BigWorld MAXScripts and the BigWorld Maya Shelf

- Installation
- Export Visual and Animation
- Make Custom Hull
- Add Portal
- Label Portal
- Import Hard Point
- Add a Note Track
- Flag as BSP (collision object)
- Munge Normals
- Attach Hitbox and Export Hitbox to XML
- Object Divider
- Prefab Assembler
- Large Object Workflow

The BigWorld MAXScripts simplify some of the BigWorld specific tasks that are required during asset creation.

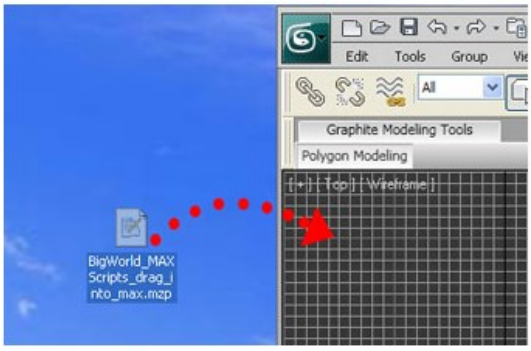
All these scripts modify your 3dsMax file, it is highly recommended that you save before using any of these scripts.

Installation

Maxscript Installation

The BigWorld Maxscripts have been tested with 3dsMax version 2010 and onwards. If using 3dsMax 2010 please ensure you have service pack 1 installed.

Drag and drop the BigWorld_MAXScripts.mzp located in `bigworld\tools\maxscripts\bigworld_maxscripts_v1.6.mzp` file onto the 3dsMax 3d view.



Restart 3dsMax.

Now your ready to use the toolbar.

Maya BigWorld Shelf Installation

To install the BigWorld Maya Shelf

1. Ensure the Maya exporters are installed. Information on how to install the exporters can be found in `bigworld\doc\content_tools_reference_guide.pdf`

2. Copy the BigWorld Shelf from `bigworld\tools\melscripts\src\shelf\shelf_BigWorld.mel` to your Maya shelf folder `Documents and Settings\username\My Documents\maya\2011\prefs\shelves`

3. Copy the script `bigworld\tools\melscripts\src\scripts\bw_common.py` to the Maya scripts folder `Documents and Settings\username\My Documents\maya\scripts\`

4. Copy the BigWorld Icons from `bigworld\tools\melscripts\src\icons` to your Maya Icon folder `Documents and Settings\adamm\My Documents\maya\2011\prefs\icons`

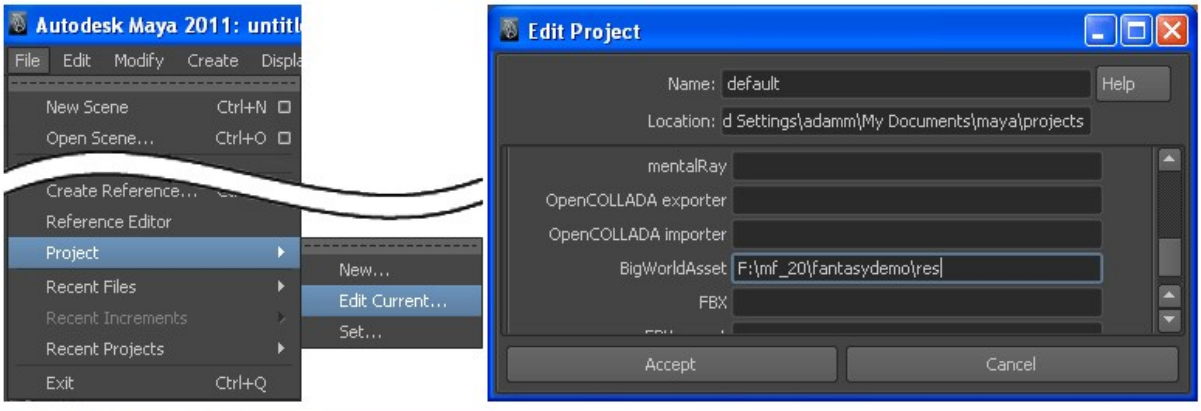
5. Copy the BigWorld helper maps from `bigworld\tools\melscripts\resources\textures` to the **textures** folder of your current Maya Project folder `Documents and Settings\username\My Documents\maya\projects\default\textures`

