**Preparing and exporting 3D scene from 3dsMax for AllCast**

AllCast is ​​a powerful tool for creating, recording and broadcasting multimedia content in the field of broadcasting and education. At the same time, this software product is aimed primarily at medium and small companies, and even individual users. Those the user of the AllCast program should not be required to have deep special knowledge beyond the scope of his professional activity.

This means that 3D scenes must be prepared in a special way that allows an ordinary AllCast user (who has no idea about KeyFrame animation, materials, texture coordinates, etc.) to control animation, assign media sources to scene elements, etc.

AllCast spends significant computer resources on processing incoming and outgoing video streams, media files, title elements, etc. This means that 3D scenes (and several of them can be used simultaneously in one AllCast project) must be optimized so as to economically use computer resources, to ensure fast loading of 3D scenes.

This article is devoted to the peculiarities of preparing 3D scenes for use in AllCast.

**1. General information.**

The virtual studio built into AllCast does not use expensive and complex tracking systems. However, in many cases it can achieve the same results. How is this achieved? In AllCast, the diffuse texture of the material of a special object of the 3D scene (a flat rectangle) is replaced with a "live" image of the actor (with an alpha channel) from a stationary real camera. When moving the virtual camera, the viewer sees a picture similar to the one as if the real camera did not stand still, but moved in concert with the movement of the virtual camera.

This technology has both pros and cons:

* The disadvantages include restrictions on the angle of view of the virtual camera relative to the plane of the "Actor".
* On the plus side - the ability to animate a virtual camera in ways that are difficult (or even impossible) for a real camera.

AllCast Render Engine allows to use, for example, real-time PBR (physically-based rendering) technology. However, in a 3D scene intended for use with AllCast, only standard materials and standard lights are allowed.

This is due to the following reasons:

* firstly, the desire to save computer resources as much as possible, to speed up the loading of 3d scenes.
* secondly, the quality of the real-time PBR picture is significantly worse than the quality of the unreal-time rendering ("Vray", "Corona", etc.).

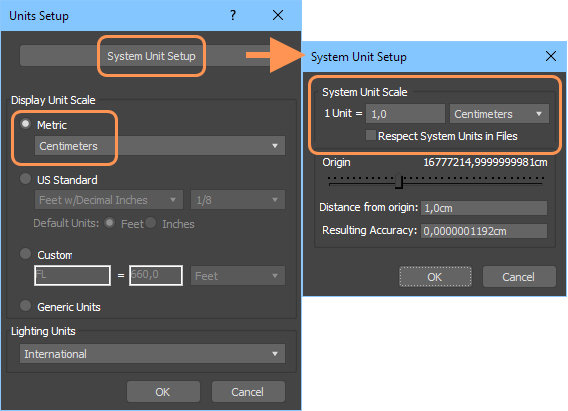
Therefore, creating a scene in 3ds Max using plugins like "Vray" or "Corona", with subsequent conversion through baking textures into a standard 3ds Max scene will make the scene "light", and will make the picture (when rendering in real time in AllCast) - high quality.

Thus, it is desirable to have two 3D scenes:

* Initial (for plugins "Vray", "Corona", etc.), in which you can use any materials and any number of any light sources.
* Result, which is obtained from the original by converting to a "standard" 3ds max scene. It is from this scene that the export to a .fbx file is made for use in AllCast.

! Use Map Channel 3 when baking.

To be compatible with the AllMix unit system, set up the 3ds Max unit system as follows:



**2. 3D Scene optimization**

**2.1. Materials, light sources, camera.**

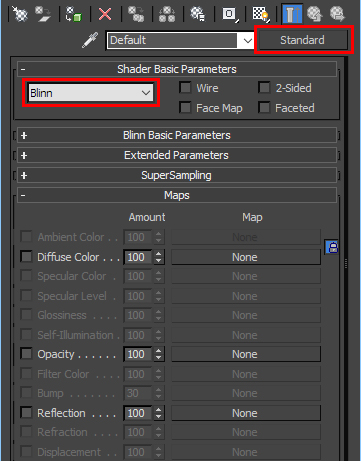
After baking textures, all materials, lights, cameras, etc. must be converted to standard 3dsMax types:

• camera - "FreeCamera"

• light sources - "Omni", "Spot", "Direct"

• material - "Standard (Blinn)"

Currently, AllCast uses only three types of textures - "Diffuse", "Opacity" and "Reflection". It is best to make baked materials self-illuminated. This will remove "unnecessary" light sources, keeping their number to a minimum. Or do without them.

