

```

ModifyGizmo showAxisCue=1
ModifyGizmo setQuaternion={0.435,-0.227,-0.404,0.777}

// Append a red, opaque sphere
AppendToGizmo/D sphere={0.25,25,25}, name=sphere0
ModifyGizmo modifyObject=sphere0, objectType=Sphere, property={colorType,1}
ModifyGizmo modifyObject=sphere0, objectType=Sphere,
    property={color,1.0,0.0,0.0,1.0}

// Append a blue, translucent box
AppendToGizmo/D box={0.5,0.5,0.5}, name=box0
ModifyGizmo modifyobject=box0, objectType=Box, property={colorType,1}
ModifyGizmo modifyobject=box0, objectType=Box,
    property={colorValue,0,0.0000,0.2444,1.0000,0.5000}

```

You now have a translucent blue box surrounding a red sphere but you can not see the red sphere because alpha blending is disabled by default. Now we enable it:

```

// Enable transparency blend
AppendToGizmo attribute blendFunc={770,771}, name=blendFunc0
ModifyGizmo insertDisplayList=0, opName=enableBlend, operation=enable,
    data=3042
ModifyGizmo insertDisplayList=0, attribute=blendFunc0

```

The last set of commands is what is executed when you choose Gizmo Menu→Enable Transparency Blend.

Converting Igor Colors to Gizmo Colors

For historical reasons, Igor represents color components as integer values from 0 to 65535. OpenGL, and consequently Gizmo, represent color components as floating point values from 0.0 to 1.0. To convert from an Igor color component value, such as you might receive from the ColorTab2Wave operation, to a Gizmo color component value for use in a Gizmo command, you need to divide the Igor color component value by 65535.

Here is an example of such a conversion:

```

ColorTab2Wave Rainbow                                // Creates M_colors
MatrixOP/O gizmoRainbowColors = M_colors/65535        // Convert to SP and scale
Redimension/N=(-1,4) gizmoRainbowColors               // Add a column for alpha
gizmoRainbowColors[][3] = 1                           // Set the alpha to 1 (opaque)

```

Colors, Materials and Lighting

The perceived color of an object depends on the combination of the color, the material and the lighting.

A color material specifies which faces of an object are to be colored by OpenGL and the way the object emits or responds to light. The "face" property can be set to GL_FRONT, GL_BACK or GL_FRONT_AND_BACK. The "mode" property can be set to GL_EMISSION, GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR or GL_AMBIENT_AND_DIFFUSE.

When you create a Gizmo object you have the option to specify a color or to leave it unspecified. If you specify a color, Gizmo creates a default color material for the object. The default color material has the GL_FRONT_AND_BACK and GL_AMBIENT_AND_DIFFUSE settings. If you don't specify a color then Gizmo does not create a default color material and you must create a color material yourself. This color material affects all objects that appear later in the display list if they have no default color material.

Whatever color material is in effect for a given object, you can modify it by adding ambient, diffuse, specular, shininess or emission attributes above it in the display list.

To create a shiny object (e.g., sphere), start with a sphere object, add to it shininess and specular attributes and then add the sphere to the display list following a light object that has matching diffuse and specular components. In this case the info window and display window like this: