

OptiCAD

Optical Analysis Program

User's Guide

Version 9

Opticad Corporation
511 Juniper Drive
Santa Fe, NM
87501 USA
505-995-1999
support@opticad.com
www.opticad.com

IMPORTANT NOTICE

Microsoft®, Excel, Visual Basic, Windows™, Windows NT™, Windows XP™ and the Windows Logo are trademarks of Microsoft Corporation.

AutoCAD™ is a trademark of Autodesk Incorporated.

Intel® and Pentium®, are trademarks of Intel Corporation.

PostScript™ is a trademark of Adobe Systems Inc.

OptiCAD® is a trademark of Opticad Corporation.

Copyright © Opticad Corporation 1987–2004. All rights reserved.

All other product names or trademarks are the property of their respective owners.

Information in this manual is subject to change without notice and does not represent a commitment on the part of the vendor. The software described in this manual is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement.

The material in this manual is copyrighted and may not be reproduced or electronically distributed without written permission from the copyright holder.

Opticad Corporation provides this publication “as is” without warranty of any kind, either express or implied, including but not limited to the implied warranties or conditions of merchantability or fitness for a particular purpose. In no event shall Opticad Corporation be liable for any loss of profits, loss of business, loss of use or data, interruption of business, or for indirect, special, incidental, or consequential damages of any kind, even if Opticad Corporation has been advised of the possibility of such damages arising from any defect or error in this publication or in the software.

INTRODUCTION	1
What OptiCAD Is	1
What OptiCAD Is Not	1
References on Imaging and Nonimaging Optics	2
Technical Support for OptiCAD	2
System Requirements	2
Installation Procedure	3
Installation Problems	3
Uninstalling OptiCAD	3
Registering OptiCAD	3
Updates of OptiCAD Software	3
Installing the Parallel Printer Port Hardware Key	3
Running OptiCAD	4
Policy on the Hardware Lock	4
Policy on Bug Fixes	4
Policy on Upgrades	5
Version 9.2 New Features	5
Manual Notation	5
Menu Sequence Description	5
Menu Table Description	6
Text Command Description	6
Default Options	6
Unconstrained Ray-Tracing Methodology	6
Ray Energy	7
Coordinate System	7
Global Object Placement (default)	7
Relative Object Placement	8
Inside Object Placement	8
Local Object Rotation (default)	8
Relative Object Rotation	8
Global Object Rotation	8
Global Object Translation (default)	8
Local Object Translation (default)	8
OptiCAD Home Directory	8
StartUp.opt	9
Rays.dat and Spot.dat	9
Opticad.log	9
Glass Files	9
Virus Checker	9
Note on Examples	9
Frequently Asked Questions	9
DEFINITIONS	1
Terminology	1
Active object	1
Placed object	1
Default object	1
Enslitted energy	1
Alias	1

Extent 1
Descendant rays 2
Ghost ray 2
Parent ray 2
Polynet 2
Fresnel reflection 2
Total internal reflection (TIR) 2
Oval extent 2
Define 2
Design 2
IGES 2
NURBS 2
OptiCAD Home Directory 2
List of Symbols 3
USER INTERFACE 1
Introduction 1
Program Functions 1
Working Environment 2
Object Definition and Modification 3
Object Location and Orientation 6
Ray-Trace Analysis 6
Tabulated Data 7
Views and Plots 8
Toolbar Buttons 8
Clear (File → New) 8
Open (File → Open) 8
Save (File → Save) 8
Print (File → Print) 9
Print Preview (File → Print Preview) 9
View System (View → Type → Perspective) 9
View Ortho (View → Type → Ortho) 9
View X or Y or Z (View → Type → X or Y or Z) 9
View Film (View → Type → Film) 9
View Irradiance Map (View → Radiometer) 9
Auto Redraw (Options → AutoRedraw) 9
Auto Retrace 9
Update Ray Trace (Options → Update Ray Trace) 9
Rerun Macro (Files → RerunLastMacro) 9
Point Select (Options → Point) 9
Workbox Sizing (Options → SizeBox) 9
View Point (Options → AzEl) 9
Zoom 9
Pan 9
Snug Workbox (Options → SnugWorkbox) 9
Add Rays 9
Rerun Energy (Analysis → Energy → RepeatLast) 10
OSPb (Edit → Object Properties) 10
View Rays (View → Rays) 10

Open GL Wire (View → Open GL Wireframe) 10
Open GL Solid (View → Open GL Solid) 10
OptiCAD Objects Box 10
Object and Surface Properties Box 10
Boxes at Bottom of Window 11
Text Out Box 11
Text In Box 11
View Azimuth & Elevation (View → Type → Perspective) 11
Object and Surface Names 11
Object Names 11
Object Name Examples 12
Surface Names 12
Surface Name Examples 12
Menu Object Name Input 12
Numerical Input 12
Menu Numerical Input 12
TextIn Box 13
TextOut Box 13
Error Messages 13
File Name Restriction 13
Directory Name Restriction 13
Macro Format 13
Text Based Command Structure 13
Text File Formats 15
Text/Macro Only Commands 15
Pause 15
Echo On 16
Reset Screen.out 16
Display Object absorptance 16
Comment Character 16
Glass Catalogs 17
Glass Catalog Data File Format 17
Display Colors 19
Seeding Random Number Generator 19
Macro Content Restrictions 20
Old Macros 20
(Old) Solid Model 20
Translators 20
BASIC TUTORIAL 1
Introduction 1
Before doing any of the exercises 1
Description of the Toolbar Buttons 2
Screen Operations 3
Zoom in on detail of an X, Y, or Z view 3
Resize the workbox to enclose all of the active objects 3
Select an object as the (selected) active object 3
Select a surface as the (selected) active surface 3
Open the Object and Surface Properties box 3

Menu Operations	3
Run a macro file	3
Edit a macro file	3
Edit the last macro that was run	4
Toolbar Operations	4
ReRun the last macro that was run	4
Repeat the last ray–trace performed	4
Change the screen view	4
View the contents of a Radiometer (irradiance map)	4
Section A	4
Exercise 1 Defining and Placing a Point Source of Rays	4
Exercise 2 Rotating a Point Source of Rays	6
Exercise 3 Defining and Placing a Mirror	7
Exercise 4 Defining and Placing a Lens	8
Exercise 5 Rotating a Lens and Zooming in on a View	9
Section B	9
Optional Procedure to Run the Macro A_Basic_Tutorial.mac	10
Exercise 6 Defining and Placing a Film Plane	10
Exercise 7 Modifying a Surface Shape - Understanding Surface Names	10
Exercise 8 Zooming in on the Film Plane	12
Exercise 9 Modifying Surface Reflectance and Setting Ray–Trace Limits	12
Section C	14
Exercise 10 Extent of a Mirror - Aperture - Film Plane	14
Exercise 11 Extent of Spherical and Cylindrical Lenses	17
Exercise 12 Extent of a Cone - Ellipsoid - Cylinder - Cavity	19
Section D	20
Exercise 13 Global and Relative Placement	20
Exercise 14 Inside Placement	22
Section E	24
Exercise 15 Fresnel Lens	24
Exercise 16 Diffraction Grating	25
Exercise 17 Surface Scattering	27
Section F	28
Exercise 18 Extruded Solid Polynet	29
Exercise 19 Sheet with Four Holes	31
Exercise 20 Extrude Four Holes into an Egg Crate (four lightpipes)	33
TUTORIAL II	1
Viewing	1
Zoom in on an area of interest	1
UnZoom	1
Rotate viewing angle	1
Unconstrained Ray–Tracing	1
Scattering from Diffuse Surfaces	2
Single-Element Lens Scattering	3
Scattering and Multiple Bounces in a Reflective Conical Light Pipe	4
Stray Light Modeling	4
Radiometry	5
Radiometry with a Compound Reflector	5

Integrating Sphere with Nearly Lambertian Surface 6

Polygonally Faceted Objects 7

Polygon Scanner 7

Abbe Prism 8

Wedge-shaped Lightpipe 8

Building a Solid CPC 10

FILE 1

New 1

Open 1

Run Macro 1

Import CAD 2

IGES 2

STL 3

Import As Single CAD Object 3

Save 4

Save Model As 4

Save History (Macro) 5

Save CAD 5

Print.... 5

Print Preview.... 6

Print Setup.... 6

Save Metafile 6

SaveRaysAsText 6

SaveSpotsAsText 6

EditFile 7

Recent Files 7

EditLastMacro 7

ReRunLastMacro 7

ClearReRunMacro 8

Exit 8

EDIT 1

CopyText 1

CopyBitmap 1

Copy Metafile 1

PasteBitmap 1

Activate 1

Deactivate 2

Delete 2

Place 2

Coordinate Systems 3

Object Local Coordinate System 3

PlaceGlobal 4

PlaceRelative 5

PlaceInside 5

PlaceRotations 6

Rotate 7

Coordinate Systems 8

Object Local Coordinate System 8

Rotate Local 8
Rotate Global 9
Rotate Relative 9
Translate 10
Coordinate Systems 10
Object Local Coordinate System 10
Translate Local 10
Translate Global 11
Remove 12
Object Properties 12
VIEW 1
Type 1
Perspective 1
Orthographic 1
X or Y or Z 1
Film 2
Energy 2
View Radiometer 3
Polar Radiometer 3
Box 3
Rays 4
Radiometer Palettes 4
Monochrome 4
PolyNormal 5
Grid 5
Frame 5
HighLight 5
RaySegment 5
Facet/Surface 6
IGES Export as Linework 6
3D Wireframe OpenGL 7
3D Solid OpenGL 7
Display Radiometer with Open GL 8
Redraw 8
View Camera Array 9
DESIGN 1
Introduction 1
Physical Extent of Objects 1
Shell Objects 1
Volumetric Objects 2
T.I.R. Option 2
Shell Surface Reflectivity Options 2
Open Surfaces (Mirrors) 2
Closed Surfaces (Cavities) 2
Lightsource 2
Power Units 3
Reverse Ray-Trace Detector 3
Film 5

Rectangular 5
Oval 5
Axis 6
Mirror 6
Flat 6
Rectangular 7
Oval 7
Spherical 8
Spherical - Oval 9
Rectangular 9
Trough 10
Parabolic 11
Oval 12
Rectangular 13
CPC 13
Lens 14
Spherical lens 15
Two Surface Spherical lens 17
Cylindrical Lens 18
Catalog lenses 20
Aperture 20
Rectangular Extent 21
HoleRectang 21
HoleOval 21
Oval Extent 21
HoleOval 21
HRectang 21
Rod 22
Cavity 22
Ellipsoid 23
Cone 25
Cylinder 27
Eltrough 28
Prism 30
CornerCube 30
Polar Radiometer 31
Replicate 32
OBJECT and SURFACE PROPERTIES 1
OptiCAD Objects Box (OOb) 1
Object Renaming 2
Object and Surface Properties Box (OSPb) 2
Definition Page 3
Optical Materials List 4
Placement Page 4
Attributes Page 6
Surface Attributes 6
Note on Polynet Surface Properties 7
Selected Coating 7

Reflectance Functions Page 8
Reflectance Functions 8
Equation Based Spatial Reflectance Function 10
Bitmap Based Spatial Reflectance Function 10
Equation Based Angular Reflectance Function 10
Absorptance Functions Page 11
Absorptance Functions 11
Equation Based Spatial Absorptance Function 12
Bitmap Based Spatial Absorptance Function 12
Equation Based Angular Absorptance Function 13
Luminance Functions Page 14
Luminance Functions 14
Equation Based Spatial Luminance Function 15
Bitmap Based Spatial Luminance Function 15
Equation Based Angular Luminance Function 15
Treatment Page 16
Treatment 16
X_BEf or Y_BEf 16
X_Hyperbolic_BEf or Y_Hyperbolic_BEf 17
Conical surface structure 18
Spherical surface structure 19
Dots surface structure 20
Remove surface treatment 21
Reflective and Transmissive Polarization Pages 22
Polarization Properties 22
X_Analyzer 23
Y_Analyzer 23
X_Quarter Wave Plate 23
Y_Quarter Wave Plate 24
X_Half Wave Plate 24
Y_Half Wave Plate 24
Jones Matrix 24
Null Matrix 24
Remove Polarization Properties 25
AltSurfs Page 26
AltSurface 26
Iterative Intercept Method 26
Multiple AltSurfaces 26
Toric 27
Aspheric 28
Fresnel Lenses 30
Grating 31
Scattering Page 33
Types of Scattering Models 33
Post Surface Scattering 33
Perturbed Surface Scattering 33
Importance Sampling Scattering 34
Sobol Sampling 34

Scattering Functions 34
None 35
Lambertian 35
SN_Lambertian 35
Gaussian 36
SN_Gaussian 37
ABg 38
SN_ABg 38
PowerLaw 39
SN_PowerLaw 40
XY Exponential 41
SN_XY Exponential 42
BSDF tabular data based scattering 42
ASCII BSDF data format 43
Fresnel Page 44
SurfaceFresnel 45
None 45
Fresnel 45
Metal 46
LWIR Metal 47
Extent Page 49
Extent 49
SurfaceHole 50
None 50
Oval 51
Rectangular 51
Object Media Page 52
Volume Absorption 52
Volumetric Luminance 53
Bulk (Volumetric) Scattering 53
Gradient Index Material 55
Standard Gradient Index Material 55
Gradient index 1 Material 55
Gradient index 2 Material 56
Gradient index 3 Material 57
Gradient index 4 Material 57
Gradient index 5 Material 58
Gradient index 6 Material 59
Gradient index 7 Material 60
Gradient index 8 Material (GRADIUM) 61
Gradient index 9 Material (SELFOC) 62
Gradient index 10 Material 63
User Defined Gradient Index Material 64
Making a user defined DLL 64
DLL Data Array 65
Object Birefringence Page 67
(Extended) Source-Detectors Page 69
Custom/Extended Lightsources and Detectors 69

Lambertian Distribution 70
Lightsources versus Detectors 70
Detector Area 70
Predefined Extended Source–Detector 70
Cylindrical Emitting Surface Source–Detector 71
Flat Emitting Surface Source–Detector 71
Gaussian Surface Source 72
Multimode Gaussian Surface Source 73
Laser Diode Source 74
Laser Diode2 Source 75
Diode Linear Array Source 76
Cylindrical Emitting Volume Source–Detector 76
Ellipsoidal Emitting Volume Source–Detector 77
Isotropic Emitting Point Source–Detector 77
Toroidal Surface Source 78
Helical Surface Source 79
Sun Source 80
Spherical Surface Source–Detector 80
Measured Data Light Sources 81
IESNA Data Sources 81
User Specified Tabulated Data 82
Tabular Ray Data 83
OptiCAD Binary Format from Radiant Imaging 84
New ASCII Text Format 85
Older Text Data Format 85
Array Using 2-D Source Profile Data 86
User Defined Custom Source 87
Making a user defined DLL 87
DLL Data Array 88
Emitance Functions Page 90
Emitance Functions 90
Bitmap Based Spatial Emitance Function 91
Equation Based Angular Emitance Function 92
Data File Based Angular Emitance Function 92
Initial Polarization Page 93
Camera Page 94
Camera 94
Flat Camera 97
Spherical Camera 99
POLYNETS 1
General Discussion 1
T.I.R. Option 1
Implementation 2
Verification 2
Polynet Commands 3
Polynet - Start 3
Polynet - Polygon 4
Polynet - Chain 4

Polynet - Point 4
Polynet - Vertex 5
Polynet - Thick 5
Polynet - Close 6
Polynet - Finish 6
Modify 6
Polynet - Normalize 7
Example I - Extruded Polynet Object 8
Example II - General Polynet Object 8
Multiple Holes in a Polynet Object 9
ANALYZE 1
Raytrace 1
Sobol Sampling 2
Amplified Spontaneous Emissions (ASE) 2
Lightsource Raytracing 3
Detector Raytracing 3
Cone 4
FullAngle 4
Eqsolid 5
EqAngle 5
Tangent 6
Direction Cosine 6
Collimated 7
Rectangular 7
Oval 8
Extended 8
Rectangular 9
Oval 9
Function 10
Rectangular 10
Oval 11
Energy 11
XLinear 12
YLinear 13
Encircled 14
RepeatLast 15
Preferences 15
EnergyFile 16
Display 17
Rays 17
Spot 17
Spot - Directions 17
Spot - Locations 17
Object 17
Altsurfaces 17
Extents 17
Geometry 17
Surface 18

Radiometer 18
Conventional radiometer 18
Irradiance radiometer 19
Intensity radiometer 19
Radiance radiometer 19
Viewing a conventional radiometer 20
Polynet radiometer 20
Notes on using a radiometer 20
Attach Irradiance Radiometer 21
Attach Intensity Radiometer 21
Attach Radiance Radiometer 22
Minimum/Maximum Display Settings 23
Smoothing Setting 23
Clear (reset) 23
Remove 24
TextOut 24
SaveToFile 25
Rebin 25
View Radiometer 26

COATINGS AND POLARIZATION 1

INTRODUCTION 1

Coating definition 2

Materials and coatings provided with OptiCAD 3

Specifying coatings on surfaces 4

Removing a coating 4

Coating on glass 5

Coating on a shell 5

If no coating is specified on a glass surface 5

Surface reflectance is zero and fresnel is off 5

Surface reflectance is zero and global fresnel is on 5

Surface reflectance is not zero and surface specific fresnel is off 6

Surface specific fresnel is on 6

If no coating is specified on a shell surface 6

Surface reflectance is 1.0 and the fresnel metal is off 6

Surface reflectance is not 1.0 and the fresnel metal is off 6

Surface fresnel metal is on 6

Defining polarization sensitive explicit surface properties 6

Y Analyze 7

X Quarter Wave Plate 7

Y Quarter Wave Plate 8

X Half Wave Plate 8

Y Half Wave Plate 8

Null Matrix 8

Activating polarization 8

Defining the incident polarization 8

Modeling frustrated total internal reflection 9

Limitations of polarization analysis 10

Special considerations 11

OPTIONS 1
AutoRetrace 1
AutoRedraw 1
UpdateAll 2
SelectorMode 2
Point (Select Location) 2
SizeBox (Zoom In) 3
AzEl (Perspective View) 3
RayTraceLimits 3
Wavelengths - Weights - Primary Wavelength 4
Iteration 5
Proximity Glue 6
Fresnel Reflect 7
Add Rays 7
Save Rays 8
Save Spots 9
Reset 9
SphLensVertex 10
Preferences 10
PopUpOnError 10
NURBS CacheSizes 10
Polarization 11
Trace mode activation 11
Polarization state initialization 11
Sew CAD Objects 11
Sobol Sampling 12
T.I.R. User Defined Reflectance 12
WINDOW 1
Toolbar 1
StatusBar 1
ObjectBox 1
Snug Workbox 1
SetWorkbox 1
SetUnits 2
SetPowerUnits 2
SunPosition 3
Relative 3
Absolute 3
GridSnap 4
FletchRays 4
Linework Resolution 4
RayRainbow 4
Color Spots By Energy 4
Perspective 5
Stereo 3D (red/blue) 5
PlotScaling 5
PARAMETER STUDIES USING A SPREADSHEET APPLICATION 1
Interfacing with MS Excel 1

Creating an Input Macro with Excel 1
Displaying Output Data with Excel 1
INDEX 1

INTRODUCTION

What OptiCAD Is

OptiCAD is a computer program for the layout and analysis of three-dimensional optical systems. The program is structured in a computer-aided design (CAD) format in which the user defines objects, then places and orients them in a global-coordinate system. The program may be used to simulate a wide variety of nonimaging and imaging optical systems.

Components include spherical and cylindrical lenses; conical, parabolic, elliptical, spherical, and flat mirrors; full (closed) cylinders and ellipsoids; transmitting and reflecting rods; parabolic and elliptical troughs; single- and double-sided reflective surfaces; apertures of finite extent; lightpipes; multiple sources and detectors; compound parabolic concentrators; Fresnel lenses; and polygonally faceted objects (that may be translated from a CAD program).

All surfaces may be modified in reflectance and absorptance; oval or rectangular holes may be added in many surfaces; diffraction gratings and polynomial aspherics may be added to many surfaces.

OptiCAD has several forms of diffuse surface scattering that may be applied to any surface and a radiometry feature with no ray limit.

A powerful feature of OptiCAD is unconstrained (nonsequential) ray-tracing. Rays are not constrained as to the order in which components are intersected, nor to the number of times the same component may be intersected. The optical system is not constrained to any rotational symmetries. Total internal reflections (TIR) are handled automatically.

The results are presented as tabular output, 3-D graphical views of the system (with or without rays), ray-spot diagrams, encircled and enslitted energy plots, rendered images (solid model display), 2-D irradiance maps, and for some data in a form that can be imported into a spreadsheet program.

What OptiCAD Is Not

OptiCAD does not perform optimization; it is an analysis-only program. OptiCAD does not perform any diffraction calculations; it is a geometric ray-trace program.

References on Imaging and Nonimaging Optics

Author	Title	Publisher
W. Smith	Modern Optical Engineering	McGraw-Hill
R. Winston	Selected Papers on Nonimaging Optics (SPIE Milestone) (1995)	SPIE
R. S. Longhurst	Geometric and Physical Optics	John Wiley & Sons
Bass	Handbook of Optics	McGraw-Hill
J.W.T. Walsh	Photometry	Dover Publications
W. B. Elmer	Optical Design of Reflectors	TLA Lighting, Salem MA
	I.E.S. Lighting Handbook	Illum. Eng. Soc. of N. Amer.
J. B. Murdoch	Illumination Engineering	Visions Communications
J. T. Stover	Optical Scattering	SPIE Press

Most importantly, OptiCAD is no substitute for good engineering practices. No design should ever be considered finished until a qualified engineer has checked the calculations performed by the software to see if the results are reasonable. This is particularly important when a design is to be fabricated and significant costs are involved. It is the engineer's responsibility to check the results of OptiCAD, not the other way around.

Technical Support for OptiCAD

Technical support is provided for 90 days after purchase, and to those users with a current Service Upgrade Agreement. Technical support is only provided for features and functions documented in this manual. Users should report problems by e-mail. It is essential for us to have a written description of the problem. We can most efficiently solve problems when the user sends a short macro file demonstrating the suspect behavior.

Our e-mail address is support@opticalcad.com, our support-only phone number is 505-995-1999, and our Fax number is 505-988-9160.

System Requirements

To run OptiCAD you need Windows 2000, XP or later; Pentium, 128 Megabytes of RAM; 10 megabytes of available disk space, graphics board with a minimum of 256 colors (graphics accelerator board highly recommended), a CD ROM, activated parallel port, a hardware lock for OptiCAD, and a password that is specific to that hardware lock.

We recommend that **Regional Settings** (found on the **Control Panel**) be set to **English (United States)**.

Please read the section entitled Policy on the Hardware Lock.

If you are running Windows NT, you must have administrative privileges in order to install required drivers.

Installation Procedure

Before you install the OptiCAD software, please verify that you have the required hardware specified in the previous section.

If you are upgrading from a previous version of OptiCAD, we recommend that you first Uninstall OptiCAD using the Add/Remove Software feature of the Windows Control Panel.

Go to X:\OptiCAD (where X: is the drive that contains the CD ROM) and double-click on setup.exe.

You may wish to create a Shortcut to Opticad.exe. A Shortcut may be created by highlighting Opticad.exe with the left mouse button, clicking the right mouse button, and then selecting Shortcut. The Shortcut icon (a black prism with colored rays) may now be dragged to the main screen.

To start OptiCAD, double-click on the OptiCAD shortcut (or select Start → Programs → OptiCAD → OptiCAD Optical Analysis Program.

Installation Problems

On rare occasions an error message will appear, stating that the number of bytes copied does not agree with the expected value. If this occurs, you should click **Ignore** and proceed with the installation.

Uninstalling OptiCAD

When running the uninstall program you may get an error that not all files could be removed from the OptiCAD directory. This is because OptiCAD writes files to the OptiCAD directory, these may be deleted manually.

Registering OptiCAD

Please be aware that the OptiCAD software was sent to the address on the purchase order, which is not always your physical location. We need your physical address in order to send you upgrade notices, and the latest product information. Therefore, **please fill out and send in the registration card now.**

Updates of OptiCAD Software

After installing OptiCAD from the CD please go to www.opticad.com for the most recent version of the OptiCAD software.

Installing the Parallel Printer Port Hardware Key

OptiCAD is copy-protected by a hardware device called a hardware key (also called a lock). Attempting to run OptiCAD on more than one computer at a time, or attempting to defeat the intention of the hardware key is a violation of the owners' copyright, and is punishable by severe criminal and civil penalties.

The parallel port hardware key (supplied with the program) attaches to the parallel (printer) port at the rear of the computer. If you do not have a printer, place the hardware key in the port by itself. If you have a printer, disconnect the printer cable, then attach the hardware key to the port, then attach the printer cable to the hardware key. Your printer must be turned on for the hardware key to function properly. Your printer and all other computer operations will be unaffected by the presence of the hardware key.

If you have other Rainbow parallel port copy-protection devices installed on the same computer, you should install the OptiCAD copy-protection device closer to the computer.

You do not need to hook up a printer to the hardware key, but if you do, the printer must be turned on. If you do not have the hardware key installed correctly, OptiCAD will issue an error message and not allow the program to run.

Please read the section entitled Policy on the Hardware Lock.

Running OptiCAD

To run OptiCAD, double-click on the OptiCAD Shortcut (the user should create a shortcut if needed).

Policy on the Hardware Lock

OptiCAD is multiple-use protected by a device called a hardware lock. The hardware lock must be attached to the computer on which OptiCAD is running. The hardware lock prevents more than one copy of OptiCAD from running at one time. The registered user of the OptiCAD software is allowed to install the OptiCAD software on more than one computer as long as there is no possibility that more than one copy of OptiCAD will be running at one time. For example, the user is allowed to install the OptiCAD software on a computer at work and on another computer at home as long as the two installations of OptiCAD are never used at the same time. The hardware lock must be physically moved from one computer to another, allowing one copy of OptiCAD to run at a time.

The hardware lock supplied with the OptiCAD software is worth the full price of the software. If the hardware lock is lost or stolen, it will not be replaced without purchase of the full software package. Insure the hardware lock as you would any other business or personal asset of comparable value.

If the hardware lock ceases to function correctly, it will be replaced free of charge if returned to OptiCAD Corporation. Please call for authorization prior to returning any defective hardware lock.

Policy on Bug Fixes

All nontrivial computer programs have bugs. Every effort is made to find and fix bugs before a new version is released. However, the program is sufficiently complex such that even a talented team of prerelease program testers will not find every bug. OptiCAD Corporation therefore may provide periodic bug fix releases.

If you find an apparent bug, please report it to us. Try to isolate the exact actions that cause or lead up to the bug. Try to see if the error is unique to the optical file you are working with. Bugs are usually easier to find if they appear on the example files provided with the program.

If you find a bug that causes significant degradation of program performance within 60 days of purchasing the program, the bug fix will be provided free of charge. OptiCAD Corporation reserves the right to determine which bugs constitute significant degradation. After 60 days, an upgrade charge may apply; in this case you will receive the latest program version and manual.

Bug fixes are not provided if the problem is traced to incompatible, nonstandard, or insufficient hardware, or if you are using an outdated version of the operating system. OptiCAD is only guaranteed to work under Windows 95, Windows 98, or Windows NT. OptiCAD may not run correctly under OS/2 or other non-Microsoft operating systems, or on any non-Intel processor.

Please also see the section **What OptiCAD Is Not** earlier in this chapter.

Policy on Upgrades

Periodically you may receive an upgrade notification that will list the new features available. If you would like to upgrade from your current version of the program to the new version, you may do so for a fee. If you prefer, you can keep your current version and not purchase the upgrade. You may upgrade at any time, and are required to pay for only the latest upgrade; you do not need to buy every upgrade. Upgrades are only available on an annual subscription basis. The subscription provides for periodic upgrades which include new versions, manual, and technical support during the year the subscription is in force.

Users whose support is current should occasionally check the OptiCAD home pages for minor upgrades.

Version 9.2 New Features

Major new features have been added, for a list of new features see the file `OP_Release_Notes_9.2.pdf` in the **Release Notes** subdirectory.

Manual Notation

This section describes the format, convention, and notation used in this manual.

Menu Sequence Description

When describing a series of menu selections, options may be shown on subsequent lines, therefore:

Choice1
Choice2
Choice3

should be interpreted as select choice 1 from the top level menu, then select choice 2 from the next level menu, then select choice 3 from the final level menu.

When menu sequences are described in sentence form they are shown as Choice1 → Choice2 → Choice3.

Two options (such as on and off) shown on the same line, such as

on | off

should be interpreted as either clicking a feature to produce a check mark next to the choice (turning the feature on), or clicking a check-marked feature to uncheck the feature (turning the feature off).

A series of dots (.....) indicates that more options exist to further refine the selection or object definition.

Parameters are often shown with an underscore between words (e.g., `object_name`, `my_lens`, `conic_mirror`). This does not indicate a restriction on object or parameter names. Names do not require an underscore. Object names need not contain the word `object`. If you wish, you even are free to name a lens *mymirror*.

Menu Table Description

A table format is used to represent a screen menu, with text in approximately the same location as in the menu.

Text Command Description

When describing a text command containing several choices, options may be shown on subsequent lines, therefore:

```
Command 1  
Option2  
Option3  
Name  
Parameter1  
Parameter2
```

should be interpreted as a single line text command of the form:

```
command1/option2/option3 name parameter1 parameter2
```

Two options (such as `on` and `off`) shown on the same line, such as

```
on | off
```

should be interpreted as either appending the command line with `/on`

```
command1/option2/option3/on
```

(turning the feature on),

or appending the command line with `/off`

```
command1/option2/option3/off
```

(turning the feature off).

Default Options

Default options, which are generally described in the Options chapter, may be set for future sessions (saved) by editing the startup file `StartUp.opt`, which is automatically read whenever OptiCAD is started. A sample `StartUp.opt` is located in the `Samples` directory.

Unconstrained Ray-Tracing Methodology

A powerful feature of OptiCAD is the method it uses to compute ray paths through an optical system. Unlike conventional ray-tracing codes, OptiCAD rays are not constrained to follow a predetermined order (sequence) of ray-surface interceptions. Rays, originating from a source, travel in a user-defined direction, and fill a user-defined solid angle (up to a full sphere), with no constraints on the order that objects must be intercepted. OptiCAD considers all ray-surface intercepts and selects the closest real path. OptiCAD can model optical systems with discontinuous pupils, arrays of non-identical elements, total internal reflections, and multiple ray-surface intercepts.

Another powerful feature of OptiCAD is the user's ability to assign both a reflection and a transmission value to any surface. Because OptiCAD has a full ghost-ray capability, a single ray will result in a binary ray-tree, which contains all the possible descendant rays for any initial ray (called a parent ray). When a ray splits at a surface, the branch with the greater amount of energy is defined as the primary descendant, while the branch with the lesser amount of energy is defined as the ghost ray. If a ghost-ray encounters another binary ray intersection, a second generation ghost-ray (ghost of a ghost) is formed. In the unlikely event that all of the surfaces are both reflective and transmissive, a single ray would result in N descendants, where N is the total number of intersections (called the maximum ray depth) for a parent ray.

The user is able to specify the criteria for terminating a ray. A ray is terminated if it fails any of the following user-defined criteria: a) exceeds a maximum number of ray-surface intersections, b) exceeds a maximum number of ghosts, c) falls below a minimum ray energy transmission (the accumulated product of the transmissions and reflections along a ray path).

Ray Energy

Ray energy is generally conserved by OptiCAD. Each ray starts out with an energy of 1.0. The software keeps track of the energy of each ray as it propagates, and ray energy is reduced by surface and volumetric absorption.

When a parent ray divides (splits) at a partially transmitting and partially reflecting surface, the ray energy is distributed to the descendent rays.

Ray energy is arbitrarily lost only when a ray fails one of the user-specified tests set by the RayLim feature. In such cases (energy falls below a threshold, too many surfaces are hit, or too high a ghost order is encountered) the ray is truncated.

Energy is not conserved if the volumetric absorption is used as an exponential gain.

Coordinate System

There are three types of coordinate systems (all of which are left handed) used in OptiCAD. They are **global** (or system, or absolute) coordinates, in which objects are placed and in which the ray paths are displayed; the local coordinates of an **object**, and the local coordinates of a **surface**. There is only one global coordinate system, in it each object has its own local coordinate system, as does each surface.

The global location and orientation of a surface is determined by its relationship to its host object's coordinate system and by the object's relationship to the global coordinate system. When rotating an object, you are actually rotating the object's coordinate system.

Global Object Placement (default)

Objects may be placed globally, in which case the user specifies the location of an object's local coordinate system with respect to the global coordinates.

Relative Object Placement

Object placement may also be relative, in which case the user specifies the location of an object's local coordinate system with respect to the local coordinate system of any previously placed (reference) object. The location of the placed object is then transformed and stored in global coordinates. A relative placement does not link the objects together.

Inside Object Placement

An object may be placed inside another (reference) object. Placing an object inside another object overrides the program default assumption – that when a ray leaves an object it exits into air – the ray instead exits into the index of refraction of the reference object. For the inside placement, the user specifies the location of an object's local coordinate system with respect to the local coordinate system of any previously placed (reference) object. The location of the placed object is then transformed and stored in global coordinates, and the exit index assumption for the imbedded object is changed. With this one difference, an inside placement is the same as a relative placement. An inside placement does not link the objects together.

Local Object Rotation (default)

Objects may be rotated locally, in which case the user specifies the axis of rotation and the rotation angle that the object's local coordinate system is to be rotated about. The location and orientation of the rotated object is then transformed and stored in the global coordinate system.

Relative Object Rotation

Objects may be rotated relative to the local coordinate system of a placed object, in which case the user specifies the axis of rotation and the rotation angle that the object's local coordinate system is to be rotated about. The location and orientation of the rotated object is then transformed and stored in the global coordinates. A relative rotation does not link the objects together.

Global Object Rotation

Objects may be rotated globally, in which case the user specifies the axis of rotation and the rotation angle that the object's local coordinate system is to be rotated about.

Global Object Translation (default)

Objects may be translated globally, in which case the user specifies the translation of an object's local coordinate system with respect to the global coordinate system. The location and orientation of the translated object is then transformed and stored in the global coordinates.

Local Object Translation (default)

Objects may be translated locally, in which case the user specifies the translation of an object's local coordinate system with respect to its own coordinate system. The location and orientation of the translated object is then transformed and stored in the global coordinates.

OptiCAD Home Directory

The OptiCAD home directory is the directory that contains the executable optacad.exe. The following files use the home directory:

StartUp.opt

This macro is run (read) when OptiCAD is started; it contains user-defined settings and may be edited by the user.

Rays.dat and Spot.dat

These binary files (written by OptiCAD) contain ray and spot data. Both rays.dat and spot.dat are read by the View routine in order to display graphical views. Spot.dat is also read by the Energy routine in order to calculate encircled (and enslitted energy).

Opticad.log

This file (written by OptiCAD) contains a record of all operations and error messages of the current session.

Glass Files

The files used by OptiCAD to calculate the index of refraction of material end in an agf suffix.

Virus Checker

Users are strongly advised to install a virus checker on their computers and to keep current the data base of virus information.

Note on Examples

All graphics were produced using OptiCAD; in some cases, graphics were later touched up using Adobe Photoshop and Adobe Illustrator to ensure clear presentation in this printed manual.

Frequently Asked Questions

Is there any way to stop a ray trace once it's under way?

Hitting the escape key should stop most ray-traces.

How can I copy and paste from the TextOut box?

You cannot currently copy and paste directly from the TextOut box. You can however, copy the last text written into the TextOut box to the clipboard. Select Edit -> Copy Text, then you can paste the text into another application using cntrl+v.

I get an error message "WARNING: The decimal character. is not allowed in an alias" what does this mean?

The message that you get is probably because you are using an incorrect object name in the previous input line.

How do I model a coiled fiber with a cladding?

Define two coiled polynet objects, one smaller (the core) in diameter and (slightly shorter) in length than the other (the cladding). Use the `place/inside` feature to place the smaller one (the core) inside the larger one (the cladding). Make sure the inner fiber is completely inside the outer fiber.

How do I handle the double surfaces on the input and exit faces when I have one fiber inside another fiber?

Given that the core fits inside the cladding and is a little shorter, set the proximity_glue tolerance to be larger than the distance between the faces of the inner and outer polynet objects. Remember that the glue distance is ray path length, not an axial distance.

Rays leak out of my polygonally faceted object, what's going on?

Rays leaking out of a polygonally faceted object is mostly an aesthetic problem, as the leakage area on a polygon is insignificant. One way to minimize leaky rays is to rotate the light source by 1 degree. This avoids the problem of both a ray and a facet being in the x-y or y-z planes (which is the main cause of leakage).

How many polynet facets do I need to approximate a continuous object?

Too many facets isn't really a problem, though it slows things down a bit. The best way to test if you have enough facets is to keep increasing the mesh density (number of facets) until the result doesn't change.

Is there a way to make only one facet of a polynet object a scatterer?

You can save (export) the polynet object as a CAD file, then reimport the same file (as a NURBS object). You will then be able to assign a scattering function to any surface (facet face) of the object.

I ray-trace a polygonally faceted object and the rays go through the object without refracting, what's going on?

The problem is most likely that you have forgotten to add the line polynet/normalize poly1 (where poly1 is the name of your polynet object) after the polynet/finish command line. The polynet/normalize command will adjust the facet surface normals to all be pointing outward. If your polynet object is not closed the polynet/normalize command will produce an error message.

How do I copy the screen view into another application?

OptiCAD can copy graphic images to the Windows Clipboard, which can then be pasted into another application. Make the view or views that you want (including zooming in on detail) and then select View → CopyMetafile. Now, run the desired application choose (usually) Edit → Paste. See the documentation for the specific application for details. For some applications it may be necessary to save the metafile as a file (File → PrintMetafile) and then read the saved file into the application.

Make the view or views that you want (including zooming in on detail) and then select View -> CopyMetafile. Now start MS Word and select Paste Special, and paste the view into your document. For some other applications (such as Adobe Illustrator 7) it may be necessary to save the metafile as a file and then read the saved file into the application.

How do I combine the rays from multiple ray-traces?

Make sure that the option Add_Rays is off (unchecked), then perform the first ray-trace (this will clear data from previous ray-traces). Select (click) the option Add_Rays (a check mark will appear next to the menu item Add_Rays), perform all of the remaining ray-traces, and then deselect the option (click again) Add_Rays, which will again appear unchecked.

How do I make a cemented doublet or triplet?

You can create a cemented lens by first defining the individual elements, and then placing them a very small distance apart (a gap of 0.00001 would be a good choice). Next set the parameter maxdistance in the Proximity Glue feature (Options -> ProximityGlue or set/proximity _glue maxdistance) to be 10 times the gap between elements (0.0001 for the example). This will fill the gap between the elements with the index of the one of the elements. You may wish to shorten the thickness of the second (right) element by the gap, and in rare cases you may need to decrease the smaller radius so that the surfaces do not intersect.

Why are rays ignored or reflected from an internal surface of a doublet or triplet?

Two surfaces cannot occupy the same place. Make sure that there is a gap between the common surface of the glued elements, and make sure that the proximity glue is set to be greater than a ray path in the gap.

Why do things that used to work no longer work?

Great effort has been made to assure that this will never happen, but here are some things to try:

a) Exit OptiCAD then restart and rerun the file, sometimes this will reset things that were overlooked by the program.

b) Inspect the saved file, use a text editor or click on File -> EditFile and select the file name. Clean up the file by getting rid of multiple definitions and placements, keeping only the most recent version. Make sure that all definitions and placements include the object name and all other relevant information is present. Try to group all information about an object together. Make the file as clean and neat and simple as possible. Then save the file and try running it again.

I place a partially reflecting mirror or film in a block of glass – the transmitted rays do not refract properly?

The default condition in OptiCAD is that when a ray leaves an object it exits into air. When you place a partially reflecting mirror in a piece of glass without using the `place/inside` option, the transmitted ray refracts as if it was exiting into air, rather than continuing in the glass. The solution is to use the `place/inside` option when placing the mirror in the glass.

No rays are displayed even though I just performed a raytrace?

Make sure that **Rays** is checked on the **View** menu, if Rays is not checked click on Rays.

Rays do not split at a partially reflecting surface?

Select **Options -> Raytrace Limits** and increase the value for **ghosts**.

I trace some rays but I don't see any rays displayed on the screen?

Select **View** and make sure that there is a check mark in front of **Rays**, if not click on Rays.

When I select Run macro or Open, I don't see any mac or opt files displayed?

When I try to select a dll from the Object and Surface Properties box I no dlls are shown?

From **Windows** (not OptiCAD), open one of the **Windows** programs that enables you to display contents of directories (such as My Computer or Explorer). Select **View -> Options -> View** then select **View All Files**.

The output macro created by the Z to OP translator is not accepted by OptiCAD?

The Z to OP translator produces an error message "Type Mismatch"?

Go to the **Windows Control Panel** and select **Regional Settings** then **English (U.S.)**, then rerun the translator.

End of Introduction Chapter

DEFINITIONS

Terminology

Active object

An active object is an object that is located and oriented in the workspace (workbox) that can be displayed and ray-traced. When an object is placed, it is automatically activated. An object can be deactivated by unchecking the Activate box on the Placement page of the Object Properties box. A deactivated object keeps its position and orientation information, but it is not displayed and cannot be ray-traced. A deactivated object can be reactivated by checking the Activate box on the Placement page of the Object Properties box. An object that has not been placed or has been removed from the workbox is considered deactivated.

Placed object

A placed object is an object that is located and oriented in the workspace (workbox) that can be displayed and ray-traced. When an object is placed, it is automatically activated. A placed object can be removed from the workspace (workbox) using the Remove function (all position and orientation information is lost).

Default object

The default object (also called the selected object) is the object whose name appears in any menu box (such as Delete) requiring an object name. The user can override the displayed object name by typing in a new object name. However, the default object is the object whose name is used with functions that do not produce an input box (such as Deactivate and Activate).

An object can be made the default object by clicking on the object's name in the OptiCAD Objects box.

Enslitted energy

Enslitted energy is the energy that falls inside a linear slit as the slit is moved across a film plane. Enslitted energy is displayed as a function of the slit position.

Alias

These identifiers are generated internally within OptiCAD and should be ignored.

Extent

The extents of an object or a surface are the physical limits of that object or surface. As an example, a singlet spherical lens is limited by two spheres (radius1 and radius2) and also by a cylinder that defines the lens diameter. Rays falling outside of the extent of an object or surface will pass undeviated.

Descendant rays

When a ray encounters a surface having both reflectance and transmittance, the ray will divide (split) into two rays (unless one branch has too little energy). The resulting rays are named descendants of the original ray. The branch with the greater energy is named the primary descendant ray, and the branch with the lesser energy is named the secondary descendant (or more commonly the ghost) ray.

Ghost ray

See Descendant ray's description in this chapter.

Parent ray

A parent ray is the name given to a ray before it divides into descendant rays. See Descendant ray's description in this chapter.

Polynet

A polynet is a continuous grouping of individual flat facets (called polygons) which describe either an open surface or a closed object. See the Polynet chapter for additional definitions.

Fresnel reflection

A reflection from an entry or exit surface of a dielectric material (glass, plastic, or water), whose reflectance and transmittance values are based on the indices of refraction on both sides of the surface.

Total internal reflection (TIR)

A reflection that occurs at the entry or exit surface of a dielectric material (glass, plastic, or water) when the exit angle predicted by Snell's refraction equation would produce a physically unreal result. In such a case the ray is said to have exceeded the critical angle.

Oval extent

This term is used to describe an elliptical (outer or inner) extent of a surface. The term oval is used to avoid confusion between an elliptical extent and an ellipsoidal surface shape.

Define

This term is used interchangeably with "design" to describe the definition (creation) of an OptiCAD object.

Design

This term is used interchangeably with "define" to describe the definition (creation) of an OptiCAD object.

IGES

A format for interchanging data between CAD programs.

NURBS

Is a Non Uniform Rational B-spline Surface used to represent objects imported from a CAD program.

OptiCAD Home Directory

The directory containing the OptiCAD executable program.

List of Symbols

Symbol	Description
.igs	Extension for an IGES CAD file
.mac	Extension for a macro file - contains system creation history
.opt	Extension for a model file - contains current system parameters
.stl	Extension for an STL CAD file
.wdm	Extension for a demo macro file
a	Half size in x of a complete surface
a	Volumetric absorptance coefficient of a volume
absor	Absorptance of a surface
b	Half size in y of a complete surface
c	Curvature (1/radius of curvature)
c	Half size in z of a complete surface
cv	Curvature (1/radius of curvature)
diameter	Radial full size (twice the radius) of a lens
fl	focal length
hfov	Maximum full angle to be traced in x
hhwidth	Half width in x
hxcen	Displacement of a hole in x
hxhw	Half size of a hole in x
hycen	Displacement of a hole in y
hyhw	Half size of a hole in y
input_angle	Maximum acceptance half angle of a CPC
k	Conic constant
length	For a rod: Full size in z
length	For a trough: Full size in x
length	For an eltrough: Full size in x

Symbol	Description
n	Index of refraction
n_rays	Total number of rays to be traced
nx	Number of rays to be traced in x
ny	Number of rays to be traced in y
output_angle	Maximum allowable exit half angle of a CPC
θ_{max_x}	Maximum full angle to be traced in x
θ_{max_y}	Maximum full angle to be traced in y
r1	Radius of curvature of the 1st (left) surface of a lens
r1	Radius of curvature of the 1st (left) surface of a lens
radius	Radial half size (half the diameter)
radius_exit	Radius of the small end of a CPC
refl	Reflectance of a surface
thickness	Axial thickness of a lens
vfov	Maximum full angle to be traced in y
vhwidth	Half width in y
width	For a trough: Full size in y
width	For an eltrough: Full size in y
xangle	Rotation about the x axis
xcenter	Displacement in x
xhw	Half width in x
xoffset	Displacement in x

Symbol	Description
xsize	Half width in x
yangle	Rotation about the y axis
ycenter	Displacement in y
yhw	Half width in y
yoffset	Displacement in y
ysize	Half width in y
zangle	Rotation about the z axis
zmax	Maximum value in z for the actual surface
zmin	Minimum value in z for the actual surface

End of Definitions Chapter

USER INTERFACE

Introduction

OptiCAD has both a text-based input and a screen input. Both types of inputs are described in the manual. In general, the screen input is the desired form for initial configuration setup and almost all analyses, while the text-based input is best for parameter studies of complex configurations. Text-based inputs are rarely typed in by the user, but are generally run from a macro file. When a configuration is Saved, the results are saved as an ASCII file consisting of text-based commands. The results can be edited and rerun (using a toolbar button) as an OptiCAD macro. The Esc key will interrupt most ray-trace operations.

Objects are defined using the **Design** menu described in the **Design** chapter. Object and Surface properties may be modified using the **Object and Surface properties** box described in the **Objects Properties** chapter.

Program Functions

The summary of program functions that follows is grouped into basic categories. The summary does not show the choice of options in complete detail and does not show required numerical input; both of these are shown in detail in future chapters. The symbol | should be read as *or*.

Working Environment

This category includes general viewing and analysis conditions.

<u>Menu Selection</u>	<u>Text Selection</u>	<u>Description</u>
Window SetWorkbox	set workbox	Sets the range of the viewing region (workbox).
Window SnugWorkbox	set workbox snug	Automatically adjusts in the viewing window (workbox) to contain all active objects.
Options AutoRetrace on off	set autoretrace on off	Reruns last ray-trace after any change.
Options AutoRedraw on off	set auto redraw on off	Redraws screen view after any change.
Options UpdateAll	Not Available	Repeats the last ray-trace and redraws the view.
Options RayTraceLimits	set raylim	Sets the criteria for terminating a ray.
Options Wavelength	set wavelength	Sets wavelengths and weights.
Options Iteration	Not Available	Sets conditions for iterative surface intersection.
Options ProximityGlue	set proximity	Sets the criteria for gluing objects together.
Options FresnelReflect on off	set fresnelreflect on off	Activates Fresnel surface reflection.
Options AddRays on off	set add_rays on off	Sets the ray-trace file to stay open.
Options SaveRays on off	set saverays on off	Saves all ray—surface intersection data.
Options SaveSpots on off	set addspots on off	Saves all film plane location and energy data.

Options Polarization TraceModeActive on off	coating polarization on off	Activates polarization raytracing.
Options Polarization Initialization	coating polarization initialization	Initializes field for polarization raytracing.
View Wireframe Open GL on off	view opengl wireframe rays	Displays in Open GL
View Solid Open GL on off	view opengl all rays	Displays in Open GL
View Type x y z sys ortho	view x y z sys ortho	Selects screen view.
View Type Energy	view energy	Views a plot of the results of the last energy calculation.
View Type Film	view film	Views a spot diagram of the film plane.
View Rays on off	view rays x y z sys ortho	Selects screen view and display ray paths.
Not Available	echo/on	Echoes (repeats) commands to the TextOut box.
File New	clear	Clears all defined objects. <i>Does not reset the operating environment.</i>
File Save	Not Available	Saves the current optical system to a macro file.
File Print	Not Available	Prints the current window.
File Exit	Not Available	Exits the program.

Object Definition and Modification

The object definitions include creating, modifying, and deactivating specific objects, as well as running, saving, and editing entire macro files.

Objects are defined using the **Design** menu described in the **Design** chapter. Object and Surface properties may be modified using the **Object and Surface properties** box described in the **Object and Surface properties** chapter.

<u>Menu Selection</u>	<u>Text Selection</u>	<u>Description</u>
File RunMacro	macro name="file_name"	Runs a macro file.
File RerunLastMacro	Not Available	Reruns the last macro that has been run.
File Save	Not Available	Saves the current optical system to a macro file.
File EditFile	Not Available	Calls up a text editor in order to edit a file.
File EditLastMacro	Not Available	Calls up a text editor in order to edit the last macro file that has been run.
File ImportCAD	Not Available	Imports a generalized CAD file
File ImportAsSingleCADObject	Not Available	Imports a CAD file merged to a single object.
Design LightSource	define light	Creates a point source of light.
Design Film Rectangular Oval	define film rect oval	Creates a flat film plane.
Design Axis	define axis	Creates an axis (reference object).
Design Mirror Flat Spher Trough Parab CPC	define mirror flat spher trough parab	Creates a mirror with numerous options for the shape and physical extent.
Design Lens Spherical Cylinder	define lens spher cyl	Creates a lens.

Design Aperture Rectang Oval	define aperture rectang oval	Creates an aperture.
Design Rod	define rod	Creates a uniform index rod.
Design Cavity Ellipsoid Cone Cylinder Eltrough	define cavity ellipsoid cone cylinder eltrough	Creates a cavity (closed conic).
Not Available	polynet/start polynet/finish	Creates a polygonally faceted object.
Design Prism Corner Cube	Not Available	Creates a prism
Design Polar Radiometer XY XZ YZ	polynet/belt/xypolar place xypolarrad x y z radiometer/attach	Creates a Polar Radiometer
Edit Deactivate	deactivate	Deactivates an object without deleting or removing it.
Edit Activate	activate	Activates (reactivates) an object that has been deactivated.
Edit Delete	delete	Deletes (destroys) an object.

Object Location and Orientation

The object location functions include the location, orientation and translation of objects. These functions may be performed in either local coordinates, global coordinates, or relative (to another object) coordinates. The commands are shown in the following table.

<u>Menu Selection</u>	<u>Text Selection</u>	<u>Description</u>
Edit Place Global Relative Inside	place global relative inside	Locates an object in either global or relative coordinates, or locates an object "inside" another object's index and coordinate system.
Edit Rotate Global Local Relative	rotate global local relative	Rotates an object.
Edit Translate Local Global	translate	Translates an object.
Edit Remove	remove	Removes an object from the system (but does not destroy the object).

Ray-Trace Analysis

This section describes ray-trace analysis features.

<u>Menu Selection</u>	<u>Text Selection</u>	<u>Description</u>
Analyze RayTrace Cone	raytrace cone	Traces a bundle of rays originating from a previously defined and placed lightsource.
Analyze RayTrace Collimated	raytrace cone collimated	Traces a collimated bundle of rays passing through a previously defined and placed lightsource.
Analyze Energy x y Encircled	energy x y encircled	Sorts a film plane ray distribution into linear or radial ray energy bins, as "enslitted" or "encircled" energy distributions.
RepeatLastEnergy	Not Available	Toolbar button only feature to repeat the last energy calculation.
Analyze Radiometer Attach	radiometer attach file/name=	Attaches a 2-D array of bins to a film plane, bins energy is incremented by rays (not saved).

Tabulated Data

This section describes the displaying of tabulated data, including ray–trace related data, object and surface data, and system parameter data. Tabulated data can be seen in the Text Out box at the bottom of the screen. Energy output data and radiometry data can be sent to a file as comma-separated value data, which can then be used as input to a spreadsheet or other program.

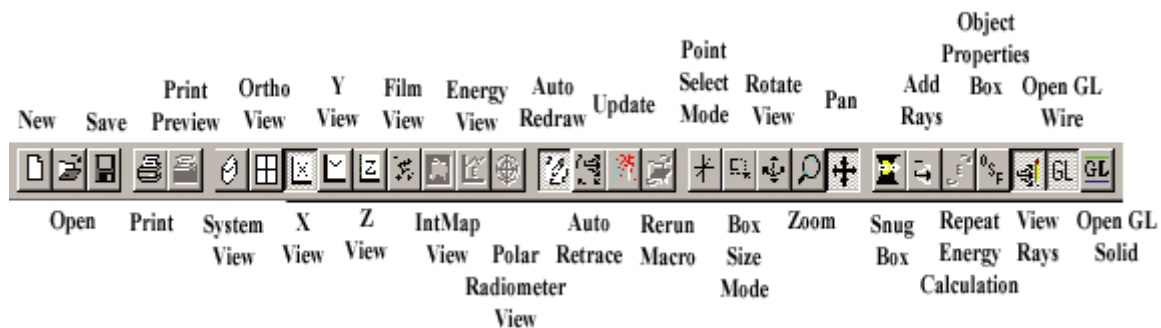
<u>Menu Selection</u>	<u>Text Selection</u>	<u>Description</u>
Analyze Display Rays	Not Available	Lists the position of every ray— surface intersection.
Analyze Display Spots	Not Available	Lists the location, direction cosines, and energy of every ray hitting the film (detector) plane.
Analyze Display Objects	Not Available	Lists objects and their status.
Analyze Display AltSurface	Not Available	Lists objects and their AltSurface status.
Analyze Display Extents	Not Available	Lists objects and their physical extents (limits).
Analyze Display Geometry	Not Available	Lists objects and their location and orientation (X Y and Z angles) in global coordinates.
Analyze Display Surface	Not Available	Lists surfaces and their reflec- tances and absorptances.
File SaveRays.....	Not Available	Saves tabulated ray–trace data to a file.
File SpotsRays.....	Not Available	Saves tabulated film (detector) plane data to a file.
Edit CopyText	Not Available	Copies the last text or tabulated data produced to the Windows clipboard.

Views and Plots

This section describes the display of graphical information.

<u>Menu Selection</u>	<u>Text Selection</u>	<u>Description</u>
View Type x y z sys ortho	view x y z sys ortho	Selects screen view.
View Solid	Not Available	Creates a rendered (shaded) image.
View Type Energy	view energy	Views a plot of the results of the last energy calculation.
View Type Film	view film	Views a spot diagram of the film plane.
View Rays on off	view rays x y z sys ortho	Selects screen view and display ray paths.

Toolbar Buttons



This section describes the OptiCAD toolbar (shown above). If the feature is described elsewhere in the manual, the appropriate reference (Chapter → Section) is shown in parentheses to the right of the toolbar button name.

Clear (File → New)

Deletes all objects, runs startup file.

Open (File → Open)

Runs a predefined macro file.

Save (File → Save)

Saves optical configuration.

Print (File → Print)

Standard Windows print routine.

Print Preview (File → Print Preview)

Standard Windows print preview routine.

View System (View → Type → Perspective)

Displays wire frame perspective view.

View Ortho (View → Type → Ortho)

Displays three orthogonal views and one perspective view.

View X or Y or Z (View → Type → X or Y or Z)

Displays a wire frame view looking into either the x, y, or z axis.

View Film (View → Type → Film)

Displays ray location (spot diagram) on film surface.

View Irradiance Map → View → Radiometer)

Displays an irradiance map from a radiometer.

Auto Redraw (Options → AutoRedraw)

Redraws view after any change.

Auto Retrace

Repeats last ray–trace operation after any change.

Update Ray Trace (Options → Update Ray Trace)

Repeats last ray–trace operation and redraws view after any change.

Rerun Macro (Files → RerunLastMacro)

Reruns last macro.

Point Select (Options → Point)

Displays coordinates of mouse cursor location.

Workbox Sizing (Options → SizeBox)

Zooms in on a subregion of the screen.

View Point (Options → AzEl)

Selects viewing location.

Zoom

Zooms in and out of the view.

Pan

Shifts the view.

Snug Workbox (Options → SnugWorkbox)

Snugs viewing box to fit objects.

Add Rays

Sets the ray data file to remain open.

Rerun Energy (Analysis → Energy → RepeatLast)

Repeats the last energy calculation.

OSPb (Edit → Object Properties)

Opens the **Object and Surface Properties** box.

View Rays (View → Rays)

Causes rays to be displayed.

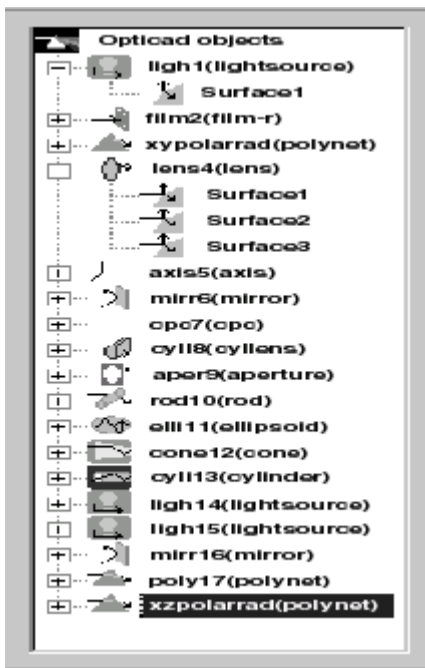
Open GL Wire (View → Open GL Wireframe)

Sets the view to wireframe Open GL.

Open GL Solid (View → Open GL Solid)

Sets the view to solid Open GL.

OptiCAD Objects Box



The OptiCAD Objects box (shown above) replaces an older box of the same name.

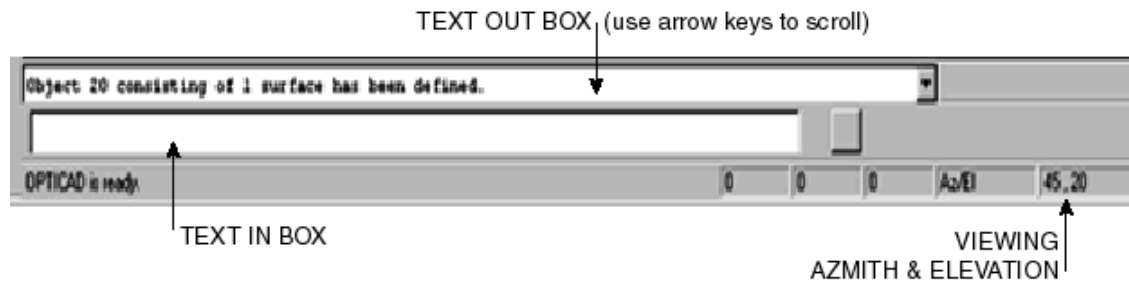
The OptiCAD Objects box displays a list of all objects. An object is selected by single-clicking on its name. The selected object name appears in all input boxes requiring an object name.

Object and Surface Properties Box

Double-clicking on an object name in the new OptiCAD Objects box will bring up the Object and Surface properties box. This box allows modification of: basic object shape, global object placement (and replacement), surface reflectivity and absorptance, surface scattering, and materials selection.

See the **Object Properties** chapter for details.

Boxes at Bottom of Window



This section describes the boxes at the bottom of the OptiCAD Window (shown above).

Text Out Box

This box contains OptiCAD output messages and tabular data. You may scroll through the text out box by using the up and down arrow keys.

Text In Box

This is an input box for typed in text commands. Text may be copied and pasted into the **Text In** box using the right mouse button. It is easier to enter sequences of text commands using a macro file.

View Azimuth & Elevation (View → Type → Perspective)

Displays orientation of view.

Object and Surface Names

Many OptiCAD functions require the user to enter an **object name** (such as the word `1ens`), or a **surface name** (also called a surface number). A surface name consists of an object name with a surface number appended (`1ens.2` for surface 2 of the object `1ens`).

Operations involving Place, Rotate, Translate, Remove, Delete, Activate, Deactivate, Design, Index, or Volumetric Absorbance require an object name (such as `1ens`). Operations involving Attributes, Holes, AltSurfaces, Extents, or Scattering require a surface name (such as the number appended to a word `1ens.1`). If an object name is entered when a surface name was required, the surface name will default to `objectname.1` (an entry of `1ens` is interpreted as `1ens.1`).

OptiCAD does not permit a blank space, or any of the following characters \ / ? # @ ! () { } [] in the name of an object.

Object Names

An object name can be comprised of lower case letters, numbers, and the underscore (`_`) character. An object name: must start with a letter, cannot contain a non-ASCII characters, cannot contain upper case characters, and must be less than 19 characters in length. OptiCAD does not permit a blank space, or any of the following characters \ / ? # @ ! () { } [] in the name of an object.

All objects should be assigned a name by the user. If you do not name an object the program will assign a name from a sequential list of names. Program-assigned names may produce confusion if independently created macros are run in the same session.

Object Name Examples

Object names are used to describe and distinguish between objects. Examples are:

`lens`, `lens1`, `lens2`, `relay_lens1`, and `relay_lens2`.

Surface Names

The surface naming (also called Surface Number or Object surface) convention is that the object is named `object_name`. For a single-surface object (such as a mirror or a cavity) the (only) surface is named `object_name.1`. For a single-element spherical lens (`objectname`), the left surface is named `objectname.1`, the right surface is named `objectname.2`, while the outer cylinder is named `objectname.3`. A cylindrical lens is named similarly with the four sides named `object_name.3` through `object_name.6`. When only the object name (`object_name`) is entered in a place that requires a surface name, the program will default to surface 1 (`object_name.1`). While this substitution is convenient for a mirror (which has only one surface), it can be a source of confusion and error for a lens.

Surface Name Examples

Surface names are used to describe and distinguish between surfaces, and are related to the object that includes the surface. Consider a single-element lens with three surface (left sphere, right sphere, and a cylinder). Examples of surface names for the left surface are:

`lens.1`, `lens1.1`, `lens2.1`, `relay_lens1.1`, and `relay_lens2.1`,

examples of surface names for the right surface are:

`lens.2`, `lens1.2`, `lens2.2`, `relay_lens1.2`, and `relay_lens2.2`.

Menu Object Name Input

The *selected* object may be set from the OptiCAD Objects box (located at the left of the screen). The *selected* object name will be used for features that do not allow explicit object name entries. Features that allow explicit object name entries will display the *selected* object name, which may be overwritten by the user.

An object name: must start with a letter, cannot contain a non-ASCII characters, cannot contain upper case characters, and must be less than 19 characters in length. OptiCAD does not permit a blank space, or any of the following characters `\ / | ? # @ ! () { } []` in the name of an object.

Numerical Input

The user is advised to be attentive to what numerical input is required: radius or focal length or curvature, full size (also called diameter, length, width, and size) or half size (also called half width). The units of numerical values may be in millimeters, centimeters, meters, or inches.

Menu Numerical Input

The user is advised to be attentive to the numerical input displayed in an input box before clicking "ok". If no changes are being made select "cancel".

When entering input in a menu, you can either click in each entry box or click in the first entry box and move between boxes using the keyboard tab key.

TextIn Box

The TextIn box is located at the center bottom of the screen, and it accepts text commands followed by a carriage return (Enter).

TextOut Box

The TextOut box is located near the center bottom of the screen, and it is opened by clicking in it. You may browse through the box by using the up and down keyboard arrows, clicking out of the box will close it.

Note that if the TextOut box appears full (or blank), you may wish to examine the optiCAD.log file located in the OptiCAD home directory. This file contains a complete history of the current session including error messages.

Error Messages

Error messages are written to the TextOut box (located near the center bottom of the screen) and to the file optiCAD.log (located in the OptiCAD home directory).

File Name Restriction

When referencing file names that contain blanks or special characters, the full name (including path if present) must be enclosed between double quotation marks (""). A file name may not contain the logical 'or' (|) character. The maximum length of a macro name (including the directory path and the quotation marks) is 250 characters.

Directory Name Restriction

When referencing directory names that contain blanks or special characters, the full name (including path if present) must be enclosed between double quotation marks (""). Directory names may not contain the logical 'or' (|) character. The maximum length of a macro name (including the directory path and the quotation marks) is 250 characters.

OptiCAD saved files will generally include the full explicit path for macros and data file, this may require some editing when moving macros to a different computer.

Macro Format

Text Based Command Structure

Text-based command line input is useful when running from a pre-defined macro file. A macro file may be created using a text editor or by saving an OptiCAD configuration using the menus File → Save or modified using an external text editor, or using File → Edit, or modified using File → EditLastMacro. A macro may be run using File → Run-Macro, or from another macro by using the text command

```
macro/name="macro_name" N
```

where macro_name is the name of the macro to be run and N is the number of times the macro is to be run. The maximum length of a macro name (including the directory path and the quotation marks) is 80 characters.

Running from a macro is useful when the optical system is complicated and when a configuration space is being explored by making small changes in system parameters. Running from a macro is also useful when an input file is created by an external application, such as a spreadsheet.

The general format of a text-based object- (or surface-) specific command is a command followed by options (also called modifiers), followed by an object or surface name, followed by numerical input. The options are separated by slashes, while object or surface names and numerical inputs are separated by blank spaces. A generic example is:

```
command/option1/option2/option3 name number1 number2 number3 number4 number5
```

and a specific example is:

```
define/lens mylens 100 -100 30 100 1.5
```

The user does not have to type an entire command or option word. The user only has to enter the minimum number of characters that make the command or option unique from all other commands or options. Currently five characters are enough to make a unique command or option. It is also noted that this shortened form of entry must be a truncation, not an abbreviation, of the command or option.

An example is:

```
defin/mirro/parab/recta my_mirror 100 100 50 50
```

instead of:

```
define/mirror/parabolic/rectangle my_mirror 100 100 50 50
```

When creating or editing a macro file, do not use a word processing package in a manner that inserts hidden control characters into the saved file. Examples of OptiCAD macros are shown in the demo files (having a WDM extension).

The entire command or option does not have to be entered; the first five characters (a truncation, not an abbreviation) are sufficient.

In many cases you may wish to begin a macro file with the `clear` command followed by the `echo/on` command. The `echo/on` command causes all macro commands to be displayed in the text out box, located at the bottom of the screen. The TextOut box is opened by clicking in the TextOut box, and explored by using the up and down keyboard arrows.

An alternative command line syntax will be found in some macros. It is of the form:

```
command/option1/option2/option3 name param1=num1 param2=num2 param3=num3
```

and a specific example is:

```
define/lens lens1 r1=100 r2=-100 thick=30 diam=100 n=1.5
```

and is identical to the example above.

Caution should be used when creating an OptiCAD macro, as spelling errors will result in an unmatched modifier. The result is that OptiCAD will substitute in a default modifier rather than give an error message.

When an object is defined from a menu, the menu box creates an input macro entry. The menu box has defaults, and it enters those defaults in the macro entry line it creates. However, a direct macro entry does not read the menu defaults. To avoid unpredictable results when directly writing a macro, you are advised to explicitly enter all of the modifiers (options).

The pound sign (#) indicates that the remainder of the line is a comment rather than an executable command. In some cases, especially when the comment line contains special characters, an error message will appear after the comment line. Comment lines should not exceed 80 characters.

Note that lines of text in a macro file should never include special characters and should never contain more than 250 characters.

Text File Formats

There are two types of saved text file formats:

The **Save Model** function creates a file with the extension `opt`, this file contains the current object properties and current object location and orientations.

The **Save History** function creates a file with the extension `mac` which contains the history of object definitions, redefinitions and the history of object placements, rotations, and translations; that result in the current object properties, location and orientation.

Files ending with `wdm` are macro files that can be run with the demonstration version of OptiCAD, these files are described in a companion file to the demo (`OPn_Demo.pdf`).

Text/Macro Only Commands

There are a few commands that can only be implemented as text commands and are only useful if run from a macro.

Pause

Function:

Temporary macro halt

Discussion:

The `pause` command causes a macro to temporarily stop, a pause box appears, and the macro continues after OK is clicked. Used in conjunction with various view options, `pause` allows time to view a plot or system. The `pause` command must be placed after a view command.

A pause command will display a text message in the pause box if the modifier `note` is used, the text cannot contain the logical 'or' (`|`) character, and must be entered between double quotation marks (`""`). See example 2 in this section.

Text Command Example 1:

```
view/x/rays
pause
```

causes a pause after the x-view appears.

Text Command Example 2:

pause/note="OptiCAD traces unconstrained rays"

displays a text message in the pause box.

Echo On

Function:

Echos every macro line

Discussion:

Sends a copy of every macro command to the TextOut box. Note that it does not check the command, it just echoes it.

Text Commands:

echo/on
echo/off

Reset Screen.out

Function:

Reset file screen.out

Discussion:

Option to reset the file screen.out when New or clear option/command

Text Commands:

set/reset_screenout/on
set/reset_screenout/off

Display Object absorptance

Function:

Displays absorptance for each object

Discussion:

Option to display in the TextOut box and scree.out file, the surface and volumetric absorptance for each object.

Text Commands:

set/absorbance_table/on
set/absorbance_table/off

Comment Character

Function:

Makes the remainder of a macro line nonoperational

Discussion:

The pound sign character (#) makes the remainder of a macro line into a comment (non-operational). The pound sign is useful for creating a comment line, appending a brief comment to a text command, and temporarily deactivating a text command (also called "commenting out a command"). Note that the entire line must not exceed 80 characters.

Text Command Examples:

```
# this macro is named doublet1.mac
define/lens/spher 10 -10 5 5 1.5 # third lens in system
# ray/col light1 10000 10000 1 1 # 100,000,000 rays will take a long time
```

Glass Catalogs

Function:

Loads a glass catalog

Discussion:

OptiCAD calculates the index of refraction only for materials that are included in glass catalogs that have been explicitly loaded into memory. This function loads a single glass catalog. Multiple glass catalogs may be loaded and used simultaneously. Index of refraction will only be calculated for wavelengths that are within the valid range for that material.

The user may wish to enter the commands for loading commonly used glass catalogs in their StartUp.mac file.

Note: Glass tables cannot be loaded from the menus and must be loaded using text commands or placed in the startup.opt file.

Parameters:

The name of a glass catalog.

Text Commands:

```
set/matcatalog/name="catalog_name.agf"
```

Text Command Examples:

```
set/matcatalog/name="corning.agf"
set/matcatalog/name="hoya.agf"
set/matcatalog/name="infrared.agf"
set/matcatalog/name="misc.agf"
set/matcatalog/name="ohara.agf"
set/matcatalog/name="old_corn.agf"
set/matcatalog/name="old_hoya.agf"
set/matcatalog/name="old_ohar.agf"
set/matcatalog/name="old_scho.agf"
set/matcatalog/name="schott.agf"
set/matcatalog/name="schottv.agf"
```

Glass Catalog Data File Format

Glass index of refraction dispersion data is stored in a file having the extension agf. User created data files and data files from other sources may be used as long as they contain the data lines read by OptiCAD. The format is as follows:

```
NM Glass_Name Dispersion_Formula_Number
CD a0 a1 a2 a3 a4 a5
LD Minimum_Wavelength Maximum_Wavelength
```

Where `Glass_Name` is the name of the glass in upper case characters,
`Dispersion_Formula_Number` is an integer number defining the dispersion formula to be used, `a0`, `a1`, `a2`, `a3`, `a4`, `a5` are the coefficients of the dispersion formula,
`Minimum_Wavelength`, `Maximum_Wavelength` are the minimum and maximum wavelengths for which an index of refraction will be calculated. Data lines beginning with characters other than NM, CD, or LD will be ignored.

Index of refraction dispersion formulas and their corresponding dispersion formula numbers are as follows:

Schott (1)

$$n^2 = a_0 + a_1 \cdot \lambda^2 + a_2 \cdot \lambda^{-2} + a_3 \cdot \lambda^{-4} + a_4 \cdot \lambda^{-6} + a_5 \cdot \lambda^{-8}$$

Sellmeier 1 (2)

$$n^2 - 1 = \frac{a_0 \cdot \lambda^2}{\lambda^2 - a_1} + \frac{a_2 \cdot \lambda^2}{\lambda^2 - a_3} + \frac{a_4 \cdot \lambda^2}{\lambda^2 - a_5}$$

Herzberger (3)

$$n = a_0 + a_1 \cdot L + a_2 \cdot L^2 + a_3 \cdot \lambda^2 + a_4 \cdot \lambda^4 + a_5 \cdot \lambda^6$$

$$L = \frac{1}{\lambda^2 - 0.028}$$

Sellmeier 2 (4)

$$n^2 - 1 = a_0 + \frac{a_1 \cdot \lambda^2}{\lambda^2 - a_2} + \frac{a_3}{\lambda^2 - a_4}$$

Conrady (5)

$$n = a_0 + \frac{a_1}{\lambda} + \frac{a_2}{\lambda^{3.5}}$$

Display Colors

Function:

Changes display (screen) colors

Discussion:

This text only feature allows the user to remap the colors used to display line drawings on the screen. Colors may vary between systems, using this feature will require some experimentation on the part of the user. The user may remap any of 16 colors to other color values.

The command `set/color/value param_1 param_2` temporarily assigns the color associated with `param_2` to also be the color associated with `param_1`. The `set/color/value` command should be included in the users startup file.

The color values for OptiCAD objects are listed in the table below.

Frame	15
Ray	14
Descendent	7
Prism	12
Lens	11
Mirror	10
Box	9
Grid	6

Example:

```
set/color/value 0 15
```

causes all program references to color 0 to instead use color 15.

The command `set/color/mapped` displays the remapping, and the command `set/color/reset` resets the colors to their default values.

Note that the system palette varies between computers. User's wishing to change the color scheme, should first display an object and then sequentially substitute colors between 0 and 12 for the number associated with that object in the table above.

Seeding Random Number Generator

Function:

Seeds the random number generator for scattering surfaces

Discussion:

This text only feature allows the user to enter a seed for the random number generator used for surface scattering. A value of zero seeds the random number generator to the system clock, a nonzero integer value seeds the random number generator to the absolute value of the entered value.

The value entered is used until it is changed or the program is restarted.

Text command

```
set/shuffle integer_value
```

Text command Example

```
set/shuffle 1
```

Macro Content Restrictions

Do not include and special characters in a macro, use only numbers and letters.

Old Macros

Older macros may not contain named objects, and may contain obsolete DOS commands. The user should edit all older macros and add object names if needed. Note that only macros referenced in this manual are supported by Opticad Corporation.

(Old) Solid Model

The function **View -> Solid (Solid Model Display Tool)** found in previous versions of OptiCAD has been replaced with the Open GL Solid option, and is no longer used.

Translators

Translators, allowing the importation of files created in other programs are located in the Utilities subdirectory, along with a pdf file describing the use of the translator.

End of User Interface Chapter

BASIC TUTORIAL

Introduction

OptiCAD is structured using objects. The user defines an object, then places and orients the object in a workspace (workbox).

Everything is an object: a point source of light (source of rays), a detector (film) plane, a mirror (infinitely thin surface), a single-element lens, a polygonally faceted surface or solid, and an aperture.

Most objects have a finite size (extent), which must be explicitly defined.

Every object has a name, and every surface within an object has a name that combines the object name with the surface number.

If a surface name is required, and an object name is entered without an appended surface number, then the surface number will default to 1.

For an object to be ray-traced, it must first be placed in the workspace (workbox). Objects that have not been placed will be ignored by the ray-trace.

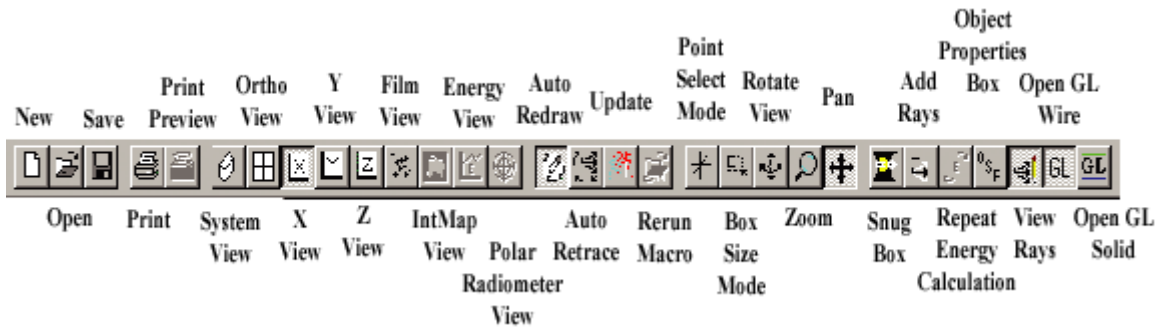
OptiCAD traces rays nonsequentially; rays are launched and may either hit or miss objects. Rays are not constrained as to the order in which components are intersected, nor to the number of times the same component may be intersected. The optical system is not constrained to any rotational symmetries. Total internal reflections (TIR) are handled automatically.

Note that the OptiCAD nomenclature uses the terms xhw (x half-width) and hhwidth (horizontal half width); also the terms yhw (y half-width) and vhwidth (vertical half width); the terms xfov (x field of view) and hfov (horizontal field of view) interchangeably; and the terms yfov (y field of view) and vfov (vertical field of view); and uses the term object name and surface name (a surface name is an object name with an appended surface number).

Before doing any of the exercises

- a) Read these chapters: **Introduction**, **User Interface**, **Design**, and **Object and Surface Properties**.
- b) Read the **Description of the Toolbar Buttons** section in this chapter.
- c) Read the **Procedure to Run a Macro** section in this chapter.
- d) Make sure that the AutoRedraw toolbar button is depressed (on), and make sure that the AutoRetrace toolbar button is not depressed (off).
- e) The exercises in this chapter assume the status of some environmental settings. Users should first run the macro SetTutorial.mac before proceeding.
- f) For most of these exercises the best view will be obtained by clicking on the Ortho View toolbar button and then on the SnugWorkbox toolbar button.
- g) Make sure that **Rays** is check marked on the **View** menu, if it is not (no rays will be displayed) click on **Rays**.

Description of the Toolbar Buttons



Clear	Deletes all objects.
Open	Opens a saved file.
Save	Saves optical configuration.
Print	Windows print routine.
Print Preview	Windows print preview routine.
View System	Displays wire frame perspective view.
View Ortho	Displays three orthogonal views and one perspective view.
View x or y or z	Displays a wire frame view in x, y, or z.
View Spots	Displays ray spot diagram.
View Intensity Map	Displays an irradiance map from a radiometer.
View Energy	Displays a plot of energy results.
Auto Redraw	Redraws view after any change.
Auto Retrace	Repeats last ray–trace after any change.
Update Trace	Repeats last ray–trace.
Rerun Macro	Reruns last macro.

Point Select	Displays coordinates of mouse cursor location.
Workbox Sizing	Zooms in on a sub-region of the screen.
View Point	Selects viewing location.
Snug Workbox	Snugs viewing box to fit objects.
Add Rays	Adds future ray-traces.
Rerun Energy	Repeats last energy calculation.
OSPb	Opens the Object and Surface Properties box
GL Wireframe	Selects Wireframe Open GL Graphics
GL Solid	Selects Solid Open GL Graphics

Screen Operations

Zoom in on detail of an X, Y, or Z view

In order to zoom in on detail in an X, Y, or Z view, select the Zoom In toolbar button, place the cursor on the screen and with the left mouse button depressed drag out a rectangle enclosing detail of interest.

Resize the workbox to enclose all of the active objects

In order to resize the workbox enclosing all of the active objects, select the Snug Workbox toolbar button.

Select an object as the (selected) active object

In order to select an object (making it the active or selected object), click on the object name in the OptiCAD Objects box at the left of the screen.

Select a surface as the (selected) active surface

In order to select a surface (making it the active or selected surface), double-click on the object name in the OptiCAD Objects box at the left of the screen (which will then expand - listing all of the surfaces contained in the object), then click on the surface number.

Open the Object and Surface Properties box

In order to open the Object and Surface Properties box and have it show the parameters of a particular object, double-click on the object name in the OptiCAD Objects box at the left of the screen.

Menu Operations

File	Edit	View	Design	Modify	Analyze	Options	Window
------	------	------	--------	--------	---------	---------	--------

Run a macro file

In order to run a macro, select File from the menu bar and then select RunMacro. Then find and double-click on the name of the macro.

Edit a macro file

In order to edit a macro file, select File from the menu bar and then select EditFile. Then

find and double-click on the name of the macro.

Edit the last macro that was run

In order to edit the last macro was run, select File from the menu bar and then select Edit-LastMacro.

Toolbar Operations

Moving the cursor to a toolbar will produce an automatic description of the toolbar function.

ReRun the last macro that was run

In order to rerun the last macro was run, click on the ReRunLastMacro toolbar button.

Repeat the last ray–trace performed

In order to repeat the last ray–trace that was run, click on the UpdateTrace toolbar button.

Change the screen view

In order to select a screen view, click on either the System View, Ortho View, X View, Y View, or Z ViewTrace toolbar button.

View the contents of a Radiometer (irradiance map)

In order to view the contents of a Radiometer, select the film object containing the Radiometer of interest, then click on the ViewIntMap button.

Section A

Exercise 1 Defining and Placing a Point Source of Rays

In this exercise we will create an object (consisting of a point source of rays), place the object, and trace a ray.

An object name may be a combination of lower case letters, numbers, and the underscore (_) character. An object name: must start with a letter, cannot contain a blank, cannot contain a special character, cannot contain a nonASCII characters, cannot contain upper case characters, and must be less than 19 characters in length.

Clear existing objects and reset OptiCAD

Select **File** then select **New** from the drop–down menu (**File** → **New**).

Click on the **View Ortho** toolbar button (in order to display into four views).

The **View Ortho** toolbar button is located to the left of the button containing an x.

Define a point source of rays

Select **Design** from the menu bar.

Then select **Lightsource** from the drop–down menu (**Design** → **Lightsource**).

Enter `my_light` for the object name, and click OK.

The **Object and Surface Properties box (OSPb)** appears.

You have now defined (created) an OptiCAD object.

But where is it and what does it look like?

The object `my_light` is now listed in the **OptiCAD Objects box (OOb)**, which is located on

the left side of the screen. You may click on the **OptiCAD Objects box (OOb)** and scroll through it using the up and down keyboard arrows.

OK, we defined an object, but we ask again: where is it and what does it look like?

Place an object using the **Object and Surface Properties box (OSPb)**

In order to use or see an object, you must first define it and then place it in the workspace (workbox).

Click on the **Placement page of the Object and Surface Properties box.**

Enter a 0 for x, 0 for y, and 0 for z (locations), and click **Place and Activate.**

Slide the **OSPb** out of the way.

What appears now on the screen is a small arrow.

The lightsource that we have designed and placed has no physical extent, but it is more than a point in space. The lightsource has an orientation. It has a local coordinate system, that has been initially placed at (0, 0, 0) and oriented parallel to the global coordinate system of the workspace. In the case of a lightsource, it is only a local coordinate system, because it has no extent.

All objects have a local coordinate system. When you place an object, you are placing its coordinate system relative to an existing coordinate system. Such a placement is then transformed (if needed) to the global coordinate system.

Trace a ray

We will now trace a ray.

Select **Analyze → Raytrace.**

Then select **Cone → Fullangle.**

Once again the name **my_light** appears below the heading Object name.

Enter 1 for nx, 1 for ny, 1 for hfov (xfov), and 1 for vfov (yfov).

Then click OK.

RayConeFullAngle			
		Object name	Cancel
		my_light	OK
nx	1		
ny	1		
hfov	1		
vfov	1		

We now see a single ray on the screen.

You can now see that the single ray that you traced is traveling along the z-axis (in the + z direction).

Review

Everything (including a point source of rays) is an object.

Every object has a local coordinate system.

In order for an object to be useful, it must first be defined, and then placed in the workspace.

Placing an object consists of defining the location of the object's local coordinate system with respect to the global coordinate system of the workspace.

Exercise 2 Rotating a Point Source of Rays

In this exercise (which continues from the previous exercise) we will rotate a lightsource (object), and trace several bundles of rays.

Let us now rotate the lightsource such that it points in the - z direction (rather than the + z direction).

Select **Edit** → **Rotate**

Select Local (the type of rotation), and x (the axis of rotation).

Enter 180 for angle (angle of rotation), and click OK.

Rotate			
Global	<input checked="" type="radio"/> Local		Relative
<input checked="" type="radio"/> x	y		z
		Object	my_light
		Angle (Degrees)	180
		Reference Object	
	OK	Cancel	

You have changed the orientation of the lightsource, but nothing has changed on the screen.

What's going on?

What you see on the screen is the location and orientation of the object (lightsource), after you changed it. This is because you made sure that the AutoRedraw feature was turned on, at the beginning of the exercises.

Repeat a ray–trace

You must trace some rays to see the effect on the rays of a configuration change.

OptiCAD does not automatically retrace rays, unless the AutoRetrace feature (which you made sure was off at the beginning of the exercises) is turned on.

Click on the **Update All** toolbar button (looks like fireworks).

You have now repeated the previous ray–trace.

The ray should now be traveling in the - z direction.

Trace a fans of rays

Repeat steps in the **Tracing a ray** section of exercise 1.

Analyze → **RayTrace** → **Cone** → **Full Angle**.

But this time enter 1 for the number of rays in x (nx), 20 for the number of rays in y (ny).

Enter 1 for the full angle in x, and 180 for the full angle in y, and click OK.

Trace another fan of rays

Repeat steps in the **Tracing a ray** section of exercise 1.

Analyze → **RayTrace** → **Cone** → **Full Angle**.

But this time enter 20 for the number of rays in x (nx), 1 for the number of rays in y (ny).

Enter 360 for the full angle in x, and 1 for the full angle in y, and click OK.

Review

A placed object (including a point source of rays) may be rotated.

Various ray bundles may be traced from a lightsource.

Exercise 3 Defining and Placing a Mirror

In this exercise (which continues from the previous exercise) let us now define something for the rays to hit.

Define a mirror

Select **Design** → **Mirror**.

Then select **Parabolic** → **Oval**.

Enter `parab_1` for the object name and click OK.

Then select **Both** and click OK.

The **Object and Surface Properties box (OSPb)** will appear.

Click on the **Definition page of the Object and Surface Properties box**.

Enter 10 for flx (focal length in x), 10 for yfl (focal length in y), 100 for xhw (half size in x), 100 for yhw (vhwidth) (half size in y), then click **Apply Now**.

Place the object using the **Placement page of the Object and Surface Properties box**

Click on the **Placement page of the Object and Surface Properties box**, which will display the object name `parab_1`.

Enter 0 for x, 0 for y, and -10 for z, then click **Place and Activate**.

Slide the **OSPb** out of the way.

Click the **SnugWorkbox** toolbar button (looks like an hour glass).

Repeat the ray–trace

Click on the **Update All** toolbar button (looks like fireworks).

The results should show most of the rays reflecting off of (and being collimated by) the parabolic mirror.

Review

A mirror is an object consisting of a single (infinitely thin) surface. This type of object is a shell, with no volumetric content.

A mirror has a finite extent (size), which must be explicitly entered.

Rays within the extent of an object intersect the object and may be altered by the object (in this case reflected).

Rays outside of the extent of an object miss the object and are unaffected (undeviated) by the object.

Exercise 4 Defining and Placing a Lens

In the previous exercise we defined an object consisting of a single surface (mirror). For multiple-surface objects, representing a volume, the concept of extent is more complicated than for shell objects.

We have previously learned that all objects have names. We will now show that surfaces also have names, and that a surface name is a concatenation of an object name and a surface number. This naming convention allows us to modify properties of an individual surface within an object.

In this exercise (which continues from the previous exercise) we will now define a single-element spherical lens, which is an object consisting of three surfaces.

Define a spherical lens

Select **Design** → **Lens** → **Spherical**.

Enter **lens1** for the object name and click **OK**.

Click on the **Definition page of the Object and Surface Properties box**.

Enter 250 for r_1 , 0 for r_2 , 30 for thickness, 200 for diameter, and 1.5 for n (the index of refraction), then click **Apply Now**.

When 0 is entered for a lens radius it is interpreted as a curvature of 0 (a plano surface).

Place the lens

Click on the **Placement page of the OSPb**.

Enter 0 for x , 0 for y , and 250 for z , then click **Place and Activate**.

Slide the **OSPb** out of the way.

Click the **SnugWorkbox** toolbar button.

Repeat the ray-trace

Click on the **Update All** toolbar button.

You should see the lens focus the rays collimated by the parabolic mirror. The lens also deviates the rays that missed the parabolic mirror.

Review

A multiple-surface object (single-element lens) can be defined.

The object is defined by specifying its conventional optical properties (r_1 , r_2 , thickness, and

index of refraction), and its extent properties (the diameter parameter defines a cylinder that limits the extent of r1 and r2.

Exercise 5 Rotating a Lens and Zooming in on a View

A solid object is represented differently in OptiCAD than in a conventional lens design program. In this exercise (which continues from the previous exercise) let us do an experiment to demonstrate that the spherical lens defined in the previous section is a true solid object.

Rotate the lens sideways

Follow the procedures in the **Rotating an object** section of exercise 2, and this time rotate the lens (**lens1**) about its local x axis by 105 degrees.

Repeat the ray-trace

Click on the **Update All** toolbar button.

The results are strange indeed, but they do demonstrate that the extent limiting cylinder is a real surface.

Zoom in to examine detail

Select the **Workbox Sizing Mode** toolbar button (a solid square in the upper left corner of a dotted square).

Place the cursor above and to the left of the lens (in either the x, y, or z views, but not in the perspective view). While keeping the left mouse button depressed, drag the cursor to below and to the right of the lens. Release the mouse button.

The results should be a zoomed-in view of the lens.

Zoom out to see all the objects

Click on the **Snug Workbox** toolbar button which should zoom out the view to display all of the objects.

Reverse the lens rotation

Follow the procedures in the **Rotating an object** section of exercise 2, and this time rotate the lens (**lens1**) about its local x axis by -105 degrees.

This brings back the system defined in exercise 4.

Verify that you have restored the system.

Click on the **Update All** toolbar button.

Review

A single-element lens consists of three real surfaces that define both the conventional optical properties and the extent of the object.

Section B

If you are continuing from Section A, you may just keep going.

If you are starting here at Section B you may get to this point by running the macro **A_Basic_Tutorial.mac**, which will bring you to the same point as if you had completed Section A.

Optional Procedure to Run the Macro A_Basic_Tutorial.mac

In order to run the macro A_Basic_Tutorial.mac, select **File** from the menu bar, and then select **RunMacro**.

Then find and double click the macro A_Basic_Tutorial.mac.

Exercise 6 Defining and Placing a Film Plane

We have some rays that start out from a point source, are collimated by a parabolic mirror, and are then focused by a lens. Let us define a film (receiver or detector) plane, and then examine the focused rays.

Define a film plane

Select **Design** → **Film** → **Rectangular**.

Enter **film1** for the object name and click **OK**.

Click on the **Definition page of the Object and Surface Properties box**.

Enter 10 for xhw (hhwidth) (the half size in x), 10 for yhw (vhwidth) (the half size in y), then click **Apply Now**.

Place the film

Click on the **Placement page of the OSPb**.

Enter 0 for x, 0 for y, and 760 for z, then click **Place and Activate**.

Slide the **OSPb** out of the way.

Click the **SnugWorkbox** toolbar button.

Repeat the ray-trace

Click on the **Update All** toolbar button.

The results should show rays focused on the film plane.

View the film (spot diagram)

Click on the **View Film** toolbar button (bunch of dots - to the right of Z).

You should now see the ray pattern on the film plane.

View the objects and rays

Click on the **View Ortho** toolbar button.

This should again produce three orthogonal views and one perspective view on the screen.

Review

A film (or detector) plane is an object of finite extent. It must be defined and placed in order to intercept rays.

Exercise 7 Modifying a Surface Shape - Understanding Surface Names

In the previous exercise we had collimated light incident on a convex-plano lens. The resulting focus was far from perfect.

In this exercise (which continues from the previous exercise) let us modify the first surface of the lens such that the rays all focus to a very small spot.

This is accomplished by setting up the same lens in an optical design program, and minimizing the r.m.s. spot size by varying the conic constant and the fourth order aspheric term of the first surface.

The results are:

conic constant = - 0.64

fourth order term = 4.988E-10

We will enter these modifications into our lens.

But first we have to understand how OptiCAD names surfaces.

Every surface has a surface name, and the surface name is a concatenation of the object name and the surface number, with only a dot (period) between them.

Generically this is **object.surface**, where object is the object name and surface is the surface number.

In the case of an object that contains only one surface, that surface is surface one. As an example the name of the parabolic surface in section A is

parab_1.1

which is a concatenation of the object name which is **parab_1** plus a dot plus the surface number which is **1**.

In the case of a single-element spherical lens of round extent, the object contains three surfaces. These surfaces are: the first (left) surface which is surface 1, the rear (right) surface which is surface 2, and the extent defining cylindrical surface which is surface 3.

In the case of an object named **lens1**:

the front (left) surface is named **lens1.1**

the rear (right) surface is named **lens1.2**

and the cylinder is named **lens1.3**

The surface that we will modify is named **lens1.1**.

Modify a surface shape

Double-click on the name **lens1** in the **OOb**.

Click on **surface1** of **lens1** in the (now expanded) **OOb**.

Click on the **AltSurfs** tab of the **OSPb**.

Make sure that the name **lens1.1** is displayed on the **AltSurfs** page of the **OSPb**.

Select **Aspheric**.

Enter 0.004 for cv (one over the radius of 250).

Enter - 0.64 for k (the conic constant), 4.988E-10 for a1 (4th order term).

Enter 0 for a2 (6th order term), 0 for a3 (8th order term).

Enter 0 for a4 (10th order term) and click **Apply Now**.

Slide the **OSPb** out of the way.

Repeat the ray–trace

Click on the **Update All** toolbar button.

View the rays on the film plane (spot diagram)

Click on the **View Film** toolbar button.

The results should be a very small spot of focused rays.

Review

Surface names are a concatenation of object name and surface number separated by a dot.

A surface shape can be modified to be aspheric.

It is better to perform lens shape optimization on a lens design (automatic optimization) program and bring the resulting lens shape into OptiCAD.

Exercise 8 Zooming in on the Film Plane

In exercise 5 we zoomed in on details of a view. In this exercise (which continues from the previous exercise) let us now zoom in on details of the film plane.

Zoom in on the film plane

While viewing the film select the **Workbox Sizing Mode** toolbar button (a solid square in the upper left corner of a dotted square).

Place the cursor above and to the left of the ray pattern. While keeping the left mouse button depressed, drag the cursor to below and to the right of the ray pattern. Release the mouse button. The results should be a zoomed in view of the ray pattern.

Repeat the above procedure until the ray pattern fills the screen.

The results show that the optimized lens produces a very small focused spot.

In order to unzoom the film plane view, move the cursor to the edge of the screen and click once.

View the objects and rays

Now click on the **View Ortho** toolbar button.

Review

You can zoom in on a region of the film plane, using the mouse.

Exercise 9 Modifying Surface Reflectance and Setting Ray–Trace Limits

There are two basic types of surface reflectance, angle of incidence independent reflection and angle of incidence dependent reflection.

A angle independent reflectance can be assigned to any surface. A ray will be reflected, transmitted and absorbed, depending on the numbers entered. If there is sufficient energy in both the transmitted and reflected ray, the ray will split into two (descendent) rays. The reflected, transmitted, and absorbed energy is not dependent on the angle of incidence that the ray makes with the surface.

An angle dependent reflectance can be assigned to any surface for which either side of the surface is glass, or metal (non-unity index of refraction). For the case of glass, the indices of refraction of both materials are used to calculate the average of the angle of incidence

dependent p and s polarization reflectance and transmittance. For the case of metal the user must enter the complex indices for both materials.

In this exercise (which continues from the previous exercise) we will show how to make one angle independent and one angle dependent type of surface reflectance modification. The differences in the resulting ray-traces are only apparent after careful radiometric analysis.

Modify an angle invariant surface reflectance

Double click on the name **lens1** (if needed) in the **OOB** in order to expand the name and display a list of surfaces.

Click on **surface1** in the **OOB** in order to open the **OSPb**.

The surfaces appear as a subset of the object name **lens1**.

Click on the **Attributes** page of the **OSPb**.

Make sure that the name **lens1.1** is displayed on the **Attributes** page.

Enter 0.5 for refl, 0 for absor, then click **Apply Now**.

Slide the **OSPb** out of the way.

Repeat the ray-trace

Click on the **Update All** toolbar button.

There are too many rays to see what is going on.

Trace fewer rays

Repeat the ray-tracing procedure shown in example 1.

Analyze → **RayTrace** → **Cone** → **Full Angle**.

But this time enter 1 for nx, 2 for ny, 1 for hfov (xfov), and 180 for vfov (yfov), and click OK.

There is still too much going on.

Set the ray-tracing limits

What we need to do is to limit the number of ray-surface intercepts, so that the rays bouncing around in the paraboloid do not totally confuse us.

Select **Options** → **Ray Trace Limits**.

Enter 5 for maxdepth (maximum number of ray-surface intercepts).

Enter 0.01 for mintrans (minimum ray energy).

Enter 2 for maxghost (maximum order of ghosting), and then click OK.

SetRayTraceLimits		
maxdepth	5	Ok
mintrans	0.01	
maxghost	2	

Repeat the ray–trace

Click on the **Update All** toolbar button.

The results are a cleaner example of rays bouncing off of surface `lens1.1`.

Modify angle dependent surface reflectance

Let us now cause surface `lens1.2` to reflect light according to the Fresnel equation (angle dependent).

Expand the name `lens1` (if needed) in the **OOb**.

Click on `surface2` in the **OOb** to open the **OSPb**.

Click on the **Fresnel page of the OSPb**.

Make sure that the name `lens1.2` is displayed on the **Fresnel page**.

Click **Fresnel**, then click **Apply Now**.

Slide the **OSPb** out of the way.

Repeat the ray–trace

Click on the **Update All** toolbar button.

You may wish to zoom in on the central region of the lens to see the effects of reflections occurring from both surfaces.

Review

The reflectance of a surface can be modified as either a fixed or angle dependent reflection.

The ray–trace can (and often should) be limited by using the **RayTraceLimits** option.

If you want to get to the same point as if you had just completed Sections A and B, then run the macro `B_Basic_Tutorial.mac`.

Section C

Section C will explain more about object and surface extents.

Exercise 10 Extent of a Mirror - Aperture - Film Plane

The extent of a single valued infinitely thin surface is determined by a method that is similar to a *cookie cutter*, by which an elliptical tube (oval definition) or a rectangular channel (actually four planes: two of which are parallel to the x-z plane and two of which are parallel to the y-z plane for the rectangular definition) limit the extent of the object.

The surfaces defining the extent of a nonsolid object are virtual surfaces in that they cannot intercept rays and are not displayed.

The example shown is for a paraboloidal mirror, but is applicable to flat mirrors, spherical mirrors, film planes, and apertures.

Clear existing objects and reset OptiCAD

Select File → New.

If prompted to save select **no**.

Define a parabolic mirror of oval extent

Select **Design** → **Mirror** → **Parabolic** → **Oval**.

Enter `pova1` for the object name and click OK.

Select **Both** and click OK.

Click the **View Ortho** toolbar button

Click on the **Definition page of the Object and Surface Properties box**.

Enter -300 for flx, -300 for yfl, 75 for xhw (hhwidth), 50 for yhw (vhwidth), and click **Apply Now**.

Place the oval paraboloid mirror

Click on the tab for the **Placement page of the OSPb**.

Enter 0 for x, 60 for y, 0 for z, then click **Place and Activate**.

Define a parabolic mirror of rectangular extent

Select **Design** → **Mirror** → **Parabolic** → **Rectangle**.

Enter `prect` for the object name and click OK.

Select **Both** and click OK.

Click on the **Definition page of the Object and Surface Properties box**.

Enter -300 for flx, -300 for yfl, 75 for xhw (hhwidth), 50 for yhw (vhwidth), and click **Apply Now**.

Place the rectangular paraboloid mirror

Click on the **Placement page of the OSPb**.

Enter 0 for x, -60 for y, 0 for z, then click **Place and Activate**.

Slide the **OSPb** out of the way.

Click the **SnugWorkbox** toolbar button.

Define a lightsource

Select **Design** → **LightSource**.

Then enter `light1` for the object name, then click OK.

Place the lightsource

Click on the **Placement page of the OSPb**.

Enter 0 for x, 0 for y, -225 for z, then click **Place and Activate**.

Note that when collimated rays are traced from the lightsource, which will be located between a film plane and the mirrors, the rays originate from the plane of the lightsource and not from infinity.

Define a film plane

Select **Design** → **Film** → **Rectangular**.

Then enter `film1` for the object name and click OK.

Click on the **Definition page of the Object and Surface Properties box.**

Enter 100 for xhw (hhwidth), 150 for yhw (vhwidth), then click **Apply Now.**

Place the film

Click on the **Placement page of the OSPb.**

Enter 0 for x, 0 for y, and -450 for z, then click **Place and Activate.**

Slide the **OSPb** out of the way.

Click the **SnugWorkbox** toolbar button.

Define another film plane

Select **Design** → **Film** → **Rectangular.**

Enter **film2** for the object name and click OK.

Click on the **Definition page of the Object and Surface Properties box.**

Enter 100 for xhw (hhwidth), 150 for yhw (vhwidth), then click **Apply Now.**

Place the second film

Click on the **Placement page.**

Enter 0 for x, 0 for y, and 100 for z, then click **Place and Activate.**

Slide the **OSPb** out of the way.

Click the **SnugWorkbox** toolbar button.

Click the **View Ortho** toolbar button

Trace a 2-D bundle of collimated rays

Let us now trace some rays to see the difference between these mirrors.

Select **Analyze** → **RayTrace** → **Collimated** → **Rectangle.**

Enter 20 for nx, 30 for ny, 80 for xhw (hhwidth), and 120 for yhw (vhwidth).

The object name should be **light1**, then click OK.

View the film plane

The perspective view shows the difference between the ray bundles, but the film planes give a better view of what is going on.

Select (by clicking on) the object **film2** from the **OOb.**

Click on the **Film View** toolbar button (bunch of spots).

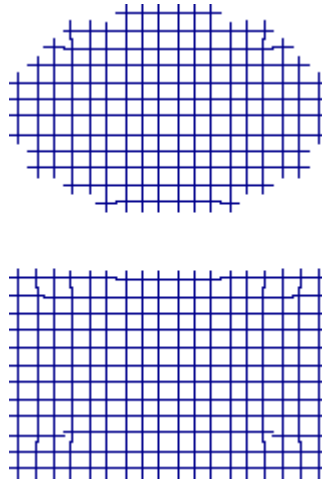
You should now see the pattern of rays that missed both mirrors and hit **film2.**

The difference between an oval and a rectangular mirror is obvious.

Next select the object **film1** from the **OOb.**

You should now see the pattern of rays that reflected off of both mirrors, and went through focus to hit **film1.**

The difference between an oval and a rectangular mirror is again obvious.



Defocussed Rays from Oval and Rectangular Mirrors

Review

A single valued thin surface (mirror, aperture, film plane) may be limited in extent by either a cylinder (oval) or by four planes (rectangular).

While this exercise shows extents that are symmetrical about the z-axis, extents that are offset from the z-axis may be specified using the **Extent page of the OSPb**.

Rays traced from a lightsource originate from the plane of the lightsource and not from infinity.

Exercise 11 Extent of Spherical and Cylindrical Lenses

Extents for lenses are similar to extents for mirrors except that the limiting surfaces are in fact explicit surfaces used to define an enclosed solid object.

These extent defining surfaces can intercept rays and are displayed.

Spherical Lens

A single-element spherical lens is a solid object whose extent is determined by a cylinder. The spherical lens consists of sections of two spheres whose centers are located on the axis of the extent defining cylinder. Wedge or tilt between the surfaces is not permitted.

Cylindrical Lens

A single-element cylindrical lens is a solid object whose extent is determined by four planar surfaces two of which are parallel to the x-z plane and two of which are parallel to the y-z plane.

The cylindrical lens consists of sections of two cylinders (either two cylinders with power in x, or two cylinders with power in y, but not a combination of an x-powered and a y-powered cylinder). The axes of the cylinders must be parallel to each other, and both parallel to either the x- or y-axes. Wedge, tilt, or rotation between the surfaces is not permitted.

In this exercise we are not going to trace any rays, we are just going to examine these two types of lenses.

Clear existing objects and reset OptiCAD

Select **File** → **New**.

If prompted to save select NO.

Define a spherical lens

Select **Design** → **Lens** → **Spherical**.

Enter `sphlens` for object name and click OK.

Click on the **Definition page of the Object and Surface Properties box**.

Enter 100 for r1, -100 for r2, 50 for thickness, 100 for diameter, and 1.5 for index, then click **Apply Now**.

Place the spherical lens

Click on the **Placement page of the OSPb**.

Enter 0 for x, 60 for y, and 0 for z, then click **Place and Activate**.

Define a cylindrical lens

Select **Design** → **Lens** → **Cylindrical** → **x_powered**.

Enter `cyl1lens` for the object name and click OK.

Click on the **Definition page of the Object and Surface Properties box**.

Enter 100 for r1, -100 for r2, 50 for xhw (hhwidth), 50 for yhw (vhwidth), 50 for thickness, and 1.5 for index, then click **Apply Now**.

Place the cylindrical lens

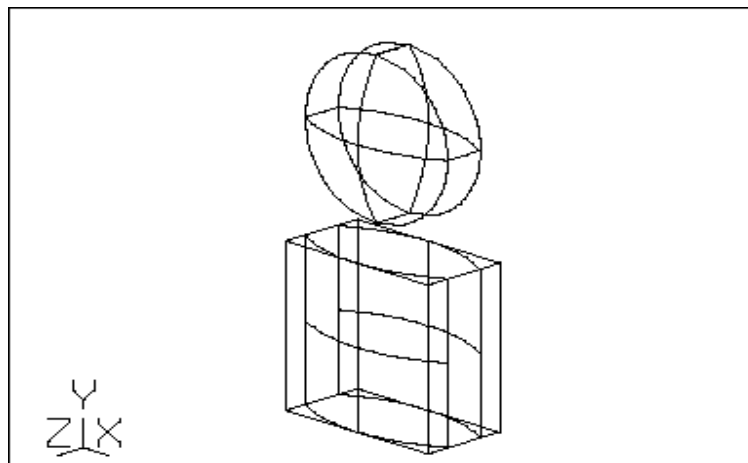
Click on the **Placement page of the OSPb**.

Enter 0 for x, -60 for y, and 0 for z, then click **Place and Activate**.

Slide the **OSPb** out of the way.

Click the **SnugWorkbox** toolbar button.

Click on the **View Ortho** toolbar button.



Spherical (top) and Cylindrical (bottom) Lenses

Review

A spherical lens is made up of three surfaces, sections of two spheres, and a section of a cylinder.

A cylindrical lens is made up of six surfaces, sections of two cylinders, and sections of four planes.

Exercise 12 Extent of a Cone - Ellipsoid - Cylinder - Cavity

Surfaces such as a cone, ellipsoid, or cylinder are defined by two (virtual) planes parallel to the x-y plane and located along the z-axis.

The surface(s) defining the extent of the above objects are virtual surfaces in that they cannot intercept rays and are not displayed.

The example shown is for an ellipsoid, but is applicable to cones and cylinders.

Clear existing objects and reset OptiCAD

Select **File** → **New**.

If prompted to save select **NO**.

Define a full ellipsoid

Ellipsoids, cones, and cylinders are called cavities in OptiCAD.

Select **Design** → **Cavity** → **Ellipsoid**.

Enter **ellip** for the object name and click **OK**.

Select **Both** and click **OK**.

Click on the **Definition page of the Object and Surface Properties box**.

Enter 50 for a, 50 for b, 100 for c, -100 for zmin, and 100 for zmax, then **Apply Now**.

Place the ellipsoid

Click on the **Placement page of the OSPb**.

Enter 0 for x, 0 for y, and 0 for z, then click **Place and Activate**.

Slide the **OSPb** out of the way.

Click the **SnugWorkbox** toolbar button.

Click on the **View Ortho** toolbar button.

A full ellipsoid should now be displayed on the screen.

The two limiting planes of the ellipsoid as defined above are located at $z=-100$ (zmin) and $z=100$ (zmax), the limiting planes lie outside of the ellipsoid, and therefore do not limit (cut off) any of the surface.

Modify the extent of the ellipsoid

We will now change the extent (but not the shape) of the ellipsoid.

Double click on the name ellip in the **OOB** which brings up the **OSPb**.

Select the **Definition page of the OSPb**.

Entering 0 for zmin, and then click **Apply Now**.

Slide the **OSPb** out of the way.

The limiting extent planes of the ellipsoid have now been redefined to be at $z=0$ and $z=100$.

The resulting surface is now only the right side ($z > 0$) of the ellipsoid.

Modify the extent of the ellipsoid (again)

We will now change the extent (but not the shape) of the ellipsoid.

Double click on the name **ellip** in the **OOB**.

Then select the **Definition page of the OSPb**.

Entering -100 for z_{min} , and 0 for z_{max} , and then click **Apply Now**.

Slide the **OSPb** out of the way.

The limiting extent planes of the ellipsoid have now been redefined to be at $z=-100$ and $z=0$.

The resulting surface is now only the left side ($z < 0$) of the ellipsoid.

Note that in defining the limiting planes z_{max} must always be greater than z_{min} .

Review

An object (thin surface) such as a cone, ellipsoid, or cylinder may be limited in extent by two virtual planes parallel to the x - y plane and located along the z -axis.

Section D

Section D will show the various options for placing and orienting objects.

The **OSPb** is an excellent way to make global placements in air, relative placement requires using the menu feature **Edit** → **Place**. Simple rotations are more easily accomplished using the menu features **Edit** → **Rotate**.

Exercise 13 Global and Relative Placement

This exercise shows the difference between global and relative object placement.

Clear existing objects and reset OptiCAD

Select **File** → **New**.

If prompted to save select **NO**.

Define a lightsource

Define a lightsource named **light1**.

Use **Design** → **Lightsource** (described in previous exercises).

Place the lightsource

Using the **Placement page of the OSPb** (described in previous exercises), place the lightsource (**light1**) at $x=0$, $y=0$, $z=0$.

Rotate the lightsource

Rotate the light (**light1**) locally about the x -axis by 45 degrees.

Use **Edit** → **Rotate** (described in previous exercises), be sure to select Local and about x.

Note that because the lightsource is located at the origin of the global coordinate system, there is no difference between a global rotation and a local rotation.

Define a spherical lens

Define a lens spherical named **lens1**, with r1 of 0, r2 of -54, a thickness of 5, a diameter of 25, and an index of refraction of 1.5.

Use **Design** → **Lens** → **Spherical** (described in previous exercises), and the **Definition page of the OSPb**.

Select a real optical glass

So far we have defined lenses for which we have specified a single index of refraction.

It now time to assign a real material to the lens. A real glass (or other material) has an index of refraction that varies with wavelength. Our choice of a glass is BK7, from the Schott catalog. The Schott catalog was loaded by the file StartUp.opt, when you started OptiCAD. OptiCAD will recognize any agf files that are explicitly loaded.

Select the **Definition page of the OSPb**.

Then open the **Optical Material list**.

Select BK7, and click **Apply Now**.

Place the lens in global coordinates

Place **lens1** at x=0, y=0, z=100.

Use the **Placement page of the OSPb** (described in previous exercises).

Slide the **OSPb** out of the way.

Click the **SnugWorkbox** toolbar button.

Click on the **View Ortho** toolbar button.

Trace a bundle of rays

Use **Analyze** → **RayTrace** → **Cone** → **FullAngle**.

Entering 1 for nx, 10 for ny, 1 for hfov (xfov), 10 for vfov (yfov).

Make sure that **light1** is entered for the object name, then click OK.

The ray bundle is centered around the +z-axis of the rotated lightsource.

The rays miss the lens.

A global placement was not what we wanted.

Define another spherical lens

Define a spherical lens named **lens2**, with r1 of 0, r2 of -54, a thickness of 5, a diameter of 25, and an index of refraction of 1.5.

Use **Design** → **Lens** → **Spherical** (described in previous exercises), and the **Definition page of the OSPb**.

Select a real optical glass (again)

Open the **Optical Material List**.

Select BK7, and click **Apply Now**.

Place the lens in relative coordinates

This time we will place the lens in the local coordinate system of the (rotated) lightsource.

Select **Edit** → **Place**, (make sure to) click relative.

Enter 0 for x, 0 for y, and 100 for z.

Enter **lens2** for the object name.

Enter **light1** for the reference object name, and click OK.

Repeat the ray-trace

Now click on the **Update All** toolbar button, and then on the **Snug Workbox** toolbar button.

The rays should now hit the lens and emerge collimated.

Rotate the lightsource (again)

Select **light1** from the **OOB** (by clicking on it's name).

Rotate **light1** locally about the x-axis by (an additional) 45 degrees.

Use **Edit** → **Rotate** (described in previous exercises), be sure to select Local and About x.

Repeat the ray-trace

Now click on the **Update All** toolbar button.

The rays should now miss both lenses.

More importantly **lens2**, which had been placed relative to the lightsource, has not moved.

Review

An object may be placed in the coordinate system of another object.

An important thing to note is that placing an object relative to (or inside of) another object, does not tie (or link) the two objects together.

Exercise 14 Inside Placement

Placing an object inside another object is a special type of relative placement. The ray-trace generally assumes that when a ray leaves an object, the ray exits into air. The **Place Inside** option tells the ray-trace function that the ray exits into the index of refraction of the host (reference) object rather than into air.

In this exercise we will place an object inside another object and examine the consequences.

Clear existing objects and reset OptiCAD

Select **File** → **New**.

If prompted to save select NO.

Define a lightsource

Define a lightsource named `light1`.

Use **Design** → **Lightsource** (described in previous exercises), and the **Definition page of the OSPb**.

Place the lightsource

Place the lightsource (`light1`) at $x=0$, $y=0$, $z=-50$.

Use the **Placement page of the OSPb** (described in previous exercises).

Define a spherical lens

Define a lens named `lens1`, with $r1$ of 0, $r2$ of 0, a thickness of 10, a diameter of 50, and an index of refraction of 1.5.

Use **Design** → **Lens** → **Spherical** (described in previous exercises), and the **Definition page of the OSPb**.

Select a real optical glass

Select BK7 as the glass for `lens1`.

Use the **Optical Material list in the Definition page of the OSPb** (described in previous exercises). Remember to click **Apply Now**.

Place the lens in global coordinates

Place the lens (`lens1`) at $x=0$, $y=0$, $z=50$.

Use the **Placement page of the OSPb** (described in previous exercises).

Click the **SnugWorkbox** toolbar button.

Click on the **View Ortho** toolbar button.

Trace a bundle of collimated rays

Use **Analyze** → **RayTrace** → **Collimated** → **Rectangle**.

Enter 1 for n_x , 10 for n_y , 1 for x_{hw} (hhwidth), 15 for y_{hw} (vhwidth), then click OK.

The rays pass through the lens undeviated because surface 1 and surface 2 are both flat.

Define a spherical lens made of air

Define a lens named `lens2`, with $r1$ of -50, $r2$ of 50, a thickness of 5, a diameter of 20, and an index of refraction of 1.0.

Use **Design** → **Lens** → **Spherical** (described in previous exercises), and the **Definition page of the OSPb**.

This lens is indeed a negative lens made of air.

Place the air lens inside the BK7 lens

When `lens2` is placed inside `lens1`, the coordinates of the (inside) placement, will be specified in the local coordinate system of (the host object) `lens1`.

Select **Edit** → **Place**.

Check Inside.

Enter 0 for x, 0 for y, and 2.5 for z.

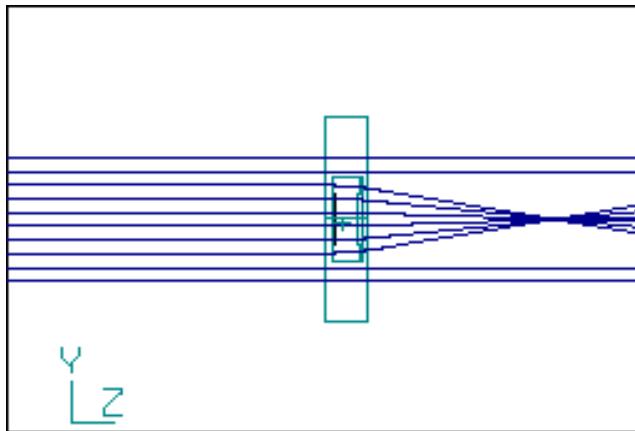
Enter `lens2` for the object name.

Enter `lens1` for the reference object name and click OK.

Repeat the ray-trace

Now click on the **Update All** toolbar button.

The rays that pass through both lenses should now converge



Negative "Air" Lens Inside a Glass Plate

Review

An object may be placed inside another object.

An inside placement tells the ray-tracing routine that a ray exiting an object exits into the index of the host object rather than into air.

An inside placement is specified in the local coordinate system of the host object.

An important thing to note is that placing an object inside of (or relative to) another object does not tie (or link) the two objects together.

Section E

In section E we will assign special properties to a surface. These include Fresnel lenses, diffraction gratings, and scattering surfaces.

Exercise 15 Fresnel Lens

The **Object and Surface Properties** chapter includes a discussion of the approximations made in modeling a Fresnel lens.

Clear existing objects and reset OptiCAD

Select **File** → **New**.

If prompted to save select **NO**.

Define a lightsource

Define a lightsource named `light1`.

Use **Design** → **Lightsource** (described in previous exercises), and the **Definition page of the OSPb**.

Place the lightsource

Place the lightsource (**light1**) at $x=0$, $y=0$, $z=-100$.

Use the **Placement page of the OSPb** (described in previous exercises).

Define a spherical lens

Define a lens named **f1ens**, with r_1 of 0, r_2 of 0, a thickness of 0.01, a diameter of 100, and an index of refraction of 1.5.

Use **Design** → **Lens** → **Spherical** (described in previous exercises), and the **Definition page of the OSPb**.

Place the lens in global coordinates

Place the lens (**f1ens**) at $x=0$, $y=0$, $z=0$.

Use the **Placement page of the OSPb** (described in previous exercises).

Click the **SnugWorkbox** toolbar button.

Click on the **View Ortho** toolbar button.

Make the lens a Fresnel lens

Double click on the name **f1ens** in the **OptiCAD Objects box (OOB)**, the name **f1ens** will expand. Click on Surface 2 of **f1ens**.

Select the **AltSurfs page of the OSPb**, then select **Fresnel** and **Radial** for Surface 2 of **f1ens** (displayed as **f1ens.2**).

Enter 100 for s , 50 for sp , and click **Apply Now**.

The parameter sp and s are the object and image distances for perfect imaging.

Trace a bundle of rays

Click on **light1** in the **OOB**.

Use **Analyze** → **RayTrace** → **Cone** → **FullAngle**.

Enter 1 for n_x , 10 for n_y , 1 for $hfov$ ($xfov$), 20 for $vfov$ ($yfov$).

The ray-trace should show near perfect imaging from a point 100 units to the left of the Fresnel lens to a point 50 units to the right of the Fresnel lens.

Exercise 16 Diffraction Grating

This exercise defines a diffraction grating.

Clear existing objects and reset OptiCAD

Select **File** → **New**.

If prompted to save select **NO**.

Define a lightsource

Define a lightsource named **light1**.

Use **Design** → **Lightsource** (described in previous exercises), and the **Definition** page of the **OSPb**.

Place the lightsource

Place the lightsource (**light1**) at $x=0$, $y=0$, $z=-50$.

Use the **Placement** page of the **Object and Surface Properties** box (described in previous exercises).

Define a flat mirror

Select **Design** → **Mirror** → **Flat** → **Rectangle**.

Enter **mirr1** for object name and click **OK**.

Select **Both** and click **OK**.

Click on the **Definition** page of the **Object and Surface Properties** box.

Enter 10 for **xhw** (hhwidth), 10 for **yhw** (vhwidth), then click **Apply Now**.

Place the mirror

Place the mirror (**mirr1**) at $x=0$, $y=0$, $z=0$.

Use the **Placement** page of the **OSPb**.

Make sure to click **Place** and **Activate**.

Click on the **View Ortho** toolbar button.

Modify the mirror to be transparent

Select the **Attributes** Page of the **OSPb**.

Enter 0 for **refl**, 0 for **absor**, and then clicking **Apply Now**.

Trace a single ray

Trace a single ray from **light1**. Click on **light1** in the **OOb**.

Use **Analyze** → **RayTrace** → **Cone** → **FullAngle** (described in a previous exercise).

Enter 1 for all four numerical parameters.

A single ray passes through the surface undeviated.

Modify the (transparent) mirror to be a grating

Double click on **mirr1** in the **OptiCAD Objects** box.

Select the **AltSurfs** page of the **OSPb**.

Select **Grating** as the **Alt-Surface** type.

Enter 500 for frequency, 0 for theta, 1 for order.

Check that **mirr1.1** is the surface name, and click **Apply Now**.

Slide the **OSPb** out of the way.

Repeat the ray–trace

Now click on the **Update All** toolbar button.

The same ray is now deviated upward according to the grating equation.

Modify the (transparent) mirror to be a grating (again)

Select `mirr1` from the **OptiCAD Objects box**.

Select the **AltSurfs page of the OSPb**.

Enter 1000 for frequency, 0 for theta, -1 for order, click **Apply Now**.

Repeat the ray-trace

Now click on the **Update All** toolbar button.

The same ray is now deviated downward (but at a greater angle) according to the grating equation.

Review

A mirror can be modified into a free standing transparent surface.

A surface can be modified into a diffraction grating.

Exercise 17 Surface Scattering

Surface scattering is a weighted random process in which a ray is perturbed about its specular (unscattered) direction. Therefore the results will be different each time you retrace a ray bundle incident on a scattering surface. The surface scattering properties follow the specified scattering function only in the average (many rays) sense.

Clear existing objects and reset OptiCAD

Select **File** → **New**.

If prompted to save select NO.

Define a lightsource

Define a lightsource named `light1`.

Use **Design** → **Lightsource** (described in previous exercises), and the **Definition page of the OSPb**.

Place the lightsource

Place the lightsource (`light1`) at $x=0$, $y=0$, $z=-50$.

Use the **Placement page of the Object and Surface Properties box** (described in previous exercises).

Define a flat mirror

Select **Design** → **Mirror** → **Flat** → **Rectangle**.

Enter `mirr1` for object name, then click OK.

Select **both** and click OK.

Select the **Definition page of the OSPb**.

Enter 10 for `xhw` (hhwidth), 10 for `yhw` (vhwidth).

Place the mirror

Place the mirror (`mirr1`) at $x=0$, $y=0$, $z=0$.

Use the **Placement** page of the OSPb.

Make sure to click **Place** and **Activate**.

Tilt the mirror

Select **Edit** → **Rotate**.

Select Local, select x.

Enter 20 for angle, `mirr1` for object, and click OK.

Trace a small collimated ray bundle

Select **Analyze** → **RayTrace** → **Collimated** → **Rectangle**

Enter 3 for nx, 3 for ny, 0.0001 for hfov (xfov), 0.0001 for vfov (yfov).

Enter `light1` for the object name, and click OK.

A very small bundle of collimated rays (specularly) reflects off of the mirror.

Modify the mirror to be scattering

Click on `mirr1` on the OOb.

Use the **Scattering** page of the OSPb.

Select **Gaussian** as the **Scattering Function Type**.

Enter 1 for fraction (scattered), 0.1 for sigma.

Click **Apply Now**.

Repeat the ray–trace

Now click on the **Update All** toolbar button.

The same very small bundle of collimated rays is now deviated into a scattered bundle.

Repeat the ray–trace several times

Now click the **Update All** toolbar button several times.

Note that the rays are scattered from the mirror with a different distribution with every retrace.

Review

A surface can be modified to be scattering.

The scattering is centered around the specular ray direction.

Scattering is a weighted random process that obeys the input scattering parameters only in the average (many rays).

Section F

Section F describes the definition of polygonally faceted objects (such as a prisms).

Polygonally faceted objects are defined from macro files, using a rather complicated command structure. You may wish to skip this section at this time, and return to it after becoming more familiar with the workings of OptiCAD.

There are two types of polygonally faceted objects: extruded and generalized. In an extruded object, the user only needs to specify a surface (usually by defining a series of points in the x-y plane), and then specifying an extrusion depth into the z-direction. OptiCAD automatically creates all of the other surfaces (the original surface is reproduced as the top and bottom surface and all of the necessary sides are created). The general polygonally faceted object, for which the user must specify all of the facets in three dimensions, will be treated at a later time.

A polygonally faceted object is either a surface or closed object (such as a cube or prism) made up of flat facets. The object is specified by specifying the corners of every facet. In the case of a closed object, the defining surface must be completely defined, gaps between facets are not permitted.

Exercise 18 Extruded Solid Polynet

In this exercise we will create an extruded polygonally faceted object (a simple wedge), by running a macro.

Clear existing objects and reset OptiCAD:

Select **File** → **New**.

If prompted to save select NO.

Examine a macro:

Select **File** → **Edit File**.

Then find and double click on the macro named Extrude_Tutorial.mac.

The macro Extrude_Tutorial.mac is also shown below.

```
clear
# This macro is template for defining
# an extruded polynet object
# You may wish to delete the clear command from the start of this file
#
# Name object and define global index,
# surface reflectance and surface absorptance
#
# polynet/start my_wedge index="index" refl="reflectance" abs="absorptance"
polynet/start my_wedge index=1.419 refl=0.0 abs=0.0
polynet/polygon
polynet/chain
# Defines points in the x-y plane (z=0)
#
# polynet/point x1 y1 0
polynet/point 10 1# Point 1
# polynet/point x2 y2 0
polynet/point 10 -1# Point 2
# polynet/point x3 y3 0
polynet/point -10 -3# Point 3
# polynet/point x4 y4 0
polynet/point -10 3# Point 4
#
# Do Not repeat the first point of a chain as the last point
polynet/close
# Extrudes the plane into +/- zthick/2
```

```
# polynet/thick zthick
polynet/thick 25
polynet/finish
#
# Normalize wedge only after completely defining it
# i.e. after "polynet/finish"
#
polynet/normalize my_wedge
```

The macro `Extrude_Tutorial.mac` may be used as a template for the design of an extruded polynet object.

Now close the Notepad editor.

Run the macro

Select **File** → **RunMacro**.

Find and double click on the macro named `Extrude_Tutorial.mac`, which is located in the `samples` subdirectory.

Place the wedge

Click on the **View Ortho** toolbar button.

Double click on the object `my_wedge` in the **OptiCAD Objects box**.

Using the **Placement page of the Object and Surface Properties box**, place the wedge at $x = 0, y = 0, z = 0$, click **Place and Activate**.

Define a lightsource

Select **Design** → **Lightsource**.

Enter `light1` for the object name and click OK.

Place the lightsource

Using the **Placement page of the Object and Surface Properties box**, place `light1` at $x = -15, y = 0, z = 0$, click **Place and Activate**.

Rotate the lightsource

Select **Edit** → **Rotate**.

Select Local and About y.

Enter -90 for the rotation angle.

Enter `light1` for the object name, and click OK.

Click on the **Snug Workbox** toolbar button

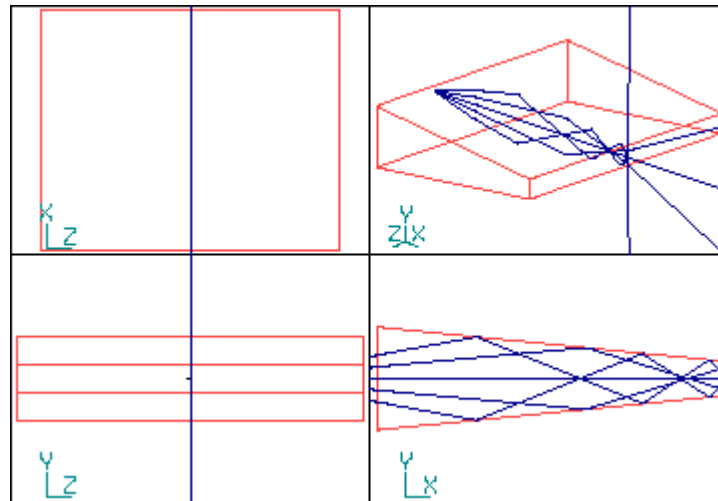
Trace rays

Select **Analyze** → **Ray Trace** → **Cone** → **Full Angle**.

Enter 1 for `nx`, 5 for `ny`, 1 for `hfov` (`xfov`), and 30 for `vfov` (`yfov`).

You may wish to select the z View toolbar button. And then zoom in on the wedge using the Zoom In toolbar button and defining a zoom region by dragging the mouse. The results are

shown in the picture shown below.



Results of Ray-Tracing the Macro Extrude_Tutorial.mac

Review

A solid polynet object may be made by defining the corner points of a flat surface located in the x-y plane and extruding the surface into the z-axis. This is the simplest solid polygon, more generalized (non-extrudable) polygon objects require the specification of all of the object's corner points.

Exercise 19 Sheet with Four Holes

In this exercise we will create a square flat sheet by running the macro Hole_Plate_Tutorial.mac. The macro: defines the sheet, places four holes in the sheet, defines and places a lightsource, defines and places a film plane, and then traces a collimated bundle of rays. In this exercise all you need to do is run the macro. In the following exercise you will modify and then rerun the macro.

Clear existing objects and reset OptiCAD:

Select **File** → **New**.

If prompted to save select NO.

Run the macro

Select **File** → **RunMacro**.

Find and double click on the macro named Hole_Plate_Tutorial.mac, which is located in the **samples** subdirectory.

Press Enter when prompted.

The macro is shown below.

```
clear
echo/on
set/saveray/on
set/savespot/on

set/fletch/off
set/add_rays/off
set/workbox 0 0 0 100 100 100
```

```

set/raylim 100 0.01 1
set/fresnel/off
set/autoretrace/off
set/autoredraw/on
#set/color/val 7 12
set/splensvertex/on
set/matcatalog/name=schott.agf
set/proximity 0
set/energy/absolute
set/energy/density
set/units/mm
set/wave 0.55 0.55 0.55 0.55 0.55 0.55 0.55 0.55 0.55 0.55 0.55 0.55
set/weight 1 0 0 0 0 0 0 0 0 0 0 0
# This macro is template for defining
# a square sheet with four square holes in it
#
# Name the object and define its global index,
# surface reflectance and surface absorptance
# polynet/start "name" index="index" refl="reflectance" abs="absorptance"
polynet/start holeplate index=1.0 refl=1.0 abs=0.0
poly/poly
poly/chain# chain of points defines outside of sheet
# Defines points in the x-y plane (z=0)
# polynet/point x1 y1 0
poly/point 2 2 0# Point 1
# polynet/point x2 y2 0
poly/point 2 -2 0# Point 2
# polynet/point x3 y3 0
poly/point -2 -2 0# Point 3
# polynet/point x4 y4 0
poly/point -2 2 0# Point 4
# Do Not repeat the first point of a chain as the last point
poly/close
#
poly/chain# chain of points defines hole 1
poly/point 0.9 0.9 0
poly/point 0.9 0.1 0
poly/point 0.1 0.1 0
poly/point 0.1 0.9 0
poly/close
#
poly/chain# chain of points defines hole 2
poly/point 0.9 -0.9 0
poly/point 0.1 -0.9 0
poly/point 0.1 -0.1 0
poly/point 0.9 -0.1 0
poly/close
#
poly/chain# chain of points defines hole 3
poly/point -0.1 -0.1 0
poly/point -0.1 -0.9 0
poly/point -0.9 -0.9 0
poly/point -0.9 -0.1 0
poly/close
#
poly/chain# chain of points defines hole 4

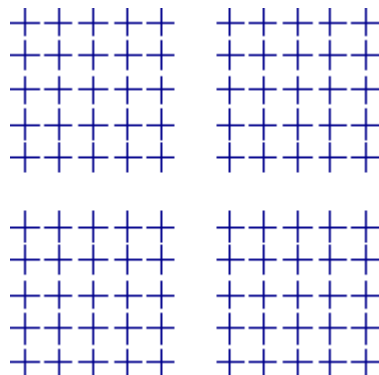
```

```

poly/point -0.1 0.9 0
poly/point -0.1 0.1 0
poly/point -0.9 0.1 0
poly/point -0.9 0.9 0
poly/close
#
#
###poly/thick 10
#
#
poly/finish
#
place holeplate 0 0 0
#
def/light light1
place light1 0 0 -10
#
define/film/rect filma 3 3
place filma 0 0 10
#
set/work/snug
#
ray/col/rect light1 25 25 1.99 1.99
view/system/ray
pause
view/film filma
pause
#
#
###ray light1 8 8 15 15
#
#
###view/ortho/ray
#
#

```

The ray pattern on the film plane (below) shows the effect of the four holes.



Ray Pattern After the Four Holes in Sheet

Exercise 20 Extrude Four Holes into an Egg Crate (four lightpipes)

You must first do exercise 19 and run the macro Hole_Plate_Tutorial.mac.

In this exercise we will modify the macro Hole_Plate_Tutorial.mac, such that the sheet

and four holes are extruded into an egg-crate-like structure (four parallel lightpipes).

Edit the macro

Select **File** → **EditLastMacro**.

The Notepad editor will open showing the macro Hole_Plate_Tutorial.mac.

Now examine the three lines in the macro that begin with three pound signs (###). The lines are:

```
###poly/thick 10
###ray light1 8 8 15 15
###view/ortho/ray
```

Edit all three of these lines.

Delete the three pound signs at the beginning of each line.

Make the three lines look like the following three lines.

```
poly/thick 10
ray light1 8 8 15 15
view/ortho/ray
```

Make sure to leave the text intact and the lines in their original locations.

Save the modified macro (in Notepad)

Select **File** → **Save As** (from the Notepad menu).

Enter H_P_Mod.mac for the name of the new file.

Select **File** → **Exit** (from the Notepad menu).

Notepad will close and you should again be in OptiCAD.

Rerun the (modified) macro

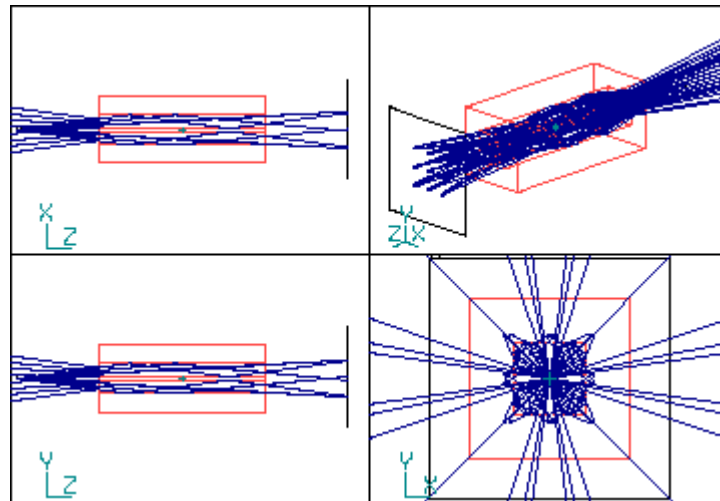
Select **File** → **RunMacro**

Enter h_p_mod.mac for the name of the new file.

Select h_p_mod.mac for the name of the file to be run.

Press Enter when prompted.

The ray-trace of the egg crate should look like the figure below.



Rays Traversing the Egg Crate Lightpipes

Review

A surface containing holes may be extruded to form an array of lightpipes.

A macro may be edited using the Notepad editor.

A revised macro may be rerun from within OptiCAD.

End of Basic Tutorial Chapter

TUTORIAL II

Viewing

Zoom in on an area of interest

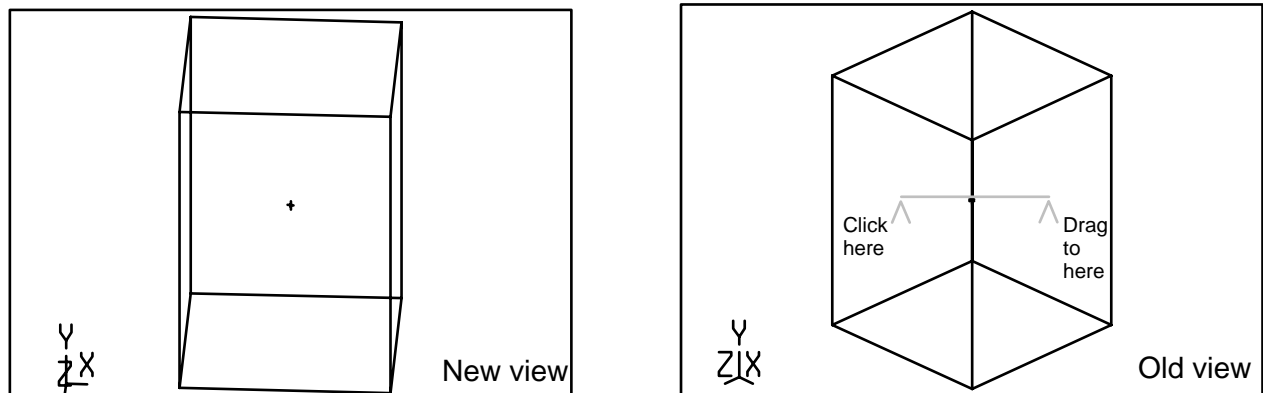
Run any OptiCAD macro and display a y view by clicking on the y toolbar button. The example figures below are from the macro cone.wdm. In order to zoom in on a subregion of interest select the ZoomIn mode by clicking on the button containing a dotted square with a solid square located in the upper right hand corner. Place the mouse cursor at the upper left of a subregion of interest and hold down the mouse button while moving the mouse cursor to the lower right of the subregion of interest. Then release the mouse button showing an enlarged view of the subregion enclosed by the stretched square. The zooming process may be repeated for x and z views.

UnZoom

The UnZoom feature redefines the 3-D viewing volume (workbox) to enclose all of the defined objects. To UnZoom press the button that looks similar to a belt being tightened (black hour glass with white belt). It is called the SnugWorkbox option and it automatically adjusts the workbox size to just enclose all objects in the system. This button is equivalent to selecting Window and then selecting SnugWorkbox from the pull down menus.

Rotate viewing angle

Press the Viewpoint Select Mode toolbar button. Place the mouse cursor at the left side of the 3-D box as indicated by the arrow marked "Click here" in the figure shown on the next page. Press the mouse button, drag it to the right, and then release the mouse button. The box will rotate with the new view angle azimuth and elevation displayed in the lower right of the window. Notice the change in orientation of the reference axis in the view below.



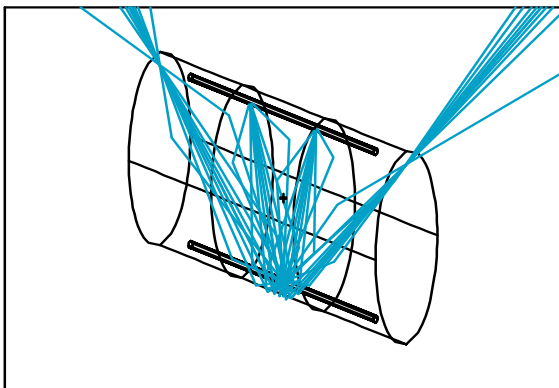
Unconstrained Ray-Tracing

Unconstrained ray-tracing is an important feature of OptiCAD. This feature was fully integrated into the program at the time of initial design. The implementation is therefore more complete than if the feature had been added to an existing sequential ray-tracing program.

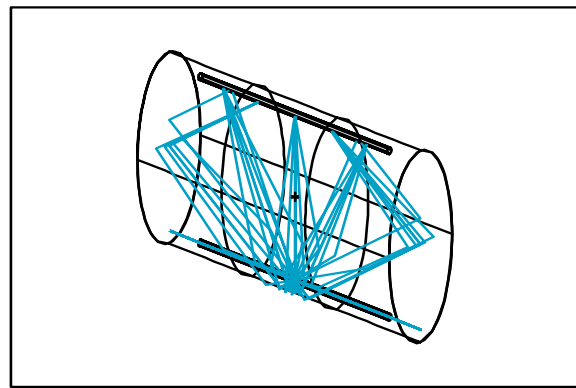
An introduction to unconstrained ray-tracing is shown in the next macro; let's run the macro Cylinder.mac (located in the Samples subdirectory). Click File, → RunMacro → Cylinder.mac → OK.

A reflective elliptical cylinder contains a lightsource at one focus and a laser rod at the other focus. The effect of adding flat end mirrors to the cylinder is that there are three possible ray paths between the lightsource and the rod. These paths are a) flash lamp to rod, b) flashlamp to elliptical cylinder to rod, and c) flash lamp to flat end mirror to elliptical cylinder to rod.

Let's deactivate the rods by clicking on the arrow in the OptiCAD Objects box (located at the left of the screen). Next find the object named `laser_rod` (the keyboard up and down arrows may be used to scroll through the box), and click on `laser_rod` to select it. Now select Edit → Deactivate. Repeat the sequence to deactivate the object named `flash_lamp`. Now select Options → RayTraceLimits and enter 1 for maxdepth. Now click on the Update toolbar button (red and blue fireworks) to repeat the last ray-trace, and click on the Ortho View toolbar button (subdivided rectangle) to display multiple views. The rays stop at the cylinder. Now try different values of maxdepth (2, 3, 4) and click the Update button for each value.



Cylinder Without End-Mirrors



Cylinder With End-Mirrors

This example shows unconstrained ray-tracing, that is, rays hitting surfaces in different orders, or striking the same surface more than once, and also shows that you can easily be overwhelmed with rays. You need to set criteria for truncating a ray by setting the raylimit parameters: the maximum number of ray-surface intersections (maxdepth), the minimum ray energy (mintrans), and the maximum order of ghost (maxghost).

Scattering from Diffuse Surfaces

Modeling of scattering from diffuse surfaces is important for applications where the scattering is desired, as with photocopiers, laptop computers, integrating spheres, and flat panel displays; and for cases where the scattering is undesired and must be minimized, as is the case with stray light analysis and baffle design. Scattering from diffuse surfaces requires unconstrained ray-tracing.

OptiCAD represents surface scattering as a gaussian, lambertian, power-law, or 2-fold exponential function. These functions may be applied to any surface; the user may also specify the fraction scattered. The OptiCAD scattering method treats the Bi-directional Reflectance Distribution Function (BRDF) or Bi-directional Transmittance Distribution Function (BTDF) as a probability density function. This statistical method eliminates the problem of ray build up that is associated with ray-cascading methods. Ray energy is conserved and the program carries the relative energy with each ray.

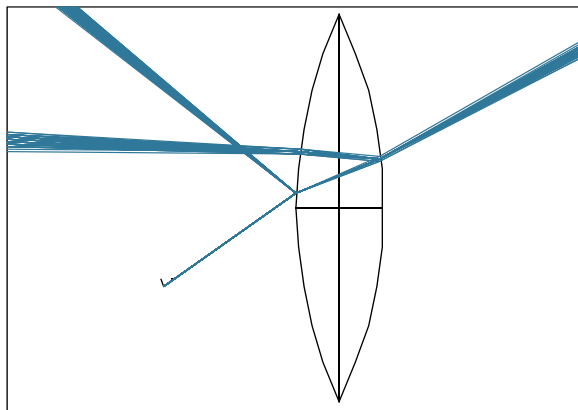
Single-Element Lens Scattering

An introduction to scattering from a diffuse surface is shown in the next macro; let's run the macro Dif_Lens.mac (located in the Samples subdirectory). Click File, → RunMacro → Dif_Lens.mac → OK.

