MTL material format (Lightwave, OBJ)

Excerpt from FILE FORMATS, Version 4.2

October 1995

Documentation created by: Diane Ramey, Linda Rose, and Lisa Tyerman


```
5.  Material Library File (.mtl)

 Material library files contain one or more material definitions, each
of which includes the color, texture, and reflection map of individual
materials.  These are applied to the surfaces and vertices of objects.
Material files are stored in ASCII format and have the .mtl extension.

 An .mtl file differs from other Alias|Wavefront property files, such as
light and atmosphere files, in that it can contain more than one
material definition (other files contain the definition of only one
item).

 An .mtl file is typically organized as shown below.


        newmtl my_red
                Material color
                & illumination
                statements

                texture map
                statements

                reflection map
                statement

        newmtl my_blue
                Material color
                & illumination
                statements

                texture map
                statements

                reflection map
                statement

        newmtl my_green
                Material color
                & illumination
                statements
```

```
                texture map
                statements

                reflection map
                statement

        Figure 5-1.  Typical organization of .mtl file
```

Each material description in an .mtl file consists of the newmtl statement, which assigns a name to the material and designates the start of a material description. This statement is followed by the material color, texture map, and reflection map statements that describe the material. An .mtl file map contain many different material descriptions.

After you specify a new material with the "newmtl" statement, you can enter the statements that describe the materials in any order. However, when the Property Editor writes an .mtl file, it puts the statements in a system-assigned order. In this chapter, the statements are described in the system-assigned order.

Format

The following is a sample format for a material definition in an .mtl file:

```
 Material
 name
 statement:
                newmtl my_mtl

 Material
 color and
 illumination
 statements:
                Ka 0.0435 0.0435 0.0435
                Kd 0.1086 0.1086 0.1086
                Ks 0.0000 0.0000 0.0000
                Tf 0.9885 0.9885 0.9885
                illum 6
                d -halo 0.6600
                Ns 10.0000
                sharpness 60
                Ni 1.19713

 Texture
 map
 statements:
                map_Ka -s 1 1 1 -o 0 0 0 -mm 0 1 chrome.mpc
                map_Kd -s 1 1 1 -o 0 0 0 -mm 0 1 chrome.mpc
                map_Ks -s 1 1 1 -o 0 0 0 -mm 0 1 chrome.mpc
                map_Ns -s 1 1 1 -o 0 0 0 -mm 0 1 wisp.mps
                map_d -s 1 1 1 -o 0 0 0 -mm 0 1 wisp.mps
```

```
                   disp -s 1 1 .5 wisp.mps
                   decal -s 1 1 1 -o 0 0 0 -mm 0 1 sand.mps
                   bump -s 1 1 1 -o 0 0 0 -bm 1 sand.mpb

 Reflection
 map
 statement:
                   refl -type sphere -mm 0 1 clouds.mpc


 Material Name

 The material name statement assigns a name to the material description.

 Syntax
 The folowing syntax describes the material name statement.

        newmtl name

 Specifies the start of a material description and assigns a name to the
material.  An .mtl file must have one newmtl statement at the start of
each material description.

        "name" is the name of the material.  Names may be any length but
cannot include blanks.  Underscores may be used in material names.


 Material color and illumination

 The statements in this section specify color, transparency, and
reflectivity values.

 Syntax
 The following syntax describes the material color and illumination
statements that apply to all .mtl files.


        Ka r g b
        Ka spectral file.rfl factor
        Ka xyz x y z

 To specify the ambient reflectivity of the current material, you can
use the "Ka" statement, the "Ka spectral" statement, or the "Ka xyz"
statement.

 Tip    These statements are mutually exclusive.  They cannot be used
concurrently in the same material.

 Ka r g b

 The Ka statement specifies the ambient reflectivity using RGB values.

 "r g b" are the values for the red, green, and blue components of the
color.  The g and b arguments are optional.  If only r is specified,
then g, and b are assumed to be equal to r.  The r g b values are
normally in the range of 0.0 to 1.0.  Values outside this range increase
```

or decrease the relectivity accordingly.

Ka spectral file.rfl factor

The "Ka spectral" statement specifies the ambient reflectivity using a spectral curve.

"file.rfl" is the name of the .rfl file.
"factor" is an optional argument.
"factor" is a multiplier for the values in the .rfl file and defaults to 1.0, if not specified.

Ka xyz x y z

The "Ka xyz" statement specifies the ambient reflectivity using CIEXYZ values.

"x y z" are the values of the CIEXYZ color space. The y and z arguments are optional. If only x is specified, then y and z are assumed to be equal to x. The x y z values are normally in the range of 0 to 1. Values outside this range increase or decrease the reflectivity accordingly.

```
Kd r g b
Kd spectral file.rfl factor
Kd xyz x y z
```

To specify the diffuse reflectivity of the current material, you can use the "Kd" statement, the "Kd spectral" statement, or the "Kd xyz" statement.

Tip   These statements are mutually exclusive. They cannot be used concurrently in the same material.

Kd r g b

The Kd statement specifies the diffuse reflectivity using RGB values.

"r g b" are the values for the red, green, and blue components of the atmosphere. The g and b arguments are optional. If only r is specified, then g, and b are assumed to be equal to r. The r g b values are normally in the range of 0.0 to 1.0. Values outside this range increase or decrease the relectivity accordingly.

Kd spectral file.rfl factor

The "Kd spectral" statement specifies the diffuse reflectivity using a spectral curve.

"file.rfl" is the name of the .rfl file.
"factor" is an optional argument.
"factor" is a multiplier for the values in the .rfl file and defaults to 1.0, if not specified.

Kd xyz x y z

The "Kd xyz" statement specifies the diffuse reflectivity using CIEXYZ values.

"x y z" are the values of the CIEXYZ color space. The y and z arguments are optional. If only x is specified, then y and z are assumed to be equal to x. The x y z values are normally in the range of 0 to 1. Values outside this range increase or decrease the reflectivity accordingly.

```
Ks r g b
Ks spectral file.rfl factor
Ks xyz x y z
```

To specify the specular reflectivity of the current material, you can use the "Ks" statement, the "Ks spectral" statement, or the "Ks xyz" statement.

Tip These statements are mutually exclusive. They cannot be used concurrently in the same material.

Ks r g b

The Ks statement specifies the specular reflectivity using RGB values.

"r g b" are the values for the red, green, and blue components of the atmosphere. The g and b arguments are optional. If only r is specified, then g, and b are assumed to be equal to r. The r g b values are normally in the range of 0.0 to 1.0. Values outside this range increase or decrease the relectivity accordingly.

Ks spectral file.rfl factor

The "Ks spectral" statement specifies the specular reflectivity using a spectral curve.

"file.rfl" is the name of the .rfl file.
"factor" is an optional argument.
"factor" is a multiplier for the values in the .rfl file and defaults to 1.0, if not specified.

Ks xyz x y z

The "Ks xyz" statement specifies the specular reflectivity using CIEXYZ values.

"x y z" are the values of the CIEXYZ color space. The y and z arguments are optional. If only x is specified, then y and z are assumed to be equal to x. The x y z values are normally in the range of 0 to 1. Values outside this range increase or decrease the reflectivity accordingly.

```
Tf r g b
Tf spectral file.rfl factor
```

Tf xyz x y z

To specify the transmission filter of the current material, you can use the "Tf" statement, the "Tf spectral" statement, or the "Tf xyz" statement.

Any light passing through the object is filtered by the transmission filter, which only allows the specifiec colors to pass through. For example, Tf 0 1 0 allows all the green to pass through and filters out all the red and blue.

Tip These statements are mutually exclusive. They cannot be used concurrently in the same material.

Tf r g b

The Tf statement specifies the transmission filter using RGB values.

"r g b" are the values for the red, green, and blue components of the atmosphere. The g and b arguments are optional. If only r is specified, then g, and b are assumed to be equal to r. The r g b values are normally in the range of 0.0 to 1.0. Values outside this range increase or decrease the relectivity accordingly.

Tf spectral file.rfl factor

The "Tf spectral" statement specifies the transmission filterusing a spectral curve.

"file.rfl" is the name of the .rfl file.
"factor" is an optional argument.
"factor" is a multiplier for the values in the .rfl file and defaults to 1.0, if not specified.

Tf xyz x y z

The "Ks xyz" statement specifies the specular reflectivity using CIEXYZ values.

"x y z" are the values of the CIEXYZ color space. The y and z arguments are optional. If only x is specified, then y and z are assumed to be equal to x. The x y z values are normally in the range of 0 to 1. Values outside this range will increase or decrease the intensity of the light transmission accordingly.

illum illum_#

The "illum" statement specifies the illumination model to use in the material. Illumination models are mathematical equations that represent various material lighting and shading effects.

"illum_#"can be a number from 0 to 10. The illumination models are summarized below; for complete descriptions see "Illumination models" on page 5-30.

Illumination model	Properties that are turned on in the Property Editor
0	Color on and Ambient off
1	Color on and Ambient on
2	Highlight on
3	Reflection on and Ray trace on
4	Transparency: Glass on Reflection: Ray trace on
5	Reflection: Fresnel on and Ray trace on
6	Transparency: Refraction on Reflection: Fresnel off and Ray trace on
7	Transparency: Refraction on Reflection: Fresnel on and Ray trace on
8	Reflection on and Ray trace off
9	Transparency: Glass on Reflection: Ray trace off
10	Casts shadows onto invisible surfaces

d factor

Specifies the dissolve for the current material.

"factor" is the amount this material dissolves into the background. A factor of 1.0 is fully opaque. This is the default when a new material is created. A factor of 0.0 is fully dissolved (completely transparent).

Unlike a real transparent material, the dissolve does not depend upon material thickness nor does it have any spectral character. Dissolve works on all illumination models.

d -halo factor

Specifies that a dissolve is dependent on the surface orientation relative to the viewer. For example, a sphere with the following dissolve, d -halo 0.0, will be fully dissolved at its center and will appear gradually more opaque toward its edge.

"factor" is the minimum amount of dissolve applied to the material. The amount of dissolve will vary between 1.0 (fully opaque) and the specified "factor". The formula is:

dissolve = 1.0 - (N*v)(1.0-factor)

For a definition of terms, see "Illumination models" on page 5-30.

Ns exponent

Specifies the specular exponent for the current material. This defines the focus of the specular highlight.

"exponent" is the value for the specular exponent. A high exponent results in a tight, concentrated highlight. Ns values normally range

```
from 0 to 1000.


 sharpness value

 Specifies the sharpness of the reflections from the local reflection
map.  If a material does not have a local reflection map defined in its
material definition, sharpness will apply to the global reflection map
defined in PreView.

 "value" can be a number from 0 to 1000.  The default is 60.  A high
value results in a clear reflection of objects in the reflection map.

 Tip    Sharpness values greater than 100 map introduce aliasing effects
in flat surfaces that are viewed at a sharp angle


 Ni optical_density

 Specifies the optical density for the surface.  This is also known as
index of refraction.

 "optical_density" is the value for the optical density.  The values can
range from 0.001 to 10.  A value of 1.0 means that light does not bend
as it passes through an object.  Increasing the optical_density
increases the amount of bending.  Glass has an index of refraction of
about 1.5.  Values of less than 1.0 produce bizarre results and are not
recommended.


 Material texture map

 Texture map statements modify the material parameters of a surface by
associating an image or texture file with material parameters that can
be mapped.  By modifying existing parameters instead of replacing them,
texture maps provide great flexibility in changing the appearance of an
object's surface.

 Image files and texture files can be used interchangeably.  If you use
an image file, that file is converted to a texture in memory and is
discarded after rendering.

 Tip Using images instead of textures saves disk space and setup time,
however, it introduces a small computational cost at the beginning of a
render.

 The material parameters that can be modified by a texture map are:

 - Ka (color)
 - Kd (color)
 - Ks (color)
 - Ns (scalar)
 - d (scalar)

 In addition to the material parameters, the surface normal can be
modified.
```

Image file types

You can link any image file type that is currently supported. Supported image file types are listed in the chapter "About Image" in the "Advanced Visualizer User's Guide". You can also use the "im_info -a" command to list Image file types, among other things.

Texture file types

The texture file types you can use are:

- mip-mapped texture files (.mpc, .mps, .mpb)
- compiled procedural texture files (.cxc, .cxs, .cxb)

Mip-mapped texture files

Mip-mapped texture files are created from images using the Create Textures panel in the Director or the "texture2D" program. There are three types of texture files:

- color texture files (.mpc)
- scalar texture files (.mps)
- bump texture files (.mpb)

Color textures. Color texture files are designated by an extension of ".mpc" in the filename, such as "chrome.mpc". Color textures modify the material color as follows:

- Ka - material ambient is multiplied by the texture value
- Kd - material diffuse is multiplied by the texture value
- Ks - material specular is multiplied by the texture value

Scalar textures. Scalar texture files are designated by an extension of ".mps" in the filename, such as "wisp.mps". Scalar textures modify the material scalar values as follows:

- Ns - material specular exponent is multiplied by the texture value
- d - material dissolve is multiplied by the texture value
- decal - uses a scalar value to deform the surface of an object to create surface roughness

Bump textures. Bump texture files are designated by an extension of ".mpb" in the filename, such as "sand.mpb". Bump textures modify surface normals. The image used for a bump texture represents the topology or height of the surface relative to the average surface. Dark areas are depressions and light areas are high points. The effect is like embossing the surface with the texture.

Procedural texture files

Procedural texture files use mathematical formulas to calculate sample values of the texture. The procedural texture file is compiled, stored, and accessed by the Image program when rendering. for more information see chapter 9, "Procedural Texture Files (.cxc, .cxb. and .cxs)".

Syntax

The following syntax describes the texture map statements that apply to .mtl files. These statements can be used alone or with any combination of options. The options and their arguments are inserted between the keyword and the "filename".

map_Ka -options args filename

Specifies that a color texture file or a color procedural texture file is applied to the ambient reflectivity of the material. During rendering, the "map_Ka" value is multiplied by the "Ka" value.

"filename" is the name of a color texture file (.mpc), a color procedural texture file (.cxc), or an image file.

Tip To make sure that the texture retains its original look, use the .rfl file "ident" as the underlying material. This applies to the "map_Ka", "map_Kd", and "map_Ks" statements. For more information on .rfl files, see chapter 8, "Spectral Curve File (.rfl)".

The options for the "map_Ka" statement are listed below. These options are described in detail in "Options for texture map statements" on page 5-18.

```
-blendu on | off
-blendv on | off
-cc on | off
-clamp on | off
-mm base gain
-o u v w
-s u v w
-t u v w
-texres value
```

map_Kd -options args filename

Specifies that a color texture file or color procedural texture file is linked to the diffuse reflectivity of the material. During rendering, the map_Kd value is multiplied by the Kd value.

