

# Riley's Mesh Reconstruction Tools

This is the user's guide for Version 0.0.22a of the Mesh Reconstruction Tools.

MSH files contain information on what props, costumes, sets, and facilities look like in the game. These utilities can read any type of MSH file, though they are primarily used to create static props.

## Installation Instructions

These tools are all command-line utilities. They are best installed in their own folder, separate from The Movies game. A good place might be "C:\Program Files\r3tm\".

1. In Windows Explorer, go to "C:\Program Files". Create a new folder named "r3tm" (or any other name of your choosing).
2. Copy all the files contained in the `r3-tm-mesh-recon-v0022a.zip` archive into the new folder.

To run these utilities, follow the steps below.

1. Go to Programs → Accessories → Command Prompt
2. Enter `cd\`
3. Enter `cd "Program Files"`
4. Enter `cd "r3tm"`
5. Run the appropriate command-line utilities, as instructed in each section below. Entering just the base name of each utility will output help information. *All commands should be typed in lower case.*

Each utility has its own user's guide in the pages below. The appendices at the end of this document provide useful quick-reference information.

## Location of Input Files

By default, most of these utilities look for the files you specify in the following locations:

1. Folders you specify with the `-newdata` option.
2. The `\content\` folder located in the directory you installed the utilities. This is where you can place custom content that you don't want to install in your game.
3. The directory you installed The Movies game.
4. The PAK files that come with The Movies game. These are compressed files located in the game's `\data\pak\` folder.

## Location of Output Files

By default, most of these utilities output your props in a folder that is given the same name as the prop. It is highly recommended that you use the `-workdata` option to specify the output folder. Doing so will allow you to better organize your props, usually by using the same folder for all props within the same project.

These utilities also automatically create a `\cache\` folder. This is used to store files contained in the game's PAK files, for faster access. The files found in this folder are safe to delete at any time, though they will be recreated if you reference those files again in a future command.

## Colour-Coded Examples

Throughout this document, you will find example commands you can enter on the command-line. All examples contain colour-coding, with the following meanings:

1. Text in **bold** is the name of the application being run. This is always the first thing on the command-line.
2. Text in **dark red** is a required input field. This is typically the filename of the input file, and / or the prop name that you want to output.
3. Text in **blue** is an input option. These usually begin with a dash, and most (but not all) require it to be followed with an option value. Each input option, combined with its option value, tells the application to perform some specific function.
4. Text in **green** is an option value. These always follow input options.

# r3-tm-meshreport

This utility outputs basic reports of the data stored in a MSH file. It also extracts individual meshes or object, each into its own separate MSH file.

This utility is primarily used in conjunction with r3-tm-extractprop, to extract specific props from existing sets or facilities. To make full use of this utility, you will first have to understand a few concepts about MSH files.

Concept	Description
Mesh Group	MSH files organize their data into <i>groups</i> . In most of the game's sets, this is used to keep "interactive" portions of the set separate from the rest of the set. This includes objects such as doors, which actors can interact with.
Group Number	Each <i>group</i> is given its own <i>group number</i> . These are numbered consecutively, beginning at zero.
Mesh	Each <i>group</i> is made up of multiple <i>meshes</i> . Each mesh is kept separate for one main reason: the texture it uses.
Mesh Number	Each <i>mesh</i> within a <i>group</i> is given its own <i>mesh number</i> . These are also numbered consecutively, beginning at zero.
3D Objects	Each <i>mesh</i> may contain one or more <i>3D objects</i> . This application is capable of isolating these 3D objects under most circumstances, with the exception of flora (trees and bushes).
Vertex Number (aka Object Number)	Each <i>3D object</i> in a <i>mesh</i> begins at a specific <i>vertex number</i> . This application provides information on which vertex number each 3D object begins at.

The most important output from this utility is the group numbers, mesh numbers, and vertex numbers. Those numbers are used as the input for r3-tm-extractprop.

## Basic Usage

Enter the following command to run this utility:

```
r3-tm-meshreport yourmesh.msh
```

Most often, you'll want to run this utility against one of the game's existing sets or facilities. Appendix A and B list all of the game's sets and facilities, and their associated MSH filename. This utility looks for the MSH file in the \data\meshes\ folder.

Quite often, you'll also want to produce individual MSH files for each individual mesh and 3D object. Add the "-xo" option (listed in the Advanced Options section below) to do this, as follows:

```
r3-tm-meshreport -xo yourmesh.msh
```

## Text Report

The default output of this utility is a text report listing the *mesh groups* and *meshes* contained in the MSH file. Much of the information listed was originally used to analyze the MSH file format.

You'll be most interested in the following information:

- The number and name of each *mesh group*.
- The number of *meshes* contained in each *mesh group*.
- The texture name used by each *mesh*.
- The *vertex numbers* listed for each *mesh* at the end of the text report.
- The 3D co-ordinate range for the entire mesh file, for each *mesh group*, and for each *mesh*.

## Output MSH Files

If you use the "-xm" or "-xo" options, this utility will also output individual MSH files for each *mesh* and *3D object*. These files will be placed in a folder with the same base name as the input MSH file.

Each individual mesh will be given the filename `group##mesh##.msh`.

Each individual 3D object will be given the filename `group##mesh##vertex####.msh`.

## Advanced Options

The following options affect the output of this utility:

Option	Parameters	Description / Usage
-newdata	<full-folder-path>	Specifies an additional directory to search for game data. This is useful if you have props that you don't want included in your copy of the game, but still want available for this utility.
-r	<report-filename>	Specifies the filename of the text report to be output. By default, this is "r3-tm-meshreport.txt".

-xm		Indicates that each individual <i>mesh</i> should be exported as its own MSH file.
-xo		Indicates that each individual <i>mesh</i> <b>and</b> <i>3D object</i> should be exported as its own MSH file.

## r3-tm-extractprop

This utility is used to extract individual objects or props from existing sets or facility buildings. It can also be used to clear out existing sets to make emptied versions of them.

This is primarily used in conjunction with r3-tm-meshreport, to determine which *group numbers*, *mesh numbers*, and *vertex numbers* represent the portion of the set or facility that you want to extract.

### Basic Usage

This utility has many options, and you'll want to read over the Full List of Options section thoroughly to make full use of its capabilities. The basic syntax of the commands you'll want to run look as follows:

```
r3-tm-extractprop -workdata <work-folder-name>  
set_myset.msh p_myprop <list-of-options>
```

<work-folder-name> is the name of the folder that the prop will be placed in. This part of the command is optional, but highly recommended. By default, this utility will place its extracted props in a folder with the same name as the output prop.

set\_myset.msh is the filename of the MSH file from which you want to extract a prop.

p\_myprop is the base name of the prop you want to create.

<list-of-options> is what will tell the utility what part of the set or facility you want to extract. This is where you will need to look over the output from r3-tm-meshreport, and determine the *group numbers*, *mesh numbers*, and *vertex numbers* you need.

To install your props into the game, copy the files that are generated from the <work-folder-name> into your game's directory. Make sure you place each file in the appropriate sub-folder.

The sections below will cover a number of scenarios. Follow along with each example, and you should begin to understand how to make full use of this utility. The examples use set\_hotelbedroom.msh (the Plush Hotel Bedroom), and each prop exported begins with "p\_ex\_" to indicate they are example props. All examples will also place the props into a folder named \examples\.

### All Meshes in a Group

To add all meshes in a specific *mesh group* to the resulting prop, use the following option.

```
-g <group-number>
```

For example:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_armchair -g 4
```

## Specific Meshes in a Group

To add one or more *meshes* from a *mesh group* to the resulting prop, use the following option.

```
-g <group-number> -m <list-of-mesh-numbers>
```

For example:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh p_ex_bed  
-g 8 -m 22 23
```

In the above example, all meshes except 22 and 23 are excluded from the resulting prop.

## Specific 3D Objects in a Mesh

To add one or more 3D objects from a mesh to the resulting prop, use the following option.

```
-g <group-number> -m <mesh-number> -v <list-of-vertex-numbers>
```

For example:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_smalldresser -g 8 -m 36 -v 51 55 59 67 83 247 263 480
```

If you want to automatically center the 3D objects for your prop, use “-cv” instead of “-v”. For example:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_smalldresser -g 8 -m 36 -cv 51 55 59 67 83 247 263 480
```

## Category in Advanced Movie Maker

To change the category that your prop will appear under in Advanced Movie Maker, use the following option. The default category is 9 (miscellaneous).

```
-cat <category-number>
```

The categories are numbered from 0 through 9, as shown in Appendix D.

For example:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_armchair -cat 5 -g 4
```

## Retaining Bone Data

This option is new as of Version 0.0.21a. Bone data is used by many mesh files to allow a prop, set, or facility to be animated in some way. Normally this utility ignores bone data. If you want to retain bone data, you can do so by including the following option.

`-bones`

For example:

```
r3-tm-extractprop -workdata examples fac_postproduction.msh
p_ex_satdish -bones -g 2 -m 22 -v 12 96 -m 23 -clm -3 7 10 0 3 14 -m
38 -v 0 -cp -2.3 5.158 10.323
```

When you include this option, the root bone will be transformed by all prop-level transformation options. The exact effect this has depends on the animation you plan to use. The above example uses the `-cp` option to translate both the prop and the root bone so that the rotating satellite dish is at ground-level.

Note that using this option does not automatically turn on the “AutoAnimated” flag (which can be done in MeshManip or a hex editor), nor does it attach an animation to the prop. To specify an animation for the prop, you must identify the animation file (\*.anm) you want to use (which you can extract using Med or Reshoot); copy the file to the `\data\animations\high\autoanimated\` folder; and rename the file to `aa_p_<propname>_v00.msh`. Additional random animations can be added by incrementally increasing the `v00` number.

If the root bone was transformed in some way, you may need to make changes to the animation file. Specifically, you may need to change the `u`, `v`, `w` values for the root bone, to match the last row of values in the root bone’s pivot matrix. This can be done with a hex editor, using Gleem’s animation file template.

## Replacing or Deactivating a Texture

By default, the props you extract will refer to the same textures as the original MSH file. In some cases, you may want to replace a texture with something you have custom-made. Use the following option to replace all instances of a given texture with a custom texture.

`-rtp <texture-file> <new-texture-file>`

For example:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh
p_ex_newbedroom -rtp marble_rk01.dds floor_rk02.dds -g 2 3 8 10
```

You may also want to deactivate certain textures, such as light-maps. Light-maps are what allow sets to be displayed in various lighting conditions. When you extract props from a set, you will likely want to remove the shadows cast upon them. Use the following option to deactivate all instances of a given texture.

