NAME

povray − POV−Ray − Persistence of Vision Ray Tracer v3.0

SYNOPSIS

povray [**+i** *input_file*] [**+o** *output_file*] [**+/−option** ...] [*INI_file* ...]

DESCRIPTION

POV−Ray is a free, full−featured ray tracer, written and maintained by a team of volunteers on the CompuServe GO GRAPHDEV Forum and on the Internet. On the Unix platform **POV−Ray** can be either a text−mode only program, or can optionally have preview capabilities using the *X Window System*. Under Linux, **POV−Ray** can optionally use the SVGA library to preview renderings.

This manual page only lists the basic **POV−Ray** and Unix specific features and command−line options for this version of **POV−Ray**. For a complete description of the features of **POV−Ray** and its scene description language, or for a better explanation of the meaning of the command−line and INI file options, please consult the documentation that should accompany all versions of **POV−Ray**. On some sites this will be installed in /usr/local/lib/povray3. Some of the Unix specific features are:

> Support for X Windows display automatically uses the best visual class and deepest depth available. For visuals that do not support 24 bits of color per pixel, Floyd−Steinberg error diffusion dithering is used, along with a dynamically allocated and optimized palette to produce the best display possible with the current visual, depth, and available colormap.
>
> ICCCM compliance for the X Windows version means that the preview window will behave like standard X Windows programs, communicate properly with the window manager, and will accept the standard command−line options. See *X(1)* for more information.
>
> Support for SVGAlib display automatically detects the available SVGA display modes to choose the size which best fits the rendered image. When rendering an image that is too large for the current display, the displayed image is scaled to fit on the screen. For displays that do not support true−color modes, Floyd−Steinberg dithering is used.
>
> ASCII graphics in the text−mode version allow a basic view of the current rendering on text−only terminals.
>
> An interrupt handler allows rendering to be interrupted in a safe way, so that any data not currently written to disk will be saved before exiting. Control−C or SIGINT will cause a user abort, and save the current rendering, before exiting. See *kill(1)* for more information.
>
> Platform and architecture independent rendering means the same scene will render in the same way on all computers and operating systems (with the exception of the rendering speed, of course).

OPTIONS

Options can be specified with either a leading ’+’ or a leading ’−’. Many options are switches, meaning a ’+’ turns the option on, and a ’−’ turns the option off. For other options, it doesn’t matter if a ’+’ or a ’−’ is used. Most options cannot have spaces in them so you should specify **+FN** rather than **+F N**, and combining options is not allowed, so **+SC is very different from +S +C. Options are not case sensitive.**

The command−line options are shown below with their corresponding INI file options. If the same option is specified multiple times, whether in INI files or on the command−line, the last such option overrides any previous ones, with the exception of the **+L** or **Library_Path** option, which is cumulative.

Parsing options:

I<*input_file_name*> or **Input_File_Name**=*file*

> Specifies the input file to use. If the input file name is ’−’, the scene description will be read from the standard input.

L<*library_path*> or **Library_Path**=*path*
Specifies a directory for to search for input files, include files, fonts, and image maps, if the specified file is not in the current directory.

MV*n.n* or **Version**=*float*
Treat scene files as if they were version *n.n* instead of the current version. This may be overridden from within the scene file.

SU or **Split_Unions**=*bool*
Split bounded CSG unions if children are finite. This allows automatic bounding of CSG objects to take place.

UR or **Remove_Bounds**=*bool*
Remove unnecessary bounding objects. This allows automatic bounding of older scene files to take place.

Output options:

H[*01234567*]
Display help on command−line options for section 0 through 7.

H*nn* or **Height**=*integer*
The image should be *nn* pixels high.

W*nn* or **Width**=*integer*
The image should be *nn* pixels wide.

SR*nn* or **Start_Row**=*integer*
Start the rendering at row *nn* from the top of the screen.

SR0.*nn* or **Start_Row**=*float*
Start the rendering *nn* percent from the top of the screen.

ER*nn* or **End_Row**=*integer*
End the rendering at row *nn* from the top of the screen.

ER0.*nn* or **End_Row**=*float*
End the rendering at *nn* percent from the top of the screen.

SC*nn* or **Start_Column**=*integer*
Start the rendering at column *nn* from the left of the screen.

SC0.*nn* or **Start_Column**=*float*
Start the rendering at *nn* percent from the left of the screen.

EC*nn* or **Start_Column**=*integer*
End the rendering at column *nn* from the left of the screen.

EC0.*nn* or **Start_Column**=*float*
End the rendering at *nn* percent from the left of the screen.

C or **Continue_Trace**=*bool*
Continue a previously interrupted trace.

P or **Pause_When_Done**=*bool*
If previewing, pause when the rendering is complete before closing the window.