"filename" is the name of a color texture file (.mpc), a color procedural texture file (.cxc), or an image file.

The options for the map_Kd statement are listed below. These options are described in detail in "Options for texture map statements" on page 5-18.

```
-blendu on | off
-blendv on | off
-cc on | off
```

```
        -clamp on | off
        -mm base gain
        -o u v w
        -s u v w
        -t u v w
        -texres value
```

map_Ks -options args filename

Specifies that a color texture file or color procedural texture file is linked to the specular reflectivity of the material. During rendering, the map_Ks value is multiplied by the Ks value.

"filename" is the name of a color texture file (.mpc), a color procedural texture file (.cxc), or an image file.

The options for the map_Ks statement are listed below. These options are described in detail in "Options for texture map statements" on page 5-18.

```
        -blendu on | off
        -blendv on | off
        -cc on | off
        -clamp on | off
        -mm base gain
        -o u v w
        -s u v w
        -t u v w
        -texres value
```

map_Ns -options args filename

Specifies that a scalar texture file or scalar procedural texture file is linked to the specular exponent of the material. During rendering, the map_Ns value is multiplied by the Ns value.

"filename" is the name of a scalar texture file (.mps), a scalar procedural texture file (.cxs), or an image file.

The options for the map_Ns statement are listed below. These options are described in detail in "Options for texture map statements" on page 5-18.

```
        -blendu on | off
        -blendv on | off
        -clamp on | off
        -imfchan r | g | b | m | l | z
        -mm base gain
        -o u v w
        -s u v w
        -t u v w
        -texres value
```

map_d -options args filename

Specifies that a scalar texture file or scalar procedural texture file is linked to the dissolve of the material. During rendering, the map_d value is multiplied by the d value.

"filename" is the name of a scalar texture file (.mps), a scalar procedural texture file (.cxs), or an image file.

The options for the map_d statement are listed below. These options are described in detail in "Options for texture map statements" on page 5-18.

```
-blendu on | off
-blendv on | off
-clamp on | off
-imfchan r | g | b | m | l | z
-mm base gain
-o u v w
-s u v w
-t u v w
-texres value
```

map_aat on

Turns on anti-aliasing of textures in this material without anti-aliasing all textures in the scene.

If you wish to selectively anti-alias textures, first insert this statement in the material file. Then, when rendering with the Image panel, choose the anti-alias settings: "shadows", "reflections polygons", or "polygons only". If using Image from the command line, use the -aa or -os options. Do not use the -aat option.

Image will anti-alias all textures in materials with the map_aat on statement, using the oversampling level you choose when you run Image. Textures in other materials will not be oversampled.

You cannot set a different oversampling level individually for each material, nor can you anti-alias some textures in a material and not others. To anti-alias all textures in all materials, use the -aat option from the Image command line. If a material with "map_aat on" includes a reflection map, all textures in that reflection map will be anti-aliased as well.

You will not see the effects of map_aat in the Property Editor.

Tip Some .mpc textures map exhibit undesirable effects around the edges of smoothed objects. The "map_aat" statement will correct this.

decal -options args filename

Specifies that a scalar texture file or a scalar procedural texture file is used to selectively replace the material color with the texture color.

"filename" is the name of a scalar texture file (.mps), a scalar procedural texture file (.cxs), or an image file.

During rendering, the Ka, Kd, and Ks values and the map_Ka, map_Kd, and map_Ks values are blended according to the following formula:

```
result_color=tex_color(tv)*decal(tv)+mtl_color*(1.0-decal(tv))
```

where tv is the texture vertex.

"result_color" is the blended Ka, Kd, and Ks values.

The options for the decal statement are listed below. These options are described in detail in "Options for texture map statements" on page 5-18.

```
-blendu on | off
-blendv on | off
-clamp on | off
-imfchan r | g | b | m | l | z
-mm base gain
-o u v w
-s u v w
-t u v w
-texres value
```

disp -options args filename

Specifies that a scalar texture is used to deform the surface of an object, creating surface roughness.

"filename" is the name of a scalar texture file (.mps), a bump procedural texture file (.cxb), or an image file.

The options for the disp statement are listed below. These options are described in detail in "Options for texture map statements" on page 5-18.

```
-blendu on | off
-blendv on | off
-clamp on | off
-imfchan r | g | b | m | l | z
-mm base gain
-o u v w
-s u v w
-t u v w
-texres value
```

bump -options args filename

Specifies that a bump texture file or a bump procedural texture file is linked to the material.

"filename" is the name of a bump texture file (.mpb), a bump procedural texture file (.cxb), or an image file.

The options for the bump statement are listed below. These options are described in detail in "Options for texture map statements" on page 5-18.

```
-bm mult
-clamp on | off
-blendu on | off
-blendv on | off
-imfchan r | g | b | m | l | z
-mm base gain
-o u v w
-s u v w
-t u v w
-texres value
```

Options for texture map statements

The following options and arguments can be used to modify the texture map statements.

