ASAP™

optical modeling software

REFERENCE GUIDE

BREAULT RESEARCH ORGANIZATION, INC.

This manual is for use with ASAP™.

Comments on this manual are welcome at: **support@breault.com**

For technical support, information on additional copies of this documentation, or technical information about other BRO products, contact:

Breault Research Organization, Inc.
6400 East Grant Road, Suite 350
Tucson, AZ 85715

US/Canada:	**1-800-882-5085**
Outside US/Canada:	**+1-520-721-0500**
Fax:	**+1-520-721-9630**
E-Mail:	
Technical Support:	**support@breault.com**
General Information:	**info@breault.com**
Web Site:	**http://www.breault.com**

Breault Research Organization, Inc., (BRO) provides this document as is without warranty of any kind, either express or implied, including, but not limited to, the implied warranty of merchantability or fitness for a particular purpose. Some states do not allow a disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you. Information in this document is subject to change without notice.

BRO-01288(06/04/03)/ref_guide*.fm, ASAP Reference Guide.doc

ASAP Commands Overview

Program commands consist of up to 10,000 unformatted alphanumeric entries separated by one or more of blanks " " or commas "**,**". Arbitrary string entries can be delimited by double quotes. Otherwise, literal entries consist of an unbroken string of alphanumeric characters (A through Z through 9, and **_**). They must start with a letter or underscore "**_**" and can be of any length (but only the first 8 may be significant).

Numeric entries can be in integer, floating point or exponential format. Exponential entries, for example "1.34E-5", cannot be longer than 40 characters.

Usually a command begins with a literal and there is only one command in each input record. However, more than one command can be placed in a record by separating them with semicolons ";". Also a command can extend over more than one record. If the last non-blank character in an input record is a comma (**,**), the command continues to the next record.

Entry delimiters	blanks and commas
Maximum number of entries per command	10000
Literal entries	begin with a letter
Commands	usually start with a literal
Maximum exponential numeric entry length	40 characters
Multiple commands per record separator	semicolon
Command continuation	comma last character of record
Command arguments	optional entries beginning with **…** (example: **…CLIP**); must appear after a command

Note to European ASAP Users

Decimal numbers must be entered with a U.S. decimal point, using the period (.) key. If you enter decimal numbers with the European decimal point, using the comma (,) key, ASAP treats it as a space, and your results will not be accurate.

See Also

Command Description Notation
Command Comment Strings
Mathematical Operators
Mathematical Functions Supported
Registers for Storing Arithmetic Results
Entries Repeated and Incremented

Overview of Kernel Capabilities

Modeling Optical/Mechanical Systems

ASAP accurately models virtually *any* optical/mechanical systems. Similar to 3-D solids modeling programs, it utilizes a very powerful geometrical approach that permits a nearly limitless variety of systems to be handled in a straightforward manner. As opposed to most other ray tracing algorithms, all surfaces and ray data are referenced by default to a single global Cartesian coordinate system. Smooth continuous object surfaces can be represented by a sequence of simple conicoids or a general 286-term polynomial (taken to 10th or 20th-order in the three Cartesian components or their squares) and can be bounded by other surfaces of the same general form. Therefore, anything from a simple plane with a polygonal boundary to an arbitrarily oriented elliptical toroid can be modeled precisely. Even more complex parametric mesh surfaces (NURBS) can be defined by connecting two or more arbitrary curves, each formed by a series of lines and/or conic arcs in space.

Tracing Rays through the System

Bundles of rays can be traced through the system such that after every reflection or refraction each ray always transfers to the nearest object. Also, rays are allowed to interact with an object any number of times; that is, multiple bounces. Therefore, the program easily handles the "funneling" effect of non-imaging radiation collectors as well as imaging systems. In addition, each ray may be assigned an arbitrary total flux that is reduced by volume and surface absorption (or increased in a gain medium) as the ray propagates through the system. The standard Fresnel equations are used to not only calculate (as a function of incidence angle and polarization) transmission losses at interfaces between two dielectric media, but also reflection losses at any dielectric/conductor interfaces. The program is also of capable of splitting any ray into reflected, transmitted, diffractive, near specular, diffuse, and backscatter components.

Modeling of Physical Optics

Each ray may also be treated as a coherent/incoherent scalar/vector beam (normally Gaussian). Groups of these beams can be combined to simulate the optical or electromagnetic field incident on a system. Each beam and thus the entire "wavefront" is then rigorously propagated. The resulting field may then be calculated and displayed at any location within the system (including on the surfaces of grazing incidence optics) and not just near focus. Therefore, ASAP not only performs geometrical optics modeling, but also accurate physical optics modeling of any system.

Command Mode

The command mode gives you access to the full power of ASAP for building and analyzing your optical system. As you become more familiar with working in the command mode, you will find it faster to enter commands at the command line. The following functions are available only in the command mode:

- Scatter and several other commands.
- Macros for performing looping, if-statements, source and geometry libraries; and for reducing frequently used command sequences to a single command.
- User-written files that you can extensively comment and easily read.
- Variables and mathematical expressions for defining quantities that are frequently changed.

Tip

You can use the command mode interactively from the Command Output window or in the background in batch mode. Since the output files are fully compatible with the graphical interface version, you can use the graphical interface to view or manipulate graphics.

Command Prompts

The top-level prompt in command mode is **ASAP>**. If you enter a key word in the Command Output window, the prompt changes to that key word prompt. Typing RETURN or pressing the Enter key returns you to the ASAP prompt. You can enter commands for constructing system geometry and source definitions at the prompt. Alternatively, you can create ASCII files with a system editor, and store system geometry and sources you create in separate files.

See Also

Command Input/Output Window
ASAP Prompts
Batch Mode
Commands by Function
Quick Reference Guide

ASAP Prompts

The top level prompt in the program is **ASAP>.** If you enter a key word in the Command Input window, the prompt changes to that key word prompt. Typing RETURN and clicking RUN restores the ASAP prompt. The following prompts may display in the status bar:

Entry	Prompt	Action
MEDIA	MED>	Accepts refractive index information
COATING	COA>	Accepts coating information
SCATTER	SCA>	Accepts scatter information
SURFACE	SUR>	Accepts surface commands
EDGE	EDG>	Accepts edge commands
LENS	LEN>	Accepts lens commands
OBJECT	OBJ>	Accepts object commands
(See Note 1)	LIN>	Performs after linear transformations
GROUP	GRU>	Groups objects as single entity
FIELD	FLD	Performs optical field calculation
DISPLAY	DIS>	Enters distribution file graphics display mode
...TEXT	TXT>	Accepts annotations to plots
USER...	USR>	Accepts user-defined data
RAY	RAY>	Accepts ray data
SEARCH	SEA>	Accepts/overrides ray trace sequence
ASYM	ASY>	Applies scaling operation to a surface
(See Note 2)	MAC>	Creates macros

Notes

The linear transformation commands are:

ALIGN
MATRIX
PLACE
ROTATE
SCALE
SHIFT
SKEW
XEQ

See Also

User-Defined Macros for more information on creating macros.

General Input Techniques

Techniques for ASAP input include: radian angle entries, relative and literal referencing, direction vectors, and linear transformations. Each technique is described below.

Radian Angle Entries

Angle entries that by default are in radians (that is, the PARABASAL divergence, ROUGHNESS RANDOM slope, and INTERFACE RMS/BSDF back cone angle) can also be entered in degrees by appending a "D" to the end of the number. For example, the following two entries are equivalent:

2.5D	2.5/57.29578 radians

If the entry is a direction cosine coordinate (CLIP DIR or SPOTS/SPREAD DIR window), the result is the sine of the angle; for example:

2.5D	SIN(2.5/57.29578)

Relative and Literal Referencing of Entities

Alternate schemes are available when referencing surface/edge/lens, media, object, ray or source numbers within commands. Instead of using the actual absolute number, you can specify a number relative to the largest number defined by using a decimal entry of the form ".i" where i is an integer between 1 and 9999 inclusive. ".i" is equivalent the 1 plus largest number defined, minus i.

Referencing Method	Format	Description
Absolute	n	actual entity number
Relative	.i	max+1-i
Literal	name	exact or abbreviation

Some examples of relative indexing are:

.2next to the largest number defined so far

-.1negative of the largest number defined

Also, for most commands that require the specification of a particular object, media or coating, the name can be used in place of directly specifying the number or using the relative indexing described above. The program first attempts an exact match (ignoring blanks). Otherwise, the characters in an abbreviation must be present in the same order as in original object, media or coating name but not necessarily consecutive; that is, any number of original characters can be skipped to make the abbreviation small enough but unique. An underscore "_" in a literal media reference always requires an exact match and is also used to separate the catalog name (file with extension CAT) from the glass name; for example, "SCHOTT_BK7".

Direction Vectors

See ...a,b,c... (ASAP Command Argument)

Linear Transformations

Linear transformations change the scaling or orientation of a geometrical entity (surface, edge, lens, object, group, and rays) by applying a general 4-by-4 linear transformation matrix to it. Any number of the following elementary operations can be applied after any entity definition to build up the final matrix. The order in which these operations are entered into the input stream is exactly the order in which they will be applied to the entity. These commands must be grouped together following an entity definition with no other commands between them (except for a comment). The LIST option causes the resulting 4-by-4 transformation matrix to be printed and decoded into simple operations if possible. A general transformation for a position vector (X,Y,Z) has the form:

$$\begin{pmatrix} 1 \\ X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ Dx & \mathrm{Axx} & \mathrm{Axy} & \mathrm{Axz} \\ Dy & \mathrm{Ayx} & \mathrm{Ayy} & \mathrm{Ayz} \\ Dz & \mathrm{Azx} & \mathrm{Azy} & \mathrm{Azz} \end{pmatrix} \begin{pmatrix} 1 \\ X \\ Y \\ Z \end{pmatrix}$$

Note

The As are the rotation submatrix while the Ds are the translation vector.

The first row is a dummy used to make the matrix square (and thus invertable).

For the transformation of a direction vector the dummy row contains all zeros.

Input Records

Input data is read sequentially from records up to 344 characters long (ASAP ignores anything past this length or a double exclamation point "!!"). ASAP first attempts to read input from a file named **defsetup.in?** (the last character in the extension depends on the particular application program). This file should contain any default input settings that would normally be used in every program run.

After processing the input from this file, ASAP starts reading either from the file specified as the first command line argument or, in batch mode, from logical unit 1 (BRO001.DAT). The file that this unit was assigned to (or the system default specification) should be created using the system editor.

If an end-of-file is reached while reading the input disk file, and before program run is terminated with the proper input command, ASAP automatically switches over to prompting you (with a greater-than sign ">") for data entry directly from the keyboard. This feature allows you to run ASAP from a file and/or in interactive mode.

Maximum input record length	344 characters
Premature record terminator	!!
Startup fetching sequence	defsetup.in? -> command argument -> Keyboard (BRO001.DAT batch)
Keyboard input prompt	???>

Command Description Notation

The format or syntax of each ASAP command is shown in the command topics, followed by a detailed explanation of the command’s function. A standard notation is used for the quantities shown.

- UPPER CASE letters or numbers represent the actual literal form of the entry.
- Single lower case letters represent numerical entries to be determined by the user.
- Lower case words represent literal entries that can take on the described set of values.
- Trailing lower case letters on literals and anything enclosed within square brackets are optional and will not trigger a program error if omitted.
- The default value for most optional numeric entries is zero, unless otherwise stated. Alternate forms for an entry are shown in the same vertical column.

See Also

Command Comment Strings

ASAP Commands Overview

Command Comment Strings

User comment strings may be imbedded in any command without usually affecting execution (they can be meaningful to some commands). Enclose them in an entry delimiter followed by one of the following:

Comment string delimiters: * = / \ < > ` '

If a command begins with any of these characters, then the rest of that command line is ignored. Comment strings delimited in this manner can be of any length (up to 344 characters) and composition (except matching delimiters), and provide the user with a convenient facility for documenting input files.

Letters opposite in case to that set by the $CASE command can also be used as imbedded comments since the program treats them as blanks. Trailing comments can be entered after an exclamation point (**!**) since this character signals the program to stop decoding input from the command.

Opposite case letters: treated as blanks

Last entry, stop parsing: exclamation point!

See Also

Command Description Notation

ASAP Commands Overview

Mathematical Operators

If two entries are separated by one of the following symbols, the entries are replaced by the result of the operation.

+	*ADD* the two entries	4
-	*SUBTRACT* the second from first	4
*****	*MULTIPLY* the two entries	5
^	*RAISE* first *TO* second *POWER*	6
/	*DIVIDE* first by second	5
\	*REMAINDER* after dividing first by second	6
<	Take *LESSER* of the two	6
>	Take *GREATER* of the two	6
%	*ARCTANGENT* (angle in degrees) of the first divided by second	6
~	Uniformly distributed *RANDOM* number between first and second	7
`	Form complex number from *REAL* and *IMAGINARY* parts	0
'	Form complex number from *MODULUS* and *PHASE* angle (in degrees)	0
([	Store value of expression and operator to the left	1
)]	Recall previous value and operator, evaluate new expression	2

- The two pair/complex operators "`'" are found under the tilde (~) (*REAL* and *IMAGINARY*) and double quote (*MODULUS* and *PHASE*) keys on most keyboards.
- Operator precedence is followed during input parsing:

  Exponentiation - first precedence

  Multiplication/division - second precedence

  Addition/subtraction - third precedence
- Parentheses may be used to force a particular order of evaluation.
- When $EXP is set to OLD, consecutive operations are always evaluated from left to right with no operator precedence, until a delimiter terminates the expression. Nested parentheses or brackets can be used when necessary.

Tips

Since curly braces { } are used in macro definitions, do not use them in mathematical expressions. Instead, use parenthesis ( ) and square brackets [ ] in mathematical expressions.

Example of Operator Precedence

2.0-1.0/4.0 is evaluated as **1.75**

(2.0-1.0)/4.0 is evaluated as **0.25**

See Also

ASAP Commands Overview

Mathematical Functions Supported

The following set of mathematical functions are also supported:

Closing right character:)]

INT	Truncate to integer	Round to nearest integer
EXP	Natural (base e) antilog	Common (base 10) antilog
LOG	Natural (base e) logarithm	Common (base 10) logarithm
ABS	Absolute value	
SGN	Sign (returns -1, 0, or +1)	
SQRT	Square root	
CBRT	Cube root with same sign	
SIN	Sine of angle in RADIANS	Sine of angle in DEGREES
COS	Cosine of angle in RADIANS	Cosine of angle in DEGREES
TAN	Tangent of angle in RADIANS	Tangent of angle in DEGREES
ASIN	Arcsine of angle in RADIANS	Arcsine of angle in DEGREES
ACOS	Arccosine of angle in RADIANS	Arccosine of angle in DEGREES
ATAN	Arctangent of angle in RADIANS	Arctangent of angle in DEGREES
BJ#	#th-order Bessel J function (# from 0 to 9)	
BK#	Modified Bessel K function (# from 0 to 9)	
STEP	0 for X<0, 1 for X>0	
RECT	0 for |X|>.5, 1 for |X|<.5	
GAUS	Gaussian EXP(-X^2)	Gaussian EXP(-PI*(X^2))
SINC	SIN(X)/X	SIN(PI*X)/(PI*X)
SOMB	2*BJ1(X)/X	2*BJ1(PI*X)/(PI*X)
FACT	X factorial (actually gamma)(|X|+1) with sign of X)	
ERF	ERROR function	Complement 1-ERF
FILE	Next available file number starting at X (usually 10).	
	Blackbody integral from 0 to X=microns*degreesK	
FBI	Fractional Energy	Fractional Photons
LPW	Lumens per Watt for X degree K blackbody	
	Normalized Visual Response at X microns	
EYE	Photopic (bright)	Scotopic (dim)
	Numeric to literal conversion	
LIT	8 or less characters	16 or less characters
RAN	Unit RMS	Unit Maximum

RAN Type	**Probability Distribution P(Y)**	**(Max/RMS)^2**
-15	Two delta functions	1
-14	|Y| ^14	(14+3) / (14+1)
:	:	:
-2	|Y| ^2	(2+3) / (2+1)
-1	Ramp	2
0	Uniform	3
1	Cosine	5
2	Convolution of 2 uniforms (triangle)	3*2
:	:	:
14	Convolution of 14 uniforms	3*14
15	Gaussian	2*LOG(2^32)

The fractional part of the argument to RAN is the relative amount (total probability) of an additional uniform variate with the same RMS or maximum; for example, RAN[2.6] has a probability distribution that looks like a house.

The ability to perform arithmetic operations during input decoding is a powerful feature when used with the macro facility.

Tip

The ability to perform arithmetic operations during input decoding is a powerful feature when used with the macro facility.

Examples

OLD		NEW	
2.-1./2.	==> .5	2.-1./2.	==> 1.5
2.-(1./2)	==> 1.5	2.-(1./2)	==> 1.5
SIN[A.]^2+(COS[A.]^2)	==> 1.0	SIN[A]^2+COS[A]^2	==> 1.0
SQRT((X.^2)+(Y.^2))’(Y.%X.)	==> X.‘Y	SQRT(X^2+Y^2)’(Y%X)	==> X‘Y
(3.-1.)^4	==> 16.	(3.-1.)^4	==> 16

See Also

Mathematical Operators
ASAP Macro Language
ASAP Commands Overview

Registers for Storing Arithmetic Results

ASAP has 286 direct registers designated by the letters A through Z by themselves or followed by the numbers 0 through 9. (A special set of registers starting with an underscore "_" instead of a letter are reserved for argument passing). These registers can be used for the storage of both intermediate arithmetic results and literals.

Three pieces of information are associated with each of these registers:

- location (A...Z9)
- name (literal up to 16 characters)
- number designation (double precision)

In the following examples, **R** stands for any register. To store the value of any valid arithmetic expression or literal in a register (forming a null entry) or recall a value in a register as an input number, use the following formats:

	Old	**New**	**Precedence Level**
Store numeric in R	expression=R	R=expression	3
Store literal in R	literal=R	R="literal"	0
Recall numeric from R	R.	(R)	
Recall literal from R	R"	R"	
Recall number from register with literal	literal.	(literal)	

where **R** is one of 286 registersA...Z or A0...Z9.

Registers are zeroed and blanked out at program start up. The capability to recall the numeric in a register by referencing the literal stored in that register allows the user to assign a register a variable name and then use that name (up to 32 characters) instead of the short fixed register name.

An unknown variable is automatically assigned to an unused indirect register location starting at 1768 and working down; that is,. up to 1482 user variables can be created before any conflict with the normal direct register set occurs.

As an example, the following input increments the contents of the register/variable, and uses the result as the current input entry:

R.+1=R.!increment register/variable and use result (OLD)

(R=R+1)!increment registe/variabler and use result (NEW)

If the period had been left off the end of the OLD expression or the parentheses left off the NEW, the register would still be incremented, but no input would be passed to the program. The current contents of the registers/variables can be displayed using the $REG predefined macro command.

See Also

GET (ASAP Command)

Mathematical Operators

ASAP Commands Overview

Files Produced by ASAP

ASAP produces and uses the files listed below. Certain files are produced only in response to particular commands (for example, a file with the extension, *.reg is produced only by the $STO command). You may not see all of these files in your directory after running ASAP. In the following table, a number symbol (#) denotes a numerical file extension. The notes referenced in column 1 are listed after the table.

Command(s) to Create File	Extension	Format	Description
$IO OUTPUT ASAP run in batch mode	*.otr	ASCII	text output file
$IO OUTPUT	*.out	ASCII	text output file
$ITER MAP	*.dis	direct-access binary	distribution data file
$STO	*.reg	binary	variables storage file
Automatically created by ASAP	*.usr	ASCII	journal of user commands
CAD DXF	*.dxf	ASCII	AutoCAD-format file
CAD IGS	*.igs	ASCII	IGES-format file
EMITTING GAUSSIAN GRID... RAYSET	virtual.pgs	direct-access binary	ray data storage file
END SYSTEM TO	lastexec.sys	binary	system geometry file
FIELD OPDMAP	bro029.dat	direct-access binary	complex optical field
MODEL ... PLOT ...PLOT	*_#.dis	direct-access binary	BSDF data
Numerous commands	*.dat	binary/ASCII	temporary data file
SAVE	*.his	direct-access binary	ray history file
SAVE k	*.#	direct-access binary	ray data file
See Note 1	bro009.dat	direct-access binary	distribution data file
See Note 2	*.plr	ASCII	screen graphics plot file
See Note 3	*.vcr	ASCII	3D vector file
User-defined	*.ies	ASCII	IES-format data file
User-defined	*.inr	ASCII	text input file
User-defined	*.lib	ASCII	text macro library
User-defined Created by ASAP while reading the file, *.inr containing embedded macros EXPLODE	*.mac	ASCII	text macro library
User-defined	usap3d.#	ASCII	apodization data file
WRITE	*.din	ASCII	distribution data file
GUI (generated	*.ent		Builder file

files)

Notes

- Commands creating files, bro009.dat include: FIELD, FMAP, MAP, OPDMAP, RADIANT, RENDER, and SPREAD.
- Commands creating files, *.plr (in some cases, specific options are required) include: CONTOUR, DIRECTIONAL, ENCLOSED, FFAD, GRAPH, HISTORY, ISOMETRIC, MODELS. . . PLOT, . . . PLOT, PLOT. . ., PLOT3D, PROFILES, RADIAL, RENDER, REPLOT, SPOTS, and TRACE.
- Commands creating files, *.vcr (in some cases, specific options are required) include: CONTOUR, HISTORY, MESH, PLOT. . ., PROFILES, RADIANT, SPOTS, and TRACE.

See Also

File Structure of ASAP

File Structure of ASAP

ASAP is a sophisticated set of 3-D system modeling and optical ray/beam propagation algorithms. It performs the entire ray tracing and derivative calculations.

Ray data is stored in an external file called **VIRTUAL.PGS**. ASAP performs a ray trace by reading and writing ray data from and to this file, while using the optical prescription data stored within data arrays. Since ray data is stored in an external file, the total number of generated rays is a function of your disk space.

Optical fields calculated from a ray trace are stored in several files. If only scalar optical field data is calculated (such as spot diagrams or irradiance patterns), it is stored in a file called **BRO009.DAT**. If complex or vector optical field data are calculated, (such as polarized field amplitudes), they are stored in a file called **BRO029.DAT**.

Plots created in ASAP, whether of a ray trace or other derivative ray trace calculation (such as a spot diagram), are written to a plot file (unless you override this). This plot file typically is named filename.**PLR**, where **filename** is the last loaded .INR file.

Text information is not written to a file unless you specify this. The text information is typically stored in filename.**OUT** or filename.**OTR**. The latter is created when ASAP is run in batch mode. Every command that is interactively entered from the keyboard is archived in a file typically called filename.**USR**.

External editor: If you are not using the Windows dialogs to perform an ASAP analysis, you are most likely creating ASCII files (***.INR**) with an external editor. These files typically contain ASAP commands that describe the optical system geometry and sources of radiation. Commands for performing an actual analysis may be included in the files, or entered interactively via the keyboard. The Editor window in ASAP is accessible from Window> Editing.

See Also

Files Produced by ASAP

Entries Repeated and Incremented

An at-sign "@" embedded between two entries allows the user to enter redundant data in an efficient manner. For example,

```
3@1.4 ==>   1.4 1.4 1.4
-4@1.4 ==> -1.4 1.4 -1.4 1.4
```

A colon (**:**) between two integer entries can be used to represent an increasing or decreasing sequence of integers; for example,

```
3:7 ==>  3 4 5 6 7
4:-1 ==> 4 3 2 1 0 -1
```

If one or both of the entries are less than one in magnitude, an increment that is also less than one in magnitude is used. For example,

```
.1:.5 ==>   .1 .2 .3 .4 .5
.11:.08 ==> .11 .10 .09 .08
```

A colon ":" between two literal entries that differ by only one character or a set of contiguous integers can be used to generate a simple sequence, for example,

```
ALITERAL:DLITERAL ==> ALITERAL BLITERAL CLITERAL DLITERAL
R11:R0 ==>            R11 R10 R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
```

See Also

Mathematical Operators
ASAP Commands Overview

Specifying Complex Numbers

You can enter complex numbers in some of the ASAP dialogs. For example, the Optical Properties dialog accepts a complex index of refraction. The two ways to specify a complex number are:

1 The real and imaginary parts are separated by a forward quote `. For example, you can write

$$0.1 + 0.2i \text{ as } 0.1\text{\`{}}0.2$$

2 An amplitude and phase (in degrees) separated by a backward quote **'**. For example, you can write:

$$2.0' e^{i3.0} \text{ as } 2.0'3.0$$

Commands by Function

The following list is a series of functional groupings of ASAP commands. The series is in the approximate order that you would follow in a typical analysis. For a list of the commands associated with each functional group and a brief description of each command, click the primary command series name: Indented commands come under the main command.

- Define/modify entities or single entity objects
- Define/modify curvedge entities
- Define/modify surfunc entities
- Define/modify lens entities
- Create/modify media, coatings, and scatter models
- Create/modify objects
- Setup plots and verify system
- Standard plot options
- Setup beam creation
- Create rays/beams
- Modify ray/beam data
- Setup trace
- Trace ray/beams
- Analyze ray/beam data
- Modify or use internal ray/beam data as input
- Calculate diffraction/propagation effects
- Display/modify energy distributions
- Save or recover system data and control execution

In addition to the above primary commands, you can define your own commands via the macro input facility. By default, ASAP first checks your input keyword against the primary command list. If it does not find a match, it searches the current macros and library file (if it exists).

If a match is found, the macro is expanded in the normal way, except that the $ or & prefix on the macro name is not needed.

The program line does not echo the internal macro lines. These new commands are therefore indistinguishable from normal program commands. This capability does not alter the normal macro expansion when the macro name prefix is present. This means that a macro can still be run and echoed even if its name conflicts with a primary command.

See Also

- Editor Window
- ASAP Commands Overview
- DOMACROS utility

Define/Modify Entities or Single Entity Objects

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

UPDATE	Control entity updating in storage
ENTITIES , OBJECTS	Start streamlined entity/object input
:	

See Also

List of ASAP Commands by Function

Define/Modify Curvedge Entities

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

CURVES/EDGES	Begin defining curvedge(s)
POINTS	Enter line/arc/control points directly
LINE ,DASHED	Equally divided straight line
ELLIPSE	Elliptical (circular) polygon
RECTANGLE	Rectangular (square)
OVAL	Something between ellipse and rectangle
ROUNDED	Rectangle with crudely rounded corners
RACETRACK	Rectangle with precisely rounded corners
CHARACTER	Edge patterned after a character
ARC	Portion or all of a circle
CONIC	Quadratic segment given conic coefficients
SAWTOOTH	Shape and number of teeth
HELIX	General helical (coiled) curve
BEZIER	Explicit polynomial as same order Bezier
SPLINE	Curvature (G2) continuous cubic segments
USERCURV	Parametric curve or surface using $FCN
COMPOSITE	Combine set of previous edges into one
REPEAT	Repeat a previous edge definition
SAG	Sag the edge
ASCALE	Non-linear asymmetric scaling
ALTER X,Y,Z,Q	Alter specific edge point(s) data
SMOOTH	Smooth a piecewise linear curve
COARSEN	Opposite of SMOOTH
INVERT	Reverse curve's parametric direction
SWEEP POS/DIR/AXIS/OFF	Sweep curve into a surface
IMAGE	Image curve through a specified lens
PATCHES	Points represent Bezier surface patches
UVSPACE	Curve in parametric space of an object
EXTEND	Linearly extend one or both ends.
MATRIX ; ROTATE; SHIFT; SCALE; SKEW; PLACE; ALIGN; XEQ	Linear transformation of edge

See Also

List of ASAP Commands by Function

Define/Modify Surfunc Entities

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

SURFACES/FUNCTIONS	Begin defining surface function(s)
HORN	Horn given profiles
TUBE	Tube-like surface (cone, cylinder)
OPTICAL	Classic rotationally symmetric optic
BICONIC	Surface with two distinct conic profiles
TORUS	Torus (doughnut)
REVOLUTION ,FIT	Rotated 2D curve
AXICONIC	Rotated conic curve with foci
CARTOVAL	Cartesian oval
CONDUIT	Circle swept along planar explicit cubic
SUPERCONIC	Special asphere used in optical design
ASYM	Distort an axially symmetric surface
ZERNIKE	Explicit Zernike polynomial surface
SAMPLED	Explicit surface interpolated from samples
GENERAL/COEFFICIENTS ,EXPLICIT	Enter surface coefficients directly
PLANE	Planar surface
ELLIPSOID	Orthogonal ellipsoid (sphere)
FITTED	Least squares fit to a set of points
USERSURF	User-programmable surface intersection
USERFUNC	User-programmable function
USERSAG	User-programmable radial or toric profile
CORNER	Axis-aligned corner of a cube
REPEAT	Repeat a previous surface definition
FCN	Wrap macro function around surfunc
FMAP	Output a surface function map
MULTIPLE	Expand surface into multiple sheets
ARRAY EXPONENT,BOUNDS/SEARCH	Replicate single patch into spatial array
TEST OFF/POINT/DIRECTION/AXIS	Set test for particular branch
BEND	Bend surface in given direction
LOCAL	Localize surface in a "box"
RENORM	Renormalize surface coefficients
SOLID	Bounding volume formed with local box
PARAMETERIZE	Set local axis for meshing surface
ALTER	Alter specific polynomial coefficients
EXPLICIT	Convert to explicit form
MATRIX; ROTATE; SHIFT; SCALE; SKEW; PLACE; ALIGN; XEQ	Linear transformation of surface

See Also

List of ASAP Commands by Function

Define/Modify Lens Entities

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

LENSES	Begin defining lens entity(s)
SEQUENCE ,CURV/RADI	Arbitrary set of conicoid surface
MIRROR	Single reflecting surface
SINGLET FL/CV/RD,APLANAT	Two refractive surface lens
MANGIN FL/CV/RD	Second surface mirror
DOUBLET	Cemented (achromatic doublet)
TELESCOPE	1 or 2 mirror telescope and corrector
RIGHT	Right-angle prism
PENTA	Penta prism
WEDGE	Wedge of glass
AFOCAL	Afocal beam expander
COMPOSITE	Combine set of previous lenses into one
PERFECT	For object at infinity but realistic
IDEAL	Specify matrices of an ideal lens
REPEAT	Duplicate a previous lens
ALTER X,Y,Z,U,V,W,H,C/R,K,O,M	Alter specific lens conicoid(s) data
IMAGE	Image a global point through lens
ABERRATIONS ,LIST,PLOT	Display aberrations of centered lens
MINIMIZE ,DIST,TLEN,GLTH	Minimize RMS spot subject to constraints
STORE	Store design in ASAP, CODE V, OSLO, or ZEMAX formats
VARIABLES, TH,CV,CC,BN,GL	Declare variables for optimizing a lens
MATRIX; ROTATE; SHIFT; SCALE; SKEW; PLACE; ALIGN; XEQ	Linear transformation of lens

See Also

List of ASAP Commands by Function

Create/Modify Media, Coatings, Scatter Models

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

UNITS	Set system geometry length units
WAVELENGTH ,UNITS	Set MEDIA and COATING wavelengths
MEDIA	Begin defining media (glasses)
,ABSORB,GRIN,CRYSTAL,SCATTER/USER	Indices, absorption, gradient, and birefringence
MODELS ,PLOT	Begin defining scatter models
LAMBERTIAN	Simple constant
HARVEY	Polished (smooth) surface
POLYNOMIAL/TRINOM/BINOM ,FIT	General polynomial with data fitting
NONLINEAR	
VANES, EDGES,DIFFRACT	Vane structure
USERBSDF	User-programmable
PARTICLES/VOLUME ,MIE	Random scattering centers or small spheres
VCAVITY	Rough (random v-cavities) surface
BSDFDATA/RAW DATA	Interpolates from entered values
RMS	Estimates scatter from surface statistics
PHYSICAL	Comprehensive physical reflective scatter
SUM	Sum of other models
COATINGS PROPERTIES /LAYERS /MODELS	Begin defining optical coatings

See Also

List of ASAP Commands by Function

Create/Modify Objects

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

BRANCH	Set position in object name hierarchy
OBJECTS	Begin defining objects
REDEFINE SURF/NORM,THICK,COLOR	Redefine basic options
DEFORM ,AXIS,FCN	Add small user-programmable deformation
BOUNDS/RBOUNDS ,MULTIPLE,POINT	Complex bounding surfaces, curves, volumes
LIMITS AXIS/REPEAT/STATS/EXPAND	Simple orthogonal limiting box
FACETS	Set subdivision of patches into facets
MATRIX;ROTATE;SHIFT,SCALE; SKEW;PLACE;ALIGN;XEQ	Linear transformation of entire object
INTERFACE COATING ,DIFFRACT	Assign optical properties to object
ROUGHNESS ,RANDOM	Affect specular by surface roughness
SCATTER MODEL/RMS/BSDF,RANDOM	Assign scattering characteristics
TOWARDS	Preferential random scattering
SPLIT	Specifies specula ray/beam splitting
LEVEL	Specifies scattered ray/beam level
FRESNEL	Flux variation with incidence angle
HALT	Sets conditions for halting a trace
GROUP	Temporary collection of objects
MATRIX; ROTATE; SHIFT; SCALE; SKEW; PLACE; ALIGN; XEQ	Linear transformation of entire group
EXPLODE	Create separate objects from lens parts

See Also

List of ASAP Commands by Function

Setup Plots and Verify System

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

TITLE	Specify user ID and/or default title
PIXELS ,ON/OFF,FILL	Sampling number and aspect
WINDOW	Define 2D graphics window on 3D data
OBLIQUE	Oblique (nonorthographic) views
VIEW ,CENTER,EYE,DOLLY,ZOOM,ORBIT	Perspective views
PRINT SURF/EDGE/LENS/COAT/MED/OBJ	Full display of system information
SPRINT SURF/EDGE/LENS/COAT/MED/OBJ	Shorter display of system information
QPRINT SURF/EDGE/LENS/COAT/MED/OBJ	Shortest display of system information
DIMENSIONS	Show dimensions of main arrays
NUMBERS ,NAMES/SUMMARY	List currently used entity numbers
PROFILES NOOPTIM,	Draw system profiles (slices)
PLOT	Plot geometry and/or ray data
REPLOT ,OPTIM,NORAYS	Replot all 3D vector graphics
DRAWING ,DIMENSIONS,NORAYS	Draw four views of 3D vector graphics
VUFACETS ,LIST	Facets and views current objects
CONSIDER	Limit current set of objects
ARROWS	Set rescaling of arrows on plots
SEGMENTS	Number of segments per arc
SHOW ,ALL	Display current command settings
COLORS	Set colors by object's interface
LIGHTS	Specify light sources for render
RENDER ,DEPTH,RAYS,MODEL	Render current object surfaces
MAP ,DEPTH,SLOPES	Map current object surfaces
TREE ,ENTITIES	Display object name hierarchy

See Also

List of ASAP Commands by Function

Standard Plot Options

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

...CLIP	Specifies limits boxes for clipping rays
...OVERLAY	Places next plot on top
...COLORS	Overrides normal colors
...TEXT	Annotates plot with 2-D or 3-D text
…PIXELS	Temporarily overrides number of PIXELS
...XY[Z] and Other Plot Window Overrides	Temporarily overrides current WINDOW

See Also

List of ASAP Commands by Function

Setup Beam Creation

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

PARABASAL ,CLIP	Controls parabasal ray calculations
WIDTHS ,EDGE	Sets relative beam width factors
WAVELENGTH	Set operating wavelength
CLIP POS/DIR	Clip rays during creation and for **POS**:
BOUNDS/RBOUNDS ,MULTIPLE,POINT	Complex bounding surfaes,vurves,volumes
USERAPOD POS/DIR/ANG/BOTH/OFF	User-programmable apodization
POLARIZ	Set polarization direction and values
XMEMORY FULL/NORM/MIN	Controls extended ray/beam paging
BEAMS INCOH/COHER/GEOM/DIFF,SHAPE	Sets future beam characteristics
SPECTRUM VIS/SCO/THERM/PHOT/FCN/OFF	Sets spectral weighting for future beams
IMMERSE	Sets starting medium for future beams

See Also

List of ASAP Commands by Function

Create Rays/Beams

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

RAYSET	Begin defining a table of rays/beams
GRID RECT/POLAR/ELLIP/OBJECT/HEX/WIN/DAT	Define a spatial grid of rays/beams
SOURCE POSIT/DIREC/FOCUS/LINE/WAVE	Specify source(s) for the rays/beams
EMITTING	Composite of random emitters
EMIT DISK/RECT	Simulate a random emitting surface
EMIT CONE PYRAMID/BOX/SPHEROID	Simulate a random emitting volume
EMIT FILAMENT/HELIX	Simulate a random emitting curve (wire)
EMIT RAYS	Arbitrary collection of rays
EMIT IES	Emitter specified by IES file data
EMIT ENTITY/OBJECT	Random emission from defined surface(s)
EMIT DATA	Emit according to distribution data file
GAUSSIAN	Rays for coherent astigmatic Gaussian mode
DECOMPOSE POSITION/DIRECTION	Decompose an existing coherent field

See Also

List of ASAP Commands by Function

Modify Ray/Beam Data

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

SELECT SOURCE/OBJECT	Select ray group for future operations
SUBSET ,RESET	Create subset of current rays/beams
FLUX	Assign total flux (power) in rays/beams
SHAPE	Specify shape and coherence of beams
REVERSE	Reverse current ray/beam directions
MOVE BY/TO, POINT/PLANE/SPHERE/FOCI	Move to new positions on rays/beams
BILATERAL	Flip ray/beam data about a plane
IMAGE	Image ray points through a specified lens
APODIZE POS/DIR/ANG/BOTH	User-programmable apodization
MATRIX;ROTATE;SHIFT;SCALE;SKEW; PLACE;ALIGN;XEQ	Linear transformation of ray data

See Also

List of ASAP Commands by Function

Setup Trace

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

WARNINGS	Controls warning messages
AXIS	Local and/or cylindrical coordinates
SEED ,QUASI	Initializes random number seed
SEARCH ALL/SEQUENTIAL/LIST	Object intersection search pattern
ALLOWED ALL/SEQUENTIAL/LIST	Object intersection halt pattern
HALT	Sets conditions for halting trace
CUTOFF	Set absolute flux and number threshold
SAVE ,file#	Saves all intersection data to file
SPLIT	Specifies specular ray/beam splitting
LEVEL	Specifies scattered ray/beam level
FRESNEL	Flux variation with incidence angle
MISSED ARROWS/LINE/OFF	Controls missed ray plotting
ACCURACY HIGH/MEDIUM/LOW	Accuracy of ray/surface intersections
VOXELS FL/AB/X/Y/Z/OFF	Setup volume energy tracking during trace

See Also

List of ASAP Commands by Function

Trace Ray/Beams

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

RAY ,PLOT,DIR,SEARCH,GALOP,LENS	Trace a single ray
TRACE ,PLOT/GRA,LIST/DIR,STATS,STEP	Trace current set of rays/beams