There should be one camera. The different camera positions are set by the animation track. Values ​​for clipping planes of the camera in the properties of the camera should be set manually (Clipping Planes / Clip Manually) in accordance with the parameters of this 3d-scene. The general rule is to set "Far" to the order of the scene dimensions, and select "Near" until the "flickering" of closely spaced edges disappears. 

**2.2. Shadows.**

As recommended above, it is better to bake the illumination of scene objects into textures. However, these tips are not suitable for dynamic shadows (shadows from moving objects and the actor). In this case, it is recommended to combine baking scene lighting into textures and using the "Spot" light source.

**2.3. 3D Objects.**

It is undesirable to use groups (command Group). Especially for animated objects.

On the contrary, to combine objects (command Attach) is always good both for saving resources and for speed of loading. This must of course be done before baking textures.

Before exporting, it is recommended to apply the "Convert to Editable Mesh" and (then) "ResetX-Form" procedures to all objects in the scene. This will also significantly speed up the loading of the scene in AllCast and ensure its correctness.

**3. Special names and properties of objects and materials in the 3d scene.**

**3.1. 3d scene objects available for editing in All¹Mix**

• The list of objects of the 3d scene, available for editing, adds objects with a name starting with the @ symbol. For example: @Table (also with a name containing the words "monitor", "video" or "actor" - for compatibility with previous versions).

• Objects with a material name beginning with the @ symbol are added to the list. For example: @screen\_material.

• In the 3D editor All¹Mix it is possible to choose to display only one object out of several. To do this, their names must have a common part to the left of the @ symbol. For example: Table@wooden, Table@glass, : Table@metal, etc.

All¹Mix allows the user to change the position, orientation and scale of such objects, as well as to replace the diffuse maps of their materials with other image sources (images, video files, streaming video, etc.). In this case, the object must contain only one material.

• If the object's material contains a transparency map, then it will be possible to assign media with an alpha channel to this object (in particular, the "ChromaKey" technique can be applied).

• If the first character in the material name is an exclamation point (for example "!MaterialName" "!@MaterialName " ), then the texture coordinates of the object in AllMix will be assigned in each frame in the "overlay" mode (similar to the "Camera Map" modifier in 3ds Max) .

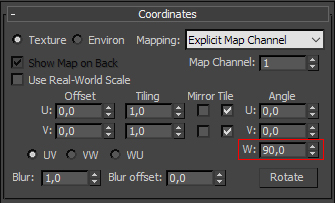
**3.2. 3D object "Actor".**

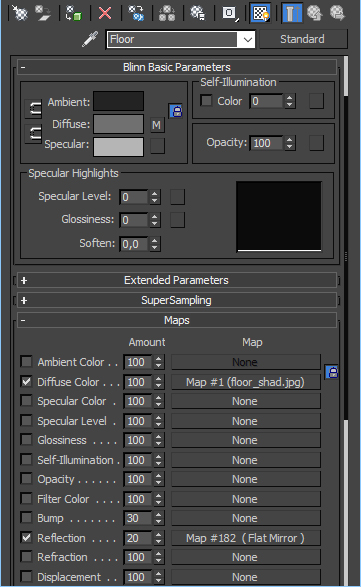
3D object "Actor"must be a planar object (rectangle) with aspect ratio 16: 9 (HD) or 4: 3 (SD), depending on the type of camera filming the real actor.

If the first character in the name of this object is an exclamation mark (for example "! MyActor"), then the object will always "look" at the virtual camera. Starting from version 2.0.076, the axis of rotation is determined by the AllCast program by the orientation of the Gizmo of the planar texture coordinates (the "top" of the Gizmo must coincide in orientation with the "top" of the scene). The exclamation mark will also work in earlier versions of AllCast if "Actor" is created in the following way:

* Create a Plane object in the Front window.
* Apply the "Reset XForm" procedure to it
* Apply to it the procedure "Convert To Editabel Mesh"

There is a technique that allows a virtual camera to "zoom in" to the waist plan of a full-length figure without losing the quality of the "Actor" image. To do this, the real video camera filming the actor is rotated 90 degrees. As a result, the image from the camera will also be rotated, and we have to compensate for this rotation in the 3d scene.