(AID_BSP.bmp, AID_exitportal.bmp, AID_hportal.bmp, AID_Hull.bmp, AID_portal.bmp)

6. Copy the contents of `bigworld\tools\melscripts\resources\assets` to the **assets** folder of your current Maya Project folder `Documents and Settings\adamm\My Documents\maya\projects\default\assets`

(maya_hard_point.mtl, maya_hard_point.obj, hardpoint_helper.bmp)

7. Add your game res/ path to the BigWorldAsset property of your current Project.



Export Visual and Animation

You can export an asset with using the export visual button  or animations using the export animation button  (3dsMax only).

If you have objects selected the script will do an **Export Selected.**

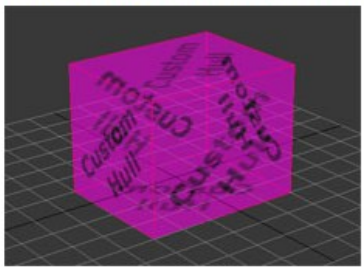
If you have nothing selected then the script will do an **Export All**.

Add Custom Hull

See the Content Creation manual bigworld/doc/content_creation.chm shells section for information on Custom Hulls

Clicking the **Add Hull** button  converts the selected object into a **Custom Hull**, by appending the suffix "_hull" to the objects name. A custom hull material is added to the object for visualisation purposes.

In 3dsMax, polygons of the hull can then be assigned portals using the scripts below.



Add Portal

See the Content Creation manual bigworld/doc/content_creation.chm shells section for information regarding Portals

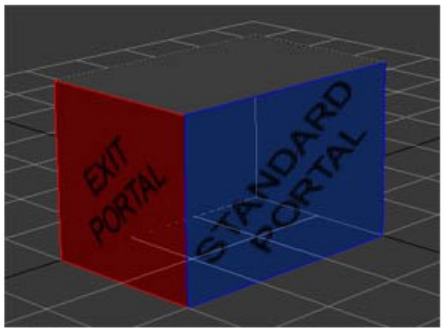
By clicking the Add Portal button the script will automatically assign a portal flag to the object selected. Subsequent clicks will remove the portal.

 Adds a standard portal flag

 Adds an Exit portal flag

 Adds a Heaven portal flag

The portal assignment will add a material to the portal to make it easily recognisable. The material library used is stored in the *BigWorld_MaterialLibrary.mat* located in *bigworld/tools/maxscripts/resources/materiallibraries/*



Portal assignment can be done in two ways using this script.

1. Whole objects (planes) can be assigned portal flags, in which case they will use the "portal = true" custom attribute.

2. Alternatively, if you are in **sub-object mode**, **Polygon** selection of a **Custom Hull**, then that polygon will be assigned a portal material.

Label Portal

This button adds a label to a selected portal.

Only geometry portals can be labeled.

 Adds a portal label

Import Hard Point

See the Content Creation manual bigworld/doc/content_creation.chm section Hard Points for information regarding Hard Points.

When multiple artists are creating assets with interdependent Hard Points (e.g. a gun and an avatars hand), discrepancies in the orientation of the Hard Points can occur. This script should reduce the number of misaligned Hard Points by standardising the Hard Point model and its alignment to the parent node.

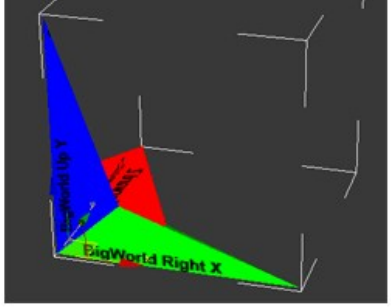
Pressing the Import Hard Point button  will automatically attach and align a standardised hard point model to the selected object.

If no object is selected the Hard Point model will be aligned to the world origin (0,0,0).

The Hard Point representation model will be automatically scaled so that it is ~20cm in scene units no matter your System Unit Settings. The model is a representation of the BigWorld coordinate system.

The hard point model *hard_point.fbx* is installed into the 3dsMax's project folder: *import* folder and uses materials from the *BigWorld_MaterialLibrary.mat*.

