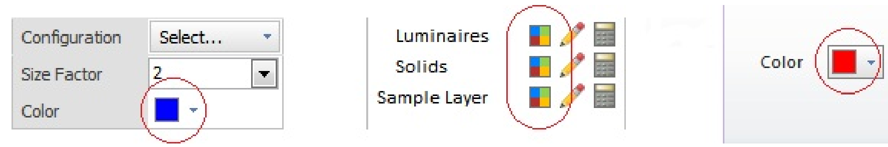
13.1 Color

The **Color** ***Dialog*** is used in many instances in Visual to allow for **Color** selection.

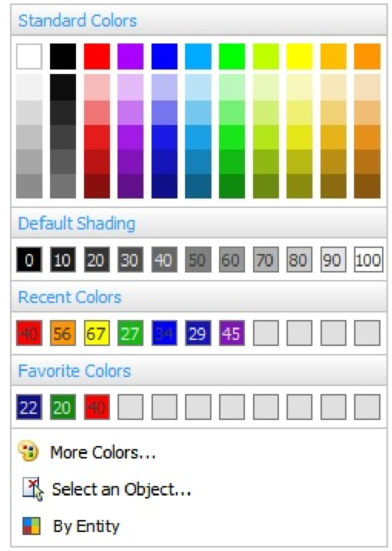
Depending on the context, there are different ways to initiate the **Color** ***Dialog***. The button is a color swatch and often has a small down-arrow next to it; the color of the swatch will vary based on default settings and user selection. Some examples of where the button appears are shown at right.



13.1.1 Color Dialog

The **Color** ***Dialog***> is used in the **Layer Manager** and the ***Luminaire*** **Editor**, as well as when constructing **Solid** and **Background** objects, **Calculation Zones**, and **Statistical Zones**. For specific information on how the **Color** ***Dialog*** relates to those commands, see the relevant sections in this manual.

The **Color** ***Dialog*** is a flyout composed of four *panels* and three commands.



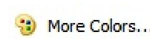
Standard Colors is a set of readily accessible choices covering a range large enough to allow for various objects to have different colors for easy reference.

Default Shading is a set of gray shades in 10% increments for ***Reflectance*** assignment when exact **Color** is not important.

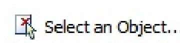
Recent Colors shows the last eleven **Colors** used. Each new **Color** chosen is amended to the left end of the row.

Favorite Colors are user-specified and saved as defaults to be used across multiple projects. To save a **Favorite Color** it must be in the **Recent Colors** *panel* (i.e. the **Color** must be previously selected for use). Right-click the mouse on the desired **Color** in the **Recent Colors** *panel* and select **Add to Favorites**. To delete colors, right-click the mouse on the desired color and select **Remove.**

The **More Colors** button initiates the **Color Selection** ***Dialog***. See [Using the Color Selection Dialog] for more information.

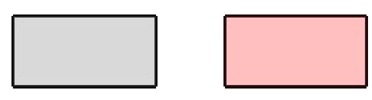


The **Select an Object…** command allows for the **Color** to be set by selecting an existing object.



The **By Entity** command appears in the **Color** ***Dialog*** when initiated from the **Layer Manager** and changes the assignment mode back to the default of **Color** being determined by entity properties if the **Layer** has been assigned a **Color**. See Layer Manager for more information.

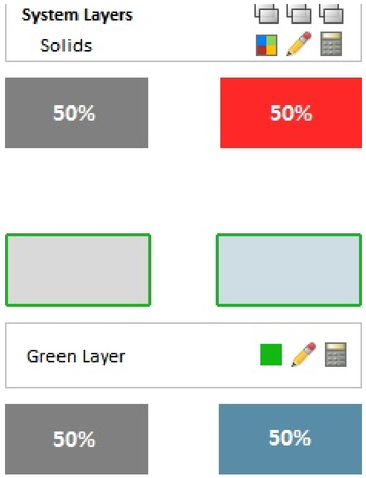
To select a **Color**, initiate the **Color** ***Dialog***, and then left-click the desired **Color**. The *dialog* will be closed and the **Color** assignment will be made.



The choice of a **Color** is only necessary if **Color Rendering** is a desired output from Visual. Grayscale choices yield the same numeric results as "colored" choices, assuming the ***Reflectance*** value is the same.

The Color and *Reflectance* chosen for **Solids** is independent of the **Layer Color**. The **Layer Color** is used to provide user feedback in the **Design Environment** and the **Color** and ***Reflectance*** is used for calculation.

On the right, the objects with a black border have different **Color** (and therefore ***Reflectance***) on the **Solids System Layer** (**Color** is **ByLayer**) and those with a green border are on a separate **Layer**, also with different **Color** (and ***Reflectance***). In **Shaded Display Mode** shown at the bottom, borders (drawn in the **Layer Color**) are not shown. The gray, red, and blue all yield the same calculational result because they are all 50% ***Reflectance*** .



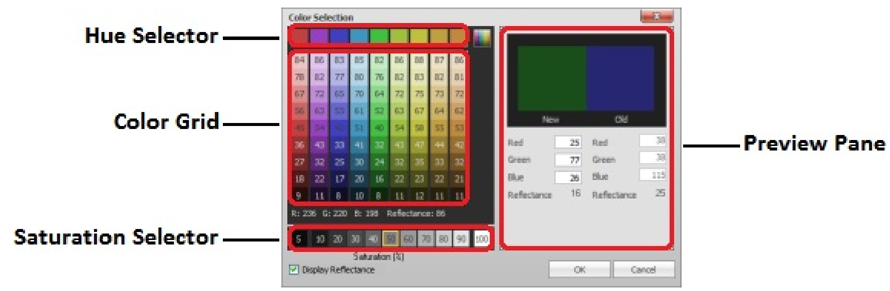
13.1.2 Using the Color Selection Dialog

The **Color Selection *Dialog*** is initiated with the **More Colors…** command in the **Color *Dialog***. This *dialog* is a hybrid 2-D implementation of the 3-D hue, saturation, and lightness (HSL) color *model* with red, green, and blue (RGB) inputs and information.

The *dialog* consists of a **Preview Pane**, a **Hue Selector**, a **Color Grid** containing **Color Swatches**, and the **Saturation Selector**.

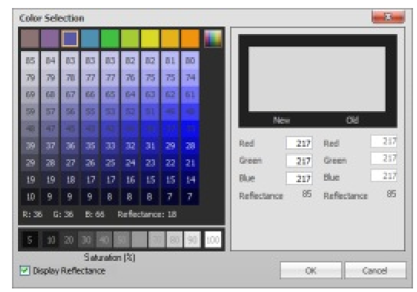
The **Display *Reflectance*** checkbox turns on or off the display of *reflectance* values in the **Color Grid**.

The **Preview Pane** shows larger swatches and more information. The Red, Green, and Blue (RGB) components are pre-loaded into text boxes for alternate modification via that color *model*. The aggregate ***Reflectance*** of the RGB values in the text boxes is shown below those fields. The currently assigned **Color** is shown as the "Old" **Color** in the **Preview Pane**.

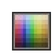


Clicking a button in the **Hue Selector** changes the grid to be (in effect) a more detailed set of hues of the chosen base color. Since saturation values are pre-set left-to-right (95% maximum on the right), the row of **Saturation Selector** buttons are disabled.

This process allows for each angular "lune" of hue (color) to be isolated in the HSL color *model*.



To reset the grid to display the original gamut of hues, click the multi-colored button at the upper right of the **Color Grid**.



Clicking a **Color Swatch** in the **Color Grid** places that **Color** in the **Preview Pane** for comparison and that **Color** will be applied if the **OK** button is clicked.

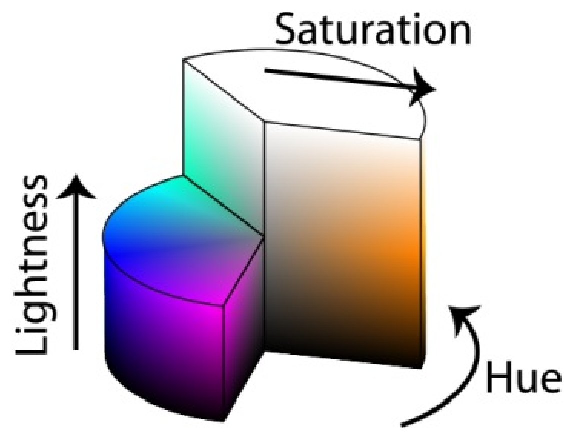
To close the *dialog* and apply the selected **Color** to the **Object** or **Layer**, click the **OK** button. Clicking the **Cancel** button closes the *dialog* without making changes.

Hue is what is most often referred to as "color" in English and this manual.

13.1.3 HSL Color Model

A brief discussion of the HSL color *model* is appropriate to understand how user input to the **Color Selection *Dialog*** impacts what is displayed.

