Focus

Virtual Studio

Creating 3D Scenes

User's Guide

Revision as of November 25, 2009

VS 1.63

Table of Contents

1 Introduction

The current User’s Guide assumes that the reader is acquainted with the *3D Studio Max* application (versions 4.2 and subsequent, unless otherwise stated) and is skilled in creating objects for use in a 3D scene. Specifically the reader should have knowledge of virtual cameras and their animation, etc. When working in *3D Studio Max*, consider the recommendations and restrictions discussed in Section 2.

In order to be used in a *Focus* project, a scene created in *3D Studio Max* must be exported to a file with the `*.3d` extension. Using the *VS Exporter* device (see Section 11) allows you to preview a scene in the 3D format directly from *3D Studio MAX*. It is strongly recommended to use the ability since the results of rendering in *Focus* can differ from the results of rendering in *3D Studio MAX*. The difference can increase appreciably if the restrictions on the export of scenes listed in Sections 2.1.4, 2.1.8 are not observed.

2 Creating 3D-Scenes for *Focus* in *3D Studio MAX*

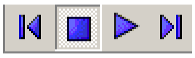
2.1 Restrictions on Exported Parameters

Export of the scene files created in *3D Studio Max* and their subsequent use in the virtual studio *Focus* assumes adherence to the minimum requirements for a 3D scene. The requirements are similar to those that face the developers of computer games. *Focus* takes full advantage of the hardware 3D graphics accelerator features. Besides, it is necessary to provide successful functioning in real-time mode, when all the required operations have to be performed in an interval shorter than a video half-frame duration time.

The requirements given below are quite strict. But following the recommendations leads to impressive results.

2.1.1 General Scene Track Export

The objects of a scene can be animated by using tracks (also, see Section 2.1.12). It is necessary to name tracks to operate them in *Focus*, since in the *3D Studio Max* application, the names of tracks are not specified.

The buttons  on the scene window toolbar (Figure 18) are used to play and stop the general (with a specified name) track of a scene. The buttons allow you to preview the general track of a scene before the scene is initialized in a project. Naming tracks of certain objects during the initialization makes it impossible to operate them with the buttons on that toolbar. The further control is performed with commands. The commands controlling tracks are described in detail in Section 4 of *VS200 Virtual Studio. Using the Script Commands User's Guide*.

To use a scene in *Focus*, add it to an existing project in the *HotActions* application or create a new project (see Section 3 of *Focus Virtual Studio. HotActions* User's Guide).

2.1.2 Geometry

The exported parameters are:

- **Standard primitives**
- **Extended primitives**
- **Compound objects.**

2.1.3 Cameras

The only cameras exported are:

- **Target camera**. Virtual cameras of this type in most cases are controlled in a scene with the help of ***Dummy*** objects (see Section 9.1). A **Target camera** can be attached to any animated object in a scene. Tracks are created just for the auxiliary objects of the kind. That is necessary to correctly rotate the camera.
- **Free camera** (not having an aiming point).

2.1.4 Light Sources

It is possible to use up to 8 light sources. Still, it is advisable to minimize the number as far as possible, since every added light source decreases the rendering speed in real-time mode. Use of the following types of light sources is available:

- **Omni** – omnidirectional point light source, located in a point; lights in all directions of the scene three-dimensional space.
- **Target Spot** – a directional point light source, consisting of the point light source itself and a Target setting the direction of the rays. The light rays conically. A surface illuminated by such a source must be tesselated, the tesselation degree is fitted experimentally.
- **Target Direct** – a directional light source, a surface giving a bundle of parallel rays.
- **Free Spot** – a point light source without a target object; the light rays conically. A direction of the light ray is specified by rotating the illuminator.

- **Free Direct** – a light source without a target object; the light source is a surface giving a bundle of parallel rays.

Only the following parameters of light sources are supported.

- **Multiplier** – the light intensity.
- **Attenuation** – is supported if the surface is tessellated.
- **Light Color**
- **Spotlight/Directional Parameters** – a range of angles controlling the parameters of a directional ray; supported only for directional light sources.

Inverse Square Decay and shadows are NOT supported.

2.1.5 Hierarchy

The hierarchies supported are:

- **Pivot**
- **Linking**
- **Unlink.**

2.1.6 Environment

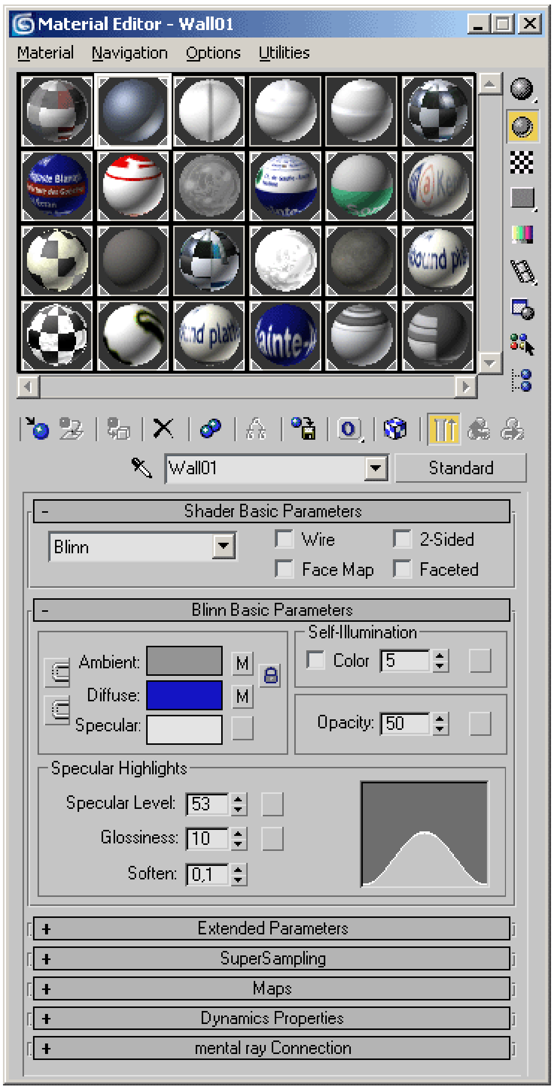
The environment parameters supported are:

- **Background Color**
- **Ambient Global Lighting**
- **Fog Atmospheric Effect.**

2.1.7 Materials

The types of materials supported are:

- **Standard**
- **Multi/sub-object**
- For the **Standard** materials, only certain **Blinn** parameters from **Shader Basic Parameters** are exported (Figure 1), the others being ignored. Of the **Blinn Basic Parameters**, the only parameters exported are: **Ambient**, **Diffuse**, **Self-illumination**, **Opacity**, **Specular**, **Specular Level**, **Glossiness**. Other shading parameters are ignored.



Wall01 Standard

- Shader Basic Parameters

Blinn □ Wire □ 2-Sided □ Face Map □ Faceted

- Blinn Basic Parameters

Ambient: M

Diffuse: M

Specular:

Self-Illumination □ Color 5

Opacity: 50

Specular Highlights

Specular Level: 53

Glossiness: 10

Soften: 0,1

+ Extended Parameters

+ SuperSampling

+ Maps

+ Dynamics Properties

+ mental ray Connection

Figure 1. Material selection dialog.

2.1.8 Textures

As texture maps, only **Bitmap** are used, both the static images and video files:

- **Diffuse color**
- **Ambient color**

> *Note: The latter is used not as* ***Ambient color*** *proper.* ***Ambient color map*** *is exported into Focus as an additional* ***Diffuse color map*** *or as* ***Lighting Map*** *if the* ***Alpha Source*** *value is set for it in* ***RGB Intensity****. The* ***Lighting Map*** *usage is convenient in that the texture has an independent set of coordinates. That means it is possible to change the main texture (****Diffuse Color****) preserving the illumination texture, shadows and highlights remaining unchanged.*

> *The allowable total size of the video files used for textures should be selected experimantally for every scene. The reason is that they are exported as a set of low-compressed bitmaps and thus might dramatically enlarge the size of the resulting exported 3D scene file. That, in turn, increases the loading time and may lead to performance problems.*

- **Opacity**
- **Reflection**

> *Note: Using the latter texture map requires acquaintance with the features described in Section 5.*

2.1.9 Texture Coordinates

The material parameters exported are:

- **UV offset**
- **UV tiling**
- **W Angles**
- from **Texture/environ** – only texture
- from **Mapping** – only **Explicit Map Channel**.

The maximum number of textures for a material is three. The following combinations are allowable:

- **Ambient color + Diffuse color + Opacity**:
    - if **Alpha source** in **Bitmap parameters** is marked as **No** (**Opaque**), the maps will be simply mixed in proportion to specified values (**Amount**). The sizes of textures in texels (pixels of texture) for **Diffuse color** and **Opacity** must be identical. Values **U** and **V** for **Diffuse color** + **Opacity** must be identical;
    - if **Alpha source** in **Bitmap parameters** is marked as **RGB Intensity**, the maps will be mixed in a special manner (**multiply**). In that case, the **Amount** values of texture parameters are ignored.
- **Diffuse color** + **reflection**:

  the same modes of mixing as in **Ambient color** + **Diffuse color**.
- **Diffuse color** + **Opacity**.

2.1.10 Sizes of Textures

The sizes of textures (in texels) for **Diffuse color** + **Opacity** should be identical. Values **U** and **V** for **Diffuse color + Opacity** should be identical.