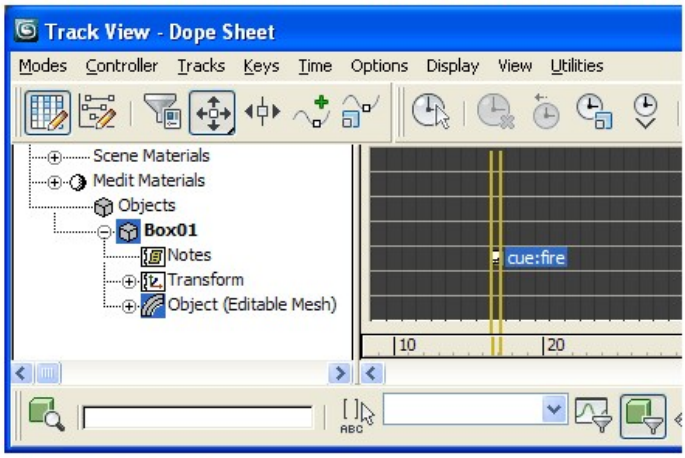
Note: The standardised **hard_point.fbx** model is located in **3dsMax's Project Path - import folder**. Changing 3dsMax's Project folder will cause this script to loose track of the hard point model. This can be fixed by re-installing the scripts, or moving the hard_point.fbx from the old /import folder to the new Project Path - import folder.



Add a Note Track

See the Content Creation manual bigworld/doc/content_creation.chm section Note Tracks for information regarding Note Tracks

Pressing the Add Note Track button  will allow the user to quickly add a note track of type "cue:" or "sound:" to the selected node at the slider Time.



Flag as BSP (collision geometry)

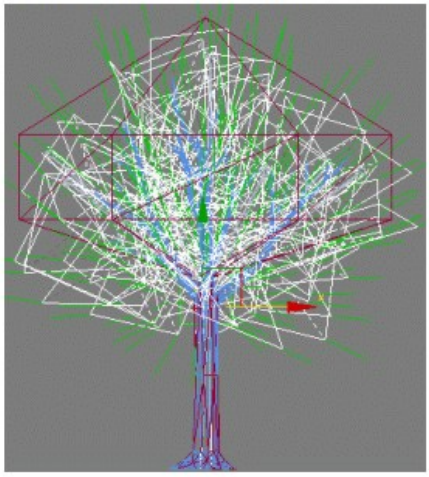
See the Content Creation manual bigworld/doc/content_creation.chm for information regarding BSP collision geometry

Pressing the Convert to BSP button  converts any geometric object to a BigWorld BSP collision object. The suffix "_bsp" is added to the objects name, and a material is applied so the object can be easily recognised within 3dsMax as a BSP object

Munge Normals

See the Content Creation manual bigworld/doc/content_creation.chm for information regarding the mungenormals script.

Pressing the Munge Normals button  will cause any normals of currently selected objects with an Edit Normals modifier to splay away from a subsequently selected object.



Attach Hitbox and Export HitBox to XML

By default BigWorld uses the entities bounding boxes for targeting. This works well for most cases, but some times more accurate targeting is needed. This script will generate a

By default BigWorld uses the entities bounding boxes for targeting. This works well for most cases, but some times more accurate targeting is needed. This script will generate a skeleton collider xml file that programmers can use for more accurate entity targeting.

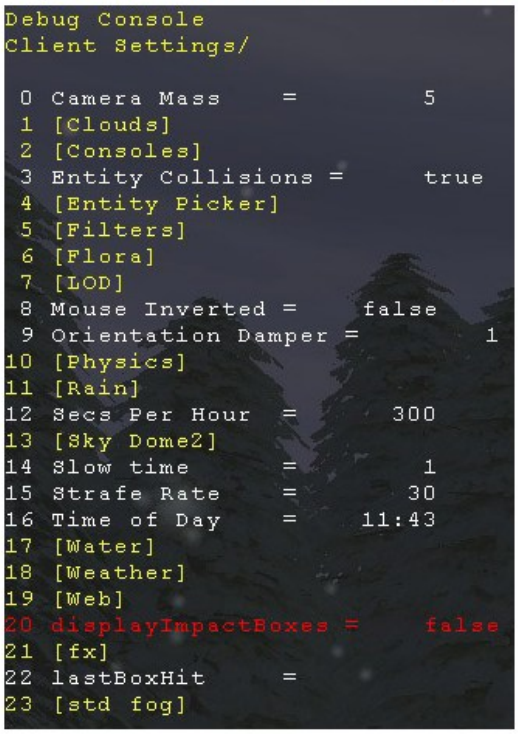
More detailed information on skeleton colliders can be found in `\bigworld\doc\howto_accurately_target_entities.pdf`

1. Select all the nodes onto which you would like to attach hitboxes. Try to be conservative, adding hitboxes to individual fingers is unnecessary and will result in poor game performance.