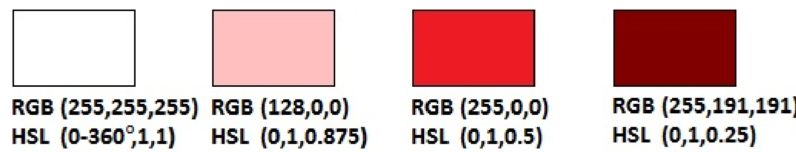
The HSL *model* is based upon traditional color mixing methods such as in mixing paint; brightly colored pigments are mixed with black or white to achieve lighter, darker, or in other words less colorful colors.



Hue is an attribute of visual sensation according to which an area appears to be similar to red, yellow, green, or blue, or a combination of any two of those colors. Hue is shown in the **Hue Selector** and the various columns in the **Color Grid**.

Saturation is the colorfulness of a stimulus relative to its own brightness. In Visual, this is a percentage value controlled by the **Saturation Selector** buttons at the bottom of the **Color Grid**. A saturation of 100% would yield the middle row in the **Color Grid** being the most colorful. For example, in RGB color space, this would be a “red” that was quantified as (255,0,0).

Lightness is the brightness relative to the brightness of a similarly illuminated white. Lightness varies with each row in the **Color Grid** with the middle row having a lightness of 0.5. Moving toward the top of the grid yields more mixture with white and moving toward the bottom of the grid yields more mixture with black. For example, this would respectively yield a “pink” and “burgundy” as shown at right.



For more discussion of color *models* see http://en.wikipedia.org/wiki/HSL_and_HSV and http://en.wikipedia.org/wiki/Comparison_of_color_models_in_computer_graphics

13.2 Calculation Engine

This appendix contains discussion of the theory and methods used to generate calculations and renderings.

13.2.1 Introduction

Scope of calculations

The calculation engine used in Visual photometrically *models* the interaction of *luminaires*, sun, and sky in a user-specified environment that may consist of surfaces that absorb and reflect light which have arbitrary orientations and *planar* shapes. The detail and accuracy of the *photometric model* is sufficient to predict direct and *interreflected* illuminances at any array of points.

Geometric input

Surfaces that block, reflect, and/or transmit light can be *planar* polygons.

Photometric input

Luminaires

For electric lighting calculations, light sources are *luminaires* that have a specified luminous extent and an arbitrary luminous intensity distribution.

IES files, Elumet file, TM14 files

Luminaire data is assumed to be contained entirely within any of the three most commonly used data files for the transfer of *photometric* information. At a minimum, these files give the luminous extent, specify which of the standard *coordinate* systems is used to describe the luminous intensity distribution, list the angles of that *coordinate* system that are used, and list the luminous intensity values of the *luminaire* at those angles.

The Visual calculation engine assumes that *photometric* data files are in that form defined by IES/ANSI standard LM-63-02. The user interface to Visual converts any of the user-supplied *photometric* files to an equivalent LM-63-02 file and submits them to the Visual calculation engine.

Surfaces

Surfaces ("solids") specified by the user are single *planar* entities with a surface normal (perpendicular) derived from the order in which the user specifies the *vertices* of the *polygon* defining the shape and orientation of the surface. Single user surfaces are treated in the calculation engine as two surfaces, back-to-back, separated by an internally-determined incremental distance. They are assumed to have identical *photometric* properties.

Reflectance

Reflectance is assumed to be perfectly *diffuse* and can have values between 0.0 and 0. 999. *Reflectance* is specified in the Visual user interface in percentage form and any value specified as 100% is reduced to 0.999. Perfect diffusion permits the assumption that the amount and distribution of reflected light is independent of incidence direction.

Specular or so-called mixed *reflectance* cannot be modeled in Visual at this time.

Transmittance

Transmittance is assumed to be either perfectly *diffuse* or perfectly image preserving. *Transmittance* can have values between 0.0 and 1.0. *Transmittance* is specified in the Visual user interface in percentage form.

Perfectly diffuse transmittance permits the assumption that the amount and distribution of transmitted light is independent of incidence direction and that transmitted light has a *diffuse* distribution.

Perfectly image preserving *transmittance* preserves the direction of travel of the light, but reduces that amount. The value specified by the user is assumed to be that value of *transmittance* perpendicular to the surface. If the user-specified value of *transmittance* is less than 1.0 (100%), then it is assumed that glass is being used, and that the *transmittance* value depends on incident angle. In this case, the calculation engine automatically determines and uses the appropriate value of *transmittance* for the incidence angles involved.

Absorbtance

If the user-specified values of *reflectance* (ρ) and *transmittance* (τ) do not sum to 1.0, then the absorbtance of the surface is assumed to be $1-\rho-\tau$, and is the fraction of light lost by absorption in the surface.

Spectral reflectance and transmittance

The Visual calculation engine makes "the gray assumption"; that is, all reflectances, transmittances, and flux from light sources are assumed to be spectrally flat. That is, the *photometric* property is uniform throughout (and therefore independent of) visible wavelengths.

Although spectral uniformity is assumed, the values of *reflectance* and *transmittance* are not entirely uncoupled from a surface color specified by the user. The Visual user interface estimates a wide-band *reflectance* from the RGB values that define a user-specified color. If the user chooses to keep the color and the *reflectance* linked, notice is given if the specified color and *reflectance* are incompatible. For example, it is not possible for "brown" to have a high, wide-band *reflectance*.

Some surface colorizing effects can be generated in the renderings. See the section of Rendering.

13.2.2 Basic Calculation Procedure

Luminaires as light sources

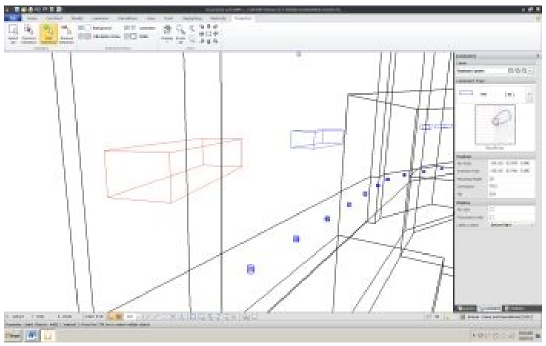
All *photometric* information about a *luminaire* is assumed to be contained in an IES/ANSI LM-63-02 formatted file. The luminous extent is specified with three local luminous dimensions: x, y, z. As defined in the standard, various combinations of zero, positive, and negative values are used to indicate various luminous forms. Regardless of the form indicated in the *photometric* data file, ALL *luminaires* are assumed to be luminous parallelepiped (rectangular boxes). The dimensions of the approximating luminous box are determined to best fit the values and shape provided in the *photometric* data file. These boxes are considered at luminous volumes in the Visual calculation engine.

Luminous volumes

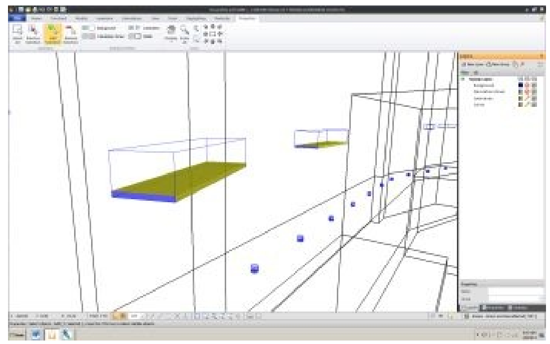
The luminous intensity distribution specified in the file is used to determine which of the six faces of the luminous volume are photometrically active. The "Luminous Volume" button in the "*Luminaire*" *tab* of the Visual user interface toggles these faces on and off. Faces colored yellow are those the calculation engine has made photometrically active, those in blue are inactive.

The total luminous radiant power of the *luminaire* (luminous flux) is distributed among the active surfaces, with the total being equal to that of the entire *luminaire*. Individual faces have individual distributions appropriate for their orientation and size. The sum of these distributions equals that of the entire *luminaire*.

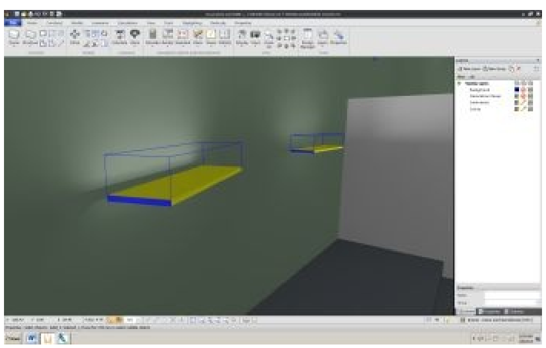
Examples are the following. A lensed *troffer* has only the local bottom surface photometrically active. A surface-mounted wraparound with have three faces active: the bottom and the two long sides. A sharp-cutoff highbay will have 5 surfaces photometrically active: the bottom and four sides.



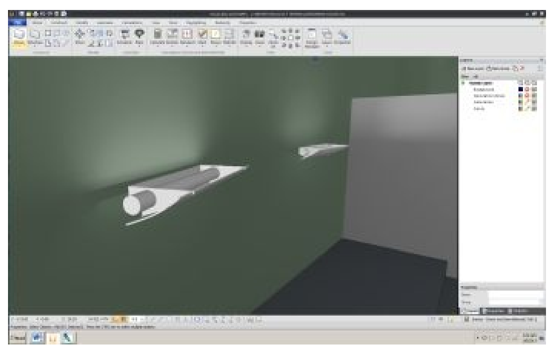
Wireframe view showing luminaire template



Wireframe view showing luminous volume



Rendered view showing luminous volume



Rendered view showing luminaire model

Luminaire photometric information and its extension

No form of commonly used *photometric* data file contains information about *luminaire* appearance or *luminaire* surface/opening *luminance* distribution. Therefore, the Visual calculation engine assumes that the luminous power of any active face of a luminous volume is homogeneous; that is, on any luminous face, the per unit luminous radiant power and distribution are everywhere the same. However, these values can and do differ from active face to active face.