See Also

List of ASAP Commands by Function

Analyze Ray/Beam Data

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

LIST POS /DIR/RAY/SOU/INT/ELL	List specific data for all or some rays
EXTREMES POS /DIR/FLUX/LEN	List data on extreme rays
STATS POSITION/DIRECTION	Statistics of ray/beam data by object
PATHS AVE/PEAK/TOT/OBJ	Grouping and listing of ray path
HISTORY ,PLOT	Display histories of rays in SAVE file
SPOTS POSITION/DIRECTION	Distribution of current ray data
FOCUS ,MODE,MOVE	Find (and move to) best focus
PLOT BEAMS,RAYS,POLARIZATION	Plot beams, rays or polarization
FFAD ,SPOTS,REFERENCE	Full Field Aberration (Spot) Display
OPDMAP file	Interpolate OPDs into file for DISPLAY
RADIANT ,MAP/AREA	Radiant pattern in spherical coordinates
DUMP	Currently selected rays to binary file
COLLECTION	Efficiency vs. aperture and cone angle

See Also

List of ASAP Commands by Function

Modify or Use Internal Ray/Beam Data as Input

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

GET	Load ray data into input registers
PUT	Move values in registers back

See Also

List of ASAP Commands by Function

Calculate Diffraction/Propagation Effects

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

IRRADIANCE	Set irradiance definition direction
VIOLATION	Controls paraxial/positivity/stability messages
SPREAD DIR/POS/APROX/NOR ,DOWN,CLIP	Specular/scattered energy distributions
FIELDSUM,DELT,ADD,MULT,COUPLE,CONT	Coherent beam summation
FIELDBPM ,DELT,MULT,COUPLE,CONT	Finite-difference beam propagation

See Also

List of ASAP Commands by Function

Display/Modify Energy Distributions

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

DISPLAY	Read distribution data, modify/display
NORMALIZE	Normalize by constant or maximum
FORM	Set to power or logarithmic form
FFT ,SIZE	Fourier Transform data
ABEL ,INVERSE	Abel transform data
AVERAGE , WEIGHT	Average data over several pixels
RADIAL ,FUNC/INTEG/BOTH	Radially average about a point
TRANSPOSE	Transpose distribution array
MODIFY	Modify regions of distribution
COMBINE	Multiply or add another data file
REDUCE	Reduce to a smaller subset of data
WRITE	Write current data to ASAP files(s)
DMAP	Print a map of data
RANGE	Override min/max for data plotting
THRESHOLD	Reset floor and/or ceiling data values
PLOT3D	Plot in 3-D data plus major profiles
ISOMETRIC	Isometric plot
GRAPH ,APPEND	Plot 1-D profiles of data
CONTOUR , LOW/HIGH/TICS/VECTOR	Contour or "color" map plot
DIRECTIONAL ,UNWRAP,RADIANCE	Create polar plot of angular distribution
MESH	Write distribution to 3D system file
ENCLOSED	Plot percent enclosed function
SECTION	Display or transfer a section of data
TABLE	Display a table of numeric values
ANGLES ,RADIANCE	Convert direction cosines to angles
VALUES	List value(s) at actual coordinates
OFFSET	Shifts coordinate origin
FOLD ,FIRST/SECOND/BOTH	Average data about one or both centers
PICTURE	Produce gray-scale picture
HISTOGRAM	Plot of data value distribution
IESFILE	Write IES file of angular distribution
TEXTFILE	Create user-definable text file of distribution
DATA	Write actual data in specified format
EOF	End of text file
HEADER	Redefine header (that is, labeling)
REPLICATE	Replicate distribution one or more times
APPEND	Append current data to given file.

See Also

List of ASAP Commands by Function

Save or Recover System Data and Control Execution

- A slash between entries indicates that only one of the entries is allowed at a time.
- Commas indicate that any combination of entries is acceptable.
- A semi-colon indicates that the commands must be grouped together.

SAVE	Writes future ray trace data to a file
SYSTEM NEW/TO/FROM file	Read/write system data to binary file
RESET	Reinitialize all settings
DOMACROS FIRST/LAST/NEVER	Transparent macro execution control
LSQFIT ,NORM/OFF,LIST	Controls SVSD fitting algorithm
FTSIZE	Sets Fourier Transform size
CADEXPORT IGS/DXF/VCR	Export object surfaces to CAD file
RETURN	Return to previous command level
END ,OFF	End execution or ignore future ENDs

See Also

List of ASAP Commands by Function

...a,b,c... (ASAP Command Argument)

Some commands (for example, PLANE NORMAL, ALIGN, RAY, SOURCE DIRECTION) require the specification of a direction vector. The following formats can be used for these three input entries (shown as simply "a,b,c" in the command descriptions).

Input Entries (3)			**Description**
`a`	`b`	`c`	Actual relative components of direction vector
`+/-`	`b`	`c`	Two actual direction cosines, sign of third
`a`	`+/-`	`c`	
`a`	`b`	`+/-`	
`X`	`a`	`a'`	Spherical angles (in degrees) from and around given axis
`Y`			
`Z`			
`DIR`	`RAY`	`n`	Current direction of given ray number
	`SOU`		Average direction of rays from given source
`OBJ`	`n`	`n'`	Direction of vector between positions of the two entities
`SUR`			
`EDG`			
`LEN`			
`RAY`			
`NOR`	`SUR`	`n`	Direction of normal to given entity at its reference point at LIMITS box center
	`LEN`		
	`EDG`		
	`OBJ`		
	`LIM`		
`LOC`	`X`	`n`	Direction of given local coordinate axis of SURFACE n
	`Y`		
	`Z`		
`TAN`	`n`	`u`	Tangent to CURVE/EDGE n at parametric value u.

ASAP automatically normalizes the resulting vector to unit length.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

...ABSORB (ASAP Command Argument)

Absorption or gain option, which is positive for an absorbing medium or negative for a gain (assigning) medium.

Syntax

```
... [ ABSORB a [ j q t ] ]
```

Option	Description
a	volume absorption in inverse length units
j	its magnitude is the SURFACE designation for this function when tracing a ray in this inhomogeneous medium
t	step length

Remarks

- If an absorption is not given (or zero), the program uses the wavelength and the imaginary part of the complex refractive index (if specified) to calculate the absorption.
- Inhomogeneous absorption or gain can be handled by assigning to the medium **a** GENERAL polynomial in the global coordinates X,Y,Z or USERFUNC function (with additional wavelength **w** dependence).
- The magnitude of **j** is the "SURFACE" designation for this function.
- The absorption coefficient at each point in the medium is then given by:

$$a\ \ X,Y,Z \qquad a\ f^{q} \quad w;\ X,Y,Z$$

- The **t** is the step length to be used by the program when tracing a ray in this inhomogeneous medium.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

...CLIP (ASAP Command Argument)

Clips the distribution data according to an object or edge.

Function

Standard Plot Options

Syntax

```
... CLIP [ i ]
          +j
          -j
```

Option	Description
i	specified OBJECT
+j	exterior of the specified closed EDGE
-j	interior of the specified closed EDGE

Remarks

- ASAP command arguments are optional and must follow a command.
- The clipping may be specified by the limits or bounds of **OBJECT i**, the exterior of closed **EDGE +j**, or the interior of closed **EDGE -j**.
- Use **…CLIP** on the SPOTS, OPDMAP, SPREAD, or FIELD commands to clip the distribution data during the calculation. This option clips the distribution by either the specified LIMITS/BOUNDS of OBJECT i, or the given side of entity j.
- A minus (**-**) symbol means the interior of a closed EDGE or the negative side of a SURFACE. A plus (**+**) symbol means the exterior of a closed EDGE or the positive side of a SURFACE.
- Data points outside the specified regions have their distribution values set to zero.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

FIELD
OPDMAP
SPOTS
SPREAD

...ANGLES (ASAP Command Argument)

Several BSDF models allow fitting to or interpolation from actual data. They all use the following common options and format.

Syntax

```
... [ ANGLES ]   [ LOG ]   [ COS ]
    [ ao bo ]                    !!first specular direction (defaults to output)
    a b f [ a' b' f' ...]        !!triplets of output direction and BSDF
    :
    ao' bo'                      !!second specular direction
    :                            !!more triplets
```

Option	Description
`ANGLES`	specifies spherical angle coordinates in degrees
`LOG`	specifies common logarithmic BSDF values
`a a' ...`	user-defined degree specular angles
`ao bo`	first specular direction, polar and spherical angle
`a b [a' b' ...]`	either direction cosine space coordinates or the spherical ANGLES from and around normal, respectively
`f [f' ...]`	either actual BSDF values or the common LOG of the BSDF
`ao' bo'`	second specular direction

Remarks

- ASAP command arguments are optional and must follow a command.
- The **a'**s and **b**'s are either direction cosine space coordinates or the spherical ANGLES from and around normal, respectively.
- The **f'**s are either actual BSDF (times scatter angle COSine) values or the common (base 10) LOG of the BSDF (times scatter angle COSine).
- Except for the RAWDATA model, BSDF values below 1.E-9 (-9 LOG) are ignored.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

BSDFDATA

...COLORS (ASAP Command Argument)

Sets color for current graphics commands.

Function

Standard Plot Options

Syntax

```
... COLORS k [ k' ... ]
```

Option	Description
`k k' ...`	color number (1 through 27)

Remarks

- ASAP command arguments are optional and must follow a command.
- Entities displayed during the current command are plotted in **only** the colors given instead of any default coloring. Remember that the maximum color number 27 (or 0) is equivalent to the background and can be used for erasing in the graphics display.
- Use the **…COLORS** command argument with any of the following **PLOT** commands: EDGES, LENSES, SURFACES, LOCALS, ENTITIES, TRACE, and CURVES.
- ASAP uses a palette of 27 colors to display graphics. The table summarizes the colors and their associated red/green/blue (RGB) values. A reference to an example is given after the table.

Number	Color Name	RGB Value
1	Reverse video	(1.00, 1.00, 1.00)
2	Red	(1.00, 0.00, 0.00)
3	Light Blue	(0.00, 0.60, 1.00)
4	Orange	(1.00, 0.60, 0.00)
5	Yellow	(1.00, 1.00, 0.00)
6	Purple	(0.50, 0.00, 1.00)
7	Sea Green	(0.00, 1.00, 0.60)
8	Maroon	(1.00, 0.00, 0.50)
9	Cyan	(0.00, 1.00, 1.00)
10	Chartreuse	(0.60, 1.00, 0.00)
11	Blue	(0.00, 0.00, 1.00)
12	Magenta	(1.00, 0.00, 1.00)
13	Green	(0.00, 1.00, 0.00)
14	Salmon	(0.88, 0.51, 0.45)
15	Olive Drab	(0.45, 0.50, 0.10)
16	Light Brown	(0.90, 0.65, 0.30)
17	Forest Green	(0.30, 0.58, 0.35)
18	Goldenrod	(1.00, 0.85, 0.40)
19	Plum	(0.55, 0.40, 0.55)
20	Tan	(0.87, 0.87, 0.65)
21	Turquoise	(0.25, 0.80, 0.85)
22	Gold	(0.85, 0.74, 0.10)
23	Thistle	(0.83, 0.65, 0.83)
24	Tomato	(0.95, 0.35, 0.20)
25	Wheat	(0.80, 0.70, 0.55)
26	Violet	(0.60, 0.30, 0.90)
27	Background	(0.00, 0.00, 0.00)

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

COLORS (ASAP Command) - reassigns color for displaying each object

...GRIN (ASAP Command Argument)

Similar to . . . ABSORB, this command argument handles the gradient index (GRIN) materials by assigning to the medium a GENERAL polynomial or USERFUNC function.

Syntax

```
... [ GRIN k p t [ l ] ]
```

Option	Description
k	its magnitude is the SURFACE designation for this function
t	step length used for tracing a ray when tracing a ray in this inhomogeneous medium
l	maximum number of steps a ray can take in the medium (default 1000)

Remarks

- The refractive index squared at each point in the medium is given by:

$$n^2(w;X,Y,Z) = n^2(w) \quad f^p([w;]X,Y,Z)$$

  where **w** is the wavelength.
- If the constant coefficient of the function **f** is unity (1), the refractive indices entered after the MEDIA command will correspond to those at the function's reference point. The **t** is again the step length to be used by the program when tracing a ray in this inhomogeneous medium. The **1** is the maximum number of steps a ray is allowed to take in the medium (default 1000).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

...MINMAX (ASAP Command Argument)

Sets BSDF bounding.

Function

Standard Plot Options

Syntax

```
... MINMAX b b'
```

Option	Description
b	minimum BSDF value
b'	maximum BSDF value

Remarks

- ASAP command arguments are optional and must follow a command.
- Limits the minimum and maximum attainable BSDF values for the commands used to specify scatter MODELS.
- This option can be used on any model definition and only applies to that particular model.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

BSDFDATA
HARVEY
LAMBERTIAN
MODELS
NONLINEAR
PARTICLES
POLYNOMIAL/TRINOMIAL/BINOMIAL
SUM
USERBSDF
VANES
VCAVITY

...MODEL... (ASAP Command Argument)

Scattering from anisotropic surfaces (for example, brushed, diamond-turned) is not rotationally symmetric at normal incidence, and not necessarily symmetric about the plane of incidence otherwise. Therefore, the orientation of the model (such as BSDFDATA, HARVEY, VANES, USERBSDF) on the surface is important and is generically specified by an axis for the second command entry, as shown below.

Syntax

```
... model X ...
          Y
          Z
          U
          V
          W
          R
          T
          A
         -X
         -Y
         -Z
         -U
         -V
         -W
         -R
         -T
         -A
          :
```

where:

Entry	Anisotropic Axis
X Y Z	Global coordinate direction
U V W	CURVE object parametric direction or local x,y,z of SURFACE object
R T A	Radial, angular, or axial direction for LENS, SWEEP AXIS, or PARAMETERized/LOCALized SURFACE object

Remarks

- ASAP command arguments are optional and must follow a command.
- The local "Alpha" direction is perpendicular to the given axis and the surface normal. The "Beta" direction is then perpendicular to the "Alpha" direction and the surface normal. The local direction cosines are the projections of the global direction vector onto these locally orthogonal directions.

Direction	Components	Magnitude
Scatter	A, B	$A^2 + B^2 = SIN^2$ (scatter angle from normal)
Specular	Ao, Bo	$A_o{}^2 + B_o{}^2 = SIN^2$ (specular angle from normal)

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

...OVERLAY (ASAP Command Argument)

Overlays plots (that is, places them on top of each other).

Function

Standard Plot Options

Syntax

```
... OVERLAY [ k ]
```

Option	Description
k	integer quadrant flag

Remarks

- ASAP command arguments are optional and must follow a command.
- The OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.
- Use as a modifier on any ASAP command that produces graphical output.
- The **k** is an integer between 1 and 4 inclusive that tells ASAP to scale the plot down to half size and place it in the **k**th quadrant of the normal plotting area. This feature can be used to overlay but not overlap successive plots (that is, spot diagrams for different source points). The quadrants are numbered in the following manner: 1=upper left, 2=upper right, 3=lower left, 4=lower right. Due to the scaled-down size, no ordinary border text is placed on the plot.
- The default value of **k** is 0, that is, normal full-sized plot, if the last plotting command did not have an OVERLAY option on it. Otherwise it is one more than the last quadrant specified.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PLOT3D
HISTORY PLOT
PLOT BEAMS
PLOT CURVES
PLOT EDGES
PLOT ENTITIES
PLOT FACETS
PLOT LENSES
PLOT LIMITS
PLOT LOCAL
PLOT MESHES
PLOT POLARIZATION
PLOT RAYS
PLOT SURFACES
PROFILES
REPLOT
SPOTS
TRACE PLOT

...PIXELS (ASAP Command Argument)

Sets the number of pixels for the current plot only.

Function

Standard Plot Options

Syntax

```
... PIXELS n
```

Option	Description
n	number of pixels

Remarks

- ASAP command arguments are optional and must follow a command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PIXELS

...PLOT (ASAP Command Argument)

Plots current model.

Syntax

```
... [ PLOT [ a a' ... ] [ OVERLAY ] ]
```

Option	Description
`a a'`	specular angles

Remarks

- ASAP command arguments are optional and must follow a command.
- Plot the model's BSDF for up to seven specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). A unit reflectivity surface in the current IMMERSE medium and at the current WAVELENGTHS is assumed when a model is plotted.
- The current PIXELS setting controls the resolution of these plots in direction cosine space.
- For the default angles and a model sharply peaked at specular, the optimum setting for PIXELS is 580.
- Optionally, the multiple graphs plotted can be OVERLAYed into separate quadrants. If the angles **a a' …** are explicitly given, PLOT also creates a full-hemispherical 2D distribution file, "name_angle.DIS", for each of the specular angles.
- Otherwise, **…PLOT** creates a 3D distribution file, **name.dis**, for each of these angles.
- Use command argument, …OVERLAY to plot multiple graphs into separate quadrants.
- Assumes a unit reflectivity surface in the current IMMERSE media and at the current WAVELENGTH when a model is plotted.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands

...TEXT (ASAP Command Argument)

Prints text annotation within the graphics window.

Function

Standard Plot Options

Syntax

```
... TEXT
x y [ z ] x' y' [ z' ] x" y" [ z" ] [ k ] 'string'
  :
```

Option	Description
`x y [ z ]`	Starting point
`x' y' [ z' ]`	Character spacing vector
`x" y" [ z" ]`	Character height vector
`k`	Color number (1-26)
`'string'`	text annotation

Remarks

- ASAP command arguments are optional and must follow a command.
- Draws text of color **k** (default 1) on 2-D and 3-D plots.
- The number of entries determines whether the starting point, character spacing vector and character height vector are in 2-D plot units or 3-D system units as follows:

  If the number of entries is seven or less, then ASAP assumes 2-D only, paper coordinates (0-10, 0-7.5).

  If the number of entries is eight or more, then ASAP assumes 3-D (and 2-D projection) system coordinates.
- With 2-D coordinates the plot window is 10 inches wide (horizontal or **x** direction) by 7.5 inches high (vertical or **y** direction). Especially useful for placing text relative to the plotting window; using 2-D paper coordinates to place text next to a system OBJECT can be difficult, since the location of plotted **OBJECT**s in the window changes according to the setting of the WINDOW command.
- The starting point and vector entries for placing text on 3-D and 2-D projection plots are expressed in global system coordinates. Remember that the **WINDOW** setting determines the apparent size of the text on a given plot, so size the text in proportion to the size of the **OBJECT**s that you wish to view in the plot **WINDOW**.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

...XY[Z] and Other Plot Window Overrides

Plotting Commands

...XY[Z] and Other Plot Window Overrides (ASAP Command Argument)

Sets the window to the given directions and size for this plot only.

Function

Standard Plot Options

Syntax

```
... XY[Z]  [ m ]
    YZ[X]  a a' d d'
    ZX[Y]
    YX[Z]
    ZY[X]
    XZ[Y]
```

Remarks

- ASAP command arguments are optional and must follow a command.
- If the third direction is given, then an OBLIQUE plot is generated.
- If the plot ranges **a a' d d'** are not given, the plot is autoscaled (and then the window optionally magnified by a factor **m**). A good use of the option is to control the direction cosine space window without affecting the current spatial WINDOW.

Example

```
SPOTS DIR YX -4@1
```

See Also

WINDOW
...OVERLAY
Plotting Commands

ABEL (ASAP Command)

Performs an Abel or inverse Abel transform on each line of data about the horizontal axis.

Function

Display/Modify Energy Distributions

Syntax

```
ABEL [c] [INVERSE]
```

Option	Description
`c`	center of data
`INVERSE`	flag to perform an inverse Abel transform

Remarks

- Performs an Abel or inverse Abel transform on each line of data assuming that the center is located at **c** (absolute or fractional pixel value, default is centroid of data).
- Use **ABEL INVERSE** on CCD images, and EMITTING DATA to model arc lamps.
- Bitmap (*.bmp) files can be read into distribution files (*.dis) using the executable file, BMP2DIS.EXE.
- Text files containing a matrix of data can be read in using DISPLAY 0.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Abel Transform Overview
Importing CCD Images

ABERRATIONS (ASAP Command)

Displays the image aberrations of all the current lens or conicoids **l** through **l'** (up to 120 total).

Function

Define/Modify Lens Entities

Syntax

```
ABERRATIONS [ l l' ] [ LIST ] [ PLOT ] [ datum d ] [ datum' d' ]
```

Remarks

Note: Since the analysis is valid only for centered lens systems, the conicoids must have a common axis. If not, the lens is temporarily "unfolded" to make it so.

- The first-order operating data must be supplied with the additional entries, which is the data for the marginal axial ray from the center of the "object" through the edge of the limiting aperture stop, and the chief ray from the edge of the "object" to the center of this stop.
- Object Space (before first conicoid):

Datum	Default	Description
U0	0	Marginal ray slope (tangent of angle)
UB0	.01	Chief ray slope (tangent of semi-field angle)
H0	1st aperture	Marginal ray height at entrance pupil plane
H1	1st aperture	Marginal ray height at first conicoid
HB1	0	Chief ray height at first conicoid
TH0	0	Distance from entrance pupil to first conicoid
KTH0	none	Conicoid number of actual aperture stop (0 to vary, .1 for telecentric in image space)
FB	0	Fractional evaluation fields (up to 5; for example, 0,.5,.7,.866,1)

- Image space (after last conicoid):

Datum	Default	Description
UF	initial	Marginal ray slope after last conicoid (curvature solve)
BKF	NA	Fix back focal distance (last conicoid to paraxial focus)
THF	0	Distance from paraxial focus to desired focal surface
CVF	0	Curvature of focal surface
CCF	0	Conic constant of focal surface
KTHF	NA	Solve for best axial position of focal surface
KCVF	NA	Solve for best curvature of focal surface
KCCF	NA	Solve for best conic constant of focal surface

- If the LIST option is present, the following surface-by-surface tables are produced:
  Paraxial marginal and chief ray traces.
  Seidel primary/secondary aberrations (3rd-order, axial/lateral color).
  Real marginal and chief ray traces.
  Glass and conicoid data (after solves).
- In any case, the final 1st-order (primary color), 3rd-order, 5th-order, and selected 7th/9th-order aberration coefficients (in waves) are listed for actual stop plane coordinates and relative to the ideal paraxial focus. These coefficients are computed using both analytical formulas and real-ray data matching.

Note

ASAP/Pro carries out all these calculations in double precision. However, since ASAP/Basic uses only single precision here, accuracy may suffer for either a large number of surfaces or delicately balanced configurations.

- The optional PLOT displays the traced lens and the following graphs:
  Ray fans (image deviation versus aperture) for each evaluation field.
  Chief ray sagittal (S) and tangential (T) foci as a function of field.
  Chief ray (D) and spot centroid (C) fractional distortion versus field.
  Actual (A) polychromatic and best (B) monochromatic RMS spot size vs. field.

- Except for the RMS spot size graphs, solid curves are computed from the approximate aberration coefficients (with color terms) while dotted curves are center-wavelength real-ray points.
- The **ABERRATIONS** command can be followed by any number and combination of the command listed below under "See Also".

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

VARIABLES
MINIMIZE
STORE

ACCURACY (ASAP Command)

Controls the accuracy of the ray trace algorithm.

Function

Setup Trace

Syntax

```
ACCURACY [ HIGH ]
           MEDIUM
           LOW
```

Remarks

- **ACCURACY** controls the tradeoff between ray/surface intersection accuracy and calculational speed.
- **HIGH** (the startup default) should be used when parabasal rays must be as accurate as the base rays and optical path lengths are required to be accurate to a small fraction of a wavelength over great distances. This is usually necessary for coherent high-resolution analyses.
- **LOW** leaves the parabasal rays on the base ray's tangent plane. This is usually sufficient for incoherent straylight and illumination analyses.
- A **PROFILE** calculation is also affected by this command.
- Typical relative error values used in iterative solvers are:

LOW	1.E-4
MEDIUM	1.E-7
HIGH	1.E-10

- If no option is specified, the setting reverts to the one before the previous **ACCURACY** command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PROFILE

AFOCAL (ASAP Command)

Creates a two-element afocal telescope.

Function

Define/Modify Lens Entities

Syntax

```
AFOCAL X x l h h' m m'
       Y y
       Z z
```

Option	Description
X or Y or Z	global coordinate axis
x or y or z	location on the global coordinate axis
l	overall length of the afocal telescope
h	input height
h'	output height
m m'	media

Remarks

- The telescope elements may be both refractive, both reflective, or mixed.
- This lens entity starts out normal to the defined global coordinate axis (X, Y or Z).
- The conic constants of the two elements are adjusted so that there is no axial aberration.
- If **h** and **h'** are of opposite sign, there is an intermediate real focus.
- For a reflecting element, **m** should be **-1** or **REFL**.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

ALIGN (ASAP Command)

Aligns a given direction vector with another.

Function

Create/Modify Objects
Define/Modify Surfunc Entities
Define/Modify Lens Entities
Create/Modify Objects
Modify Ray/Beam Data

Syntax

```
ALIGN a,b,c a',b',c' [ x,y,z ] [ LIST ]
```

Option	Description
a,b,c	first direction vector
a',b',c'	second direction vector
x,y,z	fixed point that the rotation axis passes through (default is the entity's reference point)
LIST	prints the resulting 4-by-4 transformation matrix

Remarks

- Points an entity in a particular direction if you know where it is currently pointing, and you want it to point somewhere else. Since it performs the most direct coordinate rotation, an ambiguous clocking rotation may occur. Useful when setting up systems designed on other optical software and all you have to go on are global vertex coordinates and global normal vectors.
- The rotation axis passes through a fixed point that is either point (**x,y,z**) or the entity's reference point (default).
- The LIST option causes the resulting 4-by-4 transformation matrix to be printed and decoded into simple operations if possible.
- Some commands, including SOURCE DIRECTION, require the specification of a direction vector. The following format can be used for this input entry:
- See General Input Techniques for various formats that are supported for the a, b, and c entries. For instance, to align the average source direction with the positive Z-axis direction, you can use the format, ALIGN 0 0 1 DIR SOU 1. See the following example.