-blenu on | off

The -blendu option turns texture blending in the horizontal direction (u direction) on or off. The default is on.

-blenv on | off

The -blendv option turns texture blending in the vertical direction (v direction) on or off. The default is on.

-bm mult

The -bm option specifies a bump multiplier. You can use it only with the "bump" statement. Values stored with the texture or procedural texture file are multiplied by this value before they are applied to the surface.

"mult" is the value for the bump multiplier. It can be positive or negative. Extreme bump multipliers may cause odd visual results because only the surface normal is perturbed and the surface position does not change. For best results, use values between 0 and 1.

-boost value

The -boost option increases the sharpness, or clarity, of mip-mapped texture files -- that is, color (.mpc), scalar (.mps), and bump (.mpb) files. If you render animations with boost, you may experience some texture crawling. The effects of boost are seen when you render in Image or test render in Model or PreView; they aren't as noticeable in Property Editor.

"value" is any non-negative floating point value representing the degree of increased clarity; the greater the value, the greater the clarity. You should start with a boost value of no more than 1 or 2 and increase the value as needed. Note that larger values have more potential to introduce texture crawling when animated.

-cc on | off

The -cc option turns on color correction for the texture. You can use it only with the color map statements: map_Ka, map_Kd, and map_Ks.

-clamp on | off

The -clamp option turns clamping on or off. When clamping is on, textures are restricted to 0-1 in the uvw range. The default is off.

When clamping is turned on, one copy of the texture is mapped onto the surface, rather than repeating copies of the original texture across the surface of a polygon, which is the default. Outside of the origin texture, the underlying material is unchanged.

A postage stamp on an envelope or a label on a can of soup is an example of a texture with clamping turned on. A tile floor or a sidewalk is an example of a texture with clamping turned off.

Two-dimensional textures are clamped in the u and v dimensions; 3D procedural textures are clamped in the u, v, and w dimensions.

-imfchan r | g | b | m | l | z

The -imfchan option specifies the channel used to create a scalar or bump texture. Scalar textures are applied to:

transparency
specular exponent
decal
displacement

The channel choices are:

r specifies the red channel.
g specifies the green channel.
b specifies the blue channel.
m specifies the matte channel.
l specifies the luminance channel.
z specifies the z-depth channel.

The default for bump and scalar textures is "l" (luminance), unless you are building a decal. In that case, the default is "m" (matte).

-mm base gain

The -mm option modifies the range over which scalar or color texture values may vary. This has an effect only during rendering and does not change the file.

"base" adds a base value to the texture values. A positive value makes everything brighter; a negative value makes everything dimmer. The default is 0; the range is unlimited.

"gain" expands the range of the texture values. Increasing the number increases the contrast. The default is 1; the range is unlimited.

-o u v w

The -o option offsets the position of the texture map on the surface by shifting the position of the map origin. The default is 0, 0, 0.

"u" is the value for the horizontal direction of the texture

"v" is an optional argument.
"v" is the value for the vertical direction of the texture.

"w" is an optional argument.
"w" is the value used for the depth of a 3D texture.

-s u v w

The -s option scales the size of the texture pattern on the textured surface by expanding or shrinking the pattern. The default is 1, 1, 1.

"u" is the value for the horizontal direction of the texture

"v" is an optional argument.
"v" is the value for the vertical direction of the texture.

"w" is an optional argument.
"w" is a value used for the depth of a 3D texture.
"w" is a value used for the amount of tessellation of the displacement map.

-t u v w

The -t option turns on turbulence for textures. Adding turbulence to a texture along a specified direction adds variance to the original image and allows a simple image to be repeated over a larger area without noticeable tiling effects.

turbulence also lets you use a 2D image as if it were a solid texture, similar to 3D procedural textures like marble and granite.

"u" is the value for the horizontal direction of the texture turbulence.

"v" is an optional argument.
"v" is the value for the vertical direction of the texture turbulence.

"w" is an optional argument.
"w" is a value used for the depth of the texture turbulence.

By default, the turbulence for every texture map used in a material is uvw = (0,0,0). This means that no turbulence will be applied and the 2D texture will behave normally.

Only when you raise the turbulence values above zero will you see the effects of turbulence.

-texres resolution

The -texres option specifies the resolution of texture created when an image is used. The default texture size is the largest power of two that does not exceed the original image size.

If the source image is an exact power of 2, the texture cannot be built any larger. If the source image size is not an exact power of 2, you can specify that the texture be built at the next power of 2 greater than the source image size.

The original image should be square, otherwise, it will be scaled to fit the closest square size that is not larger than the original. Scaling reduces sharpness.

Material reflection map

A reflection map is an environment that simulates reflections in specified objects. The environment is represented by a color texture file or procedural texture file that is mapped on the inside of an infinitely large, space. Reflection maps can be spherical or cubic. A spherical reflection map requires only one texture or image file, while a cubic reflection map requires six.