Accounting for luminaire luminous areas (contour integration)

Advanced techniques are used to calculate either the *illuminance* at point or the incident flux on a surface produced by any one face of a *luminaire* luminous volume. These rely on the assumed homogeneity of each active surface, account for the area of the active face, and eliminate the need to discretize the face into assumed point sources. Additionally, they are computationally faster than the point discretization technique.

Illuminance at a point

The Visual calculation engine uses a procedure to calculate the *illuminance* at a point from a face of a *luminaire* luminous volume that was first discovered in 1994. A numerical contour integration is performed around the edges of the face. Details can be found in the technical paper: "*Non-Diffuse Radiative Transfer 1: Area Sources and Point Receivers*," D.L. DiLaura and J. Quinlan, Journal of the Illuminating Engineering Society, Summer 1995, Vol. 24, No. 2, pp. 102-113.

Flux onto a surface

The Visual calculation engine uses a procedure to calculate the flux onto another surface from a face of a *luminaire* luminous volume that was first discovered in 1996. A numerical double contour integration is performed around the edges of the face and the receiving surface. Details can be found in the technical paper: "*Non-Diffuse Radiative Transfer 4: General Procedure for Planar Area Sources and Area Receivers*," D.L. DiLaura and S.R. Santoro, Journal of the Illuminating Engineering Society, Winter 1997, Vol. 26, No. 1, pp. 188-200.

Illuminance at a point

The *illuminance* calculated at any user-specified point in a calculation zone can be of several types: Directional, TV, Maximum Spill, LEEDS Trespass, Spherical, and Constrained Maximum. These can be obtained from two types of basic *illuminance* calculations: with and without one or more *illuminance* normals. In the former, incident flux is weighted with the cosine of the angle between the incidence direction and *illuminance* normals. In the latter, the incident flux is not weighted. Spherical *Illuminance*, for example, uses no *illuminance* normal, Directional *Illuminance* uses one *illuminance* normal at each point oriented in a fixed user-specified direction, TV *Illuminance* uses an *illuminance* normal that changes orientation from point to point, and Maximum Sill use 6 *illuminance* normals, one in each of the *cardinal* directions.

Illuminance is determined in a two-step process. An *illuminance* is calculated using an appropriate method and assuming the light source has an unobstructed view of the illuminated point. Then, if potential occluding objects are detected (surfaces, other *luminaires*) an occlusion factor is calculated, ranging from 0.0 to 1.0, from full occlusion to no occlusion, and is used as a multiplier for the *illuminance*, reducing it if appropriate.

The occlusion factor is determined by ray-casting. An angularly uniform spray of rays is established between the illuminated point and the surface of the source. The angular separation is ½ degree. The number of rays intercepted by *luminaires* or surfaces is determined. If a ray is intercepted by an opaque or diffusely transmissive surface, the surviving ray count is reduced by one. If the only surface(s) involved have an image-preserving *transmittance*, the ray count is reduced by the *transmittance*. The occlusion factor is the ratio of the remaining ray count to the total number of rays.

Direct illuminance at a point

Direct *illuminance* is that produced by a *luminaire* and is calculated using the numerical contour integration method described above, assuming the light source has an unobstructed view of the illuminated point. If possible occluding surfaces are present, an occlusion factor is determined.

Interreflected illuminance at a point

Interreflected illuminance is produced by: 1) the light reflecting from surfaces that are illuminated by sources or other reflecting surfaces and, 2) light transmitted through a diffusely transmissive surface illuminated from the opposite side. In both cases, the source is assumed to be perfectly *diffuse* (either because it is diffusely reflective or diffusely transmissive) and has a uniform *exitance*.

The *illuminance* at a point with an *illuminance* normal is calculated from the equation:

$$E_i = M_i C_i \alpha_i$$

Where E_i is the *illuminance* at the point due to the i^{th} diffusely luminous surface, M_i is the uniform surface *exitance*, C_i is the unoccluded geometric configuration factor from the point to the luminous surface, and α_i is the occlusion factor. The total *interreflected illuminance* at the point is the summation of that produced by all the *diffuse* surfaces:

$$E = \sum_{i=1}^{N} M_i C_i \alpha_i$$

The configuration factors are purely geometric quantities and the standard equation is used for a point and a *planar polygon*. Details can be found in the IES Lighting Handbook, Chapter 10.

The *illuminance* at a point with no *illuminance* normal is calculated from the equation:

$$E_i = \frac{1}{\pi} M_i \omega_i \alpha_i$$

In this case, ω_i is the solid angle subtended at the point by the luminous surface. The total *illuminance* is the summation over all the *diffuse* surfaces.

Discrete Radiative Transfer (Radiosity)

Interreflected illuminance calculations require the *exitance* of diffusely reflecting and transmitting surfaces. The *exitance* of these surfaces originates from light incident directly from sources (referred to as initial light or initial flux) which is increased by the *interreflection* of light between surfaces. This is known as Radiative Transfer Analysis and is also referred to as Radiosity. The procedure is described, in outline in the IES Lighting Handbook, Chapter 10.

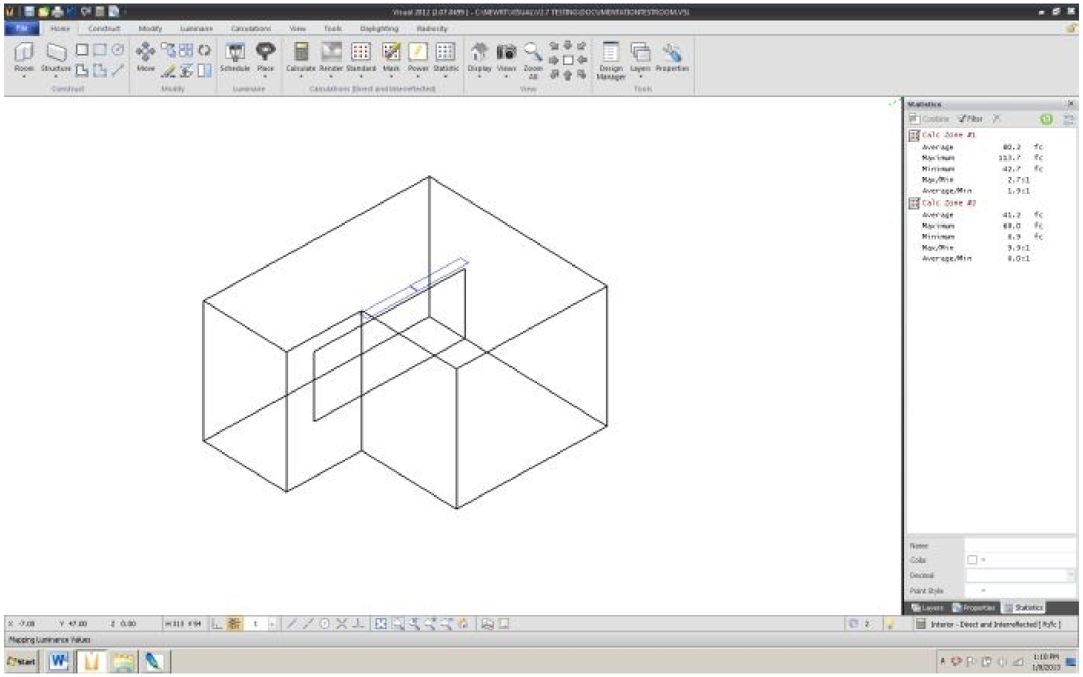
The computational procedure has two essential characteristics: otherwise continuous surfaces are broken up or discretized into small subsurfaces, and each subsurface is characterized by a single *exitance* produced by initial and *interreflected* light. How accurately the collection of uniform, individual exitances represent the actual *exitance* distribution across a large surface depends on the shape of the surface and its illumination conditions. The Visual calculation engine discretizes user-specified surfaces using several criteria and produces subsurfaces of sufficient number to balance necessary accuracy with computational time. This discretization process and the other aspects of the radiative transfer analysis used in the Visual calculation engine are described in the following sections.

