Fullscreen* mode, and some windows in the main window can be hidden or minimized depending on the settings (see Section 9.1). When switching to the *LiveAction* mode, the application performs a search in all the open *Hotsets* for

a hidden by default specific **Hotbar** with a fixed name **\$STARTUP\$**. All the **Actions** included in the **Hotbar** are started. It always contains a special initializing **Action** (see Section 3.4.7), which is necessary to execute a number of special actions for initializing a 3D scene, such as naming track fragments and others.


Note that a scene can be initialized any moment using the  button on the main application window toolbar.

 *Particularly note that initializing **Actions** can be started from the **\$STARTUP\$** of the **open Hotsets** only! The names of the open **Hotsets** as well as the other file names are in bold type in the project window.*

Usually, the user of the **LiveAction** mode is the virtual set operator. The operator starts the execution of **Actions** commands by clicking the buttons of **Hotbars**. **Actions** commands can also be started by pressing the assigned **Hotkeys** on the keyboard. A **Hotkey** can be assigned for an **Action** on a **Hotbar** in the **Hotkey** tab of the **Properties** dialog box (see Figure 40 in Section 3.6.3). Objects of a scene can be controlled with joysticks or other control devices (see Section 5) and via the network by using the broadcast automation application.


Changing a **Theme** (see Section 3.1.6) allows the virtual set operator to vary the scene appearance and behavior according to the values of the parameters making up the **Theme**. There may be **Standard** and **Custom** themes (see Section 3.8).

In the **LiveAction** mode, the project themes are displayed in the **Theme** drop-down list of the **Project** bar (see Section 3.2.5).

 *Strongly recommended to close all the other active applications before switching to the **LiveAction** mode! It is also recommended to refrain from opening files from the disk during a broadcast session since it causes temporary “freezing” of a frame, which is quite visible if there is some continuous motion in the virtual scene.*

Also note that the application interface has specific features in the **LiveAction** mode:

- clicking the Close button (‘cross’) of a document window does not close but only hides it;
- the arrangement of all the document windows and their visibility are saved independently for the **Edit** and **LiveAction** modes;
- if the buttons of **HotBars** have **Hotkeys** assigned to them (keyboard shortcut, see Section 3.6.3), it is impossible to type symbols corresponding to them in any dialog box in the **LiveAction** mode. That's why is recommended to use key combinations with **Ctrl** or **Shift** keys for **Hotkeys** since such combinations are rarely used in text editing fields.

When switching from the **LiveAction** to the **Edit** mode (the  button), the playback of all the video and audio streams is stopped.

### 3.1.10 Using Video Streams

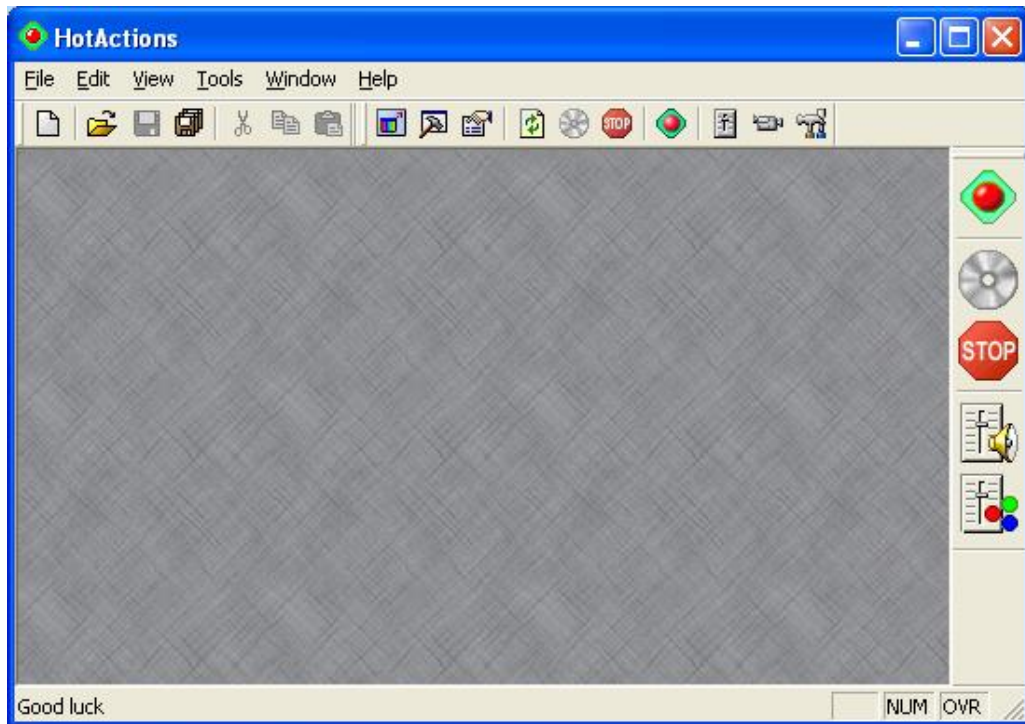
As was mentioned in the Introduction (see Section 1), the **HotActions** application uses 3D scenes created in the **3D Studio Max** or **Maya** graphics editors. After a scene is exported and a project is created, it is possible to show in the scene the video images of the video files played and what is more important, video streams of the video signals connected to the output boards (**FD300** or **Aja XENA**).

Video streams are used as texture maps (or video textures) on any objects of a scene containing **Diffuse Map**, not necessarily flat ones. Usually for operating a video texture, it is convenient to create a simple flat rectangle (**Mesh**) with an “actor” image to make it visual.

The details of assigning a video stream to a texture map are described in Section 6 of *Creating 3D Scenes. User's Guide*. After a scene is initialized (see Section 3.4.7), its appropriate **Diffuse Maps** are replaced with video images from the video sources connected, the rest of the maps and parameters of the material being ignored. Video streams are controlled using special commands, which are common for all the types of video streams. Those commands are discussed in detail in Section 7.1 of *Using the Script Commands. User's Guide*; an example of controlling the video stream of a connected video signal and video stream from a video file is given in Section 8.3.1 of *Creating 3D Scenes. User's Guide*.

The following Sections describe in detail all the components of *HotActions*.

## 3.2 The Main Application Window



**Figure 15. The *HotActions* main window**

The *HotActions* main window (Figure 15) contains various toolbars and dynamic menus for executing different commands including opening, editing, copying, and saving files.

The background color can be changed. To do so, place a wanted image BMP file named *HotActions.bmp* into the following folder: `C:\Program Files\Focus Software\HotActions\Common\Images`. If the window is larger than an image from the file, the image is duplicated as many times as necessary to fill up the window.

### 3.2.1 Toolbars

The *HotActions* main window contains three toolbars: *Standard*, *General* and *LiveAction*. Their arrangement is controlled through the **Toolbars** submenu of the **View** menu.

By default the *Standard* and *General* toolbars are docked to the upper border of the main window (*Docking View*), and the *LiveAction* bar is hidden.



Any main window toolbar as well as the *Properties* dialog box can be attached to any other convenient place of the window or rendered into the 'floating' state by seizing at their left edge and dragging. A double click on a floating toolbar title returns it to the docked state.

The functions executed by the buttons of the toolbars are listed in the next Sections. All the buttons have equivalent commands in the corresponding menus of the main application window (given in brackets).

### 3.2.2 The *Standard* Toolbar



(the **New** command of the **File** menu) creates a new file of a type used in *HotActions*;



(the **Open** command of the **File** menu) displays the *Open* dialog box;



(the **Save** command of the **File** menu) saves changes in a current active document (i.e. whose window is active);



saves changes in all the open windows and corresponding files, except for scene files (\*.3d);



(the **Cut** command of the **Edit** menu) moves a selected object of the active document to the clipboard and deletes it from the document;



(the **Copy** command of the **Edit** menu) copies a selected object of the active document to the clipboard;



(the **Paste** command of the **Edit** menu) pastes the clipboard contents to the current document.

### 3.2.3 The *General* Toolbar



(the **FullScreen Render** option of the **View** menu) switches the *Render Output* window to full-screen mode and back;



(the **Debug Output** option of the **View** menu) displays and hides the *Debug Output* window for displaying diagnostic, error, and system events messages (Section 9.3);



(the **Properties** option of the **View** menu) opens the *Properties* dialog box;



(the **Update All** command of the **Tools** menu) updates the contents of all the open files;



(the **Init All** command of the **Tools** menu) initializes a project by executing special *Startup Actions* (see Sections 3.1.9 and 3.4.6);



(the **Stop All** command of the **Tools** menu) stops all the *Actions* performance and clears their queue, stops the playback of all the animation tracks, video streams, etc. Usually after this command, it is necessary to perform the **Init All** command (see above) to restore the project initial state;



(the **LiveAction Mode** command of the **Tools** menu) switches the application to the *LiveAction* mode (see Section 4);



(the **Sound Configure...** command of the **Tools** menu) opens the Sound Configure dialog box, which is described in detail in a separate document;



(the **Video Configure...** command of the **Tools** menu) opens a dialog box for setting the video signals connected. A detailed description is given in *Video and Sound Settings*;



(the **Options** command of the **Tools** menu) opens the *Options* dialog box (see Section 9).

### 3.2.4 The *LiveAction* Toolbar



The *LiveAction* toolbar usually attached to the right border of the main window, contains the *LiveAction* mode control commands. In the mode, the toolbar buttons have the following functions:



switches the application from the *Edit* to the *LiveAction* mode and back;



the **Init All** command. Performs the same operation as the corresponding button on the *General* toolbar (see Section 3.2.3);



the **Stop All** command. Performs the same operation as the corresponding button on the *General* toolbar (see Section 3.2.3);

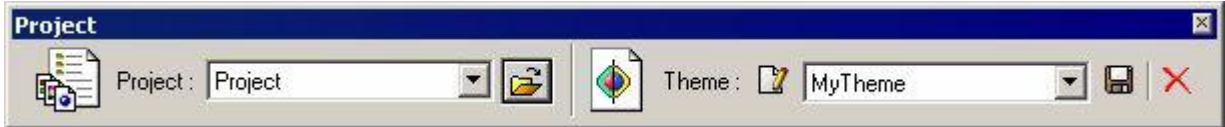


the **Sound Configure** command (see Section 3.2.3). Opens the *Sound Configure* dialog box.



the **Live Video Configure** command. Opens the *Video Configure* dialog box (the **Video Configure...** command of the **Tools** menu).

### 3.2.5 Project Bar



The *Project* bar, which is attached by default to the upper border of the main window in the *LiveAction* mode, contains commands for selecting a current project and controlling its themes. The elements of the bar are used for the following operations:



The **Project** drop-down list allows selecting a project to be loaded from the frequently used (last opened) ones;  
allows opening an arbitrary project;



The **Theme** drop-down list allows selecting a current *Theme* for the active project from the list of themes available for it. The names of the *Standard* type themes in the list begin with an asterisk (see Section 3.1.6);



opens a currently selected theme window to edit;



saves the current theme. If the current theme is *Standard*, you will be offered to save it as a *Custom* theme;



deletes the current theme. The button is available only for *Custom* themes.

### 3.2.6 Menus

The main window contains the **File**, **Edit**, **View**, **Tools**, **Window**, and **Help** menus, many of their commands duplicating the main window buttons of the toolbars described above.

The **File** menu commands allow performing standard operations, such as creating a new file or opening an existing one. The menu also has a list of the most frequently opened files. The **Exit** command is used to exit the application.

The **Edit** menu contains commands for standard editing.

The commands of the **View** menu allow displaying or hiding toolbars, the status bar, the *Render Output* and *Debug Output* windows.

The **Tools** menu contains the main commands of the *HotActions* application general control.

The commands of the **Window** menu allow arranging all the open working windows in a required order.

The **About HotActions...** command of the **Help** menu displays information on the current application version.

Depending on the active state of a window, the **Project**, **Scene**, **Action**, **Hotset** respective submenus appear, and the **View** menu contents change.

The context menu (Figure 16) opened by a right click on a free client space of the main window also contains some commands for the *LiveAction* mode control.

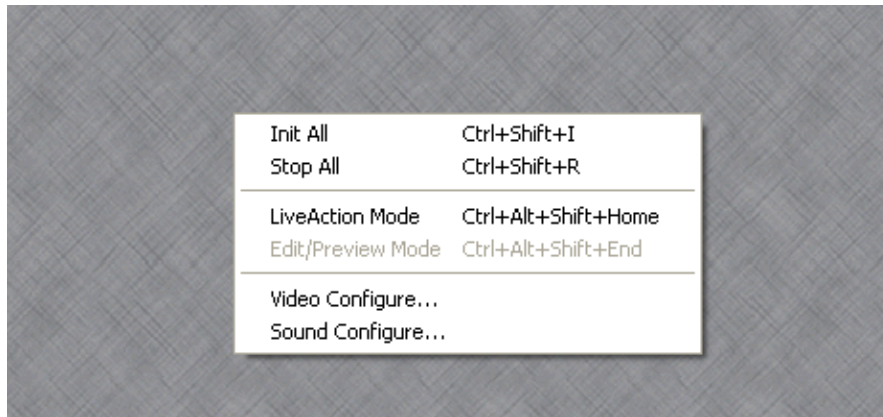


Figure 16. The context menu of a free client space of the main application window

### 3.3 Editing *Actions* and their Properties

As was mentioned above, it is possible to modify virtual scenes in the *HotActions* application using text commands (see Section 3.1.3). The commands are referred to as *Actions*. A special language for creating *Actions* is described in *Using the Script Commands. User's Guide*.

*Actions* can implement whole scenarios performed with the preset sequence automatically, since one *Action* can invoke the execution of others.

Usually an *Action* is executed on clicking the corresponding button on the *Hotbar* (see Section 3.1.5). We remind that it is possible to drag (or copy/paste) *Actions* onto *Hotbars* directly from libraries (see Section 3.6.2). *Actions* are saved in the system as *Actions Library* text documents (see Section 3.1.4).

*Action* properties are displayed in the corresponding tab of the *Properties* dialog box (Figure 17).

#### 3.3.1 The *Action* Tab of the *Properties* Dialog Box

The **Action** tab of the *Properties* window is used to edit the description of an *Action* and its contents (**Body**).

It is possible to briefly describe an *Action* in the **Description** field. The same description appears in the corresponding field of the *Properties* dialog box for a *Hotbar* button (see Section 3.6.3).

The content of an *Action* is entered into the **Body** field of the **Action** tab. Separate commands are located in individual lines (i.e. they are separated by pressing the **Enter** key). It is possible to enter several commands separated with a semicolon in the same line. It is recommended that all the command strings entered into the **Body** field start with a colon. This enables the system to automatically prompt tags (words) of the command and its parameters.

The prompts system is adjusted using the options of the **Auto Editing** submenu of the context menu (Figure 17). The submenu allows enabling/disabling the **Context Help** option (versions of the command words), the **Color Syntax** option (color syntax highlighting), and **Tooltips** (tips displaying a command function when pointing the cursor at the command).

The following rules of color syntax highlighting and separate command tags are adopted in the application:

- command parameters are typed in black;
- comments are highlighted green;
- separate command tags are highlighted blue when their names are identified by the system and the syntax is correct;
- incorrect command tags are automatically highlighted red.

The context menu opened with a right click on the **Body** field in the **Action** tab helps to perform the standard operations of text editing: copying, pasting, selecting, deleting, etc. Each command of the submenu can be performed by pressing the standard (for most text editors) key combinations on the keyboard.

After an *Action* command is created or modified, its performance can be tested with the **Test** button of the *Properties* dialog box (or using the **Ctrl-T** key combination).

The **Apply** button (as well as the **Ctrl-S** key combination) simply saves changes without launching the *Action*. Check the **Auto** check box so that the changes are saved automatically when losing the focus (for example when clicking an object different from the current *Action*).

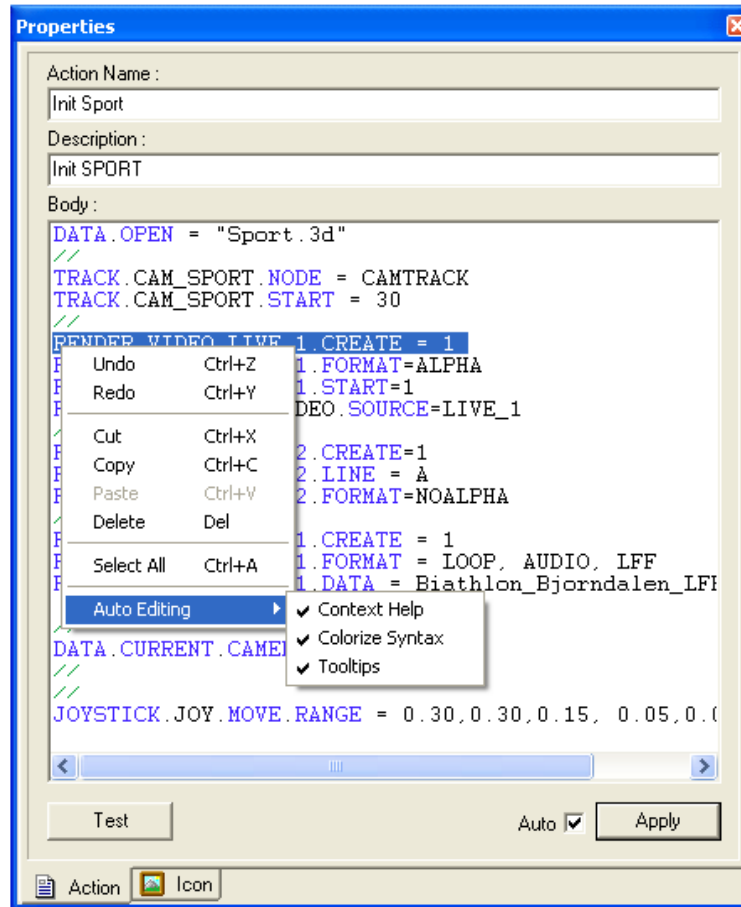


Figure 17. The *Action* tab of the *Properties* dialog box and the *Body* field context menu

When debugging operations performed by the *Action* commands, it is possible to use the command execution messages in the *Debug Output* window (see Section 6). The window, in particular, displays messages about errors occurring when executing *Actions*.

### 3.3.2 The *Icon* Tab of the *Properties* Dialog Box

Using the **Icon** tab of the *Properties* dialog box, it is possible to select an icon for the *Action* and assign it with the **Set Icon** button.

An icon is assigned to the *Hotbar* button corresponding to an *Action* from *Actions Library* (see Section 3.6.2) by default. It is possible to reassign the icon for the *Action* button on the *Hotbar* regardless of the icon assigned in the library. Icons of different *Hotbar* buttons referring to the same *Action* from the library can be different (see Section 3.6.3).





Figure 18. The *Icon* tab of the *Properties* dialog box


The **Icons Folder** drop-down list contains a list of subfolders with icons that can be assigned to *Action* buttons. The path to the folder containing the subfolders with icons is specified in the default directories of the application (see Section 9.2, Figure 78). The default directory is: **C:\Program Files\Focus Software\HotActions\Common\Icons**, which is created when installing *HotActions*. In the list of subfolders **Icon Folder**, the root directory for icons is shown as a dot: “.”. The **<default>** item presents the icons used in the application by default. Selecting the **<All Icons>** item displays all the icons available in all the subfolders of the **Icons Folder** list.

A right click on the window where the icons are shown allows changing their size: they can be large (32 by 32 pixels) or small (16 by 16 pixels).

## 3.4 Operations with the 3D Scene


### 3.4.1 Operations with the Scene File

Scenes, as well as the other files the application works with, are normally opened simultaneously with the whole project. When loading a project, the *Debug Output* window (see Section 6) displays information of the scene opened: the number of the mesh objects, polygons, textures and the value of the video adapter memory occupied by them. It is also possible to load a scene with a double click on its filename in the project window, or by the **Open** command of the file context menu in the project (Figure 49). The names of the scenes opened in the project window are in bold type. A scene file is closed either by the **Close** command of the same context menu or by closing its window via the corresponding system menu.

 If the scene window presents a disturbance when working (it overlays the main window workspace), it can be hidden by the **Hide** command from the file context menu of the project (Figure 49).

To add a new scene to the open project, use the **Add File to Group** command of the context menu for the **Scenes** standard group in the project window (Figure 51). To open a scene file, it is also possible to use the **Open...** command in the **File** menu of the *HotActions* main window. In the *Open* dialog box specify the required file extension (“**.3d;.wrl**” or “**.dxf**”, see an example in Figure 13) and select a scene name in the required folder. After that, the scene file can be opened faster: with a left click on its

name in the **Recent Files** local menu of the **File** menu. When opening a scene, the application always displays a confirmation request to add it to the project if a project is loaded.

To save changes in the scene, select the **Save** command in the menu or the equivalent  button on the toolbar when the scene window is active.

When pointing the cursor at an object of the scene in the *Render Output* window (Section 7), the name of the selected mesh object displays on the status bar in the lower part of the main application window.

The *HotActions* application does not automatically save the results of manipulating the scene objects, i.e. hiding, changing sizes and positions of objects. When a scene has some changes that can be saved, the title bar of the scene window bears an asterisk (\*). It disappears after the current scene state is saved.

To save the scene under a different name, use the **Save As...** command.

### 3.4.2 Symbols Used for the Scene Objects

When working with a scene, its name is displayed in the title bar of the scene document window (Figure 19). The structure of the scene is always presented as a tree of so-called *Nodes* with the following denotations:



the root node of a scene tree (*World*), every scene necessarily has one root node;



*Dummy* scene nodes, not displayed in the *Render Output* window. Such nodes are used as auxiliary, for example, to combine real objects into a logical whole, so that it's possible to apply changes to all the nodes of the group simultaneously. Also note that a camera in the scene is controlled not directly with the mouse or joystick but via such *Dummy* objects. For this purpose cameras must be connected with separate *Dummy* objects (see also *Creating 3D Scenes. User's Guide*);



scene nodes referring to 3D objects (having polygons);





nodes referring to virtual cameras of a particular scene;



nodes referring to light sources of a particular scene.



Object symbols may have additional indications marking the presence/absence of tracks (animation):

- small green triangle (like ) denotes that the node has its own track in the scene general track;
- small yellow triangle (like ) denotes that descendant nodes have at least one track of their own in the scene general track.

Operations with tracks are discussed in *Using the Script Commands. User's Guide*.

