Main Page **Related Pages** Files 'version'-Command

Display the version/revision of Assimp used.

Generated on Sat Jun 14 2014 18:58:46 for Tools by doxygen 1.8.7



**Main Page Related Pages Files** 'help'-Command

Display a really helpful text.

Generated on Sat Jun 14 2014 18:58:46 for Tools by doxygen 1.8.7



Main Page Related Pages Files

'info'-Command

Load a model file and print basic statistics.

Full postprocessing is applied unless the -r switch is specified. Sample output (assimp info ./test/models/3DS/mar\_rifle.3ds):

```
Launching model import ...
                                       0K
Validating postprocessing flags ...
                                       0K
Importing file ...
                                       0K
   import took approx. 0.02400 seconds
Memory consumption: 69444 B
Nodes:
Maximum depth
                     2
Meshes:
                     1
Animations:
                     0
Textures (embed.):
                     0
Materials:
                     1
Cameras:
                     0
Lights:
                     0
                     843
Vertices:
Faces:
                     572
Bones:
Animation Channels: 0
Primitive Types:
                     triangles
Average faces/mesh
                     572
Average verts/mesh
                     843
                    (-3.522588 - 11.573204 - 40.340359)
Minimum point
Maximum point
                    (3.522622 30.196556 75.941292)
Center point
                    (0.000017 9.311676 17.800467)
```

```
Named Materials:
    'mat1'
Texture Refs:
    'm_rifl.bmp'
Node hierarchy:
'<3DSRoot>', meshes: 0
-- 'm_rifle', meshes: 1
<--
```

### **Syntax:**

```
assimp info file [-r]
```

#### **Parameters:**

file

Required. Input file.

-r

Optional. Don't perform any postprocessing. The long form of this parameter is -raw.

Main Page Related Pages Files

'cmpdump'-Command

#### **Used for Assimp's internal regression testing.**

Compare two mini dumps (produced using assimp dump .. -b -s) and report any differences. In the regression test suite, this facility is used to build a database of 'known-to-be-fine' mini dumps, which are regularly compared against current results to detect regressions.

#### Syntax:

assimp cmpdmp actual expected

#### **Parameters:**

actual

Required. Mini dump now.

expected

Required. Archived dump from some point in the past.

Main Page Related Pages Files

'dump'-Command

Generate a text or binary dump of a model.

This is the core component of Assimp's internal regression test suite but it could also be useful for other developers to quickly examine the contents of a model. Note that text dumps are not intended to be used as intermediate format, Assimp is not able to read them again, nor is the file format stable or well-defined. It may change with every revision without notice. Binary dumps (\*.assbin) are backwards- and forwards-compatible.

#### Syntax:

assimp dump <model> [<out>] [-b] [-s] [common
 parameters]

#### **Parameters:**

model

Required. Relative or absolute path to the input model.

out

Optional. Relative or absolute path to write the output dump to. If it is omitted, the dump is written to <model>-dump.txt

-b

Optional. If this switch is specified, the dumb is written in binary format. The long form of this parameter is -binary.

Optional. If this switch is specified, the dumb is shortened to include only min/max values for all vertex components and animation channels. The resulting file is much smaller, but the original model can't be reconstructed from it. This is used by Assimp's regression test suite, comparing those minidumps provides a fast way to verify whether a loader works correctly or not. The long form of this parameter is -short.

common parameters

Optional. Import configuration & postprocessing. See the **common parameters page** for more information.

#### Sample:

```
assimp dump test.3ds test.txt -l -cfull
assimp dump test.3ds test.txt -include-log -
    config=full
```

Dumps 'test.3ds' to 'test.txt' after executing full post-processing on tehe imported data. The log output is included with the dump.

```
assimp dump files\*.*
assimp dump files\*.*
```

Dumps all loadable model files in the 'files' subdir. The output dumps are named <mode-file>-dump.txt. The log is not included.

/ <u>doxygen</u> 1.8.7

Main Page Related Pages Files

'extract'-Command

Extracts one or more embedded texture images from models.

#### **Syntax:**

```
assimp extract <model> [<out>] [-t<n>] [-f<fmt>] [-
    ba] [-s] [common parameters]
```

#### **Parameters:**

model

Required. Relative or absolute path to the input model.

out

Optional. Relative or absolute path to write the output images to. If the file name is omitted the output images are named <model-filename>
The suffix \_img<n> is appended to the file name if the -s switch is not specified (where <n> is the zero-based index of the texture in the model file).

The output file format is determined from the given file extension. Supported formats are BMP and TGA. If the file format can't be determined, the value specified with the -f switch is taken. Format settings are ignored for compressed embedded textures. They're always written in their native file format (e.g. jpg).

-t<n>

Optional. Specifies the (zero-based) index of the embedded texture to be extracted from the model. If this option is *not* specified all textures found

are exported. The long form of this parameter is -texture=<n>.

```
-ba<n>
```

Optional. Specifies whether output BMPs contain an alpha channel or not. The long form of this parameter is <code>-bmp-with-alpha=<n></code>.

```
-f<n>
```

Optional. Specifies the output file format. Supported formats are BMP and TGA. The default value is BMP (if a full output filename is specified, the output file format is taken from its extension, not from here). The long form of this parameter is -format=<n>.

```
-s<n>
```

Optional. Prevents the tool from adding the \_img<n> suffix to all filenames. This option must be specified together with -t to ensure that just one image is written. The long form of this parameter is -nosuffix.

```
common parameters
```

Optional. Import configuration & postprocessing. Most postprocessingsteps don't affect embedded texture images, configuring too much is probably senseless here. See the **common parameters page** for more information.