13.2.3 Geometric and Photometric Analysis

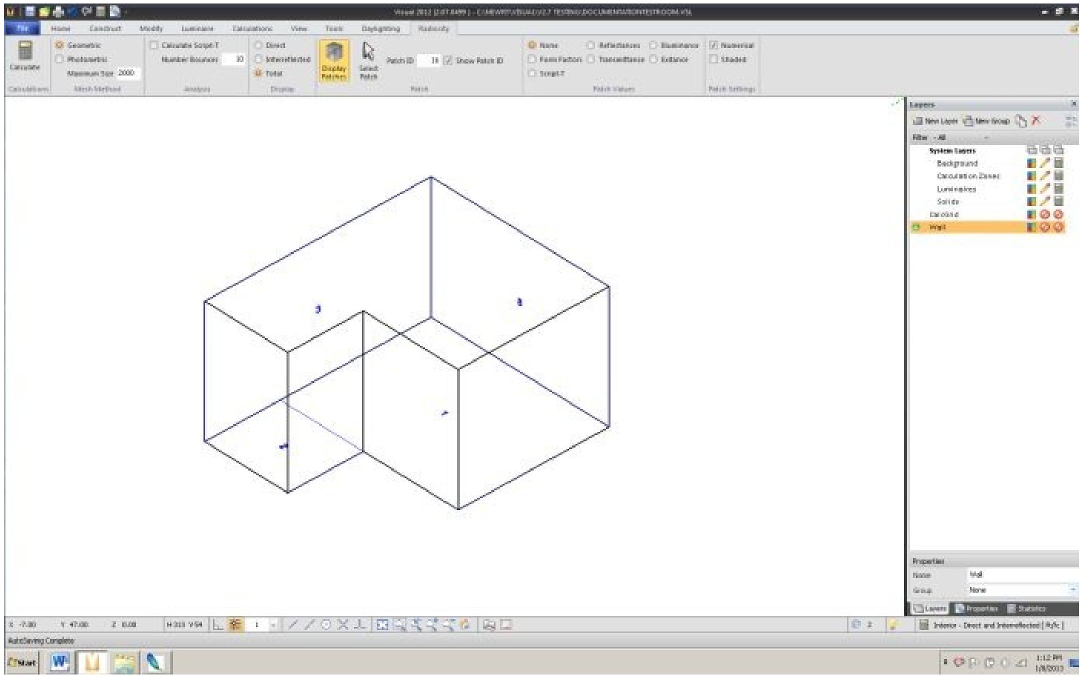
The first stage of user-specified surface discretization is purely geometric and involves two criteria: shape and proximity.

Discretization based on shape

All surfaces handled by the Visual calculation engine are assumed to have four *vertices*, that is, they are *planar* quadrilaterals. User specified surfaces with more than four *vertices* are analyzed with three discretization algorithms; the best discretization is a balance between subsurface shape and number.



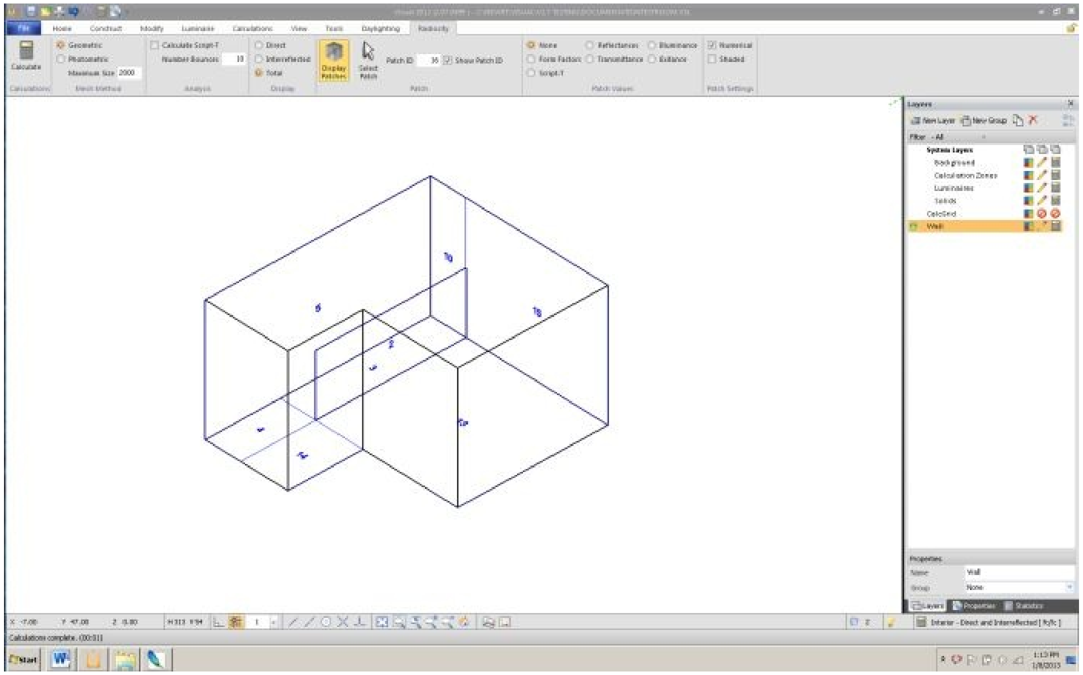
An L-shape room with a partition and two suspended *linear* direct *luminaires*



Shape discretization divides the floor and ceiling so that they are comprised of quadrilaterals.

Discretization based on proximity

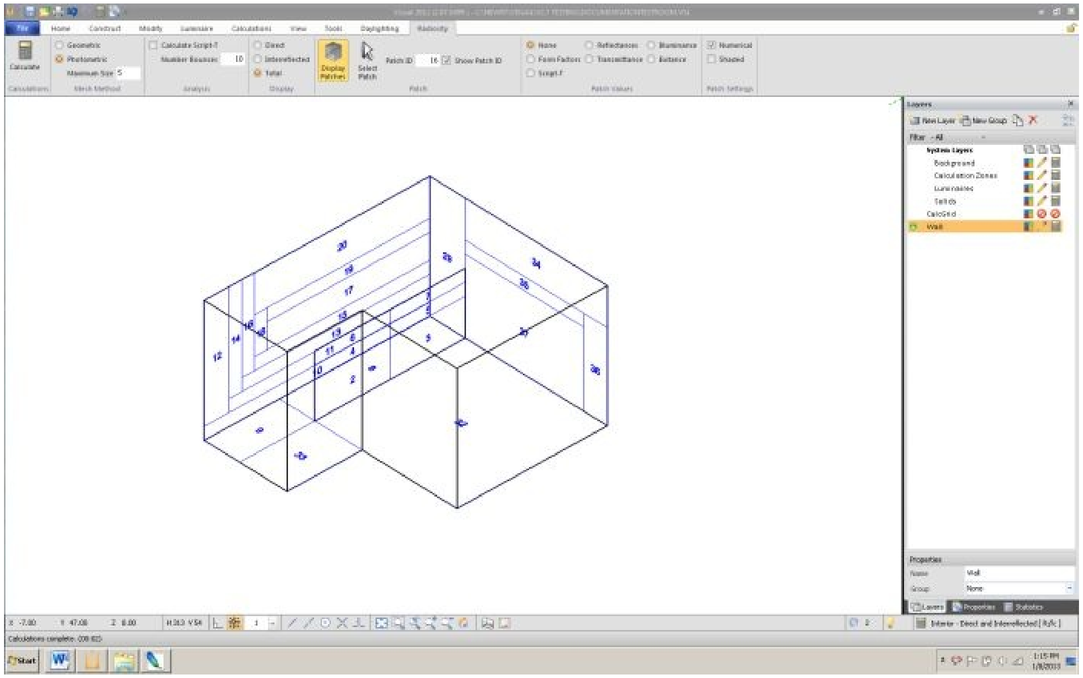
The proximity criteria for discretization accounts for the presence of neighboring or intersecting surfaces are like to produce large *exitance* differences across the intersected surface.



Proximity discretization further discretizes the floor and a wall since they are intersected by the partition.

Discretization based on illuminance gradient

After geometric and proximity, a third criterion is applied to further discretize subsurfaces: a discretization is made along any large *illuminance* gradients on a subsurface. An array of low-precision illuminances is calculated across a surface which consist of direct *illuminance* and *illuminance* produced by light reflected (only) once from other surfaces. Occlusion is accounted for. Neighboring illuminances are compared and if the ratio is greater than 3:1, the subsurface is further discretized there.



Photometric discretization finds large gradients in *illuminance* and further discretized subsurfaces. In this case, the walls contain gradients caused by both shadowing from the partition and the distribution of the *luminaires*.

Discretization of image-preserving or diffusely transmissive surfaces

These surfaces are a special case. Any user-specified surface with an image-preserving *transmittance* is not discretized for any reason.

Any user-specified surface with a diffuse transmittance is, like other surfaces, considered as two surfaces, back-to-back. In this case, the original coupling between the two is maintained throughout the entire computational process. If one surface of the pair is subjected to geometry or *photometric* discretization, that discretization is performed on its back-to-back partner. Thus, both surfaces are subject to discretization due to factors that affect either side. The result is a set of back-to-back subsurface pairs.

13.2.4 Occlusion

The three-part discretization process results in a set of subsurfaces used in the subsequent radiative transfer analysis. In most cases, any one subsurface of this system does not have an unobstructed view of all other subsurface. Subsurfaces may be facing away from each other or the line of sight between partially or fully occluded by other surfaces.

