

Extensions	Type	Read	Write	Notes
.geo	Internal	✓	✓	Houdini ASCII geometry format.
.bgeo	Internal	✓	✓	Houdini binary geometry format Prisms ASCII format (polygons only)
				This format only supports polygonal geometry types. It is included in the Geometry Editor to maintain compatibility with other, older, systems.
.poly	Internal	✓	✓	Tip In a .poly file, the first point referred to by the polygons is point number 0.
.bpoly	Internal	✓	✓	PRISMS binary format (polygons only)
.d	Internal	✓	✓	PRISMS move/draw format (polygons only)
.rib	Internal		✓	RenderMan geometry Uses gdx - polygons only, no attributes
				.dxf was invented by Autodesk for use with AutoCAD. It is a common geometry exchange format. .dxf format only supports polygonal geometry types.
.dxf	External	✓	✓	.dxf layers are mapped to groups when read into the SOP Editor. Texture and color information is lost if you save to a .dxf file. Uses gwavefront - polygons only. Only texture and pointnormal attributes (groups are kept on read).
.obj	External	✓	✓	.obj was invented by Wavefront. It is a very common geometry exchange format. .obj supports polygons, textures, normals, and groups. Uses ginventor. Does not support Bezier or metaball primitives. Only some attributes.
.iv	External/Internal	✓	✓	Inventor format was invented by SGI for use with their 3D libraries. It supports polygons, spheres, cylinders, and NURBS. VRML. Uses ginventor.
.wrl	External		✓	VRML was invented for viewing and exchanging 3D data on the Web. It is descended from the Inventor (.iv) format. Uses gsdl command.
				.sdl was invented by Alias/Wavefront. It was traditionally used as input to a renderer.
.sdl	External	✓		Houdini can read .sdl files but cannot write them. Houdini only imports spline surface geometry (and instances of spline surfaces), and the name of each patch and shader is imported. It does not import colors and texture coordinates. Uses geps command.
.eps	External	✓		Houdini reads .eps files from Illustrator 5.5 as planar geometry. It does not import fills, patterns, or colors. Uses gmed command.
.med	External	✓		Houdini will read Meta Editor metaball files, although eccentric information is lost upon loading. Uses glightwave command.
.lw	External	✓		.lw was invented by NewTek for use with LightWave. Houdini supports objects from version 3.5 and earlier. Houdini only

handles the following LightWave features:

- points
- polygons
- surface names (as primitive groups)
- surface color as primitive colors)
- surface transparency as primitive alpha)
- surface smoothing (as internal cusp operations)

Houdini will ignore all other features. It cannot load files saved with layer information. The file to be loaded *must* contain points. However, it need not contain any polygons.

Note

By default, LightWave objects are rendered as faceted (no smoothing) whereas in Houdini you must explicitly cusp polygons in order to achieve this. As a result, if you load an object that has no smoothing values set (all faceted) you will end up with all points in the resulting object being "unique-ed" - giving you a *much* larger data set. To get around this, manually convert the file using the glightwave utility with the -s option. Uses gply command

.ply was designed at Stanford University and UNC Chapel Hill, and is mostly used for academic research and with Cyberware scanners.

.ply	External	✓	✓
------	----------	---	---

The .ply format is a polygonal format. It can handle very large data sets, and supports vertex normals and colors. Houdini can read and write .ply files using the gply standalone program.

Image Formats

Extensions	Type	Read	Write	Notes
				Houdini picture format.
.pic	Internal	✓	✓	Houdini image files are run-length encoded disk files with the suffix .pic. The files contain color and color map information as well as the size of the picture.
.pic.gz **	Internal	✓	✓	Example source code for reading and writing Houdini image files is in \$HFS/houdini/public/sidefx.Pic.tar.Z. gzip compressed .pic
.pic.Z	Internal	✓	✓	Compressed .pic.
				Random Access Texture maps (RAT) are tuned for texture mapping. The format allows the renderer to access portions of the texture without having to load the whole image into memory at once.
				The RAT file format also supports arbitrary channel depth, meaning that a single channel image can be used as a texture map.
.rat	Internal	✓	✓	Using .rat files is faster for texture mapping, and typically consumes less memory than other formats because portions of the map are flushed out if more data is required for rendering. Only the portions of the map required for rendering are loaded. By default, mantra allocates a maximum of 8 MB of RAM for rat files. you can specify the maximum amount of ram (in MB) used for texturing with the environment variable SESI_RAT_USAGE. For example setenv SESI_RAT_USAGE 32.