Each material description can contain one reflection map statement that specifies a color texture file or a color procedural texture file to represent the environment. The material itself must be assigned an illumination model of 3 or greater.

The reflection map statement in the .mtl file defines a local reflection map. That is, each material assigned to an object in a scene can have an individual reflection map. In PreView, you can assign a global reflection map to an object and specify the orientation of the reflection map. Rotating the reflection map creates the effect of animating reflections independently of object motion. When you replace a global reflection map with a local reflection map, the local reflection map inherits the transformation of the global reflection map.

Syntax

The following syntax statements describe the reflection map statement for .mtl files.

refl -type sphere -options -args filename

Specifies an infinitely large sphere that casts reflections onto the material. You specify one texture file.

"filename" is the color texture file, color procedural texture file, or image file that will be mapped onto the inside of the shape.

```
 refl -type cube_side -options -args filenames

 Specifies an infinitely large sphere that casts reflections onto the
material.  You can specify different texture files for the "top",
"bottom", "front", "back", "left", and "right" with the following
statements:

 refl -type cube_top
 refl -type cube_bottom
 refl -type cube_front
 refl -type cube_back
 refl -type cube_left
 refl -type cube_right

 "filenames" are the color texture files, color procedural texture
files, or image files that will be mapped onto the inside of the shape.

 The "refl" statements for sphere and cube can be used alone or with
 any combination of the following options.  The options and their
arguments are inserted between "refl" and "filename".

 -blendu on | off
 -blendv on | off
 -cc on | off
 -clamp on | off
 -mm base gain
 -o u v w
 -s u v w
 -t u v w
 -texres value

 The options for the reflection map statement are described in detail in
"Options for texture map statements" on page 18.


 Examples

 1  Neon green

 This is a bright green material.  When applied to an object, it will
remain bright green regardless of any lighting in the scene.

 newmtl neon_green
 Kd 0.0000 1.0000 0.0000
 illum 0

 2  Flat green

 This is a flat green material.

 newmtl flat_green
 Ka 0.0000 1.0000 0.0000
 Kd 0.0000 1.0000 0.0000
 illum 1
```

3 Dissolved green

This is a flat green, partially dissolved material.

```
newmtl diss_green
Ka 0.0000 1.0000 0.0000
Kd 0.0000 1.0000 0.0000
d 0.8000
illum 1
```

4 Shiny green

This is a shiny green material. When applied to an object, it shows a white specular highlight.

```
newmtl shiny_green
Ka 0.0000 1.0000 0.0000
Kd 0.0000 1.0000 0.0000
Ks 1.0000 1.0000 1.0000
Ns 200.0000
illum 1
```

5 Green mirror

This is a reflective green material. When applied to an object, it reflects other objects in the same scene.

```
newmtl green_mirror
Ka 0.0000 1.0000 0.0000
Kd 0.0000 1.0000 0.0000
Ks 0.0000 1.0000 0.0000
Ns 200.0000
illum 3
```

6 Fake windshield

This material approximates a glass surface. Is it almost completely transparent, but it shows reflections of other objects in the scene. It will not distort the image of objects seen through the material.

```
newmtl fake_windsh
Ka 0.0000 0.0000 0.0000
Kd 0.0000 0.0000 0.0000
Ks 0.9000 0.9000 0.9000
d 0.1000
Ns 200
illum 4
```

7 Fresnel blue

This material exhibits an effect known as Fresnel reflection. When applied to an object, white fringes may appear where the object's surface is viewed at a glancing angle.

```
newmtl fresnel_blu
Ka 0.0000 0.0000 0.0000
```

```
 Kd 0.0000 0.0000 0.0000
 Ks 0.6180 0.8760 0.1430
 Ns 200
 illum 5
```

8 Real windshield

This material accurately represents a glass surface. It filters of colorizes objects that are seen through it. Filtering is done according to the transmission color of the material. The material also distorts the image of objects according to its optical density. Note that the material is not dissolved and that its ambient, diffuse, and specular reflective colors have been set to black. Only the transmission color is non-black.

```
 newmtl real_windsh
 Ka 0.0000 0.0000 0.0000
 Kd 0.0000 0.0000 0.0000
 Ks 0.0000 0.0000 0.0000
 Tf 1.0000 1.0000 1.0000
 Ns 200
 Ni 1.2000
 illum 6
```

9 Fresnel windshield

This material combines the effects in examples 7 and 8.

```
 newmtl fresnel_win
 Ka 0.0000 0.0000 1.0000
 Kd 0.0000 0.0000 1.0000
 Ks 0.6180 0.8760 0.1430
 Tf 1.0000 1.0000 1.0000
 Ns 200
 Ni 1.2000
 illum 7
```

10 Tin

This material is based on spectral reflectance samples taken from an actual piece of tin. These samples are stored in a separate .rfl file that is referred to by name in the material. Spectral sample files (.rfl) can be used in any type of material as an alternative to RGB values.

```
 newmtl tin
 Ka spectral tin.rfl
 Kd spectral tin.rfl
 Ks spectral tin.rfl
 Ns 200
 illum 3
```

11 Pine Wood

This material includes a texture map of a pine pattern. The material color is set to "ident" to preserve the texture's true color. When applied to an object, this texture map will affect only the ambient and diffuse regions of that object's surface.