An array of occlusion factors is found that describes view that all subsurfaces have of all other subsurfaces. Ray-casting is used to find these factors. Somewhat like the occlusion process described above, an array of points is established on a subsurface; the density determined adaptively by the proximity of the other subsurface of the pair being considered. From each of these points, an angularly uniform spray of rays is established to the other subsurface. The fraction of all these rays that are not either fully occluded (by opaque surfaces) or partially occluded (by image-preserving transmissive surfaces) establishes the occlusion factor between the pair of subsurfaces.

13.2.5 Form Factors

The final determination of the *exitance* on each subsurface requires knowing the fraction of direct flux that the subsurface can radiatively exchange with all other subsurfaces. Since the surfaces are assumed to be perfectly *diffuse*, these surface exchange factors are purely geometric and are called Form Factors. They are determined in a two-step process: the unoccluded form factor for a pair of subsurfaces is determined and then modified by the occlusion factor for the pair.

Unlike configuration factors, there is no single, simple equation that can be used to calculate form factors. A numerical double contour integration process is used. See the technical article: "*Calculation of Occluded Radiative Exchange Form Factors*," DiLaura, D.L., LEUKOS, July 2006, Vol 3, No. 1, pp. 51-67.

13.2.6 Initial Flux

The final determination of the *exitance* on each subsurface requires knowing the total initial or direct flux onto each subsurface. These fluxes are determined in a two-step process: the unoccluded flux from a source onto a subsurface is determined and then modified by the occlusion factor for the source-subsurface pair.

The numerical double contour integration process described above is used to determine the unoccluded flux and the result multiplied by the appropriate occlusion factor. For flux accounting purposes, the flux arriving to each subsurface from each source in the system is recorded. If it is determined that a source is completely surrounded by subsurfaces, it is possible to check that the total flux involved for the source is correct. The total flux to all surfaces from that source should equal the known total emitted by the source (*luminaire* lumens). Any imbalance is corrected on each surface, the correction being weighted by the amount of flux onto the surface. In this way and in most cases, the flux from each source distributed to all subsurfaces exactly equals the total source emitted flux.

Using the initial flux and the subsurface area, the initial *illuminance* on each subsurface can be determined.

13.2.7 Final Illuminance

Determining the *exitance* at each subsurface after all interreflections (called the final *exitance*) involves solving a system of equations that contain the initial illuminances, *diffuse* reflectances, and radiative exchange form factors. Details are in the IES Lighting Handbook, Chapter 10.

If any of the subsurfaces are diffusely transmissive, the system of equations is expanded to include the flux that back-to-back diffusely transmissive subsurfaces exchange with each other, modeling the flux that is transmitted through the original transmissive surface.

The system of equations is solved iteratively and the result is the final *illuminance* on each subsurface, accounting for all interreflections. Multiplication by the *reflectance* gives the final *exitance*.

13.2.8 Processing Calculation Zones

When the radiative transfer analysis is complete, all information required to determine the direct and *interreflected illuminance* at points in a user-specified calculation zone is available. The process used to determine the illuminances is described in the section "*Illuminance* at a Point" above.

13.2.9 Rendering

The Visual calculation engine can produce most of the data required to display a photometrically accurate rendering of a project. The assumptions underlying the calculations are the same as those for the general radiative analysis of the project; the most important are diffuse reflectance and spectral flatness. Since all surfaces are *diffuse*, the calculations required for a rendering can be performed once and provide all the data required for any view of the project desired by the user. Thus, changing views or navigating through the project does not require recalculation, only a change of the subset of data which is displayed.

Basic procedure

The basic procedure used in the Visual calculation engine to generate renderings has three steps: 1) generate arrays of triangles that cover user-specified surfaces and have exitances that are photometrically accurate, 2) display these triangles in an appropriate geometric and screen-*luminance* manner, and 3) refine the rendering with multiple calculation passes to more accurately *model* surface *exitance* distributions.

Generating triangles and their exitances

Illuminance arrays on user surfaces

The determination of the necessary triangles begins with an array of illuminances calculated on a user-specified surface. At each point in this array, the possibly-occluded direct and *interreflected illuminance* is calculated. This is a double-pass process: after the illuminances at each grid point are determined, each 2-point x 2-point subsection of the grid is examined for high gradients. If a high gradient is present, that subsection is arrayed with additional, more tightly spaced points. Illuminances are calculated at each of the points in each such subgrid during a second pass. This brings out necessary detail in the *exitance* distribution on the surface while minimizing calculation time.

Contouring

Based on the project type, and the range and gradients of *illuminance* found in the array, points defining up to 256 iso-*illuminance* contour lines are determined. The points along a contour vary in spacing; small spacing where the line is highly curved, and large where the line is straight.

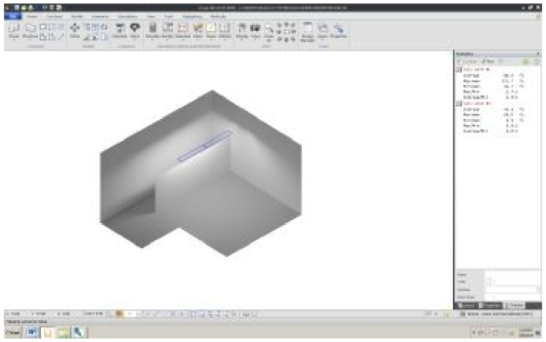
Added illuminance points

To provide for necessary detail, additional illuminances are calculated along the lines that form the boundary of the surface and along lines defined by the intersection of the surface with other surfaces in the project. These additional illuminances help define sharp shadows and accurately portray touching surfaces.

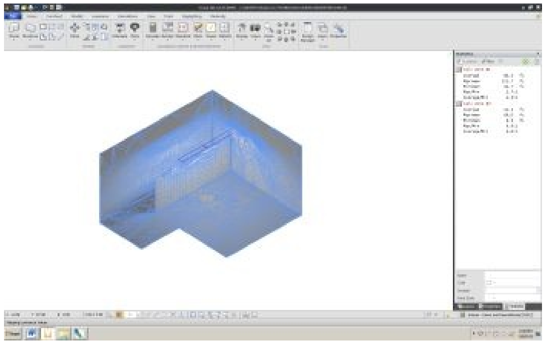
Triangularization

A collection of triangles is built from the points along the contour lines and the added *illuminance* points. Constrained Delaunay triangularization is used. The constraints are the edges formed by the sections of contour line, and the sections connecting the points along the additional lines of *illuminance*. The outline of resulting triangles can be toggled on and off in a rendering with the 7-key.

Displaying and scaling triangle exitances to screen gray scales



Rendering



Rendering showing component triangles

Multiple rendering passes

As each surface to be rendering in the project is processed, a record is kept of the number and extent of high gradients. If adaptively determined re-rendering criteria are met, the surface is flagged for an additional rendering calculation pass.

After all surfaces have been processed, the Visual user interface takes the triangle data provided by the calculation engine and generates the rendering display. Meanwhile, the calculation engine processes all surfaces that have been flagged for additional work. In this addition pass, the density of all *illuminance* points on the surface is incremented and the calculation-triangularization process outlined above is repeated. Not all surfaces may require additional calculation.

When the engine completes processing these flagged surfaces, the Visual user interface updates the data it has on all surfaces, replacing previous data for a surface with any that was generated during the addition rendering computation.

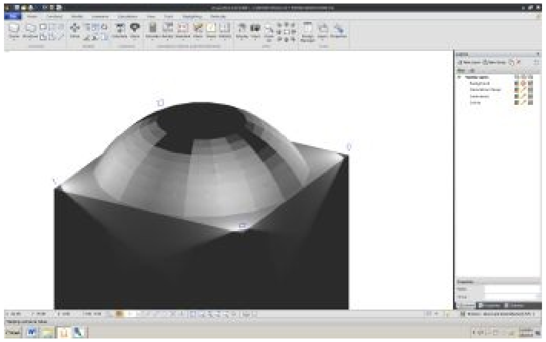
This entire process is repeated up to four times. Each time the rendering calculations are performed on a denser grid of points. It is usually the case that the list of surfaces that are recalculated and updated gets smaller with each pass.

Approximating the appearance of curved surfaces

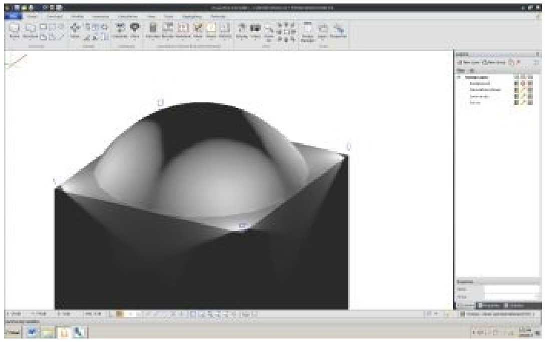
Surfaces handled in the Visual calculation engine are *planar*. In many cases any array of these approximates the surface of a dome, or a column, or a curved wall. To make renderings of such surfaces more realistic, special processing can be invoked, at the discretion of the user, to more faithfully render them by eliminating the abrupt change in *exitance* that is otherwise present on either side of an edge shared by surfaces that meet at even a slight angle.