> ☝ *Note: The map size should not be bigger than the videoadapter allows, and should not exceed 2048 texels for GeForce 3 or to 8192 for the later GPUs. It is desirable to control the sizes of textures sides. When exporting to a 3D file or loading a 3D scene file into HotActions (see Section 11.2), the sizes of textures are sometimes automatically rescaled so that they equal 2 texels in degree N, where N is an integer. Thus, in texels it is: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. Note that sizes are always rescaled to the upper closest value 2^N in order to keep the quality. For example, a texture of 257 by 513 texels in size is to be rescaled to 512 by 1024 (not 256x512), though it is very close to the lower values of 256 and 512. That way of scaling will lead to needless use of additional computer resources. Therefore, it is better to take control of the export process. At first, create a texture (for example, in Photoshop) in natural proportions and then scale it to the nearest allowable values manually, so that the quality of a picture is not reduced. The procedure may cause the image look flattened or stretched, but that's of no importance, since in the 3D scene the proportions depend exclusively on how the texture coordinates (**UVW Mapping**) are given. On the one hand, specifying sizes for textures is not a rigid requirement, on the other hand, it is extremely desirable, especially if there are a lot of different textures in a scene.*

2.1.11 Auxiliary Objects

The only auxiliary objects exported are **Dummy** objects.

2.1.12 Animation

Animation is only supported for nodes parameters: cameras, light sources and scene objects proper. **Bones** are not supported, as well as dynamic changes of a material such as changes in transparency. **Visibility Track** is not supported.

3 Morphing in Virtual Studio

The **Morphing** procedure in *Focus* is executed by the **MORPH** script command, which is a subcommand of the RENDER commands category (see Section 5 of *Focus Virtual Studuo. Using the Script Commands User's Guide*). To subject objects to morphing, perform the following steps in the initial scene of *3DS MAX*:

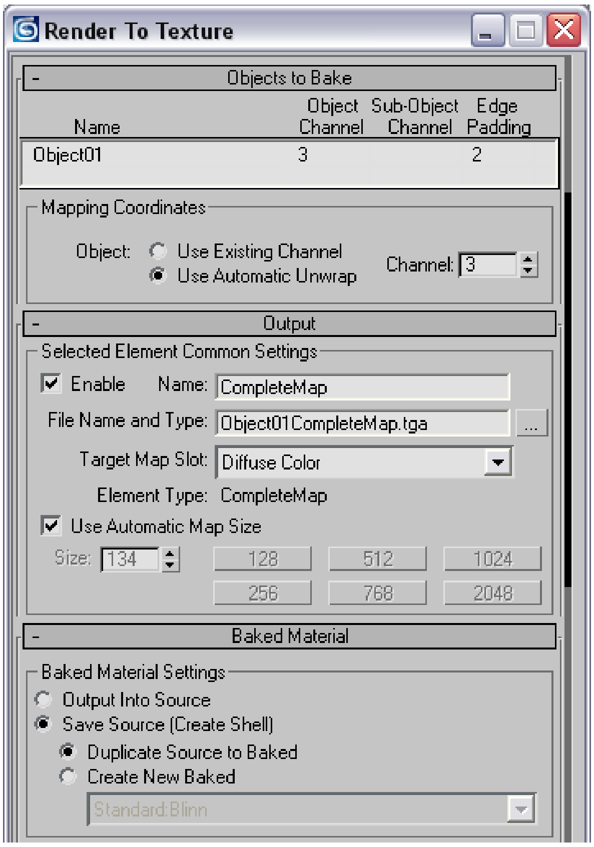
1. create a basic original object;
2. create several copies of the object to meet the supposed number of morphing stages, each stage requiring a copy;
3. edit the created object copies to achieve the stages of morphing.

> ☝ *Note: objects on all stages of the procedure must have the same number of vertices as the original object. To make the stages (not the original object) invisible, use the track editor.*

4 Generating Light Effects and Surface Roughness

Shadows and highlights are important elements for visual perception of a scene. Limited use of highlights can make a scene look plain and lifeless. Note that export of shadows caused by light sources is not supported. The solution to the problem is the imitation of shadows and highlights with the help of textures. That can be done in different ways. One of them is presented below.

- Create as many light sources of appropriate types as is necessary for an impressive illumination of the scene and creation of shadows. Ignore all the restrictions described above.
- Use the *3D Studio Max* tools to create (render) textures with “imprinted” shadows and highlights or light-and-shade textures (**Lighting Map**). Since version 5.0, 3D Studio Max has the ability of rendering to texture (the **Render To Texture** module). The recommended settings when using the module are presented in Figure 2.



Render To Texture

- Objects to Bake

Name	Object Channel	Sub-Object Channel	Edge Padding
Object01	3		2

Mapping Coordinates

Object: ○ Use Existing Channel
● Use Automatic Unwrap
Channel: 3

- Output

Selected Element Common Settings

☑ Enable Name: CompleteMap

File Name and Type: Object01CompleteMap.tga

Target Map Slot: Diffuse Color

Element Type: CompleteMap

☑ Use Automatic Map Size

Size: 134

- Baked Material

Baked Material Settings

○ Output Into Source
● Save Source (Create Shell)
● Duplicate Source to Baked
○ Create New Baked

Standard:Blinn

Figure 2. Recommended settings of the *Render To Texture* module when pre-rendering textures.

When using automatic setting texture sizes (**Use Automatic Map Size**), note that determination of sizes is performed incorrectly in some versions of *3DS MAX*. The result of setting should be verified by examining the texture sizes in the *HotActions* application.

- Remove all the auxiliary light sources, leaving only those required and no more than 8 sources (see Section 2.1.4).

Using the **Render To Texture** module enhances the visual characteristics of a scene in the *HotActions* application to a large degree. Note that the video memory size of the graphics accelerator is limited, and overrunning it can result in dropping frames when displaying. Therefore, it’s recommended to perform pre-rendering only for the most effective textures, taking note to optimize their sizes (see Section 2.1.10).

5 Using Materials with Reflective Properties

Due to hardware features of representing materials with reflective properties, there may occur breaks of image on the joints of large faces. The distortions are caused by the lack of perspective transformation correction.

If distortions are too distracting, the possible solutions to the problem are as follows:

- splitting of large faces (tesselation);
- concealing the joints of faces, i.e. arranging objects in a scene so that the edges of their large faces are not reflected;
- in case of large angles between faces, especially between large faces, it is expedient to smooth the angles by adding small superfluous faces.

6 Reduction of the Number of Polygons in a 3D Scene

> ☝ *Note: a 3D scene should not contain an excessive number of objects with many faces and should not be overloaded with a great many different textures.*

The objects of a scene are to be optimized. That can be done in various ways, below are given the fundamentals:

- It is desirable to know the possible positions of a camera in a scene. The further an object is from the camera, the less detailed it can be.
- Remove or simplify all the objects that are not important for the perception of a scene and the script.
- Remove all the objects that can be modeled by a texture, for example, a sheet of a paper on the table, etc.
- Optimize flat and poorly bent surfaces of objects, editing the parameters of the objects and/or using the **Optimize** modifier.
- Remove all the surfaces of objects that are never observed by a camera.

7 Live Video

A material used for a video image should contain a **Diffuse map**. After initialization of a video material in a scene, the map will be replaced with the video image. The details of the process are described in Section 7 of the *Focus Virtual Studio. Using the Script Commands User's Guide*. During the process, all the other maps and parameters of the material are ignored.

The material name used for the video is not important. In the example projects, which are usually located in the **VS_Samples** directory of disc **D:**, the materials are most often referred to as **VIDEO1** and **VIDEO2** if two streams of the input video are simultaneously used. If only one FD300 board is installed in the system, or only one video stream is used in the scene, the material is frequently named **VIDEO** or **VIDEO_m**. The latter might be more convenient to avoid confusion when the word "VIDEO" is used as a component of commands. For example, **VIDEO** in RENDER.VIDEO.LIVE_1.CREATE = 1 is a part of a command. And **VIDEO_m** in RENDER.MATERIAL.VIDEO_m.SOURCE = LIVE_1 is a material name.

7.1 Creating a Texture for a Video Material

An image of texture used for a video material does not matter much. It will be replaced with live video after initialization of the scene in *Focus*. Still, for the designer to better realize the general composition of a scene, it is recommended to use a static image of an actor as the video texture. At that, the actor's image should be foreshortened the same way as it will presumably look in the viewfinder (half-length, full-length, etc.).

For that purpose, use the two types of textures – **Diffuse** and **Opacity**.

For **Diffuse Map**, use a photo (picture) of the actor, observing the general recommendations for textures (see Section 2.1.10).

For **Opacity Map**, use a mask. It is created in *Adobe Photoshop* on the basis of the same photo or picture. Fill the actor's silhouette with white filling. That will be the opaque part of a material. Fill the rest field, which will be the transparent part of the mask, with black.

All the operations for **UVW Mapping** will be saved when replacing with the live video. We recommend applying video material to a rectangle with the aspect ratio of 4:3 both for PAL and NTSC (16:9 for HDTV).

> *The 4:3 aspect ratio of an object suggested for the imitation of the TV screen has nothing to do with the TV resolution (720x576 for PAL or 720x480 for NTSC). The pixel aspect ratio may vary for different TV standards, but neither for PAL nor for NTSC, it is 1:1.*

The texture is to be inscribed in the rectangle. The vertical and horizontal sizes of the texture do not matter since all the parameters are imported from the video after the material is replaced with video. After the scene export, it is possible to change the aspect ratio of the rectangle using the script commands **DATA.NODE.«Object».SCL = fa, fb, fc** in the initializing ***Action***. After the aspect ratio is changed, you may need to shift the object, which is possible with the appropriate commands (see *Focus Virtual Studio. Using the Script Commands User's Guide*). Nevertheless, it is allowable to use any material of the scene containing the **Diffuse map** as a base for video. Thus, the video image can be displayed on any object of the scene: a wall, a screen and even a teapot.

