# Grammar rules for the grammar

This is a consolidation of the entire syntax for the POV-Ray's Scene Description Language.

Note that the syntax conventions used here are slightly different than those used in the user documentation.

The following syntax conventions are used:

ITEM

An item not in brackets indicates that it is a required item.

[ITEM]

Brackets surround an optional item. If brackets are part of the item, that is noted where applicable.

ITEM...

An ellipsis indicates an item that may be used one or more times.

[ITEM...]

An ellipsis within brackets indicates an item that may be used zero or more times.

ITEM ITEM

Two or more juxtaposed items indicates that they should be used in the given order.

ITEM | ITEM

A pipe separates two or more alternatives from which only one item should be used.

ITEM & ITEM

An ampersand separates two or more items that may be used in any order.

Juxtaposition has precedence over the pipe or ampersand.

In the following example, you would select one of the keyword and vector pairs.

For that last pair, the keyword itself is optional.

rgb 3D\_VECTOR | rgbf 4D\_VECTOR | rgbt 4D\_VECTOR | [rgbft] 5D\_VECTOR

Some item names are simply descriptive in nature.

An indication of the item's type is given by a prefix on the item name, as follows:

F\_

A FLOAT item

I\_

An INT item

V\_

A VECTOR item

V4\_

A 4-D VECTOR item

# \*\*\*\*\*\*\*\*\*\* LEXICAL SPECIFICATION\*\*\*\*\*\*\*\*\*\*\*\*

FLOAT:

NUMERIC\_TERM [SIGN NUMERIC\_TERM]...

SIGN:

+ | -

NUMERIC\_TERM:

NUMERIC\_FACTOR [MULT NUMERIC\_FACTOR]...

MULT:

\* | /

NUMERIC\_EXPRESSION:

FLOAT\_LITERAL | FLOAT\_IDENTIFIER | SIGN NUMERIC\_EXPRESSION | FLOAT\_FUNCTION |

FLOAT\_BUILT\_IN\_IDENT | ( FULL\_EXPRESSION ) | ! NUMERIC\_EXPRESSION |

VECTOR.DOT\_ITEM | FLOAT\_FUNCTION\_INVOCATION

FLOAT\_LITERAL:

[DIGIT...][.]DIGIT...[EXP[SIGN]DIGIT...]

DIGIT:

0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

EXP:

e | E

FLOAT\_FUNCTION:

abs (FLOAT) | acos (FLOAT) | acosh (FLOAT) | asc (STRING) | asin (FLOAT) |

asinh (FLOAT) | atan (FLOAT) | atanh (FLOAT) | atan2 (FLOAT,FLOAT) | ceil (FLOAT) |

cos (FLOAT) | cosh (FLOAT) | defined (IDENTIFIER) | degrees (FLOAT) |

dimensions (ARRAY\_IDENTIFIER) | dimension\_size (ARRAY\_IDENTIFIER,INT) |

div (FLOAT,FLOAT) | exp (FLOAT) | file\_exists (STRING) | floor (FLOAT) |

int (FLOAT) | inside (SOLID\_OBJECT\_IDENT, VECTOR) | ln (FLOAT) | log (FLOAT) |

max (FLOAT,FLOAT[,FLOAT]...) | min (FLOAT,FLOAT[,FLOAT]...) | mod (FLOAT,FLOAT) |

pow (FLOAT,FLOAT) | radians (FLOAT) | rand (FLOAT) | seed (FLOAT) |

select (FLOAT,FLOAT,FLOAT,[FLOAT]) | sin (FLOAT) | sinh (FLOAT) | sqrt (FLOAT) |

strcmp (STRING,STRING) | strlen (STRING) | tan (FLOAT) | tanh (FLOAT) |

val (STRING) | vdot (VECTOR,VECTOR) | vlength (VECTOR)

FLOAT\_BUILT\_IN\_IDENT:

BOOLEAN\_KEYWORD | clock | clock\_delta | clock\_on | final\_clock | final\_frame |

frame\_number | image\_height | image\_width | initial\_clock | initial\_frame | pi | version

BOOLEAN\_KEYWORD:

true | yes | on | false | no | off

FULL\_EXPRESSION:

LOGICAL\_EXPRESSION [? FULL\_EXPRESSION : FULL\_EXPRESSION]

LOGICAL\_EXPRESSION:

REL\_TERM [LOGICAL\_OPERATOR REL\_TERM]...

LOGICAL\_OPERATOR:

& | |

REL\_TERM:

FLOAT [REL\_OPERATOR FLOAT]...

REL\_OPERATOR:

< | <= | = | >= | > | !=

DOT\_ITEM:

x | y | z | t | u | v | red | green | blue | filter | transmit | gray

INT:

FLOAT

Any fractional part is discarded.

BOOL:

BOOLEAN\_KEYWORD | LOGICAL\_EXPRESSION

VECTOR:

VECTOR\_TERM [SIGN VECTOR\_TERM]...

VECTOR\_TERM:

VECTOR\_EXPRESSION [MULT VECTOR\_EXPRESSION]...