 If there is no exclamation mark in the name of the "Actor" object, then simply rotate the rectangle 90 degrees. If "!" is present, we must also assign texture coordinates so that the "top" of the Gizmo coincides in orientation with the "top" of the scene. Then fix the picture by turning it in the material editor.

**4. Reflections.**

**4.1.** **Real-time reflections**

AllCast has the ability to render real-time reflection for planar objects. Such objects should also not contain more than one unique material.

A planar object can be of any shape, and even consist of separate exploded fragments. The main thing is that all the elements of the object lie in the same plane. The "pivot" of the object must also be in this plane.

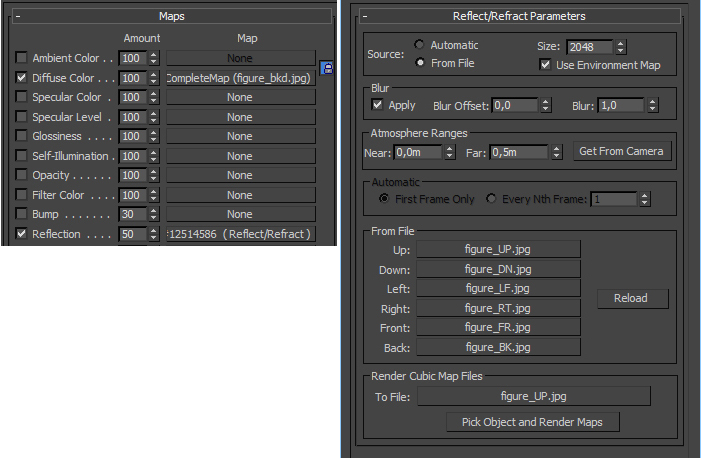
As a rule, the objects for assigning reflection are the floor, table surface, etc.

In order for AllCast to perceive the object as reflecting, it is necessary to set the "checkbox" for the "Reflection" option in the material assigned to it and set the "weight" of the reflection.

The "map" field must contain a "FlatMirror" procedural map. The amount of reflection blur is set by the "Bloor" parameter in the "FlatMirror" map properties.

**4.2. Non-dynamic reflections.**

For scene objects, it is possible to set static reflections using texture maps of the Reflect / Refract type in the material editor.



**4.3. Simulated reflections.**

Simulated reflections are set in the usual way - through a single texture map:



**5. Анимация.**

**5.1. The principle of building animation in AllCast.**

In AllCast, playback of animation tracks (created in 3ds Max) is performed between some named object "States", with unique sets of parameters: "position", "rotate", "scale".

The principle of "State" animation is that when working in AllCast, the user selects not a specific segment of the animation track to play, but “destination” - the "State" to which the object must “come”. At the same time, the AllCast program automatically selects the segment of the track, the beginning of which corresponds to the current state, and the end - to the target state. If there is no such segment of the track, then the transition from the current "State" to the final "State" is instantaneous.

Why does AllCast use “State Animation” technology to control animation instead of seemingly more understandable “Action Animation” technology? ("Action Animation" is such a control method, when each segment of the object's animation has its own button, which is responsible for starting the animation for a given segment of the track).

For example, consider a 3D scene in which there are N basic positions of a virtual camera and in 3ds Max segments of animated tracks are created that correspond to all possible “moves” of a virtual camera between these basic positions. According to the combinatorial formula, the number of such segments (and, consequently, the number of corresponding buttons for "Action Animation" technology) should be N! / (N-2)!

That is, for a scene with three, six or ten basic positions of the virtual camera, we will need, respectively, 6, 30 or 90 buttons for "Action Animation" technology. But to control the animation according to the “State Animation” technology, we will need only 3, 6 or 10 buttons, respectively. The difference is significant.

In addition, "State Animation" technology greatly simplifies the work of the operator. He does not need to think about the basic position of the virtual camera at the moment. Simply select the button corresponding to the desired next camera position.

