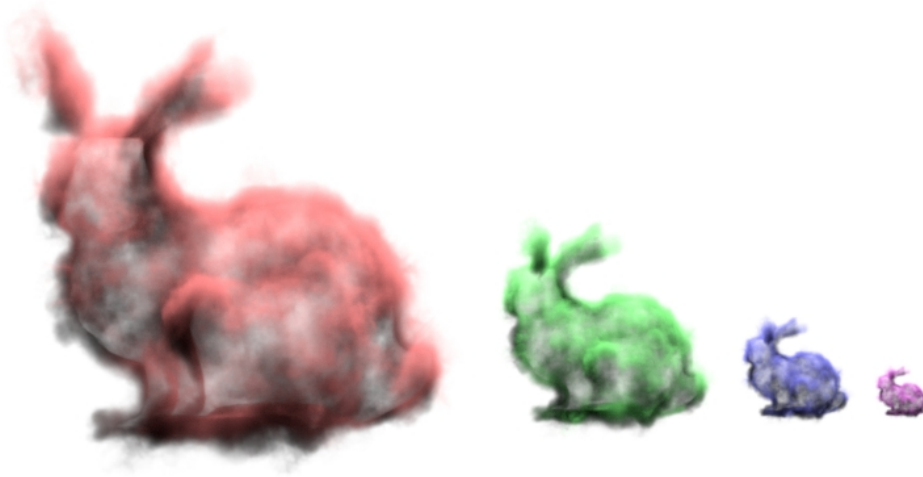


vdbmake



vdbmake is a tool that reads as input an [OpenVDB](#) file and writes out an OpenVDB file. All grids are copied from the input and written to the output, with some or all of those grids augmented with mipmap information and *minmax* information. The output file is optimized for use with [aggregate volumes](#).

Mipmapping

Mipmapping is a standard technique used to improve texture aliasing and rendering performance by automatically choosing an appropriate texture resolution level from a pyramid based on viewing size and angle of the texture on screen. This technique can also be applied to volumetric data comprised of voxels: in much the same way that a 2D mipmap pyramid level contains texels that are averages of four finer texels, a 3D mipmap pyramid level contains voxels that are each averaged from eight finer voxels. Mipmapping a volume asset will increase its disk space significantly, and may increase memory during rendering somewhat if the finest voxels are appropriately sized to the level of detail required for the camera; however, if the voxels are much finer in detail than the camera settings require, mipmapping can save significant RAM and render time because the renderer will only load coarser averaged voxels instead.

By default, when specifying:

```
> vdbmake input.vdb output.vdb
```

all grids from the input file `input.vdb` are read and processed to generate additional mipmap information. vdbmake optionally allows for a list of a subset of the grids to be specified as the last set of arguments:

```
> vdbmake input.vdb output.vdb density v
```

Grids on this list (in this case, "density" and "v") will have mipmap information generated, while grids that are not on this list will simply be copied to the output file.

The number of pyramid levels can be controlled by specifying the **-n** parameter, which takes an integer specifying the additional number of mipmap pyramid levels to generate. By default, 4 levels will be generated.

Minmax (baked acceleration data)

The *minmax* information that is generated by vdbmake is used in conjunction with volumes that have Attribute "volume" "int dsominmax" [1] to greatly decrease time to first pixel for aggregate volumes. The additional information takes the form of a dual mipmap pyramid that contains the minimum and maximum values of the source grid. By default, any float grids in the input file that are processed for mipmapping will also generate minmax information. As a space optimization, the optional **-minmax** parameter can be used to specify the name of the source grid that contains the density information; grids that do not match this name may get mipmapped but will not generate the additional dual minmax pyramid data.

```
> vdbmake -minmax density input.vdb output.vdb
```

Note that if the minmax data is not present in the VDB file, [impl_openvdb](#) will still generate it on the fly at load time if Attribute "volume" "int dsominmax" [1] is set. This may slow down time to first pixel somewhat.

Including vdbmake in your own toolchain

As part of the RenderMan distribution, we now ship libvdb.a and OpenVDBUtil.h, which allow OpenVDB savvy developers to write out aggregate volume optimized VDB files without having to invoke vdbmake directly.