

**NAME**

`oconv` - create an octree from a RADIANCE scene description

**SYNOPSIS**

`oconv` [ `-i` *octree* | `-b` *xmin ymin zmin size* ] [ `-n` *objlim* ] [ `-r` *maxres* ] [ `-f` ] [ `-w` ] [ `-` ] [ *input ..* ]

**DESCRIPTION**

`Oconv` adds each scene description *input* to *octree* and sends the result to the standard output. Each *input* can be either a file name, or a command (enclosed in quotes and preceded by a '!'). Similarly, the octree input may be given as a command preceded by a '!'. If any of the surfaces will not fit in *octree*, an error message is printed and the program aborts. If no *octree* is given, a new one is created large enough for all of the surfaces.

The `-b` option allows the user to give a bounding cube for the scene, starting at *xmin ymin zmin* and having a side length *size*. If the cube does not contain all of the surfaces, an error results. The `-b` and `-i` options are mutually exclusive.

The `-n` option specifies the maximum surface set size for each voxel. Larger numbers result in quicker octree generation, but potentially slower rendering. Smaller values may or may not produce faster renderings, since the default number (6) is close to optimal for most scenes.

The `-r` option specifies the maximum octree resolution. This should be greater than or equal to the ratio of the largest and smallest dimensions in the scene (ie. surface size or distance between surfaces). The default is 16384.

The `-f` option produces a frozen octree containing all the scene information. Normally, only a reference to the scene files is stored in the octree, and changes to those files may invalidate the result. The freeze option is useful when the octree file's integrity and loading speed is more important than its size, or when the octree is to be relocated to another directory, and is especially useful for creating library objects for the "instance" primitive type. If the input octree is frozen, the output will be also.

The `-w` option suppresses warnings.

A hyphen by itself ('-') tells `oconv` to read scene data from its standard input. This also implies the `-f` option.

The only scene file changes that do not require octree regeneration are modifications to non-surface parameters. If the coordinates of a surface are changed, or any primitives are added or deleted, `oconv` must be run again. Programs will abort with a "stale octree" message if they detect any dangerous inconsistencies between the octree and the input files.

Although the octree file format is binary, it is meant to be portable between machines. The only limitation is that machines with radically different integer sizes will not work together. For the best results, the `-f` option should be used if an octree is to be used in different environments.

**DIAGNOSTICS**

There are four basic error types reported by `oconv`:

warning - a non-fatal input-related error

fatal - an unrecoverable input-related error

system - a system-related error

internal - a fatal error related to program limitations

consistency - a program-caused error

Most errors are self-explanatory. However, the following internal errors should be mentioned:

Too many scene files

Reduce the number of scene files by combining them or using calls to `xform(1)` within files to create a hierarchy.

**Set overflow in addobject (id)**

This error occurs when too many surfaces are close together in a scene. Either too many surfaces are lying right on top of each other, or the bounding cube is inflated from an oversized object or an improper *-b* specification. If hundreds of triangles come together at a common vertex, it may not be possible to create an octree from the object. This happens most often when inane CAD systems create spheres using a polar tessellation. Chances are, the surface "id" is near one of those causing the problem.

**Hash table overflow in fullnode**

This error is caused by too many surfaces. If it is possible to create an octree for the scene at all, it will have to be done in stages using the *-i* option.

**EXAMPLE**

To add book1, book2 and a transformed book3 to the octree "scene.oct":

```
oconv -i scene.oct book1 book2 '\!xform -rz 30 book3' > newscene.oct
```

**AUTHOR**

Greg Ward

**NOTES**

In the octree, the names of the scene files are stored rather than the scene information. This means that a new octree must be generated whenever the scene files are changed or moved. Also, an octree that has been moved to a new directory will not be able to find scene files with relative pathnames. The freeze option avoids these problems. *make(1)* or *rad(1)* can be used to automate octree creation and maintenance.

**SEE ALSO**

getbbox(1), getinfo(1), make(1), obj2mesh(1), rad(1), rpict(1), rvu(1), rtrace(1), xform(1)