The singlet lens example shows that you can assign different scattering angles to different surfaces; the left surface (hazer.1) has a (Gaussian) standard deviation 0.010, and the right surface (hazer.2) has a (Gaussian) standard deviation 0.005.

The ray-trace shows that rays are scattered into a distribution centered about the specular ray direction, and that multiple surface scattering is cumulative. The next ray-trace shows that when a ray splits, both the refractive and reflective components are scattered. The final ray-trace, which also demonstrates that total internal reflections are handled automatically, shows that we can specify the fraction of the light (10% in this case) that is scattered and the fraction that remains specular. A highly specular (small scattering) surface might be an automobile windshield with a slight haze, while a high-scattering (small specular) surface might be a diffuse integrating sphere with a slight oil film.

This is an example of unconstrained ray-tracing. Rays can not only intercept surfaces in any order, different rays will scatter differently and rays which split can also be followed along all paths.



Scattering in a Single-Element Lens

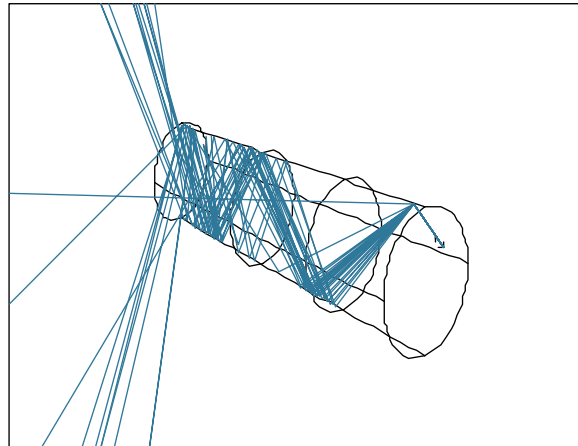
After running the macro Dif_Lens.mac, you can observe the probabilistic method used to model scattering. Click the x View button from the toolbar, and then click the Update All button from the toolbar (blue and red fireworks-like symbol), which will repeat the last ray-trace. The AutoRedraw Option should be activated. The ray pattern will change with every application of the Update All button; on the average, the rays will be distributed about the specular ray direction according to the scattering model specified.

Scattering and Multiple Bounces in a Reflective Conical Light Pipe

An example of two different types of scattering functions is shown in the next macro; let's run the macro Dif_Pipe.mac (located in the Samples subdirectory). Click File → Run-Macro → Dif_Pipe.mac → OK.

The example assigns a (rotationally symmetric) Gaussian scattering profile (of standard deviation 0.0150) to the cone surface (hazer.1). Again the angle of the scattered ray bundle is centered about the specular direction, and it increases with every bounce.

We next modify the surface scattering to have a (twofold exponential distribution) of standard deviations in x of 0.0003, and in y of 0.0150. The results show a large difference in the x and y scattering angles.



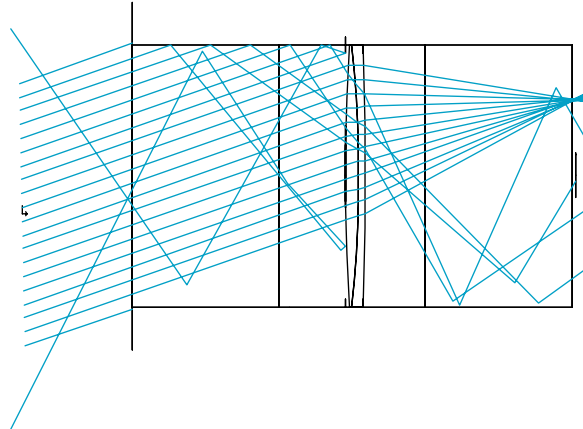
Scattering in a Conical Pipe

After running the (OptiCAD demonstration) macro Dif_Pipe.mac, you can observe the probabilistic method used to model scattering. Click the x view button from the toolbar, and then click the Update All button from the toolbar (blue and red fireworks-like symbol), which will repeat the last ray-trace. The AutoRedraw Option should be activated. The ray pattern will change with every application of the Update All button; on the average, the rays will be distributed about the specular ray direction according to the scattering model specified.

Stray Light Modeling

An example of the use of scattering from a diffuse surface to estimate stray light is shown in the next macro; let's run the macro Z_Object.mac (located in the Samples subdirectory). Click File → RunMacro → Z_Object.mac → OK.

The macro creates a doublet with a cylindrical lens shade (cylindrical tube) and traces some on-and off-axis rays. The shade reflects only 2% of the light, all of which is scattered into a Gaussian distribution. A bundle of collimated rays is then traced from an angle greater than the detector's field of view. The view shows scattered light bouncing around the tube. When a large number of rays are traced, the results show that approximately 1×10^{-6} of the incident light gets to the detector.



Stray Light Scatters off of a Baffle

After running this macro, you can change the scattering angle from the tube, the ray energy cut-off criteria, and the amount of surface scattering from the lens. You can then rerun the macro.

Radiometry

Radiometry with a Compound Reflector

An example of a compound reflector is shown in the next macro; let's run the macro Sun_Gun.mac (located in the Samples subdirectory). Click File → RunMacro → Sun_Gun.mac → OK.

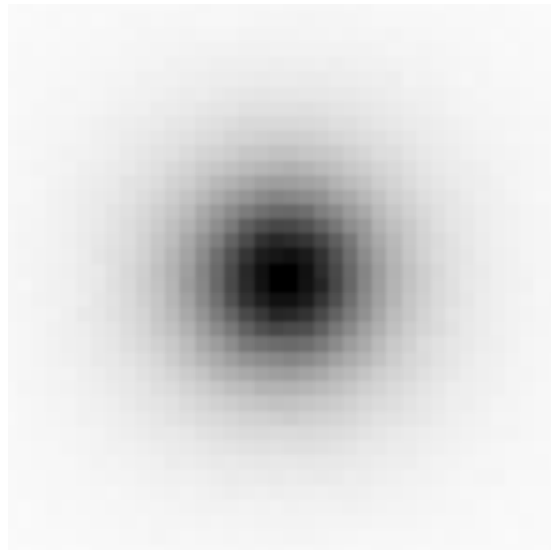
The example macro consists of a truncated parabolic mirror with flat mirrors closing the sides. A uniform bundle of rays is traced from a point lightsource located at the foci of the parabolic mirror.

Then a 3-D volumetric spherical source is placed inside the deep compound mirror. Rays originate uniformly from random locations within the spherical volume source, and are launched isotropically (no preferred direction).

These Monte Carlo ray origins and directions do not impose an artificial (numerical) structure on a final intensity pattern, as would be the case of a uniform grid of rays.

Rays are again traced into a sphere, and the film plane and enslitted energy are viewed. Energy is carried with each ray and used in radiometric calculations.

Finally, we view a 2-D irradiance map, generated by attaching a radiometer to the film plane (see the **Attach** subsection in the **Radiometer** section in the **Analyze** chapter). For reference, the result of tracing 2,000,000 rays is shown below.



After running the macro, add more rays to the irradiance map. Select Options, then click AddRays (making sure that it is now checked). Next click Update All (blue and red fire-works-like button), and now update the irradiance distribution by clicking on the View Intensity Map button (a white blob on a black background).

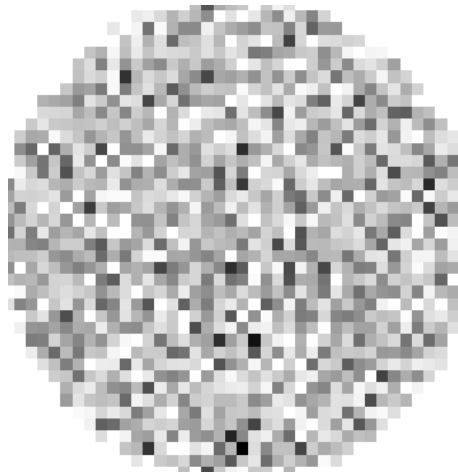
The numerical data from a radiometer can be exported to a file in a comma separated variables (csv) format, which can then be read into another application (see the **TextOut** subsection in the **Radiometer** section in the **Analyze** chapter).

Integrating Sphere with Nearly Lambertian Surface

An example of an integrating sphere is shown in the next macro; let's run the macro Int_Sph.mac (located in the Samples subdirectory). Click File → RunMacro → Int_Sph.mac → OK.

The example creates a sphere whose surface absorbs 5%, has a specular reflection of 4.75% (a slight shine), and scatters 90.25% into a broad (almost Lambertian) distribution. Rays are traced into a hemisphere and the results viewed.

We now place a microsphere having Lambertian surface scattering around the point source. The purpose of the microsphere is to randomize ray directions. These randomized rays have an advantage over a structured grid of rays in that they cannot impose any artificial (numerical) structure on a final intensity pattern. We trace more rays into a hemisphere, view the film plane, and then view a 2-D irradiance map. The result of tracing 200,000 rays is shown on the following page.



After running the macro, add more rays to the irradiance map. Click Options → AddRays (make sure it is checked), click Update All, and update the irradiance distribution by clicking ViewIntensityMap.

Polygonally Faceted Objects

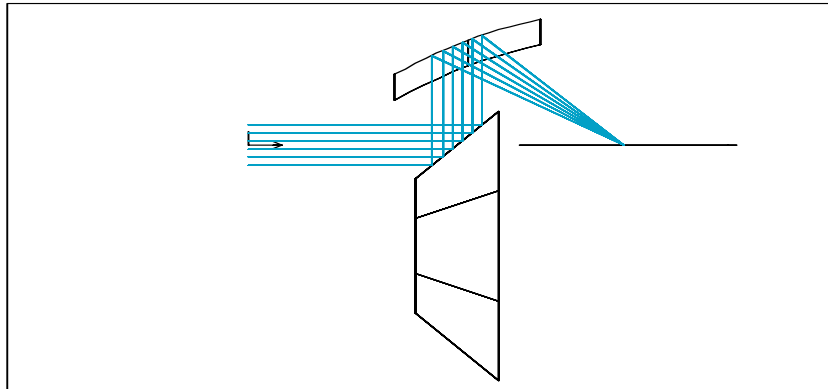
The ability to ray-trace polygonally faceted surfaces and objects allows the modeling of faceted objects, such as prisms, scanners, and wedges. A faceted representation may also be used to approximate complex surfaces and objects that could not be readily modeled as explicit objects, such as molded parts and entities imported from a CAD program. The faceted representation may be either an extruded object (such as a wedge), or a generalized 3-D object.

Polygon Scanner

An example of a polygonally faceted object is shown in the next macro; let's run the macro Polygon_Scanner.mac (located in the Samples subdirectory). Click File → Run-Macro → Polygon_Scanner.mac → OK.

The example creates an eight-sided polygonal scanner by specifying the corner points of each facet in three dimensions. Collimated light is reflected off of the scanner to an off-axis section of a parabolic mirror, then reflected to a film plane, and results are viewed. The polynet command is described in the **Polynet** chapter.

The scanner is then rotated about its axis of rotation several times, and rays are traced at each positions. The summation of rays from these positions is viewed on the detector plane, then as a perspective view.

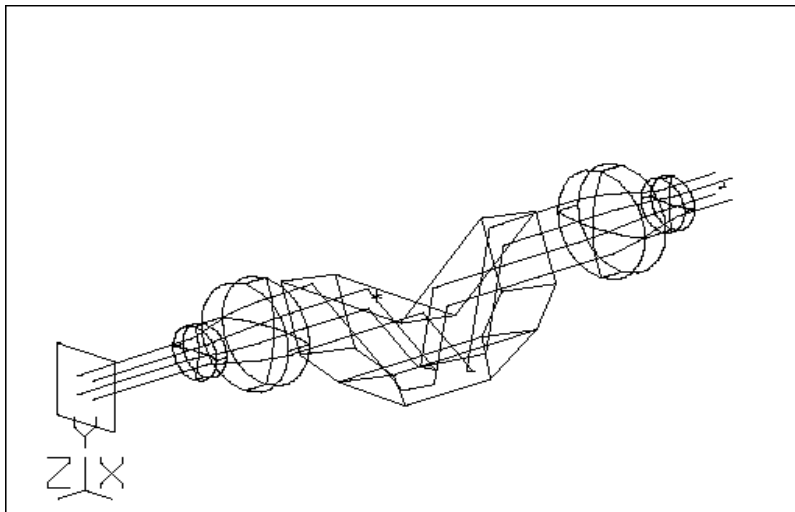


Polygonally Faceted Scanner

Abbe Prism

An example of an Abbe prism is shown in the next macro; let's run the macro Abbe_Prism.mac (located in the Samples subdirectory). Click File → RunMacro → Abbe_Prism.mac → OK.

The example creates an Abbe prism by specifying the corner points of each facet in three dimensions. The prism is fed by collimated light exiting a beam-expander (telescope); a second telescope compresses the output beam from the telescope. The polynet command is described in the **Polynet chapter**.



Abbe Prism Rotated

The results are viewed showing TIR. Finally, to demonstrate that the object is a complete solid prism, the prism is rotated several times and the rays retraced.

Wedge-shaped Lightpipe

An example of a wedge-shaped lightpipe is shown in the next macro; let's run the macro Wedge_Illuminator.mac (located in the Samples subdirectory). Click File → RunMacro → Wedge_Illuminator.mac → OK.

The example creates a solid wedge of glass as an extruded polynet surface by specifying the corner points of each facet in two dimensions and the extrusion thickness. A lightsource is located inside the wedge, and a receiver is placed slightly outside the wedge. The wedge surfaces are uncoated, and rays are reflect by either TIR or by Fresnel reflections. A uniformly spaced bundle of rays is traced and ray paths are viewed, then Monte Carlo rays are traced (but their paths are not saved) from a diode extended source (0.05 x 0.05 in extent into a 30° by 30° half angle gaussian bundle), and the enslitted energy is plotted.

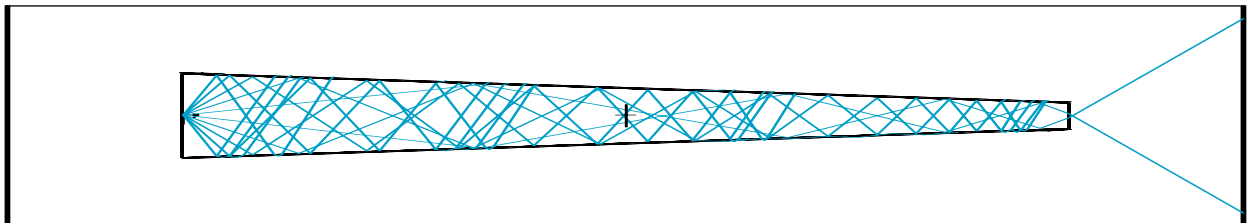
Next we change the face of the wedge opposite the receiver plane to be reflective. A uniformly spaced bundle of rays is again traced and ray paths are viewed, then Monte Carlo rays are traced from the diode source, the enslitted energy is plotted.

Finally, we attach a radiometer to the film plane, and perform a full 3-D analysis by tracing a much larger number of Monte Carlo rays from the diode source. The results are viewed as a 2-D irradiance map. The irradiance map is more informative if the user selects View → Palette → Spectrum, and then clicking on the View Intensity Map toolbar button.

For reference purposes the results of tracing 100,000 rays are shown below.



Irradiance Map Along Wedge

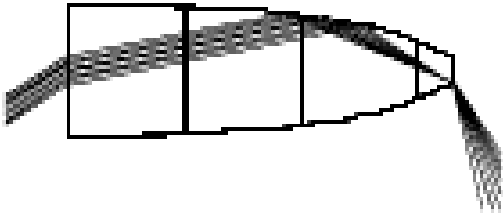


Rays Traverse Wedge

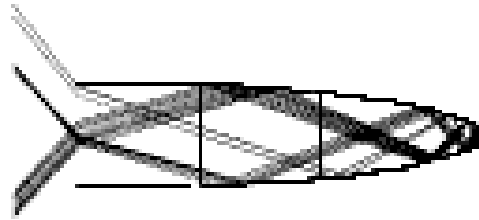
After running the macro, add more rays to the irradiance map. Select Options, then click AddRays (making sure that it is now checked). Next click Update All (blue and red fire-works-like button), and now update the irradiance distribution by clicking on the View Intensity Map button.

Building a Solid CPC

A hybrid object can be created by defining a solid object (lens or rod) and a shell object (CPC, cone, cylinder, or ellipsoid), and by having each object determine part of the extent of the hybrid object. In this case a CPC is placed to coincide with a lens (with flat ends). Note that the CPC is not placed using the Place → Inside feature — this allows (after the CPC surface has been made transmitting) TIR to occur at the CPC (because the default is to exit into air). See the macro Solid_CPC.mac (located in the Samples subdirectory).



Rays Passed by a Solid CPC



Rays Rejected by a Solid CPC

End of Tutorial II Chapter

New

Function:

Deletes all objects

Discussion:

Deletes all objects, operational parameters are not reset. If the current configuration has not been saved, OptiCAD will prompt the user to save it.

This feature can also be implemented by using a toolbar button with the same name.

Text Command:

`clear`

Open

Function:

Retrieves a stored optical configuration

Discussion:

Retrieves an optical configuration that was previously stored in the opt format. The main differences between a mac file and a opt file is that 1) the opt file can not contain operational commands (as can a mac file) such as ray-trace, view, macro, and energy; 2) the opt file will be read faster than a mac file.

This feature can also be implemented by using a toolbar button with the same name.

Parameter:

A file name.

Run Macro

Function:

Runs a macro

Discussion:

The RunMacro command allows the user to perform a predefined sequence of OptiCAD commands that are read in from a user-generated file. The file containing a command sequence may be generated externally to OptiCAD and saved as an ASCII text file. See: **Macro Command Line Structure** section of the **User Interface** chapter.

Special Note: The Escape key will interrupt ray-trace calculations and macros. If a macro is interrupted, it is wise to click File → New to clear any leftover half-created objects.

Note that there is a text only option allowing a macro to be run multiple times by specifying the parameter n (the default is 1).

Parameters:

A file name.

Text Command:

```
macro/run/name = "file_name"  
macro/run/name = "file_name" n
```

Text Command Example:

```
macro/run/name = "abbe.wdm"  
macro/run/name = "abbe.wdm" 3
```

Import CAD

Function:

Imports a CAD File

Discussion (other than STL files):

The Import CAD feature allows the user to import a CAD file. Files may contain multiple objects, each with their own location and orientation. Imported files that were previously exported by OptiCAD will contain all of the material and surface properties of the original OptiCAD objects.

IGES

Discussion:

The Import CAD feature allows the user to import the nonproprietary IGES (Initial Graphics Exchange Specification) CAD format, which is then converted to and raytraced as a continuous (unfaceted) NURBS (Nonuniform Rational B-spline Surface). Many CAD programs output data in an IGES format.

OptiCAD will import and raytrace the following IGES surface entity types as shell objects: 108 Plane (form 1: only a bounded plane); 114 Parametric Spline Surface; 118 Ruled Surface (form 0: 106 Linear Path, 110 Straight Segment; form 1: 106 Linear Path, 110 Straight Segment, 112 Cubic Spline, 126 NURBS Curve); 120 Surface of Revolution (only model space trimming curves); 122 Tabulated Cylinder (only model space trimming curves); 124 Transformation Matrix; 128 Rational B-Spline Surface; 141 Boundary; 142 Curve on a Parametric Surface; 143 Bounded Surface; 144 Trimmed (Parametric) Surface; 186 Manifold Solid B-Rep Object (limited support: this solid entity is read, and the faces that make up the solid are created as individual trimmed surfaces); 308 Subfigure Definition; 408 Singular Subfigure Instance; 504 Edge; 508 Loop; 510 Face; 514 Shell (limited support: this solid entity is read, and the faces that make up the solid are created as individual trimmed surfaces).

OptiCAD will import and raytrace IGES 186 entities as solid objects. OptiCAD will raytrace a solid object consisting of the surfaces described above if they represent a true solid, with no gaps and all surface normals pointed outward. If an IGES object does not have all of its surface normals pointed outward, OptiCAD will attempt to correct the surface normals, and optionally attempt to sew the surfaces together.

OptiCAD will not import (or draw or raytrace) any IGES line entity types, such as: 100 Circular Arc; 102 Composite Curve; 104 Conic Arc; 106 Copious Data; 110 Line; 112 Parametric Spline Curve; 116 Point; 126 Rational B-Spline Curve Entity. If you have any of these entity types in your IGES file, they will be ignored. Lines and points cannot be raytraced.

ProEngineer Note:

The following settings in ProEngineer may be helpful in generating an IGES file that ray-traces correctly (with all surface normals pointing outwards):

force_surface_normals = yes

iges_out_trim_xyz = yes

Parameters:

A file name, the file extension selected by the user determines the type of file to be imported.

If an object name is found in the CAD file, it will override an object name entered in the text command.

Text Command:

```
importcad/file="multi_object_cad_file_name"  
importcad/mergedcad/file="merged_cad_file_name" objectname refl absor n
```

Text Command Example:

```
importcad/file="C:\opticad9\mycadfile.igs"  
importcad/mergedcad/file="C:\opticad9\mymergedcadfile.igs" 0 0 1.5
```

STL

Discussion:

The Import STL feature allows the user to import an STL (stereo lithography) CAD file, and automatically convert it to an OptiCAD polynet. The STL format consists of triangles, and is simpler than IGES and DXF formats. Many CAD programs output data in an STL format.

Note that there is a limit of 32,000 polygons for an imported STL file.

Parameters:

A file name, an object (to be created) name, the reflectance, absorptance, and index of refraction of the polygon. Note the file extension selected by the user determines the type of file to be imported.

Import As Single CAD Object

Function:

Imports a CAD File as a Single OptiCAD Object

Discussion:

This function is a special case of the more generalized Import CAD function (see description above). All of the surfaces in the imported CAD file are combined into a single OptiCAD object, with individual surfaces. This merged NURBS object can then be saved to a CAD with all modified object properties intact.

It is envisioned that this function would be used once in order to build a solid or other composite object from a collection of surfaces.

If an object name is found in the CAD file, it will override an object name entered in the text command.

Text Command:

```
importcad/mergacad/file="SimpleLens.igs" objectname reflectance  
absorptance index_of_refraction
```

Text Command Example:

```
importcad/mergacad/file="SimpleLens.igs" objectname 0.1 0.2 3.1
```

Save

Function:

Saves the current optical configuration in a model (.opt) format

Discussion:

The current configuration is saved in a compressed format. The compressed format contains the most recent values of system parameters, with no history of their evolution. The model format file ends with the extension opt.

All object positions are saved as x, y, and z locations and x, y, and z rotation angles for orientation. All macro comments are removed from the body of a macro and appended to the end of the saved file.

A saved compressed file can be retrieved using File → Open. *This feature can also be implemented by using a toolbar button with the same name.*

Parameters:

A file name.

Save Model As

Function:

Saves the current optical configuration in a model (.opt) format

Discussion:

The current configuration is saved in a compressed format. The compressed format contains the most recent values of system parameters, with no history of their evolution. The model format file ends with the extension opt.

All object positions are saved as x, y, and z locations and three rotation angles for orientation. All macro comments are removed from the body of a macro and appended to the end of the saved file.

The Save Model function will save (in a separate file) and reference (in the opt file) any NURBS objects, as well as saving native objects in their original form.

A saved compressed file can be retrieved using File → Open.

Parameters:

A file name.

Save History (Macro)

Function:

Saves the **current** optical **configuration** in an **evolutionary** (.mac) **format**

Discussion:

The current configuration is saved with its full evolutionary history. The saved file also includes ray-tracing and analysis commands.

All object positions are saved as a series of incremental x, y, and z locations and changes in locations. All orientations are saved as incremental rotations about x, y, and z. All macro comments are saved in their original locations in the file.

Note that the saved macro format contains the entire history of the session after the most recent **clear** command (or **New** menu selection).

A saved History file can be run using File → Run Macro. *This feature can also be implemented by using a toolbar button with the same name.*

Parameters:

A file name.

Save CAD

Function:

Saves Objects to a **CAD File**

Discussion:

All existing NURBS objects will be directly exported, all non NURBS objects (with the exception of lightsources and films) will be converted to a NURBS object then exported.

Object material and surface properties and object position and orientation information will be saved in the exported CAD file. Saved information will be in a form that can later be retrieved when the CAD file is reimported back into OptiCAD.

The file extension selected by the user determines the type of file to be imported.

Note: The host object name for an object that is located inside another object will not be saved in an exported CAD file. When reimporting an embedded object, the user must go to the Placement page of the Object and Surface Properties box and click on the name of the host object.

An alternative to the Save CAD function is the Save Model function, which saves NURBS as NURBS and native objects as native objects (including lightsources and films).

Print....

Function:

Prints current **screen**

Discussion:

Standard Windows print routine. For more information see the Windows manual.

This feature can also be implemented by using a toolbar button with the same name.

Print Preview....

Function:

Print **previews** current **screen**

Discussion:

Standard Windows print preview routine. This function is not available in the Open GL viewing mode.

Print Setup....

Function:

Modifies Windows' printer parameters

Discussion:

Standard Windows print setup routine. For more information see the Windows manual.

Save Metafile

Function:

Saves the current screen view to a file in a Windows Enhanced Metafile format. This function is not available in the Open GL viewing mode.

Discussion:

The Windows Enhanced Metafile format saves line work (rays and line drawings) in a vector format rather than as a bit map.

Note that this function is not available when using Open GL graphics.

SaveRaysAsText

Function:

Saves ray intercept **data** to a **file**

Discussion:

The coordinates of every ray-surface intercept are saved to a text file. The user may specify file name; the default name is rays.txt, and it overwrites an existing ray text file. This function replaces the old DOS routine outrays.exe.

Parameters:

A file name.

SaveSpotsAsText

Function:

Saves film plane **data** to a **file**

Discussion:

The coordinates, direction cosines, and energy of every ray hitting the film plane are saved to a text file. The user may specify file name; the default name is spot.txt, and it overwrites existing spot text file. This function replaces the old DOS routine outspot.exe.

Parameters:

A file name.

Text Command:

```
save/spotstextfile/filename="file name"
```

Text Command Example:

```
save/spotstextfile/filename="spotdata.txt"
```

EditFile

Function:

Opens Notepad editor

Discussion:

Opens the Notepad.exe text editor from within OptiCAD. Allows editing of macro files without leaving the OptiCAD program. File selection is made from a list of mac or opt files in the current directory. For file modifications to be in effect, you must save the modified file and then close Notepad before returning to OptiCAD.

Parameters:

Menu-selectable file name.

Recent Files

Function:

List of **last four files run**

Discussion:

Displays a list of the last four files that have been run. While clicking on one of these file names will run the file, the user should use caution as this will not change the host directory and may not run subsidiary macros correctly.

EditLastMacro

Function:

Edits last macro

Discussion:

Opens the Notepad.exe text editor from within OptiCAD, allowing editing of the last macro file that has been run without leaving the OptiCAD program. Be sure to close Notepad before returning to OptiCAD.

ReRunLastMacro

Function:

Reruns last macro

Discussion:

Reruns the last macro file (.mac) that has been run, does not rerun the last.opt file that has been run. *This feature can also be implemented by using a toolbar button with the same name.*

ClearReRunMacro

Function:

Deletes all objects then Reruns last macro

Discussion:

Deletes all objects then reruns the last macro file that has been run. Operational parameters are not reset. If the current configuration has not been saved, OptiCAD will prompt the user to save it.

Exit

Function:

Exits OptiCAD

End of File Chapter

CopyText

Function:

Copies text to Windows **clipboard**

Discussion:

Copies the last text item displayed in the OptiCAD Text Out box to the Windows clipboard.

CopyBitmap

Function:

Copies bitmap to the **clipboard**

Discussion:

The current active view(s) may be copied from the screen to the Windows clipboard.

Copy Metafile

Function:

Copies the current **screen view(s)** to the **clipboard** as a Windows Enhanced Metafile.

Discussion:

The Windows Enhanced Metafile format saves line work (rays and line drawings) in a vector format rather than as a bit map. This function is not available in the Open GL viewing mode.

PasteBitmap

Function:

Copies bitmap from the **clipboard**

Discussion:

The contents of the Windows clipboard can be copied to the screen.

Activate

Function:

Reactivates the **selected object**

Discussion:

Reactivates the selected object if it is not currently active.

Parameters:

Most recent object that was selected using the OptiCAD Objects box.

Text Command:

activate **object_name**

Deactivate

Function:

Deactivates the **selected** active **object**

Discussion:

Deactivates the selected object if it is currently active. Deactivating an object is often more convenient than deleting or removing the object because a deactivated object has all of its location and orientation information saved.

Parameters:

Most recent object that was selected using the OptiCAD Objects box.

Text Command:

deactivate **object_name**

Text Command Example:

deactivate **lens1**

Delete

Function:

Deletes an **object**

Discussion:

Permanently deletes an object. See also Deactivate and Remove.

Parameters:

Object name.

Text Command:

delete **object_name**

Text Command Example:

delete **lens1**

Place

Function:

Places an **object** in the workspace

Overview Discussion:

Various methods for placing an object in the OptiCAD work space are described in this section.

Note that when placing a lightsource (or any other object) inside a dielectric material (an object with an index of refraction), you must use the Place/Inside option, otherwise rays will not be traced correctly.

Objects may be placed (and the placement conveniently changed) in global coordinates using the **Placement page of the OSPb**, however the placement functions described in this section may be used to perform relative and inside placements, whose coordinates are specified in the local coordinates of the host object.

Coordinate Systems

Coordinate systems are described in the **Coordinate Systems** section of the **Introduction** chapter.

Object Local Coordinate System

In order to place an object, you must specify the location of the object’s coordinate system with respect to the global coordinate system or with respect to the coordinate system of a placed object. Once an object is placed it can then be rotated and translated. The table below defines the object local coordinate system of each object.

<u>Object Type</u>	<u>Object Orientation</u>	<u>Object Local origin (0,0,0)</u>
Aperture	On the x-y plane	Center of the surface (on the surface)
Film	On the x-y plane	Center of the surface (on the surface)
Mirror → Flat	On the x-y plane	Center of the surface (on the surface)
Mirror → Sphere	Tangent to the x-y plane at (0,0,0)	Center of the surface (on the surface)
Mirror → Parab	Tangent to the x-y plane at (0,0,0)	Center of the surface (on the surface)
Mirror → Trough	Tangent to the x-y plane at (0,0,0)	The x-y center of the parent surface.
Mirror → CPC	Centered on the z-axis	Center of the surface (not on the surface)
Cavity → Cone	Centered on the z-axis length is defined along z	Tip (vertex) of cone

Cavity → Cylinder	Centered on the z-axis length is defined along z	Center of the object (not on the surface)
Cavity → Ellipsoid	Centered on the z-axis length is defined along z	Center of the object (not on the surface)
Axis	Parallel to coordinate system of placement	Center of the axis
Lightsource	Parallel to coordinate system of placement points in the + z direction	Apparent origin of the axis
Lens → Spher (center option)	Thickness is defined along z-axis	Center of the object (half way between surface 1 and 2 in z)
Lens → Spher (vertex option)	Thickness is defined along z-axis	Center of the first surface
Lens → Cylindrical	Thickness is defined along z-axis	Center of the first surface
Rod	Thickness is defined along z-axis	Center of the object (halfway between surface 1 and 2 in z)
Polynet	Facets defined in local 3-D coordinate system	Facets defined in local 3-D coordinate system

PlaceGlobal

Function:

Places an **object globally** in the workspace

Discussion:

Places an object in the OptiCAD workspace (workbox) by specifying the global coordinates of the object location.

Note that a global placement ignores any entries made for xangle, yangle, and zangle.

Parameters:

Object name, and x, y, and z locations.

Place					
Type					
⊗ Global	Relative			Inside	XYXRotations
Object	object_name			Reference Object	
X	x_location	Y	y_location	Z	z_location
xangle	not used	yangle	not used	zangle	not used
Clear		OK	Cancel		

Text Command:

place/global **object_name** x y z

Text Command Example:

place/global lens1 0.5 -0.5 0.0

PlaceRelative

Function:

Places an object in the workspace relative to a reference object

Discussion:

Places an object in the OptiCAD workspace (workbox) by specifying the coordinates of the object in the coordinate system of a previously placed and oriented reference object. The location of the placed object is then transformed and stored in global coordinates. A relative placement does not link the objects locations.

Note that a relative placement ignores any entries made for xangle, yangle, and zangle.

Parameters:

Object name, and x, y, and z locations.

Place					
Type					
Global	⊗ Relative			Inside	XYZRotations
Object	object_name			Reference Object	reference_object
X	x_location	Y	y_location	Z	z_location
xangle	not used	yangle	not used	zangle	not used
Clear		OK	Cancel		

Text Command:

place/relative **object_name** **reference_object** x y z

Text Command Example:

place/relative lens1 axis1 0.5 -0.5 0.0

PlaceInside

Function:

Places an object inside a solid reference object

Discussion:

Places an object inside a solid reference (host) object by specifying the coordinates of the object in the coordinate system of the previously placed and oriented reference object. The location of the placed object is then transformed and stored in global coordinates. A relative placement does not link the objects together.

Placing an object inside another object overrides the program default assumption that when a ray leaves an object it exits into air; the ray instead exits into the index of refraction of the reference object. For the inside placement, the user specifies the location of an object's local coordinate system with respect to the local coordinate system of a reference object (any previously placed object) and the name that reference object. The location of the placed object is then transformed and stored in global coordinates, and the exit index assumption for the imbedded object is changed. With this one difference, an inside placement is the same as a relative placement. An inside placement does not link the objects together.

The Place → Inside option is a flag to the ray-tracer. When leaving the embedded object, the ray is going into a medium of index equal to the host (reference) object (rather than air).

It is the user's responsibility to ensure that an object placed inside another object is completely inside that object. If an object that has been placed inside a host object actually sticks out, the ray-trace will not work properly.

A lightsource may be placed inside a solid object, but an object should not be placed inside either a lightsource or an axis.

If you want to change an object from being inside a host object to being in air, you must delete the object and then define a new object for placement in air.

Note that an inside placement ignores any entries made for xangle, yangle, and zangle.

Parameters:

x, y, and z locations.

Place					
Type					
Global	Relative			⊗ Inside	XYZRotations
Object	object_name			Reference Object	reference_object
X	x_location	Y	y_location	Z	z_location
xangle	not used	yangle	not used	zangle	not used
Clear		OK	Cancel		

Text Command:

```
place/inside object_name reference_object x y z
```

Text Command Example:

```
place/inside lens1 rod1 0.5 -0.5 0.0
```

PlaceRotations

Function:

Places and orients an **object globally** in the workspace

Discussion:

The Place → Rotations feature locates and orients an object in the OptiCAD workspace (workbox) by specifying the global coordinates and the three angles (in degrees) of the object location and rotation, and an optional reference (host) object name (that the current object is *inside*). The order of the angular rotations is about x, then about y, then about z.

The Place → Rotations option is a superset of the Place → Global option. Place → Global has the additional inputs of orientation and optional host object name. Place → Rotations is used by the compressed (.opt) save routine to store object location and orientation information on a single line, and it is a complete description of an object's orientation and the index of refraction in which the object is immersed.

Parameters:

object_name, x, y, and z global locations, x, y, and z rotation angles.

Place					
Type					
Global	Relative			Inside	⊗ XYZRotations
Object	object_name			Reference Object	host_object_name
X	x_location	Y	y_location	Z	z_location
xangle	xangle	yangle	yangle	zangle	zangle
Clear		OK	Cancel		

Text Command:

```
place/rotation object_name x y z xangle yangle zangle
place/rotation object_name x y z xangle yangle zangle host_object
```

Text Command Example:

```
place/rotation lens1 0.5 -0.5 0.0 10 20 30
place/rotation lens1 0.5 -0.5 0.0 10 20 30 lens0
place/rota lens1 0.5 -0.5 0.0 10 20 30
place/rota lens1 0.5 -0.5 0.0 10 20 30 lens0
```

Rotate

Function:

Rotates an **object** about an x, y, or z axis

Overview Discussion:

Object rotation has several options and is described in sub-sections. Rotations may be about either an x-, y-, or z-axis, but only one rotation is performed at a time. The examples shown in the tables below show only a rotation about the x axis, rotations about y or z are accomplished by checking y or z in the appropriate menu. Remember, objects that have been placed using the Place → Relative or Place → Inside features are not linked to their reference object. Therefore a future translation or rotation of the reference object will not effect the position or orientation of any objects that were placed relative or inside that reference object.

Coordinate Systems

Coordinate systems are described in the **Coordinate Systems** section of the **Introduction** chapter.

Object Local Coordinate System

The object local coordinate system is described in the **Place** section of this chapter.

Rotate Local

Function:

Rotates an **object** about the object's **local** x-, y-, or z-axis

Discussion:

Rotates an object about the object's local x-, y-, or z-axis, by specifying the axis of rotation and the angle of rotation (in degrees).

Parameters:

object_name, x-, y-, or z-axis, and the rotation angle.

Rotation Tool			
Type			
Global	⊗ Local		Relative
About Axis			
⊗ x	y		z
		Object	object_name
		Angle (Degrees)	angle
		Reference Object	
	OK	Cancel	

Text Commands:

```
rotate/local/x object_name angle
rotate/local/y object_name angle
rotate/local/z object_name angle
```

Text Command Examples:

```
rotate/local/x lens1 -30
rotate/local/y lens1 30
rotate/local/z lens1 -30
```

Rotate Global

Function:

Rotates an **object** about the **global** x-, y-, or z-axis

Discussion:

Rotates an object about the object's global x-, y-, or z-axis, by specifying the axis of rotation and the angle of rotation (in degrees).

Parameters:

object_name, x-, y-, or z-axis, and the rotation angle.

Rotation Tool			
Type			
⊗ Global	Local		Relative
About Axis			
⊗ x	y		z
		Object	object_name
		Angle (Degrees)	angle
		Reference Object	
	OK	Cancel	

Text Commands:

```
rotate/global/x object_name angle
rotate/global/y object_name angle
rotate/global/z object_name angle
```

Text Command Examples:

```
rotate/global/x lens1 30
rotate/global/y lens1 -30
rotate/global/z lens1 30
```

Rotate Relative

Function:

Rotates an **object** about an x-, y-, or z-axis of a previously placed object

Discussion:

Rotates an object about the x-, y-, or z-axis of a previously placed object, by specifying the axis of rotation, the reference object name, and the angle of rotation (in degrees).

Parameters:

object_name, x-, y-, or z-axis, reference object name, and the rotation angle.

Rotation Tool				
Type				
Global		Local		⊗ Relative
About Axis				
⊗ x		y		z
			Object	object_name
			Angle (Degrees)	angle
			Reference Object	reference_object
		OK	Cancel	

Text Commands:

```
rotate/relative/x object_name reference_object angle
rotate/relative/y object_name reference_object angle
rotate/relative/z object_name reference_object angle
```

Text Command Examples:

```
rotate/relative/x lens1 rod1 -30
rotate/relative/y lens1 rod1 30
rotate/relative/z lens1 rod1 -30
```

Translate

Function:

Translates an **object** by delta_x, delta_y, and delta_z

Overview Discussion:

Object translation has two options that are described in sub-sections below. The distances specified for the Translate feature are displacements rather than new locations. Remember, objects that have been placed using the Place → Relative or Place → Inside features are not linked to their reference object, therefore, a future translation or rotation of the reference object will not effect the position or orientation of any objects that were placed relative to or inside of that reference object.

Coordinate Systems

Coordinate systems are described in the **Coordinate Systems** section of the **Introduction** chapter.

Object Local Coordinate System

The object-local coordinate system is described in the **Place** section of this chapter.

Translate Local

Function:

Translates an **object** along the object's **local** x-, y-, and z-axis

Discussion:

Translates an object along the object's local x-, y-, and z-axes.

Parameters:

The translation distances in x, y, and z.

Translation Tool				
Type				
Global	⊗ Local		Object	object_name
X		Y		Z
delta_x		delta_y		delta_z
Clear		OK	Cancel	

Text Command:

```
translate/local object_name delta_x delta_y delta_z
```

Text Command Example:

```
translate/local lens1 0.1 -0.1 0.5
```

Translate Global

Function:

Translates an **object** along the objects global x-, y-, and z-axes

Discussion:

Translates an object along the objects global x-, y-, and z-axes.

Parameters:

The translation distances in x, y, and z.

Translation Tool				
Type				
⊗ Global	Local		Object	object_name
X		Y		Z
delta_x		delta_y		delta_z
Clear		OK	Cancel	

Text Command:

```
translate/global object_name delta_x delta_y delta_z
```

Text Command Example:

```
translate/global lens1 -0.1 0.1 -0.5
```

Remove

Function:

Removes an **object** from the workspace (workbox).

Discussion:

Removes the selected object if it is currently placed in the workbox. The removed object loses all location and orientation information. See Deactivate and Delete.

Parameters:

The selected object.

Text Command:

```
remove object_name
```

Text Command Example:

```
remove lens1
```

Object Properties

Function:

Displays the **Object and Surface Properties box**.

End of Edit Chapter

Type

Perspective

Function:

Displays wire frame perspective view

Discussion:

Displays a perspective wire frame view of all activated and placed objects in the viewing window. The view window size is set using SetWorkbox, SnugWorkbox, and Zoom In feature. The view point (azimuth and elevation angles) is set using the select view point toolbar button, and the viewing distance is set using the Windows → Perspective feature. Rays will be displayed if View → Rays is selected (checked). *This feature can also be implemented by using a toolbar button named View System.*

Text Commands:

```
view/system azimuth_angle elevation_angle perspective_distance
view/system/rays azimuth_angle elevation_angle perspective_distance
```

Text Command Examples:

```
view/system 135 -30 1E10
view/system/rays -135 30 100
```

Orthographic

Function:

Displays four wire frame views

Discussion:

The screen is subdivided into four parts. A perspective wire frame view, and an x, y, and z view are displayed simultaneously. The view window size is set using SetWorkbox, SnugWorkbox, and Zoom In feature. The view point (azimuth and elevation angles) is set using the select view point toolbar button, and the viewing distance is set using the Windows → Perspective feature. Rays will be displayed if View → Rays is selected (checked). *This feature can also be implemented by using a toolbar buttons named Side (X) View, Top (Y) View, or Front (Z) View.*

Text Commands:

```
view/ortho azimuth_angle elevation_angle perspective_distance
view/ortho/rays azimuth_angle elevation_angle perspective_distance
```

X or Y or Z

Function:

Displays wire frame view looking into the x-, y-, or z-axis

Discussion:

Displays a view looking into the x-, y-, or z-axis, of all objects that are both activated and placed. The view window size is set using SetWorkbox, SnugWorkbox, and Zoom In feature. Rays will be displayed if View → Rays is selected (checked). *These features can also be implemented by using a toolbar button with the same name.*

Text Commands:

```
view/x  
view/x/rays  
view/y  
view/y/rays  
view/z  
view/z/rays
```

Film

Function:

Displays ray locations on film surface

Discussion:

Displays a spot diagram of ray location (and color-coded energy) on a film surface. The user may Zoom In on a subarea, (double-click to ZoomOut). If the active object is not a film object, the spot diagram from the last film object used will be displayed. *This feature can also be implemented by using a toolbar button with the same name.*

Parameter:

The active **film_object** (previously selected in the OptiCAD Object box).

Text Command:

```
view/film film_object
```

Text Command Example:

```
view/film film1
```

Energy

Function:

Displays results of an Energy calculation

Discussion:

Displays a plot of the 1-D integrated energy calculation. If the active object is not a film object, the energy calculation from the last film object used will be displayed. *This feature can also be implemented by using a toolbar button with the same name.*

Parameter:

The active **film_object** (previously selected in the object selection box).

Text Command:

```
view/energy film_object
```

Text Command Example:

```
view/energy film3
```

View Radiometer

Function:

Displays results of a **Radiometer** calculation

Discussion:

The units displayed are shown below:

Irradiance radiometer displays power per area (watts/cm² or lumens/m²).

Intensity radiometer displays power per steradian.

Radiance radiometer displays power per area per steradian (watts/cm²/steradian or lumens/m²/steradian).

The menu options available using **View -> Palette** controls the radiometer palette display. See the **Radiometer Palettes** section of this chapter.

Polynet and film plane irradiance radiometers are displayed (in false color) in the system (perspective) view.

If the message "Unable to stretch bits" appears while attempting to view a Radiometer, go to the Control Panel and then select Display, reset the screen resolution the next lower resolution, making sure that the setting has at least 256 bits per pixel.

Viewing the Radiometer is accomplished by selecting an object with a radiometer attached and then clicking on the **View Radiometer Map** toolbar button.

Text Commands:

```
view/radiometer object
```

```
view/irradiance object
```

```
view/intensity object
```

Text Command Examples:

```
view/radiometer sphere1
```

```
view/irradiance film1
```

```
view/intensity sphere1
```

Polar Radiometer

Function:

Displays results of an **Polar Radiometer** calculation

Discussion:

Displays a polar plot of the polar radiometer calculation.

Text Command:

```
view/polar film_object
```

Box

Function:

Adds a 3-D box around the viewing (workbox) box

Parameters:

Toggles on (checked) or off (unchecked).

Rays

Function:

Displays rays with wire frame view

Parameters:

Toggles on (checked) or off (unchecked).

Text Commands:

```
view/system/rays azimuth_angle elevation_angle perspective_distance
view/x/rays
view/y/rays
view/z/rays
view/ortho/rays azimuth_angle elevation_angle perspective_distance
```

Text Command Examples:

```
view/system/rays 135 45 100
view/x/rays
view/y/rays
view/z/rays
view/ortho/rays -135 -45 1E5
```

Radiometer Palettes

Function:

Selects the **color palette** of a radiometer

Discussion:

This feature is used to select the palette of a radiometer. Options gray, red, and green have a black background for 0.0 irradiance, the spectrum option has a red background for 0.0 irradiance, and the blue (default) option has a white background for 0.0 irradiance.

Parameters:

Gray, red, green, blue (default), and spectrum.

Text Commands:

```
view/irradiance/gray film_object
view/irradiance/red film_object
view/irradiance/green film_object
view/irradiance/blue film_object
view/irradiance/spectrum film_object
view/intensity/gray film_object
view/intensity/red film_object
view/intensity/green film_object
view/intensity/blue film_object
view/intensity/spectrum film_object
```

Monochrome

Function:

Displays **wire frame** views using only **black lines**

Discussion:

Setting the screen to display wire frame views using only black lines will often result in better black and white hard copies.

Parameters:

Toggles on (checked) or off (unchecked).

PolyNormal

Function:

Displays local surface **normals** for **polynet** objects

Discussion:

Viewing the local surface normals for polynet objects may be useful for identifying physically unreal polynets.

Parameters:

Toggles on (checked) or off (unchecked).

Grid

Function:

Defines a **reference grid**

Discussion:

When the grid is activated or reactivated, a box appears to enter the grid spacing. This function is not available in the Open GL viewing mode.

Parameters:

Toggles on (checked) or off (unchecked).

Frame

Function:

Adds a **frame** to a printed **view**.

Discussion:

This function is not available in the Open GL viewing mode.

Parameters:

Toggles on (checked) or off (unchecked).

HighLight

RaySegment

Function:

Displays a **ray segment** in a different color.

Description:

Click on Select View → RaySegment, in order to toggle on (check marked) or off (not check marked).

Click once in the viewing screen.

Use the left and right arrows to scroll between ray segments (sections of the same ray) and use the up and down arrows to scroll between different rays.

A ray number and a segment number are displayed in the lower right of the screen (as ray number - segment number).

Facet/Surface

Function:

Displays selected surface in a different color.

Description:

Click on Select View → Surface/Facet, in order to toggle on (check marked) or off (not check marked).

Native OptiCAD Object (nonpolynet)

This feature is available in both the Open GL and non Open GL (default Windows) viewing modes.

Select and expand an object name from the OptiCAD Object box, and then click on name of the surface that you wish to highlight.

The selected surface will be displayed in a different color.

Native OptiCAD Object (polynet)

Highlighting polynet surfaces is available in the Windows (non-OpenGL) viewing mode and is controlled by the arrow keys. The program focus needs to be in the main Opticad window. This is achieved by clicking the mouse once in the main window. Step by step instructions are here:

1. With View -> Highlight Facets/Surfaces on,
2. Select the polynet object in the Object Property box
3. Click the mouse once in the main window
4. Set either the Point Select or Box Sizing mode on the toolbar.
5. Turn off the OpenGL wireframe and solid views by toggling the toolbar buttons
6. Using the right/left arrows, it is now possible to highlight the various polynet surfaces.

NURBS Object

This feature is available only in the Open GL viewing mode.

Select and expand a NURBS object name from the OptiCAD Object box, and then click on name of the surface that you wish to highlight.

The selected surface will be displayed in a different color.

IGES Export as Linework

Function:

Exports current wire frame view as an IGES file.

Discussion:

The current wire frame view is saved as an IGES file. This file can be read into some CAD programs. The exported IGES file consists of lines (IGES 110 entities) only, the saved file cannot be imported back into OptiCAD, which requires 3-D surfaces.

If rays are displayed in the view, they will be saved in the IGES file, with the exception of ray paths that do not intersect a surface (go off into space).

Parameter:

Saved file name.

Text Command:

```
iges/rays/name="file_name.igs"
```

Text Command Example:

```
iges/rays/name="outiges.igs"
```

3D Wireframe OpenGL

Function:

Uses wire frame Open GLgraphics.

Discussion:

Wireframe Open GL (only) can be activated using the toolbar button or the text command option `view/opengl/wireframe/rays`.

Both wireframe and solid Open GL can be activated using the text command `view/opengl/all/rays`.

Both wireframe and solid Open GL can be deactivated using the text command `view/noopengl/rays`.

Note that using a `view/opengl` command without the `/rays` option will deactivate the display of rays.

This feature can also be implemented by using a toolbar button with the same name.

Parameters:

Toggles on (checked) or off (unchecked).

Text Commands:

```
view/opengl/wireframe/rays
view/opengl/all/rays
view/noopengl/rays
view/opengl/wireframe
view/opengl/all
view/noopengl
```

3D Solid OpenGL

Function:

Uses Solid (shaded) Open GL graphics.

Discussion:

Solid Open GL (only) can be activated using the toolbar button or the text command option `view/opengl/solid`.

Both wireframe and solid Open GL can be activated using the text command `view/opengl/all`.

Both wireframe and solid Open GL can be deactivated using the text command `view/noopengl`.

Note that using a `view/opengl` command without the `/rays` option will deactivate the display of rays.

This feature can also be implemented by using a toolbar button with the same name.

Parameters:

Toggles on (checked) or off (unchecked).

Text Commands:

```
view/opengl/solid/rays
view/opengl/all/rays
view/noopengl/rays
view/opengl/solid
view/opengl/all
view/noopengl
```

Display Radiometer with Open GL

Function:

Display radiometer(s) with Open GL graphics.

Discussion:

Displaying the radiometer with the Open GL view is the default.

There is no menu version of this option

Not displaying radiometer(s) with Open GL views can be accomplished using the text command `view/opengl/noradiometer`.

Viewing the radiometer can be reactivated by activating wireframe or solid Open GL or both.

Text Commands:

```
view/opengl/noradiometer
view/opengl/all
view/opengl/wireframe
view/opengl/solid
```

Redraw

Function:

Redraws the wire frame view

Discussion:

This feature may be useful when redrawing the wire frame view is time consuming and the AutoRedraw option has not been selected.

However, a more convenient way to accomplish a redrawing is to double-click on the AutoRedraw toolbar button.

View Camera Array

Function:

Views a camera image resulting from a reverse ray-trace from a detector

Camera and Reverse Raytrace Radiometry Results

Numerical Data:

The total power accumulated by the reverse ray-trace, the total power normalized by the detector area (DLL based source distributions only), the total power collected by the camera array, and the total camera array power normalized by the detector area (DLL based source distributions only) are displayed in the Text Out box at the bottom of the screen.

Camera Image

The camera array is displayed in a pop-up window by using the feature **View->View Camera Array** (View menu). The image can be resized and saved to a bitmap, normalized by a user entered value, and exported in a comma separated values format by right clicking the mouse while in the pop-up display window. An array value of 0.0 will be displayed as black if OpenGL solid is selected and white if OpenGL solid is not selected.

Add Rays

The Add Rays option works differently for a detector than for a lightsource. When the Add Rays option is on a detector accumulates power from multiple ray-traces in a similar manner to film in a camera under going multiple exposures. Repeating the same ray-trace a second time will double the total accumulated power reported in the **TextOut box** and in the camera array.

See the **Add Rays** section of the **Options** chapter, the **Detector** section of the **Design chapter**, the **Raytrace** section of the **Analyze** chapter, and the **Camera page** section of the **Object Properties** chapter.

Text Command:

```
view/camera detector_name
view/camera/gray detector_name
view/camera/red detector_name
view/camera/green detector_name
view/camera/blue detector_name
view/camera/spectrum detector_name
view/camera/spectrum detector_name
```

Text Command Example:

view/camera detector1
view/camera/gray detector1
view/camera/red detector1
view/camera/green detector1
view/camera/blue detector1
view/camera/spectrum detector1
view/camera/spectrum detector1

End of View Chapter

Introduction

The Design chapter describes all of OptiCAD's continuous, explicitly defined objects. Polygonally faceted objects are described in the **Polynet** chapter.

There is considerable overlap between the **Object and Surface Properties** chapter and **Design** chapter. The user should consult both chapters for a more complete description of the definition and modification of surface and material properties.

Object definitions are accomplished using the **OSPb**, a selection from the **Design** menu opens the **Object and Surface Properties box**.

In order to design an object click on the **Definition page of the Object and Surface Properties box**, which will display the default values.

The design feature allows the creation of both solid and shell (surface) objects. Surface-only objects are included in two object types: mirror and cavity. All objects have a finite extent (outer dimension), and mirror and cavity objects have the option of specifying which side (or sides) is reflective. Mirror objects can later be modified by adding a hole. All objects should be assigned names. There is a limit of 5,000 objects.

When creating objects from a macro, you should enter object names, all of the options (modifiers), and all numerical values.

An object name consists of numbers, and lower case characters, and underscore (_). An object name must start with a letter, cannot contain a blank, cannot contain a non-ASCII characters, cannot contain upper case characters, and must be less than 19 characters in length.

Physical Extent of Objects

Shell Objects

Shell surface-only objects are defined with extent-limiting surfaces that do not exist as explicit surfaces.

Mirrors, apertures, films, and other single-valued objects (that do not close on themselves) are generally defined using a "cookie cutter" method. A limiting virtual elliptical cylinder or rectangle (four planes) is extruded into the z-axis, cutting off (limiting) any part of the surface that lies outside of its extent. The options (or modifiers) that are associated with this extent limitation method are rectangular for a rectangular extent, oval for an elliptical extent, and round for a circular extent. The options hoval and hrect are used to describe a rectangular or elliptical hole in an aperture.

Volumetric Objects

Volumetric objects are defined as having an extent limiting cylinder (spherical lenses and rods), or as having extent limiting planes (cylindrical lenses). The difference between volumetric objects and shell objects is that for volumetric objects the limiting surfaces are explicit surfaces. A single-element lens consists of three surfaces (two spherical surfaces plus a section of a cylinder), while a singlet cylindrical lens consists of six surfaces (two powered surfaces plus four sections of planes).

T.I.R. Option

The default reflectance for a ray that has undergone total internal reflectance (T.I.R.) at an uncoated surface is 100%. The **T.I.R. User Defined Reflectance** option described in the **Options** chapter uses the reflectance specified on the **Attributes** page for the surface reflection. This option only effects rays that have undergone T.I.R..

Shell Surface Reflectivity Options

Mirror surface reflectivities do not affect the shape of the surface.

Open Surfaces (Mirrors)

Open surfaces include flat, spherical, trough, parabolic, cpc, and conic.

Front: Rays hitting the +z side (in surface local coordinates) will be reflected; rays hitting the -z side will be absorbed.

Back: Rays hitting the -z side (in surface local coordinates) will be reflected; rays hitting the +z side will be absorbed.

Both: Rays hitting either side will be reflective.

Closed Surfaces (Cavities)

Closed surfaces include ellipsoid, cone, cylinder, and cavities.

Inside: Rays hitting the inside of an inside reflective surface will be reflected; rays hitting the outside of an inside reflective surface will be absorbed.

Outside: Rays hitting the outside of an outside reflective surface will be reflected; rays hitting the inside of an outside reflective surface will be absorbed.

Both: Rays hitting either side will be reflective.

Lightsource

Function:

Defines a source of **light**

Discussion:

There are two types of bundles, for an angular type a bundle of rays originate from the lightsource, for a collimated type a parallel bundle of rays is centered around (and parallel to) the lightsource.

A lightsource has no physical extent, but it does have an orientation (initially pointed parallel to the +z-axis). It can later be reoriented. OptiCAD permits multiple sources.

Note that the power_units are set using the **Windows** page.

Parameter:

User-defined lightsource name and the total power in the lightsource.

Text Command:

```
define/light my_light total_power
```

Text Command Example:

```
define/light light1 1
```

Power Units

The power units can be set from the **Windows** menu. The choice of power units determines the units of output displays. The description of the **Power Units** option in the **Windows** chapter is as follows:

Function:

Sets units of power

Discussion:

The choice of power units determines the units of output displays. Choosing Lumens will result in power density being displayed as lux (lumens/meter²) for a film plane detector or intensity displayed as candela (lumens/steradian) for a polynet detector; while choosing Watts will result in power density being displayed as watts/cm² and intensity as watts/steradian.

Parameters:

Lumens or Watts.

Text Command:

```
set/power_units/lumens  
set/power_units/watts
```

Reverse Ray-Trace Detector

Function:

Defines a reverse raytrace detector

Discussion:

Reverse (backwards) ray-tracing traces rays from a detector-source towards all objects and surfaces. Any surface can have an irradiance (watts/cm² or Lumens/m²) assigned. The reverse rays act as travelling power buckets, accumulating power along their paths, while undergoing refraction, reflection, transmission, absorption, splitting, and scattering.

This powerful feature is a direct (ray-tracing) solution to a radiometric integral. The detector-source can be a point source, or use any of the source DLLs in OptiCAD, allowing a surface or volume average calculation. This radiometry method is also called a Maxwellian view from the detector looking backwards. An example of this method is to calculate the power that a molecule (or volume of molecules) inside a laser rod receives from flashlamps used to pump the rod. Power is only accumulated from surfaces assigned an irradiance.

A detector acts like a lightsource in that it launches rays. The addition of a camera to a detector activates reverse ray-trace radiometry. The rays act as travelling power buckets, accumulating power from surfaces having a non-zero irradiance.

Rays launched from a single point on the detector represent a pinhole camera. Rays may be launched from a surface or volume using any source DLL.

The camera calculates the total accumulated power of detector launched rays and the power of rays that intercepted the camera.

The choice of a flat or spherical camera determines the assumptions made about the detector surface. Choosing a flat camera assumes a flat detector, choosing a spherical camera assumes a spherical detector. These assumptions effect the calculated power on the camera and the total calculated power. The total power will be greater or equal to the total power on the camera focal surface because rays may not intercept the focal surface.

Certain ray cone spacings are not permitted when performing radiometric backwards ray-tracing.

See the **Camera** page section of the **Object Properties** chapter, the **Raytrace** section of the **Analyze** chapter, and the **View Camera Array** section of the **View** chapter.

Note that the power_units are set using the **Windows** page.

Camera and Reverse Raytrace Radiometry Results

Numerical Data:

The total power accumulated by the reverse ray-trace, the total power normalized by the detector area (DLL based source distributions only), the total power collected by the camera array, and the total camera array power normalized by the detector area (DLL based source distributions only) are displayed in the Text Out box at the bottom of the screen.

Camera Image

The camera array is displayed in a pop-up window by using the feature **View->View Camera Array** (View menu). The image can be resized and saved to a bitmap, normalized by a user entered value, and exported in a comma separated values format by right clicking the mouse while in the pop-up display window.

Add Rays

The Add Rays option works differently for a detector than for a lightsource. When the Add Rays option is on a detector accumulates power from multiple ray-traces in a similar manner to film in a camera under going multiple exposures. Repeating the same ray-trace a second time will double the total accumulated power reported in the **TextOut box** and in the camera array. See the **Add Rays** section of the **Options** chapter.

Text Command:

```
define/detector my_detector
```

Film

Function:

Defines a detector (**film**) plane

Discussion:

Defines a detector plane having an oval (elliptical) or rectangular extent. A film plane is initially 100% absorbing. It may be modified to be 100% transmitting, however, it will not conserve energy. No other absorptance values should be used. Every time a ray hits a transparent film its' energy is added to the film (or radiometer), the film doesn't know the ray history as to whether its' energy has previously been added to the film.

See the **Film** section of the **View** chapter and the **Energy** section of the **Analyze** chapter for film display and analysis. Multiple film planes are permitted.

Parameters:

User-defined film name, extent shape (rectangular or oval), x half-size, y half-size.

Film	
Object name	object name
xhw	x half size
yhw	y half size

Rectangular

Text Command:

```
define/film/rectang my_film x_half_size y_half_size
```

Text Command Example:

```
define/film/rectang film1 1.5 1.5
```

Oval

Text Command:

```
define/film/oval my_film x_half_size y_half_size
```

Text Command Example:

```
define/film/oval film1 2.5 2.5
```

Axis

Function:

Defines a reference **axis**

Discussion:

Defines an axis that is useful as a reference for using the Place → Relative and Rotate → Relative features. The axis length is only used for drawing purposes.

Parameters:

User-defined axis name, axis length.

Axis	
Object name	<code>object name</code>
Size	<code>axis full length</code>

Text Command:

```
define/axis my_axis size
```

Text Command Example:

```
define/axis axis1 0.001
```

Mirror

General Discussion:

A mirror is an open shell object (surface of no thickness). See Cavity, in this section, for a description of closed shell objects. A mirror is initially 100% reflective, but its reflectance, absorptance, transmittance (by default), and diffuse scattering can be modified.

See sections: Physical Extent of Objects and Shell Surface Reflectivity Options at the start of the chapter.

Flat

Note that a flat mirror is treated as a special shape. The shape of a flat mirror should not be modified; instead, start with a spherical or parabolic mirror.

Parameters:

User-defined mirror name, reflective side choice, x half-size, and y half-size.

<u>Flat Mirror</u>	
Object name	object name
xhw	x half width
yhw	y half width
Front	⊗
Back	⊗
Both	⊗

Rectangular

Function:

Defines a flat mirror of **rectangular** extent

Discussion:

Defines a flat mirror of rectangular extent and reflective side options.

Text Commands:

```
define/mirror/flat/rectangle/front name xhw yhw  
define/mirror/flat/rectangle/back name xhw yhw  
define/mirror/flat/rectangle/both name xhw yhw
```

Text Command Examples:

```
define/mirror/flat/rectangle/front flat1 1.0 1.0  
define/mirror/flat/rectangle/back flat1 1.0 1.0  
define/mirror/flat/rectangle/both flat1 1.0 1.0
```

Oval

Function:

Defines a flat mirror of **elliptical** extent

Discussion:

Defines a flat mirror of elliptical extent and reflective side options.

Text Commands:

```
define/mirror/flat/oval/front name xhw yhw  
define/mirror/flat/oval/back name xhw yhw  
define/mirror/flat/oval/both name xhw yhw
```

Text Command Examples:

```
define/mirror/flat/oval/front flat2 2.0 3.0  
define/mirror/flat/oval/back flat2 2.0 3.0  
define/mirror/flat/oval/both flat2 2.0 3.0
```

Spherical

The single-valued expression for a spherical surface profile is:

$$z = \frac{cv \cdot h^2}{1 + \sqrt{1 - (cv^2 \cdot h^2)}}$$

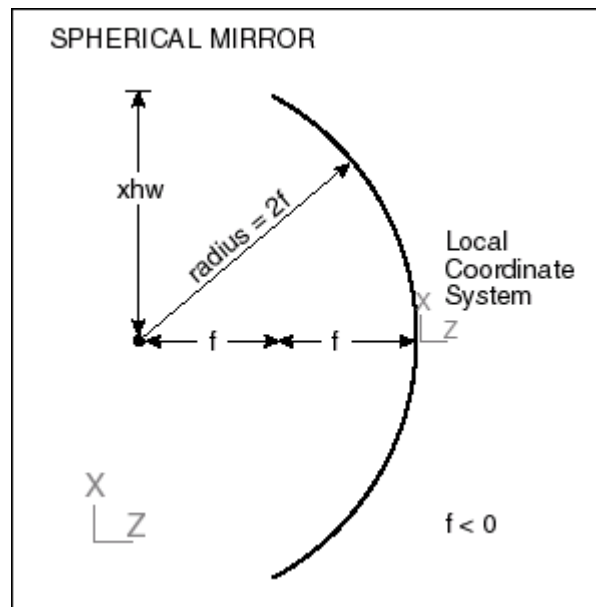
where (cv) is the curvature.

$$cv = \frac{1}{radius} = \frac{1}{2 \cdot focallength}$$

and

$$h = \sqrt{x^2 + y^2}$$

A spherical mirror is tangent to the local x-y plane.



Parameters:

User-defined mirror name, mirror focal length, reflective side choice, x half-size, and y half-size.

<u>Spherical Mirror</u>	
Object name	object name
fl	focal length
xhw	x half width
yhw	y half width
Front (Concave)	⊗
Back (Convex)	⊗
Both	⊗

Spherical - Oval

Function:

Defines a **spherical** mirror of **elliptical** extent

Discussion:

Defines a spherical mirror of elliptical extent and reflective side options.

Text Commands:

```
define/mirror/spherical/oval/concave name fl xhw yhw
define/mirror/spherical/oval/convex name fl xhw yhw
define/mirror/spherical/oval/both name fl xhw yhw
```

Text Command Examples:

```
define/mirror/spherical/oval/concave sphr1 -5 1 1
define/mirror/spherical/oval/convex sphr1 -5 1 1
define/mirror/spherical/oval/both sphr1 -5 1 1
```

Rectangular

Function:

Defines a **spherical** mirror of **rectangular** extent

Discussion:

Defines a spherical mirror of rectangular extent and reflective side options.

Text Commands:

```
define/mirror/spherical/rectangle/concave name fl xhw yhw
define/mirror/spherical/rectangle/convex name fl xhw yhw
define/mirror/spherical/rectangle/both name fl xhw yhw
```

Text Command Examples:

```
define/mirror/spherical/rectangle/concave sphr1 10 2 2
define/mirror/spherical/rectangle/convex sphr1 10 2 2
define/mirror/spherical/rectangle/both sphr1 10 2 2
```

Trough

Function:

Defines a **parabolic trough**

Discussion:

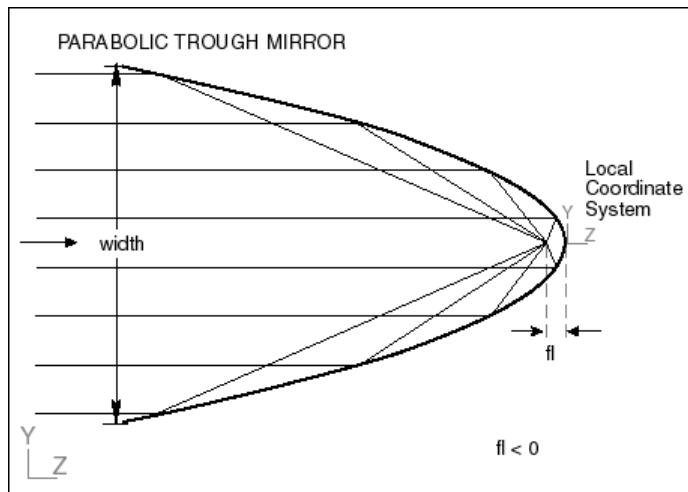
Defines a parabolic trough with power (curvature) in y, with extent and reflective side options. Note that an elliptical trough is described in this section under Cavity. The single-valued expression for a parabolic trough surface profile with power in y is:

$$z = \frac{cv_y \cdot y^2}{2}$$

where cv_y is the curvature in y

$$cv_y = \frac{1}{radius_y} = \frac{1}{2 \cdot focallength_y}$$

A parabolic trough is tangent to the local x-y plane.



Parameters:

User-defined trough name, reflective side choices, specify the y focal length, the full length in the noncurved direction (along the local x-axis), and the full width in the curved direction (along the local y-axis).

Trough Mirror	
Object name	object name
y focal length	f1
full length in x	length
full width in y	width
Both	⊗
Front (inside)	⊗
Back (outside)	⊗

Text Commands:

```
define/mirror/trough/concave name f1 length width
define/mirror/trough/convex name f1 length width
define/mirror/trough/both name f1 length width
```

Text Command Examples:

```
define/mirror/trough/concave trough1 10 5 5
define/mirror/trough/convex trough1 10 5 5
define/mirror/trough/both trough1 10 5 5
```

Parabolic

The single-valued expression for a parabolic surface profile is:

$$z = \frac{cv_x \cdot x^2 + cv_y \cdot y^2}{2}$$

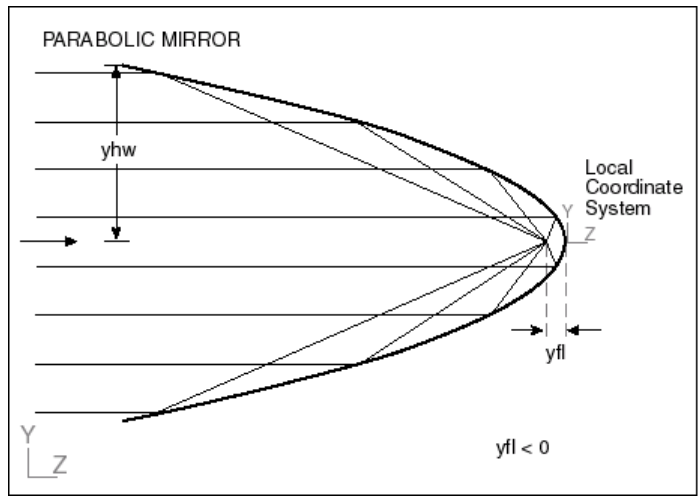
where (cv_x) is the curvature in x, (cv_y) is the curvature in y

$$cv_x = \frac{1}{radius_x} = \frac{1}{2 \cdot focallength_x}$$

and

$$cv_y = \frac{1}{radius_y} = \frac{1}{2 \cdot focallength_y}$$

A parabolic mirror is tangent to the local x-y plane.



Parameters:

User-defined mirror name, mirror x_focal length, mirror y_focal length, reflective side choice, x half-size, and y half-size. Both focal lengths must be of the same sign.

Parabolic Mirror	
Object name	<code>object name</code>
<code>flx</code>	x focal length
<code>yfl</code>	y focal length
<code>xhw</code>	x half width
<code>yhw</code>	y half width
Front (Concave)	⊗
Back (Convex)	⊗

Oval

Function:

Defines a **paraboloid** with **elliptical** extent

Discussion:

Defines a parabolic mirror of elliptical extent.

Text Commands:

```
define/mirror/parab/oval/concave name fl yfl xhw yhw
define/mirror/parab/oval/convex name fl yfl xhw yhw
define/mirror/parab/oval/both name fl yfl xhw yhw
```

Text Command Examples:

```
define/mirror/parabolic/oval/concave parab2 -10 -10 5 5
define/mirror/parabolic/oval/convex parab2 -10 -10 5 5
define/mirror/parabolic/oval/both parab2 -10 -10 5 5
```

Rectangular

Function:

Defines a **paraboloid** with **rectangular** extent

Discussion:

Define a parabolic mirror of rectangular extent.

Text Commands:

```
define/mirror/parab/rect/concave name fl yfl xhw yhw
define/mirror/parab/rect/convex name fl yfl xhw yhw
define/mirror/parab/rect/both name fl yfl xhw yhw
```

Text Command Examples:

```
define/mirror/parab/rectangle/concave 10 10 10 10
define/mirror/parab/rectangle/convex 10 10 10 10
define/mirror/parabolic/rectangle/both 10 10 10 10
```

CPC

Function:

Defines a **Compound Parabolic Concentrator**

Discussion:

OptiCAD models a modified CPC (Compound Parabolic Concentrator, also called a Winston collector) in which an input (smaller) angle space is mapped to an output (larger) angle space with a high geometric efficiency. The surface shape is a parabolic arc that is tilted and then rotated about its old axis. For maximum permitted output half angles of less than 90° a cone is appended to the rear of the CPC.

Rays entering at greater than the maximum acceptance angle will be rejected after several bounces.

A CPC object consists of two surfaces the first is a modified CPC and the second is an appended cone.

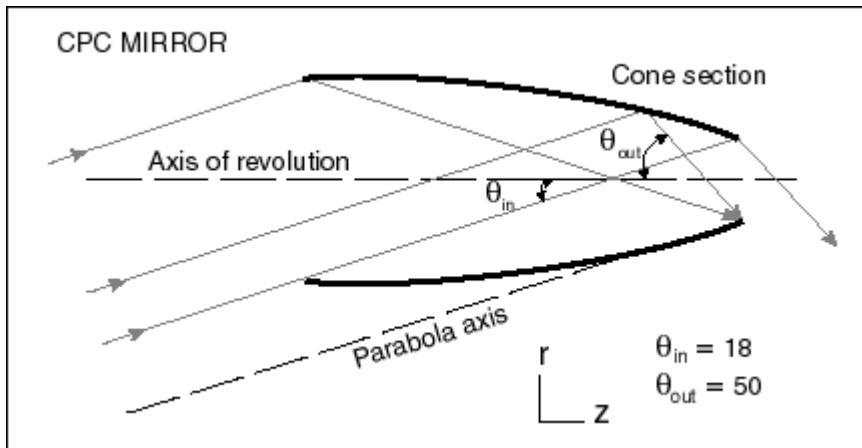
The user should avoid tracing rays that are incident on the CPC at exactly the maximum input angle.

The energy density (E) is concentrated from the input to the output by a factor of:

$$\frac{E_{OUT}}{E_{IN}} = \left(\frac{\sin \theta_{OUT}}{\sin \theta_{IN}} \right)^2$$

Note that modifying the extents of a cpc in x and y, and modifying the zmax extent beyond the maximum length of the true surface are not reflected in the drawing of the cpc surface.

The user may wish to run the DOS program CPCDATA.EXE, which gives two simple forms of the cpc equation.



Parameters:

User-defined object name, input angle (degrees), output angle (degrees), radial size of the output hole, and reflectivity side choices.

CPC Mirror	
Object name	object name
input_angle	input angle
output_angle	output angle
radius_exit	exit half size
Both	⊗
Front	⊗
Back	⊗

Text Commands:

```
define/mirror/cpc/concave name input_angle output_angle radius_exit
define/mirror/cpc/convex name input_angle output_angle radius_exit
define/mirror/cpc/both name input_angle output_angle radius_exit
```

Text Command Examples:

```
define/mirror/cpc/concave concen1 6 60 1
define/mirror/cpc/convex concen1 6 60 1
define/mirror/cpc/both concen1 6 60 1
```

Lens

Lenses described in this section have a single element. Air-spaced lenses may be built by defining multiple single-element lenses, and placing them in the proper location. Cemented lenses may be built by defining multiple single-element lenses, and gluing them together,

using the ProximityGlue feature.

OptiCAD does not check for all unreal lenses, it only checks to see that the diameter is less than twice the smallest radius.

Index of refraction will only be calculated for wavelengths that are within the valid range for that material. Glass catalogs data used by OptiCAD are in files ending with the agf suffix. The glass files must be located in the OptiCAD home directory (directory that contains opti-cad.exe).

A material name is case sensitive. Glass names must be entered in upper case. The glass catalog containing a selected glass must have been loaded (see set/matcatalog in the **Options** chapter) prior to any use of the glass in an object definition.

Spherical lens

Function:

Defines a single-element **spherical** lens

Discussion:

A single-element spherical lens is a solid object whose extent is determined by a cylinder. The spherical lens consists of sections of two spheres whose centers are located on the axis of the extent defining cylinder. Wedge or tilt between the surfaces is not permitted.

It is important to note that the default for the local coordinate system of a spherical singlet is at the vertex of the first surface. This may be overridden by selecting the option `sphlensvertex/on`

described in the **Options** chapter.

A zero entry for r1 or r2 will result in the curvature, not the radius being set to zero, indicating a plano surface.

The single-valued expression for a spherical surface profile is:

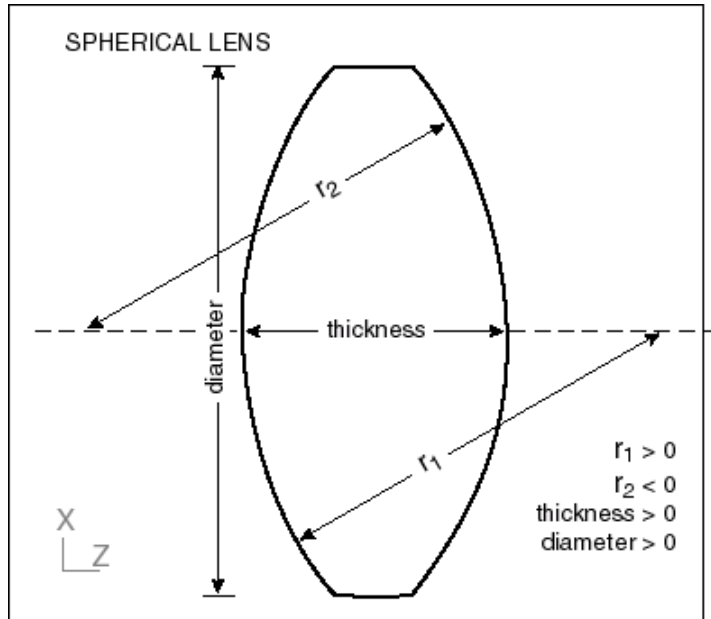
$$z = \frac{cv \cdot h^2}{1 + \sqrt{1 - cv^2 \times h^2}}$$

where (cv) is the curvature

$$cv = \frac{1}{radius}$$

and

$$h = \sqrt{x^2 + y^2}$$



Parameters:

User-defined object name, r1 (the first radius of curvature on surface my_lens.1), r2 (the second radius of curvature on surface my_lens.2), the vertex thickness (distance along the local z-axis), the lens diameter, and either the index of refraction or a material name. A lens material may be specified in the Object and Surface Properties box.

Note, if entering definition in text form the correct syntax is **spherical** or **spher**, do not type **sphere**.

Spherical Lens	
Object name:	object name
r1	radius 1
r2	radius 2
thickness	thickness
diameter	diameter
n	index of refraction

Text Command:

```
define/lens/spherical name r1 r2 thickness diameter n
define/lens/spher/material=GLASS_NAME name r1 r2 thick diameter
```

Text Command Example:

```
define/lens/spherical lens1 -10 10 3 5 1.5
define/lens/spherical/material=BK7 lens1 -10 10 3 5
```

Two Surface Spherical lens

Function:

Defines a **two surface spherical lens**

Discussion:

A two surface spherical lens is a solid object whose extent is determined by the intersection of the two surfaces rather than by a third surface (cylinder). The two surface spherical lens consists of sections of two spheres whose centers are located on the axis of the extent defining cylinder.

It is important to note that the default for the local coordinate system of a two surface spherical lens is at the vertex of the first surface. This may be overridden by selecting the option

sphlensvertex/on

described in the **Options** chapter.

A zero entry for r1 or r2 will result in the curvature, not the radius being set to zero, indicating a flat surface.

The single-valued expression for a spherical surface profile is:

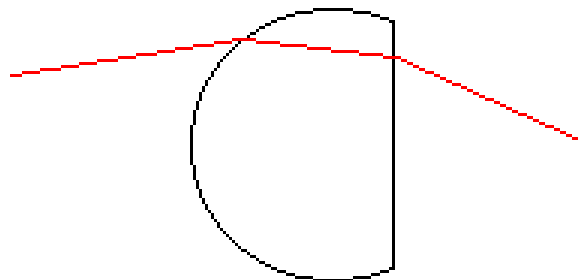
$$z = \frac{cv \cdot h^2}{1 + \sqrt{1 - cv^2 \times h^2}}$$

where (cv) is the curvature

$$cv = \frac{1}{radius}$$

and

$$h = \sqrt{x^2 + y^2}$$



Parameters:

User-defined object name, r1 (the first radius of curvature on surface my_lens.1), r2 (the second radius of curvature on surface my_lens.2), the vertex thickness (distance along the local z-axis), the lens diameter, and either the index of refraction or a material name. A lens material may be specified in the Object and Surface Properties box.

<u>Two Element Spherical Lens</u>	
Object name:	object name
r1	radius 1
r2	radius 2
thickness	thickness
diameter	diameter
n	index of refraction

Text Command:

```
define/lens/ts_spherical name r1 r2 thickness diameter n  
define/lens/ts_spherical/material=GLASS_NAME name r1 r2 thick diameter
```

Text Command Example:

```
define/lens/ts_spherical lens1 -10 10 3 5 1.5  
define/lens/ts_spherical/material=BK7 lens1 -10 10 3 5
```

Cylindrical Lens

Function:

Defines a single-element **cylindrical** lens

Discussion:

A single-element cylindrical lens is a solid object whose extent is determined by four planar surfaces (my_lens.3, my_lens.4, my_lens.5, my_lens.6) two of which are parallel to the x-z plane and two of which are parallel to the y-z plane.

The cylindrical lens consists of sections of two cylinders (either two cylinders with power in x or two cylinders with power in y, but not a combination of an x-powered and a y-powered cylinder). The axes of the cylinders must be parallel to each other and both parallel to either the x or y axes. Wedge, tilt, or rotation between the surfaces is not permitted.

The local coordinate system of a cylindrical singlet is at the vertex of the first surface. A zero entry for r1 or r2 will result in the curvature, not the radius being set to zero.

If an aspheric or toric modification will be applied to either surface 1 or surface 2 of a cylindrical lens, the initial surface must be defined as flat ($r1 = 0$ or $r2 = 0$).

The single-valued expression for a cylindrical surface profile with power in y is:

$$z = \frac{cv_y \cdot y^2}{1 + \sqrt{1 - (cv_y^2 \cdot y^2)}}$$

where cv_y is the curvature in y

$$cv_y = \frac{1}{radius_y} = \frac{1}{2 \cdot focallength_y}$$

and the single-valued expression for a cylindrical surface profile with power in x is:

$$z = \frac{cv_x \cdot x^2}{1 + \sqrt{1 - (cv_x^2 \cdot x^2)}}$$

where cv_x is the curvature in x.

$$cv_x = \frac{1}{radius_x} = \frac{1}{2 \cdot focallength_x}$$

Parameters:

User-defined object name, a power in x or y choice, r1 (the first radius of curvature on surface my_lens.1), r2 (the second radius of curvature on surface my_lens.2), the vertex thickness (distance along the local z-axis), the x half-size, the y half-size, and either the index of refraction or a material name (that is included in a previously loaded glass catalog). A lens material may be specified in the Object and Surface Properties box.

<u>Cylindrical Lens</u>	
Object name	object name
r1	radius 1
r2	radius 2
xhw	x half size
yhw	y half size
thickness	thickness
n	index of refraction

Text Commands:

```
define/lens/cylindrical/x_powered name r1 r2 xhw yhw thickness n
define/lens/cylindrical/y_powered name r1 r2 xhw yhw thickness n
define/lens/cyl/x/material=GLASS_NAME name r1 r2 xhw yhw thickness
define/lens/cyl/y/material=GLASS_NAME name r1 r2 xhw yhw thickness
```

Text Command Examples:

```
define/lens/cylindrical/x_powered cyll 10 0 5 5 5 1.6
define/lens/cylindrical/y_powered cyll 10 0 5 5 5 1.6
define/lens/cyl/x/material=BK7 cyll 10 0 5 5 5
define/lens/cyl/y/material=BK7 cyll 10 0 5 5 5
```

Catalog lenses

The OptiCAD subdirectory `catalogLenses` contains prescriptions for over 500 lenses from 15 manufacturers. Generally the file name corresponds to the manufacturer's catalog name/number. If you are using more than one identical lens, either import the first and then use the **replicate** feature to make copies or rename the first lens before entering the second lens.

Note that the lens prescriptions should be verified for performance and availability before using in a final system.

Aperture

Function:

Defines an **aperture**

Discussion:

Defines a limiting aperture, which will pass rays that are either outside of its extent (miss the object) or inside of its hole. Other rays will be absorbed. Note that the text command format uses options (modifiers) called `hoval` and `hrectangle` to indicate an elliptical- or a rectangular-shaped hole.

Parameters:

User-defined object name, x and y half-sizes of the outer extent, x and y location of the center of the hole (in surface local coordinates), x and y half-sizes of the hole, and shape choices for both the outer extent and the hole.

<u>Aperture</u>	
Object name	<code>object name</code>
xhw	<code>x outer half_size</code>
yhw	<code>y outer half size</code>
hxcen	<code>x relative location</code>
hycen	<code>y relative location</code>
hxhw	<code>x hole half_size</code>
hyhw	<code>y hole half_size</code>

Rectangular Extent

HoleRectang

Text Command:

```
define/aperture/rect/hrect name xhw yhw hxcen hycen hxhw hyhw
```

Text Command Example:

```
define/aperture/rect/hrect aper1 10 10 0 0 2 2
```

HoleOval

Text Command:

```
define/aperture/rectangle/hoval name xhw yhw hxcen hycen hxhw hyhw
```

Text Command Example:

```
define/aperture/rectangle/hoval aper1 10 10 0 0 2 2
```

Oval Extent

HoleOval

Text Command:

```
define/aperture/oval/hoval name xhw yhw hxcen hycen hxhw hyhw
```

Text Command Example:

```
define/aperture/oval/hoval aper1 10 10 0 0 2 2
```

HRectang

Text Command:

```
define/aperture/oval/hrect name xhw yhw hxcen hycen hxhw hyhw
```

Text Command Example:

```
define/aperture/oval/hrect aper1 10 10 0 0 2 2
```

Rod

Function:

Defines a **rod**

Discussion:

Defines a cylindrical rod (lens with flat ends), where the axis of the cylinder coincides with the object local z-axis. It is important to note that the local coordinate system of a rod is in the middle of the rod.

This default may be overridden by selecting the option

`set/sphlensvertex/on`

described in the Options chapter.

A material name is case sensitive. Glass names must be entered in upper case. The glass catalog containing a selected glass must have been loaded (see `set/matcatalog` in the **Options** chapter) prior to any use of the glass in an object definition.

Parameters:

Full length, diameter, index of refraction, and user defined object name.

Rod	
Object name	<code>object name</code>
length	<code>z full size</code>
diameter	<code>diameter</code>
index of refraction	<code>index of refraction</code>

Text Commands:

```
define/rod name length diameter n  
define/rod/material=glass_name name length diameter
```

Text Command Examples:

```
define/rod rod2 10 1 1.5  
define/rod/material=bk7 my_rod 10 1
```

Cavity

Function:

The object to be defined is a reflective object, such as a full ellipsoid, full cylinder, cone, or eltrough.

Discussion:

A cavity is a closed shell object (surface of no thickness). See Physical Extent of Objects, in this chapter, for a description of closed shell objects. A cavity is initially 100% reflective, but its reflectance, absorptance, transmittance, and diffuse scattering can be modified.

See sections: Physical Extent of Objects and Shell Surface Reflectivity Options at the start of the chapter.

Ellipsoid

Function:

Defines a complete **ellipsoid**

Discussion:

Defines a full (closed surface) reflective ellipsoid.

The equation for a full ellipsoid is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

where a, b, and c are the half-sizes of the ellipsoid along the x, y, and z axes respectively.

For the condition a = b, some useful formulas are

$$a = b = \frac{1}{cv \cdot \sqrt{k+1}}$$

are the (equal) minor axes of the ellipsoid,

$$c = \frac{1}{cv \cdot (k+1)}$$

is the major axis of the ellipsoid, k is the conic constant and (cv) is the vertex curvature of the conic surface previously defined,

$$cv = \frac{c}{a^2}$$

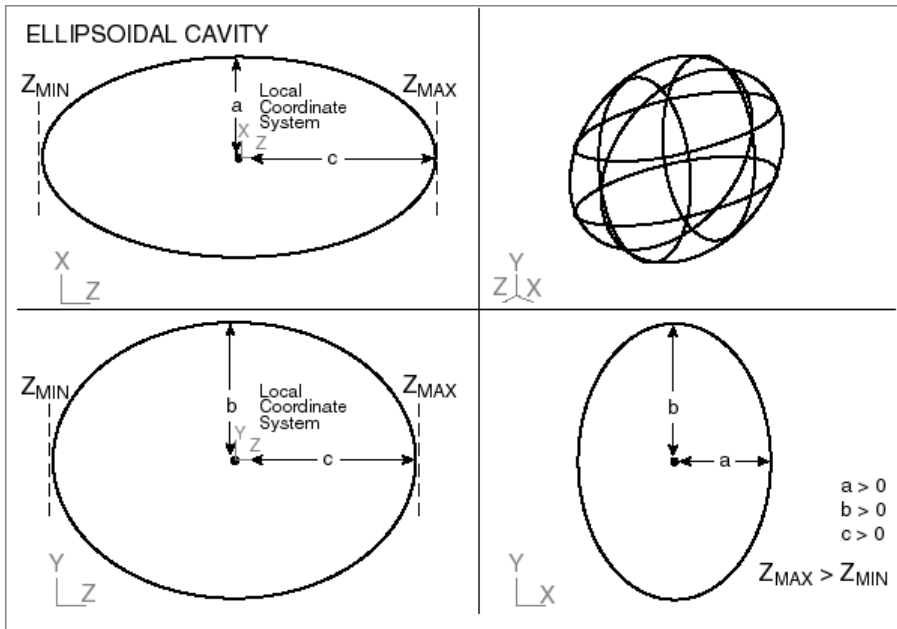
and the foci of the ellipsoid are

$$z_1 = -\sqrt{c^2 - a^2}$$

and

$$z_2 = \sqrt{c^2 - a^2}$$

The full ellipsoid is centered around $x = y = z = 0$, while the single-valued conic surface is tangent to the x-y plane and has its vertex at $x = y = z = 0$. The value entered for z_{min} must always be less than the value entered for z_{max} .



Parameters:

User-defined object name, x, y, and z half-sizes; extents along the z-axis, and inside/outside reflectivity options.

Ellipsoidal Cavity	
Object name	object name
a	x half size
b	y half size
c	z half size
zmin	minimum z extent
zmax	maximum z extent
Both	⊗
Inside	⊗
Outside	⊗

Text Commands:

```
define/cavity/ellipsoid/inside name a b c zmin zmax
define/cavity/ellipsoid/outside name a b c zmin zmax
define/cavity/ellipsoid/both name a b c zmin zmax
```

Text Command Examples:

```
define/cavity/ellipsoid/inside ell1 1 1 2 -2 2
define/cavity/ellipsoid/outside ell1 1 1 2 -2 2
define/cavity/ellipsoid/both ell1 1 1 2 -2 2
```

Cone

Function:

Defines an elliptical **cone**

Discussion:

Defines a full (both sections) reflective elliptical cone. The origin of the local coordinate system is at the common central vertex.

The equation for an elliptical cone is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = \frac{z^2}{c^2}$$

For $a = b$ the expression reduces to

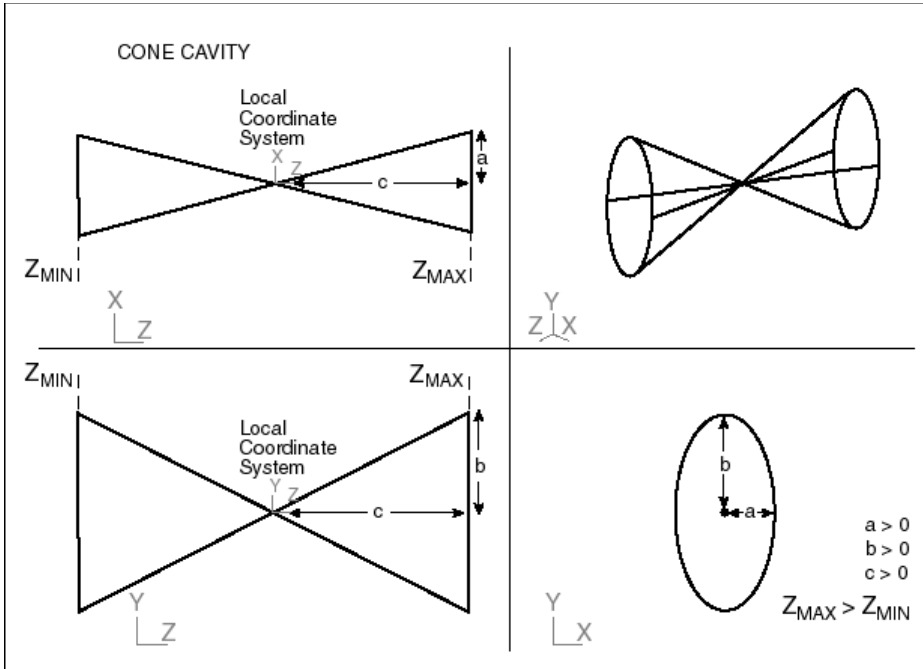
$$z = \frac{c \cdot h}{a}$$

where

$$h^2 = x^2 + y^2$$

and a/c is the tangent of the half angle of the cone.

For $a \neq b$, a cross section parallel to the x-y plane is elliptical rather than circular, a/c is the tangent of the half angle in the x-z plane (dx/dz), b/c is the tangent of the half angle in the y-z plane (dy/dz). The value entered for z_{min} must always be less than the value entered for z_{max} , and a , b , and c must be positive.



Parameters:

User-defined object name; x, y, and z half-sizes; extents along the z axis; and inside/outside reflectivity options.

Cone Cavity	
Object name	object name
a	x half size at z = c
b	y half size at z = c
c	z half size
zmin	minimum z extent
zmax	maximum z extent
Both	⊗
Inside	⊗
Outside	⊗

Text Commands:

```
define/cavity/cone/inside name a b c zmin zmax
define/cavity/cone/outside name a b c zmin zmax
define/cavity/cone/both name a b c zmin zmax
```

Text Command Examples:

```
define/cavity/cone/inside cone3 1 1 2 0 2  
define/cavity/cone/outside cone3 1 1 2 0 2  
define/cavity/cone/both cone3 1 1 2 0 2
```

Cylinder

Function:

Defines a full elliptical **cylinder**

Discussion:

Defines a full reflective elliptical cylinder, open at both ends.

The expression for an elliptical cylinder is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

where a and b are the half-sizes of the elliptical cylinder along the x- and y-axes.

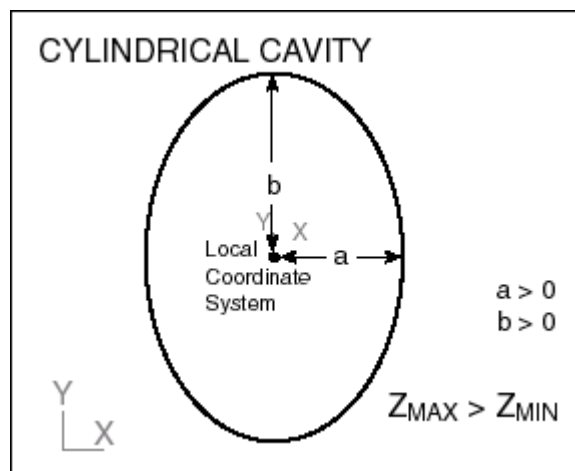
For $b > a$, the line foci (2-D foci of a section) of the ellipse are

$$y_1 = -\sqrt{b^2 - a^2}$$

and

$$y_2 = \sqrt{b^2 - a^2}$$

The full elliptical cylinder is centered around $x = y = z = 0$. The value entered for z_{min} must always be less than the value entered for z_{max} .



Parameters:

User-defined object name, x and y half-sizes, extents along the z-axis, and inside/outside reflectivity options.

Cylindrical Cavity	
Object name	<code>object name</code>
xhw	<code>x half size</code>
yhw	<code>y half size</code>
zmin	<code>minimum z extent</code>
zmax	<code>maximum z extent</code>
Both	⊗
Inside	⊗
Outside	⊗

Text Commands:

```
define/cavity/cylinder/inside name xhw yhw zmin zmax
define/cavity/cylinder/outside name xhw yhw zmin zmax
define/cavity/cylinder/both name xhw yhw zmin zmax
```

Text Command Examples:

```
define/cavity/cylinder/inside cyl1 1 1 -3 3
define/cavity/cylinder/outside cyl1 1 1 -3 3
define/cavity/cylinder/both cyl1 1 1 -3 3
```

Eltrough

Function:

Defines an **elliptical trough**

Discussion:

Defines a reflective elliptical trough. This is an open object, more like a mirror type than a cavity. The elliptical trough is a special (an open surface that is a subset) case of the elliptical cylinder. The elliptical trough differs from other cavity objects in that the surface is tangent to the x-y plane. A parabolic trough is described in this section under Mirror.

The expression for an elliptical trough is:

$$\frac{y^2}{a^2} + \frac{(z-b)^2}{b^2} = 1$$

where a and b are the half-sizes of the elliptical trough along the y- and z-axes.

The vertex curvature (2-D rather than 3-D)

$$cv_y = \frac{b}{a^2}$$

where a and b must be positive.

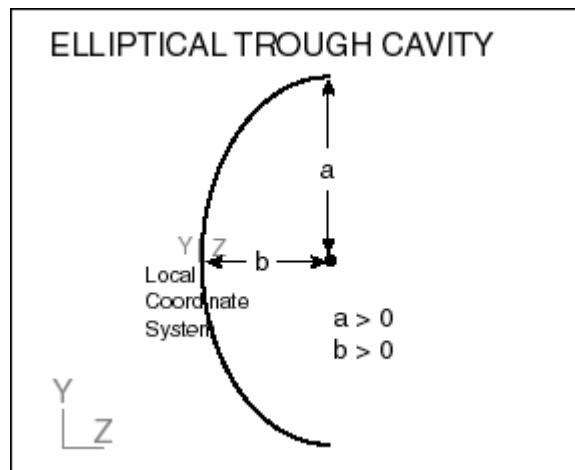
For $b > a$, the line foci (2-D foci of a section) of the ellipse are

$$z_1 = b - \sqrt{b^2 - a^2}$$

and

$$z_2 = b + \sqrt{b^2 - a^2}$$

The full elliptical cylinder is centered around $x = y = z = 0$. The value entered for z_{min} must always be less than the value entered for z_{max} .



Parameters:

User-defined object name, x and y half-sizes, extents along the z-axis, and inside/outside reflectivity options.

<u>Elliptical Trough Cavity</u>	
Object name	object name
a	y half size
b	z half size
width	y full size
length	x full size
Both	⊗
Front	⊗
Back	⊗

Text Commands:

```
define/cavity/eltrough/inside name a b width length
define/cavity/eltrough/outside name a b width length
define/cavity/eltrough/both name a b width length
```

Text Command Examples:

```
define/cavity/eltrough/inside elt1 1 1 2 10
define/cavity/eltrough/outside elt1 1 1 2 10
define/cavity/eltrough/both elt1 1 1 2 10
```

Prism

General Discussion:

The prism differs from other OptiCAD objects in that it does not directly create objects but instead uses the existing Polynet feature to define prisms. Prism is a higher-level feature than Polynet, and it saves considerable time when compared to manually defining 3-D prisms. The input parameters are generally used to scale a basic prism.

The Prism feature does not have a text command version; the polynet object formed by the Prism feature can be saved as a polygon object (not scaleable).

CornerCube

Function:

Defines a **Corner Cube**

Discussion:

Defines a solid corner cube with uncoated surfaces (TIR is handled automatically). A hollow corner cube can be made by entering an index of refraction of 1.0 and using a text editor or the modify/polygon command to change the reflectivity of the last three polygons to 1.0.

Parameters:

User-defined object name, half width, and index of refraction.

Corner Cube	
Name	<code>object name</code>
Half-width	<code>half size</code>
Refractive Index	<code>index of refraction</code>

Polar Radiometer

Function:

Defines a **1-D Faceted Radiometer**.

Discussion:

Defines and places at the origin a 1-D belt of polygons with a radiometer attached. The band of polygons (called a belt) approximates a cylinder and is positioned in either the x-y, x-z, or y-z planes, with the axis of the symmetry about either the z, y, or x axis. This allows the creation of polar plots for a detector located at a user selected radial distance from a light source. The object name for the polar radiometer is either XYPolarRad, XZPolarRad, or YZPolarRad depending on the option selected. The belt of facets is initially defined as transparent. The resulting polar plot is displayed (after selecting a polar radiometer from the **OOB**) using the View → Type → Polar Radiometer function or by clicking the `view Polar Radiometer` toolbar button.

Note that a polar radiometer is created with a default name. You can define more than one polar radiometer about the same axis by defining and then renaming polar radiometers.

For the polar radiometer defined from a macro n_x must be a multiple of 4, and n_x times n_y must be equal to n_{cells} . As an example for $n_{\text{cells}} = 360$, $n_x=4$ and $n_y=90$ would work.

Parameters:

Orientation of the polar radiometer (XY, XZ, or YZ), the radius of the belt, the half height of the belt in the orthogonal direction (expressed as a fraction of the radius), and the number of facets around the cylinder.

PolarRadiometer	
Radius of the belt	<code>radius of the cylinder</code>
Belt height	<code>cell fractional half height</code>
Number of cells	<code>number of facets</code>

Text Command Sequence (for an xy polar radiometer):

```
polynet/belt/xypolar xypolarrad belt_radius belt_height n_cells
place xypolarrad x y z
radiometer/attach xypolarrad nx ny
```

Text Command Sequence Example (for an xy polar radiometer):

```
polynet/belt/xypolar xypolarrad 100 0.01 36
place xypolarrad 0 0 0
radiometer/attach xypolarrad 4 9
```

Replicate

Function:

Defines **copies** of the **selected object**.

Discussion:

Makes copies of an existing object. The user selects the object to be copied (cloned) from the OptiCAD Objects box. The user can either enter a new base name for the family of clones or use the name of the selected object as the base name. The new object names will be variations of the base object name.

Parameters:

The object to be cloned is selected from the OptiCAD Objects box, this selected object will appear as the default base object name (which can be replaced by another name). The number of copies to be made.

<u>Replicate</u>	
Base object name	base object name
Number of copies	number of clones to be made

Text Command:

```
replicate/newname=new_base_object_name object_name ncopies
```

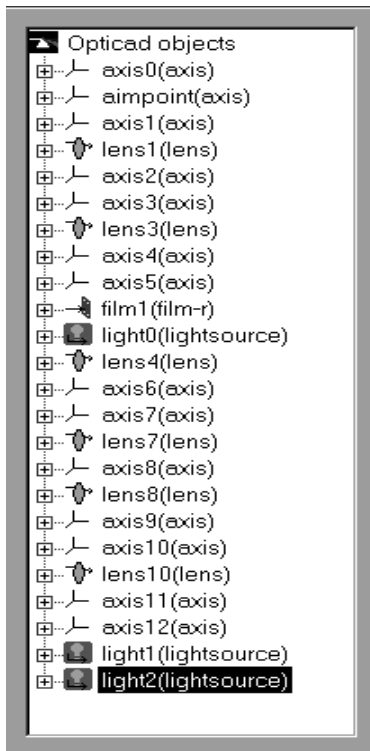
End of Design Chapter

OBJECT and SURFACE PROPERTIES

There is considerable overlap between the **Object and Surface Properties** chapter and **Design** chapter. The user should consult both chapters for a more complete description of the definition and modification of surface and material properties.

As with all OptiCAD functions that require an object or surface name, the user may select an object and surface from the **OptiCAD Objects box (OOb)**. This name will then appear in places that require an object or surface name. A surface name is simply an object name with a period and a surface number appended (`object_name.surface_number`). If only an object name is required, then any surface within that object may be selected.

OptiCAD Objects Box (OOb)



The **OptiCAD Objects box** (shown above) replaces box of the same name used in earlier versions of OptiCAD.

The **OptiCAD Objects box (OOb)** displays a list of all objects. An object is selected by single-clicking on its name. The selected object name appears in all input boxes requiring an object name.

Double-clicking on an object name will do two things: it will bring up the **Object and Surface Properties box (OSPb)**, which allows modification of most object parameters, and it will open up the object name in order to display all of the surfaces that are part of that object.

A surface may be selected by single-clicking on the surface name. The selected surface name appears in all input boxes requiring a surface name, and the **Object and Surface Properties box (OSPb)** then displays information pertaining to that surface.

Object Renaming

Function:

Renames an Object

Discussion:

An object may be renamed by clicking on the current name twice (slowly) in the **Opti-CAD Objects box (OOB)** and then replacing the entire line (including the name and object type in parenthesis) with only the new name. Do not type in the object type (in parenthesis).

Text Commands:

```
rename/newname=new_object_name old_object_name
```

Object and Surface Properties Box (OSPb)

Double-clicking on an object name in the **OOB** will bring up the **OSPb**. The **OSPb** allows modification of basic object shape, global object placement (and replacement), surface reflectivity and absorptance, surface scattering, and materials (glass) selection.

Object definitions are accomplished using the **OSPb**, the Design menu opens the **OSPb** with the default values displayed.

Modification of polynet objects is limited to glass selection, placement, and surface scattering (of all facets).

This chapter is an overview of the **OSPb**. Detailed descriptions of specific features are found in the corresponding chapters of this manual.

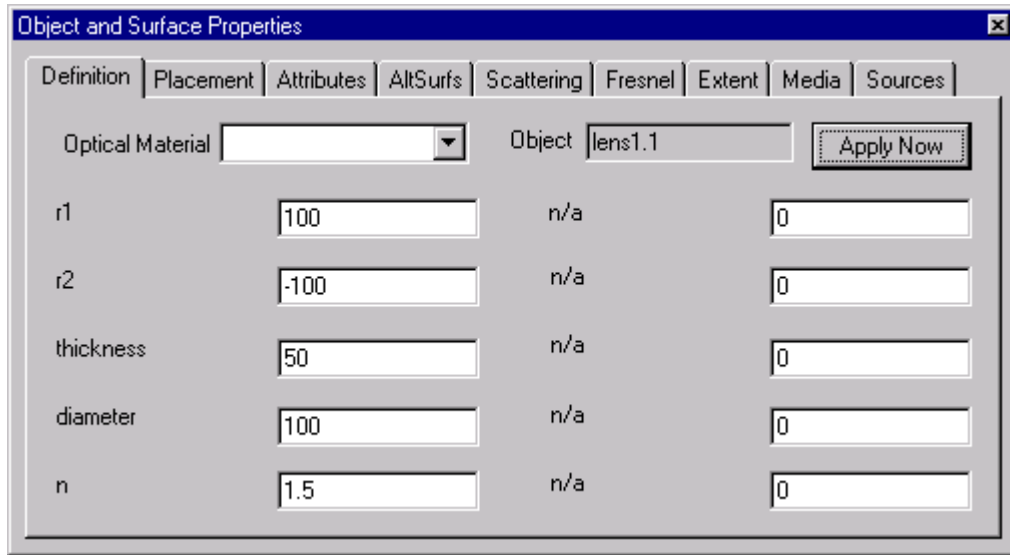
The pages described below may be selected by clicking on their names.

While viewing any page, you may select a different object or surface by clicking on the object name, or you may browse through the object list using the up and down keyboard arrows.

The Apply Now and the Place and Activate buttons only implement the values for the current page. Pushing the enter key will not implement either the Apply Now or Place and Activate operations. If you change pages without first clicking Apply Now or Place and Activate, the changes that you made will not be implemented.

Aborting an object change may be accomplished by closing the box (by clicking the x in upper right of the box), or moving to a new page and clicking the Apply Now button.

Definition Page



This page is specific to the object or surface that was selected using the **OOB** described in a previous section.

For a description of numerical entries defining to a specific object see the **Design chapter**.

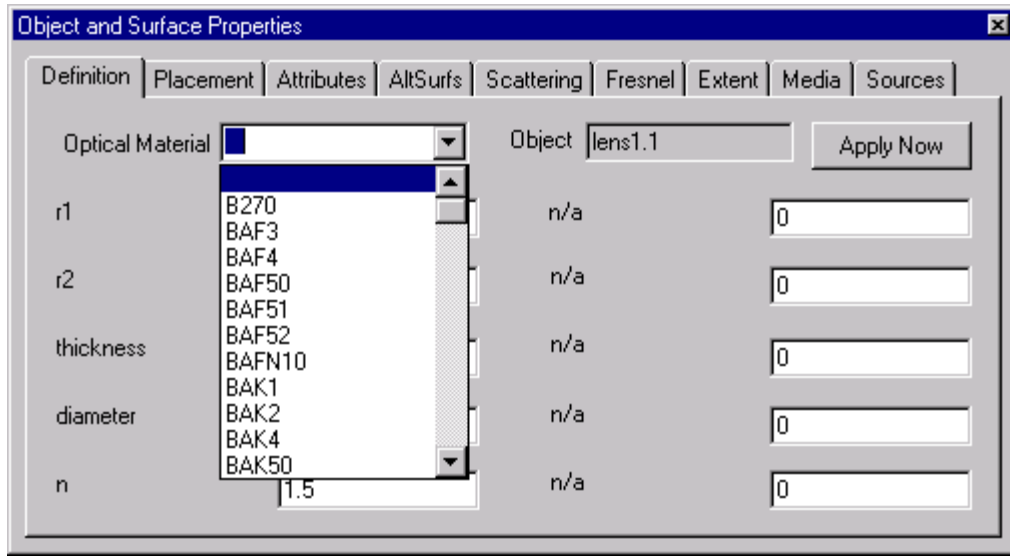
While numerical parameters can be changed using this page, the user cannot change the object type. Here are examples of changes that are not permitted: a spherical lens cannot be changed to a cylindrical lens, an elliptical cavity cannot be changed to a cone, and a cone cannot be changed to a CPC.

Note that for surfaces that have been modified using the aspheric feature, which is used by the Z_TO_OP translator for surface definitions, the **Definition page** does not display the modified radius ($1/cv$ used in the aspheric formula), but instead displays zero.

The index of refraction (n) displayed by the **Definition page** is for wavelength 1.

The value for radii or focal length displayed on this page does not reflect any surface modifications made using the Alternate Surface features. If you have assigned an aspheric or a toric to the surface, the values displayed are not accurate.

Optical Materials List



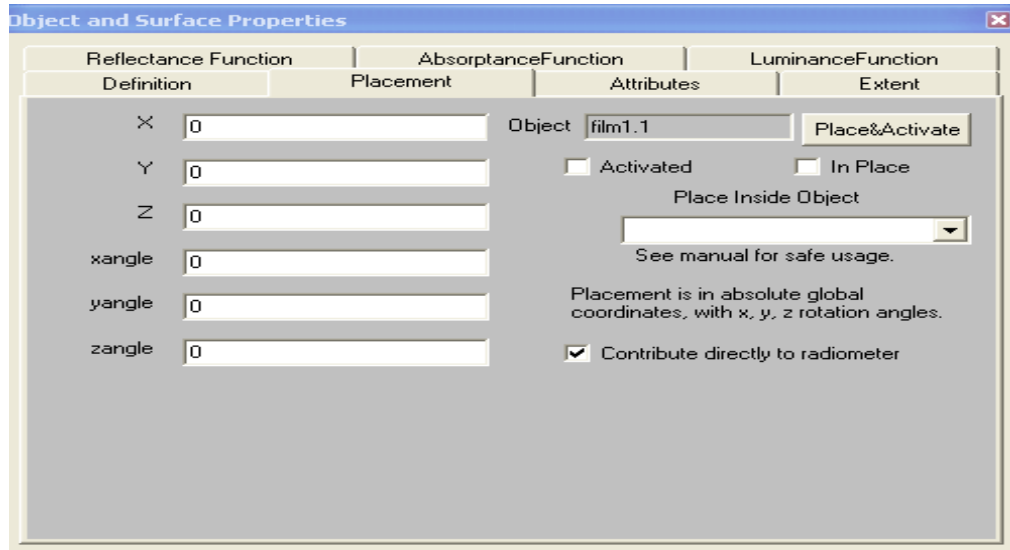
The **Optical Material Menu** is activated by clicking on the down arrow to the right of the **Optical Materials input box** on the **Definition page**.

The list of optical materials displayed are the glass catalogs (agf files) that have been previously loaded using the glass table loading command (`set/matcatalog/name=` command described in the **Options chapter**). A material is selected by clicking on its name.

Gradient index materials are selected on the **Media page**.

In order to enter an explicit (wavelength independent) index of refraction, select the blank section at the top of the list, and then enter the index in the box labeled n.

Placement Page



This page is specific to the object that was selected using the **OOB** described in a previous section.

The **Placement** page performs a global placement (or replacement) of the selected object. The values for x, y, and z, as well as the x, y, and z rotation angles must all be specified in global coordinates. OptiCAD also allows relative (and inside) placements as well as local and relative rotations.

The **Contribute directly to radiometer** selection box determines if rays leaving the object will contribute directly to a radiometer. See the description of a radiometer in the **Radiometer** section of the **Analyze** chapter.

This page is an excellent tool for placing and relocating an object that is to be located unrotated in free space, with its location specified in global coordinates.

Objects that are to be placed relative to (or inside of) other objects are best placed using the Edit → Place function, where location is specified in the local coordinates of the reference (or host) object. See the **PlaceRelative** and **PlaceInside** sections of the **Edit** chapter for a more complete description of these functions.

Objects that are to be rotated locally (about their own local coordinate system) and objects that are to be rotated about the local coordinate system of another object are best rotated using the Edit → Rotate function. See the **Edit** chapter for details.

The **Activated** and **In Place** check boxes activate and deactivate an object, and place or remove an object. These operations are performed immediately. See the **Edit** chapter for descriptions of these functions.

The **Place Inside Object** option uses global coordinates to place the object, not the coordinates of the host object.

Clicking the **Place** and **Activate** button places (or replaces) the object, and therefore automatically checks both the **Activated** and the **InPlace** buttons. If you intend to deactivate or remove an object, you should not click the **Place** and **Activate** button, as doing so will override the unchecking of both the **Activated** and the **InPlace** buttons.

Attributes Page

The screenshot shows the 'Object and Surface Properties' dialog box with the 'Attributes' tab selected. The 'Indices/wavelength' section contains a 2x6 grid of input fields for N1 through N12, all containing the value '0'. To the right, a text box contains 'surface.1' and an 'Apply Now' button. Below this, a 'Selected Coating' dropdown menu is set to '<NONE>'. The 'Surface Attributes' section contains three input fields: 'refl' set to '1', 'absor' set to '0', and 'lumins' set to '0'.

This page is specific to the object or surface that was selected using the **OOB** described in a previous section.

The **Attributes page** can be used to modify the reflectivity and absorptance of the selected surface, and to assign a coating (previously defined in the `coating.dat` data file) to a specific surface. It also lists the index of refraction of the material at the active wavelengths. Default values are listed for inactive wavelengths.

Surface Attributes

Function:

Changes surface **reflectance**, **absorptance**, and **irradiance**

Discussion:

Changes surface reflectance and absorptance; surface transmittance is automatically set to $1.0 - (\text{reflectance} + \text{absorptance})$.

In order to have a ray split at a partial reflector, the maximum ghost order must be set to at least 1. See the **Ray Trace Limits section of the Options chapter**.

For the reflectance and absorptance specified on this page to be in effect, the Surface-Fresnel option specified on the **Fresnel page** must first be set to None.

Note that entries on the **Fresnel page** take precedence over **Surface Attributes** which take precedence over **Options** -> **Fresnel Reflect**.

The surface irradiance is used by rays launched by the reverse ray-trace detector. Surface irradiance is entered in units of watts/cm^2 if power units of watts are selected or in lumens/m^2 if power units of lumens are selected. The surface is assumed to be emitting into a Lambertian distribution.

Note that the `power_units` are set using the **Windows page**.

The reverse rays act as travelling power buckets, accumulating power from surfaces having a non-zero irradiance. See the **Detector** section of the **Design chapter** and the **Camera page** section of the **Object Properties** chapter.

Parameters:

Previously defined surface name (object_name.surface_number), new surface reflectance, new surface absorptance.

Text Command:

```
modify/surface/attributes surface_name refl absor lumins
```

Text Command Example:

```
modify/surface/attributes lens1.2 0.5 0.5 1.0
```

Note on Polynet Surface Properties

The reflectance and absorptance surface properties of all polygons of a polynet object can be simultaneously modified on the **Attributes** page.

The reflectance and absorptance entered on the **Attributes** page overrides the individual polygon values that may have been entered on the **polygon refl= absor=** command line of a polynet, for the current session only. The reflectance and absorptance modification command for a polynet object is not saved in either a mac or a opt file, instead it is written to the saved file commented out.

Selected Coating

Function:

Assigns a coating to surface

Discussion:

A coating that has been previously defined in the coating.dat file can be selected from the selected coating list to be assigned to a surface.

Parameters:

A coating name.

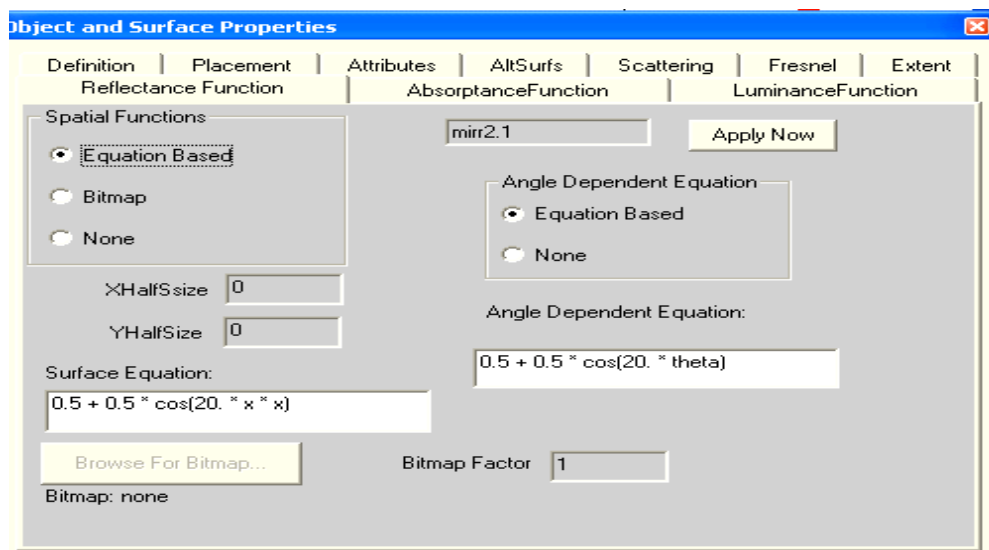
Text Command:

```
coating/name=COATING_NAME surface_name
```

Text Command Example:

```
coating/name=AR lens1.1
```

Reflectance Functions Page



This page is specific to the object.surface that was selected using the **OOB** described in a previous section.

Reflectance Functions

Function:

Defines a **surface reflectance function**

Discussion:

A surface reflectance function consists of a spatial and an angular component. The spatial component can be either a user entered equation based on the x-y intersection of each ray on the surface, or a user specified bitmap (bmp) file. The angular component can be a user entered equation based on the angle of incidence between each ray and the local surface normal. A single spatial and a single angular function may be applied to most ray-traceable surfaces.

An equation or bitmap resulting in a reflectance of less than 0 is set to 0, while a reflectance greater than 1 is set to 1. If reflectance plus absorptance is greater than 1, they are both set to the values defined on the **Attributes** page.

The selected bitmap can be either 8 bit or 24 bit. For a 24 bit bitmap, the red, green, and blue values are averaged. Bitmap values are normalized by 256, resulting in a value between 0 and 1. A table showing the permitted functions in a user entered equation is shown below.

multiply	*
add	+
subtract	-
power	^
square root	sqrt()
cosine	cos()
sine	sin()
tangent	tan()
arccosine	acos()
arcsine	asine()
arctangent	atan()
log base e	log()
log base 10	log10()
exponential base e	exp()
hyperbolic cosine	cosh()
hyperbolic sine	sinh()
hyperbolic tangent	tanh()
secant	sec()
cosecant	csc()
cotangent	cot()
hyberbolic cotangent	acoth()
hyperbolic cosecant	acsch()
archyperbolic cosine	acosh()
archyperbolic tangent	atanh()

The spatial (equation or bitmap) reflectance value is multiplied by the angular equation reflectance, resulting in the total position and angle of incidence reflectance. A non-entry for the spatial or angular function is interpreted as a value of 1.

The value for reflectance entered on the **Attributes** page is not used in this calculation.

Equation Based Spatial Reflectance Function

The user enters a permitted function of the variables: x, y, and r where.

$$r = \sqrt{x^2 + y^2}$$

Text Commands:

```
reflectancefunction/spatialequation/name="spatial_equation"
surface_name
reflectancefunction/remove_spatial surface_name
```

Text Command Examples:

```
reflectancefunction/spatialequation/name="0.5 + 0.5 * cos(20. * x * x)"
mirr2.1
reflectancefunction/remove_spatial mirr2.1
```

Bitmap Based Spatial Reflectance Function

The user selects a bitmap file and enters the x and y sizes of the bitmap and a bitmap factor which is multiplied by the normalized bitmap values.

Text Commands:

```
reflectancefunction/attach/file="bitmapfilename.bmp" surface_name
halfsize_x halfsize_y bitmapfactor
reflectancefunction/remove_spatial surface_name
```

Text Command Examples:

```
reflectancefunction/attach/file="OpticadLogo.bmp" mirr2.1 0.99 0.99 1
reflectancefunction/remove_spatial mirr2.1
```

Equation Based Angular Reflectance Function

The user enters a permitted function of the variables: xtheta, ytheta, and theta where

$$\theta = \hat{v}_{raydirection} \bullet \hat{v}_{localsurfacenormal}$$

$$x\theta = \text{atan} \frac{L_{inlocalsurfacenormal}}{N_{inlocalsurfacenormal}}$$

$$y\theta = \text{atan} \frac{M_{inlocalsurfacenormal}}{N_{inlocalsurfacenormal}}$$

Theta is the dot product of the ray direction and the local (instantaneous) surface normal at the point of intersection. Xtheta is the ratio of the x and z ray direction cosines in the coordinate system of the local (instantaneous) surface normal at the point of intersection. Ytheta is the ratio of the y and z ray direction cosines in the coordinate system of the local (instantaneous) surface normal at the point of intersection.

Note that the angles are in radians.

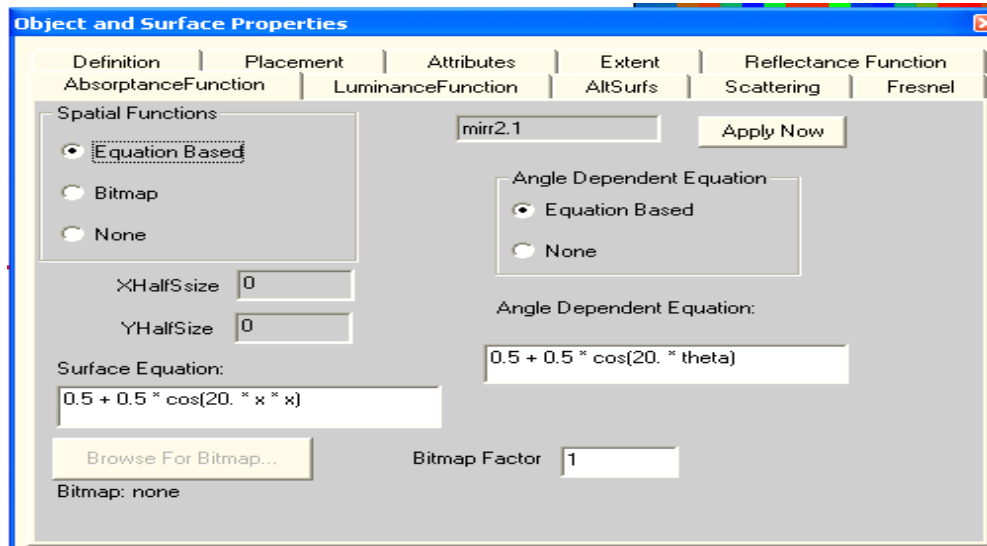
Text Commands:

```
reflectancefunction/angularequation/name="angular_equation" seed
surface_name
reflectancefunction/remove_angular surface_name
```

Text Command Examples:

```
reflectancefunction/angularequation/name="0.5 + 0.5 * cos(20. * theta)"
mirr2.1 0
reflectancefunction/remove_angular mirr2.1
```

Absorptance Functions Page



This page is specific to the object.surface that was selected using the **OOB** described in a previous section.

Absorptance Functions

Function:

Defines a **surface absorptance function**

Discussion:

A surface absorptance function consists of a spatial and an angular component. The spatial component can be either a user entered equation based on the x-y intersection of each ray on the surface, or a user specified bitmap (bmp) file. The angular part consists of a user entered equation based on the angle of incidence between each ray and the local surface normal. A single spatial and a single angular function may be applied to most ray-traceable surfaces.

The spatial and angular absorptance functions are user defined. An equation or bitmap resulting in an absorptance of less than 0 is set to 0, while an absorptance greater than 1 is set to 1. If reflectance plus absorptance is greater than 1, they are both set to the values defined on the **Attributes** page.

The selected bitmap can be either 8 bit or 24 bit. For a 24 bit bitmap, the red, green, and blue values are averaged. Bitmap values are normalized by 256, resulting in a value between 0 and 1. A table showing the permitted functions in a user entered equation is shown in the **Reflectance Functions Page** section in this chapter.

The spatial (equation or bitmap) absorptance value is multiplied by the angular equation absorptance, resulting in the total position and angle of incidence absorptance. A non-entry for the spatial or angular function is interpreted as a value of 1.

The value for absorptance entered on the **Attributes** page is not used in this calculation.

Equation Based Spatial Absorptance Function

The user enters a permitted function of the variables: x, y, and r where

$$r = \sqrt{x^2 + y^2}$$

Text Commands:

```
absorptancefunction/spatialequation/name="spatial_equation"  
surface_name  
absorptancefunction/remove_spatial surface_name
```

Text Command Examples:

```
absorptancefunction/spatialequation/name="0.5 + 0.5 * cos(20. * x * x)"  
mirr2.1  
absorptancefunction/remove_spatial mirr2.1
```

Bitmap Based Spatial Absorptance Function

The user selects a bitmap file and enters the x and y sizes of the bitmap and a bitmap factor which is multiplied by the normalized bitmap values. The user also selects a random number seed value. Seed value of 0 (recommended) seeds to the system clock.

Text Commands:

```
absorptancefunction/attach/file="bitmapfilename.bmp" surface_name  
halfsize_x halfsize_y  
absorptancefunction/remove_spatial surface_name
```

Text Command Examples:

```
absorptancefunction/attach/file="OpticadLogo.bmp" mirr2.1 0.99 0.99  
absorptancefunction/remove_spatial mirr2.1
```

Equation Based Angular Absorptance Function

The user enters a permitted function of the variables: xtheta, ytheta, and theta where

$$\theta = \hat{v}_{raydirection} \bullet \hat{v}_{localsurfacenormal}$$

$$xtheta = \text{atan} \frac{L_{inlocalsurfacenormal}}{N_{inlocalsurfacenormal}}$$

$$ytheta = \text{atan} \frac{M_{inlocalsurfacenormal}}{N_{inlocalsurfacenormal}}$$

Theta is the dot product of the ray direction and the local (instantaneous) surface normal at the point of intersection. Xtheta is the ratio of the x and z ray direction cosines in the coordinate system of the local (instantaneous) surface normal at the point of intersection. Ytheta is the ratio of the y and z ray direction cosines in the coordinate system of the local (instantaneous) surface normal at the point of intersection.

Note that the angles are in radians.

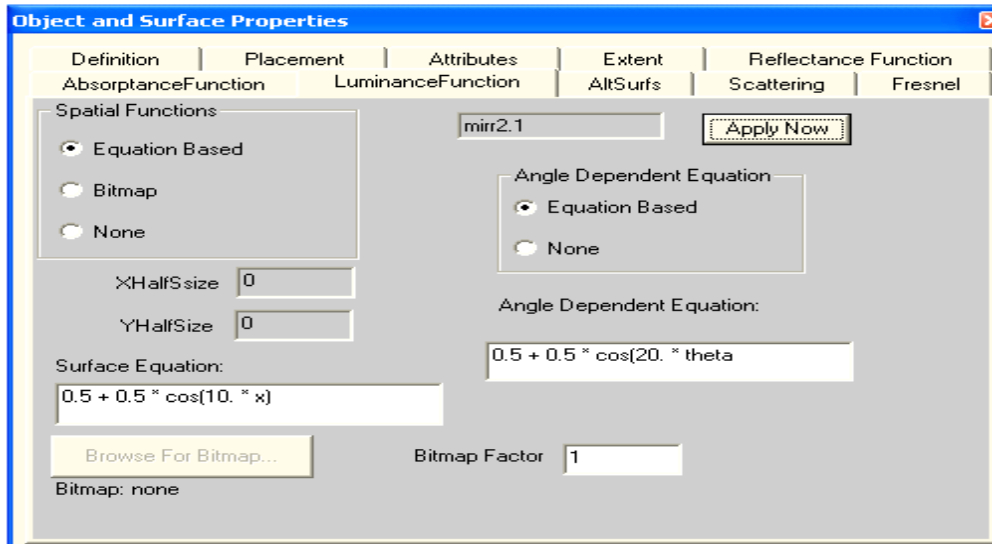
Text Commands:

```
absorptancefunction/angularequation/name="angular_equation"  
surface_name  
absorptancefunction/remove_angular surface_name
```

Text Command Examples:

```
absorptancefunction/angularequation/name="0.5 + 0.5 * cos(20. * theta)"  
mirr2.1  
absorptancefunction/remove_angular mirr2.1
```

Luminance Functions Page



This page is specific to the object.surface that was selected using the **OOB** described in a previous section.

Luminance Functions

Function:

Defines a **surface luminance function**

Discussion:

A surface luminance function consists of a spatial and an angular component. The spatial component can be either a user entered equation based on the x-y intersection of each ray on the surface, or a user specified bitmap (bmp) file. The angular part consists of a user entered equation based on the angle of incidence between each ray and the local surface normal. A single spatial and a single angular function may be applied to most ray-traceable surfaces.

The spatial and angular luminance functions are user defined. An equation or bitmap resulting in a luminance of less than 0 is set to 0.

The selected bitmap can be either 8 bit or 24 bit. For a 24 bit bitmap, the red, green, and blue values are averaged. Bitmap values are normalized by 256, resulting in a value between 0 and 1. A table showing the permitted functions in a user entered equation is shown in the **Reflectance Functions Page** section in this chapter.

The spatial (equation or bitmap) luminance value is multiplied by the angular equation luminance, resulting in the total position and angle of incidence luminance. A non-entry for the spatial or angular function is interpreted as a value of 1.

The value for luminance entered on the **Attributes** page is not used in this calculation.

Equation Based Spatial Luminance Function

The user enters a permitted function of the variables: x, y, and r where

$$r = \sqrt{x^2 + y^2}$$

Text Commands:

```
luminancefunction/spatialequation/name="spatial_equation"  
surface_name  
luminancefunction/remove_spatial surface_name
```

Text Command Examples:

```
luminancefunction/spatialequation/name="0.5 + 0.5 * cos(20. * x * x)"  
mirr2.1  
luminancefunction/remove_spatial mirr2.1
```

Bitmap Based Spatial Luminance Function

The user selects a bitmap file and enters the x and y sizes of the bitmap and a bitmap factor which is multiplied by the normalized bitmap values.

Text Commands:

```
luminancefunction/attach/file="bitmapfilename.bmp" surface_name  
halfsize_x halfsize_y  
luminancefunction/remove_spatial surface_name
```

Text Command Examples:

```
luminancefunction/attach/file="OpticadLogo.bmp" mirr2.1 0.99 0.99  
luminancefunction/remove_spatial mirr2.1
```

Equation Based Angular Luminance Function

The user enters a permitted function of the variables: xtheta, ytheta, and theta where

$$\theta = \hat{v}_{raydirection} \cdot \hat{v}_{localsurfacenormal}$$

$$x\theta = \text{atan} \frac{L_{inlocalsurfacenormal}}{N_{inlocalsurfacenormal}}$$

$$y\theta = \text{atan} \frac{M_{inlocalsurfacenormal}}{N_{inlocalsurfacenormal}}$$

Theta is the dot product of the ray direction and the local (instantaneous) surface normal at the point of intersection. Xtheta is the ratio of the x and z ray direction cosines in the coordinate system of the local (instantaneous) surface normal at the point of intersection. Ytheta is the ratio of the y and z ray direction cosines in the coordinate system of the local (instantaneous) surface normal at the point of intersection.

Note that the angles are in radians.

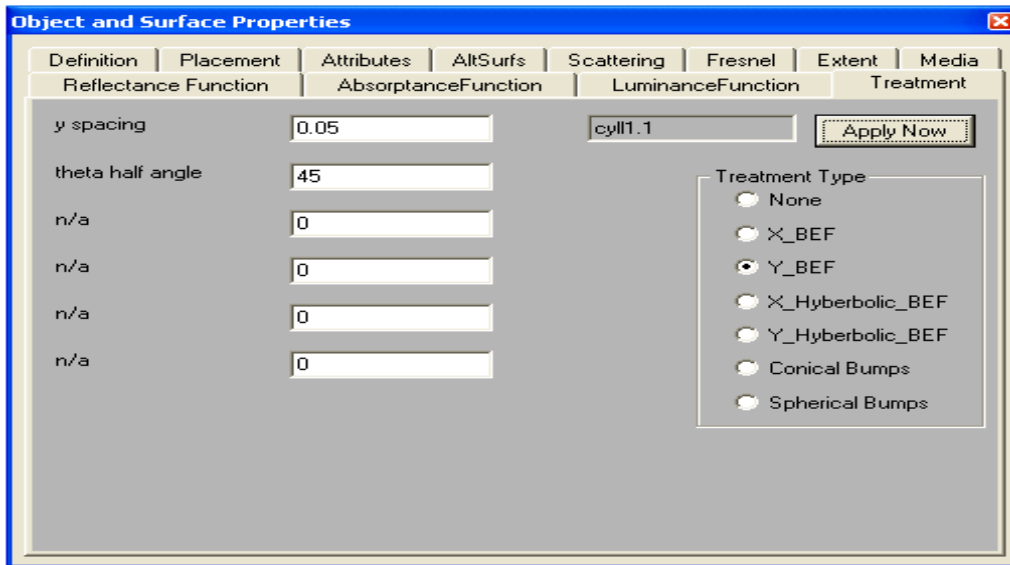
Text Commands:

```
luminancefunction/angularequation/name="angular_equation"
surface_name
luminancefunction/remove_angular surface_name
```

Text Command Examples:

```
luminancefunction/angularequation/name="0.5 + 0.5 * cos(20. * theta)"
mirr2.1
luminancefunction/remove_angular mirr2.1
```

Treatment Page



This page is specific to the surface that was selected using the **OOB** described in a previous section.

The **Treatment** page can be used to modify the surface topology and ray-tracing properties of the selected surface.

Treatment

Discussion:

A treatment represents an equally spaced array of identical surface features, such as a section of a sphere or cone, or a conical or hyperbolic trough.

A treatment should only be applied to a flat (unmodified) surface, including surface 1 or 2 of a lens where the surface curvature has been defined as flat.

Treatments are not drawn, but can be visualized by raytracing the surface with the absorptance set to 1.0.

X_BEf or Y_BEf

Function:

Modifies the surface to represent a **Brightness Enhanced Film** structure in x or y

Discussion:

A BEF (Brightness Enhanced Film) has a structured surface of parallel linear troughs. BEF objects are used to enhance the performance of backlit displays.

An example of a BEF is Vikuiti™ produced and trademarked by 3M Corporation.

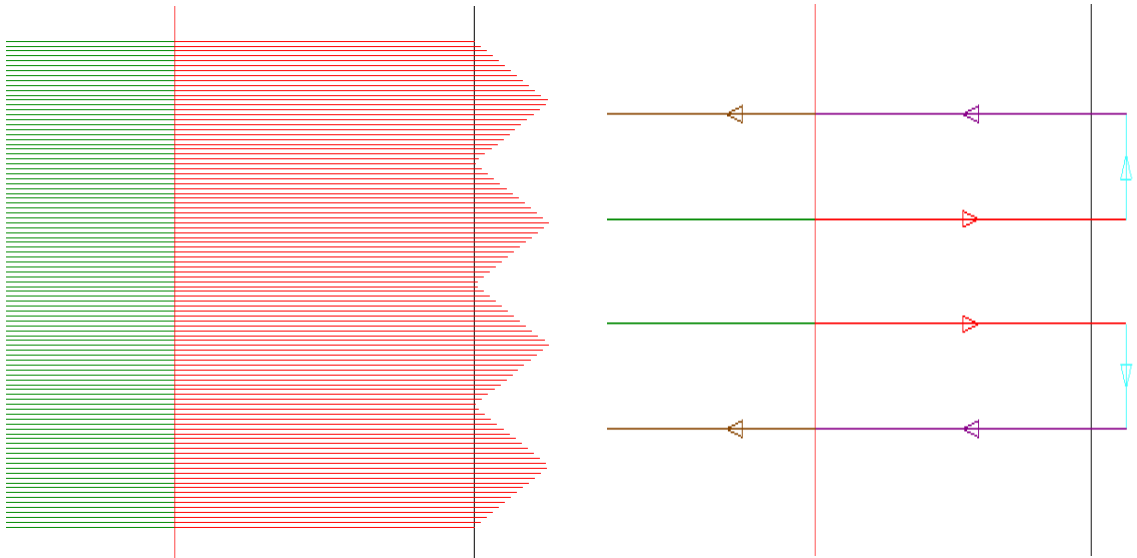


Figure above left shows the structure of a BEF, above right shows on-axis rays recycled by the BEF.

Parameters:

The surface name (`object_name.surface_number`), separation of the repeated structural elements (x spacing or y spacing), half angle (degrees) with the z-axis of the tip of the trough (`theta_half_angle`).

Note that a positive sign for the separation puts the structure to the +z side of the base surface, while a negative sign puts the structure to the left of the base surface.

Text Command:

```
treatment/x_bef_attach my_object.surface spacing theta
treatment/y_bef_attach my_object.surface spacing theta
```

Text Command Examples:

```
treatment/x_bef_attach my_bef.1 -0.05 45
treatment/y_bef_attach my_bef.1 -0.05 45
treatment/x_bef_attach my_bef.2 0.05 45
treatment/y_bef_attach my_bef.2 0.05 45
```

X_Hyperbolic_BEf or Y_Hyperbolic_BEf

Function:

Modifies the surface to represent a **Hyperbolic Brightness Enhanced Film** structure in x or y

Discussion:

A Hyperbolic BEF (Brightness Enhanced Film) has a structured surface of extruded hyperbolas, having a small radius of curvature. A Hyperbolic BEF is similar to the standard BEF describe above except that the tip is somewhat rounded. Hyperbolic BEF objects are used to enhance the performance of backlit displays.

An example of a BEF is Vikuiti™ produced and trademarked by 3M Corporation.

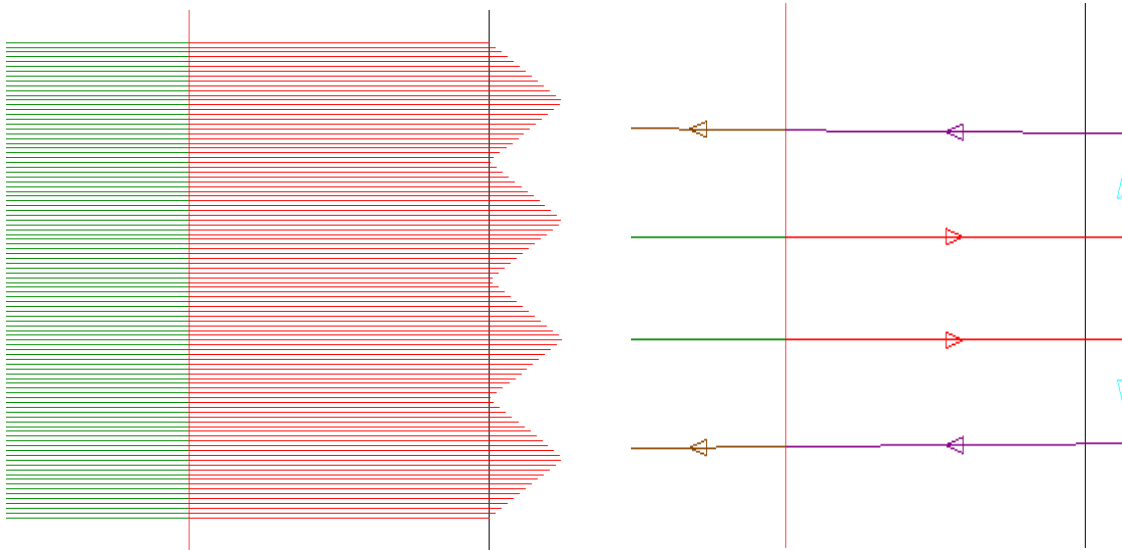


Figure above left shows the structure of a hyperbolic BEF, above right shows on-axis rays recycled by the hyperbolic BEF.

Parameters:

The surface name (`object_name.surface_number`), separation of the repeated structural elements (x spacing or y spacing), radius of the tip (radius), project half angle (degrees) with the z-axis of the hyperbolic trough (theta_half_angle).

Note that a positive sign for the separation puts the structure to the +z side of the base surface, while a negative sign puts the structure to the left of the base surface.

Text Command:

```
treatment/x_hyp_bef_attach my_object.surface spacing radius theta
treatment/y_hyp_bef_attach my_object.surface spacing radius theta
```

Text Command Examples:

```
treatment/x_bef_bef_attach my_bef.1 -0.05 0.01 45
treatment/y_bef_bef_attach my_bef.1 -0.05 0.01 45
treatment/x_bef_bef_attach my_bef.2 0.05 0.01 45
treatment/y_bef_bef_attach my_bef.2 0.05 0.01 45
```

Conical surface structure

Function:

Modifies the surface to represent a **2-D array of cones**

Discussion:

This type of surface treatment is used in backlit displays.

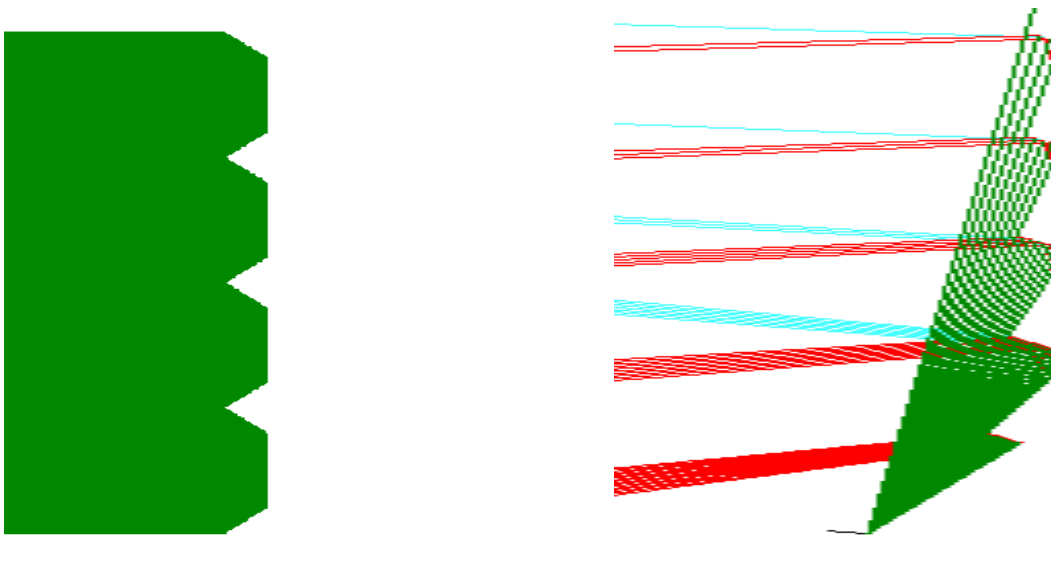


Figure above left shows the structure of an array of cones, above right shows off-axis rays redirected by the array of cones.

Parameters:

The surface name (`object_name.surface_number`), separation of the repeated structural elements in x (x spacing), separation of the repeated structural elements in y (y spacing), half size of a single cone (feature half size), height of the cone from the base surface (feature height), and the reflectance of the feature.

A positive sign for the feature height puts the structure to the +z side of the base surface, while a negative sign puts the structure to the left of the base surface.

Rays only see the treatment features from the feature side of the base surface, rays hitting the non-treatment side see the base surface.

For non-zero feature reflectance, the feature absorptance is calculated as $1 - \text{feature reflectance}$, for zero feature reflectance, the feature absorptance is zero.

Text Command:

```
treatment/cone_attach my_object.surface x_spacing y_spacing st_rhs
st_zheight st_reflectance
```

Text Command Examples:

```
treatment/cone_attach my_cones.1 0.05 0.05 0.025 0.025 1.0
treatment/cone_attach my_cones.2 0.05 0.05 0.025 -0.025 0.0
```

Spherical surface structure

Function:

Modifies the surface to represent a **2-D array of partial spheres**

Discussion:

This type of surface treatment is used in backlit displays.

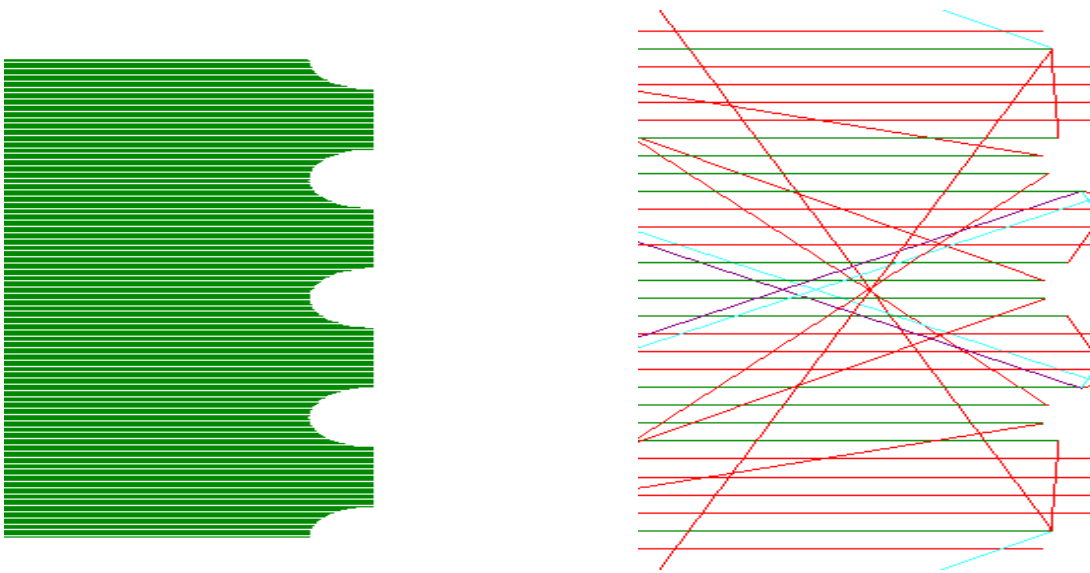


Figure above left shows the structure of an array of spheres, above right shows rays redirected by the array of spheres.

Parameters:

The surface name (`object_name.surface_number`), separation of the repeated structural elements in x (x spacing), separation of the repeated structural elements in y (y spacing), half size of a single cone (feature half size), height of the sphere from the base surface (feature height), and the reflectance of the feature.

A positive sign for the feature height puts the structure to the +z side of the base surface, while a negative sign puts the structure to the left of the base surface.

Rays only see the treatment features from the feature side of the base surface, rays hitting the non-treatment side see the base surface.

For non-zero feature reflectance, the feature absorptance is calculated as $1 - \text{feature reflectance}$, for zero feature reflectance, the feature absorptance is zero.

Text Command:

```
treatment/sphere_attach my_object.surface x_spacing y_spacing st_rhs
st_zheight st_reflectance
```

Text Command Examples:

```
treatment/sphere_attach my_cones.1 0.05 0.05 0.025 0.025 1.0
treatment/sphere_attach my_cones.2 0.05 0.05 0.025 -0.025 0.95
```

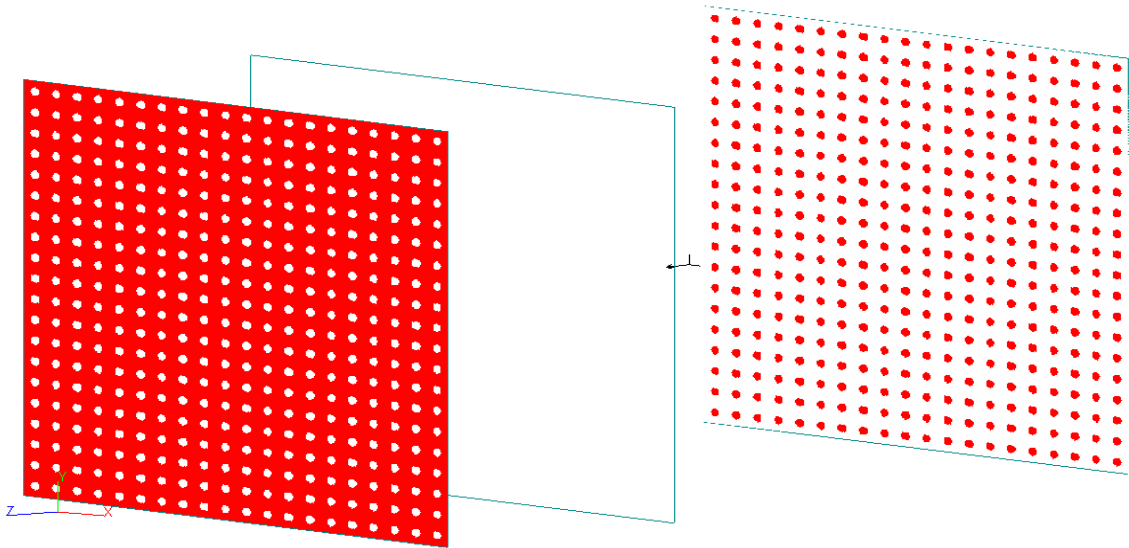
Dots surface structure

Function:

Modifies the surface to represent a **2-D array of dots**

Discussion:

This type of surface treatment is used in backlit displays.



A pattern of reflective dots on a clear substrate (center), reflected rays on a radiometer (right), transmitted rays on a second radiometer (left).

Parameters:

The surface name (`object_name.surface_number`), separation of the repeated structural elements in x (x spacing), separation of the repeated structural elements in y (y spacing), half size of a single cone (feature half size), and the reflectance of the feature.

A positive sign for the feature height puts the structure to the +z side of the base surface, while a negative sign puts the structure to the left of the base surface.

Rays only see the treatment features from the feature side of the base surface, rays hitting the non-treatment side see the base surface.

For non-zero feature reflectance, the feature absorptance is calculated as 1 - the feature reflectance, for zero feature reflectance, the feature absorptance is zero.

The dots have no thickness, a dot on a free standing surface having zero reflectance is a hole.

Text Command:

```
treatment/dots_attach my_object.surface x_spacing y_spacing st_rhs
st_reflectance
```

Text Command Examples:

```
treatment/dots_attach my_polkadots.1 0.05 0.05 0.025 1.0
treatment/dots_attach my_polkadots.2 0.05 0.05 0.025 0.0
```

Remove surface treatment

Function:

Removes all treatments from the surface

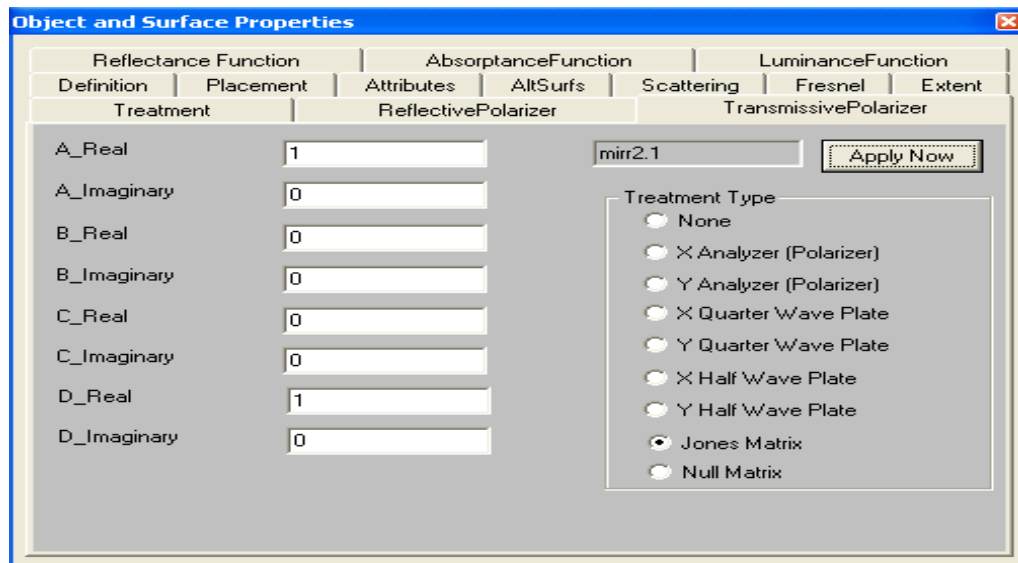
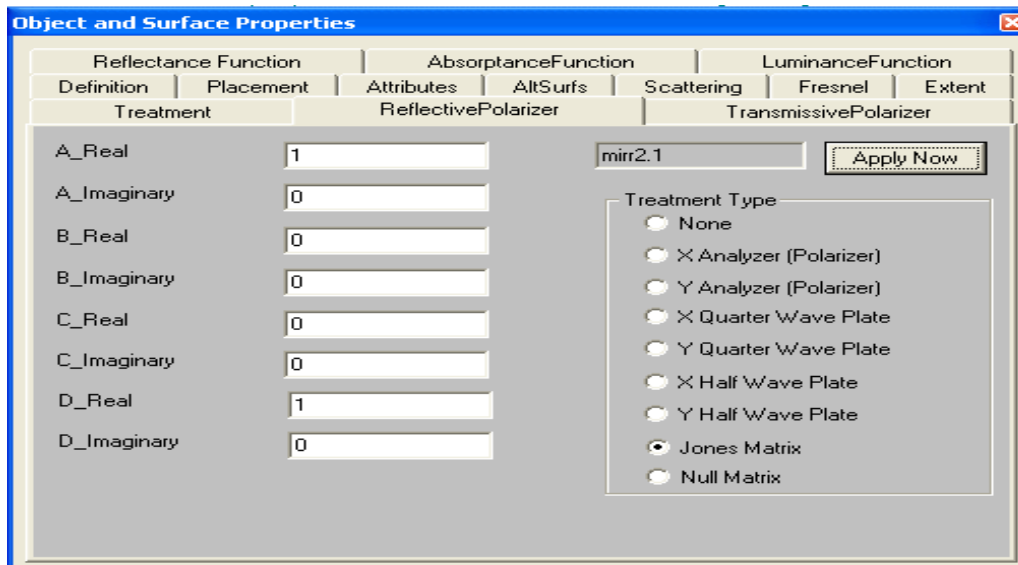
Text Command:

```
treatment/remove my_object.surface
```

Text Command Examples:

```
treatment/remove my_treatment.1
```

Reflective and Transmissive Polarization Pages



These pages are specific to the surface that was selected using the **OOB** described in a previous section.

The **Reflective and Transmissive Polarization** pages can be used to add polarization sensitivity properties to the selected surface.

Polarization Properties

Discussion:

Polarization sensitive surface properties may be explicitly defined independently for reflected and transmitted rays. Properties include: x and y analyzers (polarizers), x and y quarter wave plates, x and y half wave plates, Jones matrix (four complex coefficients), and a null Jones matrix.

Polarizing surface properties are described in the **Defining polarization sensitive explicit surface properties** section of the **Coatings and Polarization** chapter.

The reflective Jones matrix, whether defined by name or by direct entry is first multiplied by the square root of the surface reflectance defined on the **Reflective Polarization** page of the **Object and Surface Properties** box, before being used in a calculation.

The transmissive Jones matrix, whether defined by name or by direct entry is first multiplied by the square root of the surface transmittance defined on the **Transmissive Polarization** page of the **Object and Surface Properties** box, before being used in a calculation.

Analyzers, wave plates and Jones matrices take precedence over both dielectric and metal coatings, and the reflectance and absorptance entered on the **Attributes** page of the **Object and Surface Properties** box. These properties are independent of both wavelength and angle of incidence.

Warning: Because you can independently specify the reflectance and transmittance, you must use great care in order to assure a physically real result. As an example, you can specify an x analyzer with a transmittance of 1.0 and a y analyzer with reflectance of 1.0 on the same surface. This would model a perfect polarization beam splitter or polarization recycler. However, specifying an x analyzer with a transmittance of 1.0 and an x analyzer with reflectance of 1.0 on the same surface, would cause a physically unreal result.

X Analyzer

Parameter:

The surface name (`object_name.surface_number`).

Reflective Text Command:

```
polarization/r_x_analyzer object.surface_name
```

Transmissive Text Command:

```
polarization/t_x_analyzer object.surface_name
```

Y Analyzer

Parameter:

The surface name (`object_name.surface_number`).

Reflective Text Command:

```
polarization/r_y_analyzer object.surface_name
```

Transmissive Text Command:

```
polarization/t_y_analyzer object.surface_name
```

X Quarter Wave Plate

Parameter:

The surface name (`object_name.surface_number`).

Reflective Text Command:

polarization/r_x_quarter_wave **object.surface_name**

Transmissive Text Command:

polarization/t_x_quarter_wave **object.surface_name**

Y Quarter Wave Plate

Parameter:

The surface name (**object_name.surface_number**).

Reflective Text Commands:

polarization/r_y_quarter_wave **object.surface_name**

Transmissive Text Commands:

polarization/t_y_quarter_wave **object.surface_name**

X Half Wave Plate

Parameter:

The surface name (**object_name.surface_number**).

Reflective Text Command:

polarization/r_x_half_wave **object.surface_name**

Transmissive Text Command:

polarization/t_x_half_wave **object.surface_name**

Y Half Wave Plate

Parameter:

The surface name (**object_name.surface_number**).

Reflective Text Command:

polarization/r_y_half_wave **object.surface_name**

Transmissive Text Command:

polarization/t_y_half_wave **object.surface_name**

Jones Matrix

Parameters:

The surface name (**object_name.surface_number**), and the matrix elements: A_real (p1), A_imaginary (p2), B_real (p3), B_imaginary (p4), C_real (p5), C_imaginary (p6), D_real (p7), D_imaginary (p8).

Reflective Text Command:

polarization/r_jones **object.surface_name** p1 p2 p3 p4 p5 p6 p7 p8

Transmissive Text Command:

polarization/t_jones **object.surface_name** p1 p2 p3 p4 p5 p6 p7 p8

Null Matrix

Parameters:

The surface name (`object_name.surface_number`).

Reflective Text Command:

```
polarization/r_null object.surface_name p1 p2 p3 p4 p5 p6 p7 p8
```

Transmissive Text Command:

```
polarization/t_null object.surface_name p1 p2 p3 p4 p5 p6 p7 p8
```

Remove Polarization Properties

Parameters:

The surface name (`object_name.surface_number`).

Remove Reflective Polarization Surface Properties:

```
polarization/r_remove object.surface_name
```

Remove Transmissive Polarization Surface Properties:

```
polarization/t_remove object.surface_name
```

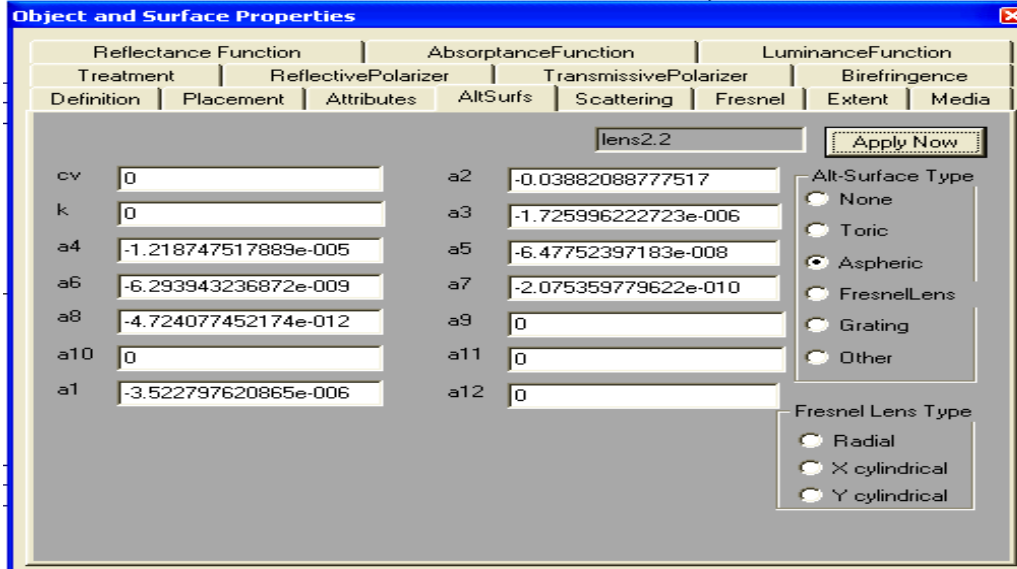
Remove Both Reflective and Transmissive Polarization Surface Properties:

```
polarization/remove object.surface_name
```

Text Command Examples:

```
polarization/r_x_analyzer myobject.1  
polarization/r_y_analyzer myobject.1  
polarization/r_x_quarter_wave myobject.1  
polarization/r_y_quarter_wave myobject.1  
polarization/r_x_half_wave myobject.1  
polarization/r_y_half_wave myobject.1  
polarization/r_null myobject.1  
polarization/r_remove myobject.1  
polarization/r_jones myobject.1 1 0 0 0 0 0 1 0  
  
polarization/t_x_analyzer myobject.1  
polarization/t_y_analyzer myobject.1  
polarization/t_x_quarter_wave myobject.1  
polarization/t_y_quarter_wave myobject.1  
polarization/t_x_half_wave myobject.1  
polarization/t_y_half_wave myobject.1  
polarization/t_null myobject.1  
polarization/t_remove myobject.1  
polarization/t_jones myobject.1 1 0 0 0 0 0 1 0  
polarization/remove myobject.1
```

AltSurfs Page



This page is specific to the surface that was selected using the **OOB** described in a previous section.

The **AltSurfs** page can be used to modify the shape or ray-tracing properties of the selected surface.

AltSurface

Discussion:

This surface feature contains several modifications requiring extra parameters (aspherics, grating frequency, Fresnel lens parameters). These surface modifications can, with a few exceptions, be applied to many open native surface (although it may not be appropriate), and, with a some exceptions, may be applied in combination.

A flat mirror is treated as a special shape, the shape of a flat mirror should not be modified; instead, start with a spherical or parabolic mirror.

Note that an **AltSurface** may be applied to either surfaces 1 or 2 of a lens. An **AltSurface** should never be applied to surface 3 of a spherical lens, or surfaces 3-6 of a cylindrical lens.

Iterative Intercept Method

Nonconic surfaces are generally ray-traced using an iterative ray-intercept method,. The method used is called the Newton-Rhapson method.

The Newton-Rhapson method works extremely well for surfaces that deviate from an initial (conic) parent surface by only a small amount. When the surface being ray-traced has extreme deviations from a best-estimate conic surface, the ray may not intercept the true surface. The user should pay particular attention to rays that pass through surfaces undeviated as these rays may not be physically correct.

Multiple AltSurfaces

The following **AltSurface** combinations are permitted: a) a grating may me added to a Toric surface, b) a grating may be added to an aspheric surface.

Toric

Function:

Modifies the surface to a **toroidal shape**

Discussion:

Models a single valued toroidal surface (section of a doughnut).

If rotationally toric is to applied to either surface 1 or surface 2 of a cylindrical lens, the initial surface must be defined as flat ($r1 = 0$ or $r2 = 0$).

Toric surfaces are ray-traced using an iterative method.

A toroidal surface is formed by defining a curve in the y-z plane, and then rotating the curve about an axis parallel to the y-axis (and intersecting the z-axis). The curve in the y-z plane is defined as:

$$z = \frac{cv_y \cdot y^2}{1 + \sqrt{1 - (cc_y + 1) \cdot cv_y^2 \cdot y^2}} + a_2 \cdot y^2 + a_4 \cdot y^4 + a_6 \cdot y^6 + a_8 \cdot y^8 + a_{10} \cdot y^{10}$$

This curve is then rotated about an axis located a distance R_x (the radius of revolution) from the vertex. The toroidal surface is shifted along the z-axis such that the point $x = y = 0$ is tangent to the x-y plane.

The following relationship is useful:

$$cv_y = \frac{1}{R_y}$$

Caution: This feature will reset the parent (quadric) surface to be an approximation to the selected toric surface. Therefore removing (see the **Remove** description below) a **Toric** will require the additional step of going to the Definitions page and clicking **ApplyNow**.

In order to model an aspheric cylindrical surface with power in the y direction, enter 0.0 for R_x .

This surface may not be applied to an imported CAD object (NURBS), and should not be applied to a closed cavity.

Parameters:

The surface name (`object_name.surface_number`), radius of revolution (R_x), the radius of curvature in y (R_y), the conic constant in y (cc_y), the second order polynomial coefficient in y (a_2), the fourth order polynomial coefficient in y (a_4), the sixth order polynomial coefficient in y (a_6), the eighth order polynomial coefficient in y (a_8), the tenth order polynomial coefficient in y (a_{10}).

Text Command:

modify/altsurface/toric **my_object.surface** Rx Ry ccy a4 a6 a8 a10 a2

Text Command Example:

```
modify/altsurface/toric tor1.1 100 50 -1  
modify/altsurface/toric tor1.1 0 50 -1 0.0001 -0.000001 0.00000001 -1e-10 0
```

Aspheric

Function:

Adds a **rotationally symmetric aspheric** to a **surface**

Discussion:

Models a spherical surface to have a conic term (conic constant) and four rotationally symmetric aspheric terms.

Caution: This feature adds a rotationally symmetric aspheric to any surface it is applied to, even a cylindrical surface. If this feature is applied to a cylinder surface, it will reset the parent (quadric) surface to be a rotationally symmetric surface. Therefore removing (see the **Remove** description below) an **Aspheric** from a formerly cylindrical surface will require the additional step of going to the Definitions page and clicking **ApplyNow**.

Note that aspheric surfaces are ray-traced using an iterative method, which has both user-set controls and basic limits.

The expression for a rotationally symmetric aspheric surface profile is:

$$z = \frac{cv \cdot h^2}{1 + \sqrt{1 - ((k + 1) \cdot cv^2 \cdot h^2)}} + a_1 \cdot h^1 + a_2 \cdot h^2 + \dots + a_{12} \cdot h^{12}$$

where (cv) is the curvature

$$cv = \frac{1}{radius} = \frac{1}{2 \cdot focallength}$$

and k is the conic constant, a_1 , a_2 , a_3 , and a_4 are user-defined coefficients, and

$$h = \sqrt{x^2 + y^2}$$

The conic constant is also $k = -e^2$, where e is the eccentricity.

Examples of values of the conic constant k (assuming $a_1 = a_2 = \dots = a_{12} = 0$)

For $k = -1$

$$z = \frac{cv \cdot h^2}{2}$$

is a paraboloid.

For $k = 0$

$$z = \frac{cv \cdot h^2}{1 + \sqrt{1 - (cv^2 \cdot h^2)}}$$

is a sphere.

For $-1 > k > 0$ the surface is a ellipsoid of rotation about the major axis (also called a prolate spheroid).

For $k > 0$ the surface is an ellipsoid of rotation about the minor axis (also called an oblate spheroid).

For $k < -1$ the surface is a hyperboloid, for $a = b$, and in the single valued region of the vertex ($z = 0$) obeys the equation

$$\frac{x^2 + y^2}{a^2} - \frac{(z - c)^2}{c^2} = -1$$

where:

$$a = \frac{cv}{|cv|} \cdot \sqrt{\frac{-1}{cv^2 \cdot (k + 1)}}$$

and

$$c = \frac{-1}{cv \cdot (k + 1)}$$

and the two focal points of the hyperboloid are

$$z_1 = -a - \sqrt{a^2 + c^2}$$

and

$$z_2 = -a + \sqrt{a^2 + c^2}$$

so that for $(cv) > 0$ light originating from $z < 0$ and traveling toward $(0,0,z_f^+)$ is redirected to $(0,0,z_f^-)$. A spherical mirror is tangent to the local x-y plane.

This surface may not be applied to an imported CAD object (NURBS), and should not be applied to a closed cavity.

Parameters:

Surface name (object_name.surface_number), curvature (1/radius), four aspheric coefficients (defined above).

Text Command:

```

modify/altsurface/aspheric/radial my_object.surface cv k a4 a6 a8 a10 a1 a2
a3 a5 a7 a9 a11 a12
modify/altsurface/aspheric my_object.surface cv k a4 a6 a8 a10 a1 a2 a3 a5
a7 a9 a11 a12

```

Text Command Examples:

```

modify/altsurf/aspheric/radial lens1.1 0.1 -1 1E-6 2E-8 3E-10 4E-12 0.1 0.01
0.001 1E-5 1E-7 1E-9 1E-11 1E-12
modify/altsurf/aspheric lens1.2 0.1 -1 1E-6 2E-8 3E-10 4E-12 0.1 0.01 0.001
1E-5 1E-7 1E-9 1E-11 1E-12

```

Fresnel Lenses

Function:

Adds a Fresnel lens

Discussion:

A Fresnel lens formulation is added to the local-surface-normal calculation. The Fresnel lens has the following constraints: a) the imaging properties of the formula assume a very thin single element lens with two flat surfaces, b) the formula is a redefinition of the local surface normals, it does not model the physical grooves.

The object distance (parameter *s*) is a positive number measured along the z-axis from left (object point) to right (the middle of the lens). The image distance (parameter *sp*) is a positive number measured along the z-axis from left (the middle of the lens) to right (image point).

The formulas for the Fresnel lens are shown below:

$$LSN = \frac{\partial \text{ApparentSag}}{\partial h} = \frac{s_1 + s_2}{\sqrt{c_1^2 + (s_1 + s_2)^2}}$$

for the radial option:

$$h = \sqrt{x^2 + y^2}$$

for the `x_cylinder` option $h = x$, and for the `y_cylinder` option $h = y$

$$s_1 = \frac{h}{\sqrt{h^2 + s^2}}$$

$$s_2 = \frac{h}{\sqrt{h^2 + sp^2}}$$

$$c_1 = sp \cdot f_2 - \sqrt{c_0}$$

$$c_0 = n^2 - s_1^2$$

$$f_2 = \frac{1}{\sqrt{h^2 + sp^2}}$$

where s is the object distance, sp is the image distance, n is the index of refraction of the material.

When the user specifies an object and image distance, a Fresnel lens is defined that images an on-axis point originating from the object point (object distance to the left of the lens) into a perfect image at the image point (image distance to the right of the lens).

The Fresnel types options in the first input box to appear has three choices: **Radial**, which produces a rotationally symmetric Fresnel lens (as if it had round grooves); **x-Cylinder**, which produces a Fresnel lens with power only in the x direction (as if it had linear grooves parallel to the y -axis); **y-Cylinder**, which produces a Fresnel lens with power only in the y direction (as if it had linear grooves parallel to the x -axis).

Note that a Fresnel lens may not be applied to a reflective surface.

Parameters:

The surface name (object_name.surface_number), which currently must be the second surface (object_name.2); object distance (s); image distance (sp); and power parameter (radial or cylindrical).

Text Commands:

```
modify/altsurface/fresnel/radial my_object.surface s sp
modify/altsurface/fresnel/x_cylindrical my_object.surface s sp
modify/altsurface/fresnel/y_cylindrical my_object.surface s sp
```

Text Command Examples:

```
modify/altsurface/fresnel/radial fres1.2 1E10 100
modify/altsurface/fresnel/x_cylindrical fres1.2 1E10 100
modify/altsurface/fresnel/y_cylindrical fres1.2 1E10 100
```

Grating

Function:

Adds a diffraction grating

Discussion:

The diffraction grating feature can be used to apply a straight-line ruled grating to a planar surface. The diffraction grating is a special case of a hologram formed by two plane wave construction beams, symmetrically incident on the surface. The grating lines are parallel to the local y-axis.

The grating is applied in the medium after refraction or reflection. For a plane grating the formulas for the ray direction cosines are

$$L' = L_0 + \frac{m \cdot \lambda \cdot f \cdot \cos \theta}{n'}$$

and

$$M' = M_0 + \frac{m \cdot \lambda \cdot f \cdot \sin \theta}{n'}$$

where L_0 and M_0 are the direction cosines after refraction or reflection but before the grating operation, m is the grating order, λ is the wavelength (microns), f is the grating frequency (mm^{-1}), n' is the index of refraction after the surface, θ is the orientation of the grating, and L' and M' are the new direction cosines after the grating operation.

OptiCAD only models gratings to the extent of deviating ray paths, grating efficiency is not modeled.

Parameters:

Surface name (`object_name.surface_number`), grating frequency, grating orientation, grating order, and the wavelength (which is set using the Options → Wavelength feature).

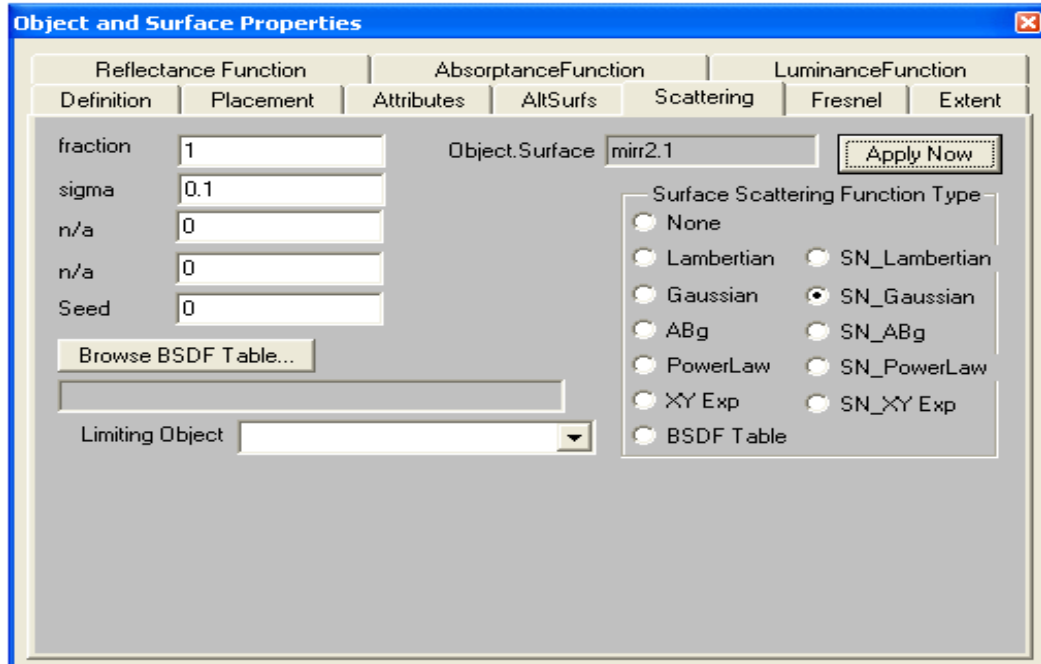
Text Command:

```
modify/altsurf/grating my_object.surface frequency theta_orientation order
```

Text Command Example:

```
modify/altsurface/grating lens1.1 500 0 -1
```

Scattering Page



This page is specific to the surface that was selected using the **OOB** described in a previous section.

The **Scattering** page allows for the definition of a surface scattering model and the entry of numerical constants for the selected surface.

Types of Scattering Models

OptiCAD has two types of scattering models

Post Surface Scattering

The **post surface** type of model calculates the specular ray direction (after refraction or reflection) and perturbs the scattered ray about the specular ray direction. Input to this type of scattering can be from a scattering distribution measured on scatterometer. The distribution can be scaled between the measuring wavelength and the evaluation wavelength. The **post surface** type of model treats scattering in a similar manner to a diffraction grating of weighted random frequency and orientation. The method is limited when the incident ray is close to the critical angle for total internal reflection (T.I.R.), because it assumes that the ray leaves the object. The **post surface** type of model distributions are named Lambertian, Gaussian, ABg, PowerLaw, XY_Exp, and BSDF_Table.

Perturbed Surface Scattering

The **perturbed surface** type of model perturbs the local surface normal of the surface according to a user selected random distribution. This surface perturbation is done before the ray hits the surface. When the ray is incident on the surface the laws of refraction and reflection (including T.I.R.) are applied, resulting in a ray distribution. Input to this type of scattering can be from a surface profiler. This method models T.I.R. in a more realistic way than the **post surface** type of model, is however purely geometric and not wavelength scalable. The **perturbed surface** type of model distributions are named SN_Lambertian, SN_Gaussian, SN_ABg, SN_PowerLaw, and SN_XY_Exp.

Importance Sampling Scattering

Importance sampling allows the selection of a target object for scattered rays. The directions of all possible rays leaving a scattering surface is limited to those rays that hit a virtual sphere centered on the user selected **Limiting Object**. The importance sampling limiting function limits the rays in free space without considering the refraction or reflection of intermediate objects. In some cases the user may wish to use a transparent mirrors or cavity object as the **Limiting Object**, in order to have greater control over the scattered ray bundle.

Sobol Sampling

Monte Carlo sampling (the OptiCAD default) is a powerful tool for optical and illumination simulation. The disadvantage of Monte Carlo sampling is the slow convergence rate of

$$O[N^{-1/2}]$$

where N is the number of samples. In qualitative terms Monte Carlo sampling can be thought of as not caring what has happened before, and therefore has no way of knowing what regions need gaps in sampling to be filled.

Sobol sampling uses Low Discrepancy Sequences instead of random numbers as a deterministic alternative (Quasi-Monte Carlo) to Monte Carlo sampling. The upper bound of Sobol sampling is thought to be

$$O\left[\frac{(\log N)^d}{N}\right]$$

where d is the number of independent dimensions. Sobol sampling is continuously filling in gaps in the sample space. For large samples Sobol is proportional to 1/N which is a big improvement.

Sobol sampling can be activated for ray origins and directions from the Options menu and by the text commands

for spatial (ray origins) Sobol

```
set/sp_sobol/on  
set/sp_sobol/off
```

and for direction (ray angles) Sobol

```
set/di_sobol/on  
set/di_sobol/off
```

Sobol sampling is implemented for functions defined using the **Emitance Function** page of the **Object and Surfaces Properties Box** and most of the standard source dlls (\sources\standard\dlls). Dlls that are not Sobol compatible will produce an error message and default to Monte Carlo sampling.

Scattering Functions

In addition to choosing a fraction scattered, a scattering function and its parameters, the user may also wish to modify the surface reflectivity. The reflected and refracted rays will be operated on by the same scattering function having the same parameters. Only one scattering function may be applied to a particular surface at a time.

A scattering fraction of 0.0 indicates no rays are scattered, 1.0 indicates that all of the rays are scattered.

Most scattering functions allow the user to enter a seed for the random number generator. Entry of 0 for the seed (the default) causes the random number generator to be seeded with the system clock (random seed).

The Apply Now button only implements the values for this page.

Aborting the object change may be accomplished by closing the box (by clicking the x in upper right of the box).

None

Function:

Makes surface **specular**

Discussion:

Deletes all scattering from the surface making it 100% specular.

Parameters:

Surface name (object_name.surface_number).

Text Command:

```
modify/scatterer/none surface_name
```

Text Command Example:

```
modify/scatterer/none mirr1.1
```

Lambertian

Function:

Defines a **Lambertian** scattering distribution

Discussion:

A Lambertian surface scatters rays uniformly in direction cosine space regardless of the direction of the specular ray.

Parameters:

Surface name (object_name.surface_number), fraction of the rays that are to be scattered (0 to 1).

Text Command:

```
modify/scatterer/Lambertian surface_name fraction_scattered seed  
modify/limitingobject="limiting_object"/scatterer/lambertian surface_name  
fraction_scattered seed
```

Text Command Example:

```
modify/scatterer/Lambertian lens2.1 1 0  
modify/limitingobject="refbot"/scatterer/lambertian lens2.1 1 5
```

SN Lambertian

Function:

Defines a **Lambertian** surface normal distribution

Discussion:

A Lambertian surface normal function distributes surface normals uniformly in direction cosine space regardless of the direction of the geometric surface normal.

Parameters:

Surface name (object_name.surface_number), fraction of the surface normals that are to be scattered (0 to 1).

Text Command:

```
modify/scatterer/sn_lambertian surface_name fraction_scattered seed  
modify/limitingobject="limiting_object"/scatterer/sn_lambertian  
surface_name fraction_scattered seed
```

Text Command Example:

```
modify/scatterer/sn_Lambertian lens2.1 1 0  
modify/limitingobject="refbot"/scatterer/sn_lambertian lens2.1 1 1
```

Gaussian

Function:

Defines a **Gaussian** scattering distribution

Discussion:

The rotationally symmetric Gaussian scattering function has the form

$$f_{l,m}(l, m) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left(-\frac{r^2}{2 \cdot \sigma^2}\right)$$

where

$$l = l_{scattered} - l_{specular}$$

and

$$m = m_{scattered} - m_{specular}$$

and

$$r = \sqrt{l^2 + m^2}$$

are all direction cosines with respect to the x and y axes, and σ is the standard deviation of the rotationally symmetric Gaussian scattering distribution.

Parameters:

Surface name (object_name.surface_number), fraction of the rays that are to be scattered (0 to 1), and standard deviation (in direction cosine space) of the ray distribution. Values of the standard deviation greater than 2 will cause the scattering function to be treated as Lambertian.

Text Command:

```
modify/scatterer/gaussian surface_name fraction_scattered sigma_gaussian
seed
modify/limitingobject="limiting_object"/scatterer/gaussian surface_name
fraction_scattered sigma_gaussian seed
```

Text Command Example:

```
modify/scatterer/gaussian lens1.2 1 0.1 0
modify/limitingobject="limiting_object"/scatterer/gaussian lens1.2 1 0.1 1
```

SN Gaussian

Function:

Defines a **Gaussian** surface normal distribution

Discussion:

The rotationally symmetric Gaussian surface normal function has the form

$$f_{l,m}(l, m) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left(-\frac{r^2}{2 \cdot \sigma^2}\right)$$

where

$$l = l_{\text{perturbation}} - l_{\text{geometric}}$$

and

$$m = m_{\text{perturbation}} - m_{\text{geometric}}$$

and

$$r = \sqrt{l^2 + m^2}$$

are all direction cosines with respect to the x and y axes, and σ is the standard deviation of the rotationally symmetric Gaussian surface normal distribution.

Parameters:

Surface name (object_name.surface_number), fraction of the surface normals that are to be scattered (0 to 1), and standard deviation (in direction cosine space) of the surface normal distribution. Values of the standard deviation greater than 2 will cause the surface normal function to be treated as Lambertian.

Text Command:

```
modify/scatterer/sn_gaussian surface_name fraction_scattered sigma_gaussian
seed
modify/limitingobject="limiting_object"/scatterer/sn_gaussian surface_name
fraction_scattered sigma_gaussian seed
```

Text Command Example:

```

modify/scatterer/sn_gaussian lens1.2 1 0.1 0
modify/limitingobject="limiting_object"/scatterer/sn_gaussian lens1.2 1 0.1
1

```

ABg

Function:

Defines an **ABg** scattering distribution

Discussion:

The rotationally symmetric ABg scattering function has the form

$$f_{l,m}(l, m) = \frac{A}{B + r^g}$$

where

$$l = l_{scattered} - l_{specular}$$

and

$$m = m_{scattered} - m_{specular}$$

and

$$r = \sqrt{l^2 + m^2}$$

are all direction cosines with respect to the x and y axes, A, B, and g are fit parameters for scattering data of the rotationally symmetric ABg scattering distribution.

Parameters:

Surface name (object_name.surface_number), fraction of the rays that are to be scattered (0 to 1), A, B, and g data fit values (in direction cosine space) of the scattering distribution.

Text Command:

```

modify/scatter/abg surface_name fraction_scattered a b g seed
modify/limitingobject="limiting_object"/scatterer/abg surface_name
fraction_scattered a b g seed

```

Text Command Example:

```

modify/scatterer/abg lens1.2 1 0.002 0.0005 2.0 0
modify/limitingobject="limiting_object"/scatterer/abg lens1.2 1 0.002 0.0005
1 2.0

```

SN_ABg

Function:

Defines an **ABg** surface normal distribution

Discussion:

The rotationally symmetric ABg surface normal function has the form

$$f_{l,m}(l, m) = \frac{A}{B + r^g}$$

where

$$l = l_{\text{perturbation}} - l_{\text{geometric}}$$

and

$$m = m_{\text{perturbation}} - m_{\text{geometric}}$$

and

$$r = \sqrt{l^2 + m^2}$$

are all direction cosines with respect to the x and y axes.

Parameters:

Surface name (object_name.surface_number), fraction of the surface normals that are to be scattered (0 to 1), A, B, and g values (in direction cosine space) of the surface normal distribution.

Text Command:

```
modify/scatter/sn_abg surface_name fraction_scattered a b g seed
modify/limitingobject="limiting_object"/scatterer/sn_abg surface_name
fraction_scattered a b g seed
```

Text Command Example:

```
modify/scatterer/sn_abg lens1.2 1 0.002 0.0005 2.0 0
modify/limitingobject="limiting_object"/scatterer/sn_abg lens1.2 1 0.002 1
0.0005 2.0
```

PowerLaw

Function:

Defines a **power-law** scattering distribution

Discussion:

The rotationally symmetric power-law scattering function has the form

$$f_{l,m}(l, m) = \frac{2-N}{2^{2-N}} \cdot r^{-N}$$

where

$$l = l_{\text{scattered}} - l_{\text{specular}}$$

and

$$m = m_{scattered} - m_{specular}$$

and

$$r = \sqrt{l^2 + m^2}$$

are all direction cosines with respect to the x and y axes, and N is the power constant of the rotationally symmetric power-law scattering distribution.

Parameters:

Surface name (object_name.surface_number), fraction of the rays that are to be scattered (0 to 1), and the power constant (between 0 and 2) in the expression above.

Text Command:

```
modify/scatterer/powerlaw surface_name fraction_scattered power seed  
modify/limitingobject="limiting_object"/scatterer/powerlaw surface_name  
fraction_scattered power seed
```

Text Command Example:

```
modify/scatterer/powerlaw mirr2.1 1 1.41 0  
modify/limitingobject="limiting_object"/scatterer/powerlaw mirr2.1 1 1.41 1
```

SN PowerLaw

Function:

Defines a **power-law** surface normal distribution

Discussion:

The rotationally symmetric power-law surface normal function has the form

$$f_{l,m}(l, m) = \frac{2-N}{2^{2-N}} \cdot r^{-N}$$

where

$$l = l_{perturbation} - l_{geometric}$$

and

$$m = m_{perturbation} - m_{geometric}$$

and

$$r = \sqrt{l^2 + m^2}$$

are all direction cosines with respect to the x and y axes, and N is the power constant of the rotationally symmetric power-law surface normal distribution.

Parameters:

Surface name (object_name.surface_number), fraction of the surface normals that are to be scattered (0 to 1), and the power constant (between 0 and 2) in the expression above.

Text Command:

```
modify/scatterer/sn_powerlaw surface_name fraction_scattered power seed
modify/limitingobject="limiting_object"/scatterer/sn_powerlaw surface_name
fraction_scattered power seed
```

Text Command Example:

```
modify/scatterer/sn_powerlaw mirr2.1 1 1.41 0
modify/limitingobject="limiting_object"/scatterer/sn_powerlaw mirr2.1 1 1.41
1
```

XY Exponential

Function:

Defines an **x-y Exponential Scattering Distribution**

Discussion:

The x-y exponential scattering distribution has the form

$$f_{l,m}(l, m) = \frac{1}{4 \cdot \sigma_l \cdot \sigma_m} \cdot \exp\left(-\frac{l}{\sigma_l}\right) \cdot \exp\left(-\frac{m}{\sigma_m}\right)$$

where

$$l = l_{scattered} - l_{specular}$$

and

$$m = m_{scattered} - m_{specular}$$

are all direction cosines with respect to the x- and y-axes, and σ_l and σ_m are the standard deviations in l and m of the x-y exponential scattering distribution.

Parameters:

Surface name (**object_name.surface_number**), fraction of the rays that are to be scattered (0 to 1), the x standard deviation (in direction cosine space) of the exponential scattering distribution, and the y standard deviation (in direction cosine space) of the exponential scattering distribution.

Text Command:

```
modify/scatterer/xy_exponential surface_name fraction_scattered xsigma
ysigma
modify/limitingobject="limiting_object"/scatterer/xy_exponential seed
surface_name fraction_scattered xsigma ysigma seed
```

Text Command Example:

```
modify/scatterer/xy_exponential flat1.1 1 2 0.1 0
modify/limitingobject="limiting_object"/scatterer/xy_exponential flat1.1 1 2
1 0.1
```

SN XY Exponential

Function:

Defines an **x-y Exponential** surface normal distribution

Discussion:

The x-y exponential surface normal distribution has the form

$$f_{l,m}(l, m) = \frac{1}{4 \cdot \sigma_l \cdot \sigma_m} \cdot \exp\left(-\frac{l}{\sigma_l}\right) \cdot \exp\left(-\frac{m}{\sigma_m}\right)$$

where

$$l = l_{perturbation} - l_{geometric}$$

and

$$m = m_{perturbation} - m_{geometric}$$

are all direction cosines with respect to the x- and y-axes, and σ_l and σ_m are the standard deviations in l and m of the x-y exponential surface normal distribution.

Parameters:

Surface name (`object_name.surface_number`), fraction of the surface normals that are to be scattered (0 to 1), the x standard deviation (in direction cosine space) of the exponential surface normal distribution, and the y standard deviation (in direction cosine space) of the exponential surface normal distribution.

Text Command:

```
modify/scatterer/sn_xy_exponential surface_name fraction_scattered xsigma  
ysigma seed  
modify/limitingobject="limiting_object"/sn_scatterer/xy_exponential  
surface_name fraction_scattered xsigma ysigma seed
```

Text Command Example:

```
modify/scatterer/sn_xy_exponential flat1.1 1 2 0.1 0  
modify/limitingobject="limiting_object"/scatterer/sn_xy_exponential flat1.1  
1 2 0.1 1
```

BPDF tabular data based scattering

General Function:

Modifies a **surface** to be **diffuse**

General Discussion:

OptiCAD models scattering (diffuse) surfaces based on a probabilistic change in the direction of the specular (unscattered) ray. The Bidirectional Scattering Distribution Function (BSDF) is treated as a probability density function, in direction cosine space.

BSDF includes BTDF (Bidirectional Transmittance Distribution Function) and BRDF (Bidirectional Reflectance Distribution Function)

The raytrace procedure is as follows:

- 1.The direction of the ray(s) leaving a surface (by either reflection or refraction or both) is calculated, based on measured BSDF tabular data.
- 2.The new ray direction is determined based on a weighted random number, a probability density function, based on interpolated BSDF tabular data. The scattered rays are centered about the direction of the specular (unscattered) ray.

The reflected and refracted rays will be operated on by the same scattering function having the same parameters. Only one scattering function may be applied to a particular surface at a time.

This scattering function is based on a dll named BSDFTABLE.dll. The file containing the tabular BSDF data and the dll (BSDFTABLE.dll) must be in the same directory.

The Apply Now button only implements the values for this page.

Aborting the object change may be accomplished by closing the box (by clicking the x in upper right of the box).

ASCII BSDF data format

Function:

Modifies a surface to be diffuse based on scattering data

Discussion:

The user selects a file containing scattering data, the dll named BSDFTABLE.dll must be in the same directory as the selected data.

OptiCAD will perform wavelength scaling of the scattering data, based on the measurement wavelength (which is included the data file), and the raytrace wavelength. An entry of zero for the measurement wavelength will disable wavelength scaling.

The data format for the BSDF data is as follows:

Data Format Description:

```
#_of_Incidence_Angles  #_of_Data_Pairs  Measurement_Wavelength(microns)
Incidence_Angle_(degrees)_for_following_Data_Pairs
theta(degrees) BSDF
theta(degrees) BSDF
theta(degrees) BSDF
theta(degrees) BSDF
theta(degrees) BSDF
Incidence_Angle_(degrees)_for_following_Data_Pairs
theta(degrees) BSDF
theta(degrees) BSDF
theta(degrees) BSDF
theta(degrees) BSDF
theta(degrees) BSDF
```

Example: Data for data taken at 2 angles of incidence (0 and 60 degrees); 10 data point pairs (scatter angle and BSDF) at each angle of incidence; all data taken at a wavelength of 0.5 microns:

```

2  10  0.50
0
1  1.99051809
2  0.42099087
3  0.157080945
4  0.077209815
5   0.044394346
60
1   0.247165939
2   0.075873981
3   0.035223021
4   0.020137071
5   0.012993554

```

Parameters:

Surface name, file name containing the BSDF data.

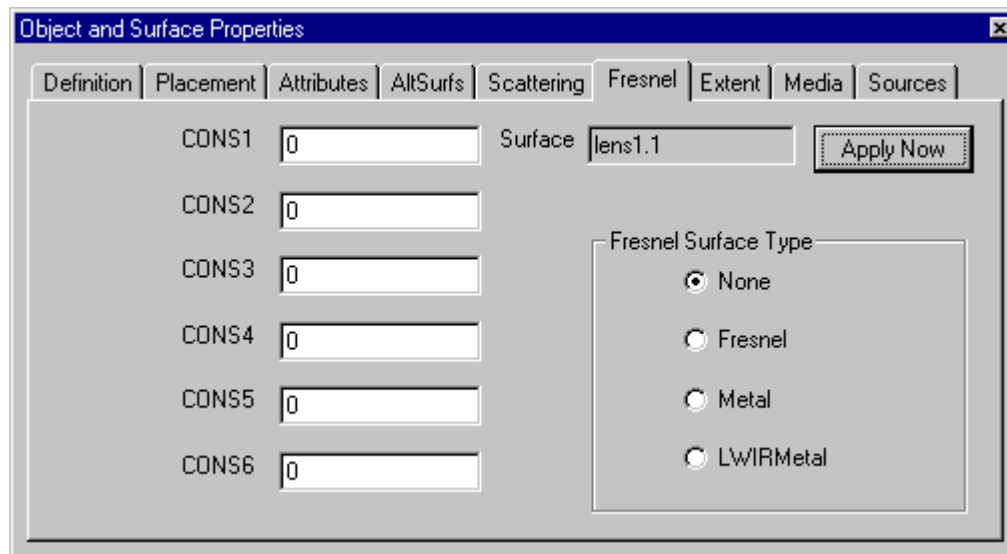
Text Command:

```
modify/scatterer/bsdftable/name="Data_File_Name" surface_name
```

Text Command Example:

```
modify/scatterer/bsdftable/name="scatterer\bsdf\BRDF_HW2.5.txt" mirr1.1
```

Fresnel Page



This page is specific to the surface that was selected using the **OOB** described in a previous section.

The **Fresnel page** allows for the selected surface to demonstrate one of several types of angle dependent reflectivity.

Note that this option takes precedence over the Surface Attributes option. For the reflectance and absorptance attributes to be used this option must first be set to None.

Note that entries on the **Fresnel** page take precedence over **Surface Attributes** which take precedence over **Options** -> **Fresnel Reflect**.

SurfaceFresnel

None

Function:

Deactivates **Fresnel reflection** for a specific dielectric surface

Parameters:

Name of the surface.

Text Commands:

```
modify/surface/fresnel/off surface_name  
modify/surface/metal/off surface_name  
modify/surface/lwirmetal/off surface_name
```

Text Command Example:

```
modify/surface/fresnel/off my_lens.1  
modify/surface/metal/off my_shell.1  
modify/surface/lwirmetal/off my_shell.1
```

Fresnel

Function:

Activates **Fresnel reflection** for a specific dielectric surface

Discussion:

Calculates the transmittance and reflectance for each ray intersection with a dielectric material (glass, plastic) based on the average of the Fresnel reflections for the two polarizations, the indices of refraction (though not the volumetric absorptance) of the materials, and the angle between the ray and the local-surface-normal at the point of intersection.

The user should pay particular attention to the ray termination criteria (see the **RayLimits** section of the **Options** chapter).

See also the global Fresnel reflection feature (Options → Fresnel Reflect) described in the Options chapter.

This feature should be used for reflections at the surface of a dielectric material, such as a lens. The feature will have no effect on a mirror in air.

Note that in order to have a Fresnel reflection the maximum ghost order must be set to at least 1.

Parameters:

Name of the surface.

Text Commands:

```
modify/surface/fresnel/on surface_name  
modify/surface/fresnel/off surface_name
```

Text Command Example:

```
modify/surface/fresnel/on my_lens.1
modify/surface/fresnel/off my_lens.1
```

Metal

Function:

Activates **Fresnel reflection** for a specific **metal** surface

Discussion:

The user should pay particular attention to the ray termination criteria (see the **RayLimits** section of the **Options** chapter).

This surface modification is used to represent a metal surface (generally in air). The indices entered by the user override existing material properties for reflection, transmission, and absorption only; the user-entered complex indices do not effect refraction.

Note that if a coating is applied to the surface it will override this calculation.

Polarization analysis is off:

Calculates the transmittance and reflectance for each ray intersection with a metal surface based on a weighted average of the Fresnel reflections for the two polarizations, the complex indices of refraction of the materials, and the angle between the ray and the local-surface-normal at the point of intersection.

Polarization analysis is on:

Calculates the transmittance and reflectance for each ray intersection with a metal surface based on the user entered complex indices off refraction before and after the surface, the polarization state of the incident light, and on the angle between the ray and the local-surface-normal at the point of intersection.

Note that cons1 defined below is ignored for this calculation.

Parameters:

Surface name,

cons1: fraction of the light that is p-polarized (p_fraction must be between 0 and 1),

cons2: an absorptance flag (absorptance_flag must be either 0 or 1, and is used to place a detector plane behind the surface),

cons3: real part of the index of refraction of the medium before the metal surface (n_index_air is 1 for air),

cons4: imaginary part of the index of refraction of the medium before the metal surface (k_index_air is 0 for air),

cons5: real part of the index of refraction of the metal surface (n_index_metal), and

cons6: imaginary part of the index of refraction of the metal surface (k_index_metal).

The form of the complex index is:

$$\eta = n + i \cdot k$$

where n is the real part of the index ($n_{\text{index_air}}$ or $n_{\text{index_metal}}$) and k is the imaginary part of the index ($n_{\text{index_air}}$ or $n_{\text{index_metal}}$). Note that different reference books use different forms of the complex index and that the user must make the appropriate conversion. As an example the inputs for Aluminum would be $n = 0.7$ and $k = -7.0$.

Text Commands:

```
modify/surf/metal/on name p_fract absorp_flag n_air k_air n_metal k_metal
modify/surface/metal/off name
```

Text Command Example:

```
modify/surface/metal/on my_shell.1 0.5 1 1.0 0.0 0.7 -7.0
modify/surface/metal/off my_shell.1
```

LWIR Metal

Function:

Activates an **angle-dependent reflection** for a specific metal surface

Discussion:

Calculates the transmittance and reflectance for each ray intersection using a metal plasma reflectivity calculation and a weighted average of the angle-dependent reflections for the two polarizations, the values of four constants, and the angle between the ray and the local surface normal at the point of intersection.

The LWIR (Long Wavelength Infrared) reflection function is appropriate for modeling the laser welding of metal at far infrared wavelengths.

The user should pay particular attention to the ray termination criteria (see the **RayLimits** section of the **Options** chapter).

Note the parameters entered for this option are ignored for polarization raytracing, which instead uses the unpolarized reflectance.

Parameters:

Surface name,

cons1: fraction of the light that is p-polarized (p_{fraction} must be between 0 and 1),

cons2: an absorptance flag (absorptance_flag must be either 0 or 1), and

three constants defined below (cons3 , cons4 , and cons5).

The ray reflectance (R) is defined as:

$$R = p_{\text{fraction}} \cdot r_p \cdot (1 - p_{\text{fraction}}) \cdot r_s$$

where

$$r_s = \frac{(1 + (1 - (\varepsilon \cdot q_1))) \cdot (1 - \varepsilon \cdot q_1)}{(1 + (1 + \varepsilon \cdot q_1)) \cdot (1 + \varepsilon \cdot q_1)}$$

and where

$$r_p = \frac{\varepsilon^2 - (2 \cdot \varepsilon \cdot q_1) + 2 \cdot q_1^2}{\varepsilon^2 + 2 \cdot \varepsilon \cdot q_1 + 2 \cdot q_1^2}$$

and where

$$\varepsilon^2 = \frac{2}{1 + \sqrt{1 + \frac{con5^2}{con4^2 \cdot con3^2}}}$$

and q_1 is the dot product between the ray and the local-surface-normal.

(See W. Schultz; "On Laser Cutting of Metals"; J. Phys. D: Appl. Phys. 20, pp481-488, 1987.)

Text Commands:

```
modify/surf/lwir/on name p_fraction absorptance_flag con3 con4 con5  
modify/surface/lwirmetal/off name
```

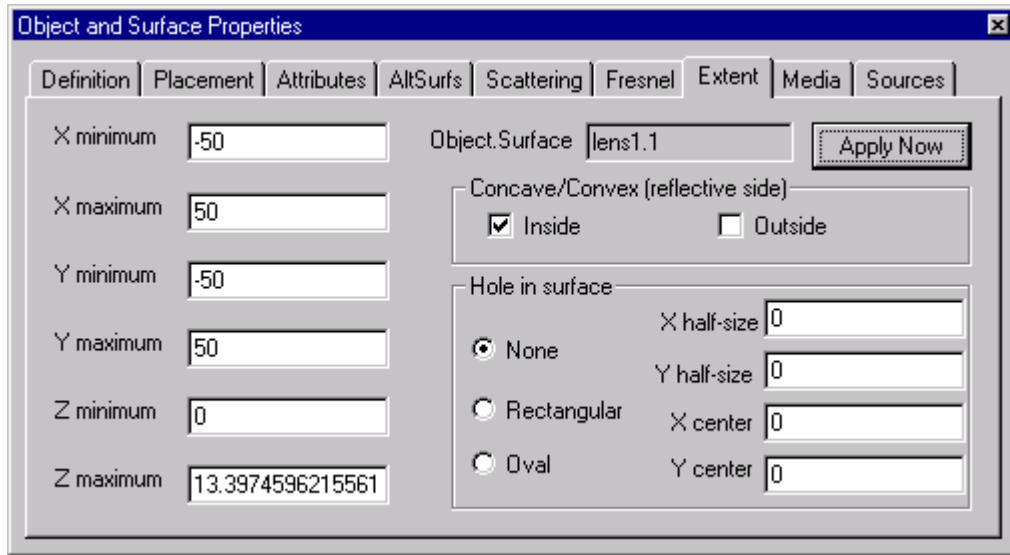
Text Command Example:

```
modify/surface/lwirmetal/on my_shell.1 0.5 1 8.854E-12 1.78E+14 5.0E+5  
modify/surface/lwirmetal/off my_shell.1
```

The Apply Now button only implements the values for this page.

Aborting the object change may be accomplished by closing the box (by clicking the x in upper right of the box).

Extent Page



This page is specific to the surface that was selected using the **OOB** described in a previous section.

The **Extent** page allows for the definition of the physical limits (including a hole) over which a surface exists (is recognized by the ray-trace function).

Extent

Function:

Redefines the limits (extents) of a **surface**

Discussion:

This feature should be used with great care, and the results of using this feature should be carefully checked graphically, by viewing test ray-traces, and by making a rendered image (Solid Model) of the modified object. In general, the object drawing routine will fail before the raytrace routine.

This is one of two methods for defining an off-centered section of a shell object (surface). The feature allows redefinition of the surface(s) that limit the extent of a single-valued surface.

For a mirror of rectangular extent, the extent is determined by an extent box, which consists of two planes parallel to the x-z plane, initially located symmetrically about the origin at +/- x_max; two planes parallel to the y-z plane, initially located symmetrically about the origin at +/- y_max, and two planes parallel to the x-y plane, initially located at z_min and z_max. Modifying the extent shifts these six planes **individually** to a user specified x_min, x_max, y_min, y_max, z_min, and z_max (all in surface-local-coordinates). The surface does not move, and remains centered about its' local x = y = 0.

For a mirror of oval extent, the extent is determined by an ellipse defined in the x-y plane that is projected along the +/-z direction. Modifying the extent changes the x and y sizes and the x and y location of the ellipse.

The value entered for `x_max` must be greater than for `x_min`, the value entered for `y_max` must be greater than for `y_min`, and the value entered for `z_max` must be greater than for `z_min`.

Remember that the surface is still defined in its local coordinate system (even if that part of the surface is not present) and that local rotations of the surface are still about the local coordinate system.

If you wish to rotate the surface about a point on the surface itself, first define an axis (as a reference object), then place the axis on the surface. Perform all rotations of the surface relative to the reference axis.

It is good practice to completely define the surface (including adding aspherics) before modifying its extent.

The extents for a solid object are the extents of each of the objects component surfaces. As an example a spherical single element lens consists of three surfaces, the extent of surface one is defined in a local coordinate system located at the vertex of the surface, the extent of surface two is defined in a local coordinate system located at the vertex of the surface, and the extent of surface three is located in the center of the lens. Do not modify the extents of a solid object.

Note that modifying the extents of a cpc in x and y, and modifying the `zmax` extent beyond the maximum length of the true surface are not reflected in the drawing of the cpc surface.

Note that a hole cannot be placed in a lens and the parameters `Concave` and `Convex` cannot be modified.

Parameters:

The object name, values for `x_min`, `x_max`, `y_min`, `y_max`, `z_min`, and `z_max`.

Text Command:

```
modify/extent object_name x1 y1 z1 x2 y2 z2
```

Text Command Example:

```
modify/extent rectmir -1 -2 -1 1 2 10
```

SurfaceHole

General Discussion:

This feature adds (or modifies) a hole to a surface. The addition (or modification) of a hole is only permitted in single valued 2-D shell objects, such as mirrors, elliptical troughs, and film planes. Holes cannot be placed in lenses. Only one hole is permitted per surface.

None

Function:

Removes a hole

Parameters:

Surface name (`object_name.surface_number`).

Text Command:

```
modify/hole/delete surface_name
```

Text Command Example:

```
modify/hole/delete mirr1.1
```

Oval

Function:

Puts an **oval hole** in a surface

Discussion:

Puts a single oval hole in a single-valued shell surface.

Parameters:

Surface name (**object_name.surface_number**), x and y (surface local) location of the hole, and x and y hole half-sizes.

Text Command:

```
modify/hole/oval surface_name xcenter ycenter xhw yhw
```

Text Command Example:

```
modify/hole/oval mirr1.1 0 0 0.5 0.5
```

Rectangular

Function:

Puts a **rectangular hole** in a surface

Discussion:

Puts a single rectangular hole in a single-valued shell surface.

Parameters:

Surface name (**object_name.surface_number**), x and y (surface local) location of the hole, and x and y hole half-sizes.

Text Command:

```
modify/hole/rectangular surface_name xcenter ycenter xhw yhw
```

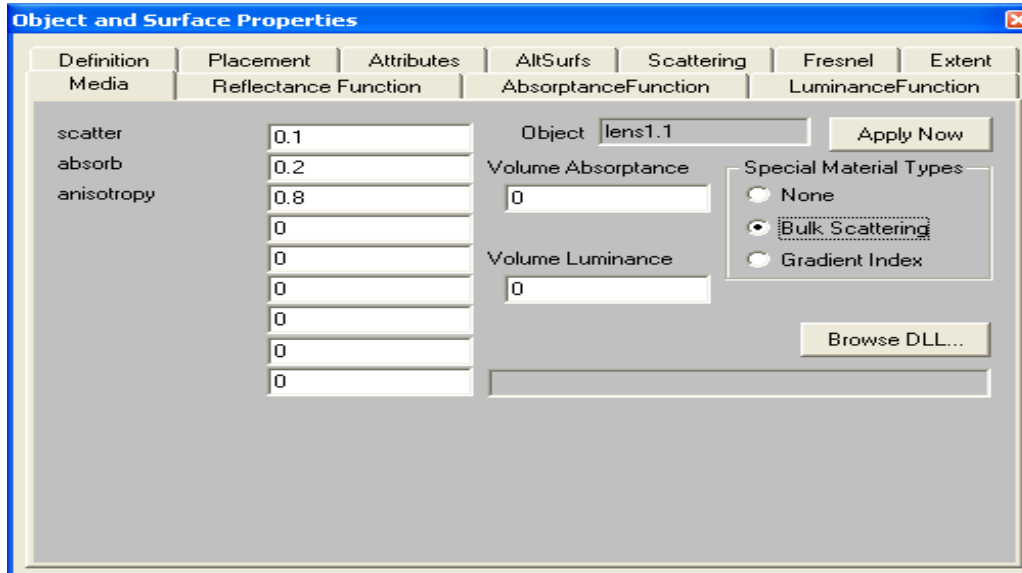
Text Command Example:

```
modify/hole/rectangular mirr2.1 0 0 1 1
```

The Apply Now button only implements the values for this page.

Aborting the object change may be accomplished by closing the box (by clicking the x in upper right of the box).

Object Media Page



This page is specific to the object that was selected using the OOb described in a previous section.

The **Media** page allows for the definition of some volumetric (bulk) properties of a material, including: volumetric absorption, volumetric luminance, gradient index properties, and bulk (volumetric) scattering.

Volume Absorption

Function:

Modifies **Volumetric (bulk) absorptance**

Discussion:

Modifies the volumetric (bulk) absorptance of an object. The expression for the transmittance through a bulk material is:

$$E_{out} = E_{in} \cdot T_1 \cdot T_2 \cdot e^{-\alpha \cdot t}$$

where E_{out} is the ray energy leaving the object, E_{in} is the ray energy entering the object, T_1 is the surface transmittance at the object's entry surface, T_2 is the surface transmittance at the object's exit surface, α is the bulk absorptance coefficient (in Meter^{-1}), and t is the actual ray path (different for each ray) through the object.

Note that the bulk absorptance parameter α is a positive number. OptiCAD will, however, permit you to enter a negative value. A negative value for α will model a volume having exponential gain.

Parameters:

The object name, and the new volumetric (bulk) absorptance.

Text Command:

```
modify/object/absorbativity object_name alpha
```

Text Command Example:

```
modify/object/absorbitivity lens2 0.001
```

Volumetric Luminance

Function:

Modifies **Volumetric luminance**

Discussion:

Modifies the volumetric luminance of an object. The expression for the luminance through a material is

$$P_{out} = \frac{P_{in} \cdot \beta \cdot T_1 \cdot (1 - e^{-\alpha \cdot t})}{\alpha}$$

where P_{out} is the accumulated power after the ray has traversed the object, p_{in} is the accumulated power before the ray has traversed the object, T_1 is the surface transmittance at the object's entry surface, α is the bulk absorptance coefficient (in Meter^{-1}), luminance is the volumetric luminance coefficient (in Meter^{-1}) and t is the actual ray path (different for each ray) through the object.

Parameters:

The object name, the luminance coefficient, and the volumetric (bulk) absorptance.

Text Command:

```
modify/object/absorbitivity object_name luminance
```

Text Command Example:

```
modify/object/absorbitivity lens2 0.001
```

Bulk (Volumetric) Scattering

Function:

Modify **bulk scattering**

Discussion:

Modifies the volumetric (bulk) scattering of an object.

The ray path in a volumetric scattering material is determined by a probabilistic function for both the distance between scattering sites and the change in direction after scattering.

The probability density function for the path between scatterers is given by

$$p(s) = \mu_t \cdot e^{-\mu_t \cdot s}$$

where μ_t is the scattering coefficient (in [local units]⁻¹ and is called scatter in OptiCAD) which is the inverse of the average (or mean free) path between scatterers, and s is the path between scatterers. See Wang, L., S.L. Jacques, "Monte Carlo Modeling of Light Transport in Multi-layered Tissues in Standard C," University of Texas M.D. Anderson Cancer Center, (1992).

The probability density function for the change in ray direction with respect to the z-axis caused by scattering is given by

$$f(N) = \frac{1 - g^2}{2 \cdot (1 + g^2 - (2 \cdot g \cdot N))^{1.5}}$$

where N is defined as

$$N = N_{scatter} - N_{specular}$$

where $N_{specular}$ is the cosine of the unscattered ray with respect to the z-axis, and $N_{scatter}$ is the cosine of the scattered ray with respect to the z-axis, and g is the anisotropy coefficient. The angular scattering function is rotationally symmetric (uniformly distributed) about the z-axis.

The anisotropy coefficient is the average direction cosine of a scattered ray. If the anisotropy has a value of zero the ray is equally likely to scatter in any direction. If the anisotropy coefficient is negative the ray is more likely to scatter backwards than forward. If the anisotropy coefficient is +1 the ray proceeds in the same direction (is not scattered). If the anisotropy coefficient is -1 the ray reverses direction (is retroreflected). See Heney, L.G., and J.L. Greenstein, "Diffuse Radiation in the Galaxy," *Astrophys. J.*, 93, 70-83 (1941).

Bulk scattering should not be used with any object that is closer to its neighbor than the proximity glue distance, should not contain an embedded object, and should not allow total internal reflections.

Parameters:

The object name, scattering coefficient, the volumetric absorptance (defined in the previous section), and the anisotropy coefficient.

Text Command:

```
material/bulk/attach object_name scatter absorb anisotropy
```

Text Command Example:

```
material/bulk/attach object_name 0.1 0.2 0.8
```


Gradient Index Material

Gradient index material is ray-traced using an iterative method. The choice of maximum step size determines the trade-off between speed and accuracy. The optimum step size depends on the numerical aperture of the system and the magnitude of the coefficients. A suitable step size can be determined by starting with a large step size (of approximately the thickness of the grin lens) and decreasing the step size by a factor of two until the results do not significantly change.

Rays traveling through a gradient index material may undergo total internal reflections, partial reflections (splitting into two rays), and surface scattering.

The surfaces defining a grin lens are restricted to a sphere or a conic (an aspheric comprising only a curvature and a conic constant).

Restrictions on the use of gradient index material are that objects may not be placed inside a gradient index material, and a gradient index material may not also be a volume scatterer.

All gradient index materials are ray-traced using a Dynamic Link Library (DLL).

The section **Standard Gradient Index Materials** describes gradient index materials whose DLL ray-trace routines are included on the OptiCAD CD. Users may write custom (DLL) routines for defining gradient index material, this procedure is described in the section **User Defined Gradient Index Materials**.

When setting the ray truncation parameter `maxdepth`, each intermediate step in tracing a ray through a gradient material is counted as a ray-surface intersection. The user should consider setting `maxdepth` to a large number (several thousand) and instead truncate a ray using the `mintrans` parameter.

Note that the step size `maxdist` is specified in local (user selected) units while all shape coefficients used in OptiCAD GRIN materials must be specified in millimeters-based (mm-1, mm-2,...) units, rather than user selected units.

When specifying a gradient index material for a polynet, care must be taken because all equations are calculated in the polynets local coordinate system. The user may wish to define a polynet with one facet in the x-y plane and avoid having any of the polynet shape extending into the negative z region. An extruded polynet may be specified in this manner by setting all of the z values equal to $-t/2$ (where t is the extrusion distance).

The DLLs for the **Standard Gradient Index Materials** do not limit the wavelength range over which an index of refraction is returned. As a general rule glasses designed for the visible region are usable between 350nm and 2000nm, however the user is advised to contact the material manufacturer for specific glass properties.

The relative paths shown in this **Gradient Index Materials** section are relative to the OptiCAD Home directory (the directory containing the OptiCAD executable program).

Standard Gradient Index Material

Gradient index 1 Material

The step size `maxdist` is specified in user selected units and all index spatial profile coefficients must be specified in millimeters-based units.

The gradient index 1 material has no dispersion, the index of refraction is therefore independent of wavelength.

The gradient index 1 material is represented by a single DLL, which must be referenced from within OptiCAD during definition. The gradient index 1 material is described by the following gradient profile:

$$n = n_0 + n_{r2} \cdot r^2$$

Within OptiCAD the user specifies material name (by the choice of the DLL name), the step size and the two terms in the equation above.

The table below shows the name of the DLL.

Material Name	DLL Name
none	g01nodis.dll

Text Commands:

```
material/grin/name="dll_name" object_name maxdist n0 nr2
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g01nodis.dll" g11 1 1.5 -2E-5
material/remove my_grin_lens
```

Gradient index 2 Material

The step size `maxdist` is specified in user selected units and all index spatial profile coefficients must be specified in millimeters-based units.

The gradient index 2 material has no dispersion, the index of refraction is independent of wavelength.

The gradient index 2 material is represented by a single DLL, which must be referenced from within OptiCAD during definition. The gradient index 2 material is described by the following gradient profile:

$$n = \left[n_0 + n_{r2} \cdot r^2 + n_{r4} \cdot r^4 + n_{r6} \cdot r^6 + n_{r8} \cdot r^8 + n_{r10} \cdot r^{10} + n_{r12} \cdot r^{12} \right]^{\frac{1}{2}}$$

Within OptiCAD the user specifies material name (by the choice of the DLL name), the step size and the seven terms in the equation above. Many radial GRIN profiles can be described by this formulation by redefining the coefficients provided by the manufacturer.

The table below shows the name of the DLL.

Material Name	DLL Name
none	g02nodis.dll

Text Commands:

```
material/grin/name="dll_name" name maxdist n0 nr2 nr4 nr6 nr8 nr10 nr12
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g02nodis.dll" gl1 1 1.5 2E-7 1E-5
material/remove my_grin_lens
```

Gradient index 3 Material

The step size `maxdist` is specified in user selected units and all index spatial profile coefficients must be specified in millimeters-based units.

The gradient index 3 material has no dispersion, the index of refraction is independent of wavelength.

The gradient index 3 material is represented by a single DLL, which must be referenced from within OptiCAD during definition. The gradient index 3 material is described by the following gradient profile:

$$n = n_0 + n_{r2} \cdot r^2 + n_{r4} \cdot r^4 + n_{r6} \cdot r^6 + n_{z1} \cdot z + n_{z2} \cdot z^2 + n_{z3} \cdot z^3$$

Within OptiCAD the user specifies material name (by the choice of the DLL name), the step size and the seven terms in the equation above. The table below shows the name of the DLL.

<u>Material Name</u>	<u>DLL Name</u>
none	g03nodis.dll

Text Commands:

```
material/grin/name="dll_name" name maxdist n0 nr2 nr4 nr6 nz1 nz2 nz3
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g03nodis.dll" glens 1 1.5 -2E-5
material/remove my_grin_lens
```

Gradient index 4 Material

The step size `maxdist` is specified in user selected units and all index spatial profile coefficients must be specified in millimeters-based units.

The gradient index 4 material has no dispersion, the index of refraction is independent of wavelength.

The gradient index 4 material is represented by a single DLL, which must be referenced from within OptiCAD during definition. The gradient index 4 material is described by the following gradient profile:

$$n = n_0 + n_{x1} \cdot x + n_{x2} \cdot x^2 + n_{y1} \cdot y + n_{y2} \cdot zy^2 + n_{z1} \cdot z + n_{z2} \cdot z^2$$

Within OptiCAD the user specifies material name (by the choice of the DLL name), the step size and the seven terms in the equation above. The table below shows the name of the DLL.

Material Name	DLL Name
none	g04nodis.dll

Text Commands:

```
material/grin/name="dll_name" name maxdist n0 nx1 nx2 ny1 ny2 nz1 nz2
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g04nodis.dll" glens 1 1.5 0 2E-5
material/remove my_grin_lens
```

Gradient index 5 Material

The step size `maxdist` is specified in user selected units and all index spatial profile coefficients must be specified in millimeters-based units.

Each material within the gradient index 5 family of materials has a separate DLL, which must be referenced from within OptiCAD during definition. The user also enters the constants of the equation below from within OptiCAD during definition.

The constants for the material dispersion calculation are part of the DLL, these constants and the current wavelength are used to calculate the index of refraction at wavelengths other than the reference wavelength. The base index of refraction n_0 entered below is assumed to be measured at the reference wavelength.

The gradient index 5 family of materials model materials with the following gradient profile:

$$n_{ref} = n_0 + n_{r2}r^2 + n_{r4}r^4 + n_{z1}z + n_{z2}z^2 + n_{z3}z^3 + n_{z4}z^4$$

Within OptiCAD the user specifies material name (by the choice of the DLL name), the step size, and the seven coefficients in the equation above.

The table below lists base materials for which a DLL exists, the name of the DLL, and the reference wavelength (microns).

Manufacturer	Base Material Name	DLL Name	Reference Wavelength
LPT	GSF	g05gsf.dll	0.58756
LPT	GK	g05gk.dll	0.58756
NONE	NONE	g05nodis.dll	NONE

The above formula describes the index profile at a single wavelength.

The index for other wavelengths is computed according to the following method based upon a general expansion of the Sellmeier formula:

$$n(\lambda)^2 = n(\lambda_{ref})^2 + \sum_{i=1}^3 \frac{K'_i \cdot [\lambda^2 - \lambda_{ref}^2]}{\lambda^2 - L_i}$$

where

$$K'_i = \sum_{j=1}^{Kmax} K'_{ij} \cdot [n(\lambda_{ref})]^{j-1}$$

and

$$L'_i = \sum_{j=1}^{Lmax} L'_{ij} \cdot [n(\lambda_{ref})]^{j-1}$$

Within the DLL the user specifies the coefficients required for the dispersion equations.

Text Commands:

```
material/grin/name="dll_name" name maxdist n0 nr2 nr4 nz1 nz2 nz3 nz4
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g05gsf.dll" glens 1 1.5 0 1.2E-3
material/grin/name="materials\grin\dlls\g05gk.dll" glens 1 1.5 0 1.2E-3
material/remove my_grin_lens
```

Gradient index 6 Material

The step size `maxdist` is specified in user selected units and all index spatial profile coefficients must be specified in millimeters-based units.

Each material within the gradient index 6 family of materials has a separate DLL, which must be referenced from within OptiCAD during definition. The user also enters the constants of the equation below from within OptiCAD during definition.

The gradient index 6 family of materials model materials with the following gradient profile:

$$n = n_{00} + n_{10}r^2 + n_{20}r^4$$

The constants for the material dispersion calculation are part of the DLL, these constants and the current wavelength are used to calculate the index of refraction.

The n_{00} value is computed from the following formula:

with similar expressions for n_{10} and n_{20} (though the values of A, B, C, and D are different). Note that the calculation based on the manufacturers coefficients use nanometers rather than microns for the wavelength.

$$n_{00} = A + B\lambda^2 + \frac{C}{\lambda^2} + \frac{D}{\lambda^4}$$

The dispersion data was provided by the Gradient Lens Corporation (GLC), Rochester, NY, (716) 235 - 2620.

Within OptiCAD the user specifies material name (by the choice of the DLL name) and the step size.

The table below lists materials for which a DLL exists and the name of the DLL.

Manufacturer	Material	DLL Name
GLC	ARS10	g06ars10.dll
GLC	ARS20	g06ars20.dll
GLC	ARS27	g06ars27.dll
GLC	ARS31	g06ars31.dll

Text Commands:

```
material/grin/name="dll_name" object_name maxdist
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g06ars10.dll" my_grin_lens 1
material/grin/name="materials\grin\dlls\g06ars12.dll" my_grin_lens 1
material/grin/name="materials\grin\dlls\g06ars27.dll" my_grin_lens 1
material/grin/name="materials\grin\dlls\g06ars31.dll" my_grin_lens 1
material/remove my_grin_lens 1.0
```

Gradient index 7 Material

The step size `maxdist` is specified in user selected units and all index spatial profile coefficients must be specified in millimeters-based units.

The gradient index 7 material has a spherical index profile and no dispersion, the index of refraction is therefore independent of wavelength.

The gradient index 7 material is represented by a single DLL, which must be referenced from within OptiCAD during definition. The gradient index 7 material is described by the following gradient profile:

$$n = n_0 + \alpha \cdot (r - R) + \beta \cdot (r - R)^2$$

where

$$r = \frac{R}{|R|} \sqrt{x^2 + y^2 + (R - z)^2}$$

The coordinates x, y, and z measured with respect to the vertex tangent plane, and R is the radius of the isoindex contours measured at the vertex. The isoindex contours are spherical shells centered about the point z = R. The starting index n₀, is measured at the vertex of the surface, not at the center of the isoindex contour.

Within OptiCAD the user specifies material name (by the choice of the DLL name), the step size and the four terms in the equations above.

The isoindex contour radius R is specified independently of the front or rear radii of the lens, R must be specified (zero is not defaulted to a plane shape).

The table below shows the name of the DLL.

Material Name	DLL Name
none	g07nodis.dll

Text Commands:

```
material/grin/name="dll_name" object_name maxdist n0 R α β
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g07nodis.dll" g11 1 1.5 1 -2.8 -1
material/remove my_grin_lens
```

Gradient index 8 Material (GRADIUM)

The step size `maxdist` is specified in user selected units and all index spatial profile coefficients must be specified in millimeters-based units.

Each material within the gradient index 8 family of materials has a separate DLL, which must be referenced from within OptiCAD during definition. The constants for the material dispersion calculation are part of the DLL, these constants and the current wavelength are used to calculate the index of refraction at wavelengths other than the reference wavelength.

This material type approximates stock gradient index blanks available from Lightpath Technologies Incorporated. The blanks have an axial gradient index profile, which describes a reference index of refraction as a function of axial position within the glass blank. All that is required to define a material is the starting point within the blank and the name of the stock blank profile.

The gradient index 8 family of materials is described by the following gradient profile:

$$n_{ref} = \sum_{i=0}^{11} n_i \cdot \left[\frac{z + \Delta z}{z_{max}} \right]^i$$

The z coordinate is the distance from the front vertex of the blank, z_{max} is the maximum z coordinate of the blank, and Δz is the offset distance along the profile.

Within OptiCAD the user specifies material name (by the choice of the DLL name), the step size and the Δz term in the equation above.

The table below lists base materials for which a DLL exists, the name of the DLL, and the reference wavelength (microns).

<u>Manufacturer</u>	<u>Material Name</u>	<u>Base Material Name</u>	<u>DLL Name</u>	<u>Reference Wavelength</u>
LPT	G1SFN	GSF	g08g1sfn.dll	0.58756
LPT	G1SFP	GSF	g08g1sfp.dll	0.58756
LPT	G2SFN	GSF	g08g2sfn.dll	0.58756
LPT	G2SFP	GSF	g08g2sfp.dll	0.58756
LPT	G4SFN	GSF	g08g4sfn.dll	0.58756
LPT	G4SFP	GSF	g08g4sfp.dll	0.58756
LPT	G5SFN	GSF	g08g5sfn.dll	0.58756
LPT	G5SFP	GSF	g08g5sfp.dll	0.58756

The above formula describes the index profile at the reference wavelength. The index for other wavelengths is computed according to the Buchdahl dispersion model defined in the **Gradient 5** section.

Text Commands:

```
material/grin/name="dll_name" object_name maxdist ΔZ
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g08g1sfn.dl" my_grin_lens 1 0
material/grin/name="materials\grin\dlls\g08g1sfp.dl" my_grin_lens 1 0
material/grin/name="materials\grin\dlls\g08g2sfn.dl" my_grin_lens 1 0
material/grin/name="materials\grin\dlls\g08g2sfp.dl" my_grin_lens 1 0
material/grin/name="materials\grin\dlls\g08g4sfn.dl" my_grin_lens 1 0
material/grin/name="materials\grin\dlls\g08g4sfp.dl" my_grin_lens 1 0
material/grin/name="materials\grin\dlls\g08g5sfn.dl" my_grin_lens 1 0
material/grin/name="materials\grin\dlls\g08g5sfp.dl" my_grin_lens 1 0
material/remove my_grin_lens
```

Gradient index 9 Material (SELFOC)

The gradient index 9 material is comparable to models **SELFOC** materials from NSG America Inc. (division of Nippon Sheet Glass) using coefficients supplied by the manufacturer.

The step size `maxdist` is specified in user selected units.

The gradient index 9 family of materials model materials with the following gradient profile:

$$n(\lambda) = n_0 + \left[1.0 - \left(\frac{Ar^2}{2} \right) \right]$$

where

$$A(\lambda) = \left[k_0 + \frac{k_1}{\lambda^2} + \frac{k_2}{\lambda^4} \right]^2$$

and

$$n_0 = B + \frac{C}{\lambda^2}$$

Within OptiCAD the user specifies material name (by the choice of the DLL name) and the step size.

The table below lists base materials for which a DLL exists, and the name of the DLL.

Manufacturer	Material Name	Function Name
NSG	SLH-1.8	g09slh18.dll
NSG	SLS-1.0	g09sls10.dll
NSG	SLS-2.0	g09sls20.dll
NSG	SLW-1.0	g09slw10.dll
NSG	SLW-1.8	g09slw18.dll
NSG	SLW-2.0	g09slw20.dll
NSG	SLW-3.0	g09slw30.dll
NSG	SLW-4.0	g09slw40.dll

Text Commands:

```
material/grin/name="dll_name" object_name maxdist  
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g09slh18.dll" my_grin_lens 1  
material/grin/name="materials\grin\dlls\g09sls10.dll" my_grin_lens 1  
material/grin/name="materials\grin\dlls\g09sls20.dll" my_grin_lens 1  
material/grin/name="materials\grin\dlls\g09slw10.dll" my_grin_lens 1  
material/grin/name="materials\grin\dlls\g09slw18.dll" my_grin_lens 1  
material/grin/name="materials\grin\dlls\g09slw20.dll" my_grin_lens 1  
material/grin/name="materials\grin\dlls\g09slw30.dll" my_grin_lens 1  
material/grin/name="materials\grin\dlls\g09slw40.dll" my_grin_lens 1  
material/remove my_grin_lens
```

Gradient index 10 Material

The step size `maxdist` is specified in user selected units and all index spatial profile coefficients must be specified in millimeters-based units.

Each material within the gradient index 10 family of materials has a separate DLL, which must be referenced from within OptiCAD during definition. The user also enters the constants of the equation below from within OptiCAD during definition.

The constants for the material dispersion calculation are part of the DLL, these constants and the current wavelength are used to calculate the index of refraction at wavelengths other than the reference wavelength. The base index of refraction n_0 below is assumed to be measured at the reference wavelength.

The gradient index 10 family of materials model materials with the following gradient profile:

$$n_{ref} = n_0 + n_{y1}|y| + n_{y2}|y^2| + n_{y3}|y^3| + n_{y4}|y^4| + n_{y5}|y^5| + n_{y6}|y^6|$$

where the || symbols indicate an absolute value. This form of the gradient has a discontinuity at the plane $y = 0$, and the gradient is bisymmetric about the $y = 0$ plane.

Within OptiCAD the user specifies material name (by the choice of the DLL name), the step size and the seven coefficients in the equation above.

The table below lists base materials for which a DLL exists, the name of the DLL, and the reference wavelength (microns)

Manufacturer	Base Material Name	DLL Name	Reference Wavelength
LPT	GSF	g10gsf.dll	0.58756
LPT	GK	g10gk.dll	0.58756

The above formula describes the index profile at the reference wavelength. The index for other wavelengths is computed according to the Buchdahl dispersion model defined in the **Gradient 5** section.

Text Commands:

```
material/grin/name="dll_name" name maxdist n0 ny1 ny2 ny3 ny4 ny5 ny6
material/remove object_name
```

Text Command Examples:

```
material/grin/name="materials\grin\dlls\g10gsf.dll" g11 1 1.5 0 1E-3 0 4E-8
material/grin/name="materials\grin\dlls\g10gk.dll" g12 1 1.5 0 1E-3 0 4E-8
material/remove my_grin_lens
```

User Defined Gradient Index Material

Making a user defined DLL

All gradient index materials are ray-traced using a Dynamic Link Library (DLL). The user may write a DLL in order to analyze a gradient index material which cannot be modeled using one of the existing DLLs included with OptiCAD.

Note that Opticad Corporation does not support or debug user written software.

In order to write a DLL you must have access to a suitable compiler or development tool that can generate 32 bit Windows Compatible DLLs. The basic procedure for making a DLL using Microsoft Visual C++ version 5.0 is described in this section.

1. Create a subdirectory in the directory that contains your C compiler (such as `c:\msdev\my_mat`) with the name of the material that you wish to model (restrict this to 8 characters).

2. Examine one of the example DLL source files (having the extension c) and copy the one that is closest to your application into your new subdirectory, also copy the definition file (of the same name but having the extension def) into your new subdirectory. Note that source files for all grin materials may not be present.
3. Rename both of these files to have the same name as your new subdirectory, but keep the extension unchanged (such as `my_mat.c` and `my_mat.def`).
4. Start the C++ program, then open one of the files having the extension mdp, and then close the file.
5. Use one of the project files (ending in mdp) included with this release of OptiCAD as a template, and create a new project with the same name as your material.
6. Select New from File menu, then select Project, and then select the generic DLL option, do not select MFC or Microsoft Foundation Class Libraries type DLL, and enter the name with the full path of your new subdirectory (such as `c:\msdev\my_mat`).
7. Insert the renamed files into your new project.
8. Now make all of the changes that are required in your source program, and select Build and Build `my_mat.dll` (or the name of your project the extension dll appended) in order to create a DLL file.
9. You can access the DLL from OptiCAD by inserting its name (with the full path in double quotes if needed) on the grin materials definition line (such as `material/grin/name="my_mat.dll" my_grin_lens`).

DLL Data Array

The user defined DLL uses a single data array (named data) to interact with OptiCAD. The table below describes the contents of that array.

<u>Array Element</u>	<u>Received from OptiCAD</u>	<u>Sent to OptiCAD</u>	<u>Description</u>
data[0]	29		maximum array size
data[1]	x		x coordinate
data[2]	y		y coordinate
data[3]	z		z coordinate
data[4]	lambda		wavelength (in microns)
data[5]	units factor		mm per length unit
data[6]		n	index of refraction
data[7]		n * dn/dx	index times dn/dx
data[8]		n * dn/dy	index times dn/dy

<u>Array Element</u>	<u>Received from OptiCAD</u>	<u>Sent to OptiCAD</u>	<u>Description</u>
data[9]		$n * dn/dz$	index times dn/dz
data[10]	p01		user defined constant
data[11]	p02		user defined constant
data[12]	p03		user defined constant
data[13]	p04		user defined constant
data[14]	p05		user defined constant
data[15]	p06		user defined constant
data[16]	p07		user defined constant
.....			user defined constant
.....			user defined constant
data[29]	p20		user defined constant

Data Sent by OptiCAD to the DLL

The element data[0] is the maximum usable address of the data array. The elements data[1], data[2], and data[3] are the x, y, and z coordinates of the current ray location, and the point at which OptiCAD needs to know the index of refraction and its spatial derivatives. The element data[4] is the wavelength in microns. The element data[5] is a scale factor equal to millimeters per length unit, this tells the DLL what units OptiCAD is using, and is equal to: 1.0 for mm, 25.4 for inches, 10.0 for cm, and 1000.0 for meters.

User entered data passed (unchecked) by OptiCAD to the DLL

The command line in OptiCAD that defines the material also allows the user to pass data to the named DLL. OptiCAD passes the second through 20th user entered parameter (named p01 through p20) to the DLL as array elements data[10] through data[29].

Note that OptiCAD does not check this data in any way, it is the sole responsibility of the user to only enter reasonable values for items p01 through p20, and to thoroughly check data inside any user defined DLL.

DLL calculated data that MUST be sent back to OptiCAD by the DLL

The element data[6] is the index of refraction (n). The elements data[7], data[8], and data[9] are $n * dn/dx$, $n * dn/dy$, and $n * dn/dz$; note that the user must return the index of refraction times the spatial derivatives of the index of refraction.

Example of user entered data for the DLL g10gsf.dll (Gradient 10 Material)

See a description of the gradient material 10 equation and definition command line in the section named `Gradient Index 10 Material`.

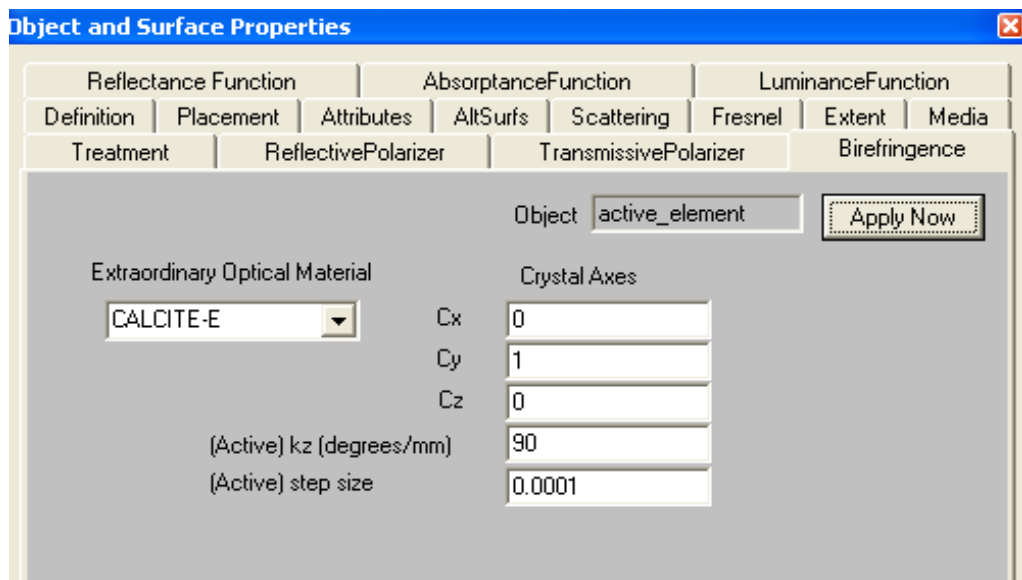
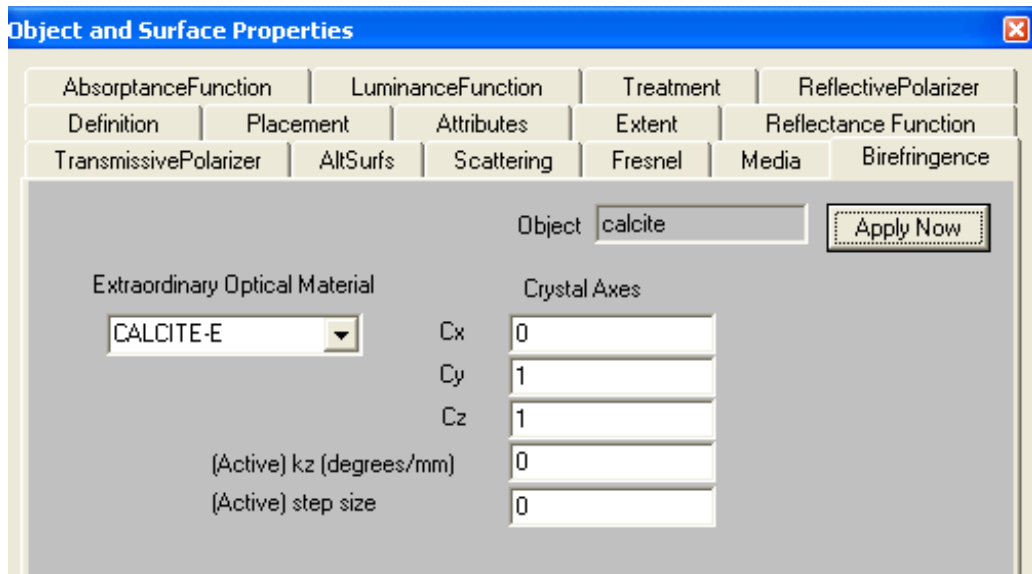
```

data[10] = n0
data[11] = ny1
data[12] = ny2
data[13] = ny3
data[14] = ny4
data[15] = ny5
data[16] = ny6

```

The Apply Now button only implements the values for this page. Aborting the object change may be accomplished by closing the box (by clicking the x in upper right of the box).

Object Birefringence Page



This page is specific to the object that was selected using the OOb described in a previous section.

The **Birefringence** page allows for the definition of a passive or active birefringent material.

Function:

Defines the **extraordinary glass, crystal axes**, and optionally the **z active material coefficient** and **step size**,

Discussion:

Modifies the material to have passive or active birefringence. If either the active birefringent coefficient or the active step size is zero, the material will have passive birefringence properties.

The active birefringence coefficient k_z rotates the x and y crystal axes about the z crystal axis as a linear function of the z location in the material. Position in the material is calculated from the vertex of surface one.

Note that the ordinary glass is selected from the **Definition** page.

Note that birefringent material may only have transmittance and absorptance (reflectance is ignored). A reflective surface may be modeled by placing (gluing) a surface (such as a transmitting mirror) near the material surface.

Parameters:

The object name, extraordinary glass, and the crystal axis vector (Cx, Cy, Cz). Optionally, nonzero values for the active z coefficient (degrees/mm) and active step size (system units).

Passive Birefringence Text Command:

```
material/extra_glass/name object_name cx cy cz 0 0
```

Active Birefringence Text Command:

```
material/extra_glass/name object_name cx cy cz kz step_size
```

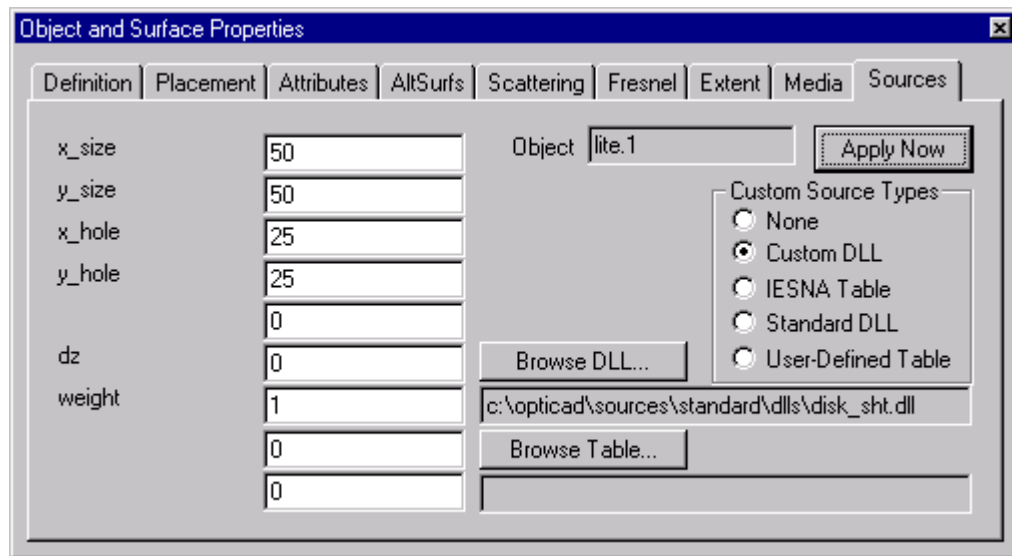
Passive Text Command Example:

```
material/extra_glass/name=CALCITE-E calcite 0 0 1 0 0
```

Active Text Command Example:

```
material/extra_glass/name=USERPOLYMER-E activematerial 0 1 0 90 0.0001
```

(Extended) Source-Detectors Page



This page is specific to the lightsource or detector object that was selected using the **OOB** described in a previous section.

The **Sources and Source-Detectors** page allows for the definition of a custom lightsource or custom detector for which the spatial and angular properties of the emitted rays are specified.

Custom/Extended Lightsources and Detectors

A custom source–detector is modeled as a virtual region (surface, volume, or point) specifying where rays originate from and in what direction they are launched. Both origination location and initial direction are determined by weighted statistical distributions. While most custom source-detectors are spatially extended, this category also includes point sources with special angular properties.

Rays will not be intercepted, absorbed, scattered, or reflected by a custom source–detector. Therefore when modeling a system in which rays are likely to interact with a source-detector, the user should design a conventional object (with appropriate reflective, absorbing, and scattering properties) and place it near the virtual source-detector. An example of such a combination of real and emitting surfaces would be a cylindrical fluorescent tube, modeled as an outwardly Lambertian emitting cylinder with conventional diffuse reflecting (slightly smaller) cylinder inside of the emitting surface.

All custom source–detectors are ray-traced using a Dynamic Link Library (DLL).

All custom source–detectors are ray–traced using the Analyze -> Raytrace -> Extended (**ray/custom**) feature. The ray–trace feature allows the ray bundle to be limited (truncated) in both x and y angles specified in the source’s local coordinate system.

The section **Standard Custom Source–Detectors** describes source–detectors whose DLL routines are included on the OptiCAD CD. DLLs for custom lightsources and custom detectors are identical. The section **External Data Sources** covers lightsources that are described by industry standard (IESNA) tabulated data, these are not appropriate for use with detectors. Users may write custom (DLL) routines for defining Custom Sources, this procedure is described in the section **User Defined Custom Source–Detectors**.

Some source-detectors have an axial shift(s) along the z-axis (`z_shift`) as a user specified parameter. A negative value specifies the apparent (virtual) location of the source, such as for a laser diode with astigmatism (different virtual line source locations in x and y). A positive value specifies the distance along the z-axis at which the rays will come to a focus, this allows the definition of a converging bundle of rays. The `z_shift` is not appropriate for detectors

Note that the parameter `weight` is used to weight each ray and that a value of 0 is interpreted as 1.0 (the default ray weight).

The relative paths shown in this **Custom and Extended Source–Detectors** section are relative to the OptiCAD Home directory (the directory containing the OptiCAD executable program).

Note that a custom (extended) source-detector is not drawn as an object, only the rays originating from the source are drawn.

Lambertian Distribution

Sources that emit into a Lambertian distribution obey Lamberts' cosine law of emission. The probability density function for this distribution is uniform in the region $L^2 + M^2 < 1.0$ and 0.0 elsewhere, where L is the cosine with the x-axis, M is the cosine with the y-axis, both L and M are defined in the local coordinate system of the surface at the point of emission.

Lightsources versus Detectors

The definition of a custom lightsource (used for forward raytracing) and a detector (use for reverse ray–tracing) are identical. However not all DLLs are appropriate for use as a radiometric detector. When a DLL is considered suitable for use only as a source, it will be described a source. When a DLL is considered appropriate as either a lightsource or a detector, it will be described as a source–detector.

Detector Area

For extended source-detectors the area of the detector is calculated in the DLL (based on the user input dimensions) and used in the radiometric calculation.

Predefined Extended Source–Detector

The ray weight is the seventh numerical input parameter. In order to enter a value for a numerical parameter you must enter values (even if they are zeros) for all parameters up to the desired parameter. As an example the isotropic point source (shown below) has only two numerical inputs (parameters six and seven), values for these parameters are entered after five (placeholder) zeros. Placeholder values (not used by the program) are shown in the text command description as a star (*), and in the text command examples as a zero.

Truncating a string of numerical input parameters causes the (ray) weight to have a value of 1.0, and all other missing input parameters to have a value of 0.0.

Cylindrical Emitting Surface Source–Detector

There are two types of cylindrical surface source–detectors because of the ray bundle limits (in x and y) available with the Analyze -> Raytrace -> Extended (*ray/custom*) feature.

The z cylinder emitting surface source–detector is a virtual cylinder extending into the z-axis whose uniformly emitting surface radiates outward into a Lambertian distribution from any point on the surface.

The x cylinder emitting surface source–detector is a virtual cylinder extending into the x-axis whose uniformly emitting surface radiates outward into a Lambertian distribution from any point on the surface.

A cylindrical emitting surface source–detector is represented by a single DLL, which must be referenced from within OptiCAD during definition.

Within OptiCAD the user specifies source–detector type name (by the choice of the DLL name), the radius, the half length of the cylinder, and the ray weight.

The table below shows the name of the DLL.

<u>Source–Detector Type Name</u>	<u>DLL Name</u>
Z Cylinder Surface	CylZ_sur.dll
X Cylinder Surface	CylX_sur.dll

Text Commands:

```
modify/source/custom/name="dll_name" name radius half_length * * * * weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\CylZ_sur.dll" 11 1 3 0 0 0  
0 100
```

```
modify/source/custom/name="sources\standard\dlls\CylX_sur.dll" 12 1 3 0 0 0  
0 100
```

```
modify/source/custom/name="sources\standard\dlls\CylZ_sur.dll" 11 1 3
```

```
modify/source/custom/name="sources\standard\dlls\CylX_sur.dll" 12 1 3
```

Flat Emitting Surface Source–Detector

There are two types of flat surface source–detectors, one is a sheet of rectangular extent and one is a sheet of oval extent.

The rectangular flat emitting surface source–detector is a virtual sheet located in the x-y plane of rectangular extent whose uniformly emitting surface radiates outward into the same angular distribution from any point on the surface.

The Lambertian source–detector now allows a shape factor to be specified, with the default value defining a Lambertian distribution.

The rotationally symmetric (about z) angular distribution is defined as

$$I(\theta) = I_0 \cdot [\cos(\theta)]^{n+1}$$

For n=0 (the default value) the distribution is Lambertian, for n=-1 the distribution is isotropic, as n increases the distribution becomes narrower about the +z axis, n must be greater than -1.75.

For the rectangular source-detector the user specifies source-detector type name (by the choice of the DLL name), the x and y half sizes of the sheet, the shape factor n, the axial shift in z, and the ray weight.

The oval flat emitting surface source-detector is a virtual sheet located in the x-y plane of oval extent whose uniformly emitting surface radiates outward into the same angular distribution from any point on the surface.

For the oval source-detector the user specifies source-detector type name (by the choice of the DLL name), the x and y half sizes of the sheet, the x and y half sizes for an oval hole, the shape factor n, the axial shift in z, and the ray weight.

A flat emitting surface source-detector is represented by a single DLL, which must be referenced from within OptiCAD during definition.

The table below shows the name of the DLL.

Source-Detector Type Name	DLL Name
Rectangular Surface	Rect_Sht.dll
Oval Surface	Disk_Sht.dll

Text Commands:

```
modify/source/custom/name="Rect_Sht.dll" name x_size y_size * * n dz weight
modify/source/custom/name="Disk_Sht.dll" name x_size y_size x_hole y_hole n
dz * weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\Rect_Sht.dll" 11 50 50
modify/source/custom/name="sources\standard\dlls\Disk_Sht.dll" 12 50 50
modify/source/custom/name="sources\standard\dlls\Disk_Sht.dll" 12 50 50 25 25
modify/source/custom/name="sources\standard\dlls\Rect_Sht.dll" 11 50 50 0 0
0 0 100
modify/source/custom/name="sources\standard\dlls\Disk_Sht.dll" 12 50 50 25 25
1 0 100
```

Gaussian Surface Source

The Gaussian source has a rotationally symmetric Gaussian spatial profile specified by the beam waist parameter ω_0 , the angular divergence (Φ_0 which is related to the beam by the equation below.

$$\tan(\Phi_0) = \frac{\lambda}{\Pi \cdot \omega_0}$$

The probability density function for the spatial ray distribution has the form

$$\exp\left(\frac{-2 \cdot [r]^2}{[\omega_0]^2}\right)$$

where $r^2 = x^2 + y^2$.

The probability density function for the angular ray distribution has the form

$$\exp\left(\frac{-2 \cdot [\tan(\Phi)]^2}{[\tan(\Phi_0)]^2}\right)$$

For the Gaussian source the user specifies source type name (by the choice of the DLL name), the x and y maximum half sizes (in mm) of the source (a clipping function not the gaussian width), the gaussian half width ω_0 (in mm), an axial shift in z (in mm), and the ray weight (a zero or blank is defaulted to 1.0).

Note that the x and y half sizes do not change the gaussian beam waist, but rather limit the maximum physical extent of the (otherwise infinite) gaussian beam.

The table below shows the name of the DLL.

Source Type Name	DLL Name
Gaussian	gaussian.dll

Detector Note:

This source DLL is probably not appropriate for a detector having an attached camera.

Text Commands:

```
modify/source/custom/name="dll_name" name x_max y_max omega_0 dz * * weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\gaussian.dll" 11 1 1 0.1
modify/source/custom/name="sources\standard\dlls\gaussian.dll" 11 1 1 0.1 0
0 0 100
```

Multimode Gaussian Surface Source

This source is identical to the **Gaussian Surface Source** (described above), except that both the beam waist parameter ω_0 and the angular divergence Φ_0 are explicitly specified.

For the Gaussian source the user specifies source type name (by the choice of the DLL name), the x and y maximum half sizes (in mm) of the source (a clipping function not the gaussian width), the (far field) gaussian half width ω_0 (in mm), the angular (far field) half width divergence Φ_0 (in degrees), an axial shift in z (in mm), and the ray weight (a zero or blank is defaulted to 1.0).

The table below shows the name of the DLL.

Source Type Name	DLL Name
Multimode Gaussian	mm_gaussian.dll

Detector Note:

This source DLL is probably not appropriate for a detector having an attached camera.

Text Commands:

```
modify/source/custom/name="dll_name" name x_max y_max omega_0 phi_0 dz *
weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\mm_gaussian.dll" l1 100 100
25 30
modify/source/custom/name="sources\standard\dlls\mm_gaussian.dll" l1 100 100
25 30 0 100
```

Laser Diode Source

A laser diode source is modeled as a beam having a different divergence in the x-direction than in the y-direction, and as having rays that could originate from different origins in the x-direction than in the y-direction (astigmatism).

The beam divergence is modeled as the product of two separable gaussian divergences, one in x and another in y. The ray pattern has a 2-fold rectangular symmetry.

Note that only one diode data set may be associated with any lightsource for a single raytrace. In order to raytrace from multiple different diode sources, first modify a source for the first set of diode parameters, then trace rays from that source, then use the add_rays option to modify a source for the second set of diode parameters, then trace rays from that source.

The probability density function for the angular ray distribution (in x and y) has the form

$$\exp\left(-\left(\frac{[\tan(\theta_x)]^2}{2 \cdot \sigma_x^2} + \frac{[\tan(\theta_y)]^2}{2 \cdot \sigma_y^2}\right)\right)$$

where

$$\left(\sigma_x = \frac{\theta_{0x}}{2}\right), \left(\sigma_y = \frac{\theta_{0y}}{2}\right)$$

and θ_{0x} and θ_{0y} are the $1/e^2$ divergence angles in x and y.

Within OptiCAD the user specifies source type name (by the choice of the DLL name), the half sizes in x and y, the x and y 1-D standard deviations (in degrees) of the gaussian (far field) half width divergences σ_x and σ_y , the axial shifts along the z-axis in the apparent ray origins in x and in y from the emitting plane (diode astigmatism), and the ray weight.

The table below shows the name of the DLL.

Source Type Name	DLL Name
Laser Diode	Diode.dll

Detector Note:

This source DLL is probably not appropriate for a detector having an attached camera.

Text Commands:

```
modify/source/custom/name="dll_name" name x_size y_size sigma_x sigma_y dz_x
dz_y weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\diode.dll" 11 5 25 0.1 1
modify/source/custom/name="sources\standard\dlls\diode.dll" 11 5 25 0.1 1 0
-10 100
```

Laser Diode2 Source

A laser diode source is modeled as a beam having a different divergence in the x-direction than in the y-direction, and as having rays that could originate from different origins in the x-direction than in the y-direction (astigmatism).

The beam divergence is modeled as the product of two separable gaussian divergences, one in x and another in y. The ray pattern has a 2-fold rectangular symmetry.

Note that only one diode data set may be associated with any lightsource for a single raytrace. In order to raytrace from multiple different diode sources, first modify a source for the first set of diode parameters, then trace rays from that source, then use the add_rays option to modify a source for the second set of diode parameters, then trace rays from that source.

The probability density function for the angular ray distribution (in x and y) has the form

$$\exp\left(-\left(\frac{2 \cdot [\tan(\theta_x)]^2}{[\tan(\theta_{0x})]^2} + \frac{2 \cdot [\tan(\theta_y)]^2}{[\tan(\theta_{0y})]^2}\right)\right)$$

where θ_{0x} and θ_{0y} are the $1/e^2$ divergence angles in x and y.

Within OptiCAD the user specifies source type name (by the choice of the DLL name), the half sizes in x and y, the x and y (far field) half width divergence angles θ_{0x} and θ_{0y} , the axial shifts along the z-axis in the apparent ray origins in x and in y from the emitting plane (diode astigmatism), and the ray weight.

The table below shows the name of the DLL.

Source Type Name	DLL Name
Laser Diode2	Diode2.dll

Detector Note:

This source DLL is probably not appropriate for a detector having an attached camera.

Text Commands:

```
modify/source/custom/name="dll_name" name x_size y_size theta_0_x theta_0_y
dz_x dz_y weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\diode2.dll" 11 5 25 0.1 1
modify/source/custom/name="sources\standard\dlls\diode2.dll" 11 5 25 0.1 1 0
-10 100
```

Diode Linear Array Source

This source models a linear array (spaced along the x-axis) of Diode2 Sources.

Within OptiCAD the user specifies source type name (by the choice of the DLL name), the number of diodes in x, the full sizes in x and y, the center-to-center diode spacing in x, the x and y (far field) half width divergence angles θ_{0x} and θ_{0y} , and the ray weight.

The table below shows the name of the DLL.

Source Type Name	DLL Name
Linear Array of Diodes	ld_array.dll

Detector Note:

This source DLL is probably not appropriate for a detector having an attached camera.

Text Commands:

```
modify/source/custom/name="dll_name" name n_x x_size y_size spacing_x
theta_0_x theta_0_y weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\ld_array.dll" lite 4 2 10 8
30 90
modify/source/custom/name="sources\standard\dlls\ld_array.dll" lite 4 2 10 8
30 90 1
```

Cylindrical Emitting Volume Source-Detector

The cylindrical emitting volume source-detector is a virtual cylinder extending into the z-axis whose uniformly emitting volume radiates isotropically from any point within its volume.

A cylindrical emitting volume source-detector is represented by a single DLL, which must be referenced from within OptiCAD during definition.

Within OptiCAD the user specifies source–detector type name (by the choice of the DLL name), the radius, half length in z for the cylinder, and the ray weight.

The table below shows the name of the DLL.

Source–Detector Type Name	DLL Name
Cylinder Volume	Cyl_Vol.dll

Text Commands:

```
modify/source/custom/name="dll_name" name radius half_length * * * * weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\cyl_vol.dll" lite 25 50
modify/source/custom/name="sources\standard\dlls\cyl_vol.dll" lite 25 50 0 0
0 0 100
```

Ellipsoidal Emitting Volume Source–Detector

The ellipsoidal emitting volume source–detector is a virtual ellipsoid whose uniformly emitting volume radiates isotropically from any point within its volume.

An ellipsoidal emitting volume source–detector is represented by a single DLL, which must be referenced from within OptiCAD during definition.

Within OptiCAD the user specifies source–detector type name (by the choice of the DLL name), the half sizes in x, y, and z of the ellipsoid, and the ray weight (a zero or blank is defaulted to 1.0).

The table below shows the name of the DLL.

Source–Detector Type Name	DLL Name
Ellipsoidal Volume	Ell_Vol.dll

Text Commands:

```
modify/source/custom/name="dll_name" name x_size y_size z_size * * * weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\Ell_vol.dll" lite 25 25 50
modify/source/custom/name="sources\standard\dlls\Ell_vol.dll" lite 25 25 50
0 0 0 100
```

Isotropic Emitting Point Source–Detector

The isotropic emitting point source–detector is a point light source–detector that is uniformly emitting into a full sphere.

The isotropic point source–detector (for the default m=0) is similar in function to the **equal solid angle** point source–detector option, but has several improvements. The isotropic point source–detector performs Monte Carlo ray–tracing, emits rays uniformly into a sphere, has no structured ray grid pattern imposed on the distribution, and can be limited in angle space as either a rectangular or oval distribution.

The isotropic point source–detector now allows a shape factor to be specified, with the default value defining an isotropic distribution.

The rotationally symmetric (about z) angular distribution is defined as

$$I(\theta) = I_0 \cdot [\cos(\theta)]^m$$

For m=0 (the default value) the distribution is isotropic, for m=1 the distribution is Lambertian, as m increases the distribution becomes narrower about the +z axis, m must be greater than --.75.

An isotropic point source–detector is represented by a single DLL, which must be referenced from within OptiCAD during definition.

Within OptiCAD the user specifies source–detector type name (by the choice of the DLL name), the axial shift in z, and the ray weight.

The table below shows the name of the DLL.

Source–Detector type name	DLL Name
Isotropic Point	Pt_Iso.dll

Text Commands:

```
modify/source/custom/name="dll_name" object_name * * * * m dz weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\pt_iso.dll" lite
modify/source/custom/name="sources\standard\dlls\pt_iso.dll" lite 0 0 0 0 0
-10 100
modify/source/custom/name="sources\standard\dlls\pt_iso.dll" lite 0 0 0 0 1
-10 100
```

Toroidal Surface Source

The toroidal source emits rays from the surface of a virtual toroid (doughnut), and into a Lambertian angular distribution. A toroid surface is formed by the rotation of a circle of radius little_r located in the x-y plane, about a line parallel to the x-axis located a distance big_R away from the center of the circle.

For the toroidal source to correctly represent a physical toroidal emitting surface, the user must place a real toroidal surface inside the virtual surface of the source. This is necessary so that rays from inside of the toroidal source are reflected at a real surface rather than just passing through the virtual surface.

For the toroidal source the user specifies the source type name (by the choice of the DLL name), little_r, big_R, the minimum rotation extent of the toroid from the y-axis (phi_min), the maximum rotation extent of the toroid from the y-axis (phi_max), and the ray weight (a zero or blank is defaulted to 1.0).

The table below shows the name of the DLL.

<u>Source Type Name</u>	<u>DLL Name</u>
Toroidal	toroid_Sur.dll

Detector Note:

This source DLL is probably not appropriate for a detector having an attached camera.

Text Commands:

```
modify/source/custom/name="dll_name" name little_r big_R phi_min phi_max * *
weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\toroidal_Surface.dll" 11 1
10 -180 180
modify/source/custom/name="sources\standard\dlls\toroidalSurface.dll" 11 1
10 -180 180 0 0 0 100
```

Helical Surface Source

The Helical source emits rays from the surface of a virtual helix, with each virtual surface element emitting into a Lambertian angular distribution. A helical surface is formed by the rotation of a circle of radius `little_r` located in the x-y plane, about a line parallel to the x-axis located a distance `big_R` away from the center of the circle, but with the x position of the circle increasing linearly with the rotation angle.

For the Helical source the user specifies the source type name (by the choice of the DLL name), `little_r` (the radius of the wire of the helix), `big_R` (the bend radius of the helix), the center-to-center extent of the helix, the minimum rotation extent of the helix from the y-z plane (`phi_min`), the maximum rotation extent of the helix from the y-z plane (`phi_max`), the axial shift in z, and the ray weight (a zero or blank is defaulted to 1.0).

The table below shows the name of the DLL.

<u>Source Type Name</u>	<u>DLL Name</u>
Helical	helix_Sur.dll

Detector Note:

This source DLL is probably not appropriate for a detector having an attached camera.

Text Commands:

```
modify/source/custom/name="dll_name" name little_r big_R c_to_c_size phi_min
phi_max dz weight
```

Text Command Examples:

```
modify/source/custom/name="sources\standard\dlls\helix_Sur.dll" helix1 1 3
10 0 1080
modify/source/custom/name="sources\standard\dlls\helix_Sur.dll" helix2 1 3
10 -540 540 0 1
```

Sun Source

The Sun source approximates incident sun light. For the newer sources sun2 places a lambertian disc at the front focal plane of a perfect lens and evaluates the distribution at rear focal plane, sun3 similarly uses a hemisphere instead of a disk.

The user may wish to use the Emitance Surface option to specify more detailed spatial and angular functions.

For the Sun sources the user specifies the source type name (by the choice of the DLL name), HalfSize (the half size of the ray distribution to be launched from the x-y plane of the source), HalfAngle (the maximum half angle in milliradians of the ray distribution).

The table below shows the names of the DLLs.

<u>Source Type Name</u>	<u>DLL Name</u>
Sun	Sun.dll
Sun2	Sun2.dll
Sun3	Sun3.dll

Detector Note:

This source DLL is probably not appropriate for a detector having an attached camera.

Text Commands:

```
modify/source/standard/name="dll_name" object_name HalfSize HalfAngle 0 0 0  
0 0 0 0
```

Text Command Examples:

```
modify/source/standard/name="sources\standard\dlls\sun.dll" light1 10 5 0 0  
0 0 0 0 0  
modify/source/standard/name="sources\standard\dlls\sun2.dll" light1 10 5 0 0  
0 0 0 0 0  
modify/source/standard/name="sources\standard\dlls\sun3.dll" light1 10 5 0 0  
0 0 0 0 0
```

Spherical Surface Source–Detector

The Spherical Surface source–detector models light uniformly emitted from the surface of a sphere and having a Lambertian angular distribution.

For the Sun source–detector the user specifies the source–detector type name (by the choice of the DLL name), HalfSize (the half size of the ray distribution to be launched from the x-y plane of the source–detector), HalfAngle (the maximum half angle in miliradians of the ray distribution).

The table below shows the name of the DLL.

<u>Source–Detector Type Name</u>	<u>DLL Name</u>
Spherical Surface	Spherical_Surface.dll

Text Commands:

modify/source/standard/name="dll_name" **object_name** Radius

Text Command Examples:

modify/source/standard/name="sources\standard\dlls\Spherical_Surface.dll" light1 1

Measured Data Light Sources

Measured data based light sources differ from other light sources in that the user specifies the name of a data file, rather than the name of a DLL. The DLL is selected based on the modifier (iesna, or usertable).

IESNA Data Sources

The Illumination Engineering Society of North America (IESNA) has established a standard file format for lamp data. This is almost always far field intensity data originating from a point source. There is a great quantity of data available in this format, which is defined in **IESNA Standard File Format for Electronic Transfer of Photometric Data (IESNA LM-63-95; IESNA, 120 Wall St., 17th Floor, New York, NY, 10005; 212 - 248 - 5000)**, and in the **IESNA Lighting Handbook, 8th Edition, New York: I.E.S. of N.A., 1993**.

While an effort has been made to faithfully trace rays based on this standard data, the standard does not define an unambiguous representation of a lamp's far field intensity.

The user is cautioned to examine two data files (both in a csv format) that are written by OptiCAD (only after an IESNA data file has been selected) based on the IESNA data. The first file, ies_log.csv displays the IES data in a clean tabulated format, it is based on a program written by Ian Ashdown of Ledalite Corporation Langley, BC, Canada, and is used with his generous permission. The second file, ies_view.csv displays the original data after being replicated (flipped and then copied as to represent symmetries) to fill the appropriate angle space (based on the IESNA specification).

The user is cautioned to trace a few rays, to verify the geometry in terms of the IESNA format and in terms of the lamp manufacturers implementation of that format, and if necessary to rotate the light source object as required for a particular application.

OptiCAD assumes a coordinate system with the z-axis representing zero horizontal and vertical IESNA angles. A horizontal IESNA angle has the positive z-axis rotated towards the positive x-axis, and a vertical IESNA angle has the positive z-axis rotated towards the positive y-axis.

This feature has been tested mostly with Type C Photometric data (the most common type used in the United States). For Type C Photometric data OptiCAD subtracts 90 degrees from the vertical angle listed in the data file, this supports the most common implementation of this data type in which the zero angle points down.

OptiCAD does not support any of the multipliers in the **LM-63-95** standard.

The data files in this format can be moderately large in size. Users requiring high performance during ray-tracing of complex lamp data should consider upgrading their computer RAM to 256Mb or greater.

Data file names should be limited to eight characters and have the three character extension ies.

Note that only one data table may be associated with any lightsource at the same time. In order to ray-trace from multiple different sources modify a source for the first table, then trace rays from that source, then use the add_rays option to modify a source for the second table, then trace rays from that source.

Note that the DLL iesna.dll must be located in the same directory as the data table, and both files must be in a subdirectory below optacad\sources.

Text Commands:

```
modify/source/iesna/name="table_name" object_name
```

Text Command Examples:

```
modify/source/iesna/name="sources\iesna\data\ge1001.ies" light1
```

User Specified Tabulated Data

The tabulated data feature allows the user to pass a data file name to a user defined DLL, would read tabulated data from the desired file.

In order to use the tabulated dated feature, you must produce a DLL named usertable.dll, and that DLL must be located in the same directory as the data table. Users requiring high performance during ray-tracing of extremely large tabulated files should consider upgrading their computer RAM to 256Mb or greater.

Note that only one data table may be associated with any lightsource at the same time. In order to ray-trace from multiple different sources modify a source for the first table, then trace rays from that source, then use the add_rays option to modify a source for the second table, then trace rays from that source.

Note that the DLL usertable.dll must be located in the same directory as the data table, and both files must be in a subdirectory below optacad\sources.

Text Commands:

```
modify/source/usertable/name="table_name" object_name p1 p2 p3 p4 p5 p6
```

Text Command Examples:

```
modify/source/usertable/name="sources\usertable\f(x)_g(y)\data2.tbl" lite
```

Tabulated Data Example

An example DLL name **usertable.dll** is included with OptiCAD. This version of a user defined tabulated data reads a data file (test.tbl) containing 1-D x-angular data and 1-D y-angular data.

The example models an array of laser diodes (called tiles), each of which has identical angular properties.

The 2-D angular function ($h(x,y)$) which describes the source angular distribution is assumed (for this sample DLL only) to be the product of $f(x) * g(y)$ [$h(x,y) = f(x) * g(y)$], x and y are assumed (for this sample DLL only) to be separable variables.

The data file must contain the full angular range of data, the dll does not assume any symmetry. The data format of energy versus angle is as follows:

nx	ny	dz(x)	dz(y)
theta_x(1)	energy_x(1)		
.....		
theta_x(nx)	energy_x(nx)		
theta_y(1)	energy_y(1)		
.....		
theta_y(ny)	energy_y(ny)		

where nx and ny are the number of data pairs in x and y, dz(x) and dz(y) are the focal shifts in x and y (diode astigmatism), the angle theta is assumed to be in milliradians.

Within OptiCAD the user specifies the data file name number of tiles in x and y, full sizes of an individual tile in x and y, and the center-to-center spacings between tiles in x and y.

Text Commands:

```
modify/source/usertable/name="table_name" object_name n_tiles_x n_tiles_y
x_size y_size dx dy
```

Text Command Examples:

```
modify/source/usertable/name="sources\usertable\f(x)_g(y)\test.tbl" lite 2 2
1 2 4 8
```

where n_tiles_x and n_tiles_y are the number of identical rectangular emitters in x and y, and x_size and y_size are the x and y sizes of a single rectangular emitter, dx and dy are the x and y separation of identical elements.

Tabular Ray Data

OptiCAD can launch rays based on tabulated values of position, direction cosine and energy.

The user enters the name of the data file, the maximum number of unique ray data (Max_Rays) in the user supplied data file, this is not the maximum number of rays you wish to launch - but the maximum number of ray data sets (rows in the file containing ray data); a factor (Scaling_Factor) that will be used to scale all of the ray location data; and a distance (Path) from the data specified location that the ray will propagate before being launched.

Note that a large (user supplied) scale factor may be required if the data is specified in detector (or CCD) coordinates rather than in source object space.

Note that the DLL usertable.dll must be located in the same directory as the data table, and both files must be in a subdirectory below optacad\sources.

The Scaling_Factor and Path are applied to the tabular data as follows:

```
x = x0 * Scaling_Factor + L * Path
y = y0 * Scaling_Factor + M * Path
z = z0 * Scaling_Factor + N * Path
```

where x0, y0, and z0 are the tabular ray origins, L, M, and N are the tabular direction cosines, and x, y, and z are the modified ray origins used by OptiCAD.

This data format is similar to Radiant Sources™ ray data produced by the program RIRayGenerator.exe, available and supported by Radiant Imaging Inc., based on their source database.

OptiCAD Binary Format from Radiant Imaging

This data is exported by the ProSource™ (version 6 and higher) program in a format labeled **OptiCAD Binary**. ProSource™ is sold and supported by Radiant Imaging Incorporated.

The Binary Source File consists of a header structure of the form:

```
typedef struct Radiant Imaging_OptiCAD_Binary_Header
{
int Identifier; // Will be set to 8675309 for quick check of proper format
int NbrRays; // The number of rays in the file
char Description[100]; // A text description of the source
float SourceFlux; // The total flux in watts of this source
float RaySetFlux; // The flux in watts represented by this Ray Set
float Wavelength; // The wavelength in microns, 0 if a composite
float AzimuthBeg, AzimuthEnd; // Angular range for ray set (Degrees)
float PolarBeg, PolarEnd; // Angular range for ray set (Degrees)
int DimensionUnits; // METERS=0, IN=1, CM=2, FEET=3, MM=4
float LocX, LocY, LocZ; // Coordinate Translation of the source
float RotX, RotY, RotZ; // Source rotation (Radians)
float ScaleX, ScaleY, ScaleZ; // Scale factor to expand/contract source
float unused1, unused2, unused3, unused4;
int reserved1, reserved2, reserved3, reserved4;
} Radiant Imaging_OptiCAD_Binary_Header;
```

The data types float and int are both 32 bit types. The identifier integer must be equal to 8675309. OptiCAD only uses the NbrRays, RaySetFlux, DimensionUnits, LocX, LocY, Loc,z, RotX, RotY, RotZ, ScaleX, ScaleY, and ScaleZ parameters. After the header the file includes NbrRays ray structures. Each ray structure is of the form:

```
typedef struct Radiant Imaging_OptiCAD_Binary_Ray_Data
{
float x, y, z;
float l, m, n;
float intensity;
} Radiant Imaging_OptiCAD_Binary_Ray_Data;
```

The maximum number of rays currently allowed is 9,500,000. A file containing this many rays would be approximately 270 Mb in size.

Text Commands:

```
modify/source/usertable/ri_opticad_binary/name="data_file_name"  
object_name Max_Rays Scaling_Factor Path
```

Text Command Examples:

```
modify/source/usertable/name=  
"sources\usertable\ri_opticad_binary\rayset.dat" light1 10000 1
```

New ASCII Text Format

This is a new text format proposed for ProSource™ (version 6 and higher) program. ProSource™ is sold and supported by Radiant Imaging Incorporated.

This format may also be useful for user's wishing to create their own text formatted data file.

The ASCII Source File consists of a single line of header data with two integer numbers:
number_of_rays dimension_flag

The number_of_rays indicates the total number of rays in the file. The dimension_flag is 0 for meters, 1 for inches, 2 for centimeters, 3 for feet, and 4 for millimeters.

The remaining lines in the file are of the format:

```
x y z L M N Energy
```

The maximum number of rays currently allowed is 9,500,000. A file containing this many rays would be approximately 2,850 Mb (2.85 Gb) in size.

Text Commands:

```
modify/source/usertable/ri_text/name="data_file_name" object_name  
Max_Rays Scaling_Factor Path
```

Text Command Examples:

```
modify/source/usertable/name="sources\usertable\ri_text\rayset.dat" light1  
10000 1
```

Older Text Data Format

OptiCAD can also read older ProSource™ text formatted data files, in the form:

```
Title line 1  
Title line 2  
Title line 3  
Title line 4  
x y z L M N Energy  
x y z L M N Energy  
x y z L M N Energy  
x y z L M N Energy  
ENDOFDATA  
RETURN
```

where x,y,z is the origin, L,M,N is the direction cosines, and Energy is the energy of the ray to be launched.

Note that title lines must not begin with a number, or one of the characters between the following square brackets [+ ' - . /].

Also note that this format does not contain data units, requiring the user to use the `Scaling_Factor` parameter in the modify command.

The maximum number of rays currently allowed is 10,000,000. A file containing this many rays would be approximately 3 Gb in size.

Text Commands:

```
modify/source/usertable/ri/name="data_file_name" object_name Max_Rays  
Scaling_Factor Path
```

Text Command Examples:

```
modify/source/usertable/name="sources\usertable\ri\temp1.txt" light1 10000  
1000 0
```

Array Using 2-D Source Profile Data

OptiCAD can launch rays based on a measured 2-D irradiance map of a source (an image of the source). The image of the source may be replicated in order to create a 2_d array of sources (such as an array of LEDs). Rays are launched uniformly over the area of each source, but are weighted based on the source profile data. Rays are launched angularly into a Lambertian distribution.

This array source also allows the user to enter exception data for up to five array elements in order to model errors in location, tilt, and relative energy output for these array elements.

There are no input parameters required from within OptiCAD, instead all input parameters are in the data file.

The data file and usertable.dll must be in the same directory.

The dll writes out the first six lines of data to the file Profile2D_Data.txt.

Data Format

All data is in a tab delimited ASCII format, such as when data is exported from a spreadsheet program as **text (tab delimited)**.

Row 1 Contains Basic Information:

NRows NColumns XSize YSize NX NY deltaX deltaY

Where NRows is the number of rows (x) of data for a single element, NColumns is the number of columns (y) of data for a single element, XSize is the x size of a single element, YSize is the y size of a single element, NX are the number of elements in x, NY are the number of elements in y, deltaX is the center to center spacing of elements in x, and deltaY is the center to center spacing of elements in y.

Rows 2 through 6 Contains Exception Data for Individual Elements:

RowNumber ColumnNumber dX dY XTilt YTilt EnergyFactor
RowNumber ColumnNumber dX dY XTilt YTilt EnergyFactor
RowNumber ColumnNumber dX dY XTilt YTilt EnergyFactor
RowNumber ColumnNumber dX dY XTilt YTilt EnergyFactor
RowNumber ColumnNumber dX dY XTilt YTilt EnergyFactor

Where RowNumber is the row number (starting from 1) of the element that has exception (to the array) data, ColumnNumber is the column number (starting from 1) of the element that has exception (to the array) data, dX is the x displacement of this element from its nominal position in the array, dY is the y displacement of this element from its nominal position in the array, XTilt is the x tilt (in miliradians) of this element from its nominal position in the array, YTilt is the y tilt (in miliradians) of this element from its nominal position in the array, EnergyFactor is the relative energy in this element compared to the energy of the LEDs in the data file.

Rows 7 through NRows + 6 contain the profile map data of a single array element.

Text Commands:

```
modify/source/usertable/name="data_file_name" object_name
```

Text Command Examples:

```
modify/source/usertable/name="sources\usertable\2dprofile\data1.txt" light1
```

User Defined Custom Source

Making a user defined DLL

All custom lightsources are ray-traced using a Dynamic Link Library (DLL). The user may write a DLL in order to represent the spatial and angular distributions of a light-source distribution which cannot be modeled using one of the existing DLLs included with OptiCAD.

Note that Opticad Corporation does not support or debug user written software.

In order to write a DLL you must have access to a suitable compiler or development tool that can generate 32 bit Windows Compatible DLLs. The basic procedure for making a DLL using Microsoft Visual C++ version 5.0 is described in this section. This description assumes that you have installed OptiCAD in the directory `c:\opticad`.

1. Create and name a new project workspace, specifying that it is a Dynamic-Link Library, do not select MFC or Microsoft Foundation Class Libraries type DLL.

A subdirectory will be created having the name of your project (such as `c:\msdev\my_source`).

2. Select a subdirectory representing a source that is closest to your application from the `c:\opticad\sources\custom` subdirectory, and copy all of the sample files in that subdirectory into your new subdirectory (as an example copy the contents of `c:\opticad\sources\custom\e11_vol` into `c:\msdev\my_source`). The subdirectories in `c:\opticad\sources\custom` also contain a project file for reference.

Note that source files for all sources may not be present.

Copy the contents of the subdirectory `c:\opticad\sources\custom\include` into the default directory for include files (usually `c:\msdev\include`).

3. You will find two files that have the same name as the source directory from which they were copied (as an example `e11_vol.c` and `e11_vol.def`).

Rename both of these files to have the same name as your new subdirectory, but keep the extension unchanged (such as `my_source.c` and `my_source.def`).

4. Now make all of the changes that are required in your new source program(s), and select Build and Build `my_source.dll` (or the name of your project the extension dll appended) in order to create a DLL file.

5. You can access the DLL from OptiCAD by inserting its name (with the full path in double quotes if needed) on the grin materials definition line (such as `sources\custom/name="e11_vol.dll" my_source`).

For the Borland C compiler the following setting choices may be helpful: Select File -> New -> Project, New Target box -> set the path, Target Type -> Dynamic Library, Platform -> Win32, Target Model -> GUI, and Static is the only other option selected no Frameworks, no Controls, no Libraries).

DLL Data Array

The user defined DLL uses a single data array (named data) to interact with OptiCAD. The table below describes the contents of that array.

<u>Array Element</u>	<u>Received from OptiCAD</u>	<u>Sent to OptiCAD</u>	<u>Description</u>
data[0]	21		Maximum array index
data[1]	lambda		wavelength (in microns)
data[2]	ray weight	ray weight	waveweight (in) new weight (out)
data[3]			
data[4]	x	x	x ray origin
data[5]	y	y	y ray origin
data[6]	z	z	z ray origin
data[7]	l	l	x ray direction cosine
data[8]	m	m	y ray direction cosine
data[9]	n	n	z ray direction cosine
data[10]	theta_x_max		maximum ray angle in x
data[11]	theta_y_max		maximum ray angle in y
data[12]	internal use only		

<u>Array Element</u>	<u>Received from OptiCAD</u>	<u>Sent to OptiCAD</u>	<u>Description</u>
data[13]	internal use only		
data[14]	p01		user defined constant
data[15]	p02		user defined constant
data[16]	p03		user defined constant
data[17]	p04		user defined constant
data[18]	p05		user defined constant
data[19]	p06		user defined constant
data[20]	p07		total power/energy
data[21]	internal use only		

Data sent by OptiCAD to the DLL

The element data[0] is the maximum usable address of the data array. The element data[1] is the wavelength, (which cannot be changed), the element data[2] is the wave-weight factor, data[4], data[5], and data[6] are the x, y, and z coordinates of the current ray location, data[7], data[8], and data[9] are the direction cosines with x, y, and z, data[10] and data[11] are the maximum full angles in x and y of the trace, data[12] and data [13] are internal flags, and data[23] is the detector area.

User entered data passed (unchecked) by OptiCAD to the DLL

The command line in OptiCAD that modifies a lightsource also allows the user to pass data to the named DLL. OptiCAD passes the 2nd (where the first is an object name) through 8th user entered parameter (named p01 through p07) to the DLL as array elements data[14] through data[20].

Note that OptiCAD does not check this data in any way, it is the sole responsibility of the user to only enter reasonable values for items p01 through p20, and to thoroughly check data inside any user defined DLL.

DLL calculated data that MUST be sent back to OptiCAD by the DLL

The element data[2] is the waveweight factor which should include the incoming wave-weight factor, data[4], data[5], and data[6] are the x, y, and z coordinates from which the ray is to be launched, data[7], data[8], and data[9] are the direction cosines with x, y, and z indicating the direction into which the ray is to be launched, data[10] and data[11] are from the ray–trace command and indicate the maximum full angles in x and y of the trace.

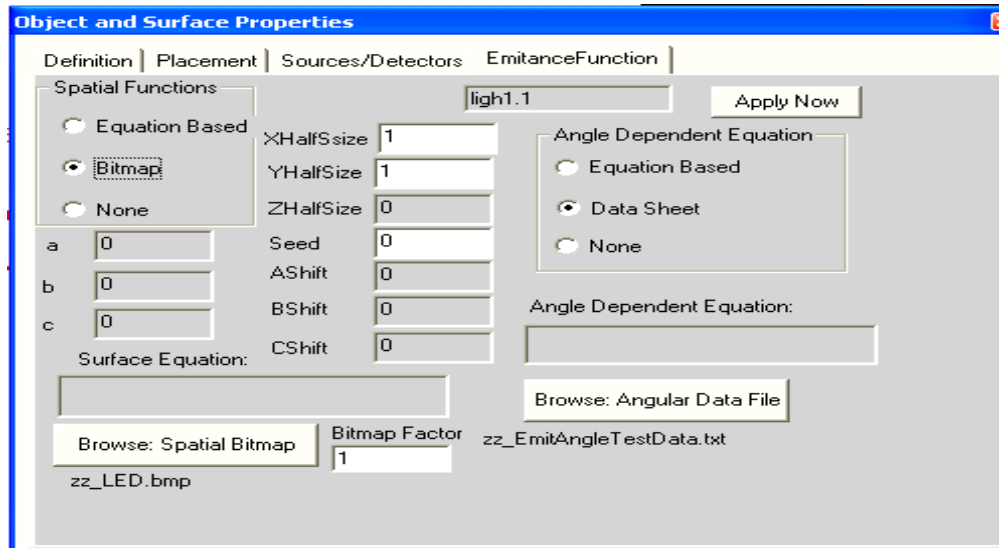
Example of a user defined DLL

See the directory sources\custom for source code for some of the standard DLL based lightsources describe above.

The Apply Now button only implements the values for this page.

Aborting the object change may be accomplished by closing the box (by clicking the x in upper right of the box).

Emitance Functions Page



This page is specific to the object.surface that was selected using the **OOB** described in a previous section.

Emitance Functions

Function:

Defines a **3-D volumetric emitance function**

Discussion:

A volumetric emitance function consists of a spatial and an angular component.

The spatial component can be either a user entered equation based on the x-y-z ray origin, or a user specified bitmap (bmp) file. The angular component can be either a user entered equation based on the ray direction or a data file containing tabulated angular data.

The emitance function can only be assigned to a lightsource.

For the 3-D equation option, rays are launched uniformly from a user defined 3-D rectangle and/or ellipsoid, and into a uniform (isotropic) distribution. The rays are then weighted in power according to the x-y-z ray origin and the ray direction.

For the bitmap option, rays are launched uniformly from the x-y plane and then weighted in power by the values of the bitmap at that x-y location and the ray direction.

A single spatial and a single angular function may only be applied to a light source.

The selected bitmap can be either 8 bit or 24 bit. For a 24 bit bitmap, the red, green, and blue values are averaged. Bitmap values are normalized by 255, resulting in a value between 0 and 1. For a bitmap source rays are launched from the x-y plane (z=0), A table showing the permitted functions in a user entered equation is shown in the **Reflection Functions Page** section in this chapter.

In order to trace rays using this emittance function use the **Raytrace->Function** feature, described in the **Function** subsection of the **Raytrace** section of the **Analyze** chapter.

The spatial emittance value is multiplied by the angular emittance value, resulting in the total position and angle emittance. A non-entry for the spatial or angular function is interpreted as a value of 1.

The value for total_power entered on the **Definition** page is not used in this calculation.

3-D Equation Based Spatial Emittance Weighting Function

The user enters a permitted function of the variables: x, y, and r where

$$r = \sqrt{x^2 + y^2}$$

Parameters

The user enters the half sizes (halfsize_x, halfsize_y, halfsize_z) of a 3-D rectangle from which rays are uniformly launched, the half sizes (halfsize_a, halfsize_b, halfsize_c) of a virtual ellipsoid that also limits the region from which rays are emitted, the location (shift_a, shift_b, shift_c) of the limiting ellipsoid, and a seed value (seed) for the random number generator used in the ray launching.

Note that a value of 0 for halfsize_x will cause rays to be launched from the y-z plane, a value of 0 for halfsize_y will cause rays to be launched from the x-z plane, and a value of 0 for halfsize_z will cause rays to be launched from the x-y plane. While a zero value for halfsize_a will cause rays to be limited by an x cylinder, a zero value for halfsize_b will cause rays to be limited by an y cylinder, and a zero value for halfsize_c will cause rays to be limited by an z cylinder. The user also selects a random number seed value. Seed value of 0 (recommended) seeds to the system clock.

Text Commands:

```
emittancefunction/spatialequation/name="spatial_equation" object_name
halfsize_x halfsize_y halfsize_z halfsize_a halfsize_b halfsize_c
shift_a shift_b shift_c seed
emittancefunction/remove_spatial object_name
```

Text Command Examples:

```
emittancefunction/spatialequation/name="0.5 + 0.5 * cos(20. * x * x)"
light1 1 1 1 0.5 1 1 0 0 0 0
emittancefunction/remove_spatial mirr2.1
```

Bitmap Based Spatial Emittance Function

The user selects a bitmap file and enters the x and y sizes of the bitmap and a bitmap factor which is multiplied by the normalized bitmap values. The user also selects a random number seed value. Seed value of 0 (recommended) seeds to the system clock.

Text Commands:

```
emitancefunction/attach/file="bitmapfilename.bmp" object_name
halfsize_x halfsize_y bmpfactor seed
emitancefunction/remove_spatial object_name
```

Text Command Examples:

```
emitancefunction/attach/file="OpticadLogo.bmp" mirr2.1 0.99 0.99 1 0
emitancefunction/remove_spatial mirr2.1
```

Equation Based Angular Emitance Function

The user enters a permitted function of the variables: $xtheta$, $ytheta$, and $theta$ where

$$theta = \arccos N$$

$$xtheta = \arctan \frac{L}{N}$$

$$ytheta = \arctan \frac{M}{N}$$

Where L, M, N are the direction cosines of the ray with respect to the +z axis. The user also selects a random number seed value. Seed value of 0 (recommended) seeds to the system clock.

Note that the angles are in radians.

Text Commands:

```
emitancefunction/angularequation/name="angular_equation" object_name
seed
emitancefunction/remove_angular object_name
```

Text Command Examples:

```
emitancefunction/angularequation/name="0.5 + 0.5 * cos(20. * theta)"
mirr2.1 0
emitancefunction/remove_angular mirr2.1
```

Data File Based Angular Emitance Function

The user selects an angular data file and a random number seed value. Seed value of 0 (recommended) seeds to the system clock.

Angular Data File Format

The angular data file consists of one header line followed by lines of data.

Note that the angles are in degrees.

The header line is:

```
Nx Ny Theta_Half_X Theta_Half_Y
```

where N_x is the number of data values in a row (number of x values), N_y is the number of rows (number of y values), Theta_Half_x is the half angle of the x data (the x data goes from $-\text{Theta_Half_x}$ to $+\text{Theta_Half_x}$), and Theta_Half_y is the half angle of the y data (the y data goes from $-\text{Theta_Half_y}$ to $+\text{Theta_Half_y}$). The N_y data lines are:

$P(1,y) P(2,y) \dots P(N_x,y)$

Example of a data file for $N_x=5, N_y=5, \text{Theta_Half_x}=10, \text{Theta_Half_y}=10$ is shown below:

```
5 5 10 10
0.125 0.250 0.500 0.250 0.125
0.188 0.375 0.750 0.375 0.188
0.250 0.500 1.000 0.500 0.250
0.188 0.375 0.750 0.375 0.188
0.125 0.250 0.500 0.250 0.125
```

Text Commands:

```
emitancefunction/angulardatafile/file="angulardatafile.txt"
object_name seed
emitancefunction/remove_angular object_name
```

Text Command Examples:

```
emitancefunction/angulardatafile/file="zz_EmitAngleTestData.txt"
mirr2.1 0
emitancefunction/remove_spatial mirr2.1
```

Initial Polarization Page



This page is specific to the object or surface that was selected using the **OOB** described in a previous section.

The **Initial Polarization** page can be used to modify the initial polarization of a specific lightsource.

Function:

Defines the initial polarization of a lightsource.

Discussion:

This initializes the polarization of a lightsource. The initial polarization is specified in the local coordinate system of the lightsource. This definition of initial polarization rotates with the lightsource. This definition differs from the global polarization of initial polarization defined using **Options -> Polarization->Initialization**.

Note that defining a global initial polarization using **Options -> Polarization->Initialization** will override (reset) all of the individual source definitions on initial polarization.

Parameters:

Previously defined lightsource or detector name, amplitude of the initial polarization vector (Ex, Ey, Ez), and the phase (in degrees) of the initial polarization vector (Phx, Phy, Phz).

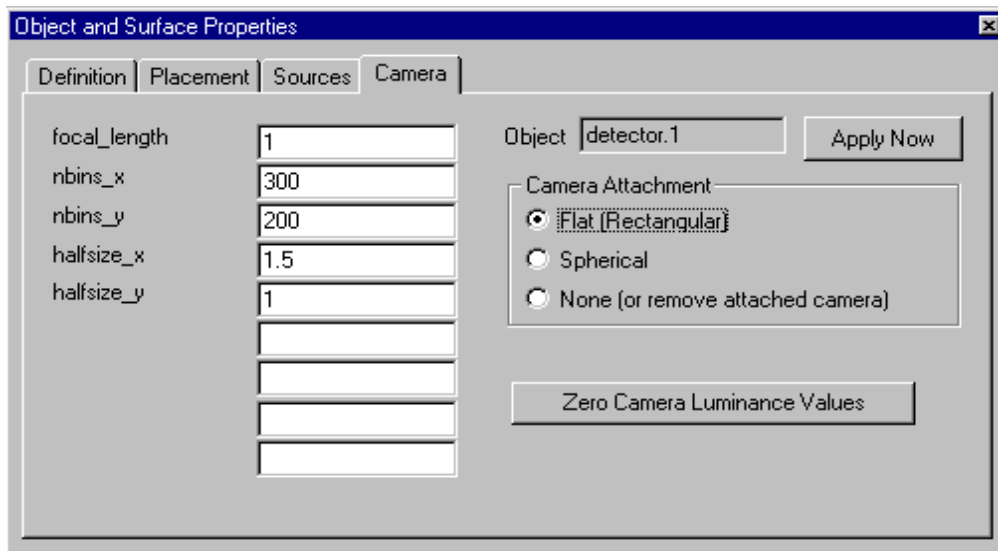
Text Command:

```
coating/polarization/source_initial object_name ex ey phx phy ez phz
```

Text Command Example:

```
coating/polarization/source_initial sourceoname 1 1 0 0 0 0
```

Camera Page



This page is specific to the object or surface that was selected using the **OOB** described in a previous section.

Camera

Function:

Defines a **camera focal surface array**

Discussion:

A camera is a virtual focal surface attached to a detector. Radiometric rays launched from a detector are projected onto this surface, and the accumulated power of the ray is added into an element of a two-dimensional array which is superimposed on the surface. A camera surface can be either flat or spherical. Rays launched from a single point on the detector represent a pinhole camera. Rays may be launched from a surface or volume using any source DLL.

The radiometric rays act as travelling power buckets, accumulating power from surfaces having a non-zero irradiance.

A camera is not a physical object, it is an appended property of a detector object. The detector object is placed and rotated not the camera, which moves with the detector. The camera is the focal plane of the detector/camera object, the detector is the source of radiometric rays. The camera is not drawn and does not intercept rays.

The addition of a camera to a detector activates reverse ray-trace radiometry. The camera calculates the total accumulated power of detector launched rays and the power of rays that intercepted the camera.

The choice of a flat or spherical camera determines the assumptions made about the detector surface. Choosing a flat camera assumes a flat detector, choosing a spherical camera assumes a spherical detector. These assumptions effect the calculated power on the camera and the total calculated power. The total power will be greater or equal to the total power on the camera focal surface because rays may not intercept the focal surface.

Certain ray cone spacings are not permitted when performing radiometric backwards ray-tracing.

See the **Detector** section of the **Design** chapter, the **Raytrace** section of the **Analyze** chapter, and the **View Camera Array** section of the **View** chapter.

Camera and Reverse Raytrace Radiometry Results

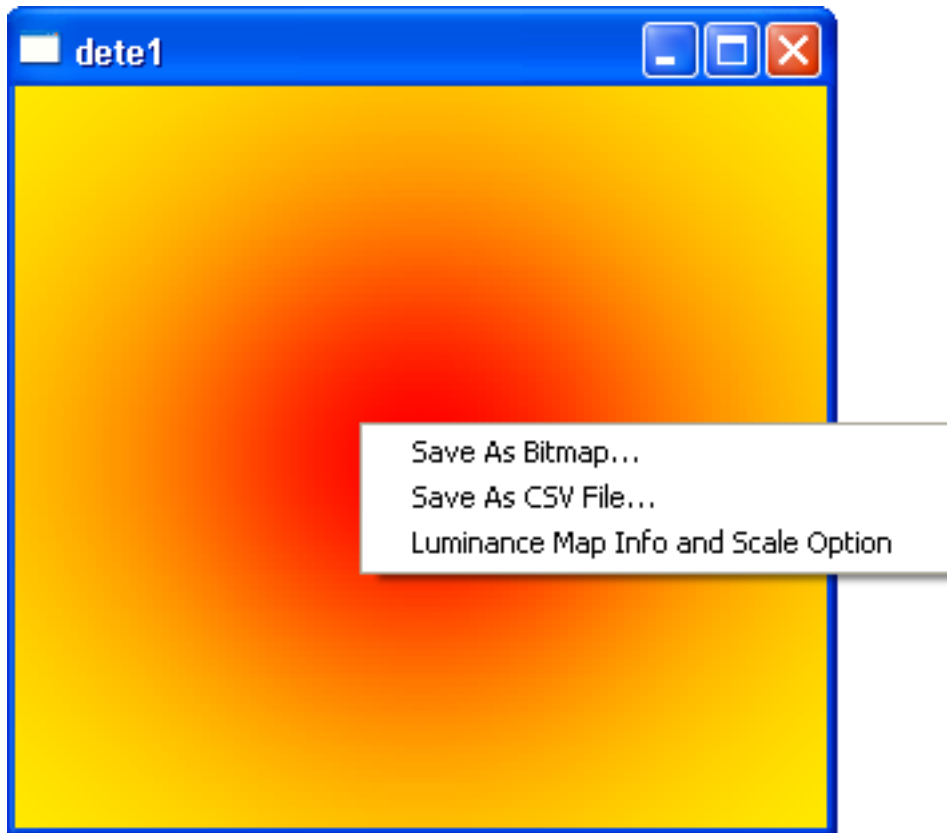
Numerical Data:

The total power accumulated by the reverse ray-trace, the total power normalized by the detector area (DLL based source distributions only), the total power collected by the camera array, and the total camera array power normalized by the detector area (DLL based source distributions only) are displayed in the Text Out box at the bottom of the screen.

Camera Image

The camera array is displayed in a pop-up window by using the feature **View->View Camera Array** (View menu). The image can be resized and saved to a bitmap, normalized by a user entered value, and exported in a comma separated values format by right clicking the mouse while in the pop-up display window. An array value of 0.0 will be displayed as black if OpenGL solid is selected and white if OpenGL solid is not selected.

Note: Click on the right mouse button while the camera image is displayed for more options.



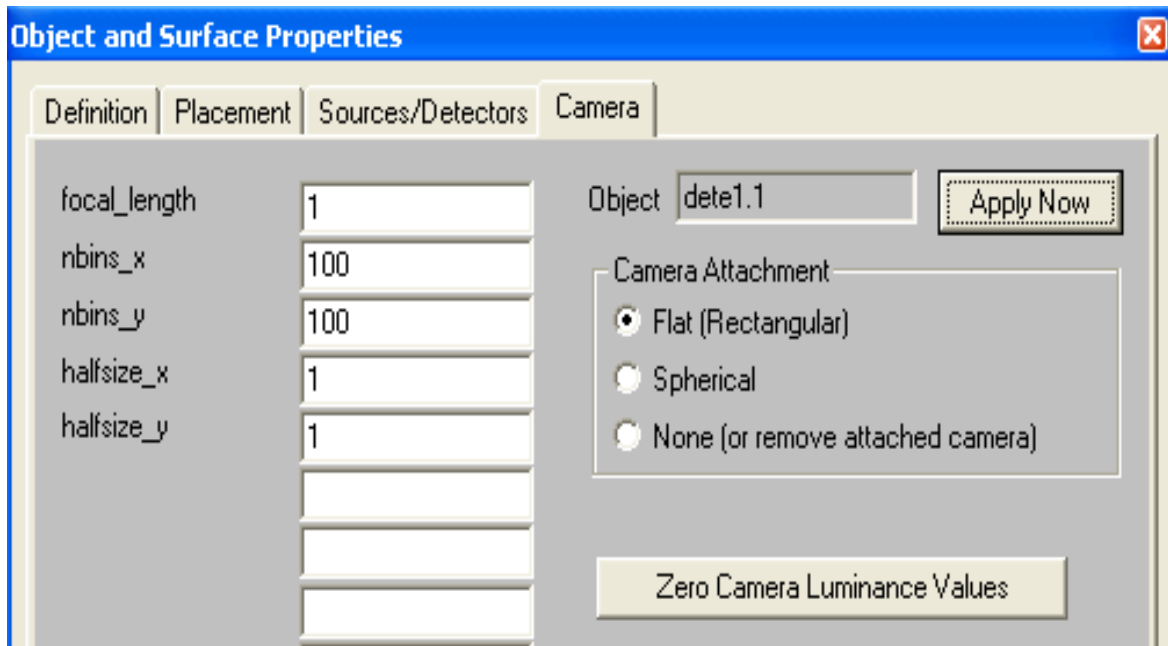
Note: Data in the camera array is normalized by the solid angle of an on-axis pixel as seen from the detector. For a point source the results are proportional to intensity.

Add Rays

The Add Rays option works differently for a detector than for a lightsource. When the Add Rays option is on a detector accumulates power from multiple ray-traces in a similar manner to film in a camera under going multiple exposures. Repeating the same ray-trace a second time will double the total accumulated power reported in the **TextOut box** and in the camera array. See the **Add Rays** section of the **Options** chapter.

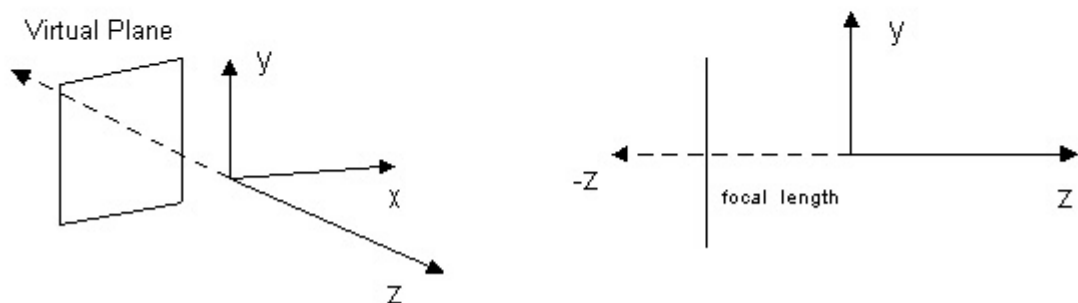
Flat Camera

Selecting a flat camera assumes a flat detector for reverse ray-tracing radiometry.

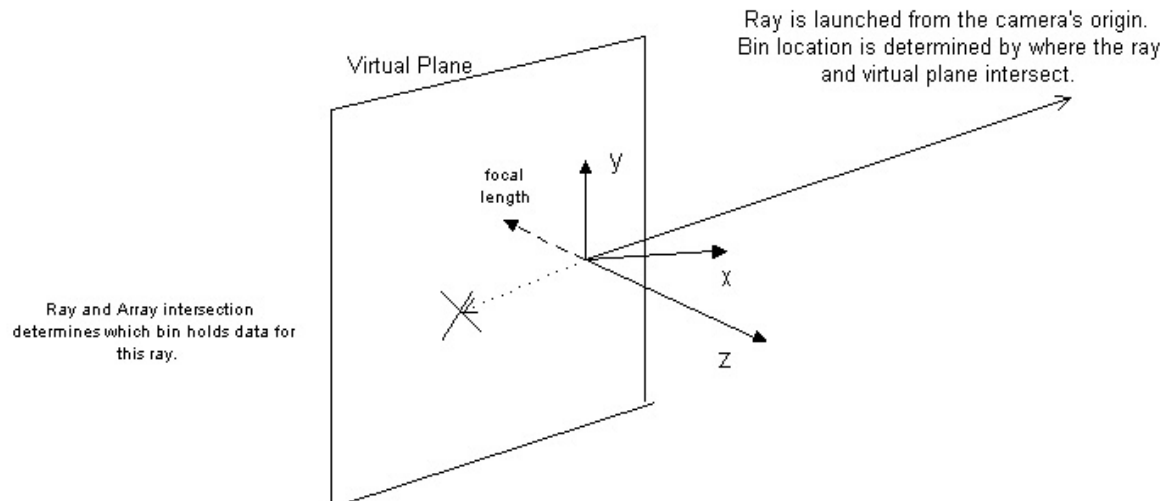


OptiCAD Camera Definition
Flat Virtual Plane

Camera's local coordinate system



OptiCAD Bin Location Scheme Flat Virtual Plane



Parameters:

For a flat camera (focal surface) the user enters the x and y half sizes, a focal length (the distance from the detector to the camera focal plane), and the number of array elements in x and y.

Text Commands:

```
camera/attach/flat detector_name focal_length nbins_x nbins_y  
halfsize_x halfsize_y  
camera/reset detector_name  
camera/remove detector_name  
camera/csvfileout/filename detector_name  
view/camera detector_name  
view/camera/gray detector_name  
view/camera/red detector_name  
view/camera/green detector_name  
view/camera/blue detector_name  
view/camera/spectrum detector_name  
view/camera/spectrum detector_name
```

Text Command Examples:

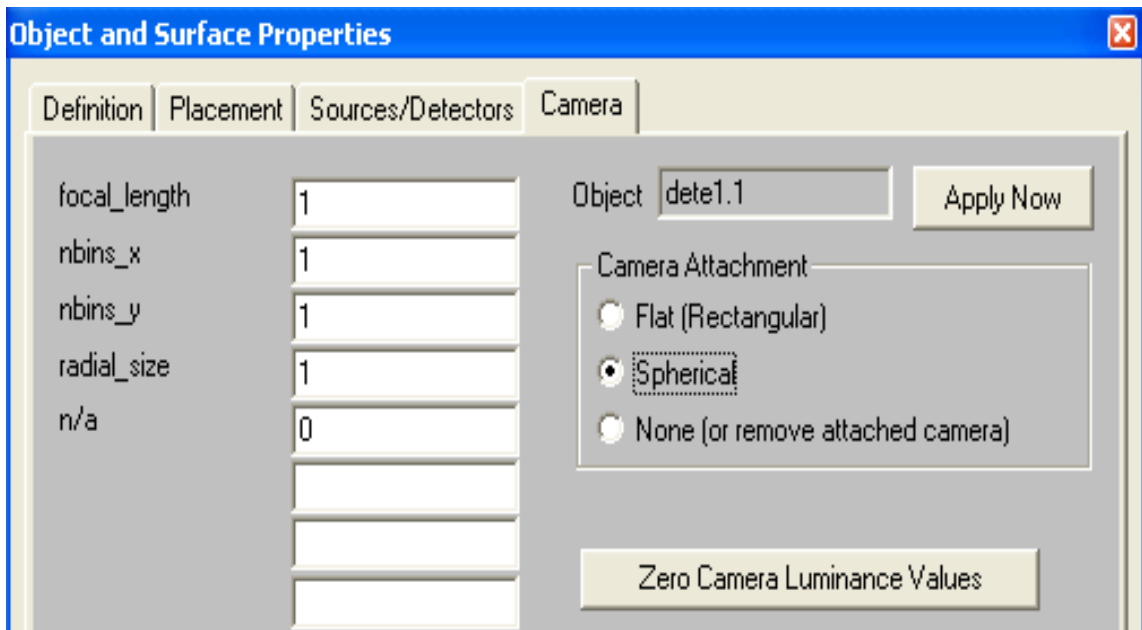
```

camera/attach/flat detector1 1 100 100 1 1
camera/reset detector1
camera/remove detector1
camera/csvfileout/filename detector1
view/camera detector1
view/camera/gray detector1
view/camera/red detector1
view/camera/green detector1
view/camera/blue detector1
view/camera/spectrum detector1
view/camera/spectrum detector1

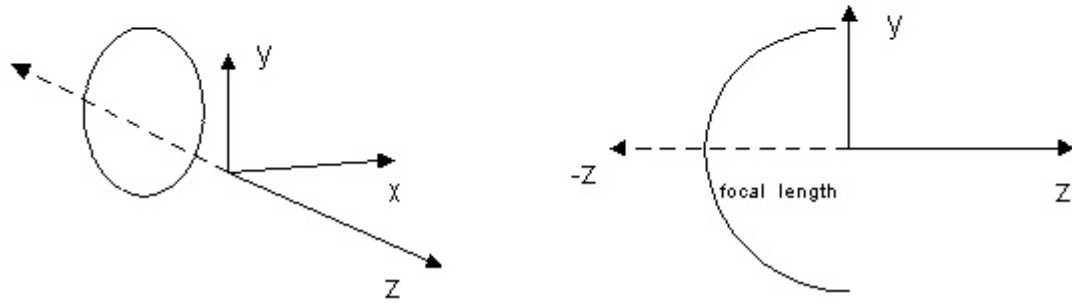
```

Spherical Camera

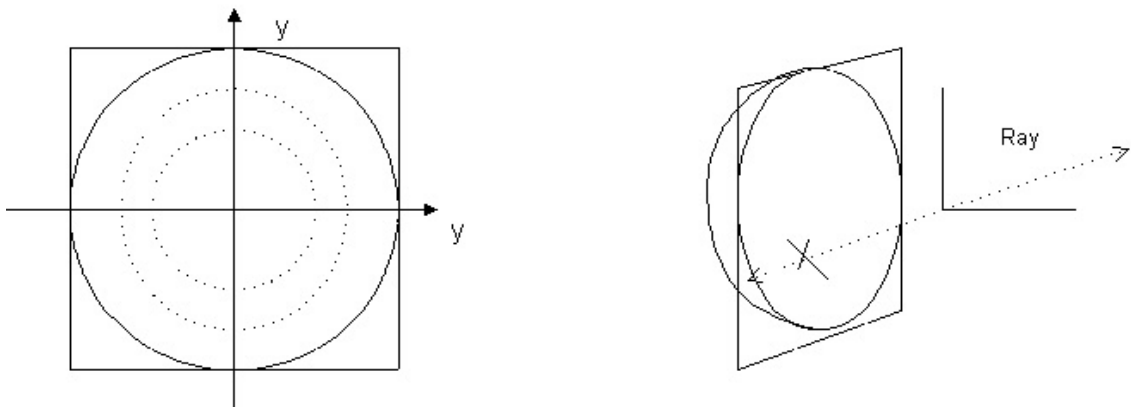
Selecting a spherical camera assumes a spherical detector for reverse ray-tracing radiometry. The array elements are equally spaced in x and y locations on the spherical surface.



OptiCAD Camera Definition
Spherical Virtual Plane



OptiCAD Camera Definition Spherical Bins



The physical size of the array is dictated by the radius of the sphere, which is the focal length.

The bin location will be determined by finding the intersection of the sphere and the ray.

Parameters:

For a spherical camera (focal surface) the user enters a radial size, a focal length (the radius of the spherical focal surface which is centered on the detector), and the number of array elements in x and y.

Text Commands:

```
camera/attach/spherical detector_name focal_length nbins_x nbins_y
radial_size
camera/reset detector_name
camera/remove detector_name
camera/csvfileout/filename detector_name
view/camera detector_name
```

Text Command Examples:

```
camera/attach/spherical detector1 1 100 100 1
camera/reset detector1
camera/remove detector1
camera/csvfileout/filename detector1
view/camera detector1
```

End of Object and Surface Properties Chapter

POLYNETS

General Discussion

The definition of a polygonally faceted object (called a polynet) allows the accurate representation of any truly faceted object (some prisms, 1-D linear Fresnel lenses, illumination homogenizers, cubes, and wedges). For continuous objects, the shape must be approximated by many adjacent (continuous with no physically unreal gaps) flat facets, and the quality of the representation is determined by the number of facets used. Generally a faceted approximation of a continuous object is adequate for (nonimaging) illumination systems and not good enough for high-quality imaging applications.

A polynet (surface or object) is defined as a continuous grouping of polygons (individual facets). A polygon is a flat surface in three dimensions. The extent of a polygon (facet) is defined by a group of 3-D points (called a chain) that must lie in a plane. These points are the limiting corners (vertices) of the polygon. The first chain associated with a polygon defines its outer limits. A second chain may be associated with a polygon in order to specify a hole.

The polygonally faceted representation requires the 3-D description of the corners of each facet. This can be accomplished by direct entry of all corners, or in the case of an extruded 2-D object, by entry of the 2-D coordinates ($z = 0$) and an extrusion (into z) distance. Faceted objects may be either shell (no thickness) or 3-D solid objects.

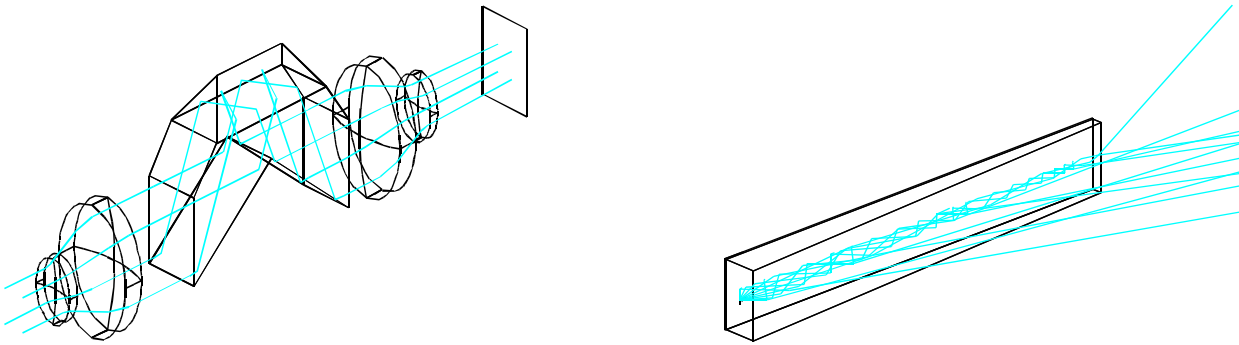
A polynet (shell) surface can be made into a detector (film) by attaching a Radiometer. See the **Radiometer** section of the **Analysis** chapter.

On rare occasions rays will hit exactly at a polynet seam and leak through, ignoring both facets. This generally represents an insignificant amount of energy and could probably be ignored. Ray leakage can be minimized by tracing an even number of rays (10 by 10 rather than 9 by 9). If leakage still occurs, rotate the lightsource about its local z -axis by a small amount (1° or less). This rotation breaks the symmetry between the x - y ray pattern and facet locations having been defined in a similar x - y system.

Note that there is a limit of 32,000 polygons for an object.

T.I.R. Option

The default reflectance for a ray that has undergone total internal reflectance (T.I.R.) at an uncoated surface is 100%. The **T.I.R. User Defined Reflectance** option described in the **Options** chapter uses the reflectance specified on the **Attributes** page for the surface reflection. This option only effects rays that have undergone T.I.R..



Generalized Polynet Object and Extruded Polynet Object

Implementation

Due to its complexity, a polynet object may only be defined from a macro.

In all cases, it is the user's responsibility to define an object or surface that is physically real.

There are several methods you can use to create a polynet object:

- 1) Direct entry using a text editor. Examples of this method are shown for a generalized polynet in the macro `abbe.wdm`, and for an extruded polynet in the macros `wedge.wdm` and `wedge1.mac`.
- 2) Generating the input macro with a user-written program. An example of a user-written program is shown in the Quick BASIC program `filament.bas`.
- 3) Generating the input macro with the output of a spreadsheet. For an example of an extruded polynet object created in Excel, see `Ex_Poly.xls`, which can be saved as `Ex_Poly.txt`, and be run as an OptiCAD macro.

Discontinuous (separate) objects should not be defined as a single polynet. Instead each separate object should be defined as a separate polynet.

Note that a local rotation of a polynet object occurs about its local coordinate system, which should be chosen with this in mind.

An example of an extruded polynet and of a generalized polynet are shown at the end of this chapter. Other examples of polynet objects are shown in the Advanced Features chapter.

Verification

Suggestions on how to verify that a polynet object is correctly defined.

1. If (and only if) the polynet is a closed solid object, then select `View -> Polynormals` from the menus and verify that all of the facet surface normals are facing outwards. If they are not or if the `polynet/normalize` command produced an error message, then the polynet that you have specified may not be physically real.
2. Produce solid model views from several directions - missing or wrong facets will often become obvious.
3. In order to color highlight a selected facet read the `Highlight -> surface/facet` section of the `view` chapter.

Polynet Commands

The polynet commands differ from other OptiCAD commands in that they must be entered in exactly the correct sequence, and they can only be entered from a macro. Polynet commands are unforgiving as to syntax. Input errors will often not be discovered until the object has been designed, placed, and ray-traced.

Spelling most definitely counts.

The polynet syntax cannot determine what the user really wanted to define, nor can it determine whether the user intended to create an open shell surface or a closed solid object. It is the user's responsibility to check all input data and to verify that the desired object is in fact being modeled.

Note: make sure the text command:

```
polynet/normalize my_polynet
```

appears immediately after the

```
polynet/finish
```

command.

Polynet - Start

Function:

Starts the definition of a polynet object

Discussion:

This command is required to begin the definition of a polynet object. The formula for the Beer's law volumetric absorptance is

$$T = e^{-(volabsorb \cdot p)}$$

where T is the transmittance through the polynet for a particular ray and p is the path length in the material for that ray.

Parameters:

User-defined object name, surface reflectance of all polygon facets, surface absorptance of all polygon facets, volumetric (ray path length dependent) Beer's law (bulk) absorptance of the polynet material, and the index of refraction of the polynet material.

A material name is case sensitive. Glass names must be entered in upper case. The glass catalog containing a selected glass must have been loaded (see set/matcatalog in the **Options** chapter) prior to any use of the glass in an object definition.

Text Commands (note alternative input format):

```
polynet/start name index=n refl=surf_reflect abs=surf_absorp volabs=bulk_abs  
polynet/start/material=GLASS name refl=surf_reflect abs=surf_absorp vol-  
abs=bulk_abs  
polynet/start name surface_reflectance surface_absorptance surface_luminance  
0.0 bulk_luminance bulk_absorptance index_of_refraction  
polynet/start/material=GLASS name surface_reflectance surface_absorptance  
surface_luminance 0.0 bulk_luminance bulk_absorptance index_of_refraction
```

Text Command Examples:

```
Polynet/start poly1 index=1.5 refl=0 abs=0 volabs=0
Polynet/start/material=BK7 poly1 refl=0 abs=0 volabs=0
polynet/start name 1 0 0 0.0 0 0 1.5
polynet/start/material=GLASS name 1 0 0 0.0 0 0 1.5
```

Polynet - Polygon

Function:

Starts the definition of a single **polygon** facet

Discussion:

The sequence of commands that defines a single facet must begin with this command and end with either the polynet/thick or polynet/close command. Parameters appended to the polynet/polygon command line are optional; if entered they will override the object parameters defined with the polynet/start command for this polygonal facet only. Either of the command lines polynet/thick or polynet/close will close a polygon.

Parameters:

Surface reflectance of this polygon facet only, and surface absorptance of this polygon facet only.

Text Command:

```
polynet/polygon refl=facet_surf_reflect abs=facet_surf_absorp
```

Text Command Example:

```
polynet/polygon refl=0.5 abs=0
```

Polynet - Chain

Function:

Starts definition of this **facet's** limits

Discussion:

Starts the definition of a series of 3-D points that together determine a limit (extent) of this polygon facet. A chain is automatically closed when a new chain is opened (with another polynet/chain command), the polynet/thick command occurs, or when the polynet/close command occurs.

Text Command:

```
polynet/chain
```

Polynet - Point

Function:

Enters the **corners** of a 3-D facet

Discussion:

This command is used multiple times to define a series of 3-D points that together determine the extent of this polygon facet.

Never repeat a point on a chain, and never repeat the first point of a chain as the last point of the same chain.

Parameters:

The x, y, and z coordinates of a vertex point (a limiting corner of the polygon facet).

Text Commands:

```
polynet/point x1 y1 z1
polynet/point x2 y2 z2
polynet/point x3 y3 z3
polynet/point x4 y4 z4
```

Text Command Examples:

```
polynet/point 1 1 0
polynet/point 1 -1 0
polynet/point -1 -1 0
polynet/point -1 1 0
```

Polynet - Vertex

Function:

Reuses a previously entered **point**

Discussion:

This rarely used command allows the indirect entry of a 3-D point by referring to (re-using) a previously defined point rather than explicitly re-entering the point location. This data entry method results in some saving in execution time and disk space.

Parameters:

The polygon number (in order of definition — starting with 1), the chain number (in order of definition — starting with 1), and the vertex number (in order of definition — starting with 1).

Note: The order of the numbers entered in the text command below is reversed from the order described above.

Text Command:

```
polynet/vertex vertex_number chain_number polygon_number
```

Text Command Example:

```
polynet/vertex 3 2 1
```

Polynet - Thick

Function:

Extrudes a 2-D surface into z

Discussion:

Extrudes a 2-D surface into the +/- z, creating a 3-D object. For a 2-D surface at z=0 the extent of the extruded object is from -thick/2 to +thick/2, where thick is the extrusion parameter. The 2-D surface to be extruded should have no variation in z (all z values must be the same); any variation in z will cause the ray-trace to work improperly. The Polynet - Thick command closes the last chain and also closes the last polygon.

Parameters:

The thickness of the extruded object in z.

Text Command:

```
polynet/thick z_full_thickness
```

Text Command Example:

```
polynet/thick 0.5
```

Polynet - Close

Function:

Closes the polygon

Discussion:

The Polynet - Close command closes the last chain and also closes the last polygon.

Text Command:

```
polynet/close
```

Polynet - Finish

Function:

Ends polynet definition

Discussion:

Finishes the series of commands that creates a polynet object. This command is mandatory.

Text Command:

```
polynet/finish
```

Modify

Function:

Modify reflectance/absorption of an **individual polygon**

Discussion:

The redefinition of the reflectance and absorptance of an individual polygon facet overrides the object-wide definitions made in the polynet/start text command. This is the only Polynet definition or redefinition operation that may be performed from a menu box (see the **Modify** chapter).

Parameters:

The polynet object name, the polygon number (assigned sequentially during definition), new reflectance, and the new absorptance.

For a generalized 3-D polynet, polygons are numbered in the order in which they are entered starting with 1. For an extruded polygon the numbering is as follows:

n + 1 is the top face at + z parallel to the x-y plane, n + 2 is the bottom face at - z parallel to the x-y plane, n + 3 is the side between point 1 and point 2, n + 4 is the side between point 2 and point 3, n + m + 2 is the side between point 3 and point m + 1, where n is number of the first polygon and m is the number of points in the surface that is to be extruded.

You can highlight individual polygons using the View -> Highlight feature.

Text Command:

```
polynet/modify my_polynet polygon_number reflect absorbtion
```

Text Command Example:

```
polynet/modify poly1 1 0.5 0.0
```

ModifyPolynet		
	Object name:	Cancel
	my_polynet	OK
polygon:	polygon_number	
reflectance:	polygon_reflectance	
absorptance:	polygon_absorptance	

Polynet - Normalize

Function:

Normalizes polygon facet normals

Discussion:

This command should only be used after defining a closed solid polynet object. This command should not be used after defining a polynet mirror or other open shell polynet objects.

For a thick polynet object polynet/normalize must always follow polynet/finish.

The polynet/normalize must never appear before polynet/finish.

Corrects the orientation of each facet's local-surface-normal. Without this correction, ray-tracing will not function properly. If this command is used on an open polynet, an error message will be issued.

Parameters:

The name of the polynet object.

Text Command:

```
polynet/normalize my_polynet
```

Text Command Example:

```
polynet/normalize poly1
```

Note:

If the polynet object was generated by a CAD translator and you are not able to edit the file and insert a name, then you will not know the object name that OptiCAD will assign to the polynet object. In this case only: put the Polynet - Normalize command immediately after the polynet definition with no object name specified. OptiCAD will substitute the name of the last Polynet defined.

Example I - Extruded Polynet Object

An extruded polynet is simple and easy to specify. The following macro (extrude.mac) defines and places an extruded wedge-shaped object.

```
# start of macro named extrude.mac
echo/on
polynet/start my_wedge refl=0 abs=0 index=1.5
polynet/polygon
polynet/chain
polynet/point 5 0.25 0
polynet/point 5 -0.25 0
polynet/point -5 -1 0
polynet/point -5 1 0
polynet/thick 0.5
polynet/finish
polynet/normalize my_wedge
place my_wedge 0 0 0
set/work/snug
view/system
```

Example II - General Polynet Object

The macro below (C_Cube.mac) defines a corner cube, which is an example of a generalized polynet object.

```
# macro named c_cube.mac
echo/on
set/workbox 0 0 0 25 25 25
define/light light1
define/film film1 2 2
polynet/start c_cube refl=1.0 index=-1.0
polynet/polygon
polynet/chain
polynet/point 0 0 17.666667
polynet/point 0 25 0
polynet/point 21.650635 -12.5 0
polynet/close
polynet/polygon
polynet/chain
polynet/point 0 0 17.666667
polynet/point 21.650635 -12.5 0
polynet/point -21.650635 -12.5 0
polynet/close
polynet/polygon
polynet/chain
polynet/point 0 0 17.666667
polynet/point -21.650635 -12.5 0
```



```

polynet/point 0 25 0
polynet/close
polynet/finish
polynet/normalize c_cube
place c_cube 0 0 0
define/lens lens1 0 -10.2501 3 10 1.5
modify/alt/asph lens1.2 -.09756 -.6286 -6.8757E-6 1.3659E-9
define/lens lens2 0 -10.2501 3 10 1.5
modify/alt/asph lens2.2 -.09756 -.6286 -6.8757E-6 1.3659E-9
place lens1 -6 0 -10
place lens2 6 0 -10
place light1 -6 0 -30
place film1 6 0 -30
ray light1 4 4 15 15
set/work/snug
view/ray/system -115 80

```

Multiple Holes in a Polynet Object

Holes can be placed in a polynet shell object (infinitely thin surface) or in an extruded polynet object. The basic rule of such polynet objects is that the first chain of a polygon defines the outer limit and the second chain defines a hole. Subsequent chains define different things depending on where they are located. All chains that are totally separated from the second chain define additional holes. However, any chain that is completely inside a hole defines an island. No chain may overlap another chain, and no chain may exceed the outer extent as defined by the first chain. In the macro ZThick.mac, note that all letters are entirely cut-out holes, except for the letter A, which contains both a cut-out and an island.

End of Polynet Chapter

Raytrace

Overview Discussion:

Rays originate from a lightsource or a detector, which are user-defined and -named objects. A lightsource or detector is placed (initially with its +z-axis parallel to the global +z-axis) and then may be rotated. Ray bundles are centered about the +z-axis of the lightsource or detector. The user has independent control over the number of rays in x and y and over the limits on the ray bundle in x and y.

All ray-trace commands will trace a bundle of rays at all of the wavelengths that have a non-zero weight, see **Wavelengths and Weights** in the **Options** chapter.

There are two explicit types of ray bundles:

1. Rays originating from the origin of the lightsource or detector's local coordinate system (where you placed the object).
2. Collimated rays that are parallel to the local z-axis of the lightsource or detector.

Rays may also be launched from a standard (included) or user written function called a Dynamic Link Library (DLL). See the **Sources** section the **Object Properties** chapter.

Rays will ignore objects that are either not active (see **Activate** and **Deactivate** sections in the **Edit** chapter) or are not present in the workbox (see **Place** and **Remove** sections in the **Edit** chapter). Lightsource or detector's have no physical extent so they will not interact with rays.

The keyboard Escape key will interrupt most ray-trace calculations.

Note on ray termination controls:

The user has control over when a ray is terminated and the ray power is lost. The ray termination function is described under **RayTraceLimits** in the **Options** chapter. A ray will be terminated when the ray power falls below a user selected minimum power (mintrans). Care must be taken when setting this factor, especially when partially transmitting surfaces are present. It is possible that multiple rays will fall below the minimum ray power and be truncated and the results would be that far more power could be lost than the value of mintrans. It is recommended that an power conservation test case be run when radiometric accuracy is required.

Note on number of rays traced:

For the Oval option, the number of rays traced is the number of rays in the inscribed ellipse rather than the product of n_x and n_y . For circular bundle of rays requested, the number traced would be approximately $n_x * n_y * \pi/4$.

Note that the power_units are set using the **Windows** page.

Sobol Sampling

Monte Carlo sampling (the OptiCAD default) is a powerful tool for optical and illumination simulation. The disadvantage of Monte Carlo sampling is the slow convergence rate of

$$O[N^{-1/2}]$$

where N is the number of samples. In qualitative terms Monte Carlo sampling can be thought of as not caring what has happened before, and therefore has no way of knowing what regions need gaps in sampling to be filled.

Sobol sampling uses Low Discrepancy Sequences instead of random numbers as a deterministic alternative (Quasi-Monte Carlo) to Monte Carlo sampling. The upper bound of Sobol sampling is thought to be

$$O\left[\frac{(\log N)^d}{N}\right]$$

where d is the number of independent dimensions. Sobol sampling is continuously filling in gaps in the sample space. For large samples Sobol is proportional to 1/N which is a big improvement.

Sobol sampling can be activated for ray origins and directions from the Options menu and by the text commands

for spatial (ray origins) Sobol

```
set/sp_sobol/on  
set/sp_sobol/off
```

and for direction (ray angles) Sobol

```
set/di_sobol/on  
set/di_sobol/off
```

Sobol sampling is implemented for functions defined using the **Emitance Function** page of the **Object and Surfaces Properties Box** and most of the standard source dlls (\sources\standard\dlls). Dlls that are not Sobol compatible will produce an error message and default to Monte Carlo sampling.

Amplified Spontaneous Emissions (ASE)

A material having small signal (exponential) gain can be modeled by entering a negative value for the volumetric absorption coefficient. See the **Volume Absorption** section of the **Object Properties** chapter.

The total power truncated after falling below a threshold power or undergoing too many bounces, and the total power exiting the material is now displayed in the TextOut box with every raytrace. For a ray launched from inside a material, conditions of surface reflection, absorption, and scattering can now be estimated to prevent trapped rays from accumulated too much power.

Lightsource Raytracing

Rays launched from a lightsource carry power based on the used define initial total power and the number of rays traced. This power can be attenuated by absorption or ray splitting. Ray power can be recorded on a film or polynet and displayed on a radiometer.

Detector Raytracing

Rays launched from a detector look like lightsource rays. However the addition of a camera to a detector activates reverse ray-trace radiometry. The detector rays then act as travelling power buckets, accumulating power from surfaces having a non-zero irradiance.

A cone of detector rays may be launched from a point, or detector rays may be launched from surface or volume using any source DLL.

Then the total accumulated power of detector launched rays and the power of rays that intercepted the camera can be calculated.

Certain ray cone spacings are not permitted when performing radiometric backwards ray-tracing.

Collimated ray bundles are not permitted from a detector having a camera attached.

For DLLs having isotropic and volumetric ray distributions see the **Isotropic Emitting Point Source–Detector**, **Cylindrical Emitting Volume Source–Detector** and **Ellipsoidal Emitting Volume Source–Detector** sections of the **Object Properties** chapter.

For DLLs having equal direction cosine and area ray distributions see the **Flat Emitting Surface Source–Detector**, **Spherical Surface Source–Detector**, and **Cylindrical Emitting Surface Source–Detector** sections of the **Object Properties** chapter.

Camera and Reverse Raytrace Radiometry Results

Numerical Data:

The total power accumulated by the reverse ray-trace, the total power normalized by the detector area (DLL based source distributions only), the total power collected by the camera array, and the total camera array power normalized by the detector area (DLL based source distributions only) are displayed in the Text Out box at the bottom of the screen.

Camera Image

The camera array is displayed in a pop-up window by using the feature **View->View Camera Array** (View menu). The image can be resized and saved to a bitmap, normalized by a user entered value, and exported in a comma separated values format by right clicking the mouse while in the pop-up display window. An array value of 0.0 will be displayed as black if OpenGL solid is selected and white if OpenGL solid is not selected.

Add Rays

The Add Rays option works differently for a detector than for a lightsource. When the Add Rays option is on a detector accumulates power from multiple ray-traces in a similar manner to film in a camera under going multiple exposures. Repeating the same ray-trace a second time will double the total accumulated power reported in the **TextOut box** and in the camera array. See the **Add Rays** section of the **Options** chapter.

Cone

Function:

Traces rays originating from a **point**

Discussion:

The rays that are traced originate from the origin of the lightsource or detector's local coordinate system (determined by its location) and are centered about the lightsource or detector's + z local axis (determined by its orientation).

The ray angle in the x-direction is limited to 360 degrees (full angle), and the ray angle in the y-direction is limited to 180 degrees (full angle). If a larger angle is needed in the y-direction, the lightsource or detector can be rotated about its local z-axis by 90°.

Note: the only ray bundle shape available with the Raytrace → Cone feature is rectangular; this differs from the Raytrace → Collimated feature, which allows a choice of either rectangular or elliptically shaped ray bundles. In order to create an elliptically shaped ray bundle, you must define an aperture of appropriate parameters (oval hole designation) and place the aperture (relative to the lightsource or detector) a small distance in front of the lightsource or detector. This procedure clips the corners of the rectangular ray bundle. You should also verify the number of rays that get through the aperture, for use in energy calculations.

The ray-trace text modifiers **new** and **yessave** are defaults and need not be entered.

RaytraceCone Parameters:

A lightsource or detector name, the number of rays in x (nx), the number of rays in y (ny), the maximum full ray angle in x (hfov), the maximum full ray angle in y (vfov), and a ray spacing selection.

RaytraceNewCone			
		Object name:	Cancel
		my_light	OK
nx	rays_in_x		
ny	rays_in_y		
hfov	x_full_angle		
vfov	y_full_angle		

FullAngle

Function:

Ray **bundle fills** angle **space**

Discussion:

This option fills the angle space specified, with rays that are equally spaced in angles, starting and ending at the extremes of the angle space.

Detector Note:

Full Angle ray spacing is not allowed for a detector having an attached camera.

Text Command:

```
raytrace/cone/full light_name nx ny hfov vfov
```

Text Command Examples:

```
raytrace/cone/full light1 10 10 30 30  
raytrace/new/cone/full/yessave light1 10 10 30 30
```

Eqsolid

Function:

Rays spaced with **equal solid angle**

Discussion:

This option spaces rays such that each ray represents an equal area on a sphere. This is accomplished by spacing the rays equally in the x-angle and equally in the sine of the y-angle. While the method is correct, better results (and better sampling) are achieved by using the diffuse microsphere method described above.

Detector Note:

Equal Solid Angle ray spacing, while allowed for a detector having an attached camera is not the recommended. While a isotropic ray distribution is a good choice for a spherical detector whose size can be neglected. A better choice for launching rays into an isotropic ray distribution from a point would be to use the DLL described in the **Isotropic Emitting Point Source–Detector** section of the **Object Properties** chapter. For volumetric ray distributions see the **Cylindrical Emitting Volume Source–Detector** and **Ellipsoidal Emitting Volume Source–Detector** sections of the **Object Properties** chapter.

Text Command:

```
raytrace/cone/eqsolid light_name nx ny hfov vfov  
raytrace/cone/eqsolid detector_name nx ny hfov vfov
```

Text Command Examples:

```
raytrace/cone/eqsolid light2 100 100 360 180  
raytrace/new/cone/eqsolid/yessave light1 10 10 30 30  
raytrace/cone/eqsolid detector2 100 100 360 180  
raytrace/new/cone/eqsolid/yessave detector1 10 10 30 30
```

EqAngle

Function:

Rays spaced with **equal angle**

Discussion:

This option underfills the specified angle space with rays that are equally spaced in angle. This option is not in common usage.

Detector Note:

Equal Angle ray spacing is not allowed for a detector having an attached camera.

Text Command:

```
raytrace/cone/eqang light_name nx ny hfov vfov
```

Text Command Examples:

```
raytrace/cone/eqang light1 10 10 30 30  
raytrace/new/cone/eqang/yessave light1 10 10 30 30
```

Tangent

Function:

Rays spaced in the **tangent space**

Discussion:

This option spaces rays to be equally spaced on a plane perpendicular to the lightsource or detector's local z-axis.

Detector Note:

Tangent ray spacing is only allowed for a flat detector having an attached camera.

This is a useful option for a flat detector when the number of rays traced in x and y are equal to the number of camera array elements in x and y. The result is that a ray when projected onto the camera focal array intersects the center of each camera array element. While this option does not provide the optimal sampling for reverse ray-trace radiometry, it does produce a picture (view of the camera array) for which every pixel (array element) is sampled.

Text Command:

```
raytrace/cone/tangent light_name nx ny hfov vfov  
raytrace/cone/tangent detector_name nx ny hfov vfov
```

Text Command Examples:

```
raytrace/cone/eqtangent light1 10 10 30 30  
raytrace/new/cone/eqtangent/yessave light1 10 10 30 30  
raytrace/cone/eqtangent detector1 10 10 30 30  
raytrace/new/cone/eqtangent/yessave detector1 10 10 30 30
```

Direction Cosine

Function:

Rays spaced in **direction cosine space**

Discussion:

This option spaces rays uniformly in direction cosine space.

Detector Note:

Launching rays spaced uniformly in direction cosine space is a good choice for a camera having a flat detector where the extent of the detector can be neglected. For area ray distributions see the **Flat Emitting Surface Source–Detector**, **Spherical Surface Source–Detector**, and **Cylindrical Emitting Surface Source–Detector** sections of the **Object Properties** chapter.

Text Command:


```
raytrace/cone/eqdcos light_name nx ny hfov vfov
raytrace/cone/eqdcos detector_name nx ny hfov vfov
```

Text Command Examples:

```
raytrace/cone/eqdcos light1 10 10 30 30
raytrace/new/cone/eqdcos/yessave light1 10 10 30 30
raytrace/cone/eqdcos detector1 10 10 30 30
raytrace/new/cone/eqdcos/yessave detector1 10 10 30 30
```

Collimated

Function:

Traces **collimated rays**

Discussion:

Traces a rectangular or elliptical bundle of collimated rays. Collimated rays do not originate at infinity, but rather originate at the local x-y plane of the lightsource and travel in the + z direction of the lightsource's local coordinate system. Care must be taken to ensure that all objects (or parts of objects) that the collimated bundle is intended to hit, lie on the + z side of the lightsource's x-y plane.

RaytraceCollimated Parameters:

A lightsource name, the number of rays in x (nx), the number of rays in y (ny), the maximum beam half-size in x (hhwidth), the maximum beam half-size in y (vhwidth), and a ray bundle shape selection.

RaytraceNewCollimated			
		Object name:	Cancel
		my_light	OK
nx	rays_in_x		
ny	rays_in_y		
hhwidth	x_half_size		
vhwidth	y_half_size		

Detector Note:

Collimated bundle ray spacing is not allowed for a detector having an attached camera.

Rectangular

Function:

Collimated **rectangularly shaped beam**

Discussion:

Traces an rectangular bundle of collimated rays that are equally spaced in x and in y.

Text Command:

```
raytrace/collimated/rectangle light_name nx ny hhwidth vhwidth
```

Text Command Examples:

```
raytrace/collimated/rectangle light1 20 20 1 1  
raytrace/new/collimated/rectangle/yessave light1 20 20 1 1
```

Oval

Function:

Collimated **elliptically shaped beam**

Discussion:

Traces an elliptical bundle of collimated rays that are equally spaced in x and in y.

Text Command:

```
raytrace/collimated/oval light_name nx ny hhwidth vhwidth
```

Text Command Example:

```
raytrace/collimated/oval light1 100 10 10 1  
raytrace/new/collimated/oval/yessave light1 20 20 1 1
```

Extended

Function:

Traces rays originating from an **extended/custom, predefined, or user defined, or IESNA, or tabular source**

Discussion:

The location and direction of traced rays are controlled by a DLL, which may be either included on the OptiCAD CD, or written by the user (see the section **Custom Sources** in the **Design** chapter),

The ray bundle in the x-direction is limited to 360 degrees (full angle), and the ray bundle in the y-direction is limited to 180 degrees (full angle). For x-direction angles greater than 180 degrees the excluded cone of rays will be made symmetric. Angular input errors will default to be a full sphere (if physically real).

The ray angle limiting parameters (hfov and vfov) limit the angular ray bundle but not the total number of rays traced. The user may wish to weight the results of the ray-trace by the fraction of the source power in the ray bundle that is actually traced, rather than by the total source power (as would be the case if all possible rays were traced).

Parameters:

The name of a lightsource or detector that has been previously modified to reference a DLL (Dynamic Link Library), the total number of rays to be traced (which for consistency is the product of the values for nx and ny which have no separate meaning), the maximum full ray angle in x (hfov), the maximum full ray angle in y (vfov), and a ray limiting shape (either oval or rectangle).

RaytraceNewCustom			
		Object name:	Cancel
		my_light	OK
nx	rays		
ny	1		
hfov	x_full_angle		
vfov	y_full_angle		

Rectangular

Function:

Extended/custom source **rectangularly shaped beam**

Discussion:

Traces an rectangular bundle of rays from an extended/custom lightsource or detector. The total number of rays to be trace is the product of nx and ny.

Text Commands:

```
raytrace/custom/rect light_name 1 nrays hfov vfov
raytrace/custom/rect light_name nrays 1 hfov vfov
```

Text Command Examples:

```
raytrace/custom/rect light1 1 100 360 180
raytrace/custom/rect light1 100 1 180 180
raytrace/new/custom/rect/yessave light1 1 100 360 180
raytrace/new/custom/rect/yessave light1 100 1 180 180
```

Oval

Function:

Extended/custom source **elliptically shaped beam**

Discussion:

Traces an rectangular bundle of rays from an extended/custom lightsource or detector. The total number of rays to be trace is the product of nx and ny.

Text Commands:

```
raytrace/custom/oval light_name 1 nrays hfov vfov
raytrace/custom/oval light_name nrays 1 hfov vfov
```

Text Command Examples:

```

raytrace/custom/oval light1 1 100 360 180
raytrace/custom/oval light1 100 1 180 180
raytrace/new/custom/oval/yessave light1 1 100 360 180
raytrace/new/custom/oval/yessave light1 100 1 180 180

```

Function

Function:

Traces rays originating from an **emittance function** created on the **Emitance Function** page of the **OSB**.

Discussion:

The location and direction of traced rays are controlled by either a spatial and an angular function, or a bitmap and an angular function defined. See the **Emitance Function** page section of the **Object and Surface Properties** box chapter.

The ray bundle is limited to a full angle of 360 degrees in the x-direction, and 180 degrees in the y-direction. For x-direction angles greater than 180 degrees the excluded cone of rays will be made symmetric. Angular input errors will default to be a full sphere (if physically real).

The ray angle limiting parameters (hfov and vfov) limit the angular ray bundle but not the total number of rays traced. The user may wish to weight the results of the ray-trace by the fraction of the source power in the ray bundle that is actually traced, rather than by the total source power (as would be the case if all possible rays were traced).

Parameters:

The name of a lightsource that has been previously modified by an emittance function, the total number of rays to be traced (which for consistency is the product of the values for nx and ny which have no separate meaning), the maximum full ray angle in x (hfov), the maximum full ray angle in y (vfov), and a ray limiting shape (either oval or rectangle).

RaytraceNewCustom			
		Object name:	Cancel
		my_light	OK
nx	rays		
ny	1		
hfov	x_full_angle		
vfov	y_full_angle		

Rectangular

Function:

Extended/custom source **rectangularly shaped beam**

Discussion:

Traces an rectangular bundle of rays from an extended/custom lightsource or detector. The total number of rays to be trace is the product of nx and ny.

Text Commands:

```
raytrace/custom/rect light_name 1 nrays hfov vfov  
raytrace/custom/rect light_name nrays 1 hfov vfov
```

Text Command Examples:

```
raytrace/custom/rect light1 1 100 360 180  
raytrace/custom/rect light1 100 1 180 180  
raytrace/new/custom/rect/yessave light1 1 100 360 180  
raytrace/new/custom/rect/yessave light1 100 1 180 180
```

Oval

Function:

Extended/custom source **elliptically shaped beam**

Discussion:

Traces an rectangular bundle of rays from an extended/custom lightsource or detector. The total number of rays to be trace is the product of nx and ny.

Text Commands:

```
raytrace/custom/oval light_name 1 nrays hfov vfov  
raytrace/custom/oval light_name nrays 1 hfov vfov
```

Text Command Examples:

```
raytrace/custom/oval light1 1 100 360 180  
raytrace/custom/oval light1 100 1 180 180  
raytrace/new/custom/oval/yessave light1 1 100 360 180  
raytrace/new/custom/oval/yessave light1 100 1 180 180
```

Energy

Function:

Integrates ray **energy** over the film plane

Ray Energy Note:

Ray energy is generally conserved by OptiCAD. Each ray starts out with an energy of 1.0. The software keeps track of the energy of each ray as it propagates, and ray energy is reduced by surface and volumetric absorption.

When a parent ray divides (splits) at a partially transmitting and partially reflecting surface, the ray energy is distributed to the descendent rays.

Ray energy is arbitrarily lost only when a ray fails one of the user-specified tests set by the RayTraceLimits feature. In such cases (energy falls below a threshold, too many surfaces are hit, or too high a ghost order is encountered) the ray is truncated.

Energy is not conserved if the volumetric absorption is used as an exponential gain.

Exported data format note:

Numerical data from the Energy function may be exported to an external file or copied to the clipboard. The exported data is in the format of comma separated values (csv). The exported data can be imported into a spreadsheet (which can put each numerical value in a separate cell) or viewed using a word processor. Pasting such data into a text editor is not recommended, as the commas and embedded carriage returns may cause confusion.

Discussion:

Ray energy on the film plane includes the effects of reflections, transmissions, and absorptions along the path of the ray. The energy feature integrates ray energy over the film plane and produces a 1-D function (either linear or radial) of energy versus slit position or aperture size.

The result of the film plane integration can be presented in various normalized forms, displayed graphically (using the View → Type → Energy feature), and written to a data file as comma-separated values (which can then be read into an external application such as a spreadsheet).

The integration of energy over a slit or aperture is only meaningful when the slit or aperture is completely contained in the film plane. Care should be used when performing an enslitted energy calculation over an oval film plane or when performing an encircled energy calculation over a noncircular film plane. The area used in energy density calculations does not consider whether the slit or aperture extends beyond the extent of the film plane. The user may wish to use the Radiometry feature in cases when the geometry is complex.

XLinear

Function:

Integrates **energy** with a **slit**

Discussion:

Defines a temporary slit in the x-y plane. The slit is oriented parallel to the y-axis. The (relative) x-width is user-defined, and the y-extent equals the y-extent of a user-specified film plane. The slit is essentially scanned across the film plane along the x-axis, and the integrated energy (Enslitted Energy) is reported as a function of the slit's x-position. The actual numerical operation involves sorting rays into bins the width of the slit. The expression for the actual entry parameter nbins is:

$$nbins = \frac{2 \cdot xhw}{xslitwidth} = \frac{xfullwidth}{xslitwidth}$$

where xhw is the half-size in x of the film plane, xslitwidth is the width of a bin (slit), and xfullwidth is the full width of the film plane in x.

There is a difference between the text command and menu implementation of this feature. The text command can only be used to perform the energy calculation and send the results to a user-defined file. The menu function sends the results to the TextOut box. The results are available for display (using the View → Type → Energy feature). In order to send energy results to a file using menu options, enter a file name in the EnergyFile box (described in this section).

Note that this function is only operational after a film plane actually contains rays.

Note that you can now append data to an existing energy file by putting a plus sign (+) in front of the file name.

Parameters:

The name of a film object, the number of bins in x (nbins), and an output file name.

EnergyXLinear			
		Object name:	Cancel
		my_film	OK
nbins	number_of_x_bins		

Text Commands:

```
energy/xlinear film_name nbins
energy/xlinear/filename="energy_file_name" film_name nbins
energy/xlinear/filename="+energy_file_name" film_name nbins
```

Text Command Examples:

```
energy/xlinear film1 20
energy/xlinear/filename="energy_out.csv" film1 20
energy/xlinear/filename="+energy_out.csv" film1 20
```

YLinear

Function:

Integrates **energy** with a **slit**

Discussion:

Defines a temporary slit in the x-y plane. The slit is oriented parallel to the x-axis. The (relative) y-width is user-defined, and the x-extent equals the x-extent of a user-specified film plane. The slit is essentially scanned across the film plane along the y-axis, and the integrated energy (Enslitted Energy) is reported as a function of the slit's y-position. The actual numerical operation involves sorting rays into bins the width of the slit. The expression for the actual entry parameter nbins is:

$$nbins = \frac{2 \cdot yhw}{y_{slitwidth}} = \frac{y_{fullwidth}}{y_{slitwidth}}$$

where yhw is the half-size in y of the film plane, yslitwidth is the width of a bin (slit), and yfullwidth is the full width of the film plane in y.

There is a difference between the text command and menu implementation of this feature. The text command can only be used to perform the energy calculation and send the results to a user-defined file. The menu function sends the results to the TextOut box. The results are available for displaying (using the View → Type → Energy feature). In order to send energy results to a file using menu options, enter a file name in the EnergyFile box (described in this section).

Note that this function is only operational after a film plane actually contains rays.

Parameters:

The name of a film object, the number of bins in y (nbins), and an output file name.

Note that you can now append data to an existing energy file by putting a plus sign (+) in front of the file name.

EnergyYLinear			
		Object name:	Cancel
		my_film	OK
nbins	number_of_y_bins		

Text Commands:

```
energy/ylinear film_name nbins
energy/ylinear/filename="energy_file_name" film_name nbins
energy/ylinear/filename="+energy_file_name" film_name nbins
```

Text Command Examples:

```
energy/ylinear film1 20
energy/ylinear/filename="energy_out.csv" film1 20
energy/ylinear/filename="+energy_out.csv" film1 20
```

Encircled

Function:

Integrates **energy** within a **circular aperture**

Discussion:

Defines a temporary circular aperture in the x-y plane. The aperture is incrementally increased in radius, and the integrated (and accumulated) energies are reported as a function of the aperture radius. The radial increments are essentially user-defined, as is the maximum radius. The actual numerical operation involves sorting rays into radial bins. The expression for the actual entry parameter nbuckets is:

$$nbuckets = \frac{radiusmax}{deltaradius}$$

where radiusmax is the radial extent of the integration, and deltaradius is the radial increment of the integration.

The aperture may only be located at the center of the film plane.

There is a difference between the text command and menu implementation of this feature. The text command can only be used to perform the energy calculation and send the results to a user-defined file. The menu function sends the results to the TextOut box, the results are available for displaying (using the View → Energy feature). In order to send energy results to a file using menu options, enter a file name in the EnergyFile box (described in this section).

Note that this function is only operational after a film plane actually contains rays.

Note that the energy density data column is the accumulated (encircled) energy divided by the accumulated area.

Parameters:

The name of a film object, the maximum radius of the integration, the number of radial buckets (nbuckets), and an output file name.

Note that you can now append data to an existing energy file by putting a plus sign (+) in front of the file name.

EnergyEncircled			
		Object name:	Cancel
		my_film	OK
radiusmax	maximum_integration_radius		
nbuckets	number_of_radial_buckets		

Text Commands:

```
energy/encircled film_name radiusmax nbuckets
energy/encircled/filename="energy_file_name" radiusmax nbuckets
energy/encircled/filename="+energy_file_name" radiusmax nbuckets
```

Text Command Examples

```
energy/encircled film1 1 20
energy/encircled/filename="energy_out.csv" film1 1 20
energy/encircled/filename="+energy_out.csv" film1 1 20
```

RepeatLast

Function:

Repeats the last energy calculation

Discussion:

This feature can also be implemented with a toolbar button of the same name.

Preferences

Function:

Determines **energy normalization** mode

Discussion:

To avoid any confusion, it is recommended that you set the first two preferences as follows: Percent should always be unselected (unchecked), and Density should always be selected (checked).

These choices will always report energy density (ray energy normalized by the area of the scanning slit).

The important choice is whether to normalize the energy density by the total number of rays traced or by the total ray energy actually incident on the film plane.

To normalize the energy density by the total number of rays traced, select the Energy Preference option: Absolute.

To normalize the energy density by the total ray energy actually incident on the film plane, select the Energy Preference option: Relative.

Text Command Examples:

```
set/energy/absolute  
set/energy/relative  
set/energy/percentage  
set/energy/density
```

EnergyFile

Function:

Writes energy **data to a file**

Discussion:

Energy data is written to a user-defined file. The output format is comma-separated values (csv), which can be read into a spreadsheet or other application. The EnergyFile title is checked when a file name has been selected. To select a file name when the EnergyFile title is unchecked, click on EnergyFile and enter a file name. To clear a file name from EnergyFile, click on the checked title, which will clear the current file name. Note: This option overwrites an existing data file of the same name (without warning).

If you select Analyze → Energy → EnergyFile and then do not enter a file name, future Energy calculations will result in an error message. You can correct the situation by clicking EnergyFile a second time, which deselects the save to a file option.

Note that you can now append data to an existing energy file by putting a plus sign (+) in front of the file name.

Parameters:

A file name must be entered (if you are going from an unchecked status to a checked status), if one is requested.

EnergyFile	
File name	Cancel
energy_out.csv	OK

Text Command:

See text commands and examples for each type of Energy option.

Display

Rays

Function:

Displays **ray paths**

Discussion:

Displays a table of the ray–surface intercept and direction cosines for each ray from the last ray–trace. Information is displayed in the TextOut box at the bottom of the screen.

A ray's path is listed until a ray ends or divides. If a ray divides, the primary ray path is listed until it ends, then the ghost ray path listing is resumed. The ray data can be sent to a text file using the File → SaveRaysAsText feature. Note that the direction cosines are not explicitly saved, but rather recalculated from ray intercept data. This will result in a loss of accuracy for extremely small ray paths.

Spot

Spot - Directions

Function:

Displays a table of the direction cosines, and relative **ray** energy for each ray hitting the **film** plane

Spot - Locations

Function:

Displays a table of the intercept, direction cosines, and relative ray energy for each **ray** hitting the **film** plane.

Object

Function:

Displays a table of object names, type, some status information.

Altsurfaces

Function:

Displays a table of some AltSurface data.

Extents

Function:

Displays a table of the x, y, and z limits for each surface

Geometry

Function:

Displays a table of the location and orientation of the center of the local coordinate systems of all objects. Information is displayed as x, y, z locations, and x, y, and z rotation angles

Surface

Function:

Displays a table of the reflectance, absorptance, index of refraction, and volumetric absorptance.

Radiometer

Function:

2-D Irradiance, intensity, or radiance distribution

Discussion:

A radiometer is a 2-D array of power bins that is stored in memory. A radiometer is fundamentally different from a film plane in that it only stores power accumulation bins; it does not store the location and power of each ray. Using a radiometer has the following advantages: a) provides a 2-D irradiance, intensity, or radiance map on a film or several other surface types; b) allows any number of rays (millions) to be traced (with SaveSpots and SaveRays both off), as you are not limited by a film file (spots.dat) filling your disk; c) saves run time (with SaveSpots and SaveRays both off), as you are not wasting time while film data is written to disk; and d) allows 2-D irradiance, intensity, or radiance information to be externally analyzed, as data can be written to a file as comma-separated values (csv formatted data can be read into a spread sheet or other application). See section **Radiometer** → **SaveToFile** in this chapter. The limitation of a radiometer is that once the rays are accumulated in bins, all location and power information for individual rays is lost.

The radiometer clears itself at the start of a ray–trace unless the Add_Rays option is set to be on, in this regard the radiometer works in a similar manner to a film plane.

Note that for ray power to be added to a radiometer it must be explicitly attached to a film, polynet, or other allowed object before the raytrace is performed.

Exported data format note:

Numerical data from the Radiometer function may be exported to an external file or copied to the clipboard. The exported data is in the format of comma separated values (csv). The exported data can be imported into a spreadsheet (which can put each numerical value in a separate cell) or viewed using a word processor. Pasting such data into a text editor is not recommended, as the commas and embedded carriage returns may cause confusion.

Conventional radiometer

A Radiometer may be attached to a previously defined film, mirror, lens, cylindrical lens, or rod. For a lens, cylindrical lens, or rod the radiometer only records rays hitting surface one. The radiometer has three ray power sorting options.

Irradiance radiometer

For the irradiance option rays are sorted into bins using the x and y ray locations on the surface, the z location is not used. The bins have equal area in the vertex x-y plane of the surface's local coordinate system. The total power in each bin is normalized by the area of the bin, the results are displayed in units of power/area (watts/cm² or lumens/m²).

Intensity radiometer

For the intensity option rays are first projected from the radiometer surface to a (virtual) hemi-spherical surface (reference sphere). The reference sphere is located tangent to the vertex x-y plane of the radiometer surface's local coordinate system and the center of curvature of the reference sphere is located on the z-axis of the radiometer surface's local coordinate system at a user defined distance (radius of the reference sphere).

Rays are sorted into bins of equal $\Delta\theta_X$ by $\Delta\theta_Y$, where the angles are with respect to the z-axis (optical axis). Note that the angles used are based on the ray intersection point on the reference sphere, and are not the actual ray directions.

This option models a source located on an x-y rotation table and a detector located a fixed distance (the radius of the reference sphere) from the center of rotation of the table.

To avoid confusion, the radius of the reference sphere is named 'focal length'.

The total ray power in each bin is normalized by the solid angle of that bin, the results are displayed in units of power/steradian (watts/steradian or lumens/steradian).

The results are a near- or far-field intensity, depending on the choice of radius for the reference sphere and the size of the source. The center of the reference sphere would generally be placed at the center of the source. Therefore the radius of the reference sphere would be the distance between the source and the radiometer (detector) surface.

A useful choice of a surface for an intensity radiometer would be a spherical surface (such as a spherical mirror with reflectance set to 1.0) centered about the source. The radius of the surface would be set equal to the radius (labeled focal_length) value chosen for the intensity radiometer.

Radiance radiometer

For the radiance option rays are sorted into bins of equal $\Delta\theta_X$ by $\Delta\theta_Y$, where the angles are with respect to the z-axis (optical axis) of the radiometer's local coordinate system.

This option models a source located on an x-y rotation table and a detector located an infinite distance from the source. This is a true far-field calculation.

For the radiance option the actual ray angles (in the radiometer's local coordinate system) are used and the ray location on the radiometer is not used.

The total ray power in each bin is normalized by the solid angle of that bin and the area of the vertex x-y plane of the radiometer surface, the results are displayed in units of power/steradian/area (watts/steradian/cm² or lumens/steradian/m²).

A useful choice of a surface for a radiance radiometer would be a spherical surface (such as a spherical mirror with reflectance set to 1.0) centered about the source.

Viewing a conventional radiometer

The contents of a conventional radiometer can be viewed using View → Type → Radiometer feature or by clicking the ViewRadiometerMap toolbar button (a white blob on a black background). The view of the distribution can be changed from a gray scale to color coded (spectrum) using View -> Palettes. See the View chapter.

Polynet radiometer

A Radiometer may be attached to a previously defined polynet shell object, in which case will have the geometric properties of that polynet object. Each facet of the polynet object will be treated as an power collection bin.

The value in the Radiometer array will be the total ray power incident on each facet of the polynet object, normalized by the area of that facet. The results are displayed in units of power/area (watts/cm² or lumens/m²).

The numerical values in a polynet radiometer array may be exported to a file using the FileOut option or copied to the clipboard using the TextOut option.

The polynet radiometer does not make use of the Energy → Preferences selections.

Note that only an irradiance radiometer may be attached to a polynet.

Notes on using a radiometer

1. In most cases the AutoRetrace option should be set off when using any of the radiometer features.
2. The ViewRadianceMap toolbar button is only activated after you have selected an object with a radiometer attached from the OptiCAD Objects box.
3. The user is advised, as a matter of good practice (to prevent memory fragmentation), to exit and restart the OptiCAD program before performing extremely long calculations.
4. Changing the active film plane does not automatically change the displayed irradiance map. In order to display the new map click the ViewRadiometerMap toolbar button.
5. Not Saving the Ray–trace Data:
 - a) In order to not save the ray paths to a file, deselect (click to uncheck mark) SaveRays from the Options menu, or from a macro include the line set/saveraysoff.
 - b) In order to resume saving the ray paths to a file, select (click to check mark) SaveRays from the Options menu, or from a macro include the line set/saverays/on.
6. Not Saving the Ray–trace Data and Not Saving the Spot Diagram Data:
 - a) In order to not save the ray paths to a file and not save the spot diagram data to a file, deselect (click to uncheck mark) SaveSpots from the Options menu and deselect (click to uncheck mark) SaveRays from the Options menu, or from a macro include the lines set/saverays/off and set/saverays/on.
 - b) In order to resume saving the ray paths to a file and resume saving the spot diagram data to a file, select (click to check mark) SaveSpots from the Options menu and select (click to check mark) SaveRays from the Options menu, or from a macro include the lines set/saverays/off and set/saverays/on.

Polynet radiometers and conventional irradiance radiometers are displayed (in false color) in the system (perspective) view when the Open GL Solid option is selected.

7. If the option to save spot data has been selected (Options->Save_Spots is check marked), data on a radiometer will be automatically resorted into the selected number of bins on the radiometer when the number of bins in the radiometer has been changed.

8. When the status of an objects contribution to radiometers has been changed, the radiometer should be explicitly rebinned using the **Radiometer->Rebin function**. See **contribute directly to radiometer** selection on the **Placement** page.

9. If the message "Unable to stretch bits" appears while attempting to view a Radiometer, go to the Control Panel and then select Display, reset the screen resolution the next lower resolution, making sure that the setting has at least 256 bits per pixel.

10. Attaching a radiometer to a surface already having a radiometer replaces (and zeros) the existing radiometer.

11. When attaching a radiometer to a polygonally faceted surface the product of nx and ny must be greater than the number of facets. Also note that the data is stored in the radiometer array in the exact order that the facets were defined, and that the View Radiometer Map toolbar button will only produce a meaningful display for a rectangular polygonally faceted mesh. Also note that the ray power is normalized by the area of each facet before being stored in the radiometer array.

Attach Irradiance Radiometer

Parameters:

The name of an existing object and the number of bins in x and y.

RadiometerAttach			
		Object name:	Cancel
		object	OK
nx	number of bins x		
ny	number of bins y		

Text Command:

```
radiometer/attach_irradiance object_name nx ny
```

Text Command Examples:

```
radiometer/attach_irradiance film2 48 48
rad/attach film2 48 48
```

Attach Intensity Radiometer

Parameters:

The name of an existing object and the number of bins in x and y.

RadiometerAttach			
		Object name:	Cancel
		object	OK
nx	number of bins x		
ny	number of bins y		
thetamaxX	maximum half angle x		
thetamaxY	maximum half angle y		
fl	reference sphere radius		

Text Command:

```
radiometer/intensity_attach object_name nx ny
```

Text Command Examples:

```
radiometer/intensity_attach film2 48 48  
rad/attach film2 48 48
```

Attach Radiance Radiometer

Parameters:

The name of an existing object and the number of bins in x and y.

RadiometerAttach			
		Object name:	Cancel
		object	OK
nx	number of bins x		
ny	number of bins y		
thetamaxX	maximum half angle x		
thetamaxY	maximum half angle y		

Text Command:

```
radiometer/radiance_attach object_name nx ny
```

Text Command Examples:


```
radiometer/radiance film2 48 48
rad/attach film2 48 48
```

Minimum/Maximum Display Settings

Function:

Limits values displayed on **Radiometer**

Discussion:

Values below Min are displayed as 0.0 (minimum color palette value), values above max are displayed as max (maximum color palette value).

Parameter:

The name of the object having a radiometer attached, minimum and maximum values to be display (in radiometer units).

Text Command:

```
radiometer/limits object_name min max
```

Text Command Example:

```
radiometer/limits film2 10 100
```

Smoothing Setting

Function:

Smooths the **Radiometer Display**

Discussion:

When a system is analyzed by tracing randomly distributed rays, a fluctuation in the displayed radiometer results will occur. This sampling noise is not physically real and can be eliminated by tracing a very large number of rays.

The smoothing function convolves the radiometer display with a gaussian function, to better simulate the results of tracing an infinite number of rays.

The smoothing function should be used with caution, and an input gaussian width of one or two (cell sizes) should be sufficient.

The Smooth feature may only be used with a film radiometer.

Parameter:

The name of the object having a radiometer attached and the width of the smoothing kernel (in cell sizes).

Text Command:

```
radiometer/smooth object_name smooth_factor
```

Text Command Example:

```
radiometer/smooth film2 1
```

Clear (reset)

Function:

Clears the Radiometer

Discussion:

Resets the numerical values in a radiometer to be zero. This function does not remove the radiometer, or change the dimensions of the array of bins. Resetting (zeroing) a radiometer attached to 'film1' does not automatically reset (zero) a radiometer attached to 'film2'.

Parameter:

The name of the object having a radiometer attached.

Text Command:

```
radiometer/reset object_name
```

Text Command Example:

```
radiometer/reset film2
```

Remove

Function:

Removes a radiometer from an object

Discussion:

Removes (unattaches) a radiometer from an object. The radiometer array of bins is zeroed (memory is released).

Note that removing a radiometer attached to 'object1' does not automatically remove a radiometer attached to 'object2'.

Parameter:

The name of the object having a radiometer attached.

Text Command:

```
radiometer/remove object_name
```

Text Command Example:

```
radiometer/remove film3
```

TextOut

Function:

Displays Radiometer data in the **TextOut box**

Discussion:

Radiometry data is written to the TextOut box as comma-separated values (csv). The data could then be copied to the Windows clipboard and then read into another application.

Displaying radiometry data to the TextOut box does not reset (zero) the radiometer. Resetting a radiometer must be performed explicitly using the Radiometer → Clear feature.

Note that for a polynet radiometer the TextOut feature normalizes the power by the area of each facet.

Parameter:

The name of the object having a radiometer attached.

Text Command:

```
radiometer/textout object_name
```

Text Command Example:

```
radiometer/textout film3
```

SaveToFile

Function:

Saves contents of a **radiometer** to a file

Discussion:

The contents of a radiometer are saved to a user-defined file as comma-separated values (csv), which can be read into a spreadsheet or other application.

Saving radiometer data to a file does not reset (zero) the radiometer. Resetting a radiometer must be performed explicitly using the Radiometer → Clear feature.

Note that you can append data to an existing radiometry file by putting a plus sign (+) in front of the file name

Parameters:

The name of the object having a radiometer attached and the name of the file to receive the radiometry data.

Text Command:

```
radiometer/fileout/name="output_file_name" object_name  
radiometer/fileout/name="+output_file_name" object_name
```

Text Command Example:

```
radiometer/fileout/name="rad_out.csv" film3  
radiometer/fileout/name="+rad_out.csv" film3
```

Rebin

Function:

Rebins Saved Spot Data onto the **Radiometer**

Discussion:

If the option to save spot data has been selected (Options->Save_Spots is check marked), data on a surface with a radiometer attached will be resorted into the selected number of bins on the radiometer.

This is done automatically when the number of bins in the radiometer is changed. But should be explicitly done when the status of an objects contribution to radiometers has been changed. See **Contribute directly to radiometer** selection on the **Placement** page.

The Rebin feature may only be used with a film radiometer.

Parameter:

The name of the object having a radiometer attached.

Text Command:

```
radiometer/rebin object_name
```

Text Command Example:

```
radiometer/rebin film2
```

View Radiometer

Function:

Displays results of a **Radiometer** calculation

Discussion:

The units displayed are shown below:

Irradiance radiometer displays power per area (watts/cm² or lumens/m²).

Intensity radiometer displays power per steradian.

Radiance radiometer displays power per area per steradian (watts/cm²/steradian or lumens/m²/steradian).

The menu options available using **View -> Palette** controls the radiometer palette display. See the **Radiometer Palettes** section of this chapter.

Polynet and film plane irradiance radiometers are displayed (in false color) in the system (perspective) view.

If the message "Unable to stretch bits" appears while attempting to view a Radiometer, go to the Control Panel and then select Display, reset the screen resolution the next lower resolution, making sure that the setting has at least 256 bits per pixel.

Viewing the Radiometer is accomplished by selecting an object with a radiometer attached and then clicking on the **View Radiometer Map** toolbar button.

Text Commands:

```
view/radiometer object
```

```
view/irradiance object
```

```
view/intensity object
```

Text Command Examples:

```
view/radiometer sphere1
```

```
view/irradiance film1
```

```
view/intensity sphere1
```

COATINGS AND POLARIZATION

INTRODUCTION

Polarization analysis is an extension of conventional ray tracing which considers the effect that optical coatings and fresnel reflections have on the transport of light.

Polarization analysis is not the default condition and must be explicitly activated.

Note that polarization (including initial polarization) and coatings are specified in local coordinates.

The amplitude and polarization state of the electric field is described by the Jones vector:

$$E = \begin{bmatrix} E_x \\ E_y \end{bmatrix}$$

where E_x and E_y are complex-valued. The x and y directions of the electric field vector are orthogonal to the propagation vector of the ray.

At a boundary between two media the electric field is divided into E_S and E_P , both of which are complex valued. The transmittance, reflectance and phase of the electric field at the boundary is different for the S and P components of the field. The S component of the field is the projection along the axis that lies orthogonal to the plane of incidence, while the P component is the projection that lies within the plane of incidence. The plane of incidence contains both the ray propagation vector and the surface normal vector at the intercept point.

The electric fields just after a surface are computed using the S and P transmittance (for the transmitted ray) are

$$E'_S = E_S \cdot \tau_S \quad E'_P = E_P \cdot \tau_P$$

and using the S and P reflectance (for the reflected ray) are

$$E'_S = E_S \cdot \rho_S \quad E'_P = E_P \cdot \rho_P$$

where the transmittance and reflectance coefficients τ_S , τ_P , ρ_S , and ρ_P are complex (amplitude) values. The new E_S and E_P components are then recomposed into the Jones electric vector and propagation continues to the next surface.

The transmittance and reflectance coefficients depends on the (possibly complex) indices of refraction of the substrate, the (possibly complex) indices of refraction and thickness of each layer of (any) optical coating, and the angle of incidence of the ray. See "**The Handbook of Optics,**" Vol. I, McGraw Hill, 1995 for details of this computation.

Coating definition

For dielectric materials the index of refraction is represented as having only a real part, while for metals the index is complex. OptiCAD uses the following definition of the complex index:

$$\eta = n + ik$$

where n is the usual (real) index of refraction and k is the extinction coefficient. It should be noted that the form of the complex index is not standard, therefore care should be taken when entering values for n and k . As an example aluminum has a complex index of refraction of:

$$\eta = 0.7 - 7.0i$$

having a negative extinction coefficient in this notation.

Coating prescription and material data is contained in a single ASCII file named `coating.dat`, which must be located in the OptiCAD home directory. The file contains two types of data: the name and index of materials, and the construction of particular coating types. The file consists of the following format:

```
MATE <first_material_name>
wavelength n k
wavelength n k

MATE <second_material_name>
wavelength n k
wavelength n k
.....

COAT <first_coating_name>
<material_name> optical_thickness
<material_name> optical_thickness

COAT <second_coating_name>
<material_name> optical_thickness
<material_name> optical_thickness
.....
```

where MATE and COAT are key words, `<material_name>` is an upper case word, wavelength, n , k , and optical_thickness are numbers

Coating and material names can be any user defined name up to 15 characters in length, with no spaces permitted. The only keywords (upper case) used in the `coating.dat` file are MATE (for material) and COAT (for coating).

The wavelength is always specified in microns, and must be specified in ascending order.

The values for n and k are the real and imaginary parts of the index of refraction at that wavelength.

If only a single wavelength value is provided for a material, then the corresponding real and imaginary parts of the index are used no matter what wavelength is being traced. Dispersion of the coating material is therefore ignored.

If two or more values are given for a material, then for wavelengths shorter than the shortest defined wavelength, the shortest wavelength data is used. For wavelengths longer than the longest defined wavelength, the longest wavelength data is used. For wavelengths in between, linear interpolation is used.

When the `coating.dat` file is first read, OptiCAD verifies that each coating consists of materials that were defined in the materials section. If referenced materials are not defined, an error message is issued, the coating with undefined material layers is removed from the coating list, and the name of the incorrectly defined coating is written to a file named `badcoat.txt` in the OptiCAD home directory.

Coating layer optical thickness is specified in units of primary wavelength thickness in that medium. Therefore, if the same coating is used by two different optical systems, with two different primary wavelengths, then the actual coating layer thickness will be different. The actual thickness of a coating layer is determined by:

$$d = \frac{\lambda_0}{n_0} \cdot T$$

where λ_0 is the primary wavelength in microns, n_0 is the real part of the index of refraction of the coating layer at the primary wavelength, and T is the optical thickness of the coating layer specified in the `coating.dat` file. See the **Wavelength** section of the **Options** chapter.

The order in which the coating layers are specified is important.

There is room in the coating catalog for 50 material types, 100 wavelength-index points per material type, 50 coating types, and up to 150 layers per coating.

The `coating.dat` file is edited outside of OptiCAD using a text editor such as the NOTEPAD Windows editor.

Note that when polarization analysis is on, coating properties of a surface will be used to compute the reflectance and transmittance, other surface properties (such as fresnel or attributes) will be ignored.

Materials and coatings provided with OptiCAD

Materials	Description
AIR	Air, index 1.0
AL2O3	Aluminum Oxide, index 1.59
ALUM	Aluminum. index 0.7 -7.0i
ALUM2	Aluminum (alternative), index 0.82 -5.99i
CEF3	Cerous Fluoride, index 1.63
LA2O3	Lanthanum Oxide, index 1.95
MGF2	Magnesium Fluoride, index 1.38

N15	Imaginary material, index 1.5
THF4	Thorium Fluoride, index 1.52
ZNS	Zinc Sulfide, index 2.35
ZRO2	Zirconium Oxide, index 2.1
<u>Coatings</u>	<u>Description</u>
AR	0.25 waves of MGF2
HEAR1	High performance anti reflection coating
HEAR2	High performance anti reflection coating
METAL	Thin layer of aluminum
WAR	Antireflection W coating, having 0.5 waves of LA2O3 + 0.25 waves of MGF2
FTIR	Small air gap between two layers of index 1.5, used to show evanescent propagation

These materials and coatings are shown as examples, and may not be appropriate for any particular application. Always check with the coating manufacturer or designer for details of coating material specifics.

Specifying coatings on surfaces

A coating may be applied to a surface by double-clicking on an object name in the **OOB** and then selecting the desired surface from the expanded surface list below the object name. Double-clicking on an object name will also bring up the **OSPb**.

Select the **Attributes** page from the **OSPb**, and then select a coating name from the list in the **Selected Coating box**. The **Selected Coating box** is activated by clicking on the down arrow to the right of the box. Then click Apply Now.

Text Command:

```
coating/name="COATING_NAME" surface_name
```

Text Command Example:

```
coating/name="AR" lens1.1
```

Removing a coating

In order to remove a coating from a surface select the coating name <NONE> from the **Selected Coating box** described above.

Text Command:

```
coating/name="" surface_name
```

Text Command Example:

```
coating/name="" lens1.1
```

OptiCAD interprets the coating definition as follows:

Coating on glass

If the coating is applied to a boundary surface between air and glass, the definition of the coating should not include the glass index. The term glass here means any solid dielectric material, including lenses, rods, and solid polygons.

If the coating described above is raytraced with a ray going from air to glass, the coating stack is interpreted exactly as specified in the `coating.dat` file.

If the same coating is raytraced with a ray going from glass to air, then the order of the coating stack layers is automatically reversed, so that the coating is the same as if it had been raytraced going from air to glass.

If the coating is applied to a boundary surface between air and air, or glass to glass, the coating stack is interpreted exactly as specified in the `coating.dat` file.

Coating on a shell

If the coating is applied to a shell object, then the coating definition must include the substrate index. The last layer in the coating definition is then assumed to be a semi-infinite thickness substrate material. The term shell here means a surface not having an explicit index of refraction, such as a mirror, cavity, or non-solid polynet.

If the shell object is placed inside a glass object (using the Place -> Inside feature), then the index of the substrate must be entered as both the first and last layers of the coating. This technique can be used to model frustrated TIR.

If no coating is specified on a glass surface

The term glass here means any solid dielectric material, including lenses, rods, and solid polygons.

The order of precedence of surface properties is the same as for unpolarized raytracing. Surface-specific fresnel takes precedence over non-default surface reflectance, which takes precedence over global fresnel.

Surface reflectance is zero and fresnel is off

If the (unpolarized) reflectance of the surface is 0.0 (default) and the surface specific fresnel reflection or the global fresnel reflection are off, then the surface will have no reflection, and a transmittance of 1.0.

Surface reflectance is zero and global fresnel is on

If the (unpolarized) reflectance of the surface is 0.0 (the default) and the global fresnel reflection is turned on, then the surface reflectance and transmittance will be calculated using Fresnel's (polarized light) equation, based on the indices of refraction on both sides of the surface, on the angle of incidence between the incoming ray and the surface normal at the point of intersection, and on the polarization state of the incoming light.

Surface reflectance is not zero and surface specific fresnel is off

If the (unpolarized) reflectance of the surface is not 0.0 (the default) and the surface specific fresnel reflection is off, then the surface reflectance and transmittance will be calculated as an idealized coating. An idealized coating is independent of the indices of refraction, angle of incidence, and polarization state. The intensity reflectance (R) is the surface reflectance, the intensity transmittance (T) is $T = 1 - R - A$ (A is the unpolarized surface absorptance). The amplitude reflectance is $\rho = \sqrt{R}$ and the amplitude transmittance is $\tau = \sqrt{T}$.

Surface specific fresnel is on

If the surface-specific fresnel reflection is turned on, then the surface reflectance and transmittance will be calculated using fresnel's (polarized light) equation, based on the indices of refraction on both sides of the surface, on the angle of incidence between the incoming ray and the surface normal at the point of intersection, and on the polarization state of the incoming light.

If no coating is specified on a shell surface

The term shell here means a surface not having an explicit index of refraction, such as a mirror, cavity, or nonsolid polynet.

Surface reflectance is 1.0 and the fresnel metal is off

If the (unpolarized) reflectance is 1.0 (the default) and the fresnel metal is off, then the surface reflectance is calculated using Fresnel's (polarized light) equation, as if the surface had an aluminum (ALUM) coating. The transmittance of the surface will be 0.0.

Surface reflectance is not 1.0 and the fresnel metal is off

If the (unpolarized) reflectance is not 1.0 (the default) and the fresnel metal is off, then the surface reflection is calculated as an idealized coating. An idealized coating is independent of the indices of refraction, angle of incidence, and polarization state. The intensity reflectance (R) is the surface reflectance, the intensity transmittance (T) is $T = 1 - R - A$ (A is the unpolarized surface absorptance). The amplitude reflectance is $\rho = \sqrt{R}$ and the amplitude transmittance is $\tau = \sqrt{T}$.

Surface fresnel metal is on

If the surface specific fresnel metal is turned on, then the surface reflectance and transmittance will be calculated based on the complex indices of refraction and the absorptance flag, which are all entered with the surface fresnel definition. The term cons1 in the fresnel metal definition is ignored. See the **Metal** section of the **Fresnel** page in the **Object and Surface Properties** chapter for details.

Defining polarization sensitive explicit surface properties

Polarization sensitive surface properties may be explicitly defined independently for reflected and transmitted rays. Properties include: x and y analyzers (polarizers), x and y quarter wave plates, x and y half wave plates, Jones matrix (four complex coefficients), and a null Jones matrix. The Jones matrix modifies a Jones vector according to

$$\begin{bmatrix} E'_x \\ E'_y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E_x \\ E_y \end{bmatrix}$$

where A, B, C, D, Ex, and Ey are complex numbers.

Analyzers, wave plates and Jones matrices take precedence over both dielectric and metal coatings, and the reflectance and absorptance entered on the **Attributes** page of the **Object and Surface Properties** box. These properties are independent of both wavelength and angle of incidence.

The reflective Jones matrix, whether defined by name or by direct entry is first multiplied by the square root of the surface reflectance defined on the **ReflectivePolarization** page of the **Object and Surface Properties** box, before being used in a calculation.

The transmissive Jones matrix, whether defined by name or by direct entry is first multiplied by the square root of the surface transmittance defined on the **TransmissivePolarization** page of the **Object and Surface Properties** box, before being used in a calculation.

The following Jones Matrices may be selected by clicking on their names as described in the **Reflective Polarization** and **Transmissive Polarization** pages in the **Object and Surface Properties Box** chapter. The user can also enter the eight coefficients of the Jones matrix in the **Reflective Polarization** and **Transmissive Polarization** pages. The notation shown below for a complex number is (f,g) where f is the real part and g is the imaginary part.

Warning: Because you can independently specify the reflectance and transmittance, you must use great care in order to assure a physically real result. As an example, you can specify an x analyzer with a transmittance of 1.0 and a y analyzer with reflectance of 1.0 on the same surface. This would model a perfect polarization beam splitter or polarization recycler. However, specifying an x analyzer with a transmittance of 1.0 and an x analyzer with reflectance of 1.0 on the same surface, would cause a physically unreal result.

X Analyzer

$$\begin{bmatrix} (1, 0) & (0, 0) \\ (0, 0) & (0, 0) \end{bmatrix}$$

Y Analyze

$$\begin{bmatrix} (0, 0) & (0, 0) \\ (0, 0) & (1, 0) \end{bmatrix}$$

X Quarter Wave Plate

$$\begin{bmatrix} (0, 1) & (0, 0) \\ (0, 0) & (1, 0) \end{bmatrix}$$

Y Quarter Wave Plate

$$\begin{bmatrix} (1, 0) & (0, 0) \\ (0, 0) & (0, 1) \end{bmatrix}$$

X Half Wave Plate

$$\begin{bmatrix} (-1, 0) & (0, 0) \\ (0, 0) & (1, 0) \end{bmatrix}$$

Y Half Wave Plate

$$\begin{bmatrix} (1, 0) & (0, 0) \\ (0, 0) & (-1, 0) \end{bmatrix}$$

Null Matrix

$$\begin{bmatrix} (1, 0) & (0, 0) \\ (0, 0) & (1, 0) \end{bmatrix}$$

Activating polarization

Polarization raytracing must be explicitly activated to in effect. Activating polarization raytracing is a global condition, and will effect raytracing for all objects. This can be accomplished by checking the polarization menu item, or including the appropriate text command in a macro. See the **Polarization section** of the **Options chapter** for details.

Text Command:

```
coating/polarization/on  
coating/polarization/off
```

Defining the incident polarization

The polarization input is defined by the Jones vector:

$$E = \begin{bmatrix} E_x \\ E_y \end{bmatrix}$$

where E_x and E_y have both a magnitude and a phase. OptiCAD normalizes the specified E_x and E_y to have a magnitude of unity. Note that E_x and E_y are measured in terms of relative electric field amplitude. The E_x and E_y values describe the electric field in the local coordinate system of the ray, since E_z (in local ray coordinates) is zero.

The electric field is then converted from the local E_x and E_y description to the global E_x , E_y , and E_z , where in general E_z is not zero. The conversion is performed by choosing the Jones vector component E_x to indicate the electric field along a vector S , which has no global y component. The Jones vector component E_y indicates the electric field along a vector P , which is orthogonal to S and has no global x component. The vectors S and P are defined such that $S \times P = R$, where R is the ray vector. The global E_x , E_y , and E_z , are then computed from the local E_x and E_y projections along S and P . The details of this computation are handled automatically by OptiCAD, and no additional input is required. The polarization vectors of rays with different input angles are not parallel, since the polarization vectors must be orthogonal to the ray direction.

The initial amplitudes and phases of E_x and E_y can be entered from either the Options menu or the macro command described below. Note that the incident polarization can only be set when polarization raytracing is activated, and that the default values for incident polarization are: $|E_x| = 0$, $|E_y| = 1$, $x_phase = 0$, $y_phase = 0$. See the **Polarization** section of the **Options** chapter.

Text Command:

```
coating/polarization/initial |Ex| |Ey| x_phase y_phase
```

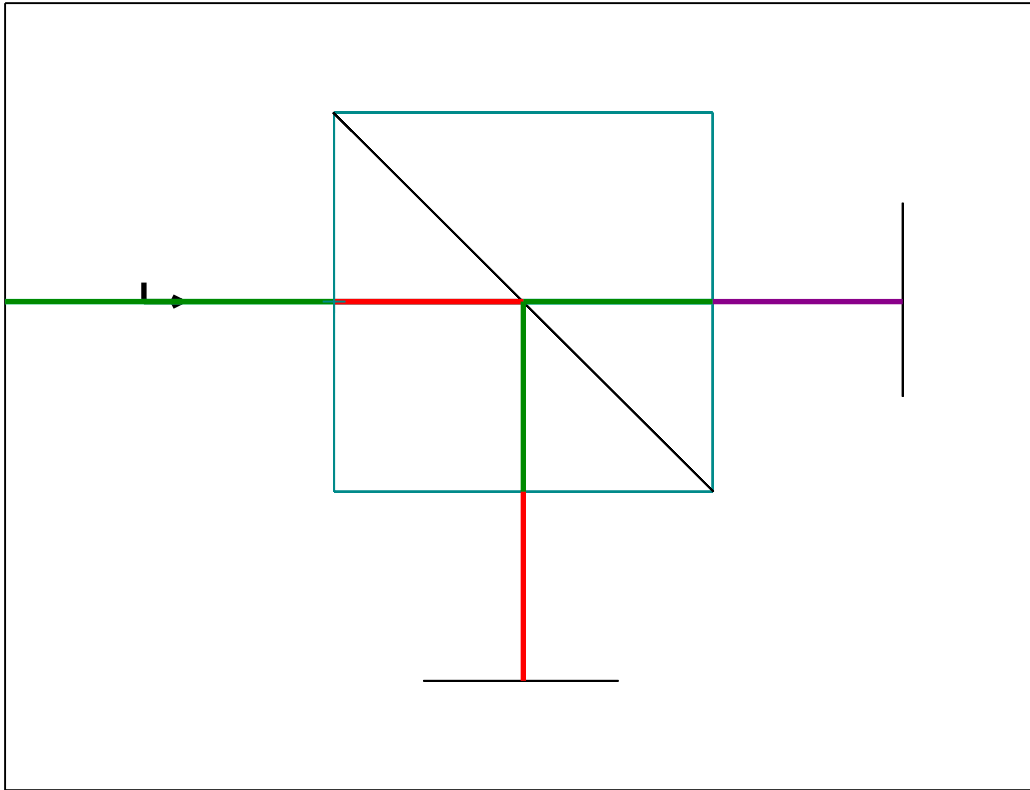
Text Command examples:

```
coating/polarization/initial 0.707107 0.707107 0 0
coating/polarization/initial 1 0 0 0
```

Modeling frustrated total internal reflection

Frustrated TIR (FTIR) occurs when a ray of light traveling through glass strikes an interface at an angle exceeding the critical angle. If another dielectric medium, such as another piece of glass, is placed close, but not touching the interface, some light will transmit through the thin gap and propagate, even though the refraction at the boundary cannot satisfy Snell's law. Both the reflected and transmitted beams will be affected, depending upon the thickness of the gap. In the limit of a gap having zero thickness, the light will continue through the gap as if there were no boundary. In the limit of a large gap, more than a fraction of a wavelength, then virtually all of the light is perfectly reflected.

The polarization routine handles FTIR in a normal manner. To model FTIR the user needs to define a mirror, add a coating consisting of one wave of glass substrate index + a small air gap + one wave of glass substrate index. Then place the mirror inside a glass object (using the Place -> Inside feature) at the location and orientation of the gap. The ray splits at the interface and both the reflected and transmitted rays are modeled. See the macro `FTIR.MAC` which has a 0.1λ gap, at 45° , inside a material of index 1.5.



FTIR - $n=1.5$, $\lambda=0.5$, $gap=0.5$, $angle=45^\circ$, $E_x=E_y=0.7071$
 transmittance=0.8415, reflectance=0.0801

Limitations of polarization analysis

There are a few limitations to the polarization analysis capabilities of OptiCAD. The data in the `coating.dat` file should always be carefully checked for accuracy whenever fabrication decisions are being made. Also, the predicted performance data should always be presented to the coating manufacturer as a check to insure that the coating will behave as intended.

The polarization raytracing algorithm generally does not return correct results for systems containing certain types of surfaces, including diffraction gratings, fresnel lens surfaces, and scattering surfaces. Generally, only surfaces which have refraction and reflection described by Snell's law have valid polarization computed. In particular, the polarization algorithm assumes that rays remain in the plane of incidence after reflection and refraction. This is true for conventional refractive optics (it is a condition of Snell's law) but is not true in general for diffractive optics (such as diffraction gratings).

The polarization raytracing algorithm generally does not return correct results for systems containing certain types of media, including gradient index media, birefringent media, and volumetric scattering media for the reasons described above.

OptiCAD decides if a ray will TIR before evaluating the refraction or reflection at a surface or coating.

Special considerations

Coating properties will be ignored by the raytrace unless polarization raytracing is activated.

For a glass surface with no coating assigned, there will be no reflection unless fresnel reflections are turned on for that surface (or globally).

For a shell object such as a mirror, with a coating assigned there will be no transmitted ray while the sum of the reflectance plus absorptance is unity (the default). If you want rays to be transmitted through a coated shell, set the reflectance to a value less than unity and the absorptance to 0.0. A reflectance value of 0.9999 would allow a transmitted ray through a coated shell.

End of Polarization Analysis Chapter

OPTIONS

There are two basic types of options: the first is setting a condition on or off, and the second is to set the value of a global parameter.

On/off conditions (such as global Fresnel reflections) are set with a toggle switch in which the on condition displays a checkmark to the left of the feature name, and an off condition does not display the checkmark. Click a feature without a checkmark (is off) and the next time the Options box is opened the feature has a checkmark (is on). Click this feature with a checkmark (is on) and the next time the Options box is opened the feature does not have a checkmark (is off).

Options may be saved by editing the startup file StartUp.opt, which is automatically read whenever OptiCAD is started.

Global parameters are set by entering a value in a box. No object or surface name is required.

AutoRetrace

Function:

Repeats last ray–trace operation **after** any **change**

Discussion:

When this option is on, the last successful ray operation is re-executed each time any system change is made. The advantage of having this option on is that it keeps the ray–trace current. The disadvantage of having the option on is that it consumes time.

This option is generally set to off, except for trivial optical configurations.

This feature is almost always implemented by using a toolbar button with the same name.

A single retrace can be implemented by double clicking the AutoRetrace toolbar button.

Parameters:

Toggle On (checked) or Off (unchecked).

Text Commands:

```
set/autoretrace/on  
set/autoretrace/off
```

AutoRedraw

Function:

Redraws views **after** any **change**

Discussion:

When this option is on, the current screen view is redrawn each time a system change is made. The advantage of having this option on is that it keeps the view current. The disadvantage of having the option on is that it consumes time.

This option is generally set on, except for extremely complex optical configurations or large polynet objects.

This feature can be implemented by using a toolbar button with the same name.

A single redraw can be implemented by double clicking the AutoRedraw toolbar button.

Parameters:

Toggle On (checked) or Off (unchecked).

Text Commands:

set/autoredraw/on
set/autoredraw/off

UpdateAll

Function:

AutoRetrace + AutoRedraw

Discussion:

This is a function rather than an option. Selecting this feature performs a single Retrace followed by a single Redraw.

This feature can be implemented by using a toolbar button with the same name.

SelectorMode

Function:

Selects mouse options

Discussion:

The selector mode determines the mouse's graphical function. Clicking one of the four buttons selects that mode, to the exclusion of the other choices.

Point (Select Location)



Function:

Displays cursor location

Discussion:

This feature displays the coordinates of the cursor location (while the mouse button is depressed) as it is moved about the screen. The operation must be performed in two orthogonal views to locate a point in 3-D space.

The last coordinates of the cursor are entered in the Place input box.

This feature can be implemented by using a toolbar button with the same name.

SizeBox (Zoom In)



Function:

Zooms in on a subregion of the display

Discussion:

The user can resize the viewing box by clicking and then dragging a variable size zoom box. See the Tutorial chapter for a detailed example. Unzooming can only be accomplished using the SnugWorkbox feature (toolbar button). It is usually easier to use this feature when the display is in orthogonal view mode.

This feature can be implemented by using a toolbar button with the same name.

AzEl (Perspective View)



Function:

Selects viewing location

Discussion:

This feature changes the position of the observer. The screen must be set to display a perspective view.

Click and drag mouse horizontally (right to left or left to right) to change azimuth.

Click and drag mouse vertically (up to down or down to up) to change elevation.

This feature can be implemented by using a toolbar button with the same name.

RayTraceLimits

Function:

Sets ray termination conditions

Discussion:

This option sets the conditions for terminating a ray. A ray is tested against the three termination conditions: number of surfaces hit, relative ray energy, and order (or level) of ghosting. A ray is terminated when one of the three termination conditions is violated, even if the other two conditions have not been met. Care must be taken to set the ghost criteria large enough when multiple beam splitters are present.

Note fro mintrans:

The addition of radiometric/photometric units changes how the mintrans value works. For Previous versions of OptiCAD, all non table based rays were lunched having unit energy, and the mintrans value (used to truncate a ray when its energy became too small) was a fixed value (default was 0.01 or 1%). Now the power in a launched ray (for a single raytrace) is

$$InitialPowerperRay = \frac{TotalLightsourcePower}{SumofWaveWeights \bullet NumberofRaysTraced}$$

and the value entered for mintrans is a fraction of the initial power for that ray, rather than an absolute value.

Parameters:

Maximum number of surfaces (maxdepth) that can be intercepted (including the original lightsource) before a ray is terminated, the minimum relative ray energy (mintrans) level that a ray must remain above or be stopped, and the maximum order of ghosting (maxghost), which is the maximum number of times a ray may divide keeping its minor energy component (only the minor component is truncated).

SetRayLimits		
		Cancel
maxdepth	max_intercepts	Ok
mintrans	min_energy	
maxghost	max_ghosts	

Text Command:

```
set/raylimits maxdepth mintrans maxghost
```

Text Command Examples:

```
set/raylimits 100 0.01 3
set/raylimit 100 0.01 3
set/raylim 100 0.01 3
```

Wavelengths - Weights - Primary Wavelength

Function:

Sets several **wavelengths**, wavelength **weights**, and Primary Wavelength

Discussion:

The user may specify up to 12 wavelengths and a corresponding weight for each wavelength. Wavelength data is used to calculate index of refraction for materials included in a previously loaded glass catalog.

A ray-trace command results in rays being traced at every wavelength with a nonzero weight. The initial ray energy of each ray is equal to the wavelength weight.

The user may select a primary wavelength which is used when specifying and analyzing dielectric coatings. The user specifies wnum (the position on the wavelength list) of the primary wavelength. The default for wnum is 2.

Parameters:

Up to 12 wavelengths (in microns), up to 12 wavelength weights, and 1 primary wavelength.

Wavelength (microns)	Weight	Primary Wavelength	
0.4	0		OK
0.45	0	x	Cancel
0.5	1		
0.55	0		
0.6	0		
0.65	0		
0.7	0		
1	0		
1	0		
1	0		
1	0		
1	0		

Text Command:

```
set/wavelength w1 w2 w3 w4 w5 w6 w7 w8 w9 w10 w11 w12
set/weightwave w1 w2 w3 w4 w5 w6 w7 w8 w9 w10 w11 w12
set/primary_wavelength wnum
```

Text Command Example:

```
set/wavelength 0.4 0.45 0.5 0.55 0.6 0.65 0.7
set/weightwave 0 0 1 0 0 0 0
set/primary_wavelength 2
```

Iteration

Function:

Sets **iterative intercept conditions**

Parameters:

The maximum number of iterations attempted before an iterative intercept is abandoned and the precision before an iterative intercept is considered to have converged.

SetIteration		
		Cancel
maxcount	max_iterations	Ok
tolerance	min_error	

Text Command:

```
set/iteration maxcount tolerance
```

Text Command Example:

```
set/iteration 20 0.000001
```

Proximity Glue

Function:

Sets **bonding distance**

Discussion:

The glue feature allows the bonding of essentially common surfaces.

Rays being traced by OptiCAD must intercept surfaces in a physically real order. Therefore, two identical surfaces cannot be placed to be exactly coincident. The user must place components such that a small gap (such as 10^{-5}) exists between surfaces. The Proximity Glue feature fills that gap with the index of refraction of one of the adjacent materials. The requirement of noncoincidence is met, and the surfaces are bonded with an index matching glue. The numerical parameter (maxdistance) in Proximity Glue is the ray path between surfaces that determines whether to glue or not to glue. The glue decision is made on a ray by ray basis. If the ray path is less than maxdistance, the gap is filled with index matching glue.

A tolerance of ten times the actual gap is reasonable. Any nonzero entry for maxdistance activates Proximity Glue, and a zero entry deactivates the feature.

Note that if a ray path is less than the max_path, the Proximity Glue feature does not stop the ray from being refracted, but instead fills the gap with the index of refraction of the object that was defined first (has a lower internal object number). In cases where three objects are within the max_path distance, the order of object definition is critical to ray-trace performance.

Parameter:

Maximum ray path (maxdistance) that will be index-filled.

SetProximity		
		Cancel
maxdistance	max_path	Ok

Text Command:

```
set/proximity_glue maxdistance
```

Text Command Examples:

```
set/proximity_glue 0.0001  
set/proximity 0.0001
```

Fresnel Reflect

Function:

Activates **Fresnel reflections** globally for dielectric materials

Discussion:

Calculates the transmittance and reflectance for each ray intersection with a dielectric material (glass, plastic) based on the average of the Fresnel reflections for the two polarizations, the index of refraction (though not the volumetric absorptance) of the materials, and the angle between the ray and the local-surface-normal at the point of intersection.

This feature may generate multiple ghost rays that clutter the screen. The user should pay particular attention to the RayTraceLimits (discussed in this section).

The global Fresnel reflection calculation may be overridden for an individual surface by setting the surface transmittance to a value other than 1.0 (the default). Setting the surface reflectance to 10^{-6} will deactivate the Fresnel reflection at a surface.

See also the surface specific Fresnel reflection feature on the fresnel page, described in the **Object and Surface Properties** chapter.

This feature should be used for reflections at the surface of a dielectric material (such as a lens). The feature will have no effect on a mirror in air.

In order to have a Fresnel reflection, the maximum ghost order must be set to at least 1.

Note that entries on the **Fresnel page** take precedence over **Surface Attributes** which take precedence over **Options** -> **Fresnel Reflect**.

Parameters:

Toggle On (checked) or Off (unchecked).

Text Commands:

```
set/fresnel/on  
set/fresnel/off
```

Add Rays

Function:

Sets the **ray file** to remain **open**

Discussion:

This option allows the user to add the results of multiple ray-traces. The default is that at the beginning of a ray-trace, the program clears the files containing ray-surface intercepts and spot data.

With `add_rays` set on, ray-traces from multiple configurations (wavelengths, lightources, component positions) can be combined.

When performing a series of ray-traces, it is prudent to perform the first trace with the `add_rays` command set to off in order to be sure that the accumulated ray files do not accidentally contain ray data from a previous (and possibly irrelevant) ray-trace. It would also be prudent to set `add_rays` off at the end of a multiple ray-trace sequence.

Lightsource

The addition of radiometric/photometric units changes how the AddRays feature works. Previously the relative power/energy results displayed were normalized by the total ray energy traced, including the results of multiple raytraces with AddRays set to be on. Therefore if AddRays were on and the user clicked on UpdateAll (retracing the last ray-trace), the radiometer and energy calculations (and displays) would be unchanged. Now a lightsource has a user defined quantity of power, and multiple raytraces (from the same or different lightsources) adds power to the system. Therefore if AddRays is on and the user clicks on UpdateAll, the values for an energy or radiometer calculation displayed for power and power density will double.

Reverse Ray-Trace Detector

The Add Rays option works differently for a detector than for a lightsource. When the Add Rays option is on a detector accumulates power from multiple ray-traces in a similar manner to film in a camera under going multiple exposures. Repeating the same ray-trace a second time will double the total accumulated power reported in the **TextOut box** and in the camera array.

Parameters:

Toggle On (checked) or Off (unchecked).

Text Commands:

```
set/add_rays/on  
set/add_rays/off
```

Text Command Sequence Example:

```
set/add_rays/off  
ray/col light1 10 10 1 1  
set/add_rays/on  
ray/col light2 10 10 1 1  
ray/col light3 10 10 1 1  
ray/col light4 10 10 1 1  
set/add_rays/off
```

Save Rays

Function:

Saves the ray paths

Discussion:

This option allows the user to not save (default is to save) the complete path for every ray. Not saving the ray paths increases ray–tracing speed and saves disk space, and is a useful option when only the results of an energy calculation are required.

Note that turning off savespots (not saving the spot data) also turns off saverays (does not save the ray path data).

Parameters:

Toggle On (checked) or Off (unchecked).

Text Commands:

```
set/saverays/on  
set/saverays/off
```

Text Command Examples:

```
set/saverays/on  
set/saverays/off  
set/saveray/on  
set/saveray/off
```

Save Spots

Function:

Saves the ray intercepts on the film plane

Discussion:

This option allows the user to not save (default is to save) the location and energy of every ray hitting the film plane. Not saving the film plane data increases ray–tracing speed and saves disk space, and is a useful option when only the results of an radiometer calculation are required.

Note that turning off savespots (not saving the spot data) also turns off saverays (does not save the ray path data).

Parameters:

Toggle On (checked) or Off (unchecked).

Text Commands:

```
set/savespots/on  
set/savespots/off
```

Text Commands:

```
set/savespots/on  
set/savespots/off  
set/savespot/on  
set/savespot/off
```

Reset

Function:

Resets Options to Default Values

Discussion:

This feature resets parameters controlled by the **Options menu** to the OptiCAD internal default values.

Parameters:

Click to implement.

SphLensVertex

Function:

Sets the **placement option** for a spherical lens

Discussion:

The default local coordinate system of a spherical lens is located at the first surface of a lens.

This option allows the user to specify whether a spherical lens is placed such that its local coordinate system is considered to be located at either the center of the lens (SphLensVertex option is off) or at the vertex of the first surface (SphLensVertex option is on).

The user must use caution to be sure that the option chosen is consistent between initial lens definition and future use. The Place command will act differently depending on which option is selected.

The SphLensVertex option on should be used for all new systems and macros.

Never change this option in the middle of an OptiCAD session or in the middle of a macro.

Parameters:

Toggle On (checked) or Off (unchecked).

Text Commands:

```
set/sphlensvertex/off  
set/sphlensvertex/on
```

Preferences

PopUpOnError

Function:

Causes a popup box explaining an error to appear when the error occurs.

Parameters:

Toggle On (checked) or Off (unchecked).

NURBS CacheSizes

Function:

Sets the cache memory sizes for raytracing NURBS objects.

Discussion:

When an NURBS object is raytraced blocks of memory are reserved as cache, this greatly speeds up the raytrace. The user can directly control the size of these caches using this option.

Note that extreme caution should be used in changing these cache parameters. Setting these parameters is not a substitute for having enough memory (RAM) in your computer. Consider changing these parameters only if an NURBS object traces correctly, but seems to trace unusually slowly for the size of the object.

Parameters:

The cache sizes for curves, surfaces, trimmed surfaces, and b-reps.

Polarization

Trace mode activation

Function:

Activates polarization raytracing

Parameters:

Toggles on (checked) or off (unchecked).

Text Commands:

```
coating/polarization/on  
coating/polarization/off
```

Polarization state initialization

Function:

Sets the initial polarization state.

Parameters:

Magnitude of the electric field in x, y, and z (E_x , E_y , E_z), phase of the electric field in x, y, and z (Ph_x , Ph_y , Ph_z).

Discussion:

The values of (E_x , E_y , and E_z) will be normalized, therefore $E_x = 1$, $E_y = 1$, $E_z = 0$ is equivalent to $E_x = 0.707107$, $E_y = 0.707107$, $E_z = 0.0$.

Note that the initial polarization are specified in global coordinates.

Text Command:

```
coating/polarization/initial |Ex| |Ey| x_phase y_phase |Ez| z_phase
```

Text Command examples:

```
coating/polarization/initial 0.707107 0.707107 0 0 0 0  
coating/polarization/initial 1 0 0 0 0 0
```

Sew CAD Objects

Function:

Sets the option to sew together surfaces of a CAD object

Discussion:

OptiCAD attempts to sew together the common edges of adjacent surfaces for both solid and shell CAD objects. This eliminates ray leakage through seams between edges of surfaces. On rare occasions, particularly when the gaps between surfaces is greater than the CAD tolerance, this sewing operation can damage the surface. The Sew CAD Objects option, which is normally on, may be switched off to allow importation of such objects.

Parameters:

Toggle On (checked) or Off (unchecked).

Text Commands:

```
set/sew_cad/on  
set/sew_cad/off
```

Sobol Sampling

Sobol sampling (described in the **Analyze** chapter) can be activated for ray origins and directions from the Options menu and by the text commands

for spatial (ray origins) Sobol

```
set/sp_sobol/on  
set/sp_sobol/off
```

and for direction (ray angles) Sobol

```
set/di_sobol/on  
set/di_sobol/off
```

Sobol sampling is implemented for functions defined using the **Emitance Function** page of the **Object and Surfaces Properties Box** and most of the standard source dlls (\sources\standard\dlls). Dlls that are not Sobol compatible will produce an error message and default to Monte Carlo sampling.

T.I.R. User Defined Reflectance

Function:

Sets the option to use the user specified reflectance for a ray that has undergone total internal reflectance.

Discussion:

The default reflectance for a ray that has undergone total internal reflectance at an uncoated surface is 100%. This option uses the reflectance specified on the **Attributes** page for the surface reflection. This option only effects rays that have undergone T.I.R..

Parameters:

Toggle On (checked) or Off (unchecked).

Text Commands:

```
set/tir_user_reflectance/on  
set/tir_user_reflectance/off
```

Toolbar

Function:

Displays or hides **toolbar**

Parameters:

Toggle On (checked) or Off (unchecked).

StatusBar

Function:

Displays or hides **status bar**

Parameters:

Toggle On (checked) or Off (unchecked).

ObjectBox

Function:

Displays or hides the OptiCAD Objects box.

Parameters:

Toggle On (checked) or Off (unchecked).

Snug Workbox



Function:

Snugs viewing box to fit objects

Discussion:

This feature is almost always implemented by using a toolbar button with the same name.

Text Command:

`set/workbox/snug`

SetWorkbox

Function:

Sets location and dimension of **view box**

Discussion:

This feature is a more precise yet less convenient way to control the viewing box.

Parameters:

The coordinates of the center of the view box (workbox) and the x, y, and z half-sizes of the workbox.

SetWorkbox			
	x	y	z
Center	x_location	y_location	z_location
Size	x_half_size	x_half_size	x_half_size
	Ok	Cancel	

Text Command:

```
set/workbox xcenter ycenter zcenter xsize ysize zsize
```

Text Command Example:

```
set/workbox 0 0 0 100 100 100
```

SetUnits

Function:

Sets **units**

Discussion:

Changing units redefines the numerical values that define a system, changing units does not scale numerical values.

Parameters:

Centimeters (cm), millimeters (mm), meters (meters), or inches (inches).

Text Commands:

```
set/units/cm  
set/units/mm  
set/units/inches  
set/units/meters
```

SetPowerUnits

Function:

Sets **units of power**

Discussion:

The choice of power units determines the units of output displays. Choosing Lumens will result in power density being displayed as lux (lumens/meter²) for a film plane detector or intensity displayed as candela (lumens/steradian) for a polynet detector; while choosing Watts will result in power density being displayed as watts/cm² and intensity as watts/steradian.

Parameters:

Lumens or Watts.

Text Command:

```
set/powerunits/lumens  
set/powerunits/watts
```

SunPosition

General Function:

Sets Open GL **Solid illumination** direction

General Discussion:

This is an advanced control feature. In most cases the default illumination condition (which put the light behind the viewers head) is a good choice.

Relative

Function:

Sets the illumination direction **relative** to the viewer

The Relative option places the sun relative to the viewer view angle, so that relative 0, 0 means that the sun is shining directly from the viewer. We suggest putting the sun just over the viewer's shoulder at 10 degrees above and 20 degrees to the side.

Parameters:

The relative illumination azimuth and the relative illumination elevation.

SetSunRelative		
		Cancel
Azimuth	light_azimuth	Ok
Elevation	light_elevation	
Intensity	255	

Text Command:

```
set/sun/relative azimuth elevation intensity
```

Text Command Example:

```
set/sun/relative 10 20 255
```

Absolute

Function:

Sets the illumination direction in the **global** coordinate system

Parameters:

The relative illumination azimuth and the relative illumination elevation.

Text Command:

```
set/sun/absolute azimuth elevation intensity
```

Text Command Example:

set/sun/absolute 135 60 255

GridSnap

Function:

Allows mouse coordinates to be rounded (**snapped**) to nearest grid location if the Grid is activated.

The Grid may be set using the View → Grid feature, described in the **View** chapter.

Parameters:

Toggle On (checked) or Off (unchecked).

FletchRays

Function:

Adds an **arrow** to each ray.

Discussion:

It is recommended that you leave the AutoFletching option On (check marked) and turn Fletching On and Off as needed. This feature may need to be reset (by turning it off and then on again) after zooming in on a sub-region of the screen.

Text Command:

```
set/fletch/off  
set/fletch/on position=0.5 size=0.5
```

Linework Resolution

Function:

Sets **density** of **wire frame** graphics

Parameters:

Low, Medium (recommended), and High.

Text Commands:

```
set/resolution/low  
set/resolution/medium  
set/resolution/high
```

RayRainbow

Function:

Causes **ray** to change **colors** at every segment

Parameters:

Toggle On (checked) or Off (unchecked).

Color Spots By Energy

Function:

Color codes spots on the film plane by relative energy.

Parameters:

Toggle On (checked) or Off (unchecked).

Perspective

Function:

Sets **viewing distance** for both wire frame and shaded (solid model) image views.

A perspective distance of three times the workbox (viewing box) size is a reasonable choice. Perspective distances of less than the workbox size will produce spurious results.

Stereo 3D (red/blue)

Function:

Produces red/blue color **stereo views** suitable for viewing with red/blue glasses (not included).

PlotScaling

Function:

Sets **plot scaling options**

Discussion:

Plots such as Energy plots default to be auto-scaled to their largest value. Auto-scaling makes the comparison of multiple plots difficult. This feature allows the user to insert an absolute scale (maximum value) on multiple plots.

Another option for comparing multiple plots is to write the data to a file for later insertion into a spreadsheet program.

SetPlotScaling		
		Ok
⊗	Automatic	Cancel
⊗	Predefined	
	Predefined value	max_value

End of Window Chapter

PARAMETER STUDIES USING A SPREADSHEET APPLICATION

Often in illumination studies it is necessary to look at many different, but similar systems to determine how variations in one or several of the system design parameters effect the performance of the system as a whole. A convenient method of doing this can be by generating the OptiCAD macro through an external application such as a spreadsheet. Although the example below describes how to do this using Excel it is equally applicable to any program which can perform calculations and generate text files.

Interfacing with MS Excel

Examples in this section show how to use Excel to create an input macro. Examples also show how to use Excel to plot output data. Numerical evaluation of output data may also be performed. Excel is not included with OptiCAD; it must be purchased separately.

Creating an Input Macro with Excel

Start Excel and open the file InputEx.xls. This Excel file resembles an OptiCAD macro (but it is not a macro). Rows that contain nonmacro commands begin with a pound sign (#); these “commented out” rows generally have input parameters. In this example, you may change the F-number (a variable named F_number) and the number of rays to be traced (a variable named Total_Rays). The ray command uses the variable Total_Rays in cell G35 to specify the total number of rays to be traced, and the variable F_number in cells I35 and I42 to calculate the ray cone angle.

You may change the F-number and the total number of rays to be traced, and then save the file as InputEx.xls (using File → Save). Now save the file a second time as a text file using File → SaveAs and selecting text (Tab delimited) from the “Save File as Type” box. If InputEx.txt already exists, choose Yes to replace, and then select OK. Now close InputEx.txt (you do not have to exit Excel), and when asked to save changes, choose No (because you only want to save the text, not the formulas). The text file that you have created, InputEx.txt, is readable by OptiCAD as a macro.

Start OptiCAD and run the macro InputEx.txt by clicking File → RunMacro, then switch to the Excel directory and select InputEx.txt (do not select InputEx.xls). In addition to the screen view and the energy plot, the macro sends Energy data to a file named Ex1.csv. Some OptiCAD data is output as comma-separated values (the Excel convention for this data is as a file ending in csv).

Displaying Output Data with Excel

In this example we use Excel to plot the Energy data created by the macro InputEx.txt. Start Excel, open the file Ex1.csv (output data file), then open Ex1out.xls (output display file). The file Ex1out.xls is linked to the file Ex1.csv, such that whenever the data file changes (which it does every time the macro InputEx.txt is run by OptiCAD) the display file is automatically updated.

The display file also makes a copy of the location of the center of each bin and of the energy density in each bin. This data is also updated when the macro is rerun, as is the smoothed scatter plot of energy density versus bin center. The display file will not add any rows if the number of bins is increased. Therefore the 20 bins in this example should be considered fixed. With the OptiCAD output data read into Excel, many computational and display options are possible. Options include display radiometer output as a contour or isometric plot, calculation of the r.m.s. uniformity of a radiometer output, and the superposition of multiple energy plots originating from multiple film planes.

End of Parameter Studies Chapter

Numerics

3M Vikuki 10-17, 10-18

A

Abbe Prism 5-8

ABg scattering distribution 10-38

Absorptance 10-6

Absorptance functions 10-12, 10-14

Absorptance Functions Page 10-11

Activate an Object 7-1

Activate polarization raytracing 14-11

active birefringence 10-68

Active Object 2-1

Add Ray Traces 3-9, 3-10, 14-7, 14-8, 14-9, 14-10, 14-11, 14-12

Add Rays 14-7

ADVANCED TOPICS 16-1

Alias 2-1

Alternative Surface Data 10-26

AltSurfs Page 10-16, 10-26

Amplified Spontaneous Emissions 12-2

ANALYZE 12-1

analyzers 13-6

Angular absorptance function 10-13, 10-15, 10-92

Angular Emitance Function 10-92

Angular reflectance function 10-10

Aperture 9-20

Array of Laser Diodes 10-82

Array of surface treatments 10-16

ASCII Source File 10-85

Aspheric Surface Shape 10-28

Attributes Page 10-6

Automatic ReDrawing 14-1

Automatic Redrawing 3-9

Automatic ReTracing 14-1

Automatic Retracing 3-9

Axis 9-6

AzEl Selector Mode 3-9

B

backlit displays 10-17, 10-18, 10-19, 10-20

BASIC TUTORIAL 4-1

BEF 10-17

Belt - Polar Radiometer 9-31
Bidirectional Reflectance Distribution Function 10-42
Bidirectional Scattering Distribution Function 10-42
Bidirectional Transmittance Distribution Function 10-42
birefringence 10-68
birefringent material 10-68
Bitmap Array Source 10-86
Bitmap spatial absorptance function 10-12, 10-15, 10-91, 10-92
Bitmap spatial reflectance function 10-10
Bonding 14-6
BRDF 10-42
BSDF tabular data 10-42
BTDF 10-42
Bug Fixes 1-4
Bulk Absorptance 10-52, 10-53
Bulk Scattering 10-53

C

Camera 10-9, 10-12, 10-14, 10-91, 10-95
Camera and Reverse Raytrace Radiometry Results 10-10, 10-12, 10-15, 10-91, 10-95
Camera Page 10-94
Camera Results 8-9
Catalog Lenses 9-20
Cavity 9-22
Clear 6-1
Clone 9-32
Coating 10-7
Coating definition 13-2
Collimated 12-7
Colors 3-19
Comment 3-16
Compound Parabolic Concentrator 9-13
Cone 9-25, 12-4, 12-8, 12-10
Cone Cavity 9-26
Conical surface structure 10-18
Conrady Dispersion 3-18
Coordinate Systems 1-7
Copy Metafile 7-1
Copy Protection 1-3
Corner Cube 9-30
CPC 5-10, 9-13
CPC Mirror 9-14
Creating an Input Macro with Excel 16-1
crystal axis 10-68
Custom Lightsources 10-69
Custom lightsources 12-8
Cylindrical Cavity 9-27, 9-28
Cylindrical Emitting Surface Source 10-70

Cylindrical Emitting Surface Source-Detector 10-71
Cylindrical Emitting Surface Sources 10-70
Cylindrical Emitting Volume Source 10-76
Cylindrical Emitting Volume Source-Detector 10-76
Cylindrical Lens 9-18

D

Data File Based Angular Emitance Function 10-92
DeActivate an Object 7-2
Define 2-2
Defining and Placing a Film Plane 4-10
Defining and Placing a Lens 4-8
Defining and Placing a Mirror 4-7
Defining and Placing a Point Source 4-4
Definition Page 10-3
DEFINITIONS 2-1
Delete All Objects 6-1
Delete an Object 7-2
Descendant Rays 1-7, 2-2
DESIGN 9-1
Detector 9-3, 9-5
Detector Raytracing 12-3
Dielectric Materials 13-2
Diffraction Grating 4-25, 10-31
Diffuse Surfaces 5-2
Diode Linear Array Source 10-76
Direction Cosine Ray Spacing 12-6
Display 12-17
Display Alternative Surface Data 12-17
Display Colors 3-19
Display Locations 12-17
Display Object Properties 12-17
Display Orientations 12-17
Display Physical Extents 12-17
Display Rays 12-17
Display Spots 12-17
Display Surface Absorptances 12-18
Display Surface Reflectances 12-18
Display Surface Transmittances 12-18
dots 10-20

E

echo command 3-16
Echo On 3-16, 10-2
EDIT 7-1
Edit Last Macro 6-7
Edit Macro File 6-7
Ellipsoidal Cavity 9-24

Ellipsoidal Emitting Volume Source 10-77
Ellipsoidal Emitting Volume Source-Detector 10-77
elliptical cone 9-25
elliptical trough 9-28
Eltrough 9-28
Emitance functions 10-90
Emitance Functions Page 10-90
Encircled Energy 12-14
Energy 1-7, 12-11
Energy View 8-2
Enslitted Energy 2-1, 12-12, 12-13
Equal Solid Angle 12-5
Equation spatial function 10-10, 10-12, 10-15, 10-91
Error Messages 3-13
Excel 16-1
Exit 6-8
Exponential Scattering 10-41, 10-42
Extended Lightsources 10-69
Extended Sources Page 10-69
Extent 2-1, 4-14, 4-17, 4-19, 9-1
Extent of a Mirror 4-14
Extent of Lenses 4-17
extraordinary glass properties 10-68
extraordinary index of refraction 10-68
Extrude 11-5

F

Faceted Radiometer 9-31
FILE 6-1
File Formats 3-15
Film 3-9, 9-5
Film Plane 4-10
Film plane radiometer 12-18
Film View 8-2
Flat Camera 10-10, 10-12, 10-13, 10-15, 10-91, 10-92, 10-97
Flat Emitting Surface Source 10-72, 12-3, 12-6
Flat Emitting Surface Source-Detector 10-71
Flat Emitting Surface Sources 10-72, 12-3, 12-6
Flat Mirror 9-6
Fletch Rays 15-4
Frequently Asked Questions 1-9
Fresnel Lens 4-24
Fresnel Lenses 10-33
Fresnel Reflection 2-2
Fresnel Reflections 10-45, 10-46, 10-47, 14-7
Fresnel Surface Page 10-44, 10-49
Frustrated Total Internal Reflection 13-9

G

Gaussian Scattering Distribution 10-36
Gaussian surface normal distribution 10-37, 10-38
Gaussian Surface Source 10-72
Ghost Ray 2-2
Glass Catalog File Format 3-17
Glass Catalogs 3-17
Global and Relative Placement 4-20
Glue 14-6
Gluing Objects 14-6
Gradient index 1 Material 10-70
Gradient index 10 Material 10-63
Gradient index 2 Material 10-56
Gradient index 3 Material 10-57
Gradient index 4 Material 10-57
Gradient index 5 Material 10-58
Gradient index 6 Material 10-59
Gradient index 7 Material 10-60
Gradient index 8 Material 10-61
Gradient index 9 Material 10-62
Gradient Index Material 10-55
GRADIUM 10-61
Graphics 8-7
Grating 4-25, 10-31
Grid Snap 15-4

H

half wave plates 10-23, 13-6
Hardware Lock 1-3, 1-4
Helical Surface Source 10-79
Herzberger Dispersion 3-18
Highlight Facets 8-5, 8-6
Highlight Surfaces 8-5, 8-6
Hole 10-50, 11-9
Hyperbolic_BEF 10-17

I

IESNA Data Sources 10-81
IGES 2-2
IGES Import 6-2
igs File Extension 2-3
Illumination Engineering Society of North America 10-81
Import IGES 6-2
Import STL 6-3
Importance Sampling Scattering 10-34
Incident Polarization 13-8
Index of Refraction 10-3

initial polarization 10-94
Inside 1-8
Inside Placement 4-22
Installation Procedure 1-3
Integrating Sphere 5-6
Interfacing with MS Excel 16-1
INTRODUCTION 1-1
Irradiance (Surface) 10-6
Irradiance Map 12-18
Irradiance Map Source 10-86
Isotropic Emitting Point Source-Detector 10-77
Iterative Intercept Tolerance 14-5

J

Jones matrix 10-23, 13-6

L

Lambertian Distribution 10-70
Lambertian Scattering 10-35, 10-36
Lambertian Scattering Distribution 10-35
Lambertian surface normal distribution 10-36
Laser Diode Source 10-74
Laser Diode2 Source 10-75
Laser Diodes 10-82
Lens 4-8, 9-14
Light Source 9-2, 9-3
Lightsource 12-1, 12-4, 12-9, 12-11
lightsource initial polarization 10-94
Lightsource Raytracing 12-3
Lightsources Versus Detectors 10-70
Linework Density 15-4
Low Discrepancy Sequences 10-34, 12-2
Lumens 9-3, 15-2
Luminance Functions Page 10-14
Lux 9-3, 15-2

M

mac File Extension 2-3
mac File Format 3-15
macro comment command 3-16
Macro Content Restrictions 3-20
Macro Only Commands 3-15
Macro Structure 3-13
Making a user defined DLL 10-64, 10-87
Manual Notation 1-5
Maximum Number of Ghosts 1-7
Measured Data Light Sources 10-81
Metafile 6-6, 7-1

Minimum Ray Transmission 1-7
Mirror 4-7, 9-6
MODIFY 13-1
Modifying a Surface Shape 4-10
Modifying Surface Reflectance 4-12
Monte Carlo sampling 10-34, 12-2
Multimode_Gaussian Surface Source 10-73
Multiple Ray-traces 14-7

N

New Features 1-5
Non-Sequential Ray Tracing 5-1
Normalize 11-7
Normalize a Polynet 11-7
Notation 1-5
Notepad 6-7

O

Object and Surface Properties Box 10-2
Object Cloning 9-32
Object Definition Page 10-3
Object Media Page 10-52
Object Names 3-11
Object Properties Box 10-2
OBJECT PROPERTIES BOX (new) 10-1
Object renaming 10-2
Object Selection Box (new) 10-1
Old Macros 3-20
OOB 10-1
OpenGL 8-6, 8-7
opt File Extension 2-3
opt File Format 3-15
OptiCAD 1-1, 4-1
OptiCAD Binary Format from Radiant Imaging 10-84
OptiCAD Objects Box 10-1
Optical Materials List 10-4
OPTIONS 14-1
Ortho View 6-2, 6-3, 8-1
OSPb 10-2

P

Parabolic Mirror 9-12
Parabolic Trough 9-10
Parabolic Trough Mirror 9-10
Parent Ray 1-7
Pause 3-15
pause command 3-15
Perspective 15-4, 15-5

Perspective View 8-1
Perturbed Surface Scattering 10-33
Physical Extent 9-1
Place an Object 7-2
Place Inside an Object 7-5
Placement 4-20, 4-22
Placement Page 10-4
Plot Scaling 15-5
Point Source 4-4, 4-6, 10-77
Polar Radiometer 8-3, 9-31, 12-20
Polar View 8-3
Polarization 14-11
Polarization analysis 13-1
Polarization sensitive surface propertie 10-23
Polarization sensitive surface properties 13-6
Polarization state initialization 14-11
polarizers 10-23, 13-6
Polygon 11-4
Polygonally Faceted Objects 5-7
Polynet 2-2, 4-29, 11-7, 11-9
Polynet Chain 11-4
Polynet Normals 8-5
Polynet Point 11-4
Polynet Radiometer 8-3, 9-31
Polynet radiometer 12-20
Polynet Thickness 11-5
Polynet Vertex 11-5
POLYNETS 11-1
Post Surface Scattering 10-33
Power Law Scattering 10-39, 10-40
Power Units 9-3, 9-5
Powerlaw surface normal distribution 10-40
Predefined Extended Light Sources 10-70
Preferences 14-10
Primary Wavelength 14-4
Print 3-9, 6-5
Print Preview 6-6
Print Setup 6-6, 7-1
Prism 9-30
Program Functions 3-1
ProSource 10-84, 10-85
Proximity Glue 14-6

Q

quarter wave plates 10-23, 13-6

R

Radiant Imaging Inc. 10-84, 10-85

Radiant Sources 10-84
Radiometer 5-5, 9-31, 12-18
Radiometer limits min/max 12-23
Radiometer min/max 12-23
Radiometer rebin radiometer 12-25
Radiometer smooth 12-23
Radiometer smoothing 12-23
Radiometry 5-5, 12-18
Ray Energy 1-7
Ray Paths 12-17
Ray Rainbow Display 15-4
Ray Termination Criteria 14-3
Ray Trace a Collimated Bundle 12-7
Ray Trace a Cone of Rays 12-4, 12-8, 12-10
Ray Trace Limits 4-12, 14-3
Ray Tracing 12-1
Ray-Trace Limits 4-12
Ray-tracing 12-1
Rebin radiometer data 12-25
Reference Axis 9-6
References 1-2
Reflectance 10-6
Reflectance functions 10-8, 10-10, 10-12, 10-14
Reflectance Functions Page 10-8
Reflectance option (T.I.R.) 14-12
Relative Placement 4-20
Remove an Object 7-12
Renaming objects 10-2
Repetitive surface structures 10-16
Replicate 9-32
ReRun Last Macro 3-9, 6-7
reset screen.out file 3-16
Retrieve 6-1
Reverse (Backward) Ray-tracing 10-95
Reverse (Backwards) Ray-tracing 9-3
Reverse Ray-Trace Detector 9-3
Reverse Raytrace Results 8-9
Rod 9-22
Rotate an Object 7-7
Rotate Viewing Angle 5-1
Rotating a Lens 4-9
Rotating a Point Source 4-6
Run a Macro 6-1, 6-2, 6-3

S

Save Current Configuration 6-4
Save Metafile 6-6
Save Ray Data 6-6

Save ray traces 14-9
Save Rays 14-8
Save Spot Data 6-6
Scattering 4-27, 5-2, 10-35, 10-36, 10-37, 10-38, 10-39, 10-41, 10-42
scattering 10-42
Scattering - Importance Sampling 10-34
Scattering - Target Object 10-34
Schott Dispersion 3-18
screen.out file 3-16
Seeding Random Number Generator 3-19
SELFOC 10-62
Sellmeier 1 Dispersion 3-18
Sellmeier 2 Dispersion 3-18
Set Viewing Window 15-1
Set Workbox 15-1
SetPowerUnits 15-2
Shell Object 9-1
Slit 12-12, 12-13
Smoothing the Radiometer Display 12-23
Snug Workbox 3-9, 15-1
Sobol sampling 10-34, 12-2, 14-12
Solid (shaded) Open GL 8-7
Solid CPC 5-10
Solid Model (old) 3-20
Solid Polynet 4-29
SolidOpenGL 8-8
Source functions 10-90
Sources Page 10-69
Spherical Camera 10-10, 10-13, 10-15, 10-92, 10-99
Spherical Lens 9-15, 9-17, 9-20
spherical lens 9-15, 9-17
Spherical Lens Placement 14-10, 14-11
Spherical Mirror 9-9
Spherical Surface Source-Detector 10-80
Spherical surface structure 10-19
Spot Diagram 12-17
Standard Gradient Index Material 10-55
Status Bar 15-1
Stereo 3D 15-5
STL 6-3
STL (stereo lithography) CAD file 6-3
STL CAD format 6-2
stl File Extension 2-3
Stock Catalog Lenses 9-20
Stray Light Modeling 5-4
Sum Ray Traces 14-7, 14-8, 14-9, 14-10, 14-11, 14-12
Sun Position 15-3
Sun Source 10-80

Support 1-2
Surface - absorptance functions 10-12
Surface - luminance functions 10-14
Surface Absorptance 10-6
surface absorptance (display) 3-16
surface absorptance table 3-16
Surface Attributes 10-6, 10-8, 10-11, 10-93, 10-94
Surface Extent Page 10-49
Surface Hole 10-50
Surface Names 3-11, 4-10
Surface Reflectance 4-12, 10-6, 10-45, 10-46, 10-47
surface reflectance function 10-8
Surface Scattering 4-27
Surface Scattering Page 10-33
Surface Shape 4-10
Surface Transmittance 10-6
Surface treatment 10-16, 10-22
System Requirements 1-2

T

T.I.R. reflectance option 9-2, 11-1, 14-12
Tabular Ray Data 10-83
Tangent Ray Spacing 12-6
Technical Support 1-2
Text Only Commands 3-15
Toolbar 15-1
Toolbar Buttons 3-8, 4-2
Toric 10-33
Toroidal 10-27
Toroidal Surface Source 10-78
Total Internal Reflection 2-2
Trace Depth 1-7
Translate an Object 7-10
Translators 3-20
Transmittance 10-6
Treatments 10-16
Trough 9-10
Trough Mirror 9-11
TUTORIAL 4-1, 5-1

U

Unconstrained Ray Tracing 1-6
Uninstalling OptiCAD 1-3
Units 15-2
UnZoom 5-1
Update Ray Trace 3-9, 14-2
Updates of OptiCAD Software 1-3
Upgrades 1-5

User Defined Custom Source 10-87
User Defined Gradient Index Material 10-64
USER INTERFACE 3-1
User Specified Tabulated Data 10-82

V

VIEW 8-1
View Camera Array 8-9
View Detector Image 8-9
View Energy 8-2
View Film 8-2
View Polar 8-3
View Polynet Normals 8-5
View with Rays 8-4
Viewing Perspective 15-4, 15-5
Viewing Point 14-3
Vikuiti 10-18
Vikuki 10-17
Volume Absorptance 10-52, 10-68
volumetric 3-16
volumetric absorptance (display) 3-16
volumetric absorptance table 3-16
Volumetric Object 9-2
Volumetric Scattering 10-53

W

Watts 9-3, 15-2
Wavelength 10-3, 14-3, 14-4
Wavelength Weights 14-4
wdm File Extension 2-3
wdm File Format 3-15
Wedge 5-8
WINDOW 15-1
Windows Metafile 7-1
Wire Frame Open GL 8-7
Wire Frame View 8-1
Working Environment 3-2

X

X View 8-1
X-Y Exponential Scattering 10-41, 10-42
X-Y Exponential Scattering Distribution 10-41
X-Y Exponential surface normal distribution 10-42

Y

Y View 8-1

Z

Z View 8-1

Zoom In 3-9, 5-1, 14-3
Zoom Out 15-1
Zooming 14-3
Zooming in on a View 4-9
Zooming in on the Film Plane 4-12