We recommend using .rat files for texture maps before any other

				format because the quality tends to be much better than using stochastic sampling.
ip / iw / md	External	✓	✓	isplay / image window.
fip	Internal	✓	✓	Flipped iplay window.
vf / a60	Internal	✓	✓	<i>VideoFramer/Abekas</i>
.cin / .kdk	Internal	✓	✓	Kodak Cineon format
.fit	External	✓	✓	FIT tiled image format
.gif	External	✓	✓	GIF
.gif89	External	✓	✓	GIF89a (GIF with 1-bit alpha)
.jpg / .jpeg	Internal	✓	✓	JPEG (Very efficient for storage; lossy)
.qtl	External	✓	✓	Quantel yuv format Wavefront format
.rla / rlb	Internal	✓	✓	Wavefront pictures are treated like other image file formats but have one additional feature. The gamma value placed in Wavefront .rla files defaults to 2.2, but can be overridden by setting the WFGAMMA environment variable to the desired value.
.rla16	Internal	✓	✓	Wavefront .rla 16 bit format
.pix	Internal	✓	✓	Alias .pix format
.sgi / .rgb	Internal	✓	✓	SGI format (a.k.a. .rgb by non-Houdini software)
.rgba				
.si / .pic	Internal	✓	✓	Softimage format TIFF (Tagged Image File Format) is the standard used by RenderMan and many Mac and PC applications.
.tif / .tiff	Internal	✓	✓	If a file has a .tif3 file extension, Houdini uses tiff version 5.0 from mid-1990: LZW compression, RGB color space, and 32-bits per pixel (i.e. four channels per pixel - RGB and alpha, at eight bits per channel). If a file has a plain .tif extension, Houdini uses a newer TIFF library from early-1996 that is compatible with RenderMan 3.6. Adobe-Deflate codecs are also supported.
.tif3	Internal	✓	✓	TIFF RGB, no alpha
.tif16	External	✓	✓	TIFF 16 bit format
.tx	External	✓	✓	RenderMan texture images (requires RenderMan t.kit) The Targa and Vista file formats are treated identically. Houdini supports:
.tga / .vst	Internal	✓	✓	<ul style="list-style-type: none"> • Image type 10 (RGB Run Length Encoded) • Image type 2 (RGB Raw Data Stream) • Data Bits 15, 16, 24 and 32
.vtg	Internal	✓	✓	...and all combinations of the above (for example Type 10, 16 bits per pixel). Houdini's preferred extension for Vertigo files is .vtg, however Vertigo uses the extension .pic by default. Houdini will intelligently load Vertigo files with a .pic extension.
.yuv	Internal	✓	✓	Compressed Vertigo files (.Z) cannot be read by the frame buffer library, nor can they be created. Use the unix <i>uncompress</i> program before using them in Houdini. Abekas yuv format

Note

Some older software packages use .pic as an extension for SGI pixmap images (.rgb or .sgi). Houdini uses .pic for its own image format. Houdini should still load SGI pixmaps with the .pic extension correctly, but if you have problems try renaming the SGI file to have an .sgi extension.

Extensions	Type	Notes
		Houdini ASCII channel format containing raw values in rows and columns.
.chan	External	This information is written out into the file as ascii text, one row per frame of data, and one column per channel. The file contains specific information regarding channel names, sample rates etc. This format can be imported or exported from the Houdini channel editor (and is even compatible with old action channel formats). <i>Houdini binary channel format, equivalent to .chan</i>
.bchan	External	This file type is the same as the .chan format, except that the data is written in binary format. That is, the information is represented in a tightly-coded short-hand rather than as ascii characters you could read as text. It is more compact and loads more quickly. Houdini ASCII native CHOP Format. Contains raw values like .chan plus all the information held by a chop. Currently this format contains named channels, each with an array of raw sample values. The clip also contains a start and end range, a sample rate, channel extend conditions and quaternion attributes.
.clip	Internal	Example source code for reading and writing Houdini clip files is in \$HFS/houdini/public/CPD.tar.Z.
.bclip	Internal	This is the binary version of a clip. This format is recognized by the magic string bclip in its first four characters. Houdini ASCII format for a group of channels expressed as keyframed segments (e.g. bezier(), ease(), etc.)
.chn	External	This format is used to describe channels which contain segments, slopes, accelerations, interpolation types and other spline based attributes. It is compatible with the internal Houdini Channel Editor.
.bchn	External	Houdini binary format, equivalent to .chn
.aiff	Internal	Common lossless audio format, descended from IFF.
.aifc	Internal	Compressed version of AIFF.
.au	Internal	Old Sun/NeXT audio format.
.sf	Internal	NeXTAudio format.
.snd	Internal	Audio format
.wav	Internal	Windows lossless audio format.