VECTOR\_EXPRESSION:

VECTOR\_LITERAL | VECTOR\_IDENTIFIER | SIGN VECTOR\_EXPRESSION | VECTOR\_FUNCTION |

VECTOR\_BUILT\_IN\_IDENT | ! VECTOR\_EXPRESSION | FLOAT | VECTOR\_FUNCTION\_INVOCATION |

COLOR\_FUNCTION\_INVOCATION | SPLINE\_INVOCATION

VECTOR\_LITERAL:

< FLOAT, FLOAT [, FLOAT [, FLOAT [, FLOAT ]]] >

VECTOR\_FUNCTION:

min\_extent (OBJECT\_IDENTIFIER) | max\_extent (OBJECT\_IDENTIFIER) |

trace (OBJECT\_IDENTIFIER,VECTOR,VECTOR[,VECTOR\_IDENTIFIER]) |

vaxis\_rotate (VECTOR,VECTOR,FLOAT) | vcross (VECTOR,VECTOR) |

vrotate (VECTOR,VECTOR) | vnormalize (VECTOR) | vturbulence (FLOAT,FLOAT,FLOAT,VECTOR)

VECTOR\_BUILT\_IN\_IDENT:

x | y | z | t | u | v

COLOR:

[color] COLOR\_BODY | colour COLOR\_BODY

COLOR\_BODY:

COLOR\_VECTOR | COLOR\_KEYWORD\_GROUP | COLOR\_IDENTIFIER

COLOR\_VECTOR:

rgb 3D\_VECTOR | rgbf 4D\_VECTOR | rgbt 4D\_VECTOR | [rgbft] 5D\_VECTOR

COLOR\_KEYWORD\_GROUP:

[COLOR\_IDENTIFIER] COLOR\_KEYWORD\_ITEMS

COLOR\_KEYWORD\_ITEMS:

[red FLOAT] & [green FLOAT] & [blue FLOAT] & [filter FLOAT] & [transmit FLOAT]

USER\_FUNCTION:

FLOAT\_USER\_FUNCTION | VECTOR\_USER\_FUNCTION | COLOR\_USER\_FUNCTION

FLOAT\_USER\_FUNCTION:

function { FN\_FLOAT } | function (IDENT\_LIST) { FN\_FLOAT } |

function { pattern { PATTERN [PATTERN\_MODIFIERS]} }

IDENT\_LIST:

IDENT\_ITEM [, IDENT\_LIST]

The maximum number of parameter identifiers is 56. An identifier may not be repeated in the list.

IDENT\_ITEM:

x | y | z | u | v | PARAM\_IDENTIFIER

PATTERN:

MAP\_PATTERN | brick [BRICK\_ITEM] | checker | hexagon | object { LIST\_OBJECT }

VECTOR\_USER\_FUNCTION:

function { SPECIAL\_VECTOR\_FUNCTION }

SPECIAL\_VECTOR\_FUNCTION:

TRANSFORM | SPLINE

COLOR\_USER\_FUNCTION:

function { PIGMENT }

FN\_FLOAT

LOGIC\_AND [OR LOGIC\_AND]

OR:

|

LOGIC\_AND:

REL\_TERM [AND REL\_TERM]

AND:

&

REL\_TERM:

TERM [REL\_OPERATOR TERM]

REL\_OPERATOR:

< | <= | = | >= | > | !=

TERM:

FACTOR [SIGN FACTOR]

SIGN:

+ | -

FACTOR:

EXPRESSION [MULT EXPRESSION]

MULT:

\* | /

EXPRESSION:

FLOAT\_LITERAL | FLOAT\_IDENTIFIER | FN\_FLOAT\_FUNCTION | FLOAT\_BUILT\_IN\_IDENT |

( FN\_FLOAT ) | IDENT\_ITEM | SIGN EXPRESSION | VECTOR\_FUNCTION\_INVOCATION.FN\_DOT\_ITEM |

COLOR\_FUNCTION\_INVOCATION.FN\_DOT\_ITEM | FLOAT\_FUNCTION\_INVOCATION

FN\_DOT\_ITEM:

DOT\_ITEM | hf

FN\_FLOAT\_FUNCTION:

abs (FN\_FLOAT) | acos (FN\_FLOAT) | acosh (FN\_FLOAT) | asin (FN\_FLOAT) |

asinh (FN\_FLOAT) | atan (FN\_FLOAT) | atanh (FN\_FLOAT) | atan2 (FN\_FLOAT,FN\_FLOAT) |

ceil (FN\_FLOAT) | cos (FN\_FLOAT) | cosh (FN\_FLOAT) | degrees (FN\_FLOAT) |

exp (FN\_FLOAT) | floor (FN\_FLOAT) | int (FN\_FLOAT) | ln (FN\_FLOAT) | log (FN\_FLOAT) |

max (FN\_FLOAT,FN\_FLOAT[,FN\_FLOAT]...) | min (FN\_FLOAT,FN\_FLOAT[,FN\_FLOAT]...) |

mod (FN\_FLOAT,FN\_FLOAT) | pow (FN\_FLOAT,FN\_FLOAT) |

prod (IDENTIFIER, FN\_FLOAT, FN\_FLOAT, FN\_FLOAT) | radians (FN\_FLOAT) | sin (FN\_FLOAT) |

sinh (FN\_FLOAT) | sqrt (FN\_FLOAT) | sum (IDENTIFIER, FN\_FLOAT, FN\_FLOAT, FN\_FLOAT) |

tan (FN\_FLOAT) | tanh (FN\_FLOAT) | select (FN\_FLOAT,FN\_FLOAT,FN\_FLOAT [,FN\_FLOAT])