### 3.4.3 The Scene Window Toolbar and Presentation of the Scene Objects according to Types

The scene working window has its own toolbar:

- The  button displays all the scene objects as a node tree (Figure 19). The context menu opened by a right click on a required node allows you to hide/show the object in a scene, copy its name, generate script commands describing the current position of the object (see the  button function), or open its properties dialog box. A double click on a node selects it for the mouse/joystick manipulation in the scene (see Section 5). The selection of a node to be manipulated is saved in the scene file.

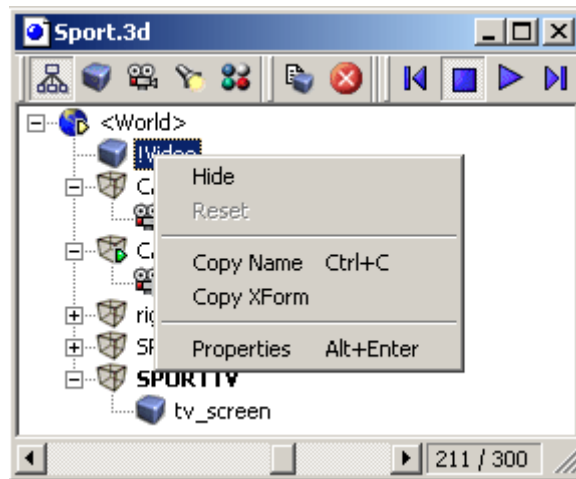



Figure 19. A working window of the scene nodes and the context menu for a node

The context menu for a node contains the following commands:

- Hide** hides the node selected in the scene. If the node refers to a light source, the latter is 'switched off';
- Reset** restores the position and size of the selected scene node as they were when opening the scene. Accordingly, the command is available after changing the node position and/or size. To show that there are changes, the object name in the scene window is added with an asterisk (\*);
- Copy Name** the command copies the selected object name to the clipboard to use, for example, in script commands. To paste the object name copied with the command, use the **Paste** command of the *Properties* dialog box context menu when editing the *Action* (see Section 3.3.1, Figure 17);
- Copy XForm** equivalent to the  button action;
- Properties** opens the *Properties* dialog box (Section 3.4.5) for the selected object.

- The  button presents all the 3D objects of the scene or *Meshes* (Figure 20).

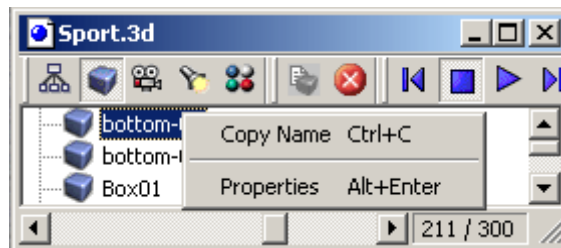


Figure 20. Presentation of all the scene meshes and the context menu for a mesh

- The  button presents all the cameras available in the scene (Figure 21).

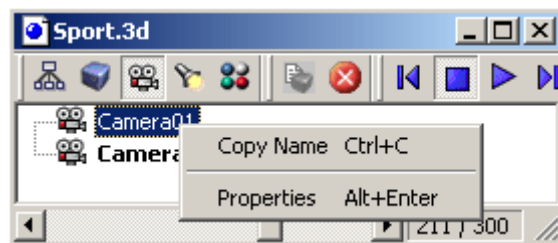


Figure 21. Presentation of all the scene cameras and the context menu for a scene camera


- The  button presents all the light sources available in the scene (Figure 22).



Figure 22. Presentation of all the scene light sources and the context menu for a light source






- The  button presents all the available materials of all the scene objects (Figure 23).



Figure 23. Presentation of all the scene materials and the context menu for a material

- The  button is used to generate the script commands that describe the current position of the object and its scale (*XForm*). When clicking it, the script commands that result in setting the current position and scale of the object are copied to the clipboard. After that, it is possible to paste these commands from the clipboard by the **Paste** command of the context menu in the **Body** field of the **Action** tab, the *Properties* dialog box. At that, the  (Paste) button in the main application window becomes unavailable, since the toolbar it refers to is used for documents, for the scene in this case, but **Paste** is not specified for the scene.
- The  button cancels all the scene modifications (of the scale and positions of all the scene objects) and restores the initial state.
- The  buttons are used to play back and stop the scene track (animation). The buttons are used mostly to preview the scene animation loaded as a separate file, without a project (i.e. without *Actions* modifying it, etc.). That is caused by the fact that when the scene is initialized in the project, the startup Action commands usually reassign the control of separate tracks, which excludes the tracks from the general scene animation controlled by the buttons.

At any type of the scene object representations, a right click on a free place of the window client area opens a menu (Figure 24). The menu contains commands for switching the presentation type (they are equivalent to the toolbar buttons), and also the **View** local menu with the options of the toolbars view.

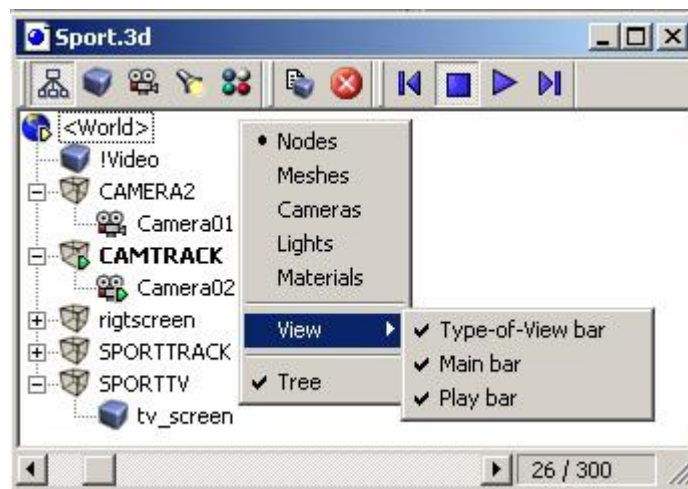


Figure 24. Scene window context menu

### 3.4.4 The *HotActions* Main Window Menu When Working with the Scene

When working with the scene, the **Scene** special menu appears in the *HotActions* main window (Figure 25), and the **Edit** menu contents change. The **Play**, **Stop**, **Rewind**, **Reset All Nodes** commands of the **Scene** menu duplicate the buttons of the scene working window. The **Copy Name**, **Copy Xform** commands of the **Edit** menu duplicate the scene working window buttons described above.



Figure 25. The *Scene* and *Edit* menus of the main application window

### 3.4.5 The *Properties* Dialog Box for Different Types of the Scene Objects

The *Properties* dialog box opened for any object of the scene (a node, camera, light source, material, etc.) from any presentation (see Section 3.4.3) provides detailed information about the object.

The information is displayed in the corresponding tabs.

The **Node** tab displaying the node name, its position, scale, and rotation angles (Figure 26) has a similar appearance for all the objects. The parameters being editable, clicking the **Apply** button applies the modifications and the **Reset** button restores the node initial state.

Checking the **Hide** check box hides the object in the scene.

The **Control** button, as well as a double click on a node in the scene tree (Figure 19), selects it for manipulation with the joystick, mouse or arrow keys, the object name being in bold type. Operations that can be performed with the joystick are discussed in Section 5.

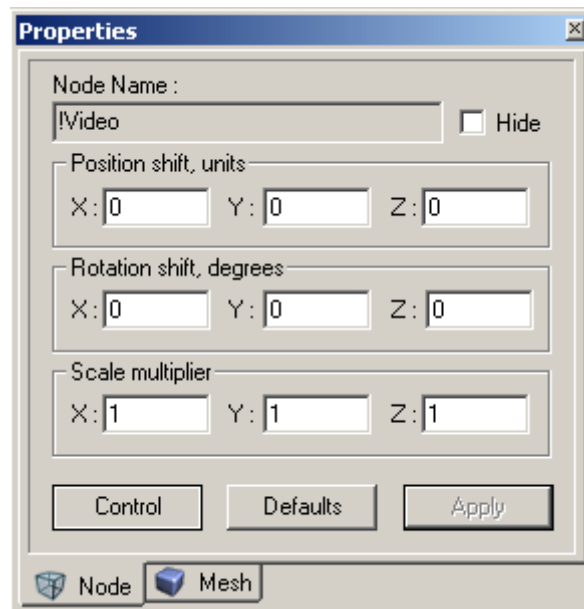



Figure 26. The *Node* tab of the *Properties* dialog box

The *Properties* dialog box contains additional tabs for the scene objects, except for *Dummy* ones designated as .

The **Camera** tab (Figure 27) is used for cameras. The tab allows specifying the **FOV** (field of view) parameter in degrees or the lens focal distance in millimeters if the **Lens** option is set. It is also possible to specify the parameters of the clipping planes and the range of the environment parameters for the selected camera.

The **Set Active** button renders the camera current in the scene, but does not make it the manipulated object. For that purpose the camera must be selected in the **Node** tab (Figure 26) or in the scene objects tree (Figure 19).

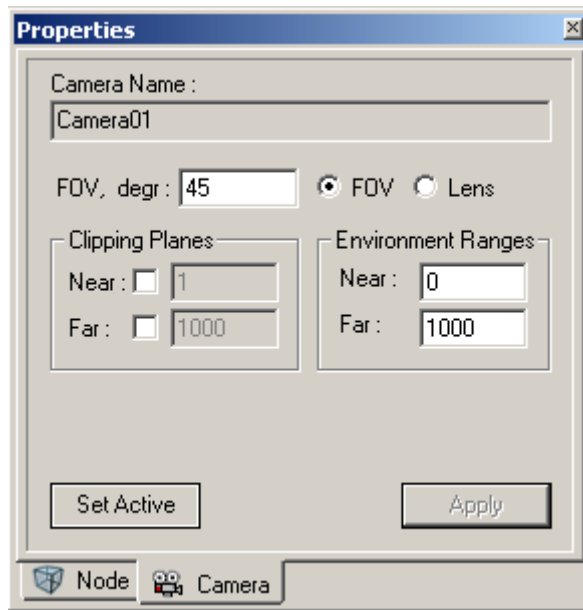


Figure 27. The *Camera* tab of the *Properties* dialog box

The additional **Mesh** tab (Figure 28) displays information about the number of the object faces, vertices and the used materials for the three-dimensional objects in the scene (*Meshes*) consisting of sets of faces.

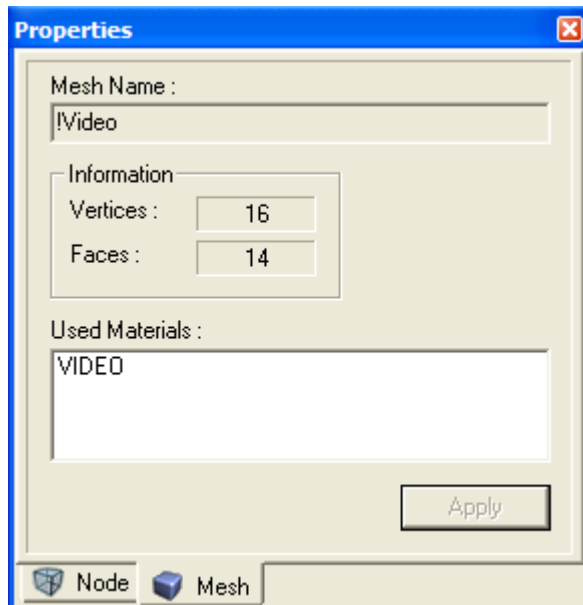


Figure 28. The *Mesh* tab of the *Properties* dialog box

The additional **Light** tab (Figure 29) for the scene light sources allows specifying a color for the light source if the **On** check box is checked. The light source can be ‘turned off’ completely by unchecking the check box. To open the corresponding color selection dialog box, click the **Color** field. The tab also allows specifying a multiplication factor for the light source intensity level in the **Multiplier** field.

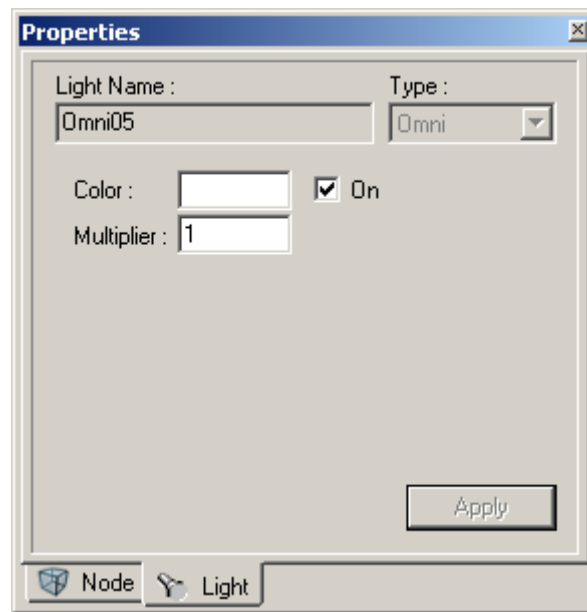


Figure 29. The *Light* tab of the *Properties* dialog box

For an object selected not from the scene node tree (Figure 19) but from its group (Figure 20, Figure 21, Figure 22, Figure 23), the *Properties* dialog box displays only the listed above additional properties tabs corresponding to the object type, without the **Node** tab.

Since materials are not attached to particular nodes of the scene, there is no **Node** tab for them. You can see their properties (Figure 30) after selecting a particular material from the list of all the scene materials (Figure 23). If a material has textures, information about them is also displayed in the properties tab (Figure 30), the texture size, bits per texel (color depth) and number of frames (if more than one) being specified above each texture.

Some of material parameters can be modified. The colors of the material are grouped on the left: the **Ambient** color in the shade, the **Diffuse** color of the illuminated surface, the **Specular** color of the flare. On the right are located: the self-luminosity value (**Self-Illum.**), **Opacity**, and **Glossiness**.

The **Find in Scene** button is designed to search the material in the scene. When clicking the button, the surface the material is assigned to starts blinking. A second click on the button or switching to another window stops the blinking.

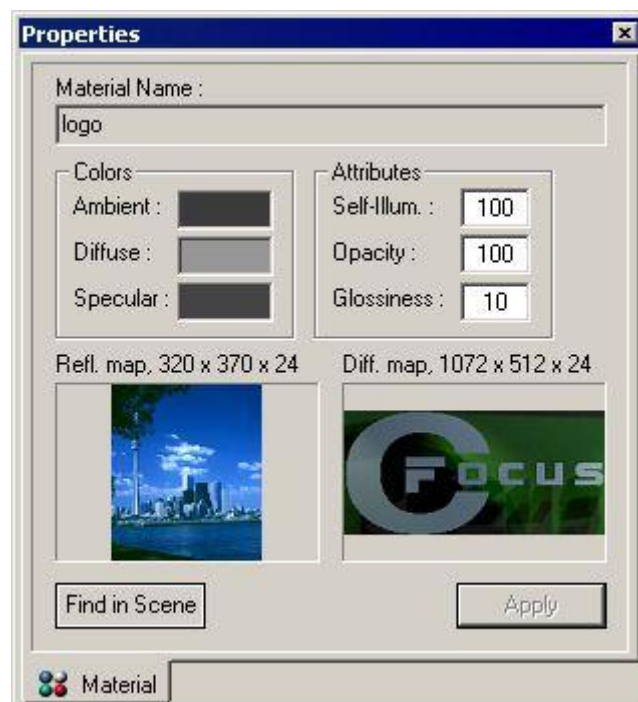


Figure 30. The *Material* tab of the *Properties* dialog box

If a scene node or its descendants have an individual track in the scene general track, the *Properties* dialog box also contains the **Track** tab (Figure 31).

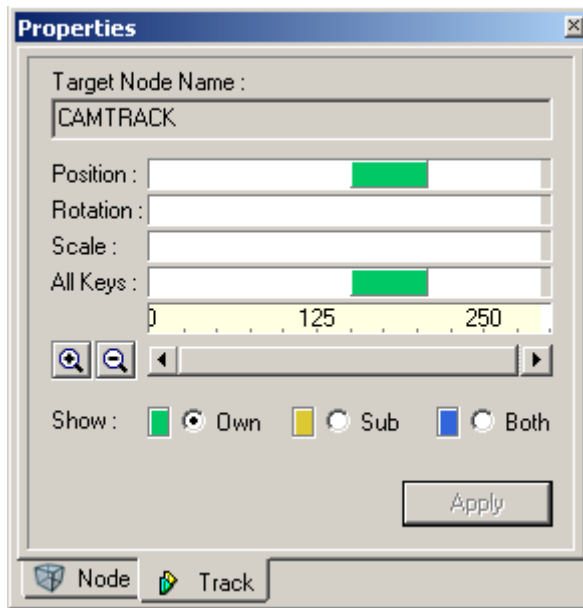


Figure 31. The *Track* tab of the *Properties* dialog box

The tab displays information about each constituent of the object track. Changes of the object position, the angle of rotation and scale are displayed in the tab as colored rectangles located on the respective bars. The rectangles are colored according to the radio button selected in the **Show** group.

The **Show** radio buttons are used to group and display the following tracks:

- the object own tracks (the **Own** radio button selected);
- the total information about its descendant tracks (the **Sub** radio button selected);
- the object own tracks along with its descendants (the **Both** radio button selected).

The *Properties* dialog box includes the **Environment** tab for the root node of the scene tree (*World*) denoted by the  icon.

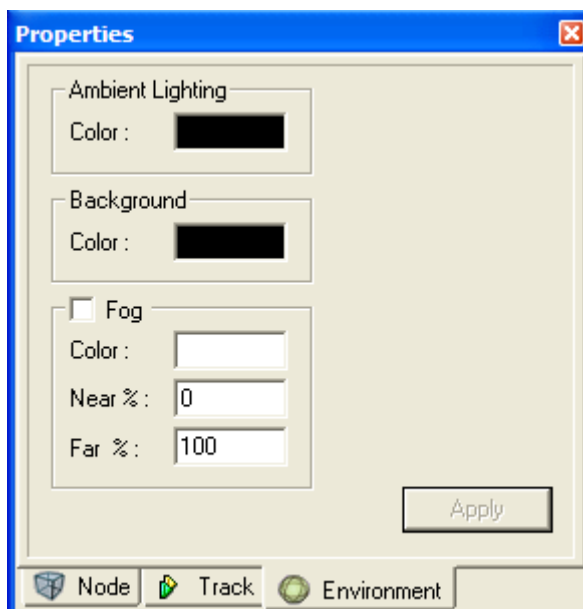


Figure 32. The *Environment* tab (for the *World* object) of the *Properties* dialog box

The **Ambient Lighting** area of the tab is used to specify the color of the scene ambient lighting. Click the **Color** field to open the color selection dialog box. The background color, i.e. the color of the space free of scene objects is specified in the same way under **Background**.



The **Fog** area is designed to use the fog effect in the scene. Check the **Fog** check box and specify a required color in the **Color** field. The **Near** and **Far** parameters set the fog density ratio (in percentage terms) at the near and far clipping planes for the virtual camera. The clipping planes are specified in the **Environment Ranges** area of the **Camera** tab, the *Properties* dialog box (Figure 27), and they can vary for different virtual cameras in the scene.



### 3.4.6 The Function of *Startup Actions*


After a scene is created in a 3D editor, such as *3D Studio Max* or *Maya*, and exported, which is discussed in detail in *Creating 3D Scenes. User's Guide*, we have a file of the scene with the **\*.3d** extension.

The scene file can be used in the existing projects of the *HotActions* application or in a new project created specially for it. In any case, it is necessary to have a handy toolset to control the scene: to load the scene, manipulate the objects, start object tracks, etc.

These operations are executed by *Actions*, which present sequences of one or several text commands. The execution of an *Action* results in some modification of the virtual scene (see Section 3.1.3).

The *Startup Actions* are an important group of *Actions*. They differ from usual *Actions* in their task, which is rendering the scene to the initial operating state. Most often such *Actions* are used after loading the scene.

- One of the advantages of the *Startup Actions* is uniformity of operations with projects: it is easy to initialize even an unfamiliar project simply by clicking the  button on the main application window toolbar. *Startup Actions* are located in the **\$STARTUP\$** folder, since the *HotActions* application automatically executes the contents of the **\$STARTUP\$** folders on clicking the  button or when switching to the *LiveAction* operating mode (see Section 3.1.9).
- Originally there is a **\$STARTUP\$** folder in each *Hotset*. By default the folder is hidden, but it can be displayed after switching to the *Hotset* configuring mode (see Section 3.6.6).

Note that the  button becomes available only when the project has at least one *Hotset* open. Adding a *Hotset* to a project and creating a new *Hotset* are discussed in Sections 3.7.2 and 3.6.1 respectively.

*Actions* located in the **\$STARTUP\$** folder are executed in a strictly specified top-down order (this refers to the *Hotset* alternative presentation, see Section 3.6.6). Consequently, it is possible to modify the order of execution by moving references to *Actions* in the folder.

 Note once again that switching to the *LiveAction* mode (see Section 3.1.9) automatically launches all the *Startup Actions*.

### 3.4.7 Creating *Startup Actions*

The main task of *Startup Actions* is initializing the project to the initial operating condition. The basic operations required for that purpose are listed below. The commands given within the example belong to the “**Init Sport**” *Startup Action* of the “Sport” project available in the software package.

1. First of all, load the scene  
:DATA.OPEN = «Sport.3d».
2. The scene objects can have tracks (animation based on trajectories). In order to control tracks in the virtual studio, they have to be named since, for example, track names are not specified in the *3D Studio Max* application (see Section 2.1.1 of *Creating 3D Scenes. User's Guide*)  
:TRACK.CAM\_SPORT.NODE = CAMTRACK.

In other words, the command assigns the name “CAMTRACK” to the “CAM\_SPORT” object track, further the name being used for operations with the track, for example, for starting it. A detailed description of the track control commands is available in Section 4 of *Using the Script Commands. User's Guide*.

3. Create the video streams for the scene and assign them to the existing materials (the commands are discussed in detail in Section 7 of *Using the Script Commands. User's Guide*):


```
//create a video stream connected with LIVE_1 in the Stream drop-down list, Figure 6
:RENDER.VIDEO.LIVE_1.CREATE = 1
//specify the video stream format including the alpha-channel
:RENDER.VIDEO.LIVE_1.FORMAT = ALPHA
//start the video stream
:RENDER.VIDEO.LIVE_1.START = 1
//assign a video texture from the LIVE_1 video stream to the material named VIDEO so that
//the texture is overlaid on all the scene objects having this material
:RENDER.MATERIAL.VIDEO.SOURCE = LIVE_1
```

After loading the scene and naming its tracks, *Startup Actions* as usual *Actions* can contain any commands for starting them.