The color information for the texture is stored in a separate .mpc file that is referred to in the material by its name, "pine.mpc". If you use different .mpc files for ambient and diffuse, you will get unrealistic results.

```
newmtl pine_wood
Ka spectral ident.rfl 1
Kd spectral ident.rfl 1
illum 1
map_Ka pine.mpc
map_Kd pine.mpc
```

12 Bumpy leather

This material includes a texture map of a leather pattern. The material color is set to "ident" to preserve the texture's true color. When applied to an object, it affects both the color of the object's surface and its apparent bumpiness.

The color information for the texture is stored in a separate .mpc file that is referred to in the material by its name, "brown.mpc". The bump information is stored in a separate .mpb file that is referred to in the material by its name, "leath.mpb". The -bm option is used to raise the apparent height of the leather bumps.

```
newmtl bumpy_leath
Ka spectral ident.rfl 1
Kd spectral ident.rfl 1
Ks spectral ident.rfl 1
illum 2
map_Ka brown.mpc
map_Kd brown.mpc
map_Ks brown.mpc
bump -bm 2.000 leath.mpb
```

13 Frosted window

This material includes a texture map used to alter the opacity of an object's surface. The material color is set to "ident" to preserve the texture's true color. When applied to an object, the object becomes transparent in certain areas and opaque in others.

The variation between opaque and transparent regions is controlled by scalar information stored in a separate .mps file that is referred to in the material by its name, "window.mps". The "-mm" option is used to shift and compress the range of opacity.

```
newmtl frost_wind
Ka 0.2 0.2 0.2
Kd 0.6 0.6 0.6
Ks 0.1 0.1 0.1
d 1
Ns 200
```

```
illum 2
map_d -mm 0.200 0.800 window.mps
```

14 Shifted logo

This material includes a texture map which illustrates how a texture's origin may be shifted left/right (the "u" direction) or up/down (the "v" direction). The material color is set to "ident" to preserve the texture's true color.

In this example, the original image of the logo is off-center to the left. To compensate, the texture's origin is shifted back to the right (the positive "u" direction) using the "-o" option to modify the origin.

```
Ka spectral ident.rfl 1
Kd spectral ident.rfl 1
Ks spectral ident.rfl 1
illum 2
map_Ka -o 0.200 0.000 0.000 logo.mpc
map_Kd -o 0.200 0.000 0.000 logo.mpc
map_Ks -o 0.200 0.000 0.000 logo.mpc
```

15 Scaled logo

This material includes a texture map showing how a texture may be scaled left or right (in the "u" direction) or up and down (in the "v" direction). The material color is set to "ident" to preserve the texture's true color.

In this example, the original image of the logo is too small. To compensate, the texture is scaled slightly to the right (in the positive "u" direction) and up (in the positive "v" direction) using the "-s" option to modify the scale.

```
Ka spectral ident.rfl 1
Kd spectral ident.rfl 1
Ks spectral ident.rfl 1
illum 2
map_Ka -s 1.200 1.200 0.000 logo.mpc
map_Kd -s 1.200 1.200 0.000 logo.mpc
map_Ks -s 1.200 1.200 0.000 logo.mpc
```

16 Chrome with spherical reflection map

This illustrates a common use for local reflection maps (defined in a material).

this material is highly reflective with no diffuse or ambient contribution. Its reflection map is an image with silver streaks that yields a chrome appearance when viewed as a reflection.

```
ka 0 0 0
kd 0 0 0
ks .7 .7 .7
illum 1
refl -type sphere chrome.rla
```

Illumination models

The following list defines the terms and vectors that are used in the illumination model equations:

```
Term   Definition

Ft     Fresnel reflectance
Ft     Fresnel transmittance
Ia     ambient light
I      light intensity
Ir     intensity from reflected direction
       (reflection map and/or ray tracing)
It     intensity from transmitted direction
Ka     ambient reflectance
Kd     diffuse reflectance
Ks     specular reflectance
Tf     transmission filter

Vector Definition

H      unit vector bisector between L and V
L      unit light vector
N      unit surface normal
V      unit view vector
```

The illumination models are:

0 This is a constant color illumination model. The color is the specified Kd for the material. The formula is:

```
color = Kd
```

1 This is a diffuse illumination model using Lambertian shading. The color includes an ambient constant term and a diffuse shading term for each light source. The formula is

```
color = KaIa + Kd { SUM j=1..ls, (N * Lj)Ij }
```

2 This is a diffuse and specular illumination model using Lambertian shading and Blinn's interpretation of Phong's specular illumination model (BLIN77). The color includes an ambient constant term, and a diffuse and specular shading term for each light source. The formula is:

```
color = KaIa
     + Kd { SUM j=1..ls, (N*Lj)Ij }
     + Ks { SUM j=1..ls, ((H*Hj)^Ns)Ij }
```