V or **Verbose**=*bool*
Output verbose status messages on the progress of the rendering.

X or **Test_Abort**=*bool*
Enable the ’q’ and ’Q’ keys to interrupt a rendering in progress.

X*nn* or **Test_Abort_Count**=*integer*
Only check every *nn* pixels for a user abort.

Output options − display related:

D[0][*GHT*] or **Display**=*bool* **Palette**=*char*

Display the rendering in progress, optionally specifying the palette. The only valid X Windows palette option is *G*, which forces grayscale preview. The X Windows palette is based on the visual used, whether selected automatically by **POV−Ray** or via the **−visual** option. SVGA options are *T* for 24−bit true−color, and *H* for 15−bit high−color display, in addition to the default 256−color palette. To specify the palette, you must first specify the display type (the second character, shown here as ’0’) for compatibility reasons, even though it is ignored in Unix versions.

SP*nn* or **Preview_Start_Size**=*integer*

Start mosaic preview with blocks *nn* pixels square.

EP*nn* or **Preview_End_Size**=*integer*

End mosaic preview with blocks *nn* pixels square.

UD or **Draw_Vistas**=*bool*

Draw vista rectangles before rendering.

Output options − file related:

B*nn* or **Buffer_Output**=*bool* **Buffer_Size**=*integer*

Use an output buffer *nn* kilobytes in size.

F[*CNPT*][*n*] or **Output_to_File**=*bool* **Output_File_Type**=*char*

Store the rendered image using one of the available formats, namely *C*ompressed TGA, P*N*G, *P*PM, and *T*GA. PNG format supports the *n* option to specify the number of bits per color, where 5 <= *n* <= 16.

O<*output_file*> or **Output_File_Name**=*file*

Write the output to the file named *output_file*, or the standard output if ’−’ is given as the output file name.

HT[*CNPTX*] or **Histogram_Type**=*char*

Create a CPU utilization histogram image in format *x*. Available formats are *C*omma−separated values (CSV), P*N*G grayscale, *P*PM POV heightfield, uncompressed *T*GA POV heightfield, or *X* for no histogram generation.

HN<*histogram_file_name*> or **Histogram_Name**=*file*

Output the histogram to the specified file.

HS*x.y* or **Histogram_Grid_Size**=*float*

Divide the histogram into *x* columns and *y* rows of buckets.

Tracing options:

MB*nn* or **Bounding**=*bool* **Bounding_Threshold**=*integer*

Use automatic bounding slabs if more than *nn* objects are in the scene.

Q*n* or **Quality**=*integer*

Render at quality *n*. Qualities range from *0* for rough images and *9* for complete ray−tracing and textures, and *10* and *11* add radiosity.

A0.*n* or **Antialias**=*bool* **Antialias_Threshold**=*integer*

Do antialiasing on the pixels until the difference between adjacent pixels is less that 0.*n*, or the maximum recursion depth is reached.

AM*n* or **Sampling_Method**=*integer*

Specify the method of antialiasing used, non−adaptive (*n* = 1), or adaptive antialiasing (*n* = 2).

J*n.n* or **Jitter**=*bool* **Jitter_Amount**=*float*

Specify maximum radius, in pixels, that antialiased samples should be jittered from their true centers.

R*n* or **Antialias_Depth**=*integer*
Set the maximum recursion depth for antialiased pixel sub−sampling.

UL or **Light_Buffer**=*bool*
Use light buffer to speed up rendering.

UV or **Vista_Buffer**=*bool*
Use vista buffer to speed up rendering.

Animation options:

K*n.n* or **Clock**=*float*
Render a single frame of an animation with the clock value *n.n*.

KFI*nn* or **Initial_Frame**=*integer*
Specify the initial frame number for an animation.

KFF*nn* or **Final_Frame**=*integer*
Specify the final frame number for an animation. This must be set at a value other that 1 in order to render multiple frames at once.

KI*n.n* or **Initial_Clock**=*float*
Specify the clock value for the initial frame of an animation.

KF*n.n* or **Final_Clock**=*float*
Specify the clock value for the frame final of an animation.

SF*nn* or **Subset_Start_Frame**=*integer*
Render a subset of frames from an animation, starting at frame *nn*.

SF*nn* or **Subset_Start_Frame**=*float*
Render a subset of frames from an animation, starting *nn* percent into the animation.

EF*nn* or **Subset_End_Frame**=*integer*
Render a subset of frames from an animation, stopping at frame *nn*.

EF*nn* or **Subset_End_Frame**=*float*
Render a subset of frames from an animation, stopping *nn* percent into the animation.

KC or **Cyclic_Animation**=*bool*
Generate clock values for a cyclic animation.