For details about any command, see *Using the Script Commands. User's Guide*.

## 3.5 Operations with the Action Library

### 3.5.1 Operations with the Library File

To create a new *Action Library* file, open the dialog box for creating new files using the **New** command of the **File** menu in the *HotActions* main window or by pressing **Ctrl-N** keys or clicking the  button on the toolbar. Then select the **Action Library** type from the list of new file types available. Specify a name for the new library being created and a folder to save it.

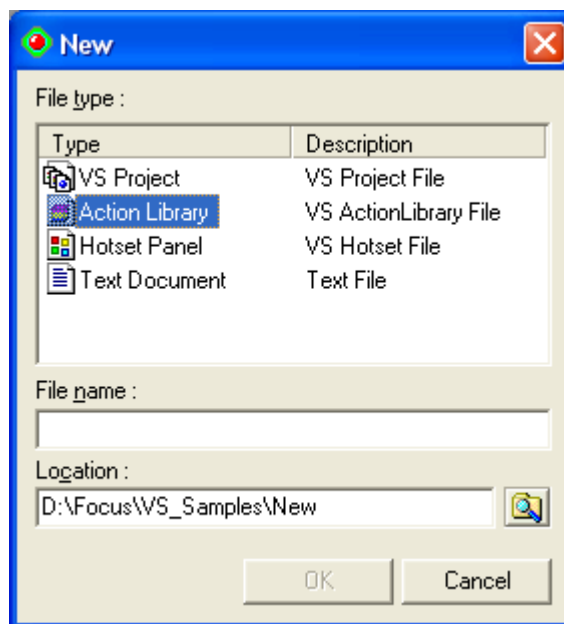



Figure 33. Selecting the file type and a file name when creating an *Action Library*

To attach the new library to the project right away, check the **Add to Project** check box in the bottom part of the dialog box (checked by default). To open a previously created library, use the *Open* dialog box. Select the **Open...** command in the *HotActions* **File** menu, in the *Open* dialog box, specify the *Action Library Files* type (\*.acl, Figure 13) to be opened, select the required file, and click the **Open** button. After that, the application displays a confirmation request to include the file in the current project. If the library is not to be included in the project, for example, when opened only to copy some of its *Actions*, click **No**. After the first opening of the library, it can be opened directly with a click on its name in the **Recent Files** local menu in the same **File** menu.

A library can be added to the project using the context menu of the **Libraries** standard group in the project window.

To save the changes in the library, select the **Save** command in the same **File** menu or the equivalent  button on the toolbar (when the library window is active). To save the library under a different name, use the **Save As...** command.

### 3.5.2 Operations with *Actions* in the Library

*Actions* in the library can be treated like usual files in a folder: they can be copied, pasted, deleted, etc. The same operations can be done with a group of *Actions* after the required ones are selected as usual – with the arrow keys or the mouse clicks with the **Ctrl** or **Shift** key pressed.

A right click in the library window empty space opens the context menu (Figure 34). The menu is a combination of some commands of the **Action** and **View** menus of the main window (see Section 3.5.3).

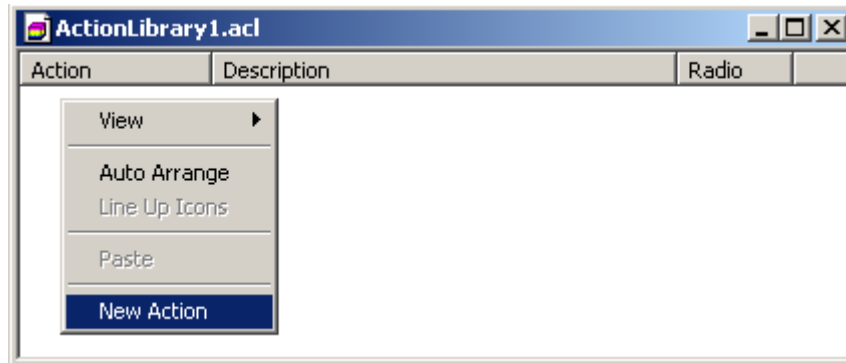


Figure 34. The *Action Library* context menu

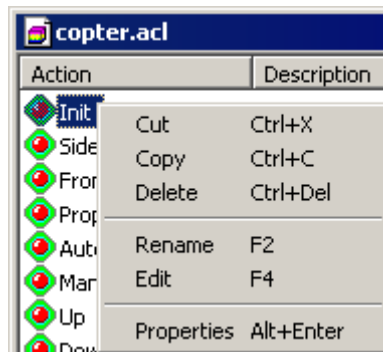


Figure 35. The context menu of an *Action* in the library

The **Edit** command of another context menu (Figure 35) is used to edit the *Body* contents of the selected *Action*. The command activates the corresponding tab of the *Properties* dialog box (Figure 36). The *Properties* dialog box can also be opened by the **Properties** command of the context menu of an *Action* or with the **Alt-Enter** key combination.

The contents of an *Action*, or its ‘body’, are a set of script commands, which can be edited in the **Body** field in the *Action* tab of the *Properties* dialog box.

After creating an *Action* or modifying its ‘body’, its commands can be tested. To do so, click the **Test** button of the *Properties* dialog box or use the **Ctrl-T** key combination.

The **Apply** button as well as the **Ctrl-S** key combination simply saves changes in the body without starting the *Action*. To enable automatic saving of changes made in an *Action* body when switching the focus from the dialog box, e.g. after a click on another object, check the **Auto** check box.

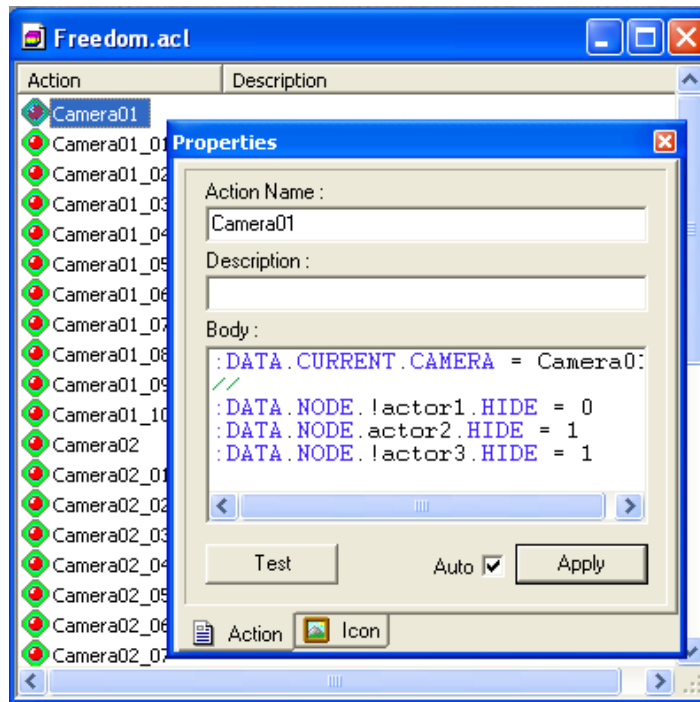


Figure 36. The *Action* tab of the *Properties* dialog box

It is possible to assign an icon to the *Action* in the **Icon** page of the *Properties* dialog box. For more detail about editing *Actions*, see Section 3.3.

It is possible to exchange *Actions* between open libraries by dragging. They can be copied or moved with the **Ctrl** key pressed when dragging.

When *Actions* are moved to a *Hotbar*, the corresponding buttons are added to it. If they are moved to an empty space on the *Hotset*, a new *Hotbar* is created (see Section 3.6.2).

### 3.5.3 The Main Window Menu When Working with an *Action Library*

When working with a library, a special **Action** menu for manipulating the library contents is added to the standard list of the main window menus, the **View** menu being modified (Figure 37).

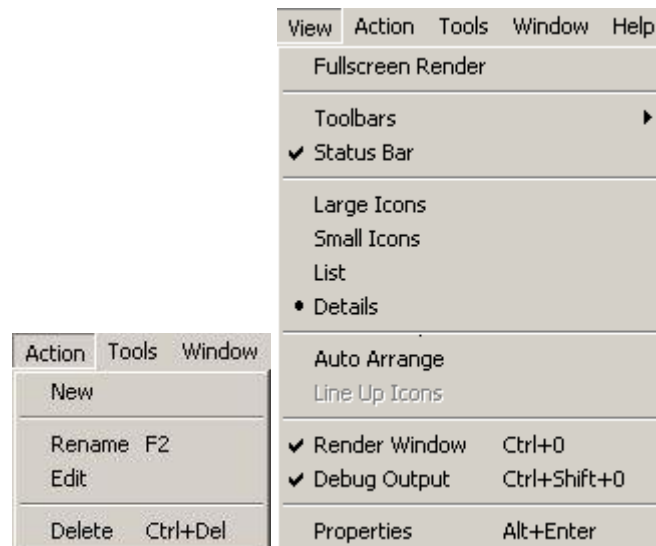


Figure 37. The *Action* and *View* menus of the main window

The **Action** menu contains the following commands:

**New** – creates a new *Action*;

**Rename** – renames an existing *Action*;

**Edit** – opens the *Action Properties* dialog box to edit the ‘body’ of an *Action*;

**Delete** – deletes the selected *Action*.

The **View** menu includes the following commands for working with *Action Libraries*:

The **Large Icons**, **Small Icons**, **List**, and **Details** commands change the presentation of *Actions* in the library window to large icons, small icons, simple list, or a detailed list (with several columns), respectively;

The **Auto Arrange** command enables automatic *Actions* arrangement in rows in the library window (if the **Large Icons** or **Small Icons** mode is selected);

The **Line up** command arranges *Actions* in rows in the library window (if the **Large Icons** or **Small Icons** mode is selected).


*Actions* in the library can also be controlled with the library context menu (Figure 34).

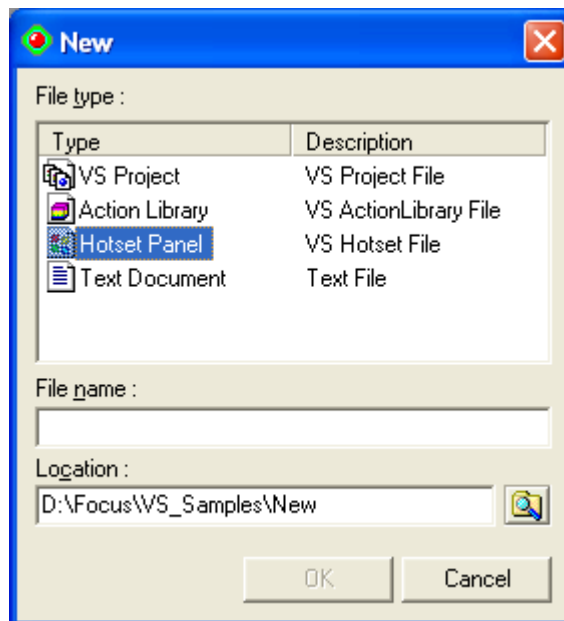
## 3.6 Operations with *Hotsets* and *HotBars*

The functions of *HotBars* and *Hotsets* are discussed in Section 3.1.5.

### 3.6.1 Operations with the *Hotset* File

*Hotsets* are created and opened similarly to other documents the *HotActions* application works with.

To create a new *Hotset*, open the *New* dialog box using the **New** command in the **File** menu of the main application window or the **Ctrl-N** key combination, or the  button on the toolbar. Select the **Hotset Panel** type from the list of available file types. Then specify the file name and the folder to save it in.




**Figure 38.** The *New* dialog box for specifying the file type and name when creating a *Hotset*

To attach the new *Hotset* to the project right away, check the **Add to Project** check box in the bottom part of the dialog box (checked by default).

As all the other files included in the project, *Hotsets* are automatically opened when opening the project. Unlike other file types, *Hotsets* are rarely used as independent files, mostly they belong to projects, for which they are, in fact, created.

A *Hotset* can be opened without a project as well using the **Open** command of the **File** menu (Figure 13). At that the application displays a request to add the *Hotset* to the project if there is a project loaded. After an existing *Hotset* is opened for the first time, or a new one is saved, it can be opened directly with a click on its name in the **Recent Files** local menu of the **File** menu.

To add a previously created *Hotset* to the open project, use the context menu for the standard group **Hotsets** in the project window (Figure 51).

A *Hotset* can be saved under a different name using the **Save As...** command of the **File** menu. The  button or the **Save** command save it with the same name (when the *Hotset* window is active).

### 3.6.2 Manipulating *HotBars* and their Buttons in the *Hotset*

The arrangement of *HotBars* in the *Hotset* can be changed, and it is possible to add and modify buttons directly on the *HotBar* in the *Edit* mode (see Section 3.1.9) of the control environment.

It is possible to add buttons to any *HotBar* by dragging *Actions* from *Action Libraries* directly onto the *HotBar*. When an *Action* is placed on the *Hotset* empty space, a new *HotBar* is created for it.

Buttons can be easily manipulated, it being possible to:

- change their text, icons, assign *Hotkeys* (keyboard shortcuts) using the **Button** tab of the *Properties* dialog box (Figure 40);
- delete, add, copy buttons, insert separators between them via the context menu (Figure 39);
- change their order on the *HotBar* by dragging and dropping;
- drag them between different *HotBars* and *Hotsets*.


In the *Edit* mode, a single click on a button selects it but not operates. To test the operation of a button, click it with the **Alt** key pressed. The **Test** command of the button context menu has the same operation (Figure 39). Some other commands of the menu are: **Insert Separator** (between buttons) and **Take&Set Snapshot** (to set a scene image onto the button, see Section 3.6.4). The rest of the context menu commands are intended for the standard operations with the button: copying (**Copy**), deletion (**Delete**), etc. The **Properties** command of the menu opens the *Button* tab of the *Properties* dialog box used to edit the button properties (see the following Section).

Note that each command of this menu has a *Hotkey*.



Figure 39. The context menu of a button

*Hotbars* themselves can be moved between *Hotsets* by dragging.

 Note once again that the *Hotset* does not contain *Actions* proper. It contains only references to specific *Action Libraries*. When opening a *Hotset*, the application automatically loads the *Actions* its buttons refer to from the corresponding libraries. That is why a *Hotset* is independent of a particular description of a certain *Action*. Thus, a *Hotset* can be formed reasoning from the subject of the prospective TV show.

### 3.6.3 Operations with the *HotBar* Buttons Properties

The *Properties* dialog box allows modifying the buttons properties (Figure 40). The dialog box can be opened using the **Properties** command of the context menu (Figure 39) or with the **Alt-Enter** key combination.

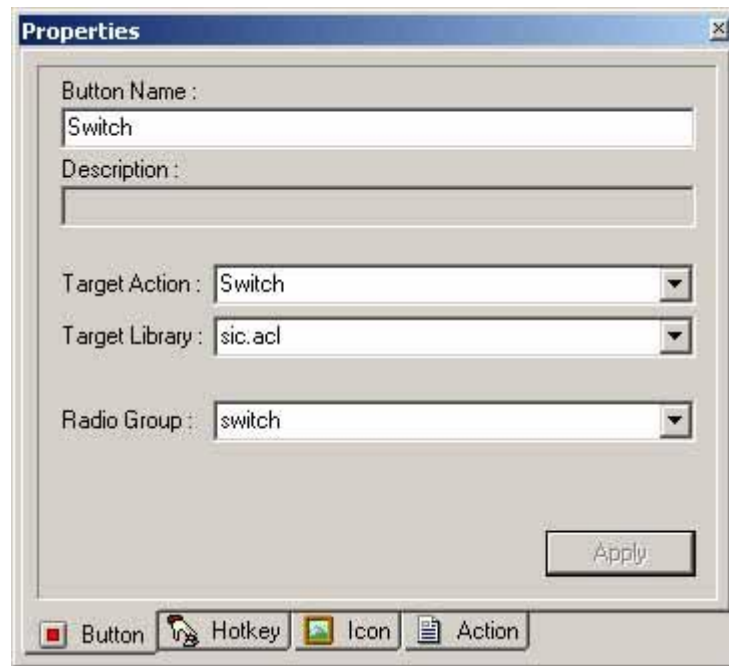



Figure 40. The *Button* tab of the *Properties* dialog box

The **Button Name** field in the *Button* tab of the *Properties* dialog box can be used to edit the button name. The button description displayed in the **Description** field is imported from the corresponding *Action* and therefore is not editable. The *Action* the button refers to can be changed in the **Target Action** drop-down list, and library where an *Action* can be selected is specified in the **Target Library** drop-down list. If the library is changed to another one, the *Actions* of the current one become available in the **Target Action** drop-down list.

The **Radio Group** combo box is used to specify a radio group for the current button. It is possible to select a radio group from the available groups in the list or create a new group by entering its name in the text field of the combo box. A radio group is a set of buttons in which only one button can be selected.


The **Apply** button of the *Properties* dialog box saves all the changes without exiting the dialog box.

The **Hotkey** tab of the dialog box allows assigning a *Hotkey* (a keyboard shortcut) to the button.

 Note that in the *LiveAction* mode, the characters corresponding to *Hotkeys* cannot be entered in any of the dialog boxes. Therefore, it is recommended that you use key combinations with **Ctrl** or **Shift** keys when assigning *Hotkeys*, since the combinations are rarely used in the text editing fields.

The button icon can be changed in the **Icon** tab. It is completely identical to the **Icon** tab of the *Properties* dialog box when working with *Actions* (see Section 3.3.2), but it sets an icon for the button, not the *Action*. The icons of different buttons referring to the same *Action* may differ. When the **Snapshots** option is enabled in the properties of the *HotBar* containing the button (Figure 43), the icons are replaced with snapshots – images copied from the **Render Output** window. For more detail about operations with snapshots, see Section 3.6.4.

The body and description of the *Action* corresponding to the button are displayed in the **Action** tab (see Section 3.3.1). Changes made in the tab are immediately reflected in the source text. In much the same way, the changes made in an *Action* body in the library immediately change the properties of the corresponding button.

The full synchronization of the *Action* bodies in the libraries and all the *Hotbar* buttons is performed by the **Update All** command of the **Tools** menu in the main application window. The  button on the **General** toolbar of the main application window has the same operation (see Section 3.2.3).

### 3.6.4 Using *Snapshots* for *HotBar* Buttons

It was already mentioned that when the **Snapshots** option is enabled in the *HotBar* properties (see Section 3.6.5), the icons of all its buttons are replaced with the snapshots – images captured from the **Render Output** window.

The easiest way to use the ability of placing a scene image on a button is selecting the **Take&Set Snapshot** command in the button context menu (Figure 39). Note that the option is unavailable in the Figure since the usual icons are used in the current *HotBar* (so remember to switch the *HotBar* to the **Snapshots** mode first). In the dialog box opened by the command, specify a name for the image file. The name usually matches the button name it is intended for. Specify a folder to save the file. Most often, it's the **Shots** folder in the project local directory. If the folder is absent, the application displays a confirmation request to create it.

When using **Snapshots**, the **Icon** tab of the *Properties* dialog box is modified correspondingly (Figure 41). The controls of this tab allow selecting a folder with images.

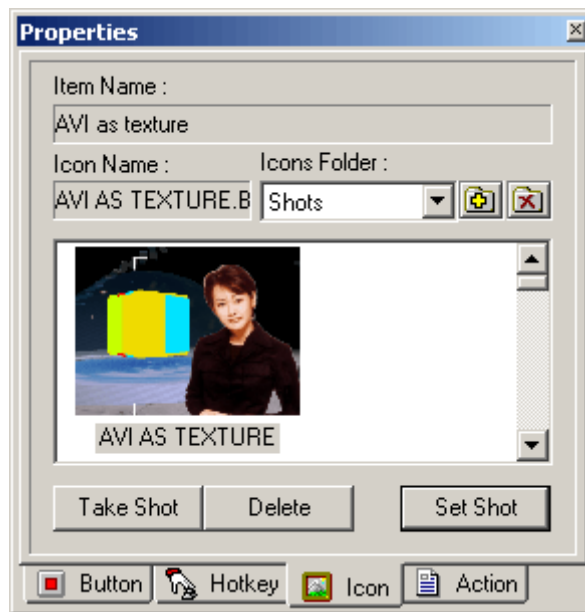


Figure 41. The *Icon* tab of the *Properties* dialog box

The **Icons Folder** drop-down list is used to specify the folder from which images for the button are selected. All the BMP files in the specified folder become available for selection (they are automatically scaled by the application when necessary). A “.” dot in the list stands for the project folder, the **<default>** folder contains the default icons.

The buttons located directly on the right of the folders drop-down list are designed to add a folder to the list or remove it. If the **Add Subpaths** option is enabled when adding a folder, all its subfolders are also added to the list. If **<All Icons>** is selected, it is possible to choose an icon from all the folders of the list.

To assign a selected image to the button, it is sufficient to double-click it or click the **Set Shot** button. The **Take Shot** button captures the current image from the **Render Output** window and adds it to the current folder for snapshots (selecting the **<All Icons>** from the folders list makes the button unavailable).



Note that the **Delete** button deletes the selected image from the disk.

### 3.6.5 The Context Menu and *Properties* Dialog Box when Working with a *HotBar* in the *Hotset*

The context menu of a *Hotset* includes the main commands for manipulating *HotBars*. The commands duplicate those of the *HotBar* menu of the main application window (see Section 3.6.7), except for the **Properties** command available.



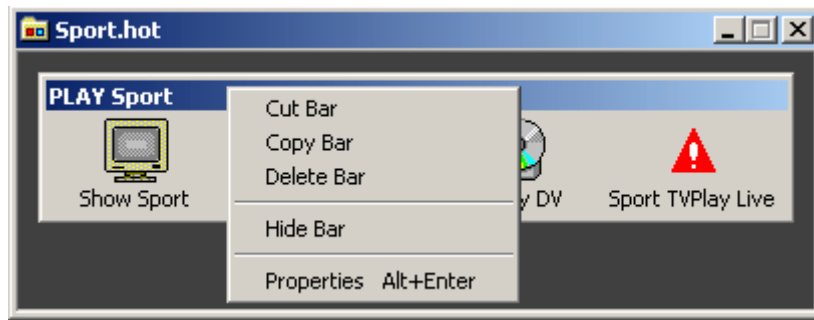


Figure 42. The context menu for a *HotBar* in the *Hotset*

The *Properties* dialog box for a *HotBar* in the *Hotset* (Figure 43) allows changing the name of the *HotBar*, its description, and presentation of buttons on it (the text arrangement and icon sizes). In particular, it is possible to change the arrangement of text on buttons: **At Bottom**, **At Right**, or **None**. Selection of the **Snapshots** option enables using snapshots – images captured from the **Render Output** window – instead of the icons for all the buttons of the *HotBar* (see Section 3.6.4 for information about operations with snapshots). After clicking the **Apply** button for saving changes, the application displays a request to create a dedicated folder named **Shots** for snapshots (if absent) in the current project folder. For details about operations with buttons see Section 3.6.2.