Example

```
EMIT RECT Z 0 0 0 1
LIST DIR
SURFACE; PLANE Z 0
ALIGN 0 0 1 DIR SOU 1
PRINT
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

ALLOWED (ASAP Command)

Controls whether a ray is allowed to proceed after intersecting an object.

Function

Setup Trace

Syntax

```
ALLOWED [ LIST ]
        [   ALL          FORWARD     ]
            SEQUENTIAL   BACKWARD
i           j           j'           j"
      :
```

Option	Description
`LIST`	lists SEARCH settings
`ALL`	all objects are candidates for intersection
`SEQUENTIAL`	searches objects sequentially
`FORWARD`	searches objects forward
`BACKWARD`	searches objects backward
`i j j' j"`	on object i, searches objects j through j' in steps of j"

Remarks

- Determines whether a ray is allowed to proceed once it has reached the next object.
- For an **ALL** input, where all objects are candidates for intersection, the current object is considered for the next intersection with the rays.
- The range of possible objects can be selectively set by additional commands, with the first entry **i** being the number for the particular object, followed by a pair of entries.
- You must run the **ALLOWED** command after defining all objects. If you do not run it, ASAP issues an error message.
- The **LIST** option prints the current allowed settings for each object.
- With **m** being the largest possible object number, the various options for ray intersection after OBJECT i are tabulated according to the following table.

Input	Object Range		Increment
`j j' j"`	j	j'	j"
`ALL`	1	m	1
`ALL FORWARD`	i	m	1
`ALL BACKWARD`	i	1	-1
`SEQUENTIAL`	i-1	i+1	2
`SEQUENTIAL FORWARD`	i+1	i+1	
`SEQUENTIAL BACKWARD`	i-1	i-1	

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SEARCH

ALTER (Edge Modifier) (ASAP Command)

Alters edge database.

Function

Define/Modify Curvedge Entities

Syntax

```
ALTER n [  n'  [n"  ]  ]  data a  [  b  [  j  ]  ]  data' a'  [  b'  [  j'  ]  ] ...
```

Option	Description
`n  n'`	edge points range
`data, data' ...`	ALTER parameters
`a b, a' b' ...`	ALTER parameter scale factors
`j, j' ...`	edge points range used for calculation

Remarks

- Alters data on a set of edges allowing, for example, pickups on other edges.
- The given data on points n through n' (default n) for the current edge are altered as follows for the current edge; that is, for i = n to n' by n" (default 1):

```
data  (i) = a  + b  data  (j)  ' defaults" b=0, j=I ]
data' (i) = a' + b' data' (j')
                 :
```

- The **data** parameters are shown in the following table.

Data	Description
`X`	x coordinate of edge point
`Y`	y coordinate of edge point
`Z`	z coordinate of edge point
`Q W`	connection or weighting factor

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

ALTER (Lens Modifier) (ASAP Command)
ALTER (Surface Modifier) (ASAP Command)
REPEAT (ASAP Command)

ALTER (Lens Modifier) (ASAP Command)

Alters lens conicoid database.

Function

Define/Modify Lens Entities

Syntax

```
ALTER n [ n' [ n"  ] ] data a [ b [ j ] ] data' a' [ b' [ j' ] ] ...
```

Option	Description
n n'	conicoid range (n' is defaulted to n)

Remarks

- Alters data on a set of conicoids allowing, for example, pickups on other conicoids.
- Conicoids **n** through **n'** (default **n**) for the current lens are altered as follows for **i=n** to **n'** by **n"** (with defaults **b**=0, **j=i**):

```
data  (i) = a +  b  data  (j)
data' (i) = a' + b' data' (j')
```

The **data** parameters are as shown:

Data	Description
X	x coordinate of vertex position
Y	y coordinate of vertex position
Z	z coordinate of vertex position
U	x component of vertex normal
V	y component of vertex normal
W	z component of vertex normal
H	Aperture height
C R	Vertex curvature or radius
K	Conic constant or 4th-order aspheric
O	Obscuration (hole) ratio
M	Medium number
C0 R0	Curvature or radius of subtracted parabola
S	Aspheric scale factor

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

ALTER (Edge Modifier) (ASAP Command)
ALTER (Surface Modifier) (ASAP Command)
REPEAT (ASAP Command)

ALTER (Surface Modifier) (ASAP Command)

Edits specific polynomial coefficients to form a differently shaped or oriented surface.

Function

Define/Modify Surfunc Entities

Syntax

```
ALTER term a [ b ] [ term' a' [ b' ] ...
```

Remarks

- ASAP is capable of handling any implicit surface that can be represented by an arbitrary function of a general **n**th order polynomial in the global coordinates X,Y,Z, with arbitrary reference point x,y,z and real coefficients c. **ALTER** changes the coefficient of the given polynomial **term** to be **a** plus **b** (default 0) times its current value. For example, to multiply the constant term by 2 and change the Z term to -1:

  ```
  ALTER C 0 2 Z -1
  ```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

ALTER (Edge Modifier) (ASAP Command)

ALTER (Lens Modifier) (ASAP Command)

ANGLES (ASAP Command)

Converts the current distribution data to an angle space RADIANCE or radiant intensity distribution.

Function

Display/Modify Energy Distributions

Syntax

```
ANGLES [RADIANCE]
```

Remarks

- ANGLES converts a direction cosines space radiance distribution, which is created by SPREAD or SPOTS DIRECTION, to an angle space RADIANCE or radiant intensity distribution.
- The polar axis of the spherical angle coordinate system is assumed to be horizontal (IES type B photometry).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DIRECTIONAL
DISPLAY

APPEND (ASAP Command)

Appends the current data to the end of distribution file number and updates the header accordingly.

Function

Display/Modify Energy Distributions

Syntax

```
APPEND [ u ]
         name
```

Option	Description
u	distribution file number
name	distribution file name

Remarks

- Appends the current data to the end of distribution file number **u** or **name.DIS** (the default is the one loaded on entry to DISPLAY) and updates the header accordingly.
- Due to the fixed structure of a BRO binary distribution file, the following restriction applies: each data set must have the same number of rows and columns.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

ARC (ASAP Command)

Creates a circular arc on the plane.

Function

Define/Modify Curvedge Entities

Syntax (long form)

```
ARC X x y [ z y' z' [ a ] ]
    Y y z   x z' x'
    Z z x   y x' y'
```

Syntax (short form)

```
ARC X x r
    Y y
    Z z
```

Option	Description
X, Y or Z	coordinate axis of the plane
x, y or z	location of the plane on the coordinate axis
y [z], z [x], or x [y]	arc starting point
y' z', z' x', or x' y'	arc center (default 0,0)
a	arc subtense (default 360 degrees)
r	radius of the arc (short form)

Reference Point

Arc center

Remarks

- The short form allows you to enter a complete circle of radius **r** centered on the axis.
- This edge is a combination of coplanar straight line and higher-order curved segments.
- Arcs up to 135 degrees are represented by a single, quadratic, rational, Bezier segment. Beyond that, the arc is subdivided into more of these segments (up to 4 for a full 360 degrees).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

ARRAY (ASAP Command)

Turns the last surface into a set of identical but linearly spaced surfaces.

Function

Define/Modify Surfunc Entities

Syntax

```
ARRAY n x y z [ n' x' y' z' ] [ EXPONENT p [ p' ] ] [ BOUNDS       ]
       X n s      Y n' s'                             SEARCH [ k ]
                  Z
       Y          Z
                  X
       Z          X
                  Y
```

Option	Description
ARRAY	elements are associated with an OBJECT
n n'	number of linearly spaced elements
x y z x' y' z'	reference point shift
EXPONENT	modifies reference point coefficient
p p'	reference point coefficient exponent
BOUNDS	treats each instance as a separate entity
k	number of additional "rings" of instances about the nearest one to also consider

Remarks

- Turns the last surface into an n+1 by n'! set of identical but linearly spaced (s and s' apart) surfaces. For the first, more general (possibly skew) syntax, the replicated surfaces are spaced according to the equation:

$$f\left(X - ax - bx', Y - ay - by', Z - az - bz'\right) = 0$$

- The **EXPONENT** option allows the reference point coefficient to be modified as follows:

$$a = i^{p} \ i = 0, n \ b = j^{p'} \ j = 0, n'$$

- The defaults for **p** and **p'** are both 1.
- All the **ARRAY** elements are associated with an OBJECT when the internal SURFACE is referenced by an **OBJECT**.
- Optionally, each instance can be treated as a separate entity on a BOUNDS command. Otherwise, if used as a base object surface, the SEARCH for any ray intersection is restricted to the instance nearest to where the ray intersects the plane of the array (default or **k** zero).
- **k** option: for large arrays, this option can speed up the trace calculation by orders of magnitude, but runs the risk of some rays unphysically missing the object. Larger values of **k** reduce this risk but slow down the trace.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

ARROWS (ASAP Command)

Toggles between displaying arrows on rays and not displaying them.

Function

Setup Plots and Verify System

Syntax

```
ARROWS [ s   ]
         OFF
```

Option	Description
s	rescaling factor
OFF	suppresses appearance of an arrow

Remarks

- Toggles the displaying of arrows on the PLOT RAYS, TRACE PLOT, and PLOT POLARIZATION commands.
- The **s** is a rescaling factor used to change the size of the arrows relative to the internal setting (default is 1).
- **OFF** is used to suppress the display of arrows.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

WINDOW

ASCALE (ASAP Command)

Deforms an edge or curve into a keystone.

Function

Define/Modify Curvedge Entities

Syntax

```
ASCALE s s'
```

Option	Description
`s`	scale factor at the bottom of the edge/curve
`s'`	scale factor at the top of the edge/curve

Remarks

- Scales the previous edge along the axis that follows the normal axis of the edge, for example, an ELLIPSE Z is scaled in the x direction.
- The scaling varies linearly from s at the bottom to s' at the top of the edge (as measured along the third coordinate direction), as shown in the following table where the coefficients **a** and **b** are computed from **s s'** and the height of the edges/curves.

Edges/curves		
Coordinate axis	**Scaled axis**	**Equation**
X	**Y**	**Yscaled = Y (a Z + b)**
Y	**Z**	**Zscaled = Z (a X + b)**
Z	**X**	**Xscaled = X (a Y + b)**

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT (ASAP Command)

ASYM (ASAP Command)

Modifies surfaces to make a cylindrical or anamorphic surface.

Function

Define/Modify Surfunc Entities

Syntax

```
ASYM u v
```

Option	Description
`u v`	scale factors

Remarks

- Distorts the rotational symmetry of the surface/functions generated by TUBE**,** OPTICAL, TORUS, REVOLUTION, HORN, CARTOVAL , AXICONIC, BICONIC, CONDUIT, and SUPERCONIC commands to essentially distort the rotational symmetry of the surfaces. For example, if **Z** is the axis of symmetry, the surfaces are only a function of the distance from the axis squared:

$$r^2 = x^2 + y^2 \rightarrow ux^2 + vy^2$$

- Allows the creation of a more general class of surfaces such as anamorphic **OPTICAL** surfaces.
- Setting either **u** or **v** to zero makes a surface with cylindrical symmetry become an infinite trough in the nonzero direction.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

AVERAGE (ASAP Command)

Smoothes current distribution data by averaging adjacent pixels.

Function

Display/Modify Energy Distributions

Syntax

```
AVERAGE [ i [ j ] ] [WEIGHT]
```

Option	Description
`i`	pixels in the first direction (default=1)
`j`	pixels in the seconds direction (default=i)

Remarks

- Smoothes the data before displaying it by averaging over **i** (default 1) adjacent pixels in the first direction and **j** (default=**i**) in the second. For example, **i**=1 and **j**=1 replaces each pixel with the average of the 3x3 box of pixels centered on it; **i**=2 and **j**=3 uses a 5x7 box, and so on.
- This operation is also equivalent to convolving the distribution with a finite window.
- If **i** and/or **j** are entered as negative numbers, the median value within the smoothing window is used instead of the normal average value.
- Optionally, a WEIGHTed average can be done assuming that the error in each pixel within the given window is proportional to the square root of its value; that is, Monte Carlo statistics.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
FOLD
RADIAL

AXICONIC (ASAP Command)

Creates a surface by spinning an arbitrarily oriented conic curve.

Function

Define/Modify Surfunc Entities

Syntax

```
AXICONIC X x h x' h' s [ x" h" ] [ LENGTH t ] [ LIST ]
         Y y h y' h' s [ y" h" ] [ HEIGHT ]
         Z z h z' h' s [ z" h" ]
```

Option	Description
X, Y, or Z	coordinate axis
x h, y h, or z h	coordinates of a point on the surface
x' h', y' h', or z' h'	coordinates of first focal point
x" h", y" h", or z" h"	coordinates of second focal point
s	angle in degrees of second focus (if it is moved to infinity) relative to coordinate axis
LENGTH	distance from 3rd entry
HEIGHT	distance from axis
LIST	lists the coefficients of the second order curve

Reference Point

At location along coordinate axis of first point

Surface Normal

Radially outward from the axis

Autolimiting

Yes, if LENGTH or HEIGHT option is used

Remarks

- Generates a surface by rotating an arbitrarily oriented conic curve about the given axis.
- Similar to the REVOLUTION command, except that the second order curve (whose coefficients can be **LIST**ed) is determined from the foci of a conic.
- Since the surface obeys Fermat's principle, the distance (on any plane) from the first focal point to the second focal point through the surface is stationary (or constant).
- **AXICONIC** has two modes of operation:
  1. The positions of the two foci can be specified, in which case, the value of **s** is irrelevant and the sign of **s** determines whether the sum or difference of the distances is stationary, that is, an ellipse or hyperbola.
  2. The position of one focus only can be specified. In this case, the second focus is moved to infinity and **s** becomes the angle (in degrees) it makes with the axis of symmetry. This form of the command is useful in constructing compound parabolic concentrators.

Specifies other end of curve with **LENGTH** or **HEIGHT** option.

An optimal **LOCAL** box for the surface can be automatically created if 1) the other end of the curve is specified either by an axial **LENGTH** (distance from 3rd entry), or 2) a **HEIGHT** (distance from axis). The program solves a quadratic for the other coordinate taking the root closest to the first unprimed point (if there are no real roots, no box is created). **TEST** and **PARAMETERIZ**ation are also automatically set so that one branch (maybe not the desired one) of the surface should mesh properly for **PLOT**ting.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

LOCAL
REVOLUTION

AXIS (ASAP Command)

Selects the coordinate system in which future ray trace data is printed.

Function

Define/Modify Surfunc Entities
Setup Trace

Syntax

```
AXIS [ OFF  ]
        X
        Y
        Z
       -X
       -Y
       -Z
        LOCAL [ i ]
```

Option	Description
`OFF`	resets ASAP to the default global Cartesian coordinates
`LOCAL`	causes data to be output in a local Cartesian coordinate system (see Remarks)
`X, Y, Z, -X, -Y, or -Z`	reports ray trace data in cylindrical coordinate relative to the specified axis

Remarks

- Allows future ray positions to be printed in either local and/or cylindrical coordinates relative to the specified axis.
- The origin of the local Cartesian coordinate system is the reference point of the object's first entity.
- **LOCAL** causes data to be output in a local Cartesian coordinate system whose origin is the reference point of object **i**'s base identity. The default for i is each ray's current object.
- If the axis designation is preceded by a minus sign, a local cylindrical coordinate system is used.
- The cylindrical coordinates are **R** (radial distance from the axis), **T** (angle theta in degrees around the axis), and the actual axial coordinate (X, Y or Z if positions, and A, B or C if directions).
- The combination of an AXIS LOCAL command and a window override on a SPOTS, RADIANT, OPDMAP, SPREAD or FIELD command forces the distribution to be calculated in a local Cartesian coordinate system.