`-rtp <texture-file> null`



For example:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh
p_ex_newbedroom -rtp lm_hotelbedroom_0.dds null -g 2 3 8 10
```

Use the text report generated by r3-tm-meshreport to determine the filename of the textures you want to replace or deactivate. Light-map textures are typically given filenames that look like “lm\_<set-name>\_0.dds”.

## Setting a Texture for a Mesh

To set the texture used by a specific mesh, use the following option.

```
-t0 <texture-file>
```

That’s a zero, not an O. The above option sets the base texture, also known as the “diffuse texture”. While you will not need to use them as often, you can also use the -t1, -t2, and -t3 options. t1 = Reflective texture. t2 = Light-map texture. t3 = Specular texture.

Similar to the -rtp option above, you can remove a texture from a mesh by setting it to null.

This option must follow a mesh number. For example, the following command will change the textures for mesh 2 in group 8:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh
p_ex_bedroomwalls -g 8 -m 2 -t0 wall_rk08.dds -t2 null
```

These options give you much more fine control over the textures used than the above -rtp option. All types of textures should be designed to fit the same shapes and co-ordinates as the base (diffuse) texture. The only exception is when you replace light-maps for meshes that already have a light-map attached: in this case, any custom light-map texture should follow the same shape and co-ordinates as the original light-map.

## Activating or Deactivating a Texture’s Alpha Channel

An “alpha channel” is what allows a texture to have see-through properties (or transparency). You can turn the alpha channel for a given *mesh* on or off with the following options.

```
-aon
-aoff
```

This option must follow a mesh number. For example, the following command will switch off the alpha channel for mesh 1 in group 8 (which happens to be the floor):

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh
p_ex_newbedroom -g 2 3 10 -g 8 -m 0 2 3 4 5 6 7 8 10 11 12 13 14 15
16 17 20 21 22 23 24 26 27 28 30 33 34 35 36 39 40 41 45 46 -m 1
-aoff
```

## Translating Texture Co-ordinates

This is a very rarely used option, but is useful in specific circumstances. If you want to create multiple copies of some "generic prop", and want all of these props to use the same base texture, you may find this useful. It may also be useful in aligning a repeating texture properly against an adjacent surface. Texture co-ordinates are measured on a scale from 0 to 1, no matter the dimensions of the image file.

To translate the texture co-ordinates for a specific mesh's base texture, use the following option:

```
-tt <x> <y>
```

For example, the following command will cause the texture of the bedroom walls to be shifted halfway down:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_bedroomwalls -g 8 -m 2 -tt 0 0.5
```

## Animating Textures

You can make your textures "animate" by setting the speed of movement along the texture's X and Y co-ordinates, as well as the speed of rotation. Making use of this capability requires that you carefully set the texture co-ordinates for each vertex in your mesh.

To animate the texture for a specific mesh's base texture, use the following option:

```
-tanim <x-speed> <y-speed> <rot-speed>
```

Each of the speed values should be an integer between 0 and 255. A value of 0 indicates that the texture should not animate. Higher values cause the scrolling animation to go faster. Texture rotation can cause strange effects if your texture co-ordinates are too close to the edges.

For example, the following command will cause the texture of the bedroom walls to scroll horizontally:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_bedroomwalls -g 8 -m 2 -tanim 50 0 0
```

## Reflective Floors

You can indicate that an individual *mesh* should act as a reflective floor. This can be done with the following options:

```
-rfon  
-rfoff
```

For example, the following command will turn off the reflective floor in the bedroom:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh
p_ex_newbedroom -g 2 3 10 -g 8 -m 0 2 3 4 5 6 7 8 10 11 12 13 14 15
16 17 20 21 22 23 24 26 27 28 30 33 34 35 36 39 40 41 45 46 -m 1
-rfoff
```

## Centering your Prop

Most props you extract can be centered properly by using the `-cv` option mentioned earlier. When you use that option, it programmatically determines the center point of all the 3D objects listed, and centers the entire prop accordingly.

To specify the exact 3D co-ordinates to center the entire prop, use the following option. This option is typically found at the end of the command.

```
-cp <x> <y> <z>
```

You can specify the exact 3D co-ordinates to center a *mesh group* around by using the following option.

```
-cg <x> <y> <z>
```

You can also set the center point for an individual *mesh* by using the following option. This is used only in rare circumstances, usually to correct situations where meshes “collide” when converted from a set to a static prop.

```
-cm <x> <y> <z>
```

For example, the following command raises the two armchairs off the floor, and pushes the dresser a few meters from its original location:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh
p_ex_newbedroom -g 2 3 -g 4 -cg 0 0 -1 -g 5 -cg 0 0 -1 -g 8 -m 0 1 2
3 4 5 6 7 8 10 11 12 13 14 15 16 17 20 21 22 23 24 26 27 28 33 34 35
36 39 40 41 45 46 -m 30 -cm 1 1 0
```

## Rotating your Prop

While not necessary, you may want to rotate your prop so that it's facing forward when being placed in Advanced Movie Maker. The angle you specify should be measured in degrees, and a positive angle will rotate counter-clockwise.

You can rotate the entire prop with the following option.

```
-rpz <angle>
```

To rotate only a specific *mesh group*, use the following option.

```
-rgz <angle>
```

You can also rotate only a specific *mesh*, using the following option.

```
-rmz <angle>
```

Typically, you will have to extract your prop without the rotation option first. Then open the mesh file in Blender to find out the rotation angle. The angle in Blender will be of the opposite sign.

For example, to extract the armchair and rotate it so it's facing forward, you would end up with the following command.

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_armchair -g 5 -m 0 -cv 0 32 44 113 196 213 -rpz -79.75
```

The letter "z" in the above options indicates that the rotation will occur around the Z-axis. This means that the rotation occurs the same way as it does in Advanced Movie Maker. If you want to "tip over" a prop (or a *mesh group* or individual *mesh*), you can also rotate around the X- or Y-axes. Replace the "z" with an "x" or "y" to do so.

For example, to extract the armchair and tip it onto its back, you would end up with the following command.

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_armchair -g 5 -m 0 -cv 0 32 44 113 196 213 -rpz -79.75 -rpx -90
```

It is a good idea to place your rotation options at the end of the appropriate section. This ensures the rotation occurs *after* any centering options.

## Scaling your Prop

If you want to make your prop larger or smaller, use the following option:

```
-sp <factor>
```

The factor value multiplies the size of the prop in every direction. Using a value of 0.5 will make the prop half the size, while using 2 will double its size.

You can also scale only a specific *mesh group*, using the following option:

```
-sg <factor>
```

Or use the following option to scale a specific *mesh*:

```
-sm <factor>
```

For example, to make a bedroom fit for giants, you might use the following command:

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_bigbedroom -g 2 3 8 10 -sp 5
```

As of version 0.0.20a, you can also stretch a prop, mesh group, or mesh by indicating the axis to scale on. These options are -spx, -spy, and -spz for scaling the entire prop; -sgx, -sgy, and -sgz for scaling a mesh group; and -smx, -smy, and -smz for scaling a mesh. Scaling on the X-axis will make the prop wider, on the Y-axis will make the prop deeper, and on the Z-axis will make the prop taller.

It is a good idea to place your scaling options at the end of the appropriate section. This ensures scaling occurs *after* any centering options.

## Manual Clipping Regions

The -v and -cv options actually work by automatically determining the appropriate clipping region to use to isolate specific *3D objects*. In some cases, you may find it is too arduous to determine all the *vertex numbers* for certain *3D objects*. In such cases, you will want to consider applying manual clipping regions. To gather 3D co-ordinates, you can open the input mesh file in Blender and use the View Properties box.

To add a new clipping region to a specific *mesh*, use the following option:

```
-clm <x1> <y1> <z1> <x2> <y2> <z2>
```

If you want to apply a clipping region to an entire *mesh group*, use the following option:

```
-clg <x1> <y1> <z1> <x2> <y2> <z2>
```

Or to apply a clipping region to the entire prop, use the following option:

```
-clp <x1> <y1> <z1> <x2> <y2> <z2>
```

For example, to extract only the wall containing the window, you would end up with the following command. The clipping region specified with the -clm option below will include all geometry that is between X = -7 and X = -0.3, Y = 4 and Y = -0.3, and Z = -1 and Z = 5.

```
r3-tm-extractprop -workdata examples set_hotelbedroom.msh  
p_ex_windowwall -g 8 -m 2 -clm -7 4 -1 -0.3 -1.5 5 -m 11 -v 27 51 -m  
39 -g 10 -cp -3.1 1.4 0
```

Clipping regions are additive, meaning you can apply as many of them as you want. Objects contained in *any one* of the clipping regions you apply will appear in the final prop.

## Full List of Options

The table below lists every input option available.

Option	Input Values	Description
<b>Global Options</b> Place these at the beginning of your command, before the input mesh filename and output prop name.		
-workdata	<work-data-folder>	Specifies the output folder that the prop will be placed in.
-newdata	<full-folder-path>	Specified an additional location to search for game data files.
<b>Prop-Level Options</b> Place these anywhere after the input mesh filename and output prop name. Note that all "transformation" options (center, scale, rotation) occur in the order specified in the command.		
-cat	<category-number>	Specified the category number the prop will be placed under in AMM.
-bones		Retains all bone data from the original mesh file. Transformations made against the prop will also be made to the root bone.
-rtp	<old-dds> <new-dds>	Replaces all instances of the old texture filename with the new texture filename.
-cp	<x> <y> <z>	Centers the entire prop around the given 3D co-ordinates.
-sp	<factor>	Scales the entire prop by the given factor.
-spx	<factor>	Scales the entire prop, in the X-direction only, by the given factor.
-spy	<factor>	Scales the entire prop, in the Y-direction only, by the given factor.
-spz	<factor>	Scales the entire prop, in the Z-direction only, by the given factor.
-rpx	<angle>	Rotates the entire prop around the X-axis by the given angle.
-rpy	<angle>	Rotates the entire prop around the Y-axis by the given angle.
-rpz	<angle>	Rotates the entire prop around the Z-axis by the given angle.
-clp	<x1> <y1> <z1> <x2> <y2> <z2>	Applies a clipping region to the entire prop, bounded by the given 3D co-ordinates.
-g	<group-number>	Indicates that the given <i>mesh group</i> should be included in the final prop.