3 This is a diffuse and specular illumination model with reflection using Lambertian shading, Blinn's interpretation of Phong's specular illumination model (BLIN77), and a reflection term similar to that in Whitted's illumination model (WHIT80). The color includes an ambient constant term and a diffuse and specular shading term for each light

```
source.  The formula is:

   color = KaIa
        + Kd { SUM j=1..ls, (N*Lj)Ij }
        + Ks ({ SUM j=1..ls, ((H*Hj)^Ns)Ij } + Ir)

   Ir = (intensity of reflection map) + (ray trace)

 4  The diffuse and specular illumination model used to simulate glass
is the same as illumination model 3.  When using a very low dissolve
(approximately 0.1), specular highlights from lights or reflections
become imperceptible.

 Simulating glass requires an almost transparent object that still
reflects strong highlights.  The maximum of the average intensity of
highlights and reflected lights is used to adjust the dissolve factor.
The formula is:

   color = KaIa
        + Kd { SUM j=1..ls, (N*Lj)Ij }
        + Ks ({ SUM j=1..ls, ((H*Hj)^Ns)Ij } + Ir)

 5  This is a diffuse and specular shading models similar to
illumination model 3, except that reflection due to Fresnel effects is
introduced into the equation.  Fresnel reflection results from light
striking a diffuse surface at a grazing or glancing angle.  When light
reflects at a grazing angle, the Ks value approaches 1.0 for all color
samples.  The formula is:

   color = KaIa
        + Kd { SUM j=1..ls, (N*Lj)Ij }
        + Ks ({ SUM j=1..ls, ((H*Hj)^Ns)Ij Fr(Lj*Hj,Ks,Ns)Ij} +
Fr(N*V,Ks,Ns)Ir})


 6  This is a diffuse and specular illumination model similar to that
used by Whitted (WHIT80) that allows rays to refract through a surface.
The amount of refraction is based on optical density (Ni).  The
intensity of light that refracts is equal to 1.0 minus the value of Ks,
and the resulting light is filtered by Tf (transmission filter) as it
passes through the object.  The formula is:

   color = KaIa
        + Kd { SUM j=1..ls, (N*Lj)Ij }
        + Ks ({ SUM j=1..ls, ((H*Hj)^Ns)Ij } + Ir)
        + (1.0 - Ks) TfIt

 7  This illumination model is similar to illumination model 6, except
that reflection and transmission due to Fresnel effects has been
introduced to the equation.  At grazing angles, more light is reflected
and less light is refracted through the object.  The formula is:

   color = KaIa
        + Kd { SUM j=1..ls, (N*Lj)Ij }
        + Ks ({ SUM j=1..ls, ((H*Hj)^Ns)Ij Fr(Lj*Hj,Ks,Ns)Ij} +
Fr(N*V,Ks,Ns)Ir})
```

```
        + (1.0 - Kx)Ft (N*V,(1.0-Ks),Ns)TfIt

 8  This illumination model is similar to illumination model 3 without
ray tracing.  The formula is:

   color = KaIa
        + Kd { SUM j=1..ls, (N*Lj)Ij }
        + Ks ({ SUM j=1..ls, ((H*Hj)^Ns)Ij } + Ir)

   Ir = (intensity of reflection map)

 9  This illumination model is similar to illumination model 4without
ray tracing.  The formula is:


   color = KaIa
        + Kd { SUM j=1..ls, (N*Lj)Ij }
        + Ks ({ SUM j=1..ls, ((H*Hj)^Ns)Ij } + Ir)

   Ir = (intensity of reflection map)

 10  This illumination model is used to cast shadows onto an invisible
surface.  This is most useful when compositing computer-generated
imagery onto live action, since it allows shadows from rendered objects
to be composited directly on top of video-grabbed images.  The equation
for computation of a shadowmatte is formulated as follows.

 color = Pixel color.  The pixel color of a shadowmatte material is
always black.

 color = black

 M = Matte channel value.  This is the image channel which typically
represents the opacity of the point on the surface.  To store the shadow
in the matte channel of the image, it is calculated as:

 M = 1 - W / P

 where:

 P = Unweighted sum.  This is the sum of all S values for each light:

 P = S1 + S2 + S3 + .....

 W = Weighted sum.  This is the sum of all S values, each weighted by
the visibility factor (Q) for the light:

 W = (S1 * Q1) + (S2 * Q2) + .....

 Q = Visibility factor.  This is the amount of light from a particular
light source that reaches the point to be shaded, after traveling
through all shadow objects between the light and the point on the
surface.  Q = 0 means no light reached the point to be shaded; it was
blocked by shadow objects, thus casting a shadow.  Q = 1 means that
nothing blocked the light, and no shadow was cast.  0 < Q < 1 means that
```

the light was partially blocked by objects that were partially dissolved.

S = Summed brightness. This is the sum of the spectral sample intensities for a particular light. The samples are variable, but the default is 3:

S = samp1 + samp2 + samp3.