External Houdini Programs (located in Houdini 8.1.704/bin). Note that the houdini_setup_bash must be sourced (Linux) in order to run these programs.

[chchan](#)

Copies channel collection to/from action channel format.

[chcp](#)

Copies channel collection file to another format.

[chinfo](#)

Prints information about a channel collection file.

[claudio](#)

Copies CHOP data (clip) to/from audio formats.

[clchan](#)

Copies CHOP data (clip) to/from action channel format.

[clchn](#)

Copies CHOP data (clip) to/from channel collection format.

[clcp](#)

Copies CHOP data (clip) to another format.

[clinfo](#)

Prints information about a CHOP data (clip) file.

[dsparse](#)

Parses and displays dialog scripts.

[gconvert](#)

Convert between Houdini polygon formats.

[gdx](#)

Converts DXF polygons to/from Houdini format.

[geps](#)

Converts EPS files to Houdini polygon format.

[gfont](#)

Install/remove/edit files in \$HFS/houdini/fonts.

[giges](#)

Converts iges geometry files into bgeos - Import Only

[ginfo](#)

Prints polygon file statistics.

[ginventor](#)

Converts inventor files to/from Houdini format.

[glightwave](#)

Converts LightWave files to/from Houdini format.

[gmed](#)

Converts MetaCorp files to Houdini format.

[gplay](#)

Geometry viewer.

[gpsx](#)

Converts Houdini geometry to Sony PSX geometry files.

[greduce](#)

Reduces polygons in a file.

[gsdl](#)

Converts Alias SDL files to Houdini format.

[gwavefront](#)

Converts .obj files to/from Houdini format.

[hmvplay](#)

Play Houdini (HVM) movies.

[hrender](#)

Command-line compositing renderer.

[hscript](#)

Command line HScript interpreter.

[hservice](#)

Installs/starts/stops/uninstalls Houdini services.

[hsop](#)

Edits objects and SOPs in a .hip file.

[ibblur](#)

Blur an image.

[ibulge](#)

Generate a bumpmap from the values of an image.

[ibumpmap](#)

Convert an image to a bumpmap.

[icineon](#)

Convert images from 10-bit Cineon format to an 8 bit format.

[iclear](#)

Clears a region of an image.

[icomposite](#)

[iconvert](#)

Converts image formats.

[icp](#)

Isolate a region of an image in a new image.

[icurve](#)

Distorts an image to simulate lens curvature.

[idof](#)

Generate an image with depth of field blurring.

[iexabyte - Exabyte Tape](#)

[ifilter](#)

Blurs, sharpens, or manipulates images.

[iflip - Flip Image](#)

[iflop](#)

Reflects an image.

[ifractal](#)

Creates a 2D fractal image.

[ihot](#)

Scans an image for video-unsafe colors.

[iinfo](#)

Outputs information about an image.

[ijpeg](#)

Converts Houdini formats to/from JPEG.

[ilookup](#)

Modifies image colors using gamma or a color lookup table.

[imdisplay](#)

Sends an image to an mdisplay window.

[inewtoold](#)

Converts old .pic format files.

[ipaint - Mini-Paint](#)

Mini paint program.

[iprint](#)

Prints the RGBA values for an image as text.

[iquantize](#)

Reduces the number of colors in an image.

[iramp](#)

Converts redit ramp files to images.

[isroll](#)

offset image content.

[ishear](#)

Skews an image.

[isi](#)

Convert Houdini images to SoftImage format.

[isixpack](#)

Generates a reflection map from the six sides of a cube.

[istipple](#)

Adds noise to an image.

[itim](#)

Convert Houdini images to Sony PSX format.

[izg](#)

Converts Z-depth images to 8-bit grayscale images.

[izmatte](#)

Composites RGB and Z-depth images.

[mcp](#)

Convert an image sequence to a movie file.

[minfo](#)

Prints information about movie files.

[redit](#)

Color ramp editor program.

[spy](#)

Shell utility for navigating the UNIX filesystem.