Option	Input Values	Description
<b>Group-Level Options</b> Place these immediately after a new -g <group-number> option. Note that all "transformation" options (center, scale, rotation) occur in the order specified in the command.		
-n	<group-name>	Specifies the name of this <i>mesh group</i> .
-rtg	<old-dds> <new-dds>	Replaces all instances of the old texture filename with the new texture filename, in this <i>mesh group</i> only.
-cg	<x> <y> <z>	Centers this <i>mesh group</i> around the given 3D co-ordinates.
-sg	<factor>	Scales this <i>mesh group</i> by the given factor.
-sgx	<factor>	Scales this <i>mesh group</i> , in the X-direction only, by the given factor.
-sgy	<factor>	Scales this <i>mesh group</i> , in the Y-direction only, by the given factor.
-sgz	<factor>	Scales this <i>mesh group</i> , in the Z-direction only, by the given factor.
-rgx	<angle>	Rotates this <i>mesh group</i> around the X-axis by the given angle.
-rgy	<angle>	Rotates this <i>mesh group</i> around the Y-axis by the given angle.
-rgz	<angle>	Rotates this <i>mesh group</i> around the Z-axis by the given angle.
-clg	<x1> <y1> <z1> <x2> <y2> <z2>	Applies a clipping region to this <i>mesh group</i> , bounded by the given 3D co-ordinates.
-m	<mesh-number>	Indicates that the given <i>mesh</i> should be included in the final prop.
<b>Mesh-Level Options</b> Place these immediately after a new -m <mesh-number> option. Note that all "transformation" options (center, scale, rotation) occur in the order specified in the command.		
-t0	<image-dds>	Sets the base texture filename to use for this <i>mesh</i> .
-t1	<image-dds>	Sets the reflective texture filename to use for this <i>mesh</i> .
-t2	<image-dds>	Sets the light-map texture filename to use for this <i>mesh</i> .
-t3	<image-dds>	Sets the specular texture filename to use for this <i>mesh</i> .

Option	Input Values	Description
-aon		Activates the base texture's alpha channel for this <i>mesh</i> (meaning it may become transparent).
-aoff		Deactivates the base texture's alpha channel for this <i>mesh</i> (meaning it will no longer be transparent).
-rfon		Activates a reflection for this <i>mesh</i> (intended for use with floors).
-rfoff		Deactivates the reflection for this <i>mesh</i> .
-tt	<x> <y>	Translates the base texture co-ordinates for this <i>mesh</i> .
-tanim	<x> <y> <rot>	Animates the base texture for this mesh, using the given speed values. Enter a value from 0 (no animation) to 255 (very fast animation).
-v	<vertex-number>	Indicates that the given 3D object should be included in the final prop.
-cv	<vertex-number>	Indicates that the given 3D object should be included in the final prop. The entire prop will also be centered around the mid-point of this 3D object.
-cm	<x> <y> <z>	Centers this <i>mesh</i> around the given 3D co-ordinates.
-sm	<factor>	Scales this <i>mesh</i> by the given factor.
-smx	<factor>	Scales this <i>mesh</i> , in the X-direction only, by the given factor.
-smy	<factor>	Scales this <i>mesh</i> , in the Y-direction only, by the given factor.
-smz	<factor>	Scales this <i>mesh</i> , in the Z-direction only, by the given factor.
-rmx	<angle>	Rotates this <i>mesh</i> around the X-axis by the given angle.
-rmy	<angle>	Rotates this <i>mesh</i> around the Y-axis by the given angle.
-rmz	<angle>	Rotates this <i>mesh</i> around the Z-axis by the given angle.
-clm	<x1> <y1> <z1> <x2> <y2> <z2>	Applies a clipping region to this <i>mesh</i> , bounded by the given 3D co-ordinates.



# r3-tm-transformprop

This utility performs basic transformations against a mesh file, and outputs the results as a static prop. It is something of a simplified version of r3-tm-extractprop.

## Basic Usage

This utility has a variety of options, so you'll want to read over the Full List of Options section to make full use of its capabilities. The basic syntax of the commands you'll want to run look as follows:

```
r3-tm-transformprop -workdata <work-folder-name>  
set_myset.msh p_myprop <list-of-options>
```

<work-folder-name> is the name of the folder that the prop will be placed in. This part of the command is optional, but highly recommended. By default, this utility will place its extracted props in a folder with the same name as the output prop.

set\_myset.msh is the filename of the MSH file from which you want to extract a prop.

p\_myprop is the base name of the prop you want to create.

<list-of-options> contains the list of transformations that should be applied to the prop.

## Full List of Options

The table below lists every input option available.

Option	Input Values	Description
<b>Global Options</b> Place these at the beginning of your command, before the input mesh filename and output prop name.		
-workdata	<work-data-folder>	Specifies the output folder that the prop will be placed in.
-newdata	<full-folder-path>	Specified an additional location to search for game data files.

Option	Input Values	Description
<b>Prop-Level Options</b> Place these anywhere after the input mesh filename and output prop name. Note that all "transformation" options (center, scale, rotation) occur in the order specified in the command.		
-cat	<category-number>	Specified the category number the prop will be placed under in AMM.
-bones		Retains all bone data from the original mesh file. Transformations made against the prop will also be made to the root bone.
-rt	<old-dds> <new-dds>	Replaces all instances of the old texture filename with the new texture filename.
-lm	<lightmap-dds>	Sets the light-map texture for all meshes that do not already have a light-map.
-lma	<lightmap-dds>	Sets the light-map texture for all meshes, even those that already have a light-map.
-c	<x> <y> <z>	Centers the entire prop around the given 3D co-ordinates.
-s	<factor>	Scales the entire prop by the given factor.
-sx	<factor>	Scales the entire prop, in the X-direction only, by the given factor.
-sy	<factor>	Scales the entire prop, in the Y-direction only, by the given factor.
-sz	<factor>	Scales the entire prop, in the Z-direction only, by the given factor.
-rx	<angle>	Rotates the entire prop around the X-axis by the given angle.
-ry	<angle>	Rotates the entire prop around the Y-axis by the given angle.
-rz	<angle>	Rotates the entire prop around the Z-axis by the given angle.
-cl	<x1> <y1> <z1> <x2> <y2> <z2>	Applies a clipping region to the entire prop, bounded by the given 3D co-ordinates.

## r3-tm-dress2prop

This utility converts saved set dressing files into a single prop. This is useful for filming on any of the blue-screen variations, where you want to position and rotate the set around the actors.

To successfully use this utility, you must first dress up your set-as-a-prop in Advanced Movie Maker. Place all your props as desired, and save the set dressing.

While not necessary, it is helpful to save it as a filename that will match the name of the prop you ultimately want. Avoid using spaces, and don't make the name longer than 25 characters. For example, if you were creating a bedroom for a character named Joe, you might call it "bedroom\_joe".

Once you're sure you've finalized the set dressing, enter the following at a command prompt (replacing the name as necessary):

```
r3-tm-dress2prop bedroom_joe.ini
```

A new folder named bedroom\_joe will be created. In this folder, the necessary files and directory structure will be created for the prop, as shown below:

```
\data\meshes\p_bedroom_joe.msh  
\data\propblueprint\p_bedroom_joe.ini  
\data\props\p_bedroom_joe.ini  
\data\setdressing\p_bedroom_joe.ini
```

Copy these files into the same folders in your game directory.

Note that if your set dressing file does contain spaces, you must surround it with quote marks (e.g. "bedroom for joe.ini"), and must set the prop name as indicated in the Advanced Options section below.

### Using Windows Explorer and Drag-and-Drop

As of Version 0.0.21a, you can now use Windows Explorer to drag-and-drop your set dressing INI files onto the r3-tm-dress2prop application. Open two Explorer windows.

1. In the first window, navigate to the \My Documents\The Movies\Set Dressing\ folder, and then to the sub-folder of the set containing your set dressing files.
2. In the second window, navigate to the folder containing the r3-tm-dress2prop application.

3. In the first window, highlight all the set dressing files you want to convert into props.
4. Drag them over to the second window, onto the r3-tm-dress2prop application.

## Advanced Options

The above instructions are for the most basic usage of this utility. The tables below specify additional input options you can include to achieve more advanced behaviour.

The following options affect all props output by this command.

Option	Parameter(s)	Description / Usage
-newdata	<full-folder-path>	Specifies an additional directory to search for game data. This is useful if you have props that you don't want included in your copy of the game, but still want available for this utility.
-workdata	<folder-name>	Specifies the name of the folder that will contain the output of this utility. By default, this is the base name of your set dressing file. This is useful if you want to create many props from various set dressing files, but want to keep them all together.

The following options affect the most recent set dressing file in the command.

Option	Parameter(s)	Description / Usage
-propname	<prop-name>	Specifies the base name of the prop to be created. This should include the "p_" at the beginning, but not the ".msh" at the end. For example, "p_bedroom_joe".
-cat	<category-num>	Specifies the category number to place the prop in AMM. This should be a number from 0 to 9. The default is 9 (miscellaneous).
-lightmap	<dds-filename>	For those props that do not have a light-map already set, this sets their light-map texture. This only works for textures that are a single colour.

## An Advanced Example

Consider the following command:

```
r3-tm-dress2prop -workdata mymovie "bedroom for  
joe.ini" -propname p_bedroom_joe "bedroom for joe.ini"  
-propname p_bedroom_joe_dark -lightmap  
lm_genericdark.dds "living room for joe.ini" -propname  
p_livingroom_joe
```

This will do this following:

1. Set the output directory to "\mymovie\".
2. Create a prop named "p\_bedroom\_joe" from the set dressing file named "bedroom for joe.ini".
3. Create a prop named "p\_bedroom\_joe\_dark" from the set dressing file named "bedroom for joe.ini", using the light-map texture "lm\_genericdark.dds".
4. Create a prop named "p\_livingroom\_joe" from the set dressing file named "living room for joe.ini".

## Editing a Set Dressing INI File

Occasionally, you might find it useful to fine-tune your set dressing files. You can open a set dressing INI file in Notepad or any other text editor. Once you have made adjustments to the location or angle-of-rotation of any props, run the r3-tm-dress2prop command against it.

Each prop in a set dressing INI file is placed in its own section, identified by a number enclosed in square brackets (on its own line). Each of these prop sections identifies the prop by filename, and provides the position and angle-of-rotation. The position information is measured (approximately) in *meters*, and the angle is given in *radians*. You can convert radians to degrees with the following formula:  $D = (R * 180) / \pi$ .

Typically, you will want to fine-tune a set dressing INI file in the following ways:

- Adjust the Z position of any set of props that should be at the same height.
- Adjust the angle-of-rotation of any set of props that should be facing the same direction.

These minor adjustments can help solve situations where props get slightly misaligned, due to the clumsiness inherent in placing props in Advanced Movie Maker.

# r3-tm-buildroom

This utility creates new room interiors, or building exteriors, based on script files containing a list of wall co-ordinates. The rooms it creates will have flat floors and ceilings, and vertical walls. Advanced options include: cutouts for doors and windows, custom texture support, and support for light-maps.

## What You Need

- You must create script files, ending in “.ini”, that define your room. Read the Basic Usage section for more information.
- To insert windows and doors into your walls, you must download and install my prior “Complete Works” modpacks. This includes the Hotel Bedroom, Suburban Living Room, and Casting Office packs.
- The file “r3-tm-cutouts-01.ini” contains the information needed for the windows and doors in my “Complete Works” modpacks to work in your rooms.