USER\_FUNCTION\_DECLARATION:

#declare FLOAT\_FUNCTION\_IDENTIFIER = FLOAT\_USER\_FUNCTION |

#local FLOAT\_FUNCTION\_IDENTIFIER = FLOAT\_USER\_FUNCTION |

#declare VECTOR\_FUNCTION\_IDENTIFIER = VECTOR\_USER\_FUNCTION |

#local VECTOR\_FUNCTION\_IDENTIFIER = VECTOR\_USER\_FUNCTION |

#declare COLOR\_FUNCTION\_IDENTIFIER = COLOR\_USER\_FUNCTION |

#local COLOR\_FUNCTION\_IDENTIFIER = COLOR\_USER\_FUNCTION

FLOAT\_FUNCTION\_INVOCATION:

FLOAT\_FUNCTION\_IDENTIFIER (FN\_PARAM\_LIST)

VECTOR\_FUNCTION\_INVOCATION:

VECTOR\_FUNCTION\_IDENTIFIER (FN\_PARAM\_LIST)

COLOR\_FUNCTION\_INVOCATION:

COLOR\_FUNCTION\_IDENTIFIER (FN\_PARAM\_LIST)

FN\_PARAM\_LIST:

FN\_PARAM\_ITEM [, FN\_PARAM\_LIST]

FN\_PARAM\_ITEM:

x | y | z | u | v | FLOAT

STRING:

STRING\_FUNCTION | STRING\_IDENTIFIER | STRING\_LITERAL

STRING\_FUNCTION:

chr (INT) | concat (STRING,STRING[,STRING]...) | str (FLOAT,INT,INT) |

strlwr (STRING) | strupr (STRING) | substr (STRING,INT,INT) | vstr (INT,VECTOR,STRING,INT,INT)

STRING\_LITERAL:

QUOTE [CHARACTER...] QUOTE

Limited to 256 characters.

QUOTE:

"

CHARACTER:

Any ASCII or Unicode character, depending on the charset setting in global\_settings.

The following escape sequences might be useful when writing to files or message streams:

\a - alarm

\b - backspace

\f - form feed

\n - new line

\r - carriage return

\t - horizontal tab

\uNNNN - unicode character four-digit code

\v - vertical tab

\\ - backslash

\' - single quote

\" - double quote

ARRAY\_DECLARATION:

#declare ARRAY\_IDENTIFIER = array DIMENSION... [ARRAY\_INITIALIZER] |

#local ARRAY\_IDENTIFIER = array DIMENSION... [ARRAY\_INITIALIZER]

Limited to five dimensions.

DIMENSION:

[ INT ]

The brackets here are part of the dimension specification. The integer must be greater than zero.

ARRAY\_INITIALIZER:

{ ARRAY\_INITIALIZER [, ARRAY\_INITIALIZER]... } |

{ RVALUE [, RVALUE]... }

ARRAY\_ELEMENT\_ASSIGNMENT:

#declare ARRAY\_REFERENCE =RVALUE [;] |

#local ARRAY\_REFERENCE = RVALUE [;]

The semicolon is required for a FLOAT, VECTOR or COLOR assignment.

ARRAY\_REFERENCE:

ARRAY\_IDENTIFIER ELEMENT...

ELEMENT:

[ INT ]

The brackets here are part of the element specification.

SPLINE:

spline { SPLINE\_ITEMS }

SPLINE\_ITEMS:

[SPLINE\_TYPE] PATH\_LIST | SPLINE\_IDENTIFIER [SPLINE\_TYPE][PATH\_LIST]

SPLINE\_TYPE:

linear\_spline | quadratic\_spline | cubic\_spline | natural\_spline

PATH\_LIST:

FLOAT, VECTOR [[,] PATH\_LIST]

SPLINE\_INVOCATION:

SPLINE\_IDENTIFIER ( FLOAT [, SPLINE\_TYPE] )

# \*\*\*\*\*\*\*\*\*\*\* GRAMMAR SPECIFICATION \*\*\*\*\*\*\*\*\*\*\*

SCENE:

SCENE\_ITEM...

SCENE\_ITEM:

LANGUAGE\_DIRECTIVE | CAMERA | LIGHT | OBJECT | ATMOSPHERIC\_EFFECT | GLOBAL\_SETTINGS

LANGUAGE\_DIRECTIVE:

INCLUDE\_DIRECTIVE | IDENTIFIER\_DECLARATION | UNDEF\_DIRECTIVE | FOPEN\_DIRECTIVE |

FCLOSE\_DIRECTIVE | READ\_DIRECTIVE | WRITE\_DIRECTIVE | DEFAULT\_DIRECTIVE |

VERSION\_DIRECTIVE | IF\_DIRECTIVE | IFDEF\_DIRECTIVE | IFNDEF\_DIRECTIVE |

SWITCH\_DIRECTIVE | WHILE\_DIRECTIVE | TEXT\_STREAM\_DIRECTIVE | MACRO\_DEFINITION

INCLUDE\_DIRECTIVE:

#include FILE\_NAME

FILE\_NAME:

STRING

IDENTIFIER\_DECLARATION:

#declare IDENTIFIER = RVALUE [;] |

#local IDENTIFIER = RVALUE [;]

RVALUE:

FLOAT | VECTOR | COLOR | USER\_FUNCTION | STRING | ARRAY\_REFERENCE | SPLINE | TRANSFORM | CAMERA | LIGHT | OBJECT | MATERIAL | INTERIOR | TEXTURE | TEXTURE\_MAP | PIGMENT | COLOR\_MAP | PIGMENT\_MAP | NORMAL | SLOPE\_MAP | NORMAL\_MAP | FINISH | MEDIA | DENSITY | DENSITY\_MAP | FOG | RAINBOW | SKY\_SPHERE

UNDEF\_DIRECTIVE:

#undef IDENTIFIER

FOPEN\_DIRECTIVE:

#fopen FILE\_HANDLE\_IDENTIFIER FILE\_NAME OPEN\_TYPE

OPEN\_TYPE:

read | write | append

FCLOSE\_DIRECTIVE:

#fclose FILE\_HANDLE\_IDENTIFIER

READ\_DIRECTIVE:

#read ( FILE\_HANDLE\_IDENTIFIER, DATA\_IDENTIFIER [, DATA\_IDENTIFIER]... )

DATA\_IDENTIFIER:

UNDECLARED\_IDENTIFIER | FLOAT\_IDENTIFIER | VECTOR\_IDENTIFIER | STRING\_IDENTIFIER | ARRAY\_REFERENCE

WRITE\_DIRECTIVE:

#write ( FILE\_HANDLE\_IDENTIFIER, DATA\_ITEM [, DATA\_ITEM]... )

DATA\_ITEM:

FLOAT | VECTOR | STRING

TRANSFORMATION:

rotate VECTOR | scale VECTOR | translate VECTOR | TRANSFORM | MATRIX

TRANSFORM:

transform TRANSFORM\_IDENTIFIER | transform { [TRANSFORM\_ITEM...] }

TRANSFORM\_ITEM:

TRANSFORM\_IDENTIFIER | TRANSFORMATION | inverse

MATRIX:

matrix < F\_VAL00, F\_VAL01, F\_VAL02, F\_VAL10, F\_VAL11, F\_VAL12, F\_VAL20,

F\_VAL21, F\_VAL22, F\_VAL30, F\_VAL31, F\_VAL32 >

CAMERA:

camera { [CAMERA\_TYPE] [CAMERA\_ITEMS] [CAMERA\_MODIFIERS] } |

camera { CAMERA\_IDENTIFIER [TANSFORMATIONS ...] }

CAMERA\_TYPE:

perspective | orthographic | fisheye | ultra\_wide\_angle | omnimax | panoramic |

spherical | cylinder CYLINDER\_TYPE

CYLINDER\_TYPE:

1 | 2 | 3 | 4

CAMERA\_ITEMS:

[location VECTOR] & [right VECTOR] & [up VECTOR] & [direction VECTOR] & [sky VECTOR]

CAMERA\_MODIFIERS:

[angle [angle F\_HORIZONTAL] [,F\_VERTICAL]] & [look\_at VECTOR] & [FOCAL\_BLUR] & [NORMAL] & [TRANSFORMATION...]

FOCAL\_BLUR:

aperture FLOAT & blur\_samples INT & [focal\_point VECTOR] & [confidence FLOAT] & [variance FLOAT]

LIGHT:

LIGHT\_SOURCE | LIGHT\_GROUP

LIGHT\_SOURCE:

light\_source { V\_LOCATION, COLOR [LIGHT\_SOURCE\_ITEMS] }

LIGHT\_SOURCE\_ITEMS:

[LIGHT\_TYPE] & [AREA\_LIGHT\_ITEMS] & [LIGHT\_MODIFIERS]

LIGHT\_TYPE:

spotlight [SPOTLIGHT\_ITEMS] | cylinder [SPOTLIGHT\_ITEMS]

SPOTLIGHT\_ITEMS:

[radius FLOAT] & [falloff FLOAT] & [tightness FLOAT] & [point\_at VECTOR]

AREA\_LIGHT\_ITEMS:

area\_light V\_AXIS1, V\_AXIS2, I\_SIZE1, I\_SIZE2 [AREA\_LIGHT\_MODIFIERS]

AREA\_LIGHT\_MODIFIERS:

[adaptive INT] & [jitter] & [circular] & [orient]

LIGHT\_MODIFIERS:

[LIGHT\_PHOTONS] & [looks\_like { OBJECT }] & [TRANSFORMATION...] &

[fade\_distance FLOAT] & [fade\_power FLOAT] & [media\_attenuation [BOOL]] &

[media\_interaction [BOOL]] & [shadowless] & [projected\_through { OBJECT\_IDENTIFIER }] &

[parallel [point\_at VECTOR]]

LIGHT\_PHOTONS:

photons { LIGHT\_PHOTON\_ITEMS }

LIGHT\_PHOTON\_ITEMS:

[refraction BOOL] & [reflection BOOL] & [area\_light]

LIGHT\_GROUP:

light\_group { LIGHT\_GROUP\_ITEM... [LIGHT\_GROUP\_MODIFIERS] }

LIGHT\_GROUP\_ITEM:

LIGHT\_SOURCE | OBJECT | LIGHT\_GROUP

LIGHT\_GROUP\_MODIFIERS:

[global\_lights BOOL] & [TRANSFORMATION...]