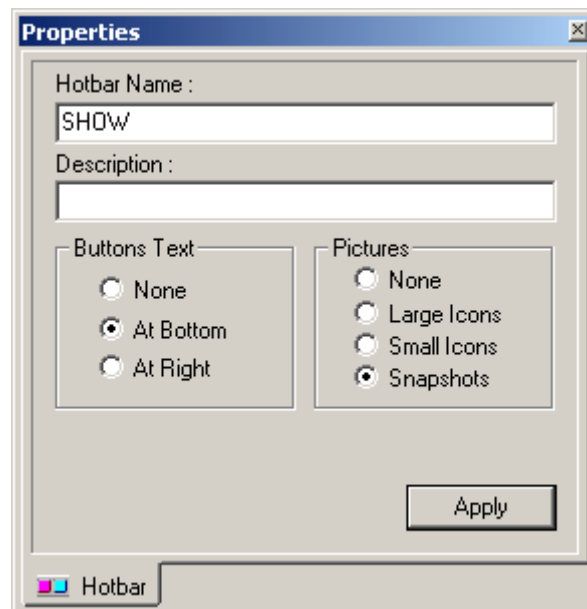
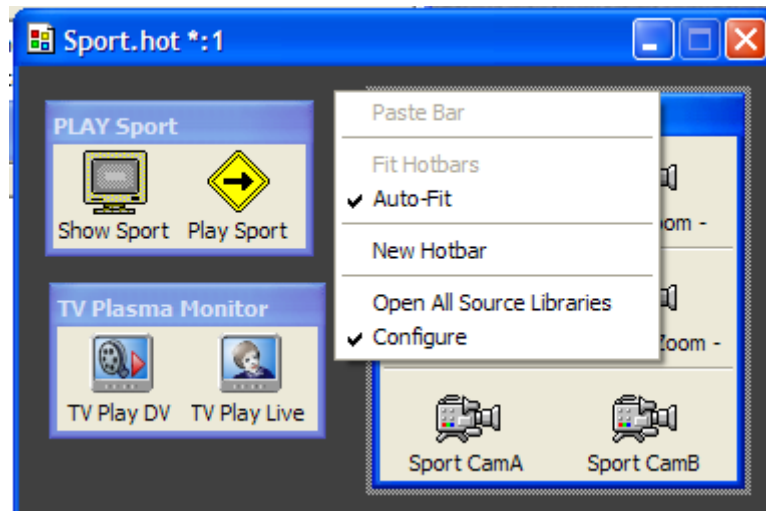
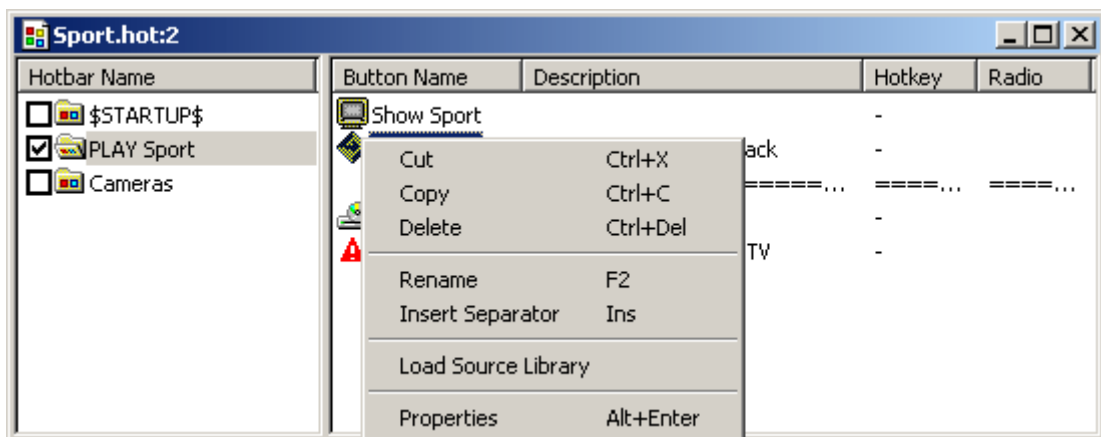


Figure 43. The *Hotbar* tab of the *Properties* dialog box

### 3.6.6 An Alternative Document-Like Presentation of the *Hotset*

You can use an alternative document-like presentation to configure the *Hotset*.

The document contains one or several folders corresponding to the *Hotbars* (Figure 45). To render the *Hotset* to this presentation, double-click it on an empty space or enable the **Configure** option in the *Hotset* context menu (Figure 44).

Figure 44. The *Hotset* context menuFigure 45. The *Hotset* document-like presentation and the context menu for a reference

To distinguish between windows of the *Hotset* different presentations when both windows are minimized, their titles are additionally marked with numbers '1' and '2' after the colon.

All the folders are displayed in the left part of the document window, and the contents of the selected folder are in the right part.

When creating a new folder in this presentation of the *Hotset*, the *Hotbar* with the same name is created automatically. An individual *Hotbar* can be shown/hidden by checking/unchecking the corresponding folder of the *Hotset*.

The contents of folders are references to the *Actions* corresponding to the *Hotbar* buttons. The order of references corresponds to the order of buttons. It is possible to add references to folders (and buttons to *Hotbars*) by dragging *Actions* from an open library, references from other folders, or buttons from other *Hotbars*.

References are manipulated in the same way as buttons: using the context menu that can be opened by a right click on a reference (Figure 45). For instance, a separator between some buttons of a floating *Hotbar* window can be added with the **Insert** key or using the **Insert Separator** command of the context menu for the button reference in the document-like presentation. The same command is available in the context menu for the button on the hotbar (Figure 39). A separator is added before the selected reference (button) and can be deleted by the **Ctrl-Del** key combination or the **Delete** command of the separator context menu on the hotbar.

References to the *Actions* that are executed at each scene initialization are located in the **\$STARTUP\$** folder available in every *Hotset*. Usually the *Startup Action* is placed into the folder, and the *Hotbar* corresponding to the folder is hidden. See Section 3.4.6 about *Startup Actions*.

### 3.6.7 The Main Window Menu When Working with the *Hotset*

When working with the *Hotset*, the **Hotset** special menu is added to the main application window toolbar (Figure 46). The menu is identical to the *Hotset* context menu and contains the following commands:

**New Hotbar** – creates a new *Hotbar*.

**Fit Hotbars** – modifies the *Hotset* proportions so that all the *Hotbars* may be entirely visible within the bounds of the *Hotset*. That doesn't influence the *Hotbars* overlapping because of which some of the *Hotbars* are not visible at all. The command is available when the **Auto-Fit** option is disabled since it is the one-time equivalent of the option.

**Auto-Fit** – enables automatic control of the *Hotset* size. All the *Hotbars* are always entirely visible, except for their overlapped areas.

**Open All Source Libraries** – opens all the *Action Libraries* referred to in the *Hotset*.

**Background Color...** – opens a dialog box for selecting a color for the *Hotset* workspace (the background for placing *Hotbars*).

**Configure** – opens the *Hotset* document-like presentation containing one or several folders corresponding to the *Hotbars* (see Section 3.6.6). This can also be done with a double click on an empty space in the *Hotset*.

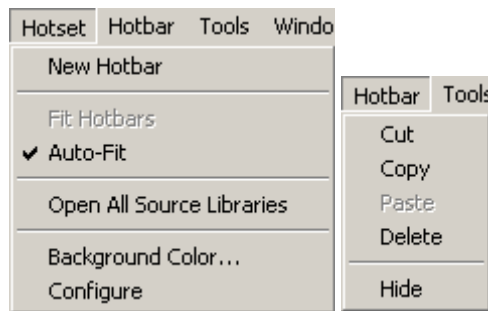


Figure 46. The *Hotset* and *Hotbar* menus of the main application window

Besides, the **HotBar** menu (Figure 46) is also added to the main application window when working with the *Hotset*. It contains the following commands for controlling *HotBars*:

**Cut** – sends the selected *HotBar* to the clipboard;

**Copy** – copies the selected *HotBar* to the clipboard;

**Paste** – pastes the selected *HotBar* from the clipboard to the *Hotset*. The command is available if there is a *HotBar* in the clipboard after one of the previous commands;

**Delete** – deletes the selected *HotBar*.


The commands of the menu are identical to the commands of the context menu for a *HotBar*.

## 3.7 Operations with the *Project*

### 3.7.1 Operations with the Project File

The *HotActions* application works with only one project at a time. When trying to open another project, the application displays a request to save changes in the current project, closes it and then opens the other one.

Existing projects are opened from the **File** menu of the main application window. Select the **Open Project...** command and specify the path to the required project or select it in its folder in the *Open* dialog box (Figure 13). After a project has been opened once, its name is added to the local menu of the **Recent Project** command and it can be opened from the menu.

New projects as well as the other files the program works with can be created in the dialog box opened by the **New** command of the **File** menu (Figure 48), with the **Ctrl-N** key combination, or the  button on the window toolbar. In the **New** dialog box, select the **VS Project** in the **File type** field (Figure 47), specify the new project name and a folder to save it.

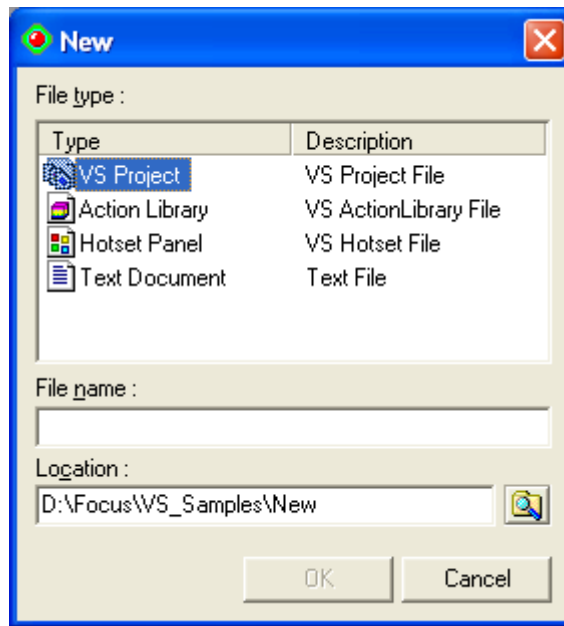


Figure 47. The New dialog box

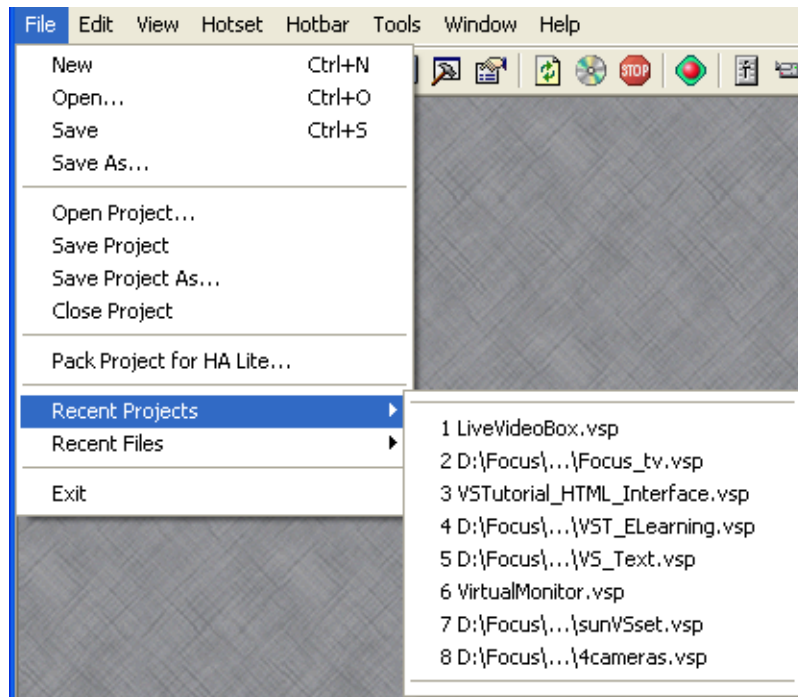





Figure 48. The File menu of the main application window when working with the project

 The open project can be saved via the **File** menu. If there is a project open, the **File** menu is extended by the **Save Project** and **Save Project As...** commands as well as the **Close Project** command (Figure 48). The  button can also be used to save the project if its window is active. If a project file is moved, for example, to another folder, without **HotActions**, the references to all the files included in it become invalid, and you will have to specify again the path to each file when it is opened. To move a project along with the paths to its files, use the **Save Project As...** command, all the references to the files being updated automatically though the files themselves remaining on their locations. Another way is to save all the files of a project in its local folder and move the whole folder with all the contents.

 The **Pack Project for HA Lite...** menu command allows transforming a project for working in the **Hot Actions Lite** application.

### 3.7.2 Working with Files and File Groups in the Project

For general information about working with files in the *HotActions* application, see Section 3.1.7.

Usually all the files included in a project are opened automatically when the project is opened. Any document file can also be loaded with a double click on its name in the project window or using the **Open** command of the project context menu (Figure 49). The names of the loaded files in the project window are in bold type.

The **Close** command of the context menu closes the file, the **Hide** command hides the document window, and the **Remove** command deletes the file from the project. The **Properties** option displays the file properties dialog box (Figure 52).

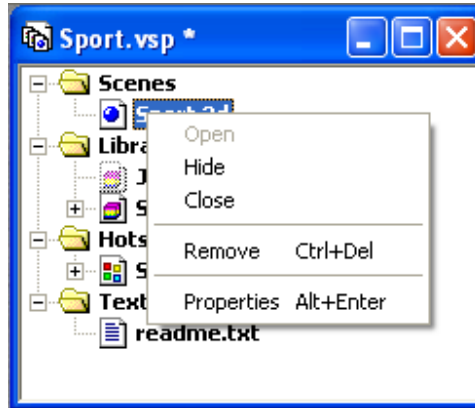


Figure 49. The context menu of a file in the project window

When opening any file (but the project file), the application prompts the user to add the file to the project if there is a project loaded and the file is not included in it.

To add a file (of a type *HotActions* works with) to the open project, use the **Add File to Group** command of the corresponding group context menu in the project window (Figure 51). To open a file, you can also use the **Open...** command of the **File** menu in the main application window. Specify the required file name in the *Open* dialog box. After that, the file can be opened directly with a click on its name in the **Recent Files** local menu of the **File** menu.

A window of each project includes **Standard Groups**, that is individual folders for files of each standard file type the *HotActions* application works with:

- The **Scenes** group for scenes (the **\*.3d** extension);
- The **Hotsets** group for *Hotsets* (the **\*.hot** extension);
- The **Libraries** group for *Action Libraries* (the **\*.acl** extension).

The *HotActions* application also supports standard operations with simple text documents (**\*.txt** files). When a text document is added or created, the **Texts** standard group appears in the project. A special group can be created for any file type, as described below.

When files are added to the project, they are automatically distributed among the proper groups. At attempt to add a file to a standard group of the wrong type, it still comes to the group of the proper type.

Standard groups for files *HotActions* work with cannot be removed or renamed unlike **Custom Groups**, which can be created via the context menu (Figure 50) opened with a right click on an empty space of the project window. Any **Custom Group** can contain references to files of different types at the same time.

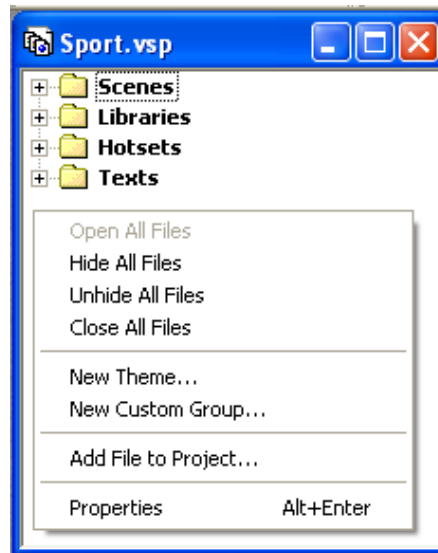


Figure 50. The project context menu

The **Open All Files** and **Close All Files** commands of the project context menu respectively open and close all the files included in the project.

The **Unhide All Files** and **Hide All Files** commands show/hide the document windows of all the files included in the project.

The **New Theme...** command creates a new *Theme* for the project. In the dialog box that opens when selecting the command, it is possible to specify a new theme name and type. For details about themes, see Section 3.1.6.


The **New Custom Group...** command creates a new custom group. In the dialog box that opens when selecting the command, it is possible to specify a new group name or accept the default name suggested by the application.

The **Add File to Project** command adds to the open project a file of one of the types supported by the application. It is necessary to specify the file path in the dialog box opened by the command.

The **Properties** command opens the *Properties* dialog box for the last selected group (Figure 52).

A right click on a group opens the context menu (Figure 51) for operations with groups:

**Open Group** and **Close Group** commands open and close all the files included in the group.

 *If some working windows of the project files are a disturbance overlapping the workspace of the main application window, they can be hidden by the commands described below.*



The **Hide Group** and **Unhide Group** commands respectively hide/show the windows of all the documents included in the group.

The **Delete Group** and **Rename...** commands respectively delete or rename the selected group if it is a *Custom* one.

The **Properties** command opens the *Properties* dialog box for the group (Figure 52).

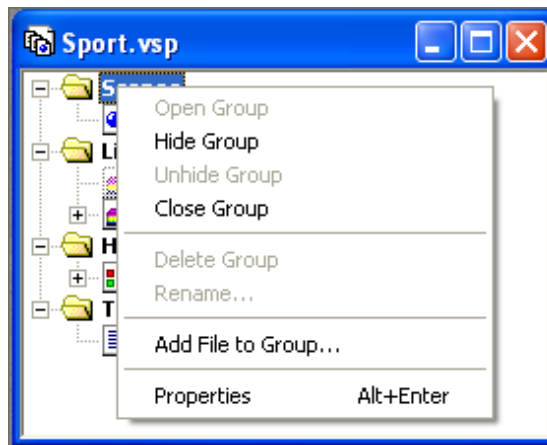


Figure 51. The context menu for a group in the open project window

The group can be renamed in the **File** tab (Figure 52, left) of the *Properties* dialog box if it is a **Custom** group. It is also possible to change the description of a group or file of any type in the tab. The **File** tab of the *Properties* dialog box additionally displays the full paths to the files and their project-relative paths (paths relative to the project file).

The **Directories** tab is used to specify the additional search paths for files in the current project. The tab relates to the whole project and is one for all the files and groups of the project (Figure 52, right).

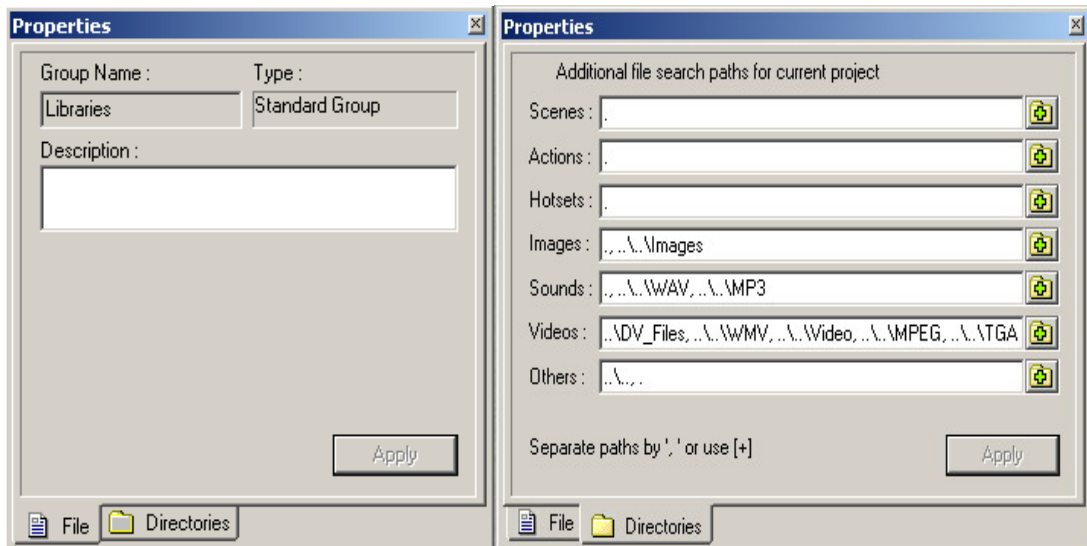


Figure 52. The *File* and *Directories* tabs of the *Properties* dialog box

If a file specified in the project is not found (for example, if moved or deleted), the application searches it using the following paths:

- in the current project folder, i.e., where the project file is located;
- in the folders specified as additional for specific file types in the project (Figure 52, right);
- in the folders specified in the application settings, in the **Default Directories** tab of the *Options* dialog box (see Section 9.2), where the full paths are specified for the current file type. The paths are identical for all the projects, that is, they are 'system' paths.

If a file is not found in any of the specified locations, the application displays an error message, like the one in Figure 53.

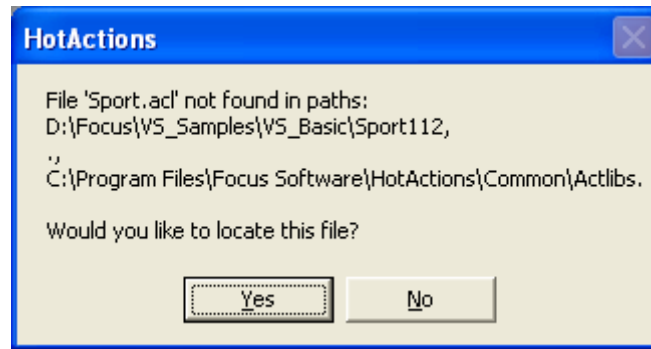


Figure 53. The *File not found* message

The **Yes** button opens the dialog box for specifying a new path. The new path is automatically added to the list of the current project additional paths and will be saved with the list.

### 3.7.3 The Main Window Menu When Working with the Project

When working with the project, the main application window menu is extended with the **Project** menu (Figure 54).