## Basic Usage

To use this utility, you must first type up one or more script files that define your room. You build your rooms much like an architect might. Draw the shape of the room on paper (graph paper, if you have it). Mark a point for the center of the room. For each corner in the room, write out how far away that point is from the center, measured in meters, in both the horizontal and vertical directions. Use negative numbers to the left and down from the center.

Once you have drawn the room, you can start typing up the script file. Open up your favourite text editor. Windows comes with a basic text editor called Notepad, which is found in Start → Programs → Accessories. The filename should end in “.ini”, and you should avoid using spaces. To start with, use the same base name as the prop you wish to create.

Each corner of the room requires a [wall] section. Start with the upper-left corner, and work your way around the room in a clockwise fashion. If you had a corner that was 5 meters to the *left* of center (left is negative, right is positive), and 3 meters *above* the center (below is negative, above is positive), you would make a [wall] section that looks like the following:

```
[wall]
x = -5
y = 3
```

Once you have all the [wall] sections typed out, you have enough information for the application to create your room! This is the most basic use of this application. If your room INI file was named “p\_mybedroom.ini”, you would type the following at the command line:

```
r3-tm-buildroom p_mybedroom.ini
```

This will create a folder called “p\_mybedroom”, and create the full directory structure for you. You should strongly consider using the `-workdata` option, which specifies the name of the folder to place the prop files in. The following command will place the prop files in a folder called “mymovie”.

```
r3-tm-buildroom -workdata mymovie p_mybedroom.ini
```

By default, the props this application creates have walls that are 3 meters high. It will be an interior with a floor and ceiling. The last corner of the room will automatically connect up with the first corner. This behaviour can be customized with a [global] section. The example below sets the wall height to 5 meters, raises the room off the floor 0.06 meters, does not include a ceiling, and does not connect the last corner to the first corner.

```
[global]
height = 5
raise = 0.06
interior = true
floor = true
ceiling = false
connect = false
```

Note the somewhat strange number used to raise the room 0.06 meters. As it turns out, this is a good number to use for filming on the various Blue-Screen sets. If it is set any lower, the floor of your room will end up being buried under the Blue-Screen floor.

The walls, floor, and ceiling will use a default texture taken from the game’s texture library. You can change the textures to use with a [texture] section. Use Appendices E, F, and G to see what all the game’s wall, floor, and ceiling textures look like. The example below changes the textures to a concrete / brick look.

```
[texture]
wall = wall_ph07.dds
floor = floor_rk13.dds
ceiling = floor_rk13.dds
```

## Cutouts – Inserting Doors and Windows into the Walls

So far, the rooms you’ve seen are awfully claustrophobic! You can create “cutouts” in the walls to allow space for doors and windows. This can be done manually by specifying the bottom and top co-ordinates, or automatically by using pre-built “cutout libraries”. For now, we will cover automatic cutouts.

This application comes packaged with a cutout library for all the door and window props released in my “Complete Works” mod packs. See Appendix H for the complete list of props that come with this application. Keep an eye on my thread in Mod Showcase for future cutout libraries. Include the following [library] section at the top of your room INI file.

```
[library]
cutouts = r3-tm-cutouts-01.ini
```

You can now place a door or window prop into a wall by specifying a [wall] section. Treat each door or window as if it is just another corner in the room. The prop will be centered around the x and y position you specify, and raised up into the wall by the bottom value you specify.

```
[wall]
x = -5
y = 3

[wall]
x = -2
y = 3
cutout = p_door01
bottom = 0

[wall]
x = 0
y = 3
```

The above example will place the door prop p\_door01, centered at (-2, 3), into the wall section that is between the two corners at (-5, 3) and (0, 3).

For more complex props, such as curved or angled windows, you must use a bottom value greater than 0. If you want such a prop to appear aligned at the bottom of the wall, use a very small value like 0.0001.

Remember that all [wall] sections in your room should be entered in the proper clockwise order. This includes [wall] sections for cutouts!



## Manual Cutouts

If you want to create any kind of opening in the wall, you can also do so manually. Each manual cutout is created by specifying a bottom and top value. For basic cutouts, the opening will occur in the section of wall between the [wall] it is specified in, and the next [wall] in the room INI file.

To leave a section of wall completely open, use a bottom value of zero, and a top value equal to the height of your room. The example below will leave the wall between (-5, 3) and (0, 3) completely open, assuming the height of the room was set to 3 meters.

```
[wall]
x = -5
y = 3
bottom = 0
top = 3

[wall]
x = 0
y = 3
```

To leave the top portion of a wall open, use a bottom value greater than zero, and a top value equal to the height of your room.

To leave the bottom portion of a wall open, use a bottom value of zero, and a top value less than the height of your room.

More advanced cutouts can be created by specifying a bottom value greater than zero, and a top value less than the height of your room. When creating advanced cutouts, you should include one final [wall] section with both the bottom and top values set to zero. This sends a signal to the application that the cutout should be closed off.

## Set Dressing – Dressing up your Room

As an advanced option, you can dress up your room with additional props. This avoids the clumsiness of using the Advanced Movie Maker, and gives you more flexibility on scaling and tipping over each prop. However, it requires you to figure out the exact co-ordinates to position and rotate each prop.

For each prop you want to add to the room, include a [prop] section. Specify the base name of the prop, the position in the room, and the angle of rotation. The example below adds a cabinet to the room.

```
[prop]
prop = p_cabinet_book
x = -3
y = 0
z = 0
rotate = 90
```

You can also apply 3D transformations to the prop, including rotation around the X or Y axis (to tip it over), and scaling. See the section titled “Room INI File Format” for complete information.

## Texturing Tiling

By default, wall textures are tiled to match the height of the walls. This means that, if the height of the room is 3 meters, the texture will repeat itself horizontally every 3 meters along the wall. It also means that the application assumes your wall texture has the same width and height. You can change this behaviour with a [wall tiling] section.

Say you are using a texture that is 512x256 in size, or twice as wide as it is high. To ensure the texture doesn't look stretched horizontally, you will need to tell the Room Builder to repeat the texture at twice the distance. The example below assumes the room has a height of 3 meters.

```
[wall tiling]
xwrap = 6
```

By default, floor and ceiling textures are also tiled to match the height of the walls. This behaviour can be changed with a [floor tiling] section.

There are many more advanced ways you can tile or stretch textures across the walls, floor, and ceiling. You can also turn on the alpha channel (to make transparent walls), or turn on texture animation. See the section titled “Room INI File Format” for complete information.

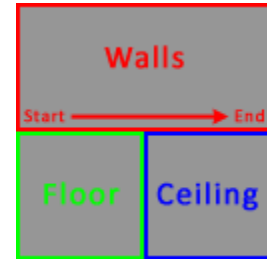
## Enhanced Light-map Support

As of version 0.0.22a, the way light-maps are applied to your rooms has been improved. The [lightmap] section has been completely changed to adapt to the new method, so if any of your existing room scripts used light-maps, you will have to update them.

You attach a light-map to your room with the [lightmap] section. You are responsible for creating the texture yourself, in image editing software such as Photoshop or GIMP. Light-maps do not need to be particularly high resolution. It is a good idea to use filenames that begin with “lm\_”, and to place them in the \data\textures\lightmap\ folder.

```
[lightmap]
lightmap = lm_myroom.dds
```

The image to the right illustrates how light-map textures will be applied to your room. The top-half of the texture is reserved for the walls, beginning at the left edge and working its way around to the right edge. The bottom-left quarter of the texture is reserved for the floor. The bottom-right quarter of the texture is reserved for the ceiling. If the room is not rectangular in shape, the floor and ceiling will not use the entire area of their portion of the texture.



In order to provide the most flexibility in designing your own custom light-maps, you may also specify the bounding region of the texture. This allows you to combine multiple portions of a room (or multiple separate rooms) onto the same light-map. This strategy is useful for saving memory. See the section titled “Room INI File Format” for complete information.

## More Complex Rooms

The Room Builder application can combine multiple room INI files into the same prop. There are many reasons you might want to do this. Your room might have multiple levels, it might require different textures for various areas along the wall, or you might need to “stack” windows on top of each other.

To combine multiple room INI files together, simply indicate all the room INI files in the command. You should also include a `-propname` option to set the name of the final prop.

```
r3-tm-buildroom -workdata mymovie -propname p_myroom
room01-lowerhalf.ini room01-upperhalf.ini room01-
platform.ini
```

The above example will process three room INI files and combine them into a single prop named `p_myroom`. The prop will be placed in a folder named `mymovie`.

## Creating Room Libraries

The Room Builder application also allows you to perform 3D transformations against each room INI file you specify on the command line. This allows you the freedom to create many related rooms, and then stitch them together piecemeal as though they were lego pieces.

For example, say you were creating a hotel interior. Many of the hallways will probably look very similar. You could create each piece of the hotel separately (one for a straight hallway with some door entrances, another for a left-turn corner, another for a right-turn corner, and another with an elevator foyer). You would then piece together all the room INI files you need, and specify centering and rotation transformations as needed.

When creating and using “Room Libraries” in this manner, you will find it convenient to ensure the upper left corner of each room is found at X=0, Y=0. This will make positioning each room much easier.

```
r3-tm-buildroom -workdata mymovie -propname
p_hotelhall01 hotel-rightturn.ini -c -14 3 0 hotel-
straight.ini -c -8 0 0 hotel-straight.ini -c 0 0 0
hotel-rightturn.ini -rz 90 -c 8 -3 0
```

In the above example, we have pieced together two stretches of straight hallways, and two right-turn corners at either end. The `-c` option is used to center each room INI file at the given co-ordinates. The other commonly used transformation is `-rz`, which rotates the room (around the Z-axis). Additional rotation and scaling operations are available. See the section titled Full List of Options for complete information.

To ensure proper customization, you could also create room INI files that only contain [prop] sections for the numbers on the doors. While a room INI file without any [wall] sections will generate a warning message, a prop will still be created.

## Room INI File Format

The following sections contain complete reference information on the file format of room INI files, and cutout library INI files.

### [library] Section

This section allows you to specify “cutout libraries”. A cutout library is a separate INI file that lists door and window props that can be used as cutouts. This application comes with a cutout library for all the door and window props from my various “Complete Works” mod packs.

Parameter Name	Type of Value	Description
cutouts	Filename	Specifies the filename of a cutout library that this room will make use of.

## [global] Section

This section specifies global information about the room or building.