> *Note: It is recommended to prefix the name of an object with the symbol "**!**" so as to avoid an angle between the actor plane and virtual camera, which makes the image look like a flat face. In that case, when moving the virtual camera, the object also moves so that the image plane always remains perpendicular to the line of sight. The object behaves similarly to a weather vane, turning round its vertical axis (Z) to face the camera (see Section 8). Strongly recommended to apply the Reset XForm modifier to such objects. The reason is that in 3D Studio Max, the directions of the created object coordinate axes do not necessarily coincide with the axes of the world coordinate system, i.e. 'own' axis Z of an object may be directed not upwards.*

8 Naming Objects in 3D Scenes and Reserved Prefixes for Names

If an object name consists of more than 40 symbols, it is reduced and assigned a number when exporting a scene to *HotActions*, so that there are no more than 40 symbols.

> ☝ *Note: It is desirable to give materials and objects exact names appropriate to their purposes, thus simplifying work with them in HotActions.*

Special names are used for the objects enabled by implied call, for example, those manipulated by the mouse or joystick straight after the scene has been opened:

- The object with the name **World** is set up for manipulation by a mouse or joystick by default when a scene has just been opened.
- It is prohibited to give an object in a scene the name <**Current**>. This special name is used in script commands as the designation for the currently chosen object. As the script changes the selection of an object for manipulation, the new object is automatically temporarily assigned the name <**Current**>.

The following symbols at the beginning of an object name have a special significance:

Symbols influencing the position of an object relative to the camera	
#	An object always faces the line of sight. All the angles of the object rotation are equal to zero in the observer's system of coordinates.
!	The same as **#** with one difference: the object behaves similarly to a weather vane, turning to the camera round its vertical axis (Z).
:	The same as **#** with one difference: usually the object begins to look smaller as the camera dollies out; here the process of demagnification progresses slower than usual. It is used for light sources, which from a distance tend to seem larger than their actual size.
Symbols varying rendering priority. The system uses the algorithm of the Z-buffer, which sorts surfaces for rendering in a priority and removes surfaces hidden to the spectator. There can be cases when is necessary to change the rendering priority for some objects, for example, in a scene with translucent objects.	
< **>**	Changes the priority of an object relative to other objects with the same ancestor on the scene tree. The names of objects that are closer to the spectator must be prefixed with '<', those that are further – with '>'.
<< **>>**	The same as '<' and '>', but these symbols are used to change the priority relative to all the objects of a scene.

9 Exporting and Using an Elementary Scene in *Focus*

To use a scene as virtual scenery, the minimum required contents of the scene are as follows:

- an object with any textural material (for live video)
- a camera
- a light source.

In the next Sections, we shall discuss an elementary scene created with the help of the *3D Studio MAX* application. That is an example of a minimal scene representing a "virtual studio".

The eventual result of a project creation described below is included in the examples supplied with the application: the project **simple.vsp** is located in the **Simple** folder, the **VS_Samples\VS_Basic** directory. The **readme.txt** file of the project contains a concise description of particular ***Actions***, their commands also have detailed comments.

9.1 Creating a Minimal Scene for Virtual Studio

Launch *3D Studio MAX*. Open the **Simple.max** file with the **Open** command of the **File** menu. The scene presents a completely ready scenery example with the elements most frequently used to create projects in *Focus*.

The required objects are as follows:

- a camera (of the **Target** type)
- a light source (of the **Omni** type)
- a flat rectangle with an actor image. The *HotActions* application will project the assigned video stream from a camera onto the material of the object.

Some optional objects, yet, difficult to manage without are as follows:

- the floor and green or dark blue background for the actor (in our example, those are the **back_blue** and **back_green** objects, from which only one is visible, and the other is hidden);
- a table or any other object hiding a part of the actor's body if the actor is not supposed to appear full-length in the camera. For example, when shooting the actor half-length, with him sitting at the table, a part of the body being hidden behind the table;
- a "monitor" for displaying video clips, slides, etc. The texture used for the material of the "monitor" screen is replaced (usually afterwards, in the *HotActions* project) with video stream from a DV file or with input video.

Besides, there is a second camera named **Camera02** in the scene controlled with a ***Dummy*** object **CAMERA2**. It is intended to demonstrate the ability to instantly switch from one camera to another in the *HotActions* application. The first camera named **Camera01** controlled with a ***Dummy*** object **CAMERA1** also demonstrates the animated movement of the camera in the sample virtual studio.

Note some important features of creating a scene:

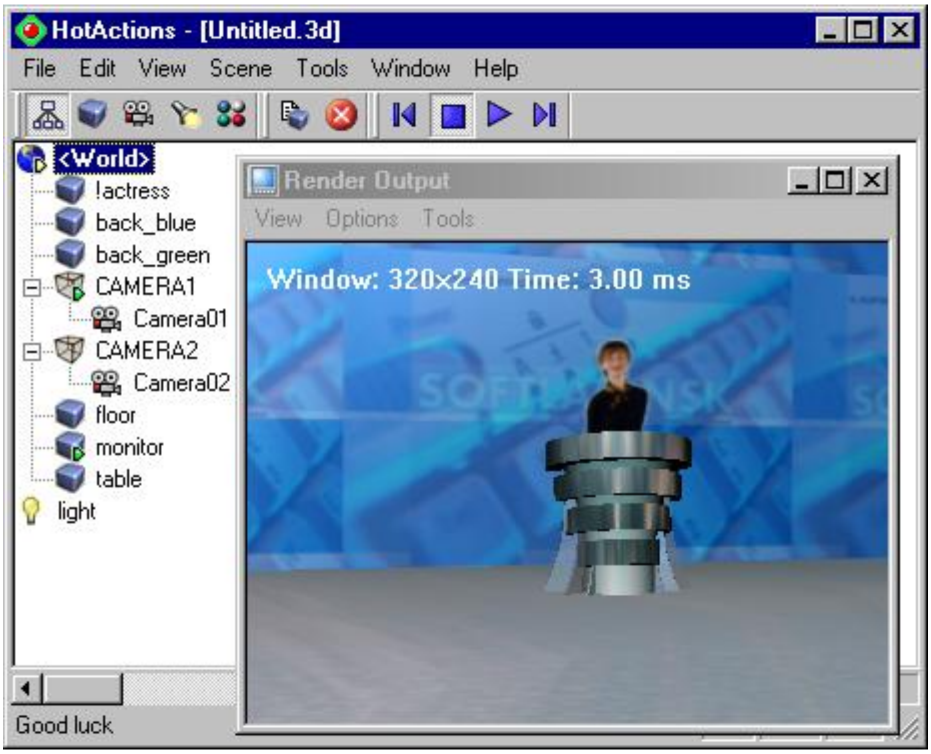
- To display the "actor" in the *Focus* project, a flat rectangle with the 4:3 (16;9) aspect ratio is created beforehand in the scene, according to the recommendations in Section 7.1. The photo of the actor is assigned to the **Diffuse** map to create a material to be replaced by live video. A black-and-white mask of the photo is used in the **Opacity** map. To avoid an angle between the actor plane and camera, prefix the name of the actor plane with the symbol "**!**". That will make the object behave similarly to a weather vane turning round its vertical axis (Z) to face the camera. An object exactly like that is present in the scene (named **!actress**).

  > ☝ *When an object behaves like that, use it's local coordinates, which do not necessarily coinside with the world coordinates. Therefore, in order to avoid confusion, it is recommended (within* 3D Studio MAX*) to apply «****Reset Xform****» operation to objects with the symbol «****!****» in their names. The operation brings the local coordinates of an object into accord with the world coordinates.*

- It is possible to use various sceneries as a background, using the animation created in the *3D Studio MAX* application. In the example, the background is an elementary flat object. As the actor is shot against a dark blue or green background, it is desirable to create a virtual background close to green or dark blue (respectively) for the actor to fit into the virtual scenery in the most natural way. It is possible, as in the given example, to create two backgrounds in parallel and hide one or the other with the help of the script commands in *HotActions* when necessary.

- The table for the actor is the closest object to the camera, therefore it requires the greatest attention to the details. It is expedient to use a reflections map on the surface of such an object. In the given example, when the camera is approaching to the actor, one can see quite well the highlights simulated with the help of the **Reflection** map.
- In the given example, "monitor" is a flat object without depth with a material having one map of the **Diffuse** type (similar to the actor plane). The example below shows the way a picture from an AVI file can be displayed on the "monitor" screen, or a slide can be shown directly in *HotActions*.

It is possible to preview a scene (Figure 3.) in the *VS Exporter* application (see Section 11.2).



Window: 320x240 Time: 3.00 ms

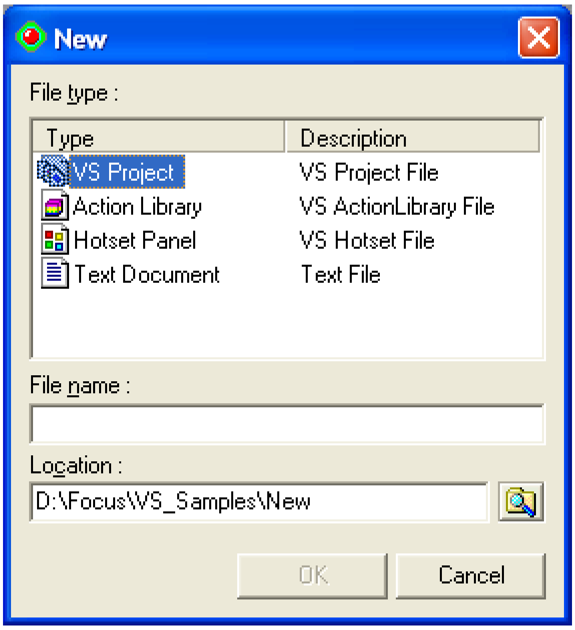
Figure 3. An example of a scene opened in *VS Exporter.*

9.2 Importing a Scene and Creating a New Project in *Focus*

To import a scene, it's sufficient to save it with the **Save As...** command in the **File** menu of the *HotActions* main window. Specify a name, e.g., "Simple" and folder in the opened dialog. The procedure is described in detail in Section 11.3.