The commands of this menu duplicate the ones of the project context menu (Figure 50. The project context menu).

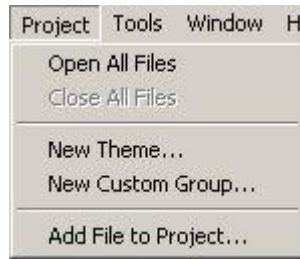


Figure 54. The main window *Project* menu for operations with the project

## 3.8 Operations with the *Theme*

A *Theme* is a description of the project variables (parameters) used in its *Actions* (see Section 3.1.6).

### 3.8.1 Operations with the File of a *Theme*

A new *Theme* is created by the **New Theme** command of the project context menu (which is opened with a right click on an empty space of the project window, Figure 50) or of the main window **Project** menu when working with the project (Figure 54).

Specify a name for the new theme in the **Theme name** field of the dialog box (Figure 55):

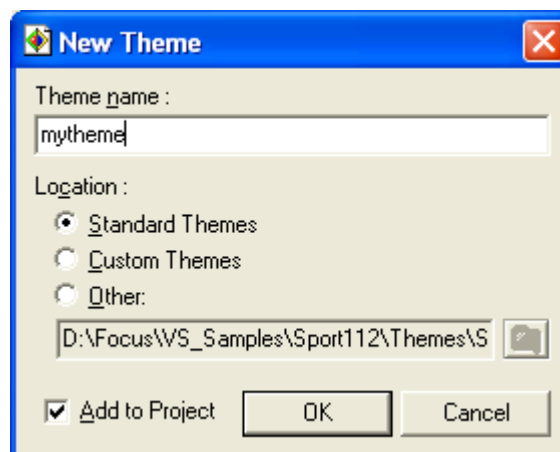


Figure 55. The *New Theme* dialog box

The new theme type is specified in the **Location** group.



The difference between the **Standard**, **Custom** and **Other** theme types is described in Section 3.1.6. The **Standard** and **Custom** themes differ in that only **Custom** themes can be edited and saved in the *LiveAction* mode. In the **Theme** drop-down list of the **Project** toolbar, the names of the **Standard** themes begin with an asterisk (\*) (see Section 3.2.5). The **Other** themes belong to the **Custom** type as well but such files can be created in an arbitrary folder. When creating a theme of the type, the button of selecting a folder for saving it becomes available. If such a theme is added to the project, the specified path is added to the **Others** field in the additional search paths of the current project (Figure 52).



All the project themes are available for selection in the *LiveAction* mode (see Section 3.1.9). Only one theme can be loaded (opened) in the project. If it is not specified explicitly, the application loads the default theme (**Default** in the **Theme** drop-down list, see Section 3.2.5), which is the last saved or last selected one.

When adding a new theme to the open project, use the context menu for the **Themes** standard group in the project window if there is a theme.

A previously created theme can always be opened in the *Open* dialog box. To do so, select the **Open...** command in the **File** menu of the *HotActions* window, specify the *Theme files (\*.vst)* file type in the *Open* dialog box (Figure 13), select the required file, and click the **Open** button.

After the operations, the application displays a confirmation request to include the file in the current project. If you want to open the theme without including it into the project, for example, to copy some variables from it, click **No**.

After a theme is opened once, it can be opened directly like any other file via the **Recent Files** submenu of the same **File** menu.

To save changes in the theme, select the **Save** command of the same **File** menu (when the theme window is active) or click the equivalent  button on the **Project** toolbar (see Section 3.2.5). The same operation is performed with the **Ctrl-S** key combination. To save the theme under a different name, use the **Save As...** command. If the current theme belongs to the **Standard** type, the application prompts the user to save it as a **Custom** theme. A **Custom** theme can be deleted with the  button on the same toolbar.

### 3.8.2 Operations with the Theme Variables

As was described in Section 3.1.6, if there is a variable name of the loaded theme in the commands of an *Action*, the name is replaced with the variable value when starting the *Action* (so-called macro substitution).

A right click on an empty space of the theme window opens the context menu (Figure 56). To create a new variable, select the **New Variable** item of the menu.



Figure 56. The context menu of a theme document

When working with the theme document, the main application window menu is extended with the **Variable** menu (Figure 57).

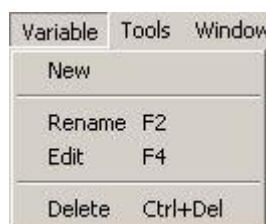
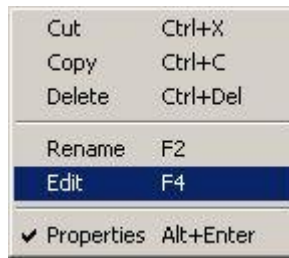


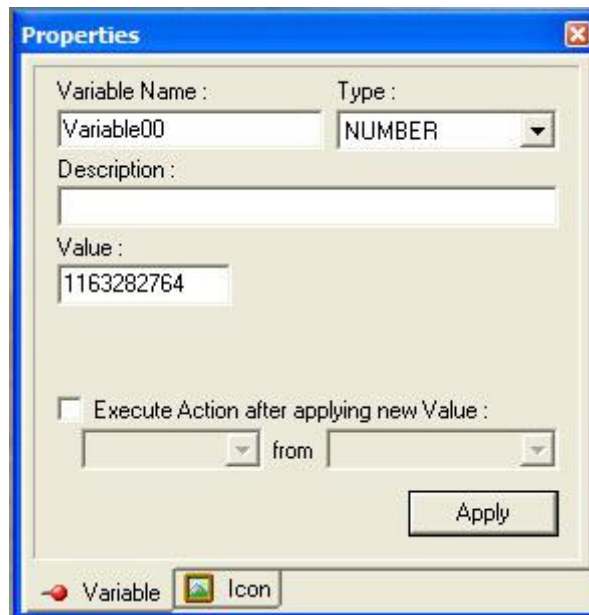
Figure 57. The *Variable* menu of the main application window when working with theme variables

The context menu of a variable is as follows:



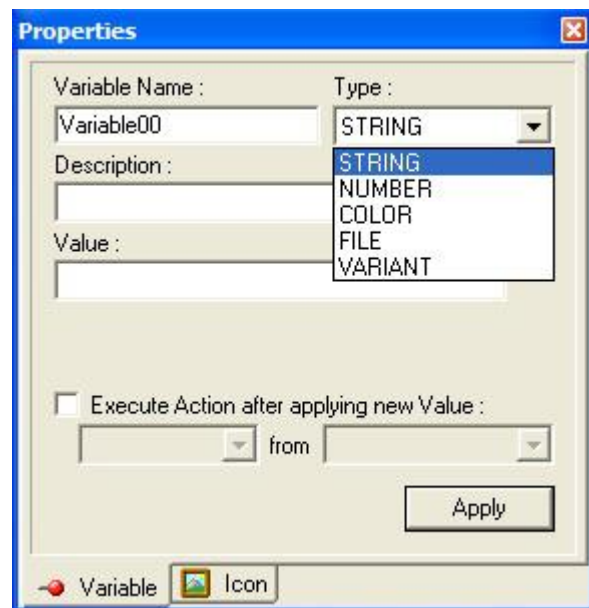
**Figure 58. The context menu of a variable**

Right after a new variable is created, it is necessary to specify its properties (in particular, its type and default value). The *Variable* tab of the *Properties* dialog box (Figure 59) can be opened by the **Properties** command of the context menu or with the **Alt-Enter** key combination.



**Figure 59. The Variable tab of the Properties dialog box**

The variable type, a most important property, is specified in the **Type** drop-down list of the *Variable* tab, the *Properties* dialog box (Figure 60).



**Figure 60. The Type drop-down list of the Variable tab**

The variables can be of the following types:

- **STRING** – a string value. Newly created variables belong to the type by default;
- **NUMBER** – a numerical value;
- **COLOR** – a chromatic variable;
- **FILE** – a file name variable, with the possible file extensions mask;
- **VARIANT** – the variable allows selecting from a set of predefined values (of the 'string' type).

According to the selected variable type, the tab modifies its appearance and functionality. For example, for the **COLOR** variable type, the tab looks like in Figure 61. A click on the **Value** color field in the tab opens a standard color selection dialog box.

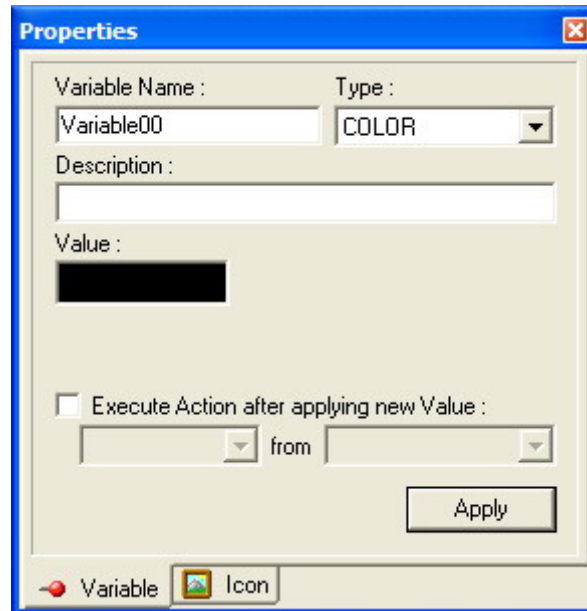



Figure 61. The *Properties* dialog box for a variable of the **COLOR** type

A value of the **FILE** type variable is selected in the *Open* dialog box (opened by the  button) by the mask specified in the **File Extensions** field of the *Variable* tab, *Properties* dialog box (Figure 62). The possible file extensions are separated by a comma or a space.

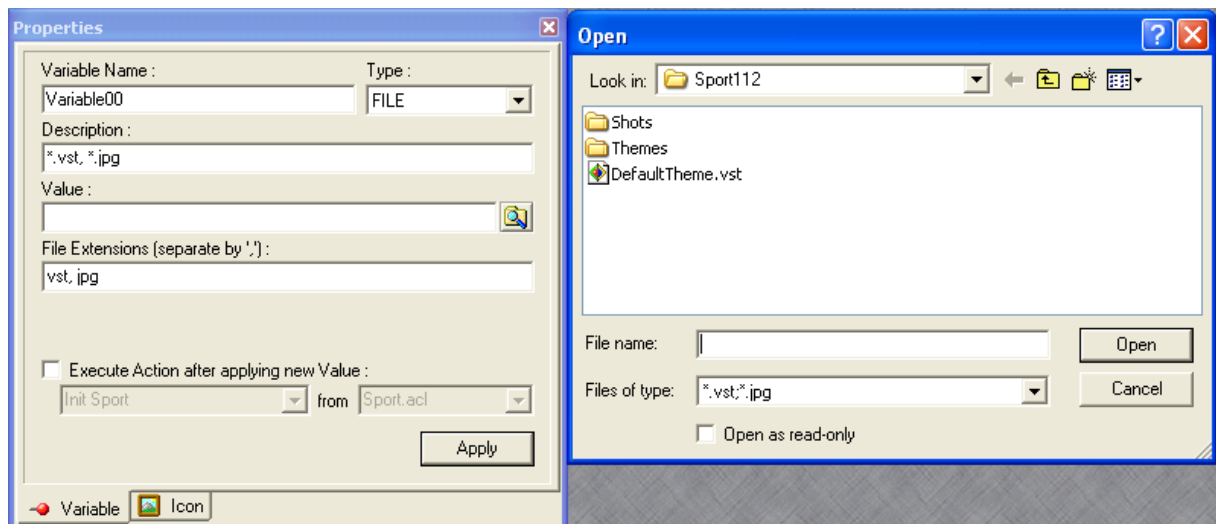


Figure 62. Selecting a value for a variable of the **FILE** type

The possible (alternative) values of the **VARIANT** type variable are separated with line feeds (**Enter**), it being possible to specify any initial value in the **Value** field (Figure 63).

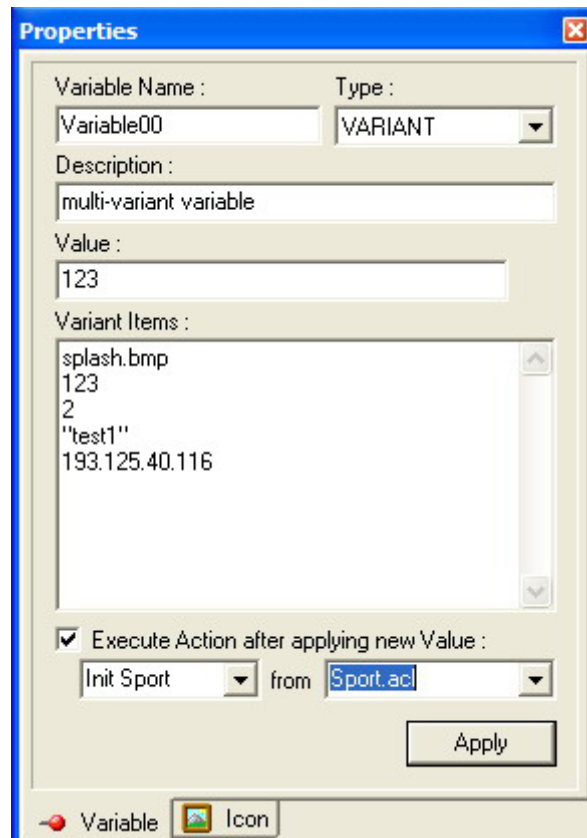


Figure 63. Editing a variable of the **VARIANT** type

The variable description is specified in the **Description** field of the *Variable* tab. When a description is specified and applied, it is displayed in the document window on the right of the variable name.

Click **Apply** (Figure 63) to save the changes in the variable properties.

The **Edit** command of the context menu (Figure 58) is used to edit the variable value. A double click on the variable is equivalent to the command. The dialog box for editing the value depends on the variable type. The standard file open dialog box and color selection dialog box are used, respectively, for the **FILE** and **COLOR** types of variables, and a dialog box with a drop-down list of possible values is used for variables of the **VARIANT** type (Figure 64).

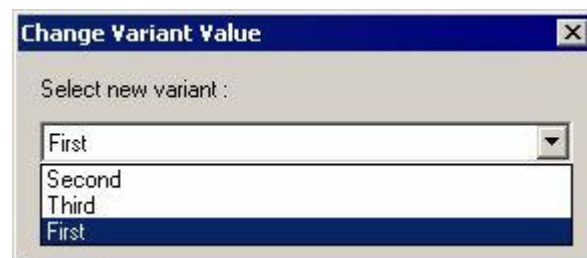



Figure 64. Editing a variable value of the **VARIANT** type


 Particularly note that in the **LiveAction** mode, editing any theme variable value is possible only using the dialog box opened with a double click on the variable name in the theme document window.

### 3.9 Operations with Text Documents

The *HotActions* application supports standard operations with simple text documents. That may be useful for the description of working with a specific project or for various comments.

Like files of other types processed in the application, text documents (files with the **\*.txt** extension) can be opened either automatically as belonging to the project or independently. When adding or creating a text document, the **Texts** standard group appears in the project (see Section 3.7.2).

To open a file individually (not as a part of the project), select the **Open...** command of the **File** menu and in the *Open* dialog box (Figure 13) select the file name from the required folder having set the **\*.txt** extension in the file types drop-down list. After that, the file can be opened directly by a click on its name in the **Recent Files** local menu of the **File** menu. When opening a file, the application always prompts the user to add it to the project, if there is one loaded.



To save the changes in a file, select the **Save** command of the **File** menu or click the equivalent  button on the toolbar (when the text document window is active). To save the file under a different name, with a different extension, different location, etc., use the **Save As...** command.

A new text document is created by selecting **Text Document** in the *New* dialog box opened by the **New** command of the **File** menu in the main application window or with the **Ctrl-N** key combination.

## 4 Controlling the Application Modes of Operation

As was mentioned above (see Section 3.1.9), the application has two modes of operation: *Edit* and *LiveAction*.

The application is launched in the *Edit* mode with two working windows: the main window and *Render Output* window (Section 7).

The application is switched to the *LiveAction* mode by clicking the **Start LiveAction**  button on the *LiveAction* toolbar of the main window (see Section 3.2.4) or its equivalent on the *General* toolbar (see Section 3.2.3). The button icon changes to . A second click on the button switches the application back to the *Edit* mode.

When switching to the *LiveAction* mode, the main working window switches to full-screen mode by default, and the rendering monitor displays the scene image if there is a project loaded. When not, the TV monitor displays an image from the EmptyRender.bmp file located in the folder C:\Program Files\Focus Software\HotActions\Common\Images folder. The same image is displayed on the monitor when finishing the work with the project, when switching from the *LiveAction* mode to the *Edit* mode, and when stopping the application (if the image is rendered without the DVM module (see Section 11)). The same image is displayed in the *Render Output* window (Section 7) in the *Edit* mode in the absence of a loaded scene.


When changing projects, the TV monitor displays an image from the LoadingProgress.bmp file from the same folder C:\Program Files\Focus Software\HotActions\Common\Images.

If you want to have different images, the required \*.bmp files must be placed to the folder C:\Program Files\Focus Software\HotActions\Common\Images and renamed accordingly: EmptyRender.bmp or/and LoadingProgress.bmp.

The settings of switching to the *LiveAction* mode can be changed in the *Options* dialog box (see Section 9.1). A click on the close button of a window (a cross in the top right corner) in the *LiveAction* mode doesn't close the document (file), but only hides its window.

The **Ctrl-Alt-Shift-Home** and **Ctrl-Alt-Shift-End** key combinations are also used to switch to the *LiveAction* mode and back. The combinations can be handy in a one-monitor configuration.

The context menu opened by a right click on an empty client area of the main window also includes commands for controlling the *LiveAction* mode. The commands as well as the rest of the buttons of the main window toolbars are described in Section 3.2. Having tried them, you can choose the handiest of them.

The *Render Output* window can be switched to the **Fullscreen** mode and back by the  button of the *General* toolbar (see Section 3.2.3) or with the **Ctrl-Shift-F** key combination. The operation does not imply switching the whole application to the *LiveAction* mode. Switching to the **Fullscreen** mode is also performed with the **Ctrl-Shift-Home** key combination, and switching back – the **Ctrl-Shift-End** combination. See Section 9.1 for information about the **Fullscreen** mode settings for the *Render Output* window.

## 5 Using the Joystick and Mouse to Control Objects


The scene objects can be controlled (dynamically rotated and scaled) by the joystick. In the *Edit* mode (see Section 4), the scene objects in the *Render Output* window can also be manipulated using the mouse and the keyboard keys (see Section 5.2). And when working with two outputs of the video adapter in an HD Studio configuration (Section 11), controlling the objects with the mouse and the keyboard keys in the *LiveAction* mode is possible as well.

### 5.1 Initializing the Joystick

To use a joystick, put a special *Action* named **Init Joystick** from the **Joystick.acl** standard library in the **\$STARTUP\$** of the *Hotset* (see Section 3.4.6). To do so, it is recommended to add the library to the project first (see the instructions for the operation in Section 3.5.1). The library is specially designed to work with the joystick and is located by default in the **\Common\ActLibs** subdirectory of the *HotActions.exe* application folder (the folder **C:\Program Files\Focus Software\HotActions\Common\ActLibs** if the virtual studio is installed on the **C** drive in the standard way).

Another way to initialize the joystick is direct inclusion of joystick initialization commands copied from the same *Action* named **Init Joystick** of the **Joystick.acl** standard library in the *Startup Action* for a particular scene.

Joystick sensitivity by axes and its responses to clicking/releasing/holding specific buttons are specified in the joystick initialization commands. Note that in the **Init Joystick Action**, the object for manipulation is always the current selected node in the scene, since all the commands refer to the node named **<CURRENT>**. The description of specific commands see in *Using the Script Commands. User's Guide*, enclosed with the software package.

 *The two foregoing ways of the joystick initialization can be effectively combined as follows: the **Init Joystick Action** initializing the joystick on the whole must be included in the **\$STARTUP\$** folder of the *Hotset*, and only the commands specifying the joystick sensitivity must be added to the *Action* initializing a specific scene. In this case make sure that the *Startup Action* of the scene **FOLLOWS** the **Init Joystick Action** in the **\$STARTUP\$** folder (see information about the order of Actions in Section 3.4.6). The order is necessary for the commands specifying the joystick sensitivity to be executed after the joystick initialization. The described method is convenient to specify different sensitivity of the joystick for different scenes, which may be required, for example, because of the difference in the scaling units of the scenes.*

### 5.2 Selecting a Manipulation Object in the *Edit* Mode

Usually after a scene is opened (for example as a part of the project), the object (node) selected for manipulation by default is the scene node named “**World**”. It is possible to select any other object for manipulation, for example, a camera with a double click on it in the scene nodes window (Figure 19).

The name of the node selected for manipulation is in bold type in the scene window. There is a special system name **<CURRENT>** for using in the script commands. The name always indicates the currently selected node. When changing the manipulation object, all the commands addressed to the **<CURRENT>** object are executed to the object selected now.

Using some additional keys of the keyboard allows enlarging the variety of operations that can be performed on an object selected for manipulation:

- **Tab** cyclic switching from one axis to another at rotation/movement/scaling;
- **+ (plus)** increasing the speed of the object onward movement (positive direction along the selected axis, Y by default) starting with zero value (position of rest);
- **– (minus)** decreasing the speed of the object onward movement (for an object moving in the positive direction of the movement axis) or increasing the speed if the object is moving in the negative direction of the selected axis or is not moving; the position of rest is zero value;
- **Space** stops the object movement, sets the movement speed equal to zero;
- **Ctrl+R** resets, restores the object to the initial state (as the one after the scene is loaded).

- **Modifiers:**

- **Ctrl** the object is scaled along the current axis with a mouse gesture or joystick movement, or arrow keystroke;
- **Alt** the object is moved along the current axis in the scene area.


If a selected object is a virtual camera, the following keys are also used (see description of the *Camera* tab, the *Properties* dialog box for a camera in the scene, Section 3.4.5):

- **PgUp / PgDown** increases/decreases the focal distance (**FOV**). The **Mouse Wheel** scrolling performs the same operation.
- **Home** sets the camera focal distance (**Zoom**) to the default position;

Objects can also be manipulated in the **Node** tab of the *Properties* dialog box (see Section 3.4.5).



## 6 Using the *Debug Output* Window

The *HotActions* application has the ability to track diagnostic, error, and warning messages, the course of executing the script commands and other events. The *Debug Output* window is used for this purpose. It can be opened by enabling the **Debug Output** option of the **View** menu or using the **Ctrl-Shift-0** key combination. Besides, you can use the  button on the *General* toolbar (see Section 3.2.3) of the main application window:

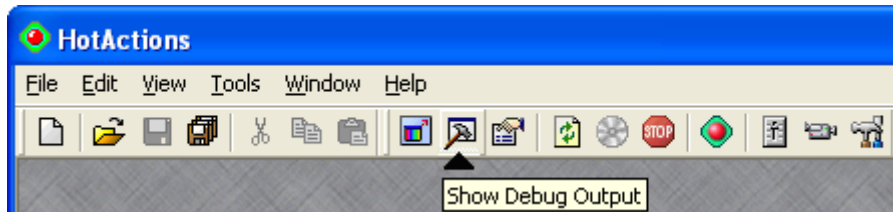



Figure 65. The *Debug Output* button on the toolbar

The type of messages displayed in the *Debug Output* window can be specified in the *Debug Output* tab, the *Options* dialog box of the application general settings opened by the **Options...** command of the **Tools** menu or the **F10** button. See Section 9.3 for more details about the tab settings. Whether the *Debug Output* window is open or closed, the messages of all the selected types are registered in the log file, whose is specified in the application general settings (see Section 9.3). The file can be sent to the developers to receive technical support when necessary.

By default, the *Debug Output* window is docked to the left bottom edge of the main application window (*Docking View*). The window can also be displayed in any part of the main window. That can be achieved by dragging it at the left bound keeping the **Ctrl** key pressed, using the  button in the top left corner of the window, or by switching off the **Docking View** command of the context menu (Figure 66). To dock the *Debug Output* window back to a border of the main application window, drag it to the border and double-click it, or select the **Docking View** command in the context menu of the window title bar.

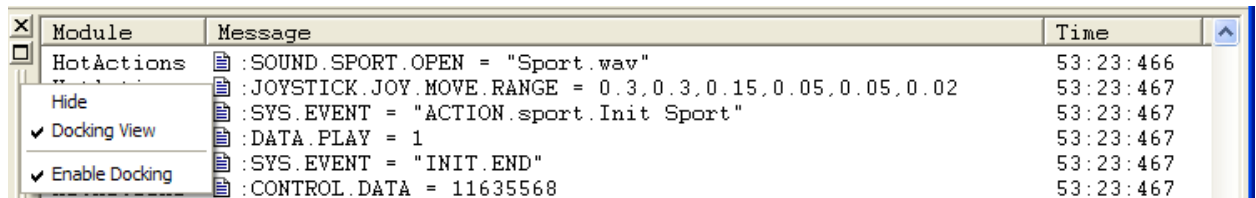



Figure 66. The *Debug Output* window title bar context menu

The **Enable Docking** option should be enabled in the context menu to make the docking of the window possible. The **Hide** command or clicking the  button hides the window.

The commands of the *Debug Output* window context menu (Figure 67) allow performing the following operations:

### Show Columns

**Sending Time** show the column of the message receiving time, time being counted from the start of the application session in the “hours : minutes : seconds : milliseconds” format (the **Time** column);

**Module Name** show the column with the name of the virtual studio software component from which the message is received in the *Debug Output* window (the **Module** column);

**Copy** (the **Ctrl-C** key combination) copy to the clipboard the lines selected in the window. Later, the lines can be inserted from the clipboard to any text application, for example, to the Notepad or an e-mail message. Selecting lines is performed in a standard way when working with files in *Windows*: left click with the **Ctrl** or **Shift** keys pressed. The **Ctrl-A** key combination is used to select all the messages in the window;

**Clear** delete all the messages from the window.

The icons displayed in the **Message** column on the left of the text (i, x, 📄, etc.) indicate the type of the message: information about the loaded data and modules, error message, command execution message, etc. For more details about the message types, see Section 9.3.

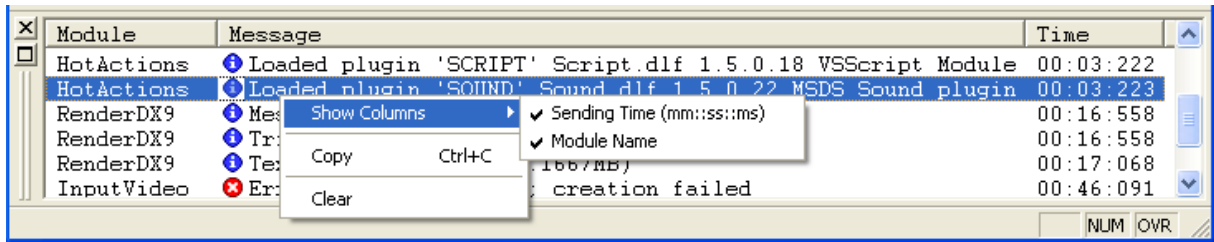



Figure 67. The *Debug Output* window context menu

## 7 The Render Output Window

The *Render Output* window is designed to display the results of the studio operation (the synthesized 3D graphics image with real-time video streams). When switching to the *LiveAction* mode, displaying the results of the studio operation is moved from the *Render Output* window to a TV monitor in the **Fullscreen** mode.

The *Render Output* window can be switched to the **Fullscreen** mode and back by the  button on the *General* toolbar (see Section 3.2.3) or using the **Ctrl-Shift-F** key combination. The operation does not imply switching the whole application to the *LiveAction* mode. Switching to the **Fullscreen** mode is also performed with the **Ctrl-Shift-Home** key combination, and switching back – with the **Ctrl-Shift-End** key combination.

The graphics processor must provide the necessary performance when displaying the 3D scene. The processor load level can be seen if enable the load indication with the **Shift-~** key combination. To estimate the load, you are presented with two indicators (Figure 68). The first one (left) indicates the number of the video streams being played in the scene (live video or files), a scale division of the abscissa axis corresponding to one video stream. A scip in the indication presents non-renewed videotexture, which may indicate a dropped frame. To increase the number of the scale divisions, execute the `RENDER.INFO.STREAMS = iStreams` command, where *iStreams* is the number of the video streams being played (see *Using the Script Commands. User's Guide*).

The second one indicates the time of the scene displaying preparation in the 'frame' – 'the frame preparation time' coordinate system. The origin of coordinates is located in the left bottom corner of the indicator, and the red line on the right marks the allowable limit of indication when displaying the image. The ordinate axis (bottom-up) indicates the frames being displayed; the abscissa axis (left-to-right) – the frame preparation time in milliseconds, one scale division being the time of the half-frame generation (the reciprocal coefficient of the output monitor frequency; in the PAL mode – 20 ms; NTSC – 16.7 ms). The number of the indicator scale divisions is always by 2 more than the value displayed in **Back Buffers Count** of the *Render Options* dialog box (see Section 8). Red indicates the scene objects displaying preparation (the CPU operation time); blue – the data transfer from the CPU to the 3D graphics accelerator for rendering; violet – the rendering time (the 3D graphics accelerator operation); yellow – the frame rendering waiting time.

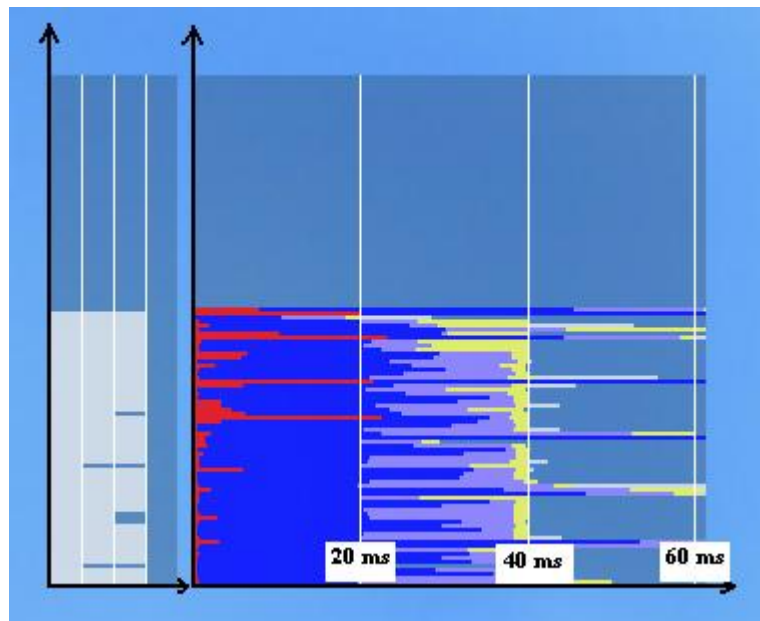


Figure 68. The 3D graphics accelerator load indicator when displaying the scene (PAL)

### 7.1 The Commands of the Main Application Window Menus for Working with the *Render Output* Window

The **Center Render Window** command of the *HotActions* main window **Window** menu (Figure 69) places the *Render Output* window in the centre of the application window.

The **Fullscreen Render** command of the **View** menu switches the *Render Output* window to full-screen mode and back. The **Render Window** command of the **View** menu (Figure 69, right) closes the *Render Output* window or opens if closed.

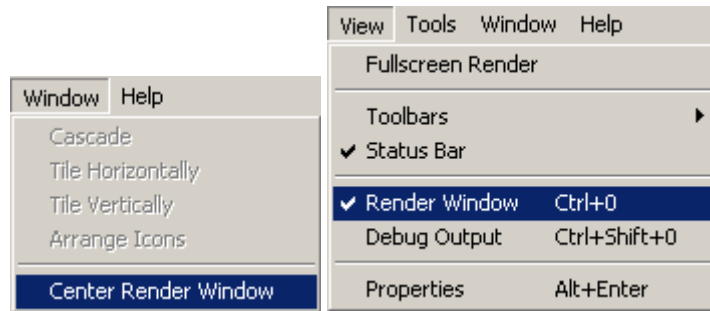



Figure 69. The *Window* and *View* menus of the main application window

## 7.2 The Menu Commands of the *Render Output* Window

The *Render Output* window also has its own menus. The **View** menu (Figure 70, left) of the *Render Output* window contains the following commands:

- **Frame Rate** displays information about the window size and some average statistical information about the studio performance for the currently displayed scene: the minimal time of calculation and displaying of one of the last four frames (in milliseconds);
- **Window Aspect** sets the window aspect ratio: **Custom** – arbitrary, **Standard** – 4:3, or **Widescreen** – 16:9;
- standard window sizes (**320x240**, **640x480**, and **768x576**) that can also be set by pressing the respective key combination (**Alt+1**, **Alt+2**, **Alt+3**);
- **Fullscreen (Ctrl+Shift+Home)** allows switching to full-screen mode, the entire application being not switched to the *LiveAction* mode. Switching to the **Fullscreen** mode is also performed with the **Ctrl-Shift-Home** key combination or the  button of the *General* toolbar (see Section 3.2.3). Switching back is performed with the **Ctrl-Shift-End** key combination.

The **Engine (F8)** command of the **Options** menu (Figure 70, right) opens a dialog box with information about the program displaying module used in the virtual studio (Figure 72). The **Render (F11)** command opens the *Render Options* dialog box with information about the displaying module settings (Figure 73). Both dialog boxes are described in the next Section.




Figure 70. The *View* and *Options* menus of the *Render Output* window

Using the **Tools** menu commands (Figure 71), the current image of the *Render Output* window can be printed (**Print...**) or saved to a BMP file (**Take Snapshot...** or **Ctrl+Shift+S**).

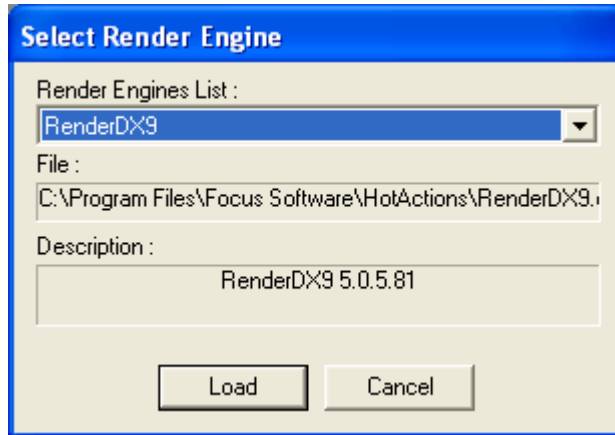


**Figure 71.** The *Tools* menu of the *Render Output* window

## 8 Dialog Boxes for Display Settings

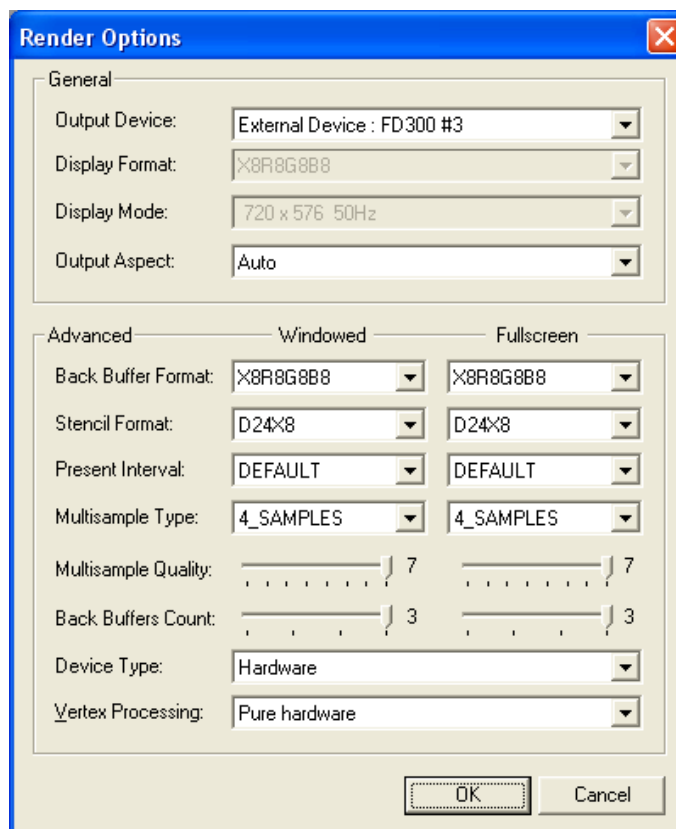
 It is not recommended to change the settings of the dialog boxes described below without special instructions of the technical support service. The settings are described for general overview.

The *Select Render Engine* dialog box displays information about the name, version and the path to the program displaying module used (Figure 72). It is opened by pressing the **F8** key on the keyboard or by the **Engine** command of the **Options** menu in the *Render Output* window (Figure 70, right ). The information can be necessary to the developers and the technical support service (Section 10.4).



**Figure 72. The *Select Render Engine* dialog box**

The *Render Options* dialog box (Figure 73) contains information about the settings of the image displaying module in the virtual studio. It is opened by pressing the **F11** key on the keyboard or by the **Render** command of the **Options** menu in the *Render Output* window (Figure 70, right).



**Figure 73. The *Render Options* dialog box**

The **General** area of the dialog box is used to set the general parameters of the resultant video image rendering.

The **Output Device** drop-down list is used to select a device for displaying the Studio resultant video image. The drop-down list may contain:

*Display 1* (with the name of the monitor connected to it) – rendering the result of the Studio operation is to be performed using the video adapter;

*Display 2: DVM62* – rendering through the *DVM62* (See Section 11) or a monitor connected to the second output of the video adapter;

*External Device: FD300 #1, External Device: FD300 #2, External Device: FD300 #3* (according to the number of the *FD300* boards installed in the system) – the image is to be rendered by the selected *FD300* board. The board number must match the logical index **L** of the output board in the *FDConfiguration* application (Section 2.3, Figure 4). Usually it is the largest number.

**Display Format** – is used to select a format of the image, that is, how many bits of the 32 are allocated to represent each chroma component R(8), G(8), and B(8).

**Display mode** – the image resolution and refresh rate.

**Output aspect** – the pixel aspect ratio – in the *Auto* mode (determined by the monitor settings), **4:3**, **16:9**, or *Square pixel* (1:1).

The **Advanced** area contains parameters for the image rendering fine-tuning: for window mode (*Windowed*) and full-screen mode (*Fullscreen*).

**Back Buffer Format** – a format of picture coding in bits for each chroma component (RGB);

**Stencil Format** – the number of bits of the 32 allocated to determine the Z order (D24) and used to store the object number in the scene (X8).

**Present Interval** – the frame rendering rate. When *DEFAULT*, the value is equal to the horizontal scan rate, as well as when *ONE*. The *ONE* mode differs from *DEFAULT* in more precision and activating more resources. In the *IMMEDIATE* mode, a frame is rendered on the monitor irrespective of its horizontal scan rate, the output time being the full rendering time.

**Multisample Type** – a mode of smoothing when rendering. When *NONMASKABLE*, *4\_SAMPLES* and *8\_SAMPLES* are selected, the level of smoothing can be varied by the **Multisample Quality** slider located below. The recommended value is 7 in the *4\_SAMPLES* mode;

**Back Buffers Count** – the number of half-frames stored into the buffer before rendering.

The **Device Type** drop-down list is used to select a type of the device which is to be used for the image rendering. *Hardware* – the graphics accelerator of the video card; *Reference* – the image is to be rendered in one of the *DirectX* modes (that is, the central processor of the motherboard is to be used).


**Vertex Processing** – is used to select a means of geometric transformations:

*Software* – transformations are performed by the central processor. Recommended when working with graphics processors of no high capacity and/or with scenes having a large quantity of dynamic objects (e.g., using morphing).

*Hardware* and *Pure hardware* – transformations are performed by the video card graphics processor, but when selecting *Pure hardware*, the data transmission of the transformation results is impossible, which allows to optimize the process in the best way. Optimal when working with scenes having a large quantity of static objects.

*Mixed* – the selection of a transformation means is performed programmatically from those described above.

## 9 The *Options* Dialog Box: the Virtual Studio General Settings

The **Options**  button of the *HotActions* main window toolbar opens the *Options* dialog box for adjusting the application settings.

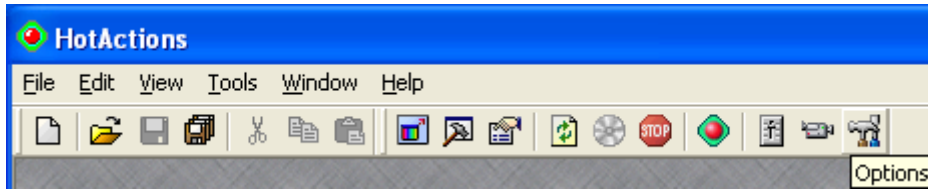


Figure 74. The Options button of the *HotActions* main window toolbar

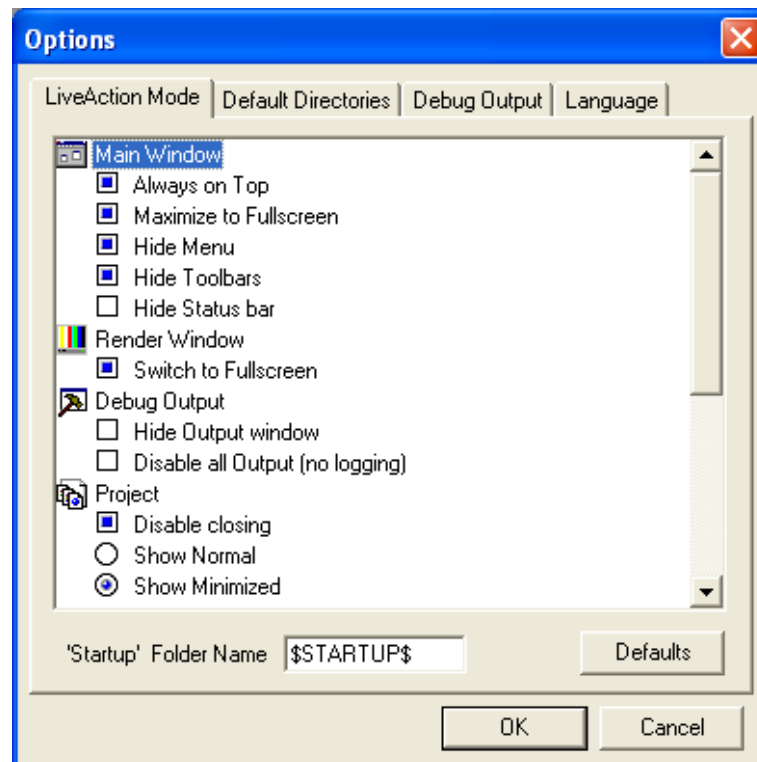



Figure 75. The *Options* dialog box with the application settings

### 9.1 The *LiveAction Mode* Settings

The **LiveAction Mode** tab (Figure 75 and Figure 76) of the *Options* dialog box is designed to adjust the *LiveAction* mode.

 *It is not recommended for inexperienced users to change the default settings. To restore them, click the **Defaults** button.*



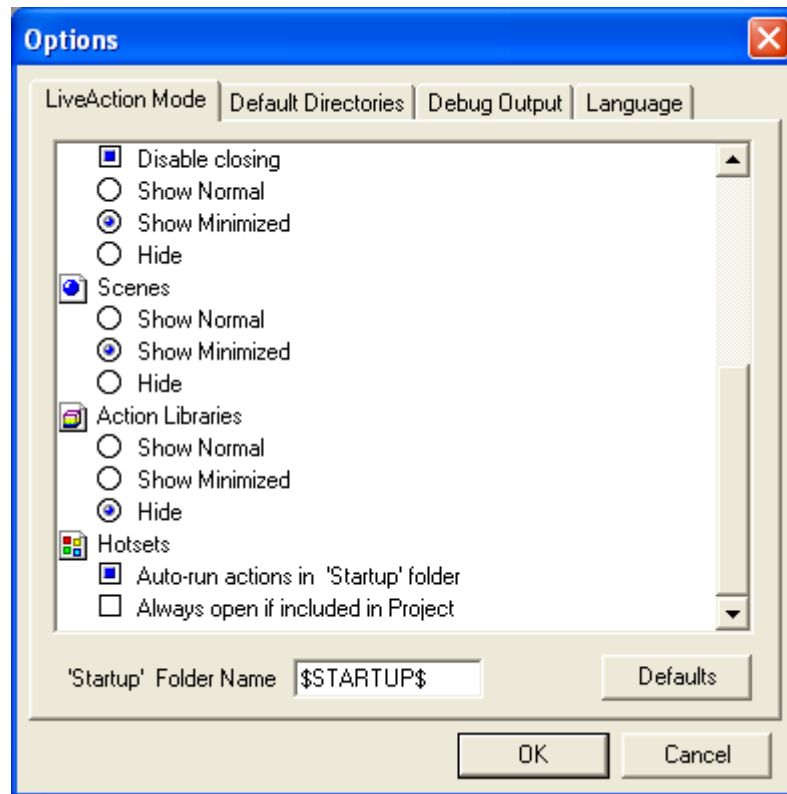


Figure 76. The *LiveAction Mode* tab of the application settings

You can set the following parameters of the *LiveAction* mode in the **Main Window** group, which refers to the main application window:

- **Always On Top** – always places the main application window above the windows of other applications;
- **Maximize to Fullscreen** – maximizes the main application window to full screen;
- **Hide Menu** – hides the menus in the main application window;
- **Hide Toolbars** – hides all the toolbars in the main application window except for the *LiveAction* and *Project* panels;
- **Hide Status Bar** – hides the status bar in the main application window.