Parameter Name	Type of Value	Description
height	Number	Height of the walls, measured in meters. This is 3 by default
raise	Number	Raised the entire room or building by the given amount. Use negative numbers to lower the room. This is 0 by default.
interior	true / false	Indicates whether this should be an interior or exterior. This is true by default
connect	true / false	Indicates whether the last [wall] section should automatically connect up to the first [wall] section. This is true by default.
floor	true / false	Indicates whether to include the floor as part of the room. This is true by default.
ceiling	true / false	Indicates whether to include the ceiling as part of the room. This is true by default.

## [texture] Section

This section specifies the texture files to use for the room, as well as optional texture-related information. If this section is missing, the room will use default textures that come from the game.

Parameter Name	Type of Value	Description
wall	Filename	Specifies the filename of the base texture to use for the walls.
wallalpha	true / false	Indicates the wall texture should use the alpha channel. This is false by default.
wallxscroll	Number	Sets the horizontal scroll speed of the wall texture (0 to 255). This is zero by default.
wallyscroll	Number	Sets the vertical scroll speed of the wall texture (0 to 255). This is zero by default.
wallrotate	Number	Sets the rotation speed of the wall texture (0 to 255). This is zero by default.

Parameter Name	Type of Value	Description
wallreflective	Filename	Specifies the filename of the reflective texture to use for the walls.
wallspecular	Filename	Specifies the filename of the specular texture to use for the walls.
floor	Filename	Specifies the filename of the base texture to use for the floor.
flooralpha	true / false	Indicates the floor texture should use the alpha channel. This is false by default.
floorxscroll	Number	Sets the horizontal scroll speed of the floor texture (0 to 255). This is zero by default.
flooryscroll	Number	Sets the vertical scroll speed of the floor texture (0 to 255). This is zero by default.
floorrotate	Number	Sets the rotation speed of the floor texture (0 to 255). This is zero by default.
reflection	true / false	Indicates the floor should reflect the geometry above it. This is false by default.
floorreflective	Filename	Specifies the filename of the reflective texture to use for the floor.
floorspecular	Filename	Specifies the filename of the specular texture to use for the floor.
ceiling	Filename	Specifies the filename of the base texture to use for the ceiling.
ceilingalpha	true / false	Indicates the ceiling texture should use the alpha channel. This is false by default.
ceilingxscroll	Number	Sets the horizontal scroll speed of the ceiling texture (0 to 255). This is zero by default.
ceilingyscroll	Number	Sets the vertical scroll speed of the ceiling texture (0 to 255). This is zero by default.
ceilingrotate	Number	Sets the rotation speed of the ceiling texture (0 to 255). This is zero by default.
ceilingreflective	Filename	Specifies the filename of the reflective texture to use for the ceiling.
ceilingspecular	Filename	Specifies the filename of the specular texture to use for the ceiling.

## [lightmap] Section

This section specifies the texture files to use as light-maps for the room. If this section is missing, the room will have no light-maps.

Parameter Name	Type of Value	Description
lightmap	Filename	Specifies the filename of the light-map texture to use.
xoffset	Number	Indicates how far horizontally into the light-map texture to begin (0.0 to 1.0). This is 0 by default.
yoffset	Number	Indicates how far vertically into the light-map texture to begin (0.0 to 1.0). This is 0 by default.
xsize	Number	Indicates how wide of an area in the light-map texture to use (0.0 to 1.0). This is 1 by default.
ysize	Number	Indicates how high of an area in the light-map texture to use (0.0 to 1.0). This is 1 by default.

## [wall tiling] Section

This section tells the Room Builder how to tile, or stretch, the texture around the walls. The texture is assumed to be square in shape, and by default will stretch across the entire height of the wall.

Parameter Name	Type of Value	Description
xoffset	Number	Indicates how far horizontally into the texture to begin (0.0 to 1.0). This is 0 by default.
yoffset	Number	Indicates how far vertically into the texture to begin (0.0 to 1.0). This is 0 by default.
xreverse	true / false	Indicates the texture should be flipped horizontally. This is false by default.
yreverse	true / false	Indicates the texture should be flipped vertically. This is false by default.
xwrap	Number	Specifies the distance (in meters) before the texture will wrap, or repeat, horizontally. This is the height of the room by default.
ywrap	Number	Specifies the distance (in meters) before the texture will wrap, or repeat, vertically. This is the height of the room by default.

## [floor tiling] Section

This section tells the Room Builder how to tile, or stretch, the texture across the floor and ceiling. The upper-left corner of the texture will be aligned to X=0, Y=0; unless you specify an xoffset and yoffset parameter.

The parameters in this section are identical to the parameters found in the [wall tiling] section described above.

## [wall] Section

This section specifies the corner of a room, or the center position for an automatic cutout.

Parameter Name	Type of Value	Description
x	Number	X co-ordinate value.
y	Number	Y co-ordinate value.
bottom	Number	Distance from the floor to open the cutout, along the bottom edge.
top	Number	Distance from the floor to open the cutout, along the top edge. This parameter is only used for manual cutouts.
exclude-bottom	true / false	Indicates whether to render the bottom portion of the wall. This parameter is only used for manual cutouts, and is false by default.
exclude-top	true / false	Indicates whether to render the top portion of the wall. This parameter is only used for manual cutouts, and is false by default.
cutout	Text	Name of the cutout to use. This parameter, and all parameters below, is only used for automatic cutouts.
useprop	true / false	Indicates whether to include the prop for the cutout. This is true by default.
prop	Text	Base name of the prop to insert. This is optional, and acts as a replacement for the prop specified in the cutout.
lightmap	Filename	Specifies the filename of the single-colour light-map texture to use for the prop.



Parameter Name	Type of Value	Description
scale	Number	Scale factor for the cutout and prop. This is 1.0 by default.
flip	true / false	Indicates whether to flip the cutout and prop horizontally. The prop will be rotated by 180 degrees. This is false by default.

## [prop] Section

This section adds a prop, as set dressing, to the room. The translate, rotate, and scale transformations are performed in the order specified.

Parameter Name	Type of Value	Description
prop	Text	Base name of the prop to insert. This parameter is required; all others are optional.
x	Number	Final X position of the prop.
y	Number	Final Y position of the prop.
z	Number	Final Z position of the prop.
lightmap	Filename	Specifies the filename of the single-colour light-map texture to use.
xtranslate	Number	Amount to translate the prop in the X direction.
ytranslate	Number	Amount to translate the prop in the Y direction.
ztranslate	Number	Amount to translate the prop in the Z direction.
rotate (or zrotate)	Number	Angle (in degrees) to rotate the prop around the Z-axis.
xrotate	Number	Angle (in degrees) to rotate the prop around the X-axis.
yrotate	Number	Angle (in degrees) to rotate the prop around the Y-axis.
scale	Number	Scale factor for the prop, in every direction.
xscale	Number	Scale factor for the prop, in the X direction only.
yscale	Number	Scale factor for the prop, in the Y direction only.
zscale	Number	Scale factor for the prop, in the Z direction only.

## [cutout] Section

This section exists only in cutout library INI files. Each cutout specifies the prop to use, and co-ordinate information for that prop. The prop should be flat against the X-axis. Each set of co-ordinate values specifies the X and Z co-ordinates for a vertical pair of vertices.

You can gather the co-ordinate information for a prop by using the Front view in Blender, and the 3D cursor information in the View Properties box.

The following parameters affect the cutout as a whole. The ycenter parameter should be used to ensure the prop will be snug against the X-axis. The xcenter and rotate parameters are only used in rare circumstances (if the prop itself is off center, or rotated improperly). When they are used, the x, bottom, and top co-ordinate values must be adjusted to compensate.

Parameter Name	Type of Value	Description
name	Text	Specifies the identifying name of this cutout. This can be the base name of the associated prop.
prop	Text	Specifies the base name of the associated prop. If this is missing, it defaults to the value of "name".
xcenter	Number	Specifies the center position of the prop along the X-axis. If this is missing, it defaults to 0.
ycenter	Number	Specifies the flat edge of the front of the prop (in the Y-direction). If this is missing, it defaults to 0.
rotate	Number	Specifies the angle (in degrees) to rotate the prop, around the Z-axis. If this is missing, it defaults to 0.

The following parameters are used together for each set of co-ordinates. There should be at least two sets of these parameters for the entire prop. All three parameters are required.

Parameter Name	Type of Value	Description
x	Number	Specifies the X co-ordinate value for this vertex (or vertical vertex pair).
bottom	Number	Specifies the bottom Z co-ordinate value for the bottom vertex.
top	Number	Specifies the top Z co-ordinate value for the top vertex.

## Full List of Options

The table below lists every input option available.

Option	Input Values	Description
<b>Global Options</b> Place these at the beginning of your command, before the input room INI files.		
-workdata	<work-data-folder>	Specifies the output folder that the prop will be placed in.
-newdata	<full-folder-path>	Specifies an additional location to search for game data files.
-propname	<prop-name>	Specifies the base name of the prop to be created. This should include the "p_" at the beginning, but not the ".msh" at the end.
-cat	<category-num>	Specifies the category number to place the prop in AMM. This should be a number from 0 to 9. The default is 9 (miscellaneous).
<b>Room-Level Options</b> Place these immediately after the input room INI filename. Note that all transformations occur in the order specified in the command. Perform rotation and scaling operations before centering.		
-c	<x> <y> <z>	Centers the room around the given 3D co-ordinates.
-s	<factor>	Scales the room by the given factor.
-sx	<factor>	Scales the room in the X-direction by the given factor.
-sy	<factor>	Scales the room in the Y-direction by the given factor.
-sz	<factor>	Scales the room in the Z-direction by the given factor.
-rx	<angle>	Rotates the room around the X-axis by the given angle, in degrees.
-ry	<angle>	Rotates the room around the Y-axis by the given angle, in degrees.
-rz	<angle>	Rotates the room around the Z-axis by the given angle, in degrees.

Option	Input Values	Description
-wt0	filename	Filename of the base wall texture to use. If specified, this will override the value given in the [texture] section.
-wt1	filename	Filename of the reflective wall texture to use.
-wt3	filename	Filename of the specular wall texture to use.
-ft0	filename	Filename of the base floor texture to use. If specified, this will override the value given in the [texture] section.
-ft1	filename	Filename of the reflective floor texture to use.
-ft3	filename	Filename of the specular floor texture to use.
-ct0	filename	Filename of the base ceiling texture to use. If specified, this will override the value given in the [texture] section.
-ct1	filename	Filename of the reflective ceiling texture to use.
-ct3	filename	Filename of the specular ceiling texture to use.
-t2	filename	Filename of the light-map texture to use. If specified, this will override the value given in the [lightmap] section.
-lm	<x1> <y1> <x2> <y2>	Specifies the texture co-ordinates to use for the light-map texture. If specified, these values will override those given in the [lightmap] section.