As a result of the scene import, we have a scene file of the *.3d format. Thereby, there are two sample files included in the *Focus* application – the first one is a scene file directly created in 3DStudio, that is **Simple.max**. The second one is a version of the same file after the import (in the same directory with the *.3D extension). That is **Simple.3d**.

Scene files can be used in the existing *HotActions* projects, or they can be used to create a new project. To create a new project, launch the *HotActions* application, select the **New** command in the **File** menu of the *HotActions* main window. The **New** dialog opens. Select the **VS Project**, enter the project name, e.g., "Simple" in the **File Name:** field, and specify a folder to save it in the **Location:** field (see Figure 4.).



Type	Description
VS Project	VS Project File
Action Library	VS ActionLibrary File
Hotset Panel	VS Hotset File
Text Document	Text File

Location : D:\Focus\VS_Samples\New

Figure 4. The *New* dialog to create a new Project.

Now add the **Simple.3d** scene file to the **Scenes** group of the **Simple** project with the **Add File to Group** command of the contextual menu.

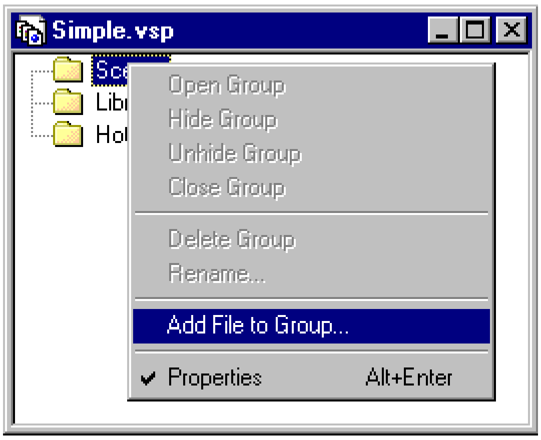


Figure 5. Adding a scene file to the project.

Create the ***Action Library*** in the new project to execute the basic actions with a scene. Also, we need at the minimum one ***Hotset*** to hold the buttons starting the ***Actions*** from the library.

To create the ***Action Library***, select **Action Library** in the **File Type:** area of the **New** dialog (see Figure 4.), which can be opened by the **New** command in the **File** menu of the *HotActions* application main window.

To create a ***Hotset,*** select **Hotset Panel** in the **New** dialog (Figure 4.). When first saving the created files, specify the file name and location. Most often all the files of the same project are created with the same name and differ only in extensions.

Specifying directories for video files and image files (operations with them are discussed in the further Sections) is a very important point. That can be done on the **Directories** tab of the *Properties* dialog for any group of files or a file in the project window.

9.3 Working with a Scene through Actions

Actions are an easy way to provide a scene control, such as displaying a scene, starting separate tracks of objects, etc. The following sequence of operations is standard for creating an ***Action*** in an open ***Actions Library***.

1. Right-click on an empty area of the ***Actions Library*** window. The contextual menu for the ***Actions Library*** window opens (see Figure 6.).

2. Select the **New Action** command. A new action is created with the default name *New Action*, it is possible to change its name immediately.

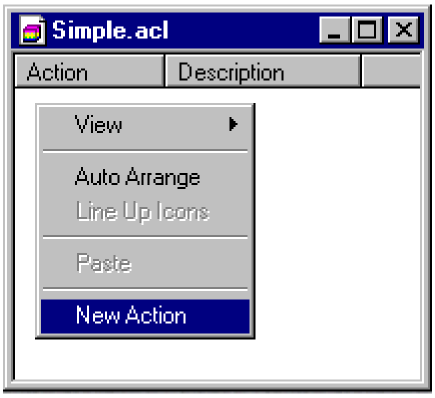
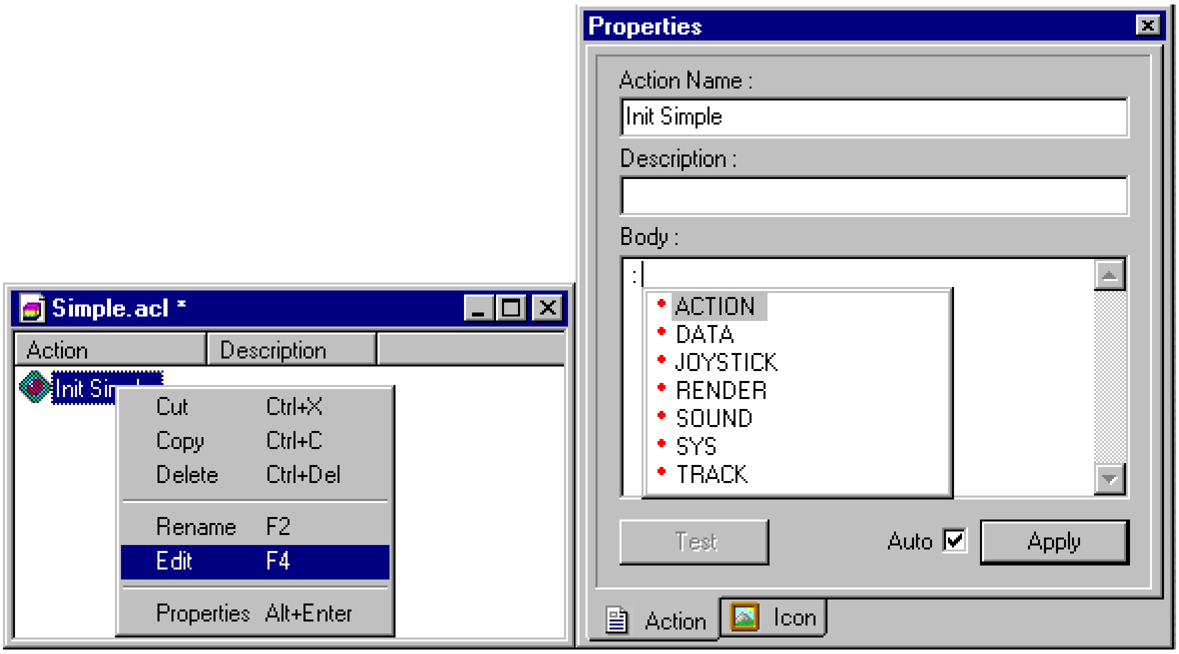


Figure 6. Creating a New Action in the library.

3. After a new ***Action*** is created, it is possible to specify its script commands in the assigned field. For that, right-click on the ***Action***. The contextual menu opens. Select the **Edit** command (see Figure 7, left). The *Properties* dialog opens (see Figure 7, right). The **Body** field of the **Action** tab is intended to enter and edit the script commands. Entering a colon («**:**») at the beginning of a line opens a local menu (Figure 7, right), where can select one of the available commands. Using a period symbol («**.**») brings up a corresponding menu which presents the available continuations for the command. That can be done several times if there are several continuations such as – **DATA.CURRENT.CAMERA**. The procedure makes easier creating commands and reduces the error probability. Commands can also be typed in manually into the **Body** field of the **Action** tab.



Action Name : Init Simple

Description :

Body : :

Auto ☑

Figure 7. Local menu for *Action* in the library (left) and Action tab of its *Properties* dialog (right).

4. After creating or changing the ***Action*** commands, it is possible to test their execution by pressing the **Test** button in the **Action** tab of the *Properties* dialog. Testing can also be done by double clicking on the Action name in the ***Action Library*** window. The **Apply** button simply saves the changes without starting the ***Action***.

Bringing a scene to the initial execution state is the task of ***STARTUP Actions*** discussed in the following Section.

9.3.1 Creating a *Startup Action*

The ***STARTUP Actions*** present a special group of actions. The task of ***STARTUP Actions*** is to bring a scene to the initial execution state, for example, when loading a scene.

Create a new ***Action*** in the ***Actions Library*** and name it **Init Simple**. The ***Action*** must contain the following commands.

- Load the scene file: DATA.OPEN = «simple.3d.»
- Make the required background visible while removing (hiding) the other. If the dark blue background is required, the green background is hidden. These commands are placed in individual ***Actions*** in the project, so they can be executed separately (**Blue backdrop** and **Green backdrop**):

  ```
  :DATA.NODE.back_blue.HIDE = 0
  :DATA.NODE.back_green.HIDE = 1
  :DATA.NODE.monitor.HIDE = 0.
  ```

- To control tracks in *Focus,* they must be named. There are two animated objects in the scene: the virtual camera, whose control is carried out by the **CAMERA1** ***Dummy*** object, and the screen named **monitor**. The following commands name the tracks belonging to the objects:

  ```
  :TRACK.TV.NODE = «monitor»
  :TRACK.TV.START = 30
  :TRACK.CAMERA.NODE = «CAMERA1»
  :TRACK.CAMERA.START = 0.
  ```

  The next to last command gives the **CAMERA1** track the name CAMERA, which is used for the further operations with the tracks of the object, for example, for startup.