The **Render Window** group (the result scene image output) allows adjusting the following parameters:

- **Switch to Fullscreen** – enables switching the *Render Output* window to the full-screen mode when switching to the *LiveAction* mode.

The **Debug Output** group parameters:


- **Hide Output window** – hides the *Debug Output* window in the operating mode;
- **Disable all Output (no logging)** – completely disables all the output to the log file (see Section 9.3).

The **Project** group parameters allow setting the project window view mode.

- **Disable Closing** – disables the ability of closing a project in the *LiveAction* mode;
- **Show Normal** – displays the project window in a usual way;
- **Show Minimized** – minimizes the project window;
- **Hide** – hides the project window.

The **Scenes** parameters group sets the scene windows view:

- **Show Normal** – displays scene windows in a usual way;
- **Show Minimized** – minimizes the scene windows;
- **Hide** – hides the scene windows.

 Note that the above parameters concerning the windows displaying mode are actual only when first switching the project to the **LiveAction** mode. After the project is switched to the **LiveAction** mode and then saved, its windows state in the mode is also saved in the project. Later the saved state has a higher priority than the parameters set in this tab.

The **Action Libraries** group parameters allow determining the size of the **Action Library** window in the **LiveAction** mode:

- **Show Normal** – displays the library window in a usual way;
- **Show Minimized** – the library window is minimized;
- **Hide** – the library window is not displayed.

The **Hotsets** group parameters determine the mode of operation with **Hotsets**:

- **Auto-run actions in 'Startup' folder** – enables automatically executing the **Actions** located in the **\$STARTUP\$** folder;
- **Always open if included in Project** – automatically displays (maximizes) windows of all the **Hotsets** when switching to the **LiveAction** mode.

The **'Startup' Folder Name** text field is used to set a name of the **Hotbar** folder containing the **Startup Actions** in the **Hotset(s)**. **Actions** in folders with such a name are executed automatically every time when switching to the **LiveAction** mode (see Sections 3.1.9 and 3.4.6).


The **Default** button restores the default values for all the parameters.

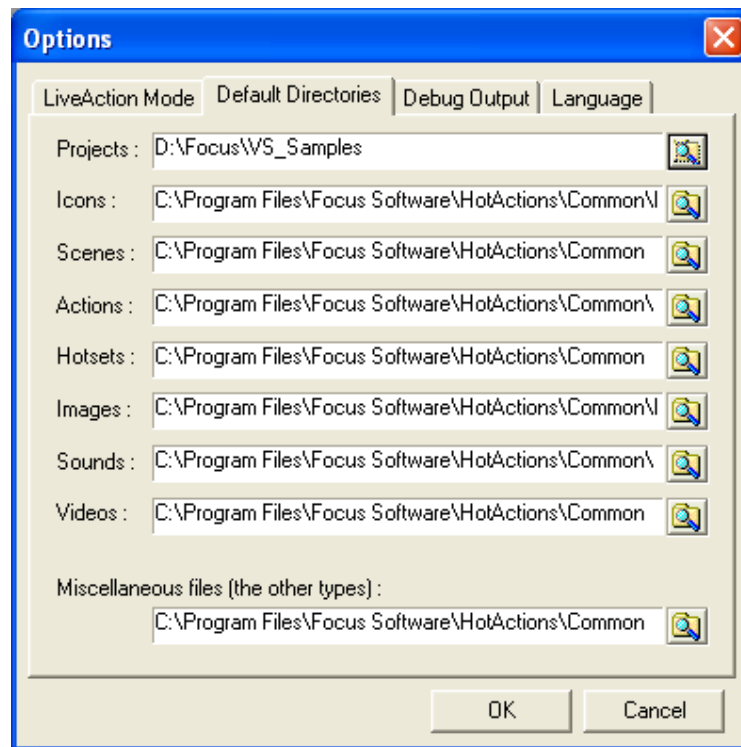
## 9.2 The *Default Directories* Settings

The **Default Directories** tab (Figure 77) is used to specify the directories for searching files of different types (projects, icons, media files, **Action Libraries**, etc.) used by the application when working with any project. Those are system paths for the application.

Files of any type used when working with projects are searched in the following order:

1. In the current project folder, i.e. in the folder where the project file (\*.vsp) is saved.
2. In the directories marked in the project as additional for specific file types (Figure 52, right). Note that if the additional directories are located on the same disk as the project, when saving the project, the search is performed by the project-relative paths.
3. In the folders of full system paths specified in the **Default Directories** tab for specific file types (Figure 77).

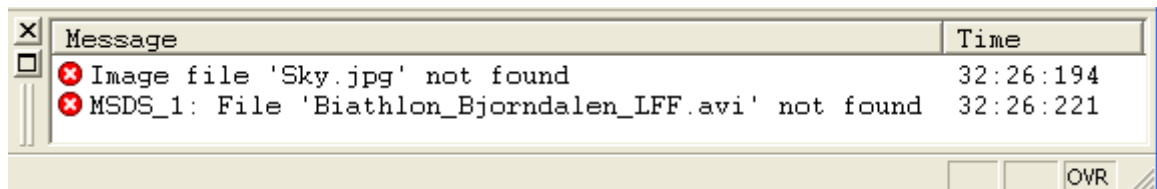
 It is very important to specify a directory correctly. Since only the file names attached to the project are saved in it, the described searching algorithm is used when opening any file included in the project.



**Figure 77. The *Default Directories* tab of the *Options* dialog box**

If the required file is not found by any path specified for it, the application displays an error message like the one in Figure 53. If specify the correct path to the file then, it is automatically added to the list of the project additional paths and saved when saving the project.

If the application can not find the graphics, audio and other media files it referred to by the script commands, it generates an error message as well. When the **Errors** option is enabled (see Section 9.3), the messages are displayed in the *Debug Output* window (Figure 78).



**Figure 78. Error messages: a file specified in the commands is not found**

A video file error message is also displayed if the file is damaged. For more detailed information about operations with media files, see *Using the Script Commands. User's Guide*.

### 9.3 The *Debug Output* Settings

The **Debug Output** tab of the *Options* dialog box (Figure 79) contains the control options of the messages that will be displayed in the *Debug Output* window when working with the application.

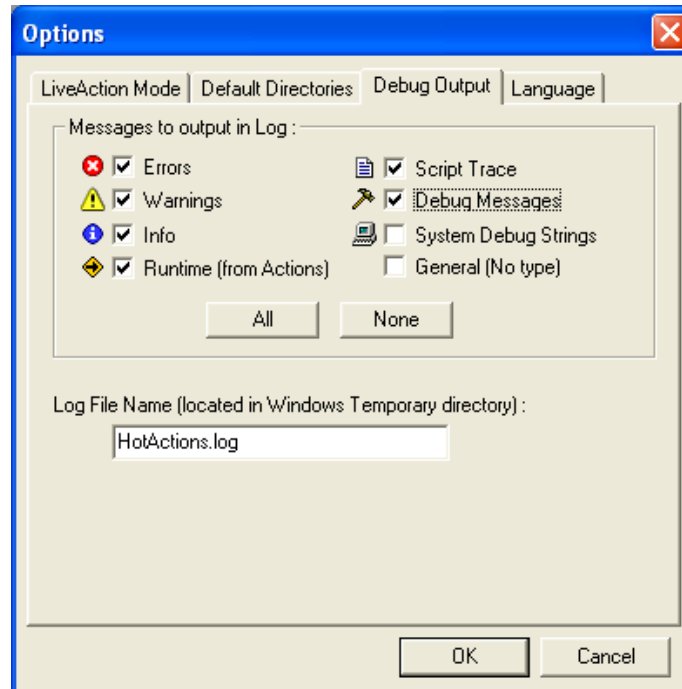


Figure 79. The *Debug Output* tab of the *Options* dialog box

According to the options selected in the tab, the following messages can be displayed in the *Debug Output* window:

- **Errors** – error messages;
- **Warnings** – non-critical warnings;
- **Info** – various information about the loaded data, program modules, etc.;
- **Runtime (from Actions)** – user messages about the execution of *Actions* (see *Using the Script Commands. User's Guide*, Section 8, the SYS.OUTPUT command for displaying such messages);
- **Script Trace** – information about the course of executing the *Action* commands. Used for debugging scripts;
- **Debug Messages** – various debug messages about the application operation (the option is designed mainly for the developers);
- **System Debug Strings** – the system debugging information (also designed for the developers): if the option is enabled, the application traps and displays all the debugging messages of the applications operating under *Windows* and using the OutputDebugString system command;
- **General (No type)** – all the other (not typified) messages.

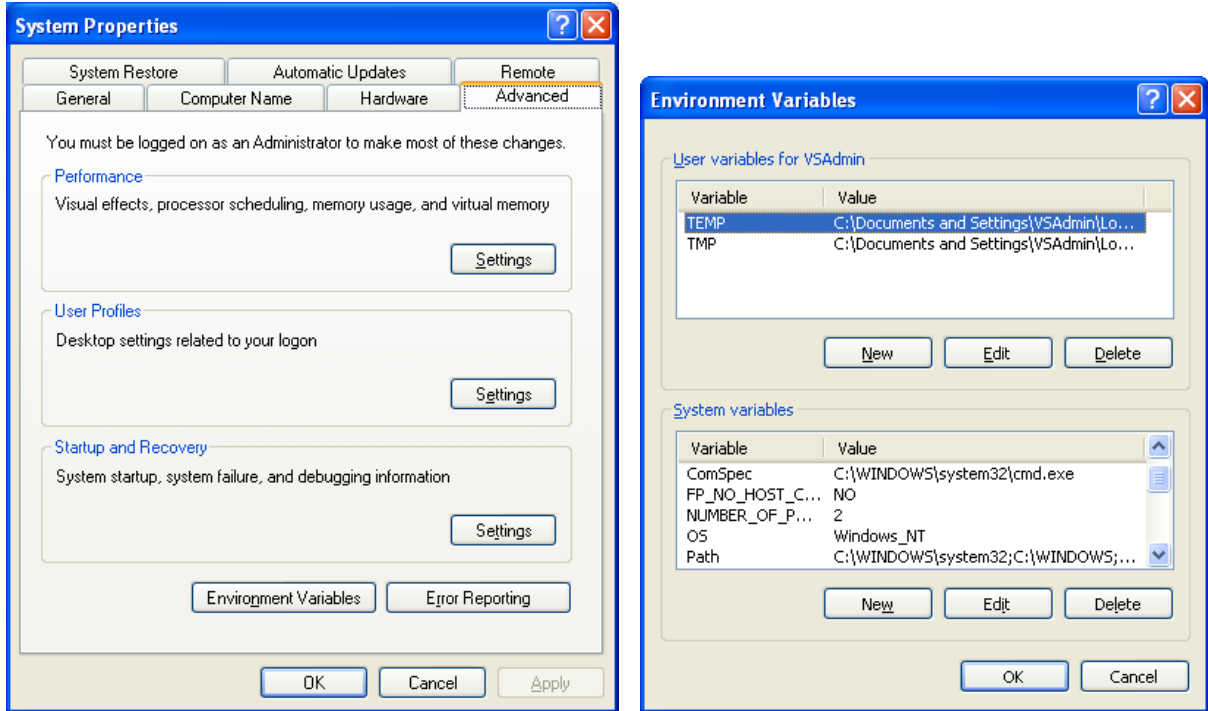
The **None** and **All** buttons allow enabling or disabling all the options simultaneously.

The icons on the left of each option correspond to the icons that will be displayed in the *Debug Output* window (see Section 6), on the left of the message text according to its type. Thus, the icons identify the types of messages.

The **Log File Name** field is used to specify the name of a log file for logging the messages of all the selected types. A text file with the name will be created in the *Windows* directory for the temporary files of the current user. The default filename is **HotActions.log**. Messages of all the selected types are registered in the log file when working in the application whether the *Debug Output* window is opened or closed. The path to the *Windows* directory containing temporary files of the current system user can be found in the *Environment Variables* dialog box. The dialog box is opened as follows:

1. Right-click on the **My computer** icon on the *Windows* desktop.

2. The context menu opens. Select the **Properties** command. The *System Properties* dialog box opens. Select the **Advanced** tab (Figure 80, left) and click the **Environment Variables** button.
3. In the *Environment Variables* dialog box, place the mouse cursor on the **User variables for ...** list (Figure 80, right), where the current user name is specified in place of the ellipsis.
4. Find the variable named **Temp** in the **Variable** field. The **Value** field of the variable displays the path to the *Windows* temporary files directory, where the log file for the debug messages will be located.

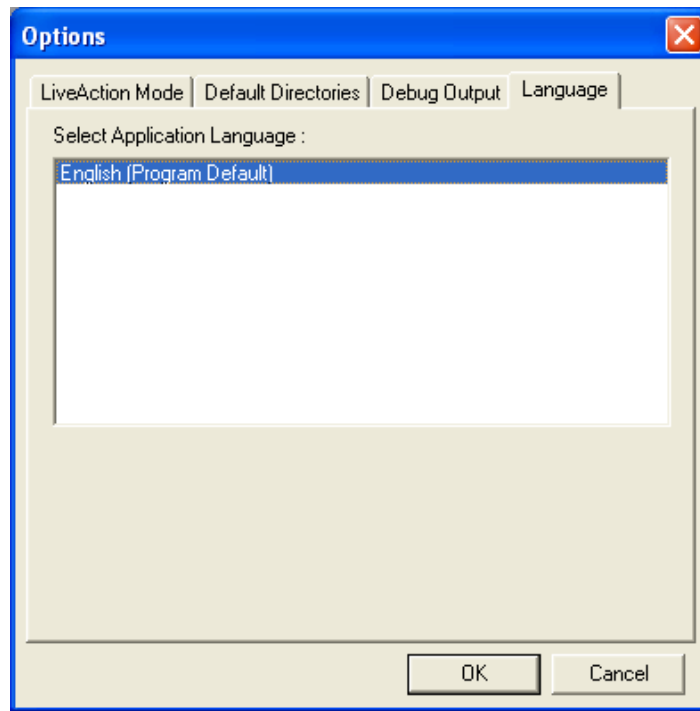


**Figure 80. The *Advanced* tab with the *Environment Variables* button (left) and the dialog box of the same name (right)**

A log file with the list of required debug messages can be sent to the developers in case of problems in order to perform fault tracing and debugging. Possible problems and solutions are described in the following Section.

## 9.4 The *Language* Tab

The *Language* tab of the *Options* dialog box (Figure 81) is designed to select a language for working in the application. The text of the information messages, menu items, prompts, etc. will be displayed in the selected language.



**Figure 81. The *Language* tab of the *Options* dialog box**

The languages list depends on the components included in the application installer designed to be used in a specific country.



*The names of all the languages are always displayed in English.*

After you change the language and click **OK** to exit the *Options* dialog box, the application must be restarted.

## 10 Troubleshooting

In case of a problem not discussed in this Section, contact Support (Section 10.4). To establish the cause of a problem you may need the log file with the debug messages. The file is named **HotActions.log** by default and is located in the folder for temporary files of the current system user (about using debug messages, see Section 6 and 9.3).

### 10.1 Problems with Default Directories

When launching the application, a dialog box for selecting a working folder, for example, for the project files, icons (Figure 82), scenes, libraries, *Hotsets*, graphics, audio and video files is displayed.

The dialog box is displayed if the **Common** folder with all the standard working folders was not created for some reason when setting up the application (or reinstalling updates), or if the folder specified in the **Default Directories** settings of the application (see Section 9.2) is absent, for example, if it was deleted.

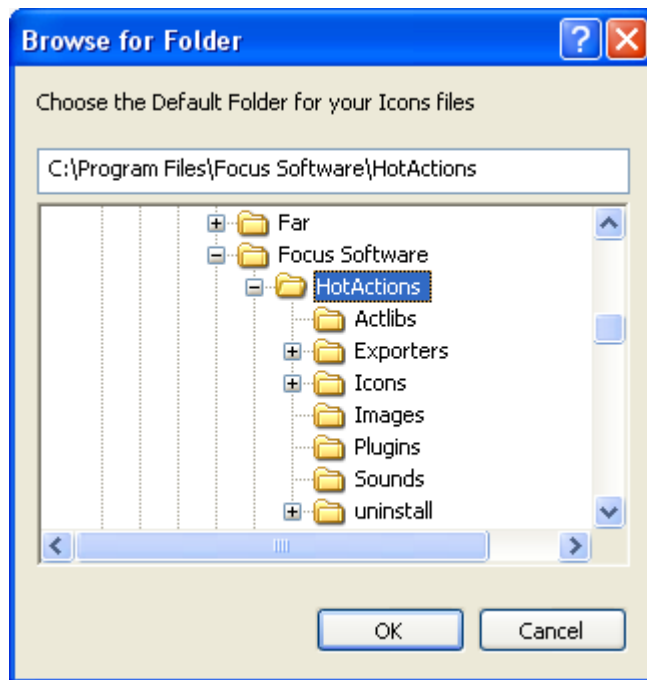


Figure 82. The Browse for Folder dialog box

To create the **Common** folder with standard working folders of the application, you can reinstall *HotActions*.

If the reinstallation has no effect, or other non-standard working folders are used, the application being could not restore the paths to them from the system register, it is possible to specify the path to the required folder in the dialog box displayed (Figure 82) or change the path in the **Default Directories** tab of the *Options* dialog box (see Section 9.2).

### 10.2 Problems with the *FD300* Board(s)

#### 10.2.1 One or Several Boards are not Found in the System

If the number of the list entries about *FD300* boards under **Board's Indexes** of the *FDConfiguration* application (Figure 2) does not match the number of boards installed in the system, it is necessary to make sure whether the boards are correctly presented in the *Device Manager* dialog box of the operating system. The dialog box is opened as follows: **Start > Settings > Control Panel > System > Hardware > Device manager**. If the boards are not found in the **Sound, video and game controllers** area of the dialog box (the corresponding entry **SoftLab-NSK FD300C Card** is not displayed), the board, most likely, does not physically contact with the slot. In this case it is required to switch off the computer power supply and check whether the boards are

properly installed in the slots. It is also possible that one of the boards is invalid, in this case contact Support (Section 10.4).

If the board is found in the system but not identified as **SoftLab-NSK FD300C Card**, perform the following steps:

1. Delete all the incorrectly installed boards from the system devices list. To do so, right-click the icon of each board and select the **Delete...** command in the context menu.
2. Right-click on the Section title and select **Update driver** in the context menu.
3. The system will find the new devices and install the corresponding drivers if the software was installed correctly.

## 10.3 Problems with Output Video

### 10.3.1 Absent *Render Output Window*

Make sure the **Render Window** command of the **View** menu is enabled (Figure 37) and execute the **Center Render Window** command of the **Window** menu (Section 7.1).

## 10.4 Technical Support Service

For complete consultation on all the questions concerning the Virtual Studio operation, please contact Support at: [vrset@sl.iae.nsk.su](mailto:vrset@sl.iae.nsk.su)

SoftLab-Nsk Co Ltd., Novosibirsk, [www.softlab-nsk.com](http://www.softlab-nsk.com)



## 11 Connecting Video Signals and Configuring HD Virtual Studio

### 11.1 The Studio Components Connection Scheme

To work with HD-SDI video signals, the Virtual Studio uses input boards *Aja Xena 2Ke*. The video signal sources are connected to the board connectors. The operating system must have drivers for board version 7.1.0.10 (installation file *AJA\_Windows\_Software-4.2.0\_x86.msi*).

Connect a switching device to the *FD300* board (the cable set and *Breakout Box* are described in Section 12, Appendix) to have the audio signal sources and receivers connected.

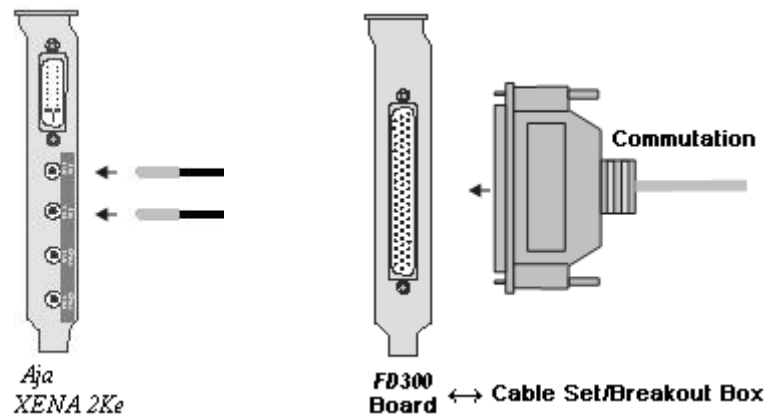


Figure 83. Connecting video signal sources to the *Aja XENA 2Ke* board and a switching device to the *FD300* board.

The HD Studio resulting image is output through the *DVM62* converter connected to the second connector of the video card. The first connector is used to connect the VGA monitor. If it is analog, the connection is performed via a DVI - D-SUB adapter.