#### Sample:

```
assimp extract test.mdl test.bmp --texture=0 --
   validate-data-structure
assimp extract test.mdl test.bmp -t=0 -vds
```

Extracts the first embedded texture (if any) from test.mdl after validating the imported data structure and writes it to test\_img0.bmp.

```
assimp extract files\*.mdl *.bmp
assimp extract files\*.mdl *.bmp
```

Extracts all embedded textures from all loadable .mdl files in the 'files' subdirectory and writes them to bitmaps which are named <model-file>\_img<image-index>.bmp



Main Page Related Pages Files

### **Common parameters**

The parameters described on this page are commonly used by almost every assimp command.

They specify how the library will postprocess the imported data. This is done by several configurable pipeline stages, called 'post processing steps'. Below you can find a list of all supported steps along with short descriptions of what they're doing.

**Programmers**: more information can be found in the aiPostProcess.h header.

Parameter	Long parameter	Description
-ptv	- pretransform- vertices	Move all vertices into worldspace and collapse the scene graph. Animation data is lost. This is intended for applications which don't support scenegraph-oriented rendering.
-gsn		Computes 'smooth' per-vertex normal vectors if necessary. Mutually exclusive with -gn
-gn	-gen-normals	Computes 'hard' per-face normal vectors if necessary. Mutually exclusive with -gsn
-cts		If one UV channel and normal vectors are given, compute tangents and bitangents
-jiv	-join- identical- vertices	Optimize the index buffer. If this flag is not specified all vertices are referenced once.
-rrm	-remove- redundant- materials	Remove redundant materials from the imported data.
-fd	-find- degenerates	Find and process degenerates primitives.

-split-large-	Split large meshes over a specific threshold
meshes	in smaller sub meshes. The default vertex & face limit is 1000000
-limit-bone- weights	Limit the number of bones influencing a single vertex. The default limit is 4.
-validate- data- structure	Performs a full validation of the imported data structure. Recommended to avoid crashes if an import plugin produces rubbish
-improve- cache- locality	Improve the cache locality of the vertex buffer by reordering the index buffer to achieve a lower ACMR (average post-transform vertex cache miss ratio)
-sort-by- ptype	Splits meshes which consist of more than one kind of primitives (e.g. lines and triangles mixed up) in 'clean' submeshes.
-convert-to- lh	Converts the imported data to left-handed coordinate space
-flip-uv	Flip UV coordinates from upper-left origin to lower-left origin
-flip- winding-order	Flip face winding order from CCW to CW
-evaluate- texture- transform	Evaluate per-texture UV transformations (e.g scaling, offset) and build pretransformed UV channels
-gen-uvcoords	Replace abstract mapping descriptions, such as 'spherical' or 'cylindrical' with proper UV channels
-fix-normals	Run a heuristic algorithm to detect meshes with wrong face winding order/normals.
-triangulate	Triangulate poylgons with 4 and more points. Lines, points and triangles are not affected.
-find- instances	Search the data structure for instanced meshes and replace them by references. This can reduce vertex/face counts but the postprocessing-step takes some time.
	-limit-bone-weights -validate-data-structure -improve-cache-locality -sort-by-ptype -convert-to-lh -flip-winding-order -evaluate-texture-transform -gen-uvcoords -fix-normals -triangulate -find-

- og	-optimize-	Simplify and optimize the scenegraph. Use it with care, all hierarchy information could be lost. Animations remain untouched.
- om	mesh	Optimize mesh usage. Meshes are merged, if possible. Very effective in combination with - optimize-graph

For convenience some default postprocessing configurations are provided. The corresponding command line parameter is -c<name> (or -config=<name>).

Name	Description	List of steps executed
fast	Fast post processing config, performs some essential optimizations and computes tangents	-cts, -gn, -jiv, -tri, -guv, -sbpt
default	Balanced post processing config; performs most optimizations	-cts, -gsn, -jiv, -icl, -lbw, -rrm, -slm, -tri, -guv, - sbpt, -fd, -fiv
full	Full post processing. May take a while but results in best output quality for most purposes	-cts, -gsn, -jiv, -icl, -lbw, -rrm, -slm, -tri, -guv, - sbpt, -fd, -fiv, -fi, -vds - om

The -tuv, -ptv, -og flags always need to be enabled manually.

There are also some common flags to customize Assimp's logging behaviour:

Name	Description		
-1 Or – show-log	Show log file on console window (stderr)		
-			
lo <file></file>			
	Streams the log to <file></file>		
out=			
<file></file>			

-v or – verbose

Enables verbose logging. Debug messages will be produced too. This might decrease loading performance and result in very long logs ... use with caution if you experience strange issues.

Generated on Sat Jun 14 2014 18:58:46 for Tools by doxygen 1.8.7





Here is a list of all files with brief descriptions:

dox\_cmd.h General documentation for assimp\_cmd





General documentation for assimp\_cmd. More...

### **Detailed Description**

General documentation for assimp\_cmd.



Main Page Related Pages Files

Related Pages

Here is a list of all related documentation pages:

'version'-Command	Display the version/revision of Assimp used
'help'-Command	Display a really helpful text
'info'-Command	Load a model file and print basic statistics
'cmpdump'-Command	Used for Assimp's internal regression testing
'dump'-Command	Generate a text or binary dump of a model
'extract'-Command	Extracts one or more embedded texture images from models
Common parameters	The parameters described on this page are commonly used by almost every assimp command