If a user-specified surface meets another at an angle less than 20-degrees then the *illuminance* normals used in the surface *illuminance* calculation are modified. All edges of a surface are examined to see if the angle to the adjoining surface is less than 20-degrees. If so, the normal at the *vertices* involved are changed from that of the original *planar* surface (which is the default case) to an average formed with the normals of the original surface and those adjoining it at the required small angle. These new, interpolated normals can spread outward, defining a convex surface, or bend inward, defining a concave surface.

The position and orientation of the *vertex* normals are used to define a new, temporary convex or concave surface that passed through the original surface *vertices*. This temporary surface is used to define new calculation points and *illuminance* normals than produce illuminances for a local, curved surface. These points and illuminances are used in the manner described above to generate rendering triangles.



Rendering of a dome without curve approximation



Rendering of a dome with curve approximation

Colorized surfaces

Luminaire models

13.2.10 Daylighting

Visual can perform daylighting calculations and provide daylighting renderings in a single-instance mode; that is, for a particular place at a particular time. The basic calculation procedure is that same as that described for electric lighting; with the sky and sun considered as additional light sources.

Additional data

To add the sky and sun as light sources, additional user data is required; used to find either the appropriate weather data from the Visual Weather Database, or to calibrate a CIE sky specified by the user. In addition, the diffuse reflectance of the surrounding ground *plane* must be specified, as well as glazing information.

Project location

Location is specified by Longitude and Latitude, specified in degrees. Positive and negative values of Latitude specify north and south the equator, respectively. Positive and negative values of Longitude specify east and west values from the Prime Meridian at Greenwich England. Longitude and Latitude are input by the user of come automatically from the Visual user interface Site Locator.

Project orientation

The default orientation of the Visual project site is that sky and sun North (geographic North) corresponds to +y in local Visual *coordinates*. A project orientation angle changes the relative angular position of the sky and sun with respect to the project. Positive values, in degrees, rotate the site clockwise when viewed from above. NB: this does NOT change the significance of the local Visual *coordinates* nor does it rotate the Visual *drawing*.

Date and time of analysis

Date and time are local civil time. If in effect, daylight saving time should be indicated.

Weather data

The Visual Weather Database is derived from all of the more than 2100 EnergyPlus data sets that cover most of the globe. For each location with a data set, only required radiometric or *photometric* data has been extracted from the hourly data. If only radiometric data is available at a particular site, *photometric* data is derived using a process devised by Perez. See: "*Modeling Daylight Availability and Irradiance Components from Direct and Global Irradiance*," R. Perez, P. Ineichen, R. Seals, J. Michalsky, and R. Stewart. Solar Energy. Vo. 44, No.5. pp 271-289.

In all cases, the primary *photometric* data that is extracted or generated for every available site are hourly values of direct solar *illuminance* and total horizontal sky *illuminance* for each data of the year. Data in the EnergyPlus weather file for a particular site are usually constructed from several years of measurements, aggregated together to establish a Typical Meteorological Year for that site.

Sun and Sky

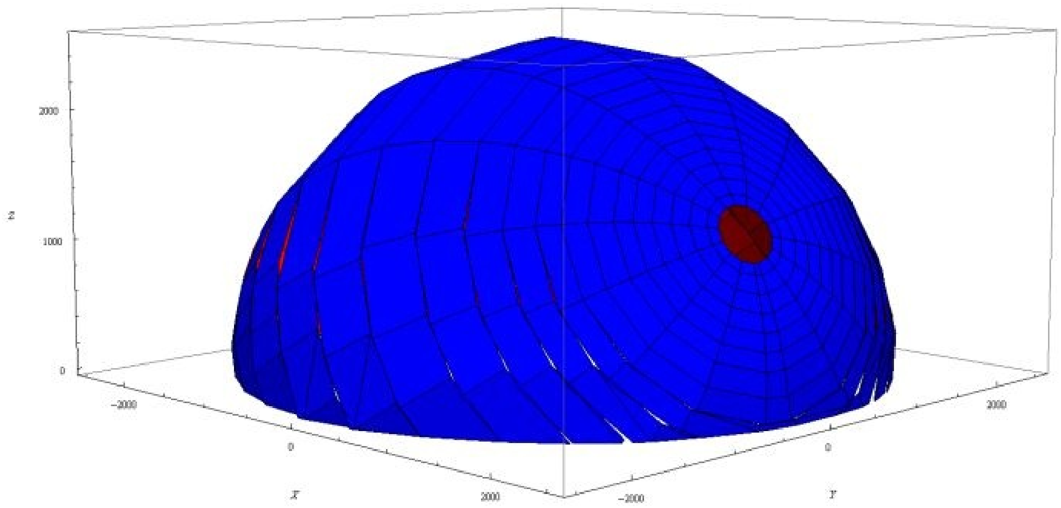
The fundamental luminous properties of sun and sky are the direct solar *illuminance* and the total horizontal *illuminance* produced by the sky. These are either: 1) derived from weather data, or 2) calculated using IES standard sun and sky parameters. See: IES Lighting Handbook, Chapter 7, Section 7.9 Formulary.

The sun is modeled as a luminous disc, ½-degree in diameter. Its luminous power is expressed as the direct, unoccluded *illuminance* produced on a surface with its normal pointed to the sun. Solar position is determined from the location and local time. See: IES Lighting Handbook, Chapter 7, Section 7.1.5 Solar Position.

The sky is modeled as a luminous dome, with a relative *luminance* distribution determined according to ISO/CIE Standard 15469, 2nd edition. The parameters that determine the distribution are either derived from the appropriate weather data or come directly from the standard set of parameters if the user specifies a specific CIE sky. See: "*All-weather Model for Sky Luminance Distribution – Preliminary Configuration and Validation*," R. Perez, R. Seals, and J. Michalsky, Solar Energy, Vol 50, No 3, pp 235-245.

NB: Standard weather data aggregates the *illuminance* from a circum-solar 5-degree circular patch of the sky with the direct *illuminance* from the sun. Therefore, the sky is modeled with a 5-degree hole centered on the sun.

The Visual calculation engine establishes a distance, based on the maximum extents of the project that defines the distance to the sun and the *radius* of the sky dome. The distance is such that that parallax error over the extent of the project to any point on the sky dome is less than ½-degree. The sky is discretized into *planar* rectangles, accounting for *luminance* gradient. Using the *luminance* distribution, element size, and sky dome *radius*, each discretized sky element is assigned a luminous power defined by the direct normal *illuminance* it produces at the center of the Visual project.



Sky dome approximated with planar rectangles, sized and distributed according to the luminance distribution of the sky. Notice the 5-degree hole left for the sun and its circum-solar component.

Ground

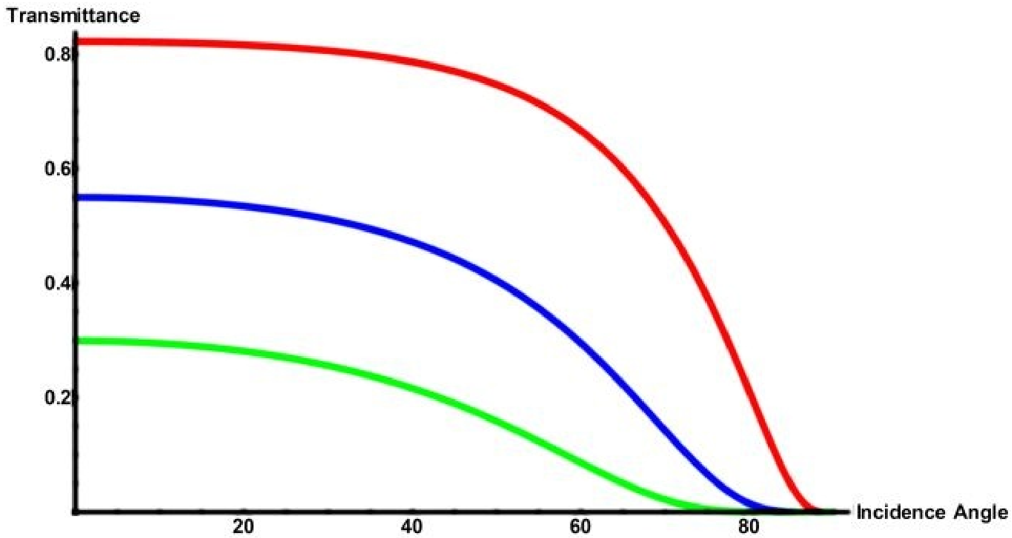
A ground *plane* is automatically established around the project and horizontally positioned at Visual local z=0.0. Its *reflectance* is user-specified. The *plane* is automatically discretized into elements that are truncated wedges surrounding the project.

Apertures, windows, and skylights

Skylight and sunlight illuminate any outward-facing user-specified surfaces and the elements of the ground *plane*. Occlusion by any other surfaces in taken into account. Skylight and sunlight illuminate any surface of an otherwise inward-facing or closed set of surfaces if admitted by an aperture, window or skylight. An aperture is a user-specified opening in an otherwise opaque surface. A window is an aperture into which a user has specified an image-preserving *transmittance* less than 100%. A skylight is an aperture into which a user has specified either an image-preserving or a diffuse transmittance.