**5.2. Creating a "State" - animation in 3DS Max**

**5.2.1.** To avoid mistakes and confusion, we recommend starting with the following:

* First of all, the time configuration of the animation should be converted to the “Frames” format (Time Configuration / Time Display / Frames), since for AllCast the description of the lengths of the animation tracks is set in this format.
* Then, in the "Auto Key" mode, after some (small) intervals on the frame track, we create the base positions of the virtual camera. The keys corresponding to these positions must contain values ​​for "position", "rotate", "scale". (For the camera - only "position" and "rotate") To make sure that the keys contain all the necessary parameters, you can apply the "Create Key" option to each created base position on the personnel track.
* Each basic camera position (i.e., “State” with a unique set of parameters “position” and “rotate”) will come up with unique and “speaking” names (for example, we have four “States” and we will call them “Near”, "Center", "Left", "Right") and for now keep them in mind. They will come in handy later.

**5.2.2. Creating animated transitions and their description.**

We will need the created keys as samples when creating segments of animated transitions between the basic positions of the virtual camera. We will copy them to the beginning and end of the segments of the track that define the animation of transitions between a pair of "Steits". This will guarantee that the values ​​of the keys will be the same for the same "State", no matter what position of the track it is.

* By the way, it is useful for our "exemplary" keys to set properties immediately (for example: Easy From = 25, Easy To = 25, Continuty = 0).

The next step is to create segments of the object animation. The segments themselves can be of any length and contain any internal keys. However, as we have already indicated, they must begin and end with keys copied from "exemplary" "Steits".

We need to tell the AllCast program which segments of the animation to use for transitions between "States".

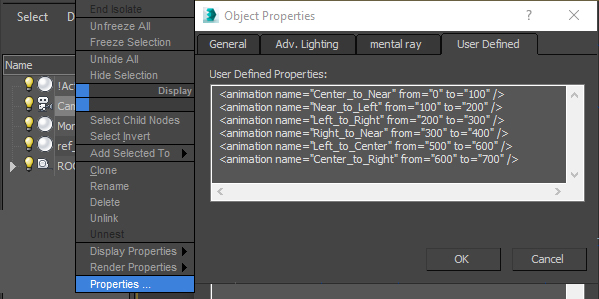
The description of the used segments of animated tracks, the corresponding transitions between the steits of each animated object is set in the user properties of the same object in the following format:

**<animation name="<StateName>\_to\_<StateName>" from="<number>" to="<number>" />**

Spaces inside quotation marks ("") are not allowed.

As a pair of <StateName>, we use the names for the states that were invented above, and as <number> we use the numbers of the first and last frames of the animation segments responsible for the animation of the transition between this pair of "States".

As a result, we get the following:



In this example, the lengths of the animation segments are the same and have common edges, but this is not necessary at all. Transitions can be of any length. If it is more convenient to create segments of transition animation at some distance from each other, do as it is more convenient. Sections of the track that are not described in the user properties of the object will be simply ignored.

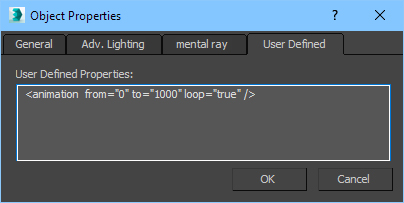
It is also seen from the example that it is not necessary to create and describe segments of animation corresponding to reverse transitions. Since AllCast can play tracks both in the forward and in the opposite direction.

If, after importing the scene into AllCast, "something went wrong", then first of all you need to look for the error in the description of transitions between the states in "Object Proporties / User Defined".

**5.2.3. Parameter "loop"**

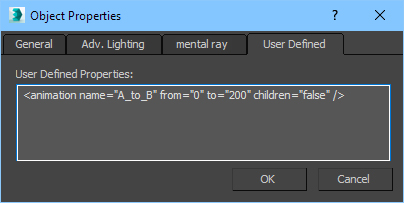
If the value of the loop parameter = "true", then the described segment of the animation of the object will be looped. At the same time, this animation starts automatically after the scene is loaded into AllCast. By default (in the absence of the loop parameter) its value is set to "false". Therefore, for non-looped animation segments, this parameter can be omitted. On the contrary, for looped segments, you can omit the names of the steits.