## DOS Batch Files

Microsoft DOS, the old 1980's operating system, still resides (in simulated form) in all versions of Windows. One very helpful thing in DOS is the ability to place a list of commands in a **batch file**. Batch files are just text files that end in \*.bat. Each command is run consecutively.

You are encouraged to place all related commands for a movie project in the same batch file. Just open up Notepad or your favourite text editor, and type away! Once you are ready to run your commands, enter the name of the file at a command prompt.

This technique is particularly useful if you are building many individual props for a particular movie project. You might have a series of r3-tm-extractprop commands to extract items from an existing set, followed by several r3-tm-dressprop commands to construct new sets-as-props.

## Appendix A - List of Sets

Set Name	Mesh Filename
Rural: Field	set_landscape_field.msh
Rural: Graveyard	set_graveyard.msh
Rural: Musty Cellar	set_cellarinterior.msh
Rural: Ranch Exterior	set_ranchexterior.msh
Rural: Shack Exterior	set_shackexterior.msh
Rural: Shack Interior	set_shackinterior.msh
Sci-Fi: Alien World	set_landscape_alienplanet.msh
Sci-Fi: Corridor 1	set_corridortz1.msh
Sci-Fi: Corridor 2	set_corridortz2.msh
Sci-Fi: Corridor 3	set_corridortz3.msh
Sci-Fi: Corridor 4	set_corridortz3_v2.msh
Sci-Fi: Corridor 5	set_corridortz4.msh
Sci-Fi: Starship Bridge 1	set_bridgetz1.msh
Sci-Fi: Starship Bridge 2	set_bridgetz2.msh
Sci-Fi: Starship Bridge 3	set_bridgetz3.msh
Sci-Fi: Starship Bridge 4	set_bridge_variation.msh
SFX: Blue Screen	set_blue_screen.msh
SFX: Green Screen	set_greenscreen.msh
SFX: Miniature City	set_mincity.msh
SFX: Devastated Miniature City	set_mincity_destroyed.msh
SFX: Scrolling Landscape	set_scrolling_miniature.msh
Stage	set_stage.msh
Suburban: Bathroom	set_bathroom_30s.msh
Suburban: Bathroom 2	set_bathroom.msh
Suburban: Haunted Corridor	set_corridor_haunted.msh
Suburban: Living Room 1	set_livingroom_1930s.msh
Suburban: Living Room 2	set_livingroom.msh
Suburban: Modern Jail	set_jail_1980s.msh
Suburban: School Corridor	set_schoolcorridor.msh
Suburban: School Library	set_schoollibrary.msh
Suburban: Street	set_suburbanstreet.msh
Traveling Vehicle: Automobile	set_car.msh
Urban: Alleyway	set_alleyway.msh
Urban: Bar	set_bar.msh
Urban: City Block Corridor	set_citycorridor.msh
Urban: City Street	set_citystreet.msh
Urban: Commercial City Street	set_citystreet_commercial.msh
Urban: Modern Bank	set_bank.msh
Urban: Modern Office	set_office_1980s.msh

<b>Set Name</b>	<b>Mesh Filename</b>
Urban: Municipal Reception	set_municipal_reception.msh
Urban: Office	set_office.msh
Urban: Plush Hotel Bedroom	set_hotelbedroom.msh
Urban: Rooftop	set_rooftop.msh
Urban: Shabby Hotel Bedroom	set_hotelbedroom_v01.msh
Urban: Subway Carriage	set_subwaycarriage.msh
Urban: Subway Station	set_subwaystation.msh
War: Barracks	set_barracks.msh
War: Battlefield	set_landscape_battlefield.msh
War: Bombed Street	set_bombedstreet.msh
Wild West: Bank	set_wwbank.msh
Wild West: Desert	set_landscape_desert.msh
Wild West: Jail	set_jail.msh
Wild West: Saloon	set_saloon.msh
Wild West: Saloon Bedroom	set_saloobedroom.msh
Wild West: Street	set_wwstreet.msh
Wild West: Traveling Vehicle	set_stagecoach.msh

## Appendix B - List of Facilities

Facility Name	Mesh Filename
Bar	fac_catering0.msh
Billboard	outl_billboardt.msh
Casting Office	fac_casting_office.msh
Cinema (outside studio lot)	p_debug12.msh
Commercial Shops and Municipal Buildings (outside studio lot)	p_debug00.msh → p_debug11.msh p_debug13.msh → p_debug27.msh p_debug38.msh → p_debug44.msh
Cosmetic Surgery	fac_cosmeticsurgery.msh
Crew Facility	fac_crewfacility.msh
Custom Scriptwriting Office	fac_customscriptoffice.msh
Hospital	fac_hospital.msh
Laboratory	fac_research.msh
Makeover Department	fac_wardrobe.msh
Post Production	fac_postproduction.msh
Production Office	fac_prod_office.msh
Publicity Office	fac_publicity_office.msh
Rehab	fac_detox.msh
Restaurant	fac_catering2.msh
Restroom: Small	fac_toilet1.msh
Restroom: Large	fac_toilet2.msh
Restroom: VIP	fac_toilet3.msh
Script Office: Basic	fac_scriptoffice_1.msh
Script Office: Intermediate	fac_scriptoffice_2.msh
Script Office: Proficient	fac_scriptoffice_3.msh
Script Office: First Class	fac_scriptoffice_4.msh
Snack Van	fac_catering1.msh
Stage School	fac_stageschool.msh
Staff Office	fac_gatehouse.msh
Star & Script Selling Facility	fac_sell.msh
Street Lights and other Municipal Infrastructure (outside studio lot)	p_debug28.msh → p_debug31.msh p_debug34.msh → p_debug37.msh lot_road_mark.msh
Studio Gates (including Gatehouse) and Studio Lot Walls	lot_gate.msh lot_wall_concrete.msh lot_wall_wood.msh lot_wallcorner.msh
Stunt School	fac_stuntschool.msh
Trailer: Rickety	fac_trailer1.msh
Trailer: Cheap	fac_trailer2.msh



Facility Name	Mesh Filename
Trailer: Comfortable	fac_trailer3.msh
Trailer: Plush	fac_trailer4.msh
Trailer: Palatial	fac_trailer5.msh
Training: Baptism of Fire	fac_fullbodyburn.msh
Training: Danger Dojo	fac_fightroom.msh
Training: Spartan Torture	fac_spartantd.msh

## Appendix C - List of Game Folders

The Movies game places all its game files in specific folders, depending on the purpose of each game file. Knowing which files go where is the key to creating mods for The Movies.

These utilities primarily deal with the following game folders:

- \data\meshes\
- \data\propblueprint\
- \data\props\
- \data\setdressing\

The table below lists all game folders used by The Movies.

Folder Name	Description of files
\data\	Initialization (*.ini) files for the game, mostly for the Tycoon mode.
\data\advisor\	Initialization (*.ini) files for the Tycoon mode.
\data\animation\facials\	Facial animation (*.fan) files for actors.
\data\animation\fx\	Effects (*.fx) files.
\data\animation\high\	Animation (*.anm) files for actors, animals, and all other animated objects in the game.
\data\animation\high\autoanimated\	Animation (*.anm) files for "auto-animated props", as well as facility buildings.
\data\animation\sequences\	Animation sequence (*.seq) files, which list a sequence of animations.
\data\backdrops\	Texture (*.dds) files for some backdrops. You should use \textures\backdrops\ for your custom backdrops.
\data\consoles\	Texture (*.dds) files for console fonts.
\data\costume\	Initialization (*.ini) files for costumes and costume categories.
\data\costume\datas\	Costume (*.cos) files for costumes. These can be edited by the Dead Sane Costume Editor.
\data\dlc\	Unknown DLC (*.dlc) files. (These look like "description" files - possibly unused, or possibly show up as little popup balloons when you hover the mouse over an object.)
\data\explainers\	Initialization (*.ini) files for interactive objects that can be placed on the studio lot. This is used for Tycoon mode.
\data\facility\	Initialization (*.ini) files for facility buildings.
\data\genre\	Initialization (*.ini) files for movie genres.

\data\head\	Head (*.hd) files for use by the Starmaker application.
\data\lots\	Initialization (*.ini) files for pre-built studio lots.
\data\meshes\	Mesh (*.msh) files for props, costumes, sets, facilities, and other 3D objects.
\data\meshes\extrainfo\	Extra information (*.inf) files for mesh files.
\data\meshes\fx\	Mesh (*.msh) files used when visual effects change the shape of an object (eg. broken glass).
\data\ornament\	Initialization (*.ini) files for ornaments that can be placed on the studio lot.
\data\person\	Initialization (*.ini) files for people in the game, such as the "celebrities" that appear in the Tycoon mode awards ceremonies.
\data\postproc\	Initialization (*.ini) files used in Post Production. This includes font information on titles and credits, as well as viewing filters.
\data\propblueprint\	Initialization (*.ini) files that indicate basic information about props, including which category they are placed under in Advanced Movie Maker.
\data\props\	Initialization (*.ini) files for props and prop categories. The game uses the information found here for "interactive props", such as guns, cigarettes, horses, etc.
\data\researchpacks\	Initialization (*.ini) files for research packs, used in Tycoon mode.
\data\savedefaults\	Initialization (*.ini) files for default studio lots for each era.
\data\scene\###\	Initialization (*.ini) files for scenes for a given type of set. The ### identifies the set.
\data\scene\interactions\	Film (*.flm) files for tycoon-mode scenes, such as when actors rehearse on a set.
\data\scene\interactions\###\	Film (*.flm) files for movie scenes, for a given type of set. The ### identifies the set.
\data\scripts\	Initialization (*.ini) files for game-generated movie scripts. This contains a number of sub-folders.
\data\serenitydata\	Texture (*.dds) file, unknown purpose.
\data\set\	Initialization (*.ini) files for sets.
\data\setdressing\	Initialization (*.ini) files for props that can be used as set dressing.
\data\sittsets\	Initialization (*.ini) files for various objects used in Tycoon mode.
\data\slider\	Initialization (*.ini) files for various Tycoon-mode sliders.
\data\technology\	Initialization (*.ini) files for technologies, such as camera, film stock, and sound.