2. Press the Attach Hitbox button  . This will generate hitboxes and attach them to the selected nodes.

3. Adjust the size, and position of the hitboxes to best match your target model. Do not break the links between parent and hitbox. If you do, the hitbox will be ignored on export.

4. Export the hitboxes to a skeleton collider xml file by pressing the Export Hitbox to XML button  .



Once loaded (see details in `howto_accurately_target_entities.pdf`) the skeleton collider can be visualised by turning on the client watcher (~tilde + F7) **Client Settings - displayImpactBoxes = True**

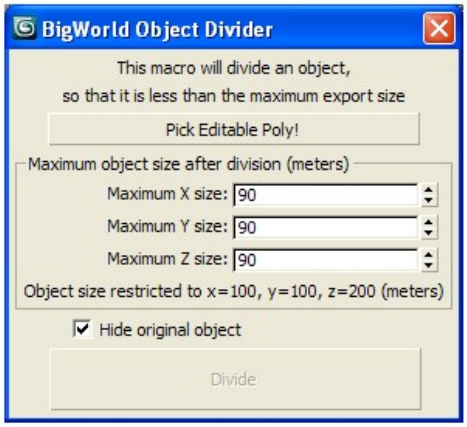
Object Divider

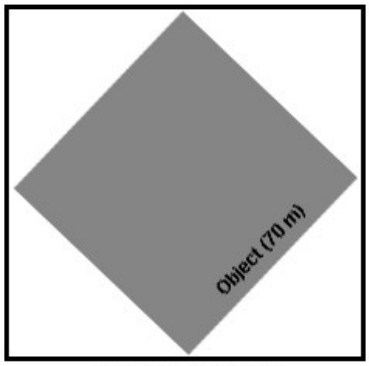
Objects imported into BigWorld have a maximum size restriction of 100m, 100m, 200m (XYZ). Sometimes it can be necessary to divide an object into several sections which can be exported individually.

The Object Divider script  will automatically divide an object up into sections smaller than the user defined Maximum size setting.

This script is usually used in conjunction with the Prefab Assembler script

Its a good idea to save your work before using this script as its going to chop up your model into little pieces.





Chunk (100m x 100m)

The Object divider will recommend a default setting of 70 meters, this is because if an object with a side length of 70 meters is rotated it will exceed the 100M limit if rotated 45 degrees. This concept is illustrated in the picture above.

Both the 3dsMax and Maya object dividers will place the divided object pieces on separate special named layers. This is to help facilitate the Prefab Assembler script.

Prefab Assembler

Due to object size restrictions it is often necessary to divide a very large object into multiple pieces and export each piece individually. These objects are manually reassembled within WorldEditor and a prefab is created for future placement.

To aid in manually reassembling the objects relative to each other, users will often export objects in-place from 3dsMax so that they all share a common pivot point. This can cause serious issues with chunk lending if the pivot point is greater than 100 meters from the objects bounding box and it makes object manipulation extremely difficult, this is NOT a recommended technique.

The correct technique is to move each object to the origin and then export so that the objects pivot point is in close proximity to the objects bounding box. This creates the problem that the user must now reassemble the assets in WorldEditor relative to each other, either by eye, or by first recording the transformation done prior to export.