UF or **Field_Render**=*bool*
Render alternate frames using odd/even fields, suitable for interlaced output.

UO or **Odd_Field**=*bool*
Start a field rendered animation on the odd field, rather than the even field.

Redirecting options:

GI<name> or **Create_Ini**=*bool* or **Create_Ini**=*file*
Write all INI parameters to a file named after the input scene file, or one with the specified name.

G[*ADFRSW*]<name> or **<Stream>_File**=*bool* or **<Stream>_File**=*file*
Write the stream to the console and/or the specified file. The streams are *A*ll_File (except status), *D*ebug_File, *F*atal_File, *R*ender_File, *S*tatistics_File, and the *W*arning_File.

X Windows options

In addition to the standard command−line options, the X Windows version recognizes additional command−line switches. See *X(1)* for a complete description of these options.

−display <*display_name*>
Display preview on *display_name* rather than the default display. This is meant to be used to change the display to a remote host. The normal dispay option /fB+d/fP is still valid.

−geometry [*WIDTHxHEIGHT*][+*XOFF*+*YOFF*]
Render the image with *WIDTH* and *HEIGHT* as the dimensions, and locate the window *XOFF* from the left edge, and *YOFF* from the top edge of the screen (or if negative the right and bottom edges respectively). The *WIDTH* and *HEIGHT*, if given, override any previous **W***nn* and **H***nn* settings.

−help Display the X Windows specific options. Use **−H** by itself on the command−line to output the general **POV−Ray** options.

−icon Start the preview window as an icon.

−title <*window_title*>
Override the default preview window title with *window_title*.

−visual <*visual_type*>
Use the deepest visual of *visual_type*, if available, instead of the visual automatically selected visual. Valid visuals are StaticGray, GrayScale, StaticColor, PseudoColor, TrueColor, or DirectColor.

RESOURCES

Currently no X resource or app−default files are supported for the X Windows options.

FILES

All of the Unix versions look for the file *.povrayrc* in the user's home directory upon startup for any initial configuration information, like the Library_Path settings, which gives the location for the standard include files. If this file is not found, the file *povray.ini* from the current directory is used, if available.

POVLEGAL.DOC should be accompany all installations of **POV−Ray**, and outlines specific conditions and restrictions on the **POV−Ray** software. A condition of POVLEGAL.DOC requires that documentation, INI, and scene files be available to all users of **POV−Ray** They are often installed in /usr/local/lib/povray3, but may be in other locations on some systems.

The most recent version of **POV−Ray** and its documentation can always be retrieved via anonymous FTP at ftp.povray.org, or via HTTP at http://www.povray.org/, as well as many other locations.

SEE ALSO

X(1), kill(1)

COPYRIGHT

POV−Ray is Copyright (C) 1991, 1996 the POV−Ray Development Team. Although it is distributed as freeware, it is **NOT PUBLIC DOMAIN** software. **POV−Ray** is subject to the restrictions as defined in POVLEGAL.DOC.

X Windows is Copyright (C) 1984 − 1991 the Massachusetts Institute of Technology, and also Copyright (C) 1992 − 1996 the X Consortium, Inc.

TRADEMARKS

Persistence of Vision, *POV−Ray*, and *POV−Help* are registered trademarks of the POV−Ray Development Team.

BUGS

The SVGA version does not currently generate an optimized grayscale palette for grayscale preview, so displays using the *G* palette option will use very few gray levels, and a few non−gray colors.

Before reporting a bug to the authors, you should make sure you have the latest version of the software, in case the bug has already been fixed. There are a large number of **POV−Ray** users on the USENET newsgroup comp.graphics.rendering.raytracing that may be able to help you with your problem. If you are having problems compiling, installing, or running the software, you should seek help locally or on USENET, rather than from the authors.

If you have a repeatable bug in the most recent version, try to isolate the bug in the smallest scene file possible. The POV−Ray Team Co−ordinator is Chris Young, and can be reached at 76702.1655@compuserve.com. Do not send large binary or uuencoded files to Chris without first asking permission to do so.

AUTHORS

The primary developers, in alphabetical order are:

Steve Anger Dieter Bayer Chris Cason
Chris Dailey Andreas Dilger Steve Demlow
Alexander Enzmann Dan Farmer Timothy Wegner
Chris Young

Major contributing authors, in alphabetical order are:

Steve A. Bennett David K. Buck Aaron A. Collins
Pascal Massimino Jim McElhiney Douglas Muir
Bill Pulver Robert Skinner Zsolt Szalavari
Scott Taylor Drew Wells

Other authors are listed in the documentation.

ACKNOWLEDGEMENT

POV−Ray is based on DKBTrace 2.12 by David K. Buck and Aaron A. Collins.