```
--- AXIS LOCAL
--- SPOTS ...window
OPDMAP
SPREAD
FIELD
RADIANT ... MAP
AREA
```

- To distinguish this from the global case, the coordinate axes in the header of the resulting distribution file (or files) are in lower case. Also, the Euler angles and positional offsets needed to transform the distribution back into global coordinates are stored in the header. If a reference object is not specified on the AXIS LOCAL command, the distribution is in the local coordinates of the base entity associated with the object that the first valid ray/beam resides on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

BEAMS (ASAP Command)

Sets the default coherence, propagation, and shape characteristics for future ray creation.

Function

Setup Beam Creation

Syntax

```
BEAMS [ INCOHERENT ] [ GEOMETRIC ] [ SHAPES k [ s ] ]
        COHERENT       DIFFRACT
```

Option	Description
`INCOHERENT or COHERENT`	flag for default coherence
`GEOMETRIC or DIFFRACT`	flag for default propagation
`SHAPES`	flag for beam shape characteristics
`k`	Hermite-Gaussian mode
`s`	optional Hermite-Gaussian mode data

Remarks

- Sets the default coherence, propagation, and shape characteristics for future beam creation, **except** for the EMITTING and SCATTER rays, which are incoherent, geometric, and of a fixed shape by definition.
- Refer to the SHAPE command for a detailed explanation of the arguments **k** and **s**.
- In most circumstances, the **INCOHERENT** option should be used with the **GEOMETRIC** option and the **COHERENT** option with **DIFFRACT** option. This is the default if only one is specified. Although it is possible to specify the other two "cross" combinations, the results can be unexpected and, therefore, useful in only a few rare cases (for example, BEAMS COHERENT GEOMETRIC SHAPE FIBR/SLAB, for a non-diffracting, guided mode). .
- The default in the absence of a **BEAMS** command depends upon the WAVELENGTH and XMEMORY settings at the time of the first ray/beam creation. If WAVELENGTH = 0 or XMEMORY MIN, then the default is **INCOHERENT GEOMETRIC**; otherwise it is **COHERENT DIFFRACT**.
- Once set, the **BEAMS** settings are not affected by future **WAVELENGTH** and **XMEMORY** commands unless a **BEAMS** command with no arguments is issued.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING
SHAPE
WAVELENGTH
XMEMORY

BEND (ASAP Command)

Bends the last surface along a parabolic curve.

Function

Define/Modify Surfunc Entities

Syntax

```
BEND X c [ c' ]
     Y
     Z
```

Option	Description
`X Y or Z`	axis which is bent
`c c'`	curvature coefficients of the bending

Remarks

- Bends the surface (relative to its reference point) by replacing the given coordinate in its polynomial function with a quadratic in the other two coordinates, for example, for a BEND Z:

$$Z \rightarrow Z + \frac{1}{2}\left(cX^2 + c'Y^2\right)$$

- The **c** coefficients are the curvatures (inverse radii) of the bending. Note that this operation may double the order of the polynomial.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

BEZIER (ASAP Command)

Creates a Bezier edge/curve in the plane.

Function

Define/Modify Curvedge Entities

Syntax

```
BEZIER X  x  y  y' c [ c' [ c" ... ] ]
       Y  y  z  z'
       Z  z  x  x'
```

Option	Description
`X, Y or Z`	coordinate axis
`x, y, or z`	axial location of the plane
`y y', z z' or x x'`	range of the segment in the plane
`c c' c" ...`	order of the explicit polynomial that defines the Bezier segment (not more than 20)

Remarks

- Creates a single Bezier segment in the plane.
- The order of the Bezier segment is the same as the order of the explicit polynomial given by the remaining coefficients and must not exceed 20.
- For a segment in the Z=z plane, the exact equation of the curve is:

$$Y = c + c'X + c''X^2 + \ldots \quad x < X < x'$$

- This edge is a combination of coplanar straight line and higher-order curved segments.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

CONIC

BICONIC (ASAP Command)

Creates a cubic or quartic biconic surface.

Function

Define/Modify Surfunc Entities

Syntax

```
BICONIC X  x r  r' k  k'  [ 3 ]  [aperture...  ]
        Y  y               4
        Z  z
```

Option	Description
X, Y or Z	axis of symmetry
x, y or z	location along axis
r r'	vertex radii of curvature
k k'	conic constants (0 circle, -1 parabola, etc.)
3 or 4	specifies implicit cubic or explicit quartic surface
aperture	ELLIPSE, RECTANGLE, or HEXAGONAL

Reference Point

At intersection of the surface and coordinate axis.

Surface Normal

Along positive coordinate direction.

Remarks

- Creates a cubic (default) or quartic biconic surface with a given vertex radii of curvature (**r,r'**) and conic constants (**k,k'**).
- The second entry designates the axis of symmetry (either **X**, **Y**, or **Z**) for the surface.
- In the case of **Z**, the actual equations of the surfaces are given by:

  Implicit Cubic (3):

$$Z-z=\frac{x^2}{2-e(Z-z)}+\frac{y^2}{2-e'(Z-z)}$$

  Explicit Quartic (4):

$$Z-z=\frac{x^2+y^2}{1+\sqrt{1-ex^2-e'y^2}}$$

$$\text{where :}$$

$$c=\frac{1}{r},\ c'=\frac{1}{r'},\ e=(1+k)c,\ e'=\left(1^2+k'\right)c',\ x^2 2=cX,\ y2=c'Y$$

  The two representations differ in how they blend the two conic profiles together (See the USERAG command for yet another distinct blending).

  If **e(Z-z)** and **e'(Z-z)** are much less than one, then the differences between the two versions are normally negligible. However, the cubic is also conic (usually an ellipse) in any plane perpendicular to the given axis, and its ray intersection is more robust, while the quartic profile is a conic in any plane perpendicular to the given axis, and its ray intersection is more robust, but it degenerates on-axis when converted to implicit polynomial form.
- This surface can extend to infinity unless a LOCAL command follows, or a trailing aperture option of the following form is specified:

```
    ELLIPSE   a [ a' [ o [ s [ s' ] ] ] ]
    RECTANGLE
```

```
     HEXAGONAL a [ o [ s [ s' ] ] ]
```

- **a a'** are the heights in the other two transverse directions.
- For the **HEXAGONAL** form, **a** is the center-to-vertex distance (maximum height).
- **o** is an optional central hole ratio.
- **s s'** are the transverse coordinates of the center of the aperture.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

BILATERAL (ASAP Command)

Folds data around a coordinate axis/axes to reduce the number of rays needed for a calculation where symmetry is expected.

Function

Modify Ray/Beam Data

Syntax

```
BILATERAL  POSITION   X [ Y,Z ]
           DIRECTION  Y [ X,Z ]
                      Z [ Y,X ]
           OFF
```

Option	Description
POSITION	folds positional ray data
DIRECTION	folds directional ray data
X Y Z	coordinate axes
OFF	turns off previous BILATERAL settings

Remarks

- In systems with bilateral symmetry, the BILATERAL command and its options can be used to reduce the number of rays traced, and still produce meaningful plots from the SPOTS command.
- Any ray, whose data specified by the second and third option (POSITION, DIRECTION) is negative, is mirror-imaged about the zero coordinate plane, so that the particular data becomes positive.
- For systems with symmetrical quadrants, two coordinate directions should be entered.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

FOLD

BOUNDS (ASAP Command)

Assigns surfaces and edges as boundaries to objects.

Function

Create/Modify Objects

Syntax

```
BOUNDS [   k   [ k' ... ] [ MULTIPLE m [ m '... ] ] ]   [ POINT x y z ]
RBOUNDS  OFF
```

Option	Description
`k k'`	numbers of the surface and/or edge entities
`MULTIPLE m [ m' ... ]`	makes a group of surfaces/edges equivalent to one standard constraint
`POINT x y z`	makes point (**x y z**) a valid point on the object
`RBOUNDS OFF`	temporarily turns off all boundary surfaces for the current object

Remarks

- The **k**'s are the numbers of the surface and/or edge entities that are to be used to define the boundaries of the object surface for future ray creation. If these numbers of positive, a valid ray is assumed to be on the positive side of the corresponding surface, outside the surface's SOLID or outside the closed edge. A negative number means that a valid ray is on the surface's negative side, inside the surface's SOLID or inside the edge.

BOUNDS	Edge Effect
+	Object is assumed to be on positive side of the corresponding surface, outside the surface's SOLID or outside the closed edge.
-	Object is on the surface's negative side, inside the surface's SOLID or inside the edge.

- The **MULTIPLE** option makes a group of surfaces/edges equivalent to one standard constraint; In other words, a point that is valid for of any one or more of the given **m** bounds is a valid object point. To efficiently enter a very large number of boundaries, use instead the **RBOUNDS** version, where every pair of **k**'s or **m**'s specifies a range of entity numbers instead of individual entities.
- The **POINT** option switches the signs of the entities, if necessary, to make point (**x y z**) a valid point on the object.
- A **BOUNDS OFF** command can be used to temporarily turn off all boundary surfaces for the current object. A BOUNDS command by itself turns them back on.
- Consult your version of ASAP for the maximum number of bounding surfaces/edges that may be applied to a given object (See the DIMENSION command).
- **BOUNDS** is used for more sophisticated surface boundaries. For very simple bounding of an object the **LIMITS** command can be used.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

LIMITS

BRANCH (ASAP Command)

Specify an absolute or relative branch of the object name tree for future object definitions.

Function

Create/Modify Objects

Syntax

```
BRANCH [ name ]
         .name
           ^
        ^.name
```

Remarks

- Specify an absolute or relative branch of the object name tree for future object definitions and display the current or new full branch name.
- Periods (**.**) are used as node separators in the name. Carets (**^**) are used to move up one node. If the entry begins with either of these two characters, then the branch is defined relative to the current branch.
- Otherwise, it is an absolute complete branch specification (with possibly many embedded periods).
- The number of levels is limited by the total object name length, which is 344 characters.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

TREE

BSDFDATA (ASAP Command)

Indirectly or directly interpolates from entered BSDF values.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax: Indirect (isotropic) interpolation

```
BSDFDATA [ n ]  [ d ][options…]  [ PLOT  [ a a' ... ] ]
RAWDATA
 data…
  :
```

Syntax: Direct (anisotropic) interpolation