- Create video streams for the scene and connect them with the available materials:

  ```
  //create a video stream LIVE_1
  :RENDER.VIDEO.LIVE_1.CREATE = 1
  //set the format of the video stream, initiate alpha-mixing, crop the frame edges:
  RENDER.VIDEO.LIVE_1.FORMAT = ALPHA, CROP
  //start the videostream:
  RENDER.VIDEO.LIVE_1.START = 1
  //wait when starting the video stream
  SYS.WAIT = VIDEO.LIVE_1.START
  //assign to the material named VIDEO_m a video texture from the LIVE_1 video stream,
  //which is to be imposed on all the objects of the scene with the material:
  RENDER.MATERIAL.VIDEO_m.SOURCE = LIVE_1
  //create the file video stream MSDS_1:
  :RENDER.VIDEO.MSDS_1.CREATE = 1
  //set the data source for the video stream MSDS_1:
  :RENDER.VIDEO.MSDS_1.DATA = UTRO.avi
  //start the file video stream MSDS_1:
  :RENDER.VIDEO.MSDS_1.START = 1
  //wait when starting the video stream
  //set the mode of continuous cycled playback (loop) of the video file
  :RENDER.VIDEO.MSDS_1.FORMAT=LOOP
  //assign the video stream MSDS_1 to the "monitor" material:
  :RENDER.MATERIAL.monitor_m.SOURCE = MSDS_1.
  ```

The only object in our scene with the material named **VIDEO_m** is the **!actress** plane. Thereby, we have provided the scene with live video.

The details about each command are discussed in *Focus Virtual Studio. Using the Script Commands User's Guide*. We shall only note that startup ***Actions***, in addition to the commands of loading a scene and naming tracks can contain any other commands modifying the ***Actions***.

Now every time the **Simple.3d** scene is opened within a project, the created ***Action*** named **Init Simple** must be executed. *HotActions* automatically executes the contents of the ***$STARTUP$*** folder when pressing the 💿 button on the main window toolbar. Therefore it is recommended to place the ***Startup Actions*** to the ***$STARTUP$*** folder. When observing the recommendation, just press the 💿 button to bring even an unfamiliar project to the execution state.

The ***$STARTUP$*** folder is always created when creating a ***Hotset***. By default, the folder is hidden, but it is possible to display its contents by using the alternative view of the ***Hotset*** (see Figure 8, right). For that, double right click on an empty area of the ***Hotset*** or select the **Configure** command in the contextual menu of the ***Hotset*** (see Figure 8, left).

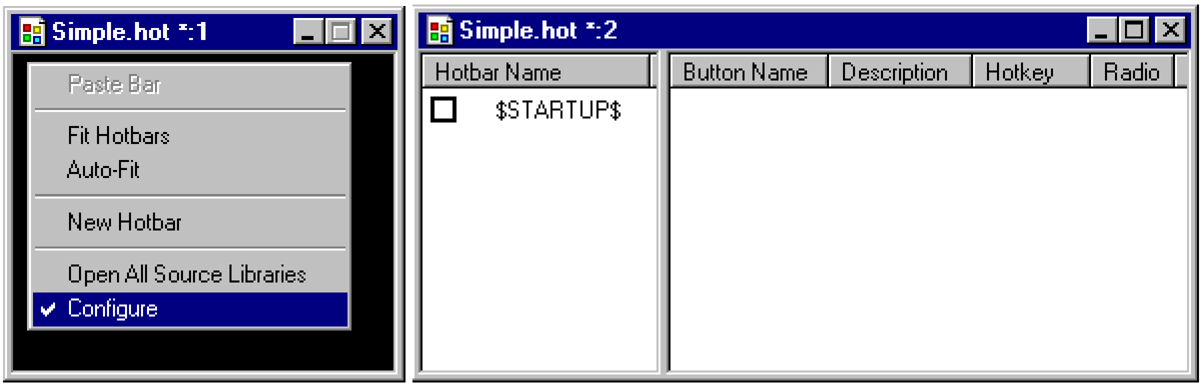


Figure 8. Contextual menu of a *Hotset* (left) and its alternative representation (right).

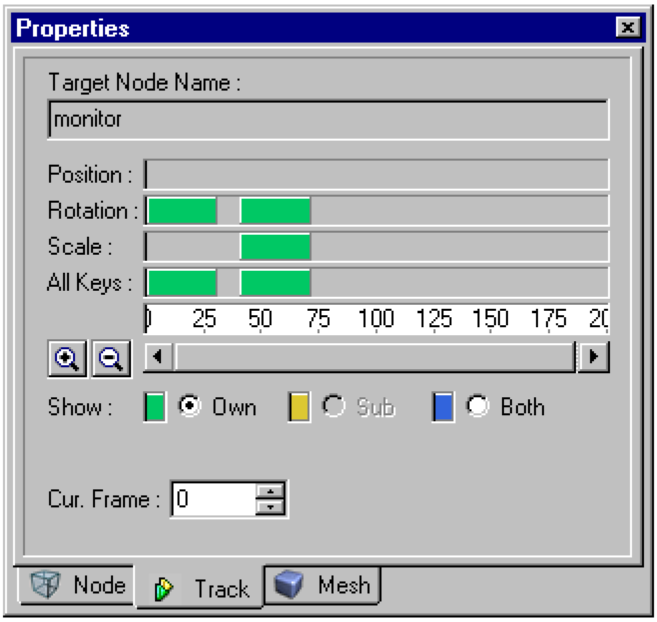
In the **Configure** mode, a ***Hotset*** is presented as a tree structure with the groups/folders of the **Hotset** in the left area and the contents of the selected folder in the right area. The contents present references to ***Actions*** corresponding to the ***Hotbar*** buttons. When a ***Hotset*** is empty, the only folder it contains is ***$STARTUP$*** .

Drag and drop the ***Action*** named **Init Sample** from the ***Actions Library*** to the ***$STARTUP$*** folder and close the window of the ***Hotset*** alternative representation. Since the ***$STARTUP$*** folder is hidden by default (the check box to the left of its name is unchecked), no buttons corresponding to **Init Sample** appear in the ***Hotset***.

9.3.2 Using Named Tracks

Now the tracks of the objects **CAMERA1** and **monitor** are named (see Section 9.3.1) and available for control.

The **Track** tab of the *Properties* dialog displays information of any object track separately by its components (Figure 9).



Target Node Name : monitor

Show : 🟩 ● Own 🟨 ○ Sub 🟦 ○ Both

Cur. Frame : 0

Figure 9. The *Track* tab of the *Properties* dialog.

The colored rectangles, each on the corresponding line, display the changes in the position, rotation, and scale of the object (see Figure 9). The color of the rectangles depends on which radio button is selected in the **Show** group.

The **Show** radio-buttons provide displaying information for the tracks grouped as follows:

- **Own** – displays the own tracks of the object;
- **Sub** – displays the the tracks of the object descendants;
- **Both** – displays the tracks of the object and its descendants.

The information presented can be effectively used in the script commands. For example, create an ***Action*** for a monitor to appear in the scene. Name the action **Show TV** and place the following commands in its body:

```
:DATA.NODE.monitor.HIDE = 0
:TRACK.TV.GOTO = 30.
:SYS.WAIT = “TRACK.TV”.
```

The :TRACK.TV.GOTO = 30 command starts the **TV** track from the current frame to the 30th providing the appearance of a monitor.

The result of an analogous command (place it in an ***Action*** named, e.g., **Hide TV**) is gradual removing the monitor from the scene:

```
//start the track from the 40th to the 72nd frames:
:TRACK.TV.START = 40,72
:SYS.WAIT=”TRACK.TV”
:TRACK.TV.START = 0
:DATA.NODE.monitor.HIDE = 1.
```

In the same way, create the ***Actions*** that use the **CAMERA** track to control the virtual camera:

The **Go to Close Up** action is used to make the camera approach the actor. The script commands:

```
:DATA.CURRENT.CAMERA = «Camera01»
:TRACK.CAMTRACK.GOTO = 90
SYS.WAIT = “TRACK.CAMERA”.
```

The **Go to Long shot** action with the following script commands:

```
:DATA.CURRENT.CAMERA = «Camera01»
:TRACK.CAMTRACK.GOTO= 50
SYS.WAIT = “TRACK.CAMERA”.
```

We will alsow need an ***Action*** to make the camera approach to the monitor object and focus on it. That is the **Go to TV** action with the following script commands:

```
ACTION.START = “Show TV”
SYS.WAIT = “ACTION.Show TV”
:DATA.CURRENT.CAMERA = «Camera01»
:TRACK.CAMERA.GOTO = 120
SYS.WAIT = “TRACK.CAMERA”
```

The last ***Action*** using named tracks that we will mention here is **Distant View** used to switch to the distant view. The the script command is:

```
:DATA.CURRENT.CAMERA = «Camera01»
:TRACK.CAMERA.START = 0.
```

The use of specific frame numbers is based on the knowledge of what is happening to object in the frames. At the same time, the **Track** panel gives sufficient information to figure out the tracks of objects even if you are unfamiliar with the contents of a scene.

After naming the track of any object with a command like this: TRACK.*NameofTrack*.NODE = *NameofObject* in the startup ***Action***, we can start the entire track. The initial frame is zero, and the last frame can be approximately set with the help of the **All Keys** line of the **Track** tab. For example: TRACK.TV.START = 0, 180.

After that, it is possible to allocate a separate range of frames to an *Action* by narrowing the range of frames.

9.3.3 Working with Virtual Cameras

There are two cameras – **Camera01** and **Camera02** in the scene controlled by the **CAMERA1** and **CAMERA2** ***Dummy*** objects.

Camera01 is animated. The animation is used for such actions as zooming the camera on the *Actor* or appearance of a “*monitor*” as described in the previous Section. The **Camera02** virtual camera has no animation and remains directly focused on the “*monitor*”.