For example, if there is a rotating earth globe in a scene that makes a complete revolution in 1000 frames, then in order to create its looped animation, we need to make the following entry in the user properties of the globe object:



**5.2.4. Parameter "children"**

The "children" parameter is used when describing the animation of the parent objects of the hierarchical tree of 3d objects of the scene. If the value of children = "true", then both the animation of the parent object itself and the animation of its "children" will play. In this case, the animation of the "children" in their user properties is not necessary. By default (in the absence of the children parameter) its value set to "true".



**5.3. Animation with hysteresis.**

Sometimes it is required that the transition from state "A" to state "B" and back is carried out on different animation tracks and/or for a different number of frames. In this case, you should create for such an object in 3DS Max two corresponding segments of the animation of the transitions between states "A" and "B". And in the user properties of this object, it is necessary to describe both of these transitions. For example:

**<animation name="A\_to\_B" from="0" to="50" />**

**<animation name="B\_to\_A" from="60" to="75" />**

**5.4. "Auto State".**

What will happen if the animation of the object is created according to the rules described above, but there is no description of transitions in the camera’s “Object Proporties / User Defined”?

In this case, after loading the scene, AllCast will try to determine the “States” by itself, create a description of the transitions and the corresponding control icon buttons. For simple animation, quite often the automaton guesses "what the author meant." Especially if the segments of the tracks responsible for transitions between "States" have common edges. Otherwise, it is possible to automatically create unnecessary transitions, and in some cases, the loss of some " States".

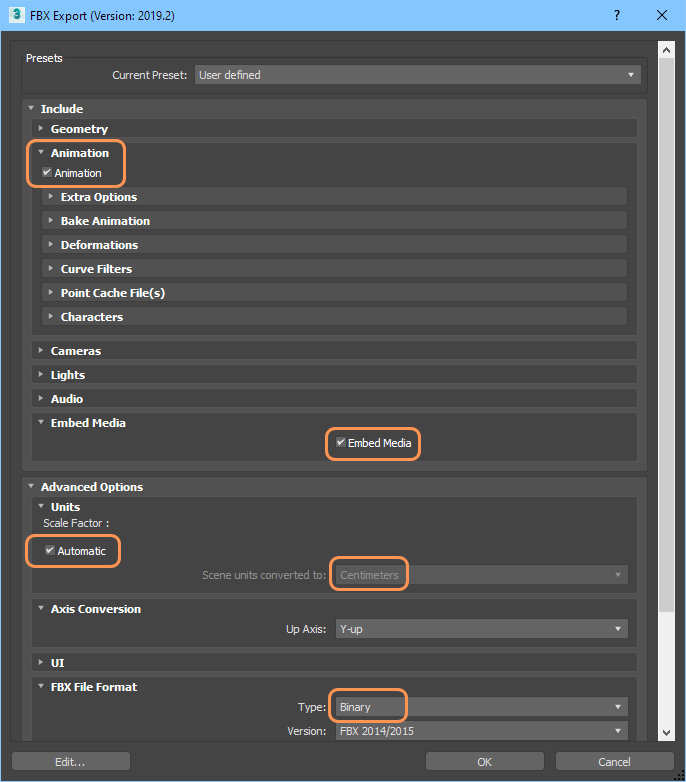
Something can be fixed in the built-in AllCast editor, but something is not.

Therefore, it is better not to be lazy and describe transitions between the " States" in "Object Proporties / User Defined".

**5.5. Some useful additional information.**

For the creator of the 3D scene, it is useful to know how AllCast uses the contents of the custom properties of the object:

* After loading the scene, the program automatically creates icon buttons in the interface that correspond to the "States" that have custom properties, designed according to the above rules.
* When you click on the icon of any "State", the program finds a line from the list in the user properties of this object, which contains a couple of "States" (current state and destination state) and plays a section of the track that is specified in this line in one direction or another.
* The images on the icon are generated as a "snapshot" from the virtual camera corresponding to the result of performing the action by clicking on this button-icon.
* In addition, the icon displays the name of the "State" to which this icon corresponds. That is why it is desirable that the name of the state corresponds to the purpose of the button.
* If only one transition is described in the properties of some "Object Proporties/User Defined" object, then when importing a file (containing this description) into All¹Mix FBX, only one control button-icon will be generated. When you click on this button, the animation segment will play in the specified direction, and when you click it again, it will play in the opposite direction.

**6. To use the 3D scene in AllCast, you must export the scene in FBX format:**