Glass transmittance

The image-preserving *transmittance* specified by the user is assumed to be the perpendicular or normal *transmittance*. The Visual calculation engine accounts for the reduction from this value due to increased incident angles. The Fresnel Laws of *Reflectance* and Transmission are used to determine this *transmittance* value. This calculation is done automatically whenever flux is passing through a surface with an image-preserving *transmittance* and uses the normal *transmittance*, the incidence angle, the number of glazing layers, and the assumed index of refraction for glass of 1.5.



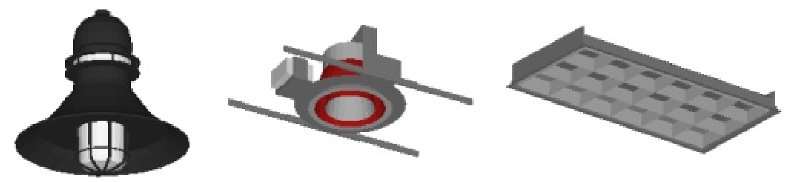
Example of transmittance as a function of incidence angle for single, double, and triple glazing.

13.3.1 Introduction

Visual includes the ability to display detailed **Solid *Models*** of ***Luminaires*** in **Rendered** and **Shaded Display** modes. This appendix describes the basics of building these *models* in Visual for use in the program when they are not present. At the outset, the user who attempts to create a **Solid *Model*** should have a strong drafting background and be expertly familiar with both software use and 3-D visualization and *drawing*.

Solid *Models* can be created in Visual or in any program capable of generating a *DWG* file.

Note that **Solid *Models*** are included in the product database for Acuity Brands Lighting products. The database is accessible when creating entries in the ***Luminaire* Schedule** and *model* data is automatically included in ***Luminaire Type*** definitions. See [3.2 The Luminaire Editor] for more information.

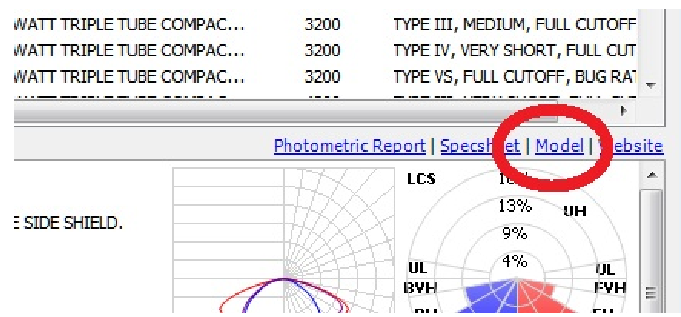


Luminaire* Solid *Models are representations of ***Luminaires*** with much more detail than the wireframe **Symbols** used in Visual, but yet less detail a solid *model* that might be used in the mechanical engineering of a *luminaire*. They are related to BIM files but are not interchangeable with those files.

Prior to constructing a **Solid *Model*** it is strongly suggested that *models* of similar products be examined in Visual.

To view existing **Solid *Models***, navigate to the desired product and click the ***Model*** link in the **Select a *Photometric File*** *dialog*. See [3.1.z Selecting a Photometric File] for more information.

Clicking the link will open the file in the Windows application associated to *DWG* files on the host computer.



Creating Solid *Models* is arguably the most advanced task in Visual and it should by no means be assumed that an advanced Visual user would be able to complete this process. The information is provided for completeness and for the more adventurous users with drafting skills and 3-D visualization aptitude.

VISUAL SUPPORT IS NOT AVAILABLE FOR CREATING *MODELS*.

Unless otherwise noted, terminology used in this chapter is related to creating *models* in Visual. The necessary steps to create *models* in other software should be discernible from the Visual-based text.

13.3.2 Drawing Input

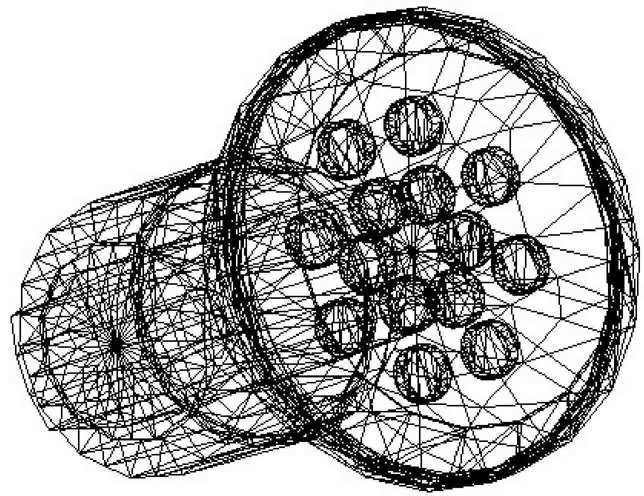
Creating a **Solid** ***Model*** will require dimensional information, and the source will vary depending on the *luminaire* manufacturer.

DWG files are often made available by *luminaire* manufacturers as part of their Building Information Modeling efforts.

These files are a valid basis for a Visual **Solid** ***Model***, but they will normally contain a large amount of extraneous information: internal details, small parts, material thickness, and other information that is excess baggage in Visual.

In most cases, BIM files have entirely more facets and segments than will be necessary.

Creating a Visual **Solid** ***Model*** from a BIM file involves redrawing the desired elements on top of the BIM information. It is *possible* to simply convert the existing closed *polygonal* surfaces to solids using that command in Visual, but this is not often *practical* as it will result in large files and some editing is still necessary to ensure proper *model* content.

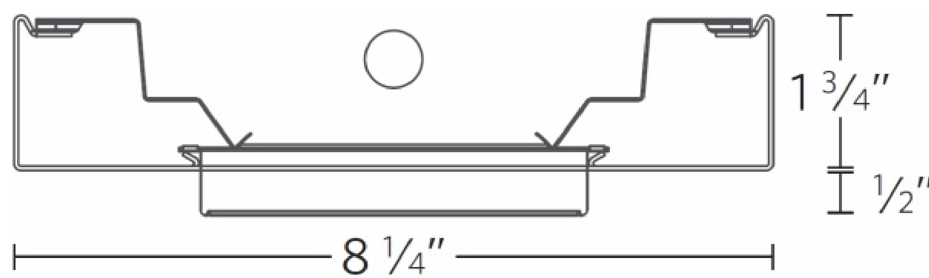


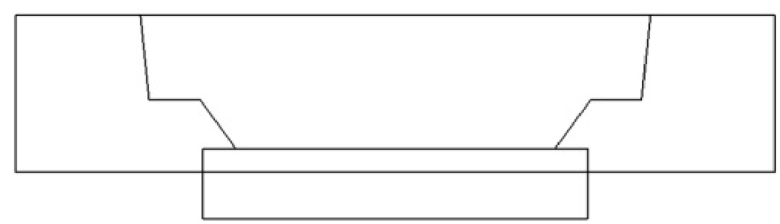
Manufacturer **Specification Sheets** include drawings that may not at the outset look like they contain enough information but realize that, in Visual, Solid *Models* only need to be the basic form and detail.

Most **Specification Sheets** are available in a PDF format that inherently has a detailed *drawing* (top right) that can easily be printed on paper. The basic form can be taken from this *drawing* and scaled using drafting techniques and duplicated in Visual (bottom right)

This 2-D information is then extruded and modified (often by making logical assumptions about product design) to achieve the desired result.

In a great deal of cases, assumptions will have to be made about louver spacing and details in the longitudinal direction that are rarely dimensioned on **Specification Sheets**. This can yield a more than acceptable *model*.





13.3.3 Drawing Layers

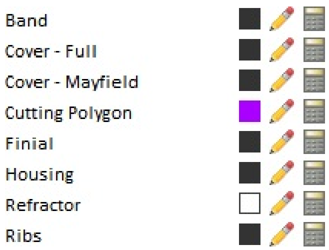
Luminaire* Solid *Models will have multiple **Layers** to which different components and materials will be assigned.

Any number of **Layers** can be created to properly define the *luminaire*. Each different material or component should be assigned to a different **Layer**.

For example, an outdoor decorative acorn *luminaire* would have the **Layers** shown at right. These are the basics of the *luminaire* with additional **Layers** included for available option components.

Note that internal detail not visible (i.e. reflectors behind lenses) do not need to be drawn.

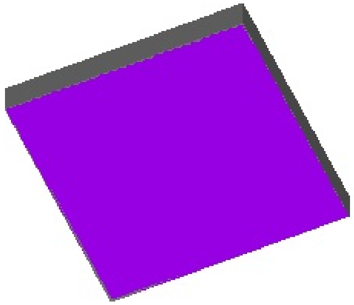
Layer creation outside of Visual is of course analogous.

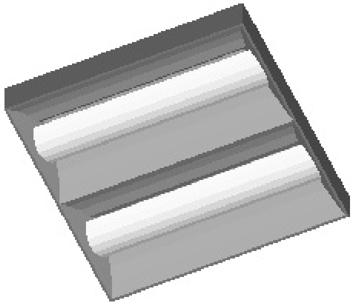


<u>CRITICAL</u>:

For any *luminaire* that is recessed or semi-recessed, the CuttingPolygon **Layer** must be included and a solid must be constructed that indicates what portion of the surface in which the *luminaire* is mounted should be removed. This allows the **Solid *Model*** to be seen. The **Color** of this **Layer** is unimportant, but it is recommended that it be assigned the magenta **Color** to avoid confusion with other elements; this color will not likely be used for any realistic *luminaire model*.