\data\textures\	Texture (*.dds) files used for the game's sets, facility buildings, and other general purposes.
\data\textures\ accessories\	Texture (*.dds) files used for costume accessories. These files always begin with "acc_". Place your textures for custom accessories here.
\data\textures\ backdrops\	Texture (*.dds) files used for backdrops. These files always begin with "bd_". Place your custom backdrop textures here.
\data\textures\ blueprint\	Texture (*.dds) files used for specific game purposes.
\data\textures\ costumes\	Texture (*.dds) files used for costumes. Most of these files begin with "com_". These textures provide the various colour options for shirts, pants, boots, etc.
\data\textures\ flash\	Texture (*.dds) files used for the game's user interface.
\data\textures\ fx\	Texture (*.dds) files used for visual effects.
\data\textures\ hair\	Texture (*.dds) files used for actor's hair. These files always begin with "hair_".
\data\textures\ lightmap\	Texture (*.dds) files used for light-maps, which provide varied lighting schemes for sets and facilities. Files that begin with "lf#_" are used to cast shadows on actors and props. Files that begin with "lm_" are used to cast shadows on the sets and facility buildings themselves.
\data\textures\ makeup\	Texture (*.dds) files used to apply to the face, such as make-up, beards, eyebrows, etc.
\data\textures\ miniatures\	Texture (*.dds) files used as backdrops on the scrolling landscape set.
\data\textures\ overlays\	Texture (*.dds) files used as overlays. These files always begin with "overlay_". Place your custom overlays here.
\data\textures\ overlays\ overlay_thumbnails\	Texture (*.dds) files used for the thumbnails for overlays, displayed in Advanced Movie Maker.
\data\textures\ people\	Texture (*.dds) files used for actor heads and body skin. These files always begin with "head_" or "skin_".
\data\textures\ props\	Texture (*.dds) files used for props. Most of the files located here begin with "p_".
\data\textures\ sm\	Texture (*.dds) files used for the Starmaker application.

\data\textures\ thumbs\	Texture (*.dds) files used for thumbnails for various purposes. This contains several sub-folders, each with its own purpose. Most important are the \backdrops\, \props\, and \sets\ sub-folders.
\data\textures\ ui\	Texture (*.dds) files used for the game's user interface. This contains several sub-folders.
\data\tutorials\	Initialization (*.ini) files used for in-game tutorials in the Advanced Movie Maker and Post Production screens.
\data\ui\	Initialization (*.ini) files used for the game's user interface. This contains several sub-folders.
\data\vehicle\	Initialization (*.ini) files used for vehicles.

## Appendix D – List of Set Dressing Categories

The Movies game organizes props into various categories, when they are used as set dressing. The table below shows the available categories.

Use this table to decide which category to place your custom props under. By default, props are placed in category 9 (Miscellaneous).

Category Number	Category Name
0	Sci-Fi
1	Action
2	Flora
3	Vehicles
4	Professional
5	Household
6	Personal – Home
7	Personal – Other
8	Decorative
9	Miscellaneous

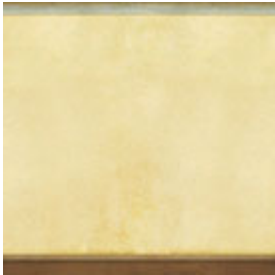
# Appendix E – List of Wall Textures



wall\_db01.dds



wall\_db02.dds



wall\_db02a.dds

(2 by 1)



wall\_db03.dds



wall\_db04.dds



wall\_db05.dds



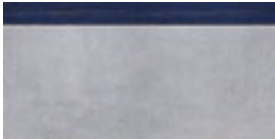
wall\_db08.dds

(2 by 1)

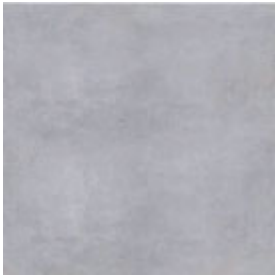


wall\_db10.dds

(2 by 1)



wall\_db11.dds



wall\_db13.ddb

(2 by 1)



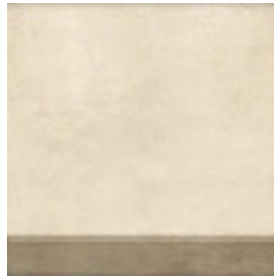
wall\_db14.dds



wall\_db15.dds



wall\_db18.dds



wall\_db19.dds

(2 by 1)

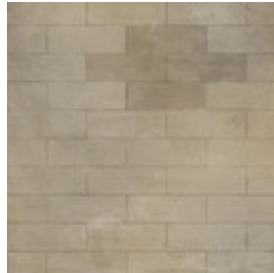


wall\_db22.dds

(2 by 1)



wall\_db23.dds



wall\_db24.dds



wall\_db32.dds

(2 by 1)



wall\_db33.dds

(2 by 1)



wall\_db34.dds



wall\_db35.dds

(2 by 1)



wall\_db36.dds



wall\_db37.dds



wall\_db38.dds

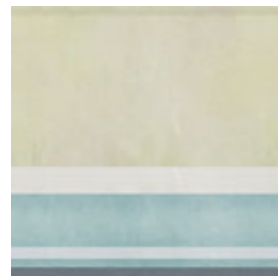
(2 by 1)



wall\_db39.dds



wall\_db40.dds



wall\_er01.dds



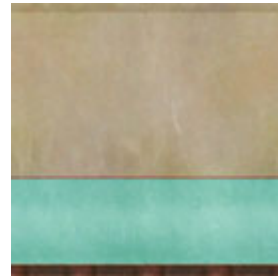
(4 by 1)



wall\_floortrim.dds



wall\_green\_er02.dds



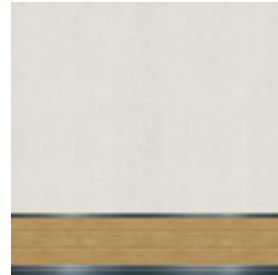
wall\_jt01.dds



wall\_mw01.dds



wall\_mw03.dds

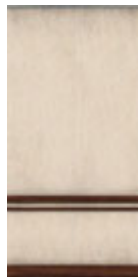


wall\_office\_rk01.dds



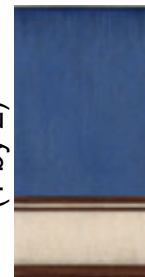
wall\_panel.dds

(1 by 2)



wall\_ph00.dds

(1 by 2)



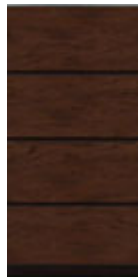
wall\_ph00a.dds

(1 by 2)



wall\_ph00b.dds

(1 by 2)



wall\_ph00c.dds

(1 by 2)



wall\_ph01.dds

(4 by 1)



wall\_ph02.dds

(8 by 1)

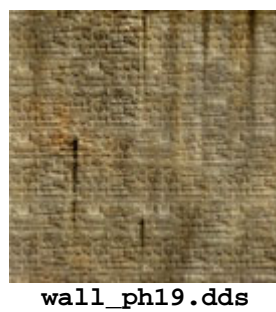
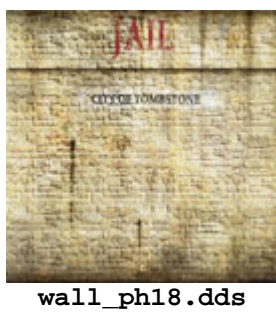
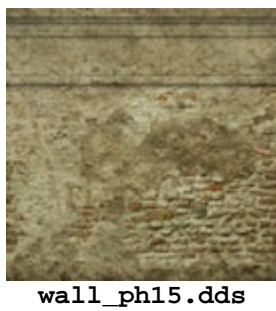
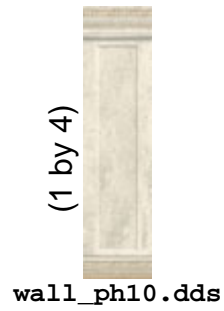
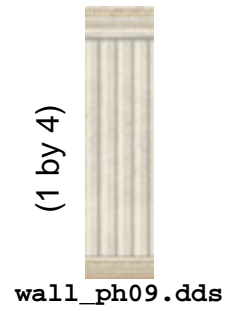
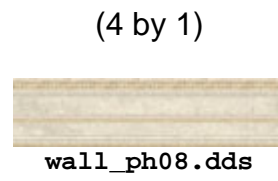
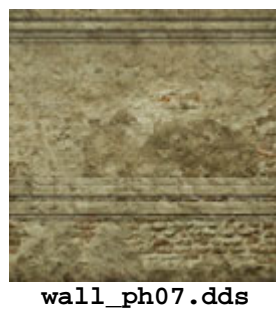
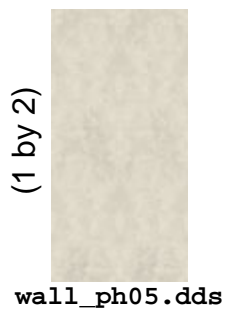


wall\_ph02b.dds

(1 by 4)



wall\_ph04.dds



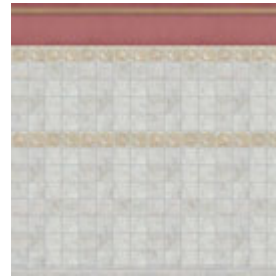
(1 by 2)



wall\_ph21.dds



wall\_ph22.dds



wall\_ph23.dds

(1 by 2)



wall\_ph24a.dds

(1 by 2)



wall\_ph25.dds



wall\_ph26.dds



wall\_ph27.dds



wall\_ph28.dds



wall\_ph29.dds



wall\_ph30.dds



wall\_ph30a.dds



wall\_ph31.dds



wall\_ph32.dds



wall\_ph34.dds



wall\_ph36.dds



wall\_ph36a.dds



wall\_ph36b.dds

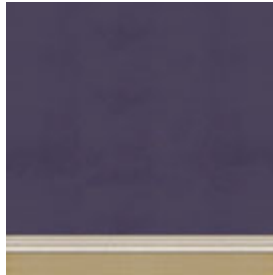


(1 by 2)

wall\_ph37.dds



wall\_ph38.dds



wall\_ph39.dds



(1 by 2)

wall\_ph40.dds



(1 by 2)

wall\_ph40a.dds



(1 by 2)

wall\_ph40b.dds



(1 by 2)

wall\_ph41.dds

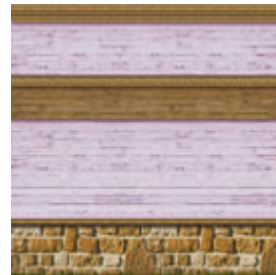


wall\_44.dds

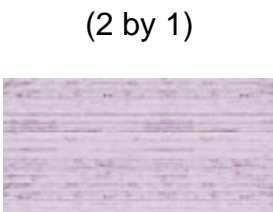


(2 by 1)

wall\_44a.dds



wall\_44b.dds

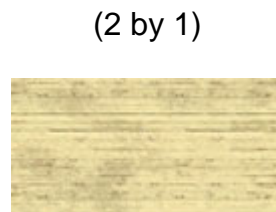


(2 by 1)

wall\_44c.dds



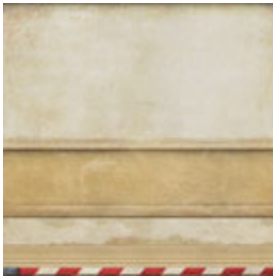
wall\_ph45.dds



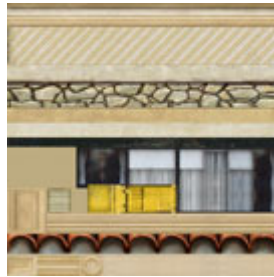
(2 by 1)