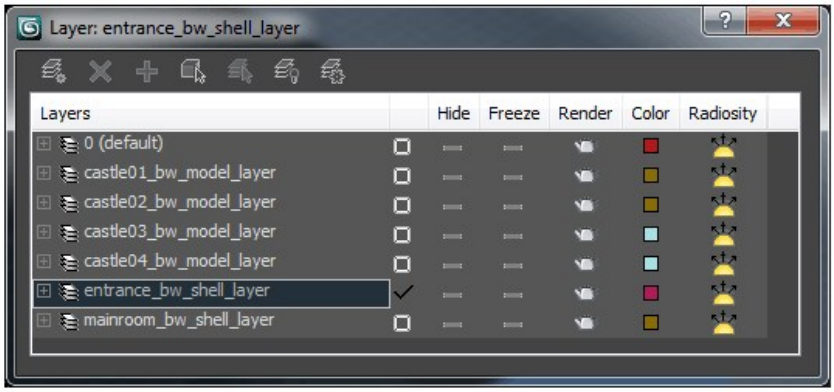
To prevent such problems the Prefab Assembler macro script  allows you to create a prefab file from multiple objects in 3dsMax.

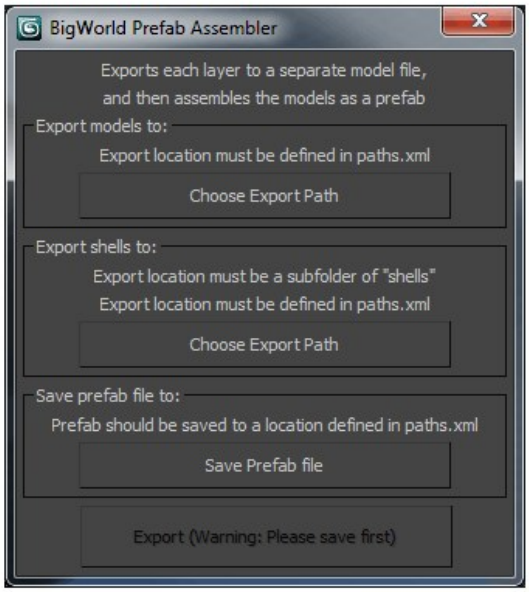
The script does the following.

1. Scans the scene for layers with the suffix "**_bw_shell_layer**" or "**_bw_model_layer**" and considers all objects on each layer a single .model
2. Records the position of each object in each Layer
3. Moves each object, temporarily to the world origin (0,0,0) to ensure that the objects pivot point resides within the objects bounding box.
4. Exports each layer generating a .model, .visual and .primitive file.
5. Generates a `.prefab` file of all the above models

The option to "Export Shells" allows users to export a scene containing multiple shells. Shells are made up of multiple components (portals, custom hulls, shell geometry).

Users must manually place each shell and all its parts on separate layers with the suffix "**_bw_shell_layer**" so that the Prefab Assembler will correctly export them.





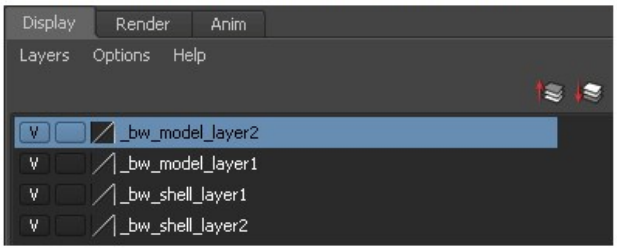
Note: Save your work file before using this script.

Maya Prefab Assembler

The Maya version of the prefab assembler also requires specifically named layers.

Model objects should be placed on layers with the suffix or prefix "_bw_model_layer" (Note the object divider script does this automatically to objects it cuts up)

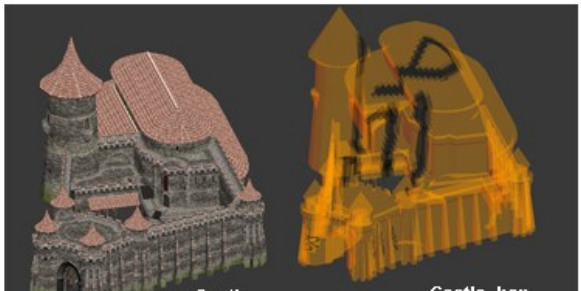
Shell objects should be placed on layers with the suffix or prefix "_bw_shell_layer"



Large object workflow

This tutorial will take you through step by step how to create a prefab out of a very large object using the object divider and prefab assembler max/mel scripts.