To switch the current camera in a scene, perform the following commands (place them in separate ***Actions*** named **Camera1** and **Camera2**):

```
//sets Camera01 as the current camera :DATA.CURRENT.CAMERA=«Camera01»;
//sets Camera02 as the current camera :DATA.CURRENT.CAMERA=«Camera02».
```

9.3.4 Working with video files

Script commands allow to create an ***Action*** placing the video image from, e.g., an video file on a virtual monitor. This can be done in the initial ***Action*** (see Section 9.3.1), as well as in a separate ***Action*** named **Play DV**:

```
//create a file video stream MSDS_1:
:RENDER.VIDEO.MSDS_1.CREATE = 1
//set a data source for the video stream MSDS_1:
:RENDER.VIDEO.MSDS_1.DATA = UTRO.avi
//start the video stream MSDS_1:
:RENDER.VIDEO.MSDS_1.START = 1
//wait for the stream to be started:
:SYS.WAIT = VIDEO.MSDS_1.START
// assign the video stream MSDS_1 to the TV material:
:RENDER.MATERIAL.monitor_m.SOURCE = MSDS_1
//after the video stream (file or «live» video) is created and started, it can be assigned as a material to any object in a scene, for example, the video from LIVE_1 can be placed on the “monitor”:
```

:RENDER.MATERIAL.monitor_m.SOURCE = LIVE_1

//wait for the end of the file playback:

:SYS.WAIT = «VIDEO.MSDS_1.END»

//restore the initial material of the “monitor”:

:RENDER.MATERIAL.monitor_m.SOURCE = 0.

The script commands of the ***Action*** are in many respects similar to those when creating the **LIVE_1** video stream (see Section 9.3.1).

To return the material of an object to its initial condition after working with a video file, assign to it a nonexistent stream as the video source. If a stream with the assigned name is not found, the initial map is used. For example, return the original texture to the object “monitor”:

:RENDER.MATERIAL.monitor_m.SOURCE = 0.

An example of more comprehensive work with a video file using the ***Actions*** described above is given in Section 9.3.6.

9.3.5 Working with Graphic Files

The *HotActions* application provides working with graphic files with the help of script commands.

As an example, let us create an ***Action*** named **Show Slide** which will place a JPG file image on our virtual monitor. It is implied that the Slide.jpg file exists in the project folder:

:DATA.MATERIAL.monitor_m.MAP = «Slide.jpg»

To restore the initial condition of the object material, it is sufficient to perform the following command:

:DATA.MATERIAL. monitor_m.RESET = 1

When executing the command, the Slide.jpg file will be deleted from the video adapter working memory. That means that the next assignment of the texture to the “monitor” material will require using additional system resources to convert the Slide.jpg file into the internal format of the video adapter memory. To avoid that, use the following command to returt the initial image to the monitor_m material:

:DATA.MATERIAL.monitor_m.MAP = 0

When executing the command, the Slide.jpg file is not deleted from the video adapter working memory and the next assignment will not require an additional convertion.

At the same time, there is no point in displaying a slide on the “monitor” if the latter is absent in the scene. Sure, it’s possible to manually execute the ***Action*** named «**Show TV**» and then the ***Action*** named «**Show Slide**». But there is a better way commected with the ability to start one ***Action*** from another. The following Section covers working with a sequence of actions in a scene.

9.3.6 Launching Actions from Other Actions

The previous Section covers creating the «**Show Slide**» ***Action***, which places an image from a JPG file onto the virutal monitor. It is possible to create a more complex action using existing ***Actions***. For example, let us create an ***Action*** named «**TV Show Slide**», in which a “monitor” appears, shows a JPG file image slide for five seconds, and then disappears. Besides, it’s required to focus the camera at what’s going on at appropriate moments. Since all described actions have already been created and stored in the *Actions* library, now we will use them.

Use the “ACTION.Action_Name” message generated on the completion of each ***Action*** in order to make all the actions occur in consecutive order. Such messages are displayed in the **Debug Output** window of the application (see Section 6 of *Focus Virtual Studio. HotActions User’s Guide* and Section 18 of *Focus Virtual Studio. Using the Script Commands User’s Guide*). The right parts of the messages can be used as the parameters for the SYS.WAIT = “event” command to wait for the completion of an ***Action*** execution:

//focus the camera at the position where the “monitor” will appear and produce it:

:ACTION.START = «Go to TV»

//waiting for the completion of the ***Action*** focusing the camera:

:SYS.WAIT = ACTION.« Go to TV»

```
//produce the JPG image file on screen with the help of the Action discussed in the previous Section:
:ACTION.START = «Show Slide»
//a 5 second pause:
:SYS.DELAY = 5.0
//start the Action removing the “monitor” from the scene:
:ACTION.START = «Hide TV»
//waiting for the comletion of the Action removing the “monitor” from the scene:
:SYS.WAIT = ACTION.« Hide TV»
//refocus the camera on the actor:
:ACTION.START = «Go to Close Up»
:SYS.WAIT = ACTION.« Go to Close Up»
//return the initial material to the “monitor”:
:DATA.MATERIAL.monitor_m.MAP = 0
```

To make an ***Action*** produce a certain video file, create an ***Action*** named «**TV Show DV**» slightly differing from the one afore-cited:

```
//focus the camera on the place where the “monitor” will appear and produce it on screen:
:ACTION.START = «Go to TV»
// waiting for the completion of the Action focusing the camera:
:SYS.WAIT = ACTION.«Go to TV»
//play the video from a file on the virtual monitor using the Action created in Section 9.3.4
:ACTION.START = «Play DV»
//waiting for the completion of the Action execution:
:SYS.WAIT = ACTION.«Play DV»
//start the Action removing the virtual“monitor” from the scene:
:ACTION.START = «Hide TV»
//waiting for the completion of the Action removing the virtual “monitor”:
:SYS.WAIT = ACTION.«Hide TV»
// refocus the camera on the actor:
:ACTION.START = «Go to Close Up»
:SYS.WAIT = ACTION.«Go to Close Up».
```

9.3.7 Creating Buttons for the Execution of Specific Actions

Having a ready supply of ***Actions*** in the *Actions Library* we can create button panels assigned to the ***Hotset*** and used as the major interface for controlling a scene.

Actions in the *Actions Library* can be managed as files in a folder. They can be copied, deleted etc. The operations can be performed with a group of ***Actions*** selected in a standard way (clicking with the **Shift** or **Ctrl** key pressed).

When a group of selected ***Actions*** is dragged to an empty space in the ***Hotset***, a new ***Hotbar*** is automatically created for them. Each button on a ***Hotbar*** refers to a certain ***Action***. By default, a button name coincides with the action name. It is possible to add buttons to any ***Hotbar*** by dragging ***Actions*** directly onto it from the ***Actions library***.

Thus, a ***Hotset*** is a set of ***Hotbars***. The ***Hotsets*** are convenient in that the Hotbars belonging to them can be hidden, displayed or moved simultaneously.

,You can change the name of a ***Hotbar*** and representation of any button on it in the *Properties* dialog for the ***Hotbar*** (the ***Hotbar*** tab of the ***Properties*** dialog). You can set or change any button icon of a ***Hotbar*** in the *Properties* dialog for the button (the ***Button*** tab of the ***Properties*** dialog) The work with ***Hot Buttons***, ***Hotbars*** and ***Hotsets*** is described in more detail in the corresponding Sections of *Focus Virtual Studio. HotActions User’s guide*. An example of a ready ***Hotset*** is shown on Figure 10.

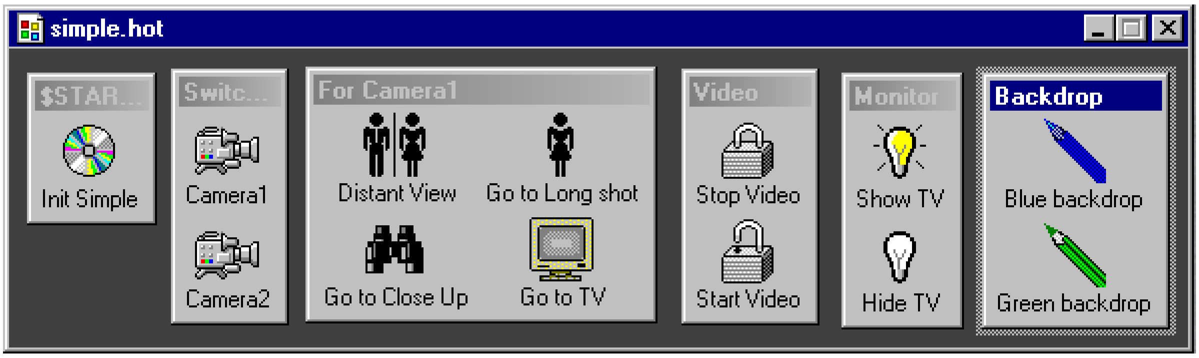


Figure 10. Ready *Hotset* and its *Hotbars* – the basic tools for working in *HotActions.*

10 Exporting Scenes from 3D Studio MAX to a *.3D File

After a scene is created in *3D Studio MAX*, export the file with the *.MAX extension to a file with the *.3D extension, which presents the working format of scenes to be used in the *HotActions* application.