OBJECT:

FINITE\_SOLID\_OBJECT | FINITE\_PATCH\_OBJECT | INFINITE\_SOLID\_OBJECT | ISOSURFACE |

PARAMETRIC | CSG\_OBJECT | OBJECT\_STATEMENT

OBJECT\_STATEMENT:

object { OBJECT\_IDENTIFIER [OBJECT\_MODIFIERS] }

OBJECT\_MODIFIERS:

[OBJECT\_PHOTONS] & [CLIPPED\_BY] & [BOUNDED\_BY] & [MATERIAL] & [INTERIOR] &

[INTERIOR\_TEXTURE] & [TEXTURE] & [PIGMENT] & [NORMAL] & [FINISH] &

[TRANSFORMATION...] & [no\_shadow] & [no\_image[BOOL]] & [no\_reflection{BOOL]] &

[inverse] & [double\_illuminate[BOOL]] & [hollow [BOOL]]

OBJECT\_PHOTONS:

photons { OBJECT\_PHOTON\_ITEMS }

OBJECT\_PHOTON\_ITEMS:

[target [F\_SPACING\_MULT]] & [refraction BOOL] & [reflection BOOL] & [collect BOOL] & [pass\_through [BOOL]]

UV\_MAPPING:

uv\_mapping PIGMENT | pigment { uv\_mapping PIGMENT\_BODY } |

uv\_mapping NORMAL | normal { uv\_mapping NORMAL\_BODY } |

uv\_mapping TEXTURE | texture { uv\_mapping TEXTURE\_BODY }

MATERIAL:

material { [MATERIAL\_IDENTIFIER] [MATERIAL\_ITEM ...] }

MATERIAL\_ITEMS:

TEXTURE | INTERIOR\_TEXTURE | INTERIOR | TRANSFORMATION

INTERIOR:

interior { [INTERIOR\_IDENTIFIER] [INTERIOR\_ITEMS] }

INTERIOR\_ITEMS:

[ior FLOAT] & [dispersion FLOAT] & [dispersion\_samples INT] & [caustics FLOAT] &

[fade\_distance FLOAT] & [fade\_power FLOAT] & [fade\_color COLOR] & [MEDIA...]

INTERIOR\_TEXTURE:

interior\_texture { TEXTURE\_BODY }

CLIPPED\_BY:

clipped\_by { UNTEXTURED\_SOLID\_OBJECT... } |

clipped\_by { bounded\_by }

UNTEXTURED\_SOLID\_OBJECT:

FINITE\_SOLID\_OBJECT | INFINITE\_SOLID\_OBJECT

Note, neither with a texture applied.

BOUNDED\_BY:

bounded\_by { UNTEXTURED\_SOLID\_OBJECT... } |

bounded\_by { clipped\_by }

ATMOSPHERIC\_EFFECT:

MEDIA | BACKGROUND | FOG | SKY\_SPHERE | RAINBOW

GLOBAL\_SETTINGS:

global\_settings { GLOBAL\_SETTING\_ITEMS }

GLOBAL\_SETTING\_ITEMS:

[adc\_bailout FLOAT] & [ambient\_light COLOR] & [assumed\_gamma FLOAT] &

[hf\_gray\_16 [BOOL]] & [irid\_wavelength COLOR] & [charset GLOBAL\_CHARSET] &

[max\_intersections INT] & [max\_trace\_level INT] & [number\_of\_waves INT] &

[noise\_generator NG\_TYPE] & [RADIOSITY] & [PHOTONS]

GLOBAL\_CHARSET:

ascii | utf8 | sys

NG\_TYPE:

1 | 2 | 3

FINITE\_SOLID\_OBJECT:

BLOB | BOX | CONE | CYLINDER | HEIGHT\_FIELD | JULIA\_FRACTAL | LATHE |

PRISM | SPHERE | SPHERE\_SWEEP | SUPERELLIPSOID | SOR | TEXT | TORUS

BLOB:

blob { [threshold FLOAT] BLOB\_ITEM... [BLOB\_MODIFIERS] }

BLOB\_ITEM:

sphere { V\_CENTER, F\_RADIUS, [strength] F\_STRENGTH [COMPONENT\_MODIFIERS] } |

cylinder { V\_END1, V\_END2, F\_RADIUS, [strength] F\_STRENGTH [COMPONENT\_MODIFIERS] }

COMPONENT\_MODIFIERS:

[TEXTURE] & [PIGMENT] & [NORMAL] & [FINISH] & [TRANSFORMATION...]

BLOB\_MODIFIERS:

[hierarchy [BOOL]] & [sturm [BOOL]] & [OBJECT\_MODIFIERS]

BOX:

box { V\_CORNER1, V\_CORNER2 [BOX\_MODIFIERS] }

BOX\_MODIFIERS:

[UV\_MAPPING] & [OBJECT\_MODIFIERS]

CONE:

cone { V\_BASE\_CENTER, F\_BASE\_RADIUS, V\_CAP\_CENTER, F\_CAP\_RADIUS [open] [OBJECT\_MODIFIERS] }

CYLINDER:

cylinder { V\_BASE\_CENTER, V\_CAP\_CENTER, F\_RADIUS [open] [OBJECT\_MODIFIERS] }

HEIGHT\_FIELD:

height\_field { HF\_IMAGE [HF\_MODIFIERS] }

HF\_IMAGE:

FUNCTION\_IMAGE | [HF\_TYPE] FILE\_NAME

HF\_TYPE:

gif | tga | pot | png | pgm | ppm | jpeg | tiff | sys

HF\_MODIFIERS:

[hierarchy [BOOL]] & [smooth] & [water\_level FLOAT] & [OBJECT\_MODIFIERS]

JULIA\_FRACTAL:

julia\_fractal { 4D\_VECTOR [JF\_ITEMS] [OBJECT\_MODIFIERS] }

JF\_ITEMS:

[ALGEBRA\_ITEM] & [max\_iteration INT] & [precision FLOAT] & [slice V4\_NORMAL, F\_DISTANCE]

ALGEBRA\_ITEM:

quaternion [QUATER\_FUNCTION] | hypercomplex [HYPER\_FUNCTION]

QUATER\_FUNCTION:

sqr | cube

HYPER\_FUNCTION:

sqr | cube | exp | reciprocal | sin | asin | sinh | asinh | cos |

acos | cosh | acosh | tan | atan | tanh | atanh | ln | pwr (FLOAT,FLOAT)

LATHE:

lathe { [LATHE\_SPLINE\_TYPE] I\_NUM\_POINTS, POINT\_LIST [LATHE\_MODIFIERS] }

LATHE\_SPLINE\_TYPE:

linear\_spline | quadratic\_spline | cubic\_spline | bezier\_spline

POINT\_LIST:

2D\_VECTOR [, 2D\_VECTOR]...

The quantity of 2D\_VECTORs is specified by the I\_NUM\_POINTS value.

LATHE\_MODIFIERS:

[sturm [BOOL]] & [UV\_MAPPING] & [OBJECT\_MODIFIERS]

PRISM:

prism { [PRISM\_ITEMS] F\_HEIGHT1, F\_HEIGHT2, I\_NUM\_POINTS, POINT\_LIST [open] [PRISM\_MODIFIERS] }

PRISM\_ITEMS:

[PRISM\_SPLINE\_TYPE] & [PRISM\_SWEEP\_TYPE]

PRISM\_SPLINE\_TYPE:

linear\_spline | quadratic\_spline | cubic\_spline | bezier\_spline

PRISM\_SWEEP\_TYPE:

linear\_sweep | conic\_sweep

PRISM\_MODIFIERS:

[sturm [BOOL]] & [OBJECT\_MODIFIERS]

SPHERE:

sphere { V\_CENTER, F\_RADIUS [SPHERE\_MODIFIERS] }

SPHERE\_MODIFIERS:

[UV\_MAPPING] & [OBJECT\_MODIFIERS]

SPHERE\_SWEEP:

sphere\_sweep { SWEEP\_SPLINE\_TYPE I\_NUM\_SPHERES, SPHERE\_LIST [tolerance F\_DEPTH\_TOLERANCE] [OBJECT\_MODIFIERS] }

SWEEP\_SPLINE\_TYPE:

linear\_spline | b\_spline | cubic\_spline

SPHERE\_LIST:

V\_CENTER, F\_RADIUS [, SPHERE\_LIST]

The quantity of V\_CENTER, F\_RADIUS pairs is specified by the I\_NUM\_SPHERES value.

SUPERELLIPSOID:

superellipsoid { < FLOAT, FLOAT > [OBJECT\_MODIFIERS] }

SOR:

sor { I\_NUM\_POINTS, POINT\_LIST [open] [SOR\_MODIFIERS] }

SOR\_MODIFIERS:

[sturm [BOOL]] & [UV\_MAPPING] & [OBJECT\_MODIFIERS]

TEXT:

text { ttf FILE\_NAME STRING F\_THICKNESS, V\_OFFSET [OBJECT\_MODIFIERS] }

TORUS:

torus { F\_MAJOR\_RADIUS, F\_MINOR\_RADIUS [TORUS\_MODIFIERS] }

TORUS\_MODIFIERS:

[sturm [BOOL]] & [UV\_MAPPING] & [OBJECT\_MODIFIERS]

FINITE\_PATCH\_OBJECT:

BICUBIC\_PATCH | DISC | MESH | MESH2 | POLYGON | TRIANGLE | SMOOTH\_TRIANGLE

BICUBIC\_PATCH:

bicubic\_patch { PATCH\_ITEMS [PATCH\_UV\_VECTORS] CONTROL\_POINTS [BICUBIC\_PATCH\_MODIFIERS] }

PATCH\_ITEMS:

type PATCH\_TYPE & [u\_steps INT] & [v\_steps INT] & [flatness FLOAT]

PATCH\_TYPE:

0 | 1

PATCH\_UV\_VECTORS:

uv\_vectors V2\_CORNER1, V2\_CORNER2, V2\_CORNER3, V2\_CORNER4

CONTROL\_POINTS:

16 VECTORs, optionally separated by commas.

BICUBIC\_PATCH\_MODIFIERS:

[UV\_MAPPING] & [OBJECT\_MODIFIERS]

DISC:

Jump to SDL

disc { V\_CENTER, V\_NORMAL, F\_RADIUS [, F\_HOLE\_RADIUS] [OBJECT\_MODIFIERS] }

MESH:

Jump to SDL

mesh { MESH\_TRIANGLE... [MESH\_MODIFIERS] }

MESH\_TRIANGLE:

triangle { V\_CORNER1, V\_CORNER2, V\_CORNER3 [MESH\_UV\_VECTORS] [MESH\_TEXTURE] } |

smooth\_triangle { V\_CORNER1, V\_NORMAL1, V\_CORNER2, V\_NORMAL2, V\_CORNER3,

V\_NORMAL3 [MESH\_UV\_VECTORS] [MESH\_TEXTURE] }

MESH\_UV\_VECTORS:

uv\_vectors V2\_CORNER1, V2\_CORNER2, V2\_CORNER3

MESH\_TEXTURE:

texture { TEXTURE\_IDENTIFIER } |

texture\_list { TEXTURE\_IDENTIFIER TEXTURE\_IDENTIFIER TEXTURE\_IDENTIFIER }

MESH\_MODIFIERS:

[inside\_vector V\_DIRECTION] & [hierarchy [BOOL]] & [UV\_MAPPING] & [OBJECT\_MODIFIERS]

MESH2:

mesh2 { MESH2\_VECTORS [TEXTURE\_LIST] MESH2\_INDICES [MESH2\_MODIFIERS] }

MESH2\_VECTORS:

VERTEX\_VECTORS [NORMAL\_VECTORS] [UV\_VECTORS]

VERTEX\_VECTORS:

vertex\_vectors { I\_NUM\_VERTICES, VECTOR [, VECTOR]... }

NORMAL\_VECTORS:

normal\_vectors { I\_NUM\_NORMALS, VECTOR [, VECTOR]... }

UV\_VECTORS:

uv\_vectors { I\_NUM\_UV\_VECTORS, 2D\_VECTOR [, 2D\_VECTOR]... }

TEXTURE\_LIST:

texture\_list { I\_NUM\_TEXTURES, TEXTURE [, TEXTURE]... }

MESH2\_INDICES:

FACE\_INDICES [NORMAL\_INDICES] [UV\_INDICES]

FACE\_INDICES:

face\_indices { I\_NUM\_FACES, FACE\_INDICES\_ITEM [, FACE\_INDICES\_ITEM]... }

FACE\_INDICES\_ITEM:

VECTOR [, I\_TEXTURE\_INDEX [, I\_TEXTURE\_INDEX, I\_TEXTURE\_INDEX ]]

NORMAL\_INDICES:

normal\_indices { I\_NUM\_FACES, VECTOR [, VECTOR]... }

UV\_INDICES:

uv\_indices { I\_NUM\_FACES, VECTOR [, VECTOR]... }

MESH2\_MODIFIERS:

[inside\_vector V\_DIRECTION] & [UV\_MAPPING] & [OBJECT\_MODIFIERS]

POLYGON:

polygon { I\_NUM\_POINTS, V\_POINT [, V\_POINT]... [OBJECT\_MODIFIERS] }

The quantity of V\_POINTs is specified by the I\_NUM\_POINTS value.

TRIANGLE:

triangle { V\_CORNER1, V\_CORNER2, V\_CORNER3 [OBJECT\_MODIFIERS] }

SMOOTH\_TRIANGLE:

smooth\_triangle { V\_CORNER1, V\_NORMAL1, V\_CORNER2, V\_NORMAL2, V\_CORNER3, V\_NORMAL3 [OBJECT\_MODIFIERS] }

INFINITE\_SOLID\_OBJECT:

PLANE | POLY | CUBIC | QUARTIC | QUADRIC

PLANE:

plane { V\_NORMAL, F\_DISTANCE [OBJECT\_MODIFIERS] }

POLY:

poly { ORDER, < POLY\_COEFFICIENTS > [POLY\_MODIFIERS] }

ORDER:

An integer value between 2 and 15 inclusive.

POLY\_COEFFICIENTS:

A quantity n of FLOATs separated by commas, where n is ((ORDER+1)\*(ORDER+2)\*(ORDER+3))/6.

POLY\_MODIFIERS:

[sturm [BOOL]] & [OBJECT\_MODIFIERS]

CUBIC:

cubic { < CUBIC\_COEFFICIENTS > [POLY\_MODIFIERS] }

CUBIC\_COEFFICIENTS:

20 FLOATs separated by commas.

QUARTIC:

quartic { < QUARTIC\_COEFFICIENTS > [POLY\_MODIFIERS] }

QUARTIC\_COEFFICIENTS:

35 FLOATs separated by commas.

QUADRIC:

quadric { < FLOAT, FLOAT, FLOAT >, < FLOAT, FLOAT, FLOAT >, < FLOAT, FLOAT, FLOAT >, FLOAT [OBJECT\_MODIFIERS] }

ISOSURFACE:

isosurface { FLOAT\_USER\_FUNCTION [ISOSURFACE\_ITEMS] [OBJECT\_MODIFIERS] }

ISOSURFACE\_ITEMS:

[contained\_by { CONTAINER }] & [threshold FLOAT] & [accuracy FLOAT] &

[max\_gradient FLOAT [evaluate F\_MIN\_ESTIMATE, F\_MAX\_ESTIMATE, F\_ATTENUATION]] & [open] & [INTERSECTION\_LIMIT]

CONTAINER:

sphere { V\_CENTER, F\_RADIUS } | box { V\_CORNER1, V\_CORNER2 }

INTERSECTION\_LIMIT:

max\_trace INT | all\_intersections

PARAMETRIC:

parametric { FLOAT\_USER\_FUNCTION, FLOAT\_USER\_FUNCTION, FLOAT\_USER\_FUNCTION 2D\_VECTOR,

2D\_VECTOR [PARAMETRIC\_ITEMS] [UV\_MAPPING] & [OBJECT\_MODIFIERS] }

PARAMETRIC\_ITEMS:

[contained\_by { CONTAINER }] & [max\_gradient FLOAT] & [accuracy FLOAT] & [precompute I\_DEPTH, x, y, z]

CONTAINER:

sphere { V\_CENTER, F\_RADIUS } | box { V\_CORNER1, V\_CORNER2 }

CSG\_OBJECT:

UNION | INTERSECTION | DIFFERENCE | MERGE

UNION:

union { UNION\_OBJECT UNION\_OBJECT... [UNION\_MODIFIERS] }

UNION\_OBJECT:

OBJECT | LIGHT

UNION\_MODIFIERS:

[split\_union BOOL] & [OBJECT\_MODIFIERS]

INTERSECTION:

intersection { SOLID\_OBJECT SOLID\_OBJECT... [INTERSECTION\_MODIFIERS] }

SOLID\_OBJECT:

FINITE\_SOLID\_OBJECT | INFINITE\_SOLID\_OBJECT | ISOSURFACE | CSG\_OBJECT

INTERSECTION\_MODIFIERS:

[cutaway\_textures] & [OBJECT\_MODIFIERS]

DIFFERENCE:

difference { SOLID\_OBJECT SOLID\_OBJECT... [DIFFERENCE\_MODIFIERS] }

DIFFERENCE\_MODIFIERS:

[cutaway\_textures] & [OBJECT\_MODIFIERS]

MERGE:

merge { SOLID\_OBJECT SOLID\_OBJECT... [OBJECT\_MODIFIERS] }

PIGMENT:

pigment { PIGMENT\_BODY }

PIGMENT\_BODY:

[PIGMENT\_IDENTIFIER] [PIGMENT\_TYPE] [PIGMENT\_MODIFIERS]

PIGMENT\_TYPE:

COLOR | COLOR\_LIST\_PATTERN | PIGMENT\_LIST\_PATTERN | IMAGE\_MAP | MAP\_PATTERN [COLOR\_MAP] | MAP\_PATTERN PIGMENT\_MAP

COLOR\_LIST\_PATTERN:

brick [COLOR [, COLOR]] [BRICK\_ITEMS] |

checker [COLOR [, COLOR]] |

hexagon [COLOR [, COLOR [, COLOR]]] |

object { LIST\_OBJECT [COLOR [, COLOR]] }

PIGMENT\_LIST\_PATTERN:

brick PIGMENT, PIGMENT [BRICK\_ITEMS] |

checker PIGMENT, PIGMENT |

hexagon PIGMENT, PIGMENT, PIGMENT |

object { LIST\_OBJECT PIGMENT, PIGMENT }

IMAGE\_MAP:

image\_map {BITMAP\_IMAGE [IMAGE\_MAP\_MODIFIER...] [BITMAP\_MODIFIERS] }

IMAGE\_MAP\_MODIFIER:

filter I\_PALETTE, F\_AMOUNT | filter all F\_AMOUNT | transmit I\_PALETTE, F\_AMOUNT | transmit all F\_AMOUNT

COLOR\_MAP:

color\_map { COLOR\_MAP\_BODY } [BLEND\_MAP\_MODIFIERS] |

colour\_map { COLOR\_MAP\_BODY } [BLEND\_MAP\_MODIFIERS]

COLOR\_MAP\_BODY:

COLOR\_MAP\_IDENTIFIER | COLOR\_MAP\_ENTRY...

There may be from 2 to 256 map entries.

COLOR\_MAP\_ENTRY:

[ FLOAT COLOR ]

The brackets here are part of the map entry.

PIGMENT\_MAP:

pigment\_map { PIGMENT\_MAP\_BODY } [BLEND\_MAP\_MODIFIERS]

PIGMENT\_MAP\_BODY:

PIGMENT\_MAP\_IDENTIFIER | PIGMENT\_MAP\_ENTRY...

There may be from 2 to 256 map entries.

PIGMENT\_MAP\_ENTRY:

[ FLOAT PIGMENT\_BODY ]

The brackets here are part of the map entry.

PIGMENT\_MODIFIERS:

[QUICK\_COLOR] & [PATTERN\_MODIFIERS]

QUICK\_COLOR:

quick\_color COLOR | quick\_colour COLOR

FINISH:

finish { [FINISH\_IDENTIFIER] [FINISH\_ITEMS] }

FINISH\_ITEMS:

[ambient COLOR] & [diffuse FLOAT] & [brilliance FLOAT] & [PHONG] &

[SPECULAR] & [REFLECTION] & [IRID] & [crand FLOAT] & [conserve\_energy [BOOL]]

PHONG:

phong FLOAT & [phong\_size FLOAT] & [metallic [FLOAT]]

SPECULAR:

specular FLOAT & [roughness FLOAT] & [metallic [FLOAT]]

REFLECTION:

reflection COLOR [reflection\_exponent FLOAT] |

reflection { [COLOR,] COLOR [REFLECTION\_ITEMS] }

REFLECTION\_ITEMS:

[fresnel BOOL] & [falloff FLOAT] & [exponent FLOAT] & [metallic [FLOAT]]

Must also use interior {ior FLOAT} in the object when fresnel is used.

IRID:

irid { F\_AMOUNT [IRID\_ITEMS] }

IRID\_ITEMS:

[thickness FLOAT] & [turbulence FLOAT]