1. Always save your work to a separate backup file first.
2. Firstly I have created a very large (greater than 100m) single piece model in 3dsMax of a castle and a second lower resolution model of its BSP



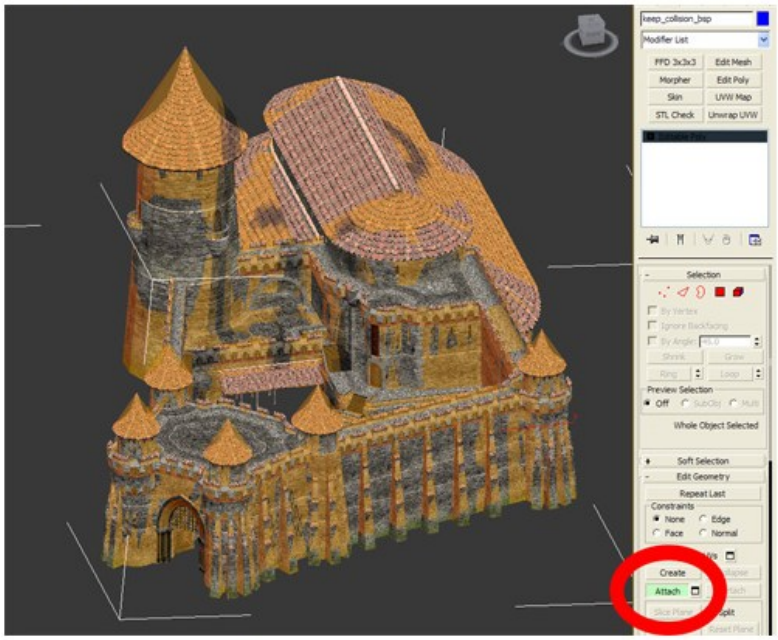
3. The Models are placed over the top of each other and collapsed into a single model using the *Attach* command in the 3dsMax modifier panel.

This will make the two models into a single piece model. We need to do this so that the object divider script can divide both objects up equally along the same lines.

If the model's LODs are already made, they too would be attached to the primary model to make a single object for the object divider. Alternatively the LOD models can be created after the primary model is exported and the prefab created.

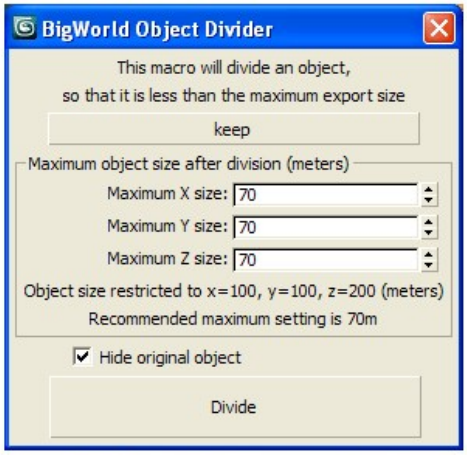
If you model contains neither custom BSP or LOD's then this, and step 6 are unecessary.

The separate BSP and LOD models must be given unique material names or mesh ID's, to enable the user to separate the pieces into individual models once the object divider had done its job.



4. Run the object divider script. A dialog will prompt you for an Editable Poly object and for maximum size settings.

If you plan to rotate your object in WorldEditor then we recommend that you do not exceed 70 meters in any direction. A 70 meter square cube is 100 meters from corner to corner.



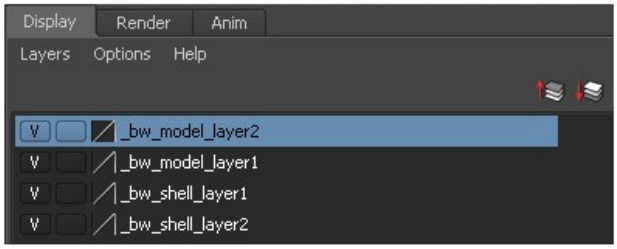
5. The Object Divider script will put the divided objects on individual layers with specific names.

Each layer is considered a single BigWorld model by the Prefab assembler script, so if you have multiple mesh on a single layer they will be exported as a single model.

Only layers with specific names will be exported by the prefab assembler.