10.1 Installing Plug-ins for *3D Studio MAX* (*MAYA*)

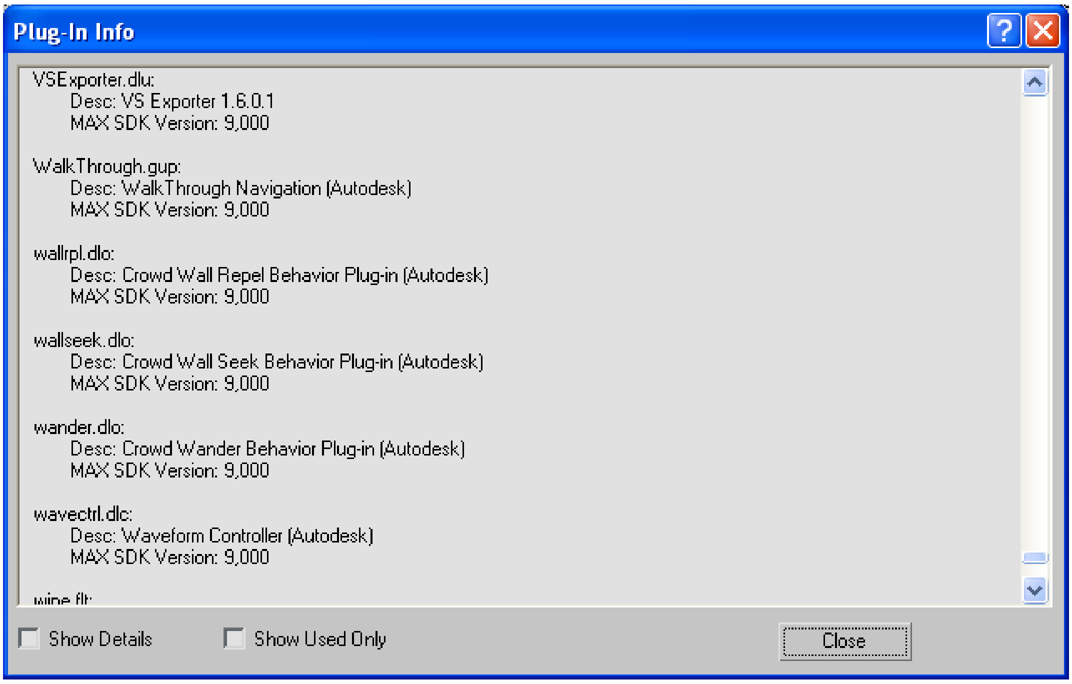
To convert the format of a scene created in *3D Studio MAX* into the scene format of the *Focus* virtual studio, install a special export plug-in included with the *Focus* software package.

If the *3D Studio MAX* application is installed on your computer, the installation of the export plug-in is performed automatically when installing the *Focus* software in the folder with the appropriate modules of the *3D Studio MAX* application (as a rule, the folder is named “plug-ins”). Besides, when installing *HotActions*, a special *Exporters* folder is created in the directory the *HotActions* application is installed in. The *Exporters* folder contains export modules from the *3D Studio MAX* graphics editor (versions from 4.0 to 3DSMax 2010 are supported) and from different versions of *Maya*: **MayaExporter4.5**, **MayaExporter5.0**, and so on. As a rule, the directory is as follows **C:\Program Files\Focus Software\HotActions**, if *Focus* is installed on the **C:** drive of your computer.

If required to install the export modules on a different computer, where *3D Studio MAX* or *Maya* is installed, copy the **VSExporter.dlu** file from the *MaxExporter* folder or **VSExporter.mll** file from the folder with the number of the corresponding version of *Maya* into the plug-ins folder of *3D Studio MAX* (*MAYA*) of that computer. Another way to build in the module is installing the *HotActionsDesign* application.

To check if the installation of the plug-in is successful, open the *Plug-In Info* dialog

Figure 11) with the **Summary Info ...** command of the **File** menu of the *3D Studio MAX* application.



```
VSExporter.dlu:
      Desc: VS Exporter 1.6.0.1
      MAX SDK Version: 9,000

WalkThrough.gup:
      Desc: WalkThrough Navigation (Autodesk)
      MAX SDK Version: 9,000

wallrpl.dlo:
      Desc: Crowd Wall Repel Behavior Plug-in (Autodesk)
      MAX SDK Version: 9,000

wallseek.dlo:
      Desc: Crowd Wall Seek Behavior Plug-in (Autodesk)
      MAX SDK Version: 9,000

wander.dlo:
      Desc: Crowd Wander Behavior Plug-in (Autodesk)
      MAX SDK Version: 9,000

wavectrl.dlc:
      Desc: Waveform Controller (Autodesk)
      MAX SDK Version: 9,000

wine.flt:
```

□ Show Details □ Show Used Only

Figure 11. The *Plug–In Info* dialog of the *3D Studio MAX* application.

If the **VSExporter.dlu** file is not present in the plug-ins list, check if the full path to the directory is specified correctly. For that, select the **Configure Paths...** command in the **Customize** menu (Figure 12). The *Configure Paths* dialog opens. Select the *Plug-Ins* tab (Figure 13) to perform this check, in the **Customize** menu (Figure 12) choose the command **Configure Paths...**. In the opened *Configure Paths* dialog select the. If *Additional 3D Studio MAX plug-ins* are not set there, add the path manually with the help of the **Add...** button.

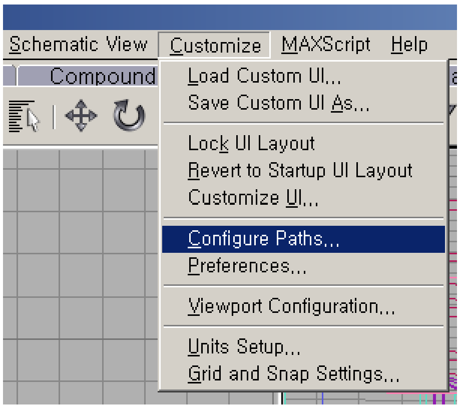
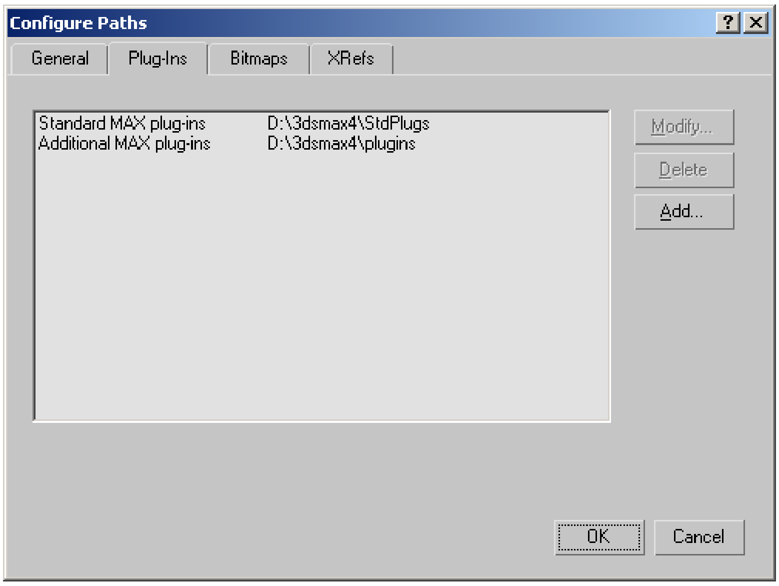


Figure 12. The *Customize* menu.

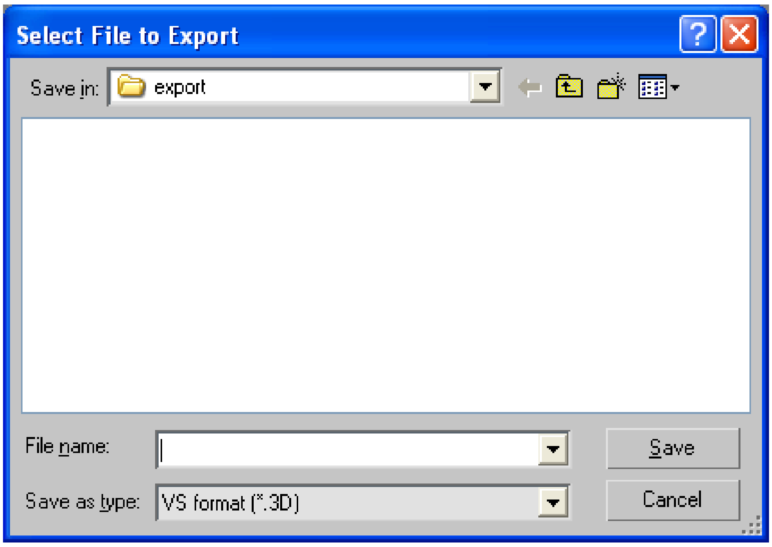


Standard MAX plug-ins	D:\3dsmax4\StdPlugs
Additional MAX plug-ins	D:\3dsmax4\plugins

Figure 13. The *Plug-Ins* tab of the *Configure Paths* dialog.

10.2 Using the Export Utility

To export a scene, select the **Export...** command in the **File** menu. The *Select File to Expot* dialog opens (Figure 14). Select a folder, enter the file name in the **File name** field. Press **Save** to complete the export.