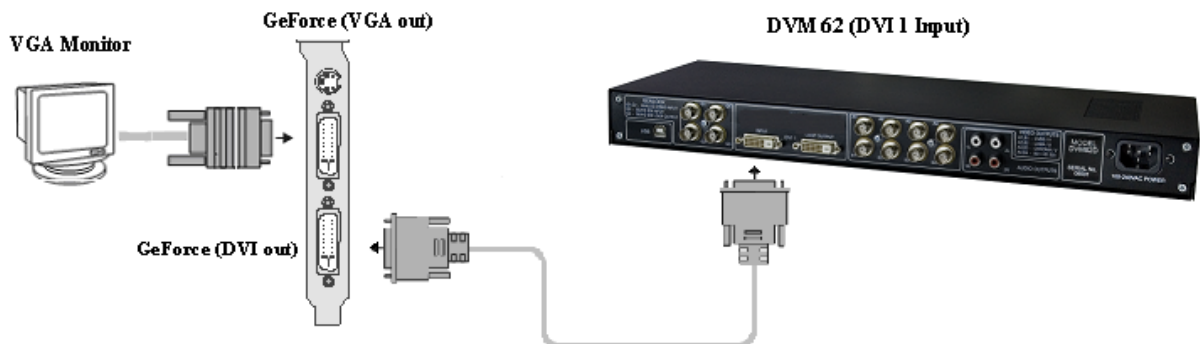


Figure 84. Connecting the Studio output video signal converter to one of the video card connectors.

When the connections are complete, connect *DVM 62* to power supply and flick the **Power** switch (on the front panel) to **On**.

The video signal receivers are connected to the back panel connectors. It may be the upper group (A1, A2, A3, A4) or the lower group (B1, B2, B3, B4). If required that the Studio video signal frequency be synchronized with an external source, connect the video source to connector G3 of the Genlock group. The front panel contains buttons that determine the output video signal format.

### 11.2 Setting the Video Card to Work in a Configuration with Two Monitors

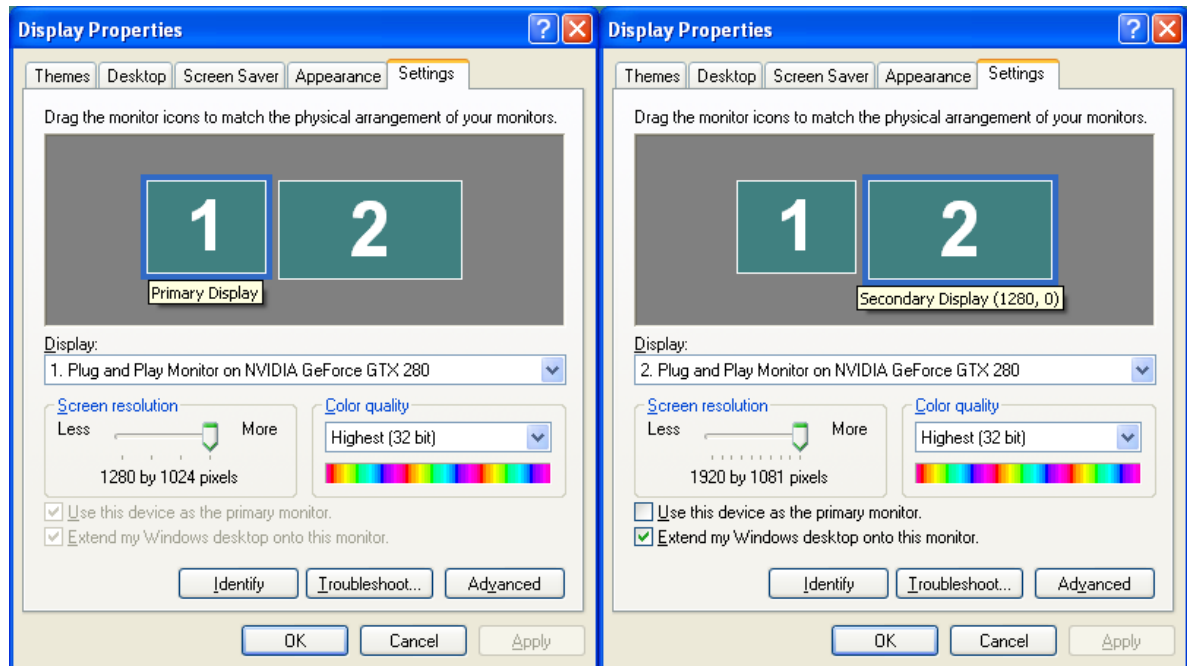
When the connections are properly done, and the computer is started, check whether the *NVIDIA GeForce* video adapter is properly set.

To do so, open the **Display Properties** dialog box (Figure 85) using the **Start** menu: **Start** → **Settings** → **Control Panel** → **Display** or right-click on the desktop and select **Properties** in the local menu.

In the **Settings** tab of the dialog box, select **1. Plug and Play Monitor on NVIDIA GeForce 8800 GTX** for the monitor icon **1** in the **Display** drop-down list.

Select **Highest (32 bit)** from the **Color Quality** drop-down list,.

For the monitor icon **2**, select **2. Plug and Play Monitor on NVIDIA GeForce 8800 GTX** from the **Display** drop-down list and enable the **"Extend my Windows desktop onto this monitor"** option located in the lower part of the dialog box



**Figure 85. Setting the NVIDIA GeForce video card to work in a configuration with two monitors.**

It is also necessary that the required standard be selected in the video adapter settings. To do so, perform as follows:

1. Press the **Advanced** button in the lower right part of the **Display Properties** dialog box. The **Plug and Play Monitor and NVIDIA GeForce GTX 280...** dialog box opens (Figure 86).
2. Select the **Monitor** tab and disable the option **Hide modes that this monitor cannot display** under **Monitor settings**.

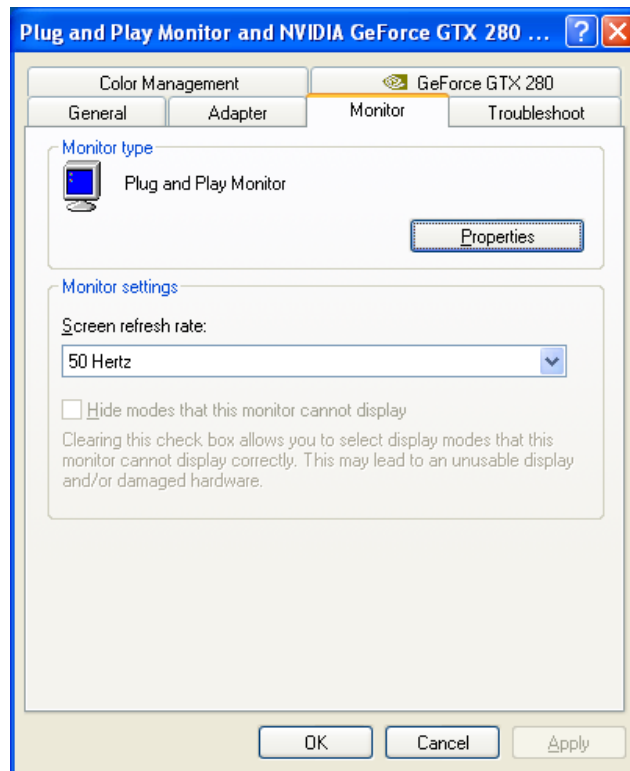


Figure 86. Dialog box for setting the monitor properties.

3. Select the **Adapter** tab (Figure 87). Click the **List All Modes** button. The **List All Modes** dialog box opens. Select the required mode.

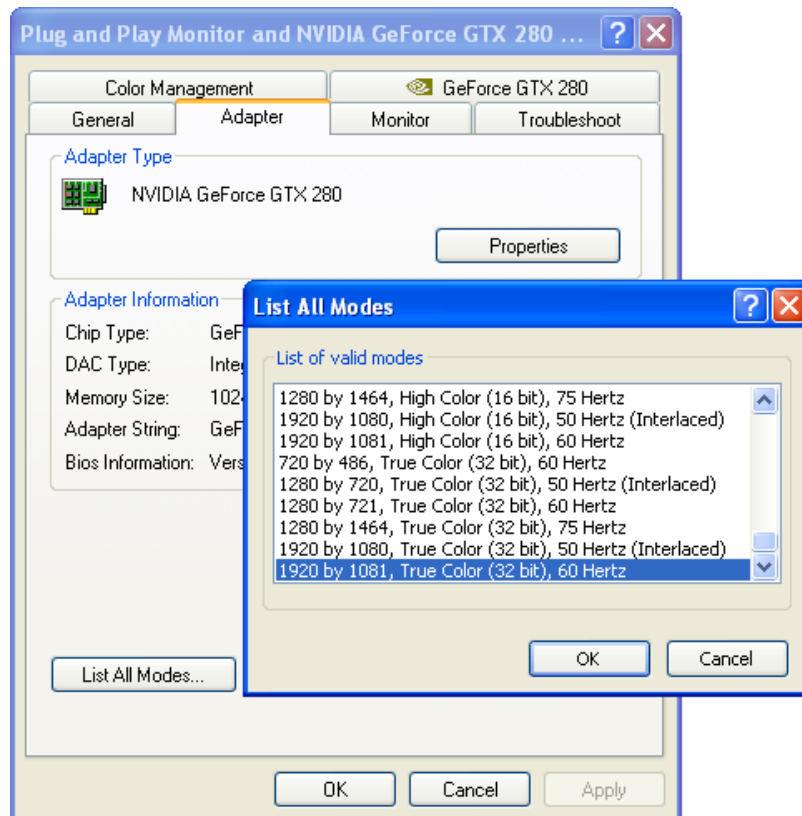
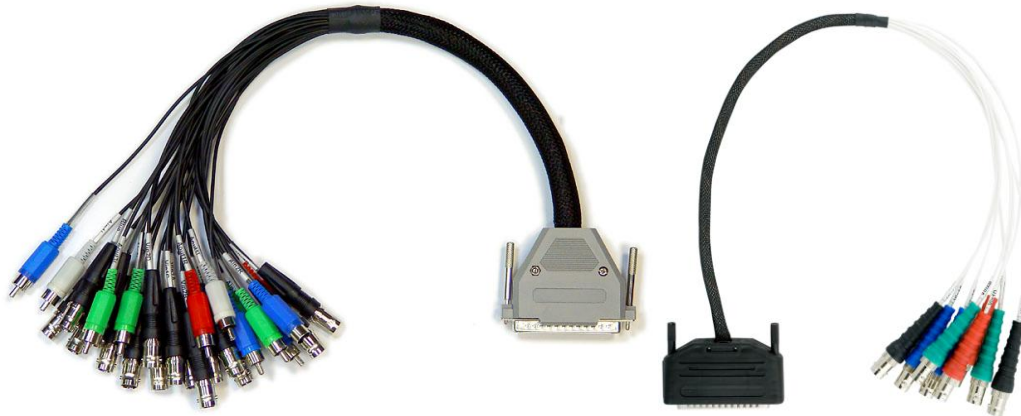


Figure 87. Dialog box for setting the second video adapter of the video card.

To further set the Virtual Studio for operation, refer to Section 2.2 ff. that covers the initial setup stages.

## 12 Appendix

### 12.1 Cable Sets for the *FD300* Boards



**Figure 88.** Cables for connecting the input and output audio and video signals to the *FD300* board

A complete cable set has 28 connectors corresponding to the connectors on the passive breakout box (except for SPDIF). All the connectors are marked according to the following rules:

All the video inputs are divided into three identical groups for handiness, each of them having the inscription **Vin** and a number of the input group. Each group contains 4 coaxial connectors (**A**, **B**, **C**, **D**). Depending on the device operating mode the following signals can be received on the video inputs:

- on input **A (Vin1A, Vin2A, Vin3A)** – CVBS-A, LUMA-AB, CSYNC;
- on input **B (Vin1B, Vin2B, Vin3B)**– CVBS-B, CHROMA-AB, Y, YS, G, GS;
- on input **C (Vin1C, Vin2C, Vin3C)** – CVBS-C, LUMA-CD, U,B;
- on input **D (Vin1D, Vin2D, Vin3D)** – CVBS-D, CHROMA-CD, V, R,

where:

- CVBS – composite video signal;
- LUMA, CHROMA – the S-video signal constituents;
- Y, U, V – the YUV component signal constituents;
- R, G, B – the RGB component signal constituents;
- CSYNC – a single signal of the component input synchronix;
- YS, GS – a component input synchronix is contained in the Y or G constituent.

The video output connectors **Red, Green, Blue, CVBS**, correspond to the RED/V/C, GREEN/Y, BLUE/U/CVBS and CVBS/CSYNC video outputs. The possible combinations are presented in Table 1. For more detailed information about switching video inputs, see Section 2 of *FD300 Board Hardware Setup*.

The audio inputs are marked: **Ain1L, Ain1R; Ain2L, Ain2R; Ain3L, Ain3R**.

The audio outputs: **Aout1L, Aout1R; Aout2L, Aout2R; Aout3L, Aout3R**, where **L** is the stereo signal left channel, **R** – right channel.

The complete set of cables is connected, as a rule to the *FD300* boards for the image output.

A set of cables connected to the board for inputting video signals to the studio has only 8 connectors, as there are no commutation for working with audio (audio connectors), connectors for the video output, there being only the video input connectors of the two groups: **Vin1 (A, B, C, D)** and **Vin2 (A, B, C, D)**.

## 12.2 The Passive Breakout Box (TF782)

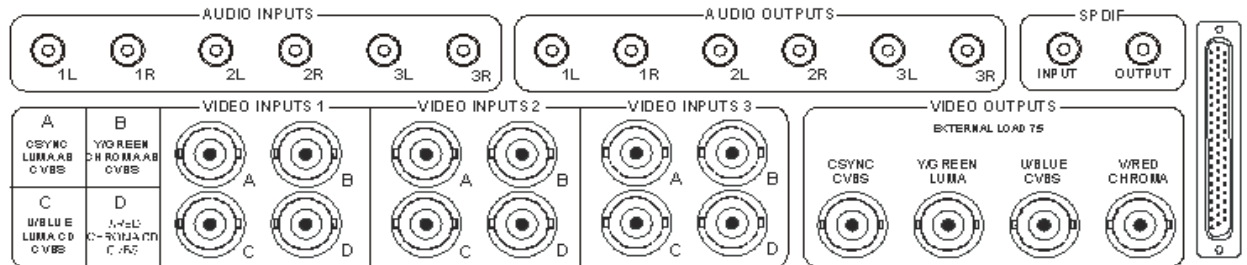
The passive breakout box (Figure 89, left) is designed to connect analog video and audio sources and signal receivers. It is connected to the *FD300* board connector with a special cable that comes with the box.



**Figure 89.** The appearance of the passive breakout box with the connection cable

Schematically the front view of the passive breakout box is shown in Figure 90. The passive breakout box scheme All the video inputs are divided into three identical groups, each of them having a frame with the inscription **VIDEO INPUT** and the input group number. Each group contains 4 coaxial connectors (A, B, C, D). Depending on the device operating mode the following signals can be received on the video inputs:

- on input A – CVBS-A, LUMA-AB, CSYNC;
- on input B – CVBS-B, CHROMA-AB, Y, YS, G, GS;
- on input C – CVBS-C, LUMA-CD, U,B;
- on input D – CVBS-D, CHROMA-CD, V, R.





**Figure 90.** The passive breakout box scheme

The video outputs are grouped in **VIDEO OUTPUTS**, where are the 4 outputs (RED/V/C, GREEN/Y, BLUE/U/CVBS, and CVBS/CSYNC). The possible combinations are given in the table below.

**Table 1.** Combinations of the video outputs

Possible combinations	VIDEO OUTPUTS group connectors			
	RED	BLUE	GREEN	CVBS
<b>Y/C+CVBS</b>	C(Chroma)	CVBS	Y(Luma)	CVBS
<b>YUV+CVBS</b>	V	U	Y	CVBS
<b>YUVS</b>	V	U	Y	Sync
<b>RGBS</b>	R	G	B	Sync
<b>YUV+Key/α</b>	V	U	Y	Key/α

 You may **NOT** receive the Y/C (S-video) signal simultaneously on the miniDIN-4 **S-VIDEO** connector and **GREEN** (Y – brightness signal) and **RED** (C – color information) connectors.

 The chroma signal (C) must be loaded at 75 ohm. Some SONY equipment (monitors, videorecorders) do **NOT** have such load and it must be provided for.

The audio signals are divided into 2 groups: the output signals – **AUDIO OUTPUTS**, input signals **AUDIO INPUTS**. Each group has six patchholes as three L-R stereo pairs.

### 12.3 The Active Breakout Box (TF785 M)



Figure 91. The front view of the active breakout box

There are 3 identical channels of **VIDEO INPUTS** (Figure 92) on the active breakout box (Figure 91). Each channel contains four pairs of parallel-connected through-pass coaxial connectors BNC (A, B, C, D), and also 2 MiniDIN4 connectors for connecting with the S-video sources (inputs AB and CD). The connectors used are selected with the **COAXIAL/S-VIDEO** switches. The **OFF/ON** switches are used to connect/disconnect the 75 ohm terminal resistors of the corresponding coaxial video input.

Depending on the device operating mode the following signals can be received on the video inputs:

- on input A – CVBS-A, LUMA-AB, CSYNC;
- on input B – CVBS-B, CHROMA-AB, Y, YS, G, GS;
- on input C – CVBS-C, LUMA-CD, U,B;
- on input D – CVBS-D, CHROMA-CD, V, R;

where:

- CVBS – composite video signal;
- LUMA, CHROMA – the S-video signal constituents;
- Y, U, V – the YUV component signal constituents;
- R, G, B – the RGB component signal constituents;
- CSYNC – a single signal of the component input synchronix;
- YS, GS – a component input synchronix contained in the Y or G constituent.

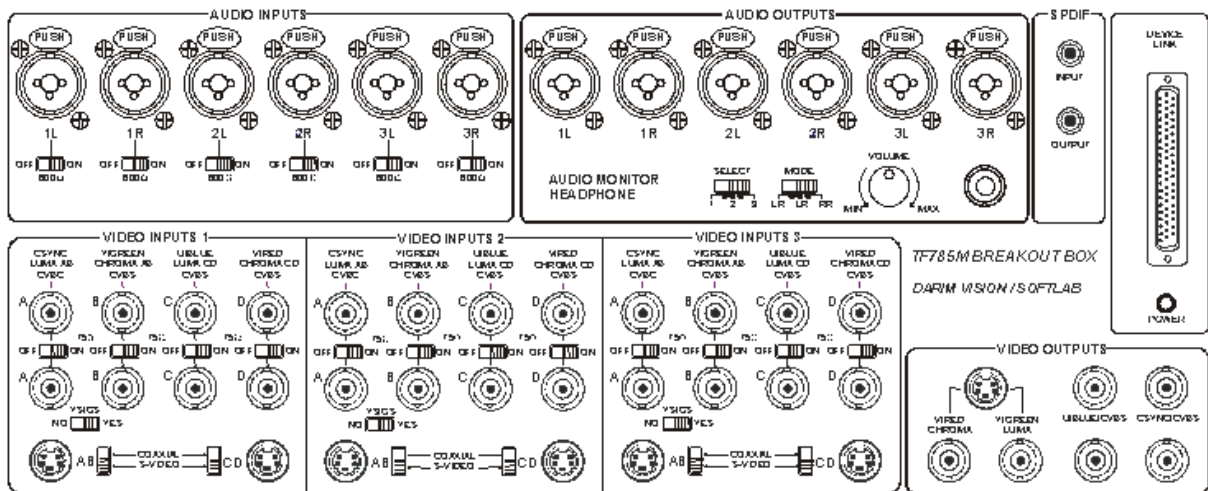


Figure 92. The active breakout box scheme

When a component signal with YS or GS is sent to the channel video input, the **YES/NO** switch must be in the **YES** position. In all other cases it must be in the **NO** position (normal position).


There are also four video outputs (RED/V/C, GREEN/Y, BLUE/U/CVBS, CVBS/CSYNC) on the box combined into the **VIDEO OUTPUTS** functional group. Each video output is duplicated, which allows connecting the preview monitor simultaneously with the working signal.

Six patchholes grouped in three L-R stereo pairs form the **AUDIO INPUTS** functional group. All the audio inputs are balanced, for which three-pole stereo plugs are used. The **OFF/ON** switches are used to connect/disconnect the 600 ohm terminal resistors of the corresponding audio input.

The **AUDIO OUTPUTS** functional group is formed by six balanced audio outputs. They can be used as unbalanced ones with the earth connection of one balance line, the same way as the audio inputs.

There is a patchhole for a stereo headphone connection in the **AUDIO MONITOR HEADPHONE** group. The output level can be set by the **VOLUME** controller. The number (1, 2, or 3) of an audio output for listening is selected by the **SELECT** switch. The **MODE** switch allows setting a necessary mode: LR – stereo, LL – left channel only, RR – right channel only.

The **POWER** light-emitting diode indicator shows the power supply availability.

 If the **POWER LED** indicator is not alight after the **FD300** board is connected to the active breakout box, contact Support (Section 10.4).

## 12.4 Breakout Boxes for a System with Several *FD300* Boards (TF783 A, TF783 B, TF783 C)

If several *FD300* boards are installed in the system, and it is desirable to perform all the necessary connections of audio and video sources and signal receivers on the same breakout box, the TF783 A, TF783 B, or TF783 C boxes can be used for this purpose (Figure 93).



**Figure 93.** The front view of the breakout boxes for connecting several *FD300* boards

As a rule, the *TF783 A Breakout box* is used in systems with two *FD300* boards for video data input and the option of synchronizing the output video signal to an external device. The **VIDEO INPUTS 1** (A, B, C, D) and **VIDEO INPUTS 2** (A, B, C, D) groups are designed to connect video sources to the first board decoders, the **VIDEO INPUTS 3** (A, B, C, D) and **VIDEO INPUTS 4** (A, B, C, D) groups – to connect to the second board, the **VIDEO INPUTS 5** group – to connect the external synchronization device. The video outputs are presented by the **VIDEO OUTPUTS** group: RED/V/C, GREEN/Y, BLUE/U/CVBS, and CVBS/CSYNC. The input **AUDIO INPUTS** and output **AUDIO OUTPUTS** audio signals are always connected to the *FD300* board that is for the signals output.

The *TF783 B Breakout box* and *TF783 C Breakout box* are usually used only to connect the input signals.

The cable used to connect a breakout box to the *FD300* boards also comes in the package with the breakout box (Figure 94).



**Figure 94. Cable for connecting *FD300* boards to the breakout box**

The cable branches are designed for the connection with the *FD300* boards. Usually there are color marks or numbers at the ends of the branches for determining the correspondence of connection to the boards. A connection type is usually determined by the current system configuration file with the \*.cfg extension located in the **C:\Program Files\ForwardT Software\FDConfig** folder next to the *FDConfig.exe* file. The file name is selected from the **Breakout Box Cable Connectors** drop-down list in the **FD300 Configuration** application. All the available configuration files are also usually located in the folder **C:\Program Files\ForwardT Software\FDConfig\CableInfo**.