This is basically the only requirement of a ***Luminaire* Solid *Model*.**





Layer Color is what is used to display the components. Any **Color** assigned "ByObject" (in Visual or *CAD* software) is ignored.

Recommended **Colors** for components are:

Lamp - White
Refractor - White
Reflector 90% Gray (or White if a *lamp* is not included)
Housing - 70% Gray

What **Layers** are necessary is based on user preference for what components are included.

13.3.4 Construction

How a ***Luminaire*** **Solid** ***Model*** is constructed is largely at the discretion of the creator but there are a few basic concepts to be used as a guide.

The amount of detail included should be logical. The rule of thumb would be to imagine what basic parts of the *luminaire* are visible in a normal usage scenario, and most importantly from normal viewing distances.

For example, a lensed *troffer* should certainly have a flange in the *model*, but it may not be desirable to draw a flange and a door frame in lieu of one element that *models* both components because the detail would not be seen in normal usage of Visual.

Models are drawn in units of inches, where "1" is one inch. In Visual, this is not the same as the normal one foot or one meter in the normal lighting design process in the Design Environment.

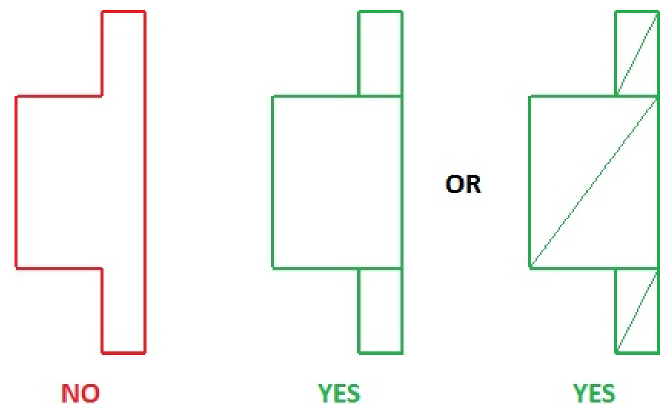
One Unit = 1"

Ideally, files should be less than 500KB in size. Very complex *models* have been created with a file size under 350KB.

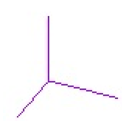
Each surface should be a "Solid" in Visual, or a 3DFace or Closed *Polyline* in other software.

Surfaces can be *polygonal* but CANNOT be concave in shape. Every *vertex* must be able to "see" all other *vertices* of the *polygon*.

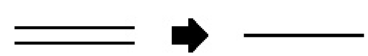
Visual breaks down all surfaces into "child" triangles, so there is no direct benefit to making high-*vertex*-count *polygonal* **Solids**. However, *drawing* nothing but triangles can also be unnecessary; at far right, the triangles are correct but perhaps more effort than logical. Construct what is convenient and logical.



Lines (Visual Background), XREFs, and blocks are ignored. This can be advantageous in that some reference markers can be left in the file for later use. It is useful to draw indicators of the X, Y, and Z axes on the CuttingPolygon **Layer**.



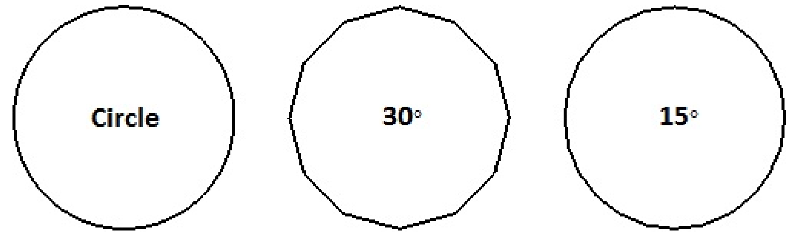
DO NOT *model* surfaces as having thickness. i.e. placing identical surfaces 0.060" apart for cold rolled steel. This will effectively double the file size for detail that can't possibly be seen in normal situations.



Do not use too many facets (*polygon* sides) to *model* curved objects. Generally 3-6 facets per 90 degrees of arc is sufficient. This yields facet angular extent of 30 degrees or 15 degrees.

As can be seen at right, at this zoom level, 15 degrees of facet extent is more than sufficient to approximate a circle. Zoomed out farther, 30 degrees would be acceptable.

Use fewer facets for small details and more facets for larger details like curved housings.

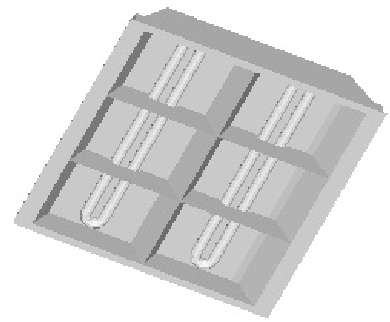


Do not include lamps if they aren't normally seen.

Striplights for example would include lamps. Metal halide highbays might include lamps if extra detail was desired. Lensed troffers certainly do not need lamps in the *model*.

There are numerous cases where lamps (or other details inside the *luminaire* might be seen in abnormal situations (i.e. an occupant looks up at the ceiling), and in these cases, it is up to the *model* creator to decide if the extra detail is of value.

LEDs can be modeled individually, but collectively they produce a "glowing *panel*" in most cases" and the recommendation is then to *model* an LED array as a single **Solid**.

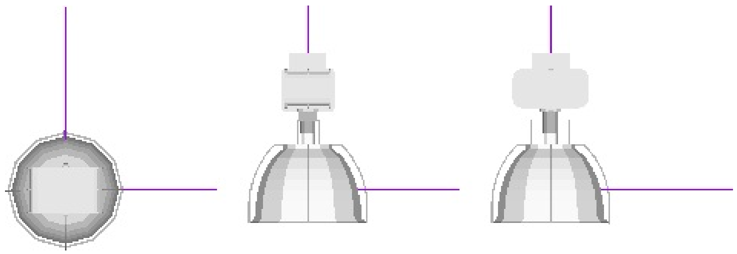


There is never a need to make a *model* look photo-realistic when it is filling the computer screen. There are remotely few situations where the *model* would ever be "seen" in Visual at that size.

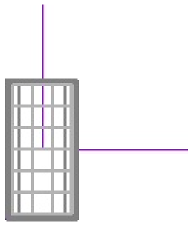
13.3.5 Alignment

The ***Luminaire* Solid *Model*** needs to be aligned properly so that the *model* and the ***Symbol** coordinate*.

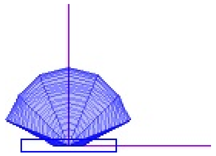
The origin (0,0,0) should be at the center of the luminous area in all three dimensions.



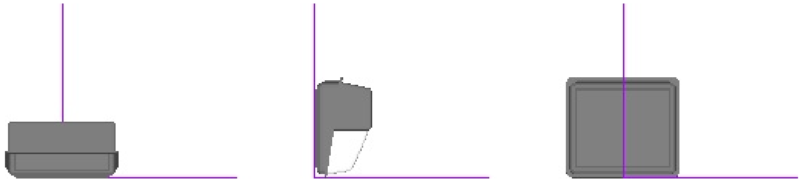
The *model* should have the *lamp* axis in line with the positive-y axis.



Luminaires with an asymmetric distribution should have the asymmetry ("punch") in the positive-y direction to agree with the *photometric* testing and reporting standards of IES LM-63. This may require in-depth knowledge of how *luminaire* optics function.



Wall-mounted *luminaires* are a bit more complex in that the origin needs to be positioned such that the **Solid *Model*** won't be stuck into a wall in **Shaded** or **Rendered Display Modes**. This means that the origin should be at the bottom and rear of the housing.



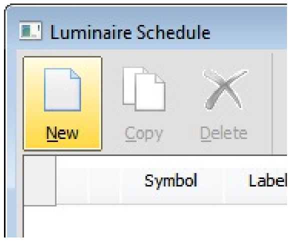
13.3.6 Verification in Visual

A **Solid *Model*** should of course look correct when it is being built and **Shaded Display Mode** is active. It is however sometimes not obvious how *luminaires* are oriented in IES files.

Export the proposed **Solid *Model*** to a *DWG* file from Visual or other *CAD* software.

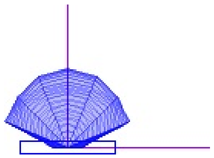
To verify *photometric* alignment while constructing a *model* in Visual:

Create a new ***Luminaire Type*** in the ***Luminaire* Schedule**.



Import the *DWG model* file into the ***Luminaire* Type** definition in the ***Model*** *tab* of the ***Luminaire* Editor**.

Insert a ***Luminaire*** in the **Design Window** with the **Place** command. Make sure the ***Photometric* Web** is on.



Activate **Shaded Display Mode** and verify that the ***Luminaire*** and its distribution are aligned properly. Switch between **Wireframe** and **Shaded Display Modes** if necessary.

13.4 UVGI Calculations

Under construction. Contact Visual Support.