Save in: export

Save as type: VS format (*.3D)

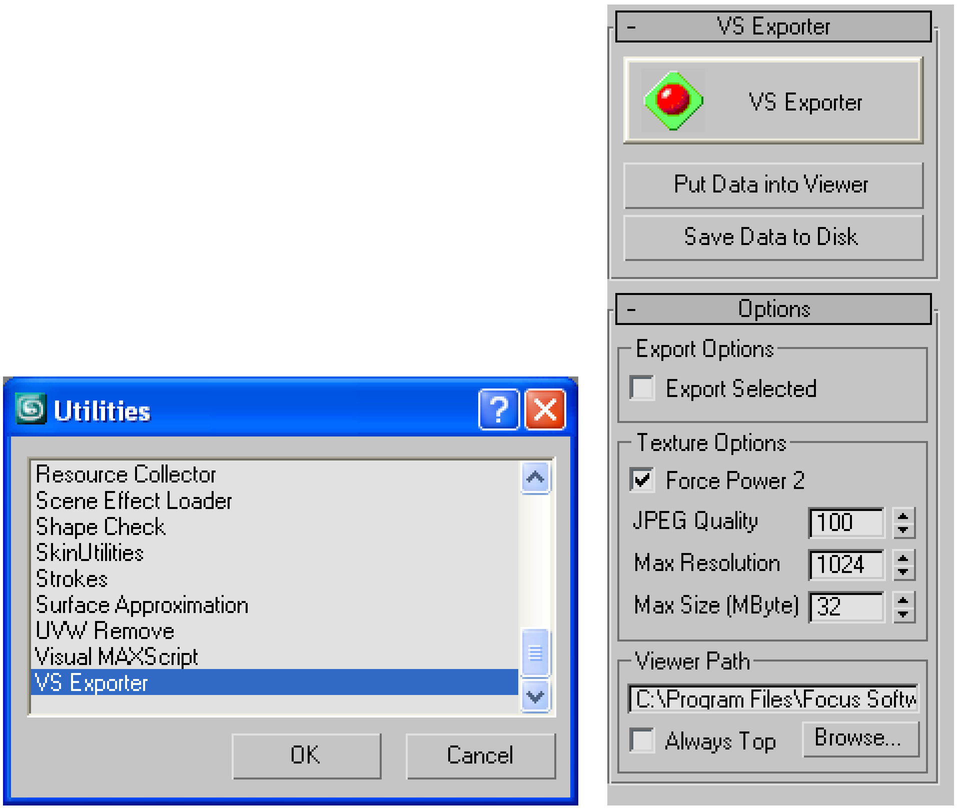
Figure 14. Dialog for exporting files.

11 Using *VS Exporter*

Using *VS Exporter* (Figure 15, right) allows you to preview and save a scene in the 3D format without exiting the *3D Studio Max* application.

11.1 Loading *VS Exporter*

Open the **Utilities** tab page and press the button of the *VS Exporter* child page (Figure 15, right). If the button is absent on the tab, press **More...** and select *VS Exporter* in the list of the **Utilities** dialog (Figure 15, left), then press **OK**. The button of the *VS Exporter* child page will appear on the **Utilities** tab page (Figure 15, right).



□ Export Selected

☑ Force Power 2

JPEG Quality 100

Max Resolution 1024

Max Size (MByte) 32

C:\Program Files\Focus Softw

□ Always Top

Figure 15. List of the additional utilities of *3Ds MAX* and the *VS Exporter* dialog.

On the **Options** child page of the *VS Exporter* dialog, set the export parameters (in the **Export Options area**), texture parameters (in the **Texture Options** area), and specify the path to the ***HotActions.exe*** application to launch it when previewing a scene (in the **Viewer Path** area).

Checking the **Export Selected** check box of the **Export Options** area provides exporting not all the objects of a scene but previously selected only.

The **Force Power 2** option (of the **Texture Options** area) determines scaling all the textures of a scene so that they equal 2^N texels, where N is an integer (see Section 2.1.10). If the option is off when exporting, the sizes of textures will not transform and remain as they were appointed in the 3DMAX scene. In that case, the rescaling to 2^N texels will be performed each time when loading the scene into the ***HotActions*** application. Thus, checking the **Force Power 2** check box provides performing the procedure of rescaling textures only once, which saves time when further loading the scene into the ***HotActions*** application.

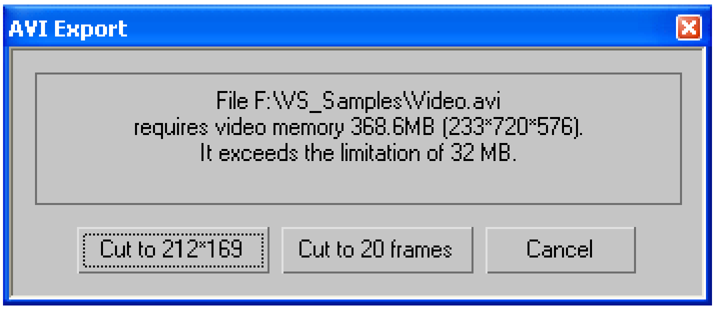
When exporting, you may set the textures quality (**JPEG Quality**), which affects the size of the exported file, set the maximum allowable texture resolution (**Max Resolution**), which may be important when exporting a video file as a texture, or set the maximum allowable texture size (**Max Size**) in MB.

Previewing a scene before export in the ***HotActions*** application is described in the following Section. The preview is possible provided the path to ***HotActions.exe*** is specified in the **Viewer Path** field. The path can be specified in the **Open** dialog after pressing the **Browse...** button. Checking the **Always Top** check box provides placing the *HotActions* windows always over the windows of other applications.

11.2Viewing a Scene in *VS Exporter*

To see the results of the last changes of a scene in ***HotActions***, press the **Put Data into Viewer** button. That launches *VS Viewer* with the loaded data of the current scene.

If the size of a texture video file exceeds the maximum allowable texture size specified in the **Max Size** field of the **Texture Options** area (Figure 15), when previewing the scene in *VS Viewer* or exporting the scene (see the next Section), a message prompting to reduce the video file size by changing the frame size or shortening the duration of the video file (number of frames) (Figure 16) is displayed.


AVI Export

File F:\VS_Samples\Video.avi
requires video memory 368.6MB (233*720*576).
It exceeds the limitation of 32 MB.

Figure 16. Message about exceeding the allowable texture size when exporting a scene.

Exporting a scene or preparing a preview may take some time, the percentage of already exported data being displayed in the lower part of the *3DS MAX* application window (Figure 17).

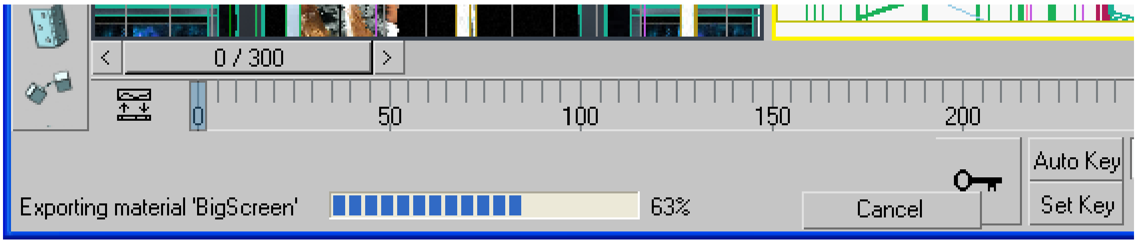

Exporting material 'BigScreen' 63%

Figure 17. Status bar of exporting a scene from 3DS MAX to HotActions.

There are some qualifications on the scene contents when exporting: a scene must not be empty and must contain a virtual camera.

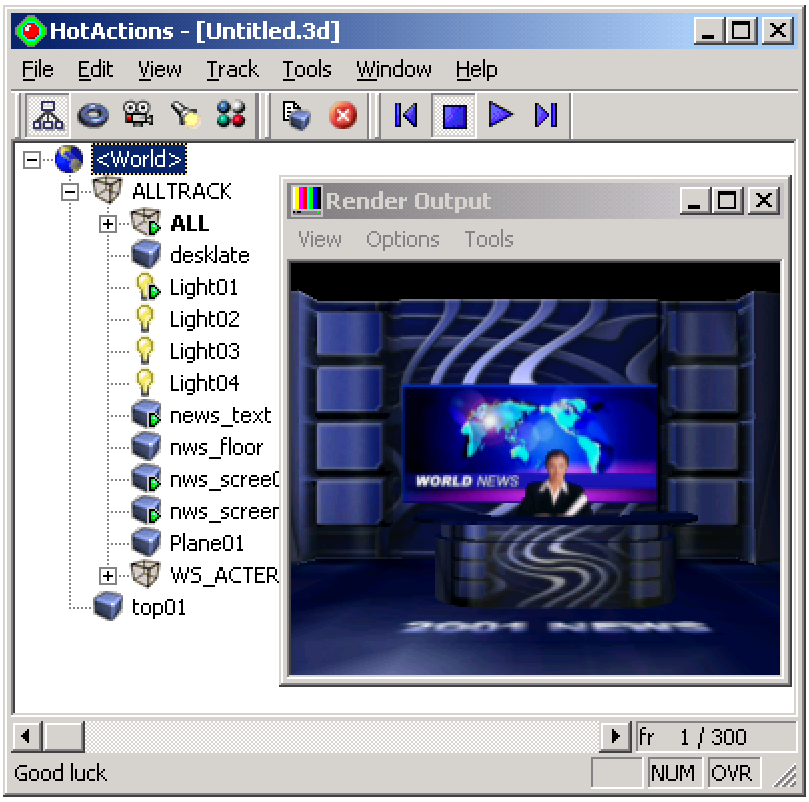


Figure 18. The application window after the launching of *VS Viewer*.

The scene components in *VS Viewer* are distributed among the appropriate panes according to their types. The toolbar allows to switch from one pane to another.

The buttons are used to play back and stop the general track of the scene. For more details about work with a scene, see Section 3.3.4 of *Focus Virtual Studio. HotActions User's Guide*.

11.3 Saving a Scene in the *Focus* Format with *VS Exporter*

To save a scene, select the **Save as...** command in **File** menu of the *HotActions* menu bar (Figure 18). The *Save As* dialog opens. Specify a directory and a file name. It is possible to save a scene as a *.3d file without previewing. That can be done with the **Save Data to Disk** button on the **Utilities** tab (Figure 15).