

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 [TRANSFORMATIONS ...] }
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]

```