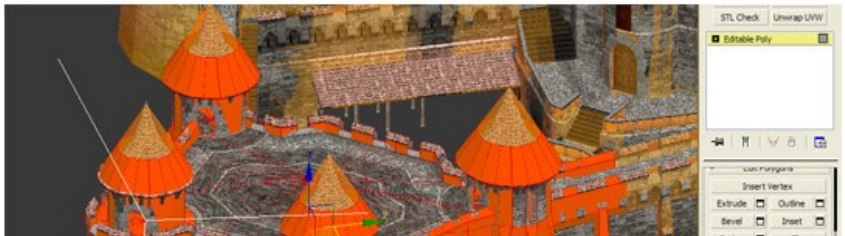
Layers containing the "**_bw_model_layer**" are considered models by the prefab assembler.

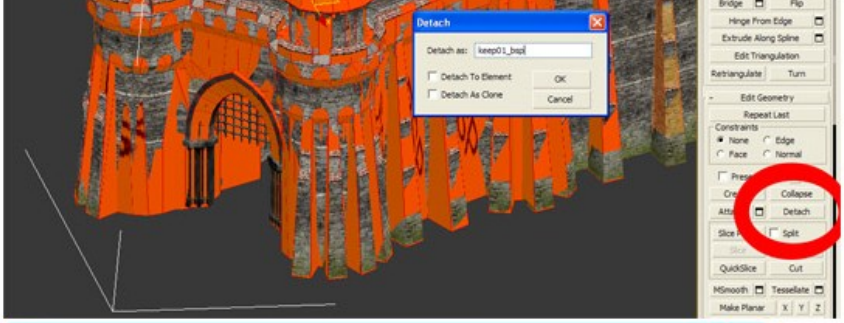
Layers containing the "**_bw_shell_layer**" are considered shells by the prefab assembler. The placement of individual shells on separate layers must be done manually by the user.

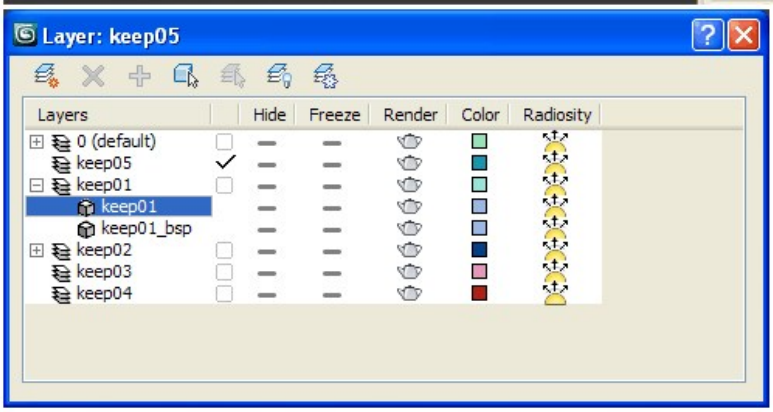


The layer name in conjunction with the model export name will be used as the BigWorld model name when exported by the prefab assembler

6. Following object division you will then have to separate the BSP and LOD sections from your now divided model using the *Detach* command in the 3dsMax modifier panel. Ensure that each detached BSP model is appropriately named with the "_bsp" suffix and is located on the appropriate layer.



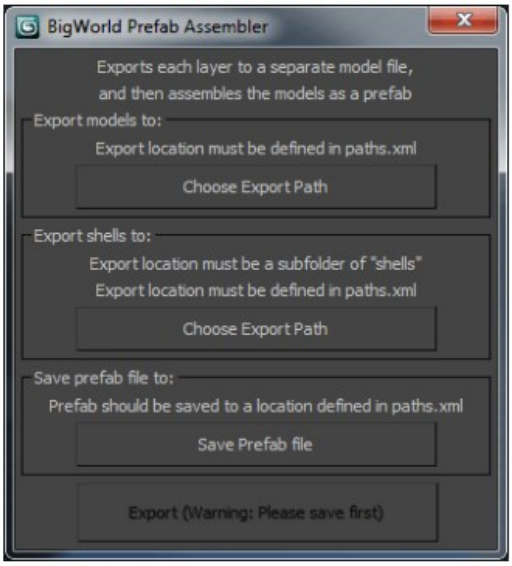




7. You are now ready to export the models using the Prefab Assembler macro script 

Choose an export location within your resource path.

Choose where to save your prefab file within your resource path.



8. You are now ready to place your prefab file in the WorldEditor