wall\_ph45a.dds

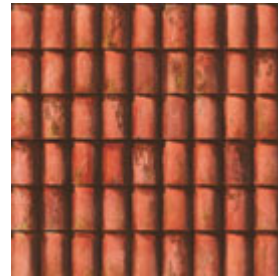




wall\_ph46.dds



wall\_ph46a.dds



wall\_ph46b.dds



wall\_ph46c.dds



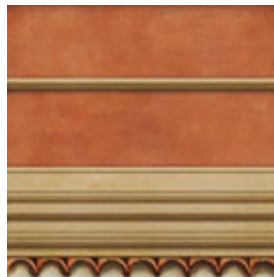
wall\_ph46d.dds



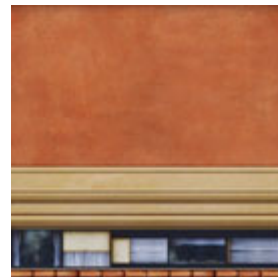
wall\_ph46e.dds



wall\_ph46f.dds



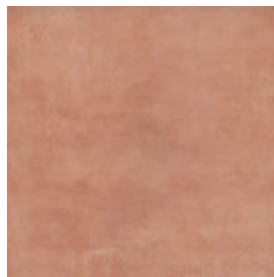
wall\_ph47.dds



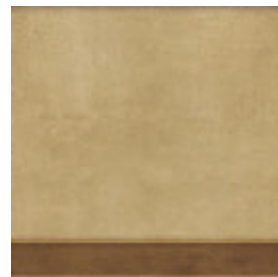
wall\_ph47a.dds



wall\_ph47b.dds



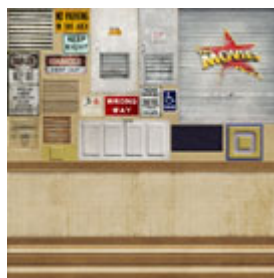
wall\_ph47c.dds



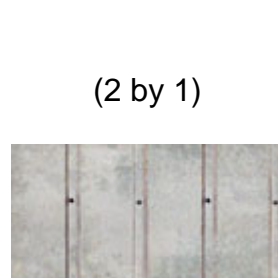
wall\_ph48.dds



wall\_ph48a.dds



wall\_ph48b.dds



wall\_ph48c.dds

(2 by 1)



wall\_ph49.dds



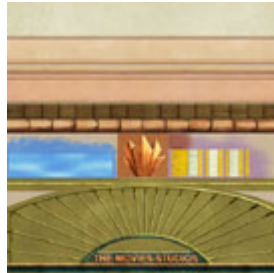
wall\_ph49a.dds



wall\_ph50.dds



wall\_ph50a.dds



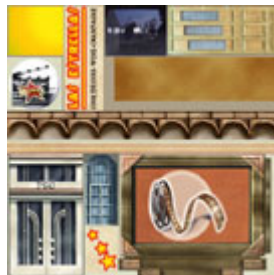
wall\_ph50b.dds



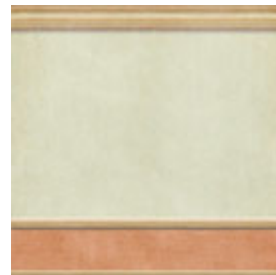
wall\_ph50c1.dds



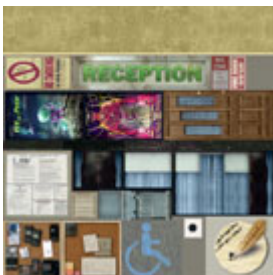
wall\_ph50c.dds



wall\_ph50d.dds



wall\_ph50e.dds

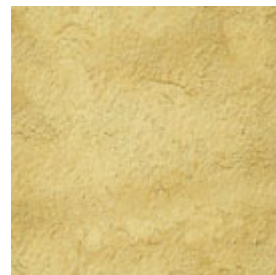


wall\_ph50f.dds

(2 by 1)



wall\_ph51.dds



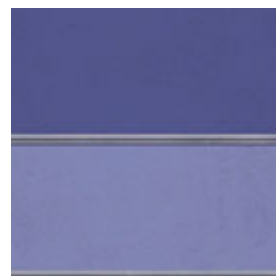
wall\_ph52.dds



wall\_ph53.dds



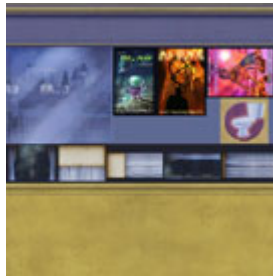
wall\_ph54.dds



wall\_ph54a.dds



wall\_ph54b.dds



wall\_ph54c.dds



wall\_ph54d.dds



wall\_ph55.dds



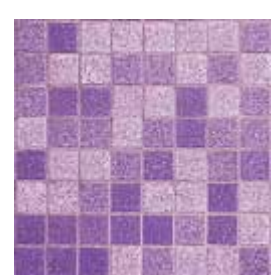
wall\_ph55a.dds



wall\_ph55b.dds



wall\_ph55c.dds



wall\_ph55d.dds



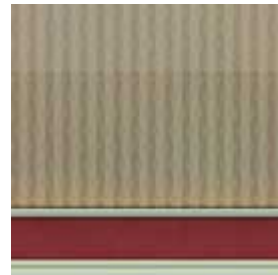
wall\_rk01.dds



wall\_rk02.dds



wall\_rk03.dds



wall\_rk07.dds



wall\_rk07b.dds



wall\_rk08.dds



wall\_rk10.dds



wall\_rk11.dds



wall\_rk12.dds



wall\_rk15.dds



wall\_rk16.dds



(1 by 2)

wall\_rk17.dds



wall\_rk18.dds



(1 by 2)

wall\_rk19.dds



wall\_rk21.dds



wall\_rk23.dds



wall\_rk23a.dds



wall\_rk24.dds



wall\_rk25.dds



wall\_rk27.dds



wall\_rk28.dds



wall\_rk30.dds





wall\_rk32.dds



wall\_rk33.dds



wall\_rk34.dds



wall\_rk35.dds



wall\_rk36.dds



wall\_rk37.dds



wall\_rk38.dds



wall\_rk40.dds



wall\_rk42.dds



wall\_rk43.dds



wall\_rw01.dds



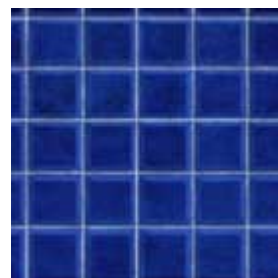
wall\_rw02.dds



wall\_rw02a.dds



wall\_rw02c.dds



wall\_rw03.dds

(1 by 2)



wall\_steel\_rw04.dds

(1 by 2)



wall\_steel\_rw05.dds



wall\_stucco\_rk01.dds



walltz2\_rk01.dds



walltz2\_rk02.dds



walltz2\_rk03.dds



walltz2\_rk04.dds



walltz2\_rk05.dds

## Appendix F – List of Floor Textures



floor\_blue\_er01.dds

(2 by 1)



floor\_jt01.dds



floor\_jt02.dds

(2 by 1)



floor\_patternlino\_er0.dds

(2 by 1)



floor\_rk01.dds



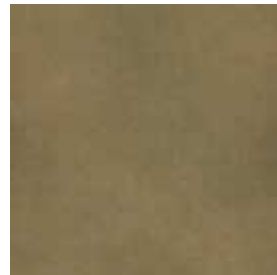
floor\_rk02.dds



floor\_rk02a.dds



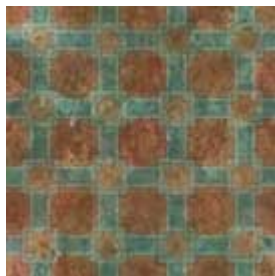
floor\_rk06.dds



floor\_rk08.dds



floor\_rk13.dds



floor\_rk18.dds



floor\_rk19.dds



**floor\_tilesbluegrey\_er01.dds**



**floortz2\_rk01.dds**



**floortz2\_rk02.dds**

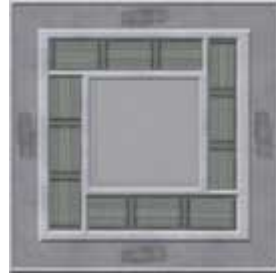
## Appendix G – List of Ceiling Textures



ceiling\_ph00.dds



ceiling\_rk00.dds



ceiling\_aircon\_cb01.dds



ceiling\_rk01.dds



ceiling\_wood\_rk01.dds

## Appendix H – List of Cutout Props

The Room Builder application comes with a library of pre-built cutout props. These can be referenced in room INI files in the `[wall]` section, with the `cutout` property.

### Library `r3-tm-cutouts-01.ini`

To use this cutout library, include the following at the top of your room INI file.

```
[library]
cutouts = r3-tm-cutouts-01.ini
```

The following door props are included in this cutout library.



`p_door01`



`p_door02`



`p_door03`



`p_door04`



`p_door05`

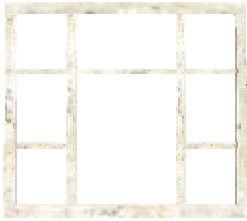


`p_door06`



`p_door07`

The following window props are included in this cutout library.



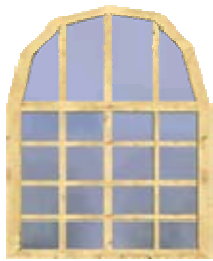
p\_windowframe01



p\_windowframe02



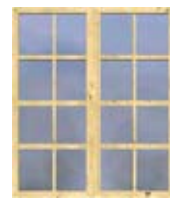
p\_windowframe03



p\_windowframe04



p\_windowframe05



p\_windowframe06



p\_windowframe07



p\_windowframe08

# Credits

These utilities were written in C++, and compiled with the MinGW compiler using Eclipse.

<http://www.mingw.org/>

<http://www.eclipse.org/>

These utilities make use of the ZLib compression library, written by Jean-loup Gailly and Mark Adler. This is a free API written wholly in the C programming language.

<http://www.zlib.net/>

Special thanks go to Gleem (Glen Rickey), who supported my The Movies modding efforts during the creation of these utilities.

Thanks also to stvndysn and rysto, for helping me to test these applications.

Thanks also to Reacher, for his excellent MeshManip application.

Thanks also to Nick Hudson and Mark Andrews, whose Blender scripts helped me decipher the MSH file format before I joined the DCMF team.

And thanks to Derobrash for The Movies Editor (MED)... Now that I've released these utilities in command-line form, it's time to look into incorporating their functionality into MED!