```
BSDFDATA  X [ d ]  [ options ... ]
RAWDATA   Y
          Z
          :
  data ...
   :
```

Option	Description
n	collapsed 4-D direction space
d	small directional distance (default 0.001 radians)
PLOT	plots the model in log(b-b$_o$) and angle space
a a' ...	user-defined degree specular angles
X,Y,Z	see . . . MODEL… command argument

Remarks

1. Indirect (isotropic) interpolation:

- The four-dimensional direction space is collapsed to just **n** (default 2) that enforce isotropic-surface symmetry and reciprocity (See the BINOMIAL model).
- The **d** is a small directional distance (default 0.001 radians) used to determine if two input data points have collapsed to the same **n**D point. The BSDF or log(BSDF) (RAWDATA or BSDFDATA, respectively) is then interpolated as a function of these **n** directional coordinates:

n	Coordinates	Interpolation
1	sqrt(T)	linear
2	sqrt(T),V	bilinear
3	U,V,W	nearest point or reciprocal point

2. Direct (anisotropic) interpolation:

- Creates a model that directly interpolates from the entered BSDF data. The option **d** is a small directional distance (default .001 radians for BSDFDATA, .000001 for RAWDATA) used to determine if two input data points are reciprocal. The BSDF or log(BSDF) (RAWDATA or BSDFDATA, respectively) is then interpolated as a function of the four directional coordinates.
- Since no assumptions concerning symmetry are made, data must be provided over most of the output hemisphere for most of the input hemisphere.

3. Both indirect and direct interpolation:

- Scattering from anisotropic surfaces (for example, brushed, diamond-turned) is not rotationally symmetric at normal incidence, and not necessarily symmetric about the plane of incidence otherwise. Therefore, the orientation of the model on the surface is important and is generally specified by an axis for the second command entry. For syntax information, see ...MODEL... (ASAP Command Argument).
- If the entered data is not from an isotropic surface or if it violates reciprocity, the interpolation may not sufficiently match the original data (that is, it may be noisy).
- Works with a single normal-incidence scan.
- Processed points are written TO and read FROM SYSTEM file.

- Small data sets (<95 points) are placed in main memory storage.
- This model uses an internal 10000-point buffer. If the final number of data points the model needs to store exceeds this, a disk file is used and runtimes can increase dramatically. In this case, use a larger value of **d** to further collapse the input data to fewer than 10000 final data points.
- The …PLOT option plots the model (common base 10 logarithm of the BSDF) for up to seven specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). The current PIXELS setting controls the resolution of these plots in direction cosine space.
- Creates a distribution file **name_angle.dis** for each of these angles.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.
- Several BSDF models allow fitting to or interpolation from actual data. See …ANGLES command argument for common options and format.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

...ANGLES (ASAP Command Argument)

MODELS Overview

MODELS

CADEXPORT (ASAP Command)

Writes out CAD-compatible system geometry file.

Function

Save or Recover System Data and Control Execution

Syntax

```
CADEXPORT [ IGS [ name          ] ]
            DXF        [ block ]
            VCR
```

Option	Description
IGS	writes out a fixed IGES file (the default)
DXF	writes out an AutoCAD DXF file
name	names the file
block	holds the DXF file standards in the named file

Remarks

- This command was formerly named "CAD"; however, CAD is an accepted abbreviation.
- Writes a 3D CAD surface representation of the currently considered objects, or the 3D graphics in the current VCR file.
- CADEXPORT supports the fixed IGES **IGS** (the default) and AutoCAD 12 **DXF** file standards (including the option to put everything into a named "block"). The file created is "name" followed by one of these extensions.
- The current limitations on surface export are:
  1. Advanced surface effects (REDEFINE SURF/NORM, DEFORM) and trimming information (LIMITS, BOUNDS) are not translated.
  2. IGS usually requires that the system UNITS must have already been set.
  3. All base surfaces in DXF are, by necessity, approximated by facets.
  4. LENS, connected CURVE/EDGE, and SAMPLED base surfaces are exported exactly to IGES. Since no CAD standard supports general algebraic surfaces, all other SURFACES are approximated by cubic spline patches or revolved cubic spline curves when possible.
  5. An often better translation to AutoCAD can be done by post-processing the VCR file created by PLOT FACETS or VUFACETS with the VECT2DXF.EXE Command Prompt utility.
- Currently, trimming information (LIMITS, BOUNDS) is not translated.
- IGS usually requires that the system UNITS must have already been set.

Example

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

CARTOVAL (ASAP Command)

Creates a Cartesian oval surface.

Function

Define/Modify Surfunc Entities

Syntax

```
CARTOVAL X x d n d' n' [ aperture... ]
         Y y
         Z z
```

Option	Description
`X, Y, or Z`	specifies axis of symmetry
`x, y, or z`	location along axis of symmetry
`d`	distance from surface to first focus point
`n`	refractive index of first medium
`d'`	distance from surface to second focus point
`n'`	refractive index of second medium
`aperture`	ELLIPSE, RECTANGLE, or HEXAGONAL

Reference Point

At intersection of surface and coordinate axis

Surface Normal

Along positive coordinate direction

Remarks

- Creates a Cartesian Oval surface at the given location on the given axis.
- The second entry designates the axis of symmetry (either **X**, **Y**, or **Z**) for the surface.
- The surface perfectly images an axial point a distance **d** in the first medium of index **n** into an axial point a distance **d'** in the second medium of index **n'**.
- This surface can extend to infinity unless a LOCAL command follows, or a trailing aperture option of the following form is specified:

```
ELLIPSE   a [ a' [ o [ s [ s' ] ] ] ]
RECTANGLE
HEXAGONAL a [ o [ s [ s' ] ] ]
```

- **a a'** are the heights in the other two transverse directions.
- For the **HEXAGONAL** form, **a** is the center-to-vertex distance (maximum height).
- **o** is an optional central hole ratio.
- **s s'** are the transverse coordinates of the center of the aperture.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

CHARACTER (ASAP Command)

Creates an edge of a specified string.

Function

Define/Modify Curvedge Entities

Syntax

```
CHARACTER X x y [ z ] string
          Y y z   x   'string'
          Z z x   y
```

Option	Description
`X, Y or Z`	character plane
`x, y or z`	character plane location
`y, z or x`	character width
`z, x or y`	character height (defaulted to the width)
`string or 'string'`	edge character string

Reference Point

First point of character

Remarks

- Creates an edge patterned after the character **string**.
- This edge is a combination of coplanar straight line segments.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

CLIP (ASAP Command)

Specifies positional or directional ray limiting or clipping boxes.

Function

Setup Beam Creation

Syntax (Short Form):

```
CLIP POS x x' [ y y' z z' ] [ X [ f ] ]
     DIR a a'   b b' c c'     Y
                              Z
                              SHORTEST
                              LONGEST
```

Syntax (Long Form):

```
CLIP POS [ OFF       ]
     DIR   AXIS        X [ f ] ]
           -X   x        Y
                a        Z
           -Y   y        LONGEST
                b        SHORTEST
           -Z   z        OFF
                c
           +X   x'
                a'
           +Y   y'
                b'
           +Z   z'
                c'
           EXPAND r
                  X r
                  Y
                  Z
```

Option	Description
`POS or DIR`	type of ray clipping
`x x' y y' z z'`	lower and upper **POS**itional clipping entries
`a a' b b' c c'`	lower and upper **DIR**ectional clipping entries
`X Y Z`	specifies cylindrical (as opposed to rectangular) clipping
`f`	specifies fractional hole
`SHORTEST`	selects clipping axis according to shortest clip box dimension
`LONGEST`	selects clipping axis according to longest clip box dimension
`OFF`	turns off cylindrical clipping
`EXPAND`	scales clipping box by relative amount **r**

Remarks

- The **CLIP** command is useful for restricting propagation directions of entity rays to improve the ray trace efficiency.
- If initialized, positional clipping is done in all cases except the single RAY command. Directional clipping is done only for SOURCE DIR GRID, EMITTING, and DECOMPOSE commands.
- The command specifies the **POS**itional or **DIR**ectional (direction cosines) limits for boxes clipping rays during ray creation. Any ray with coordinates falling outside the specified clipping box is not created. The unprimed entries are the lower bounds of the box; primed entries, the upper bounds.
- If a coordinate direction (that is, **X, Y,** or **Z**) is specified, the coordinates are constrained by a cylinder of constant elliptical (as opposed to rectangular) cross section in planes perpendicular to the given axis. The SHORTEST limit box dimension or the LONGEST can also determine this axis.
- An optional inner boundary of fractional height **f** (default zero) may be used to put a proportional hole in this the given

coordinate direction.

- The short form can also be used to temporarily turn **OFF** future clipping or reset the limits **AXIS** or just one of the six limit values.
- **EXPAND** can be used to enlarge (or shrink) by a relative amount **r** the entire limits box or just in one direction, that is, **EXPAND -.1** shrinks the entire box by 10 percent.
- For CLIP POS, more complex ray bounding can be done by following it with the BOUNDS command syntax.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DECOMPOSE
EMITTING
RAY
SOURCE DIR GRID

COARSEN (ASAP Command)

Converts the current edge to a piecewise linear curve.

Function

Define/Modify Curvedge Entities

Syntax

```
COARSEN [ m ]
```

Option	Description
m	use every **m**th point

Remarks

- Converts the current edge to a piecewise linear curve using every **m**th (default 1) point. This is the opposite of the **SMOOTH** command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SMOOTH

COATINGS LAYERS (ASAP Command)

Creates a coating specified by its multilayer prescription.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
COATINGS [ k ] LAYERS [ w [ a ] ]
d m [ d' m' d" m" ... ] [ 'name' ]
      :
```

Option	Description
`k`	starting coating number
`w`	reference wavelength in vacuum
`a`	angle of incidence (degrees)
`d  d' d" ...`	layer thicknesses
`m  m' m" ...`	media numbers (or names)

Remarks

- Specifies coatings on the basis of their actual layer structure.
- The default value of **k** is one more than the largest coating number defined and is set to 1 at the start of program execution.
- The layer thicknesses **d** are entered in fractional optical waves if reference (vacuum) wavelength **w** is entered. The **a** is the angle of incidence at which the thicknesses are desired; the default for **a** is zero (normal incidence).
- The layer thicknesses **d** are entered in absolute WAVELENGTH units if reference wavelength **w** is absent.
- Since the MEDIA command can define complex refractive indices at several wavelengths, the effects of absorption and dispersion in each layer can be modeled precisely.
- Groups of layers may be replicated any number of times by following a set of layer definitions with a negative **d** (the magnitude of **d** is the number of previous layers to replicate) and positive **m** (the number of times the layers are to be replicated).
- Layers are defined from the low index side (usually the air side) to the high index side (usually the substrate side). ASAP flips the coating prescription as needed so the same coating may be applied to both sides of a lens.
- In ASAP/Pro, the maximum number of unique **LAYERS** is 100.
- ASAP can use the transmission (or reflection) values from a COATING PROPERTIES table or calculate the coefficients from a **COATINGS LAYERS** table (based on the normal incident form of Fresnel's formulae) to determine the optical properties of a given object INTERFACE.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Media Overview

Coatings Overview

COATINGS MODELS (ASAP Command)

Specify angular properties for reflection and transmission.

Syntax

```
COATINGS [ k ] MODELS i j m n
r t  r' t' r" t" ... [ 'name' ]
     :
COATINGS [ k ] MODELS
r i j t m n  r' i' j' t' m' n'  r" ... [ 'name' ]
     :
```

Option	Description
`k`	starting coating number
`r  r'  r" ...`	real energy (or complex amplitude) reflectances
`t  t'  t" ...`	real energy (or complex amplitude) transmittances

Remarks

- Starting with coating **k**, coatings with real energy (or complex amplitude) reflectances **r** and transmittances **t** are entered.
- The default value for **k** is one more than the largest coating number defined and is set to one at the start of program execution.
- Separate angular properties can be specified by using previously defined (usually RAWDATA) MODELS where:

  iModel for reflected S polarization
  jModel for reflected P polarization
  mModel for transmitted S polarization
  nModel for transmitted P polarization
- Optionally, groups of six numbers can be entered so that each group corresponds to a wavelength entered on the last multiple WAVELENGTH(S) command. For example, the actual reflectivities and transmissions at an incidence angle **a** and a wavelength **b** between the first two

  WAVELENGTH(S) w w' would be:

$$R_s(a,v) = \left| r(1-u)f(i,a) + r'uf(i',a) \right|^2$$

$$R_p(a,v) = \left| r(1-u)f(j,a) + r'uf(j',a) \right|^2$$

$$T_s(a,v) = \left| t(1-u)f(m,a) + t'uf(m',a) \right|^2$$

$$T_p(a,v) = \left| t(1-u)f(n,a) + t'uf(n',a) \right|^2$$

where $u = \dfrac{(v-w)}{w'-w}$.

- In the above equations, **r**, **r'** and **t**, **t'** are the entered complex amplitudes or the square roots of the real energy coefficients. The **f** is a normalized angular amplitude:

$$f(i,a) = \sqrt{\frac{BSDF_i(a,a)}{BSDF_i(0,0)}}$$

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Coatings Overview
MODELS
WAVELENGTH(S)

COATINGS PROPERTIES (ASAP Command)

Creates a coating specified by its transmissive and reflective properties.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
COATINGS [ k ]  [ PROPERTIES [ a ] ]
r [ t  r'  t'  r"  t" ... ]   [ 'name' ]
    :
```

Option	Description
k	starting coating number
a	angle of incidence
r r' r" ...	real energy (or complex amplitude) reflectances
t t t" ...	real energy (or complex amplitude) transmittances
'name'	string for the name is limited to 16 characters

Remarks

- Specifies coatings on the basis of their wavelength properties, angular **MODELS**, or their actual **LAYER** structure.
- The default value of **k** is one more than the largest coating number defined, and is set to 1 at the start of program execution.
- The **a** is the angle of incidence at which the thicknesses are desired; the default for **a** is zero (normal incidence).
- If more than one pair of data are entered, the coefficients correspond to the wavelengths entered on the last WAVELENGTHS command.
- ASAP linearly interpolates, if necessary, to obtain the coefficients at any desired wavelength.
- ASAP can use the transmission (or reflection) values from a **COATING PROPERTIES** table or calculate the coefficients from a COATINGS LAYERS table (based on the normal incident form of Fresnel's formulae) to determine the optical properties of a given object INTERFACE.
- The program then interpolates (linearly in complex amplitude) to get the coefficients at any desired wavelength. For example, the actual reflectivities and transmissions at a wavelength **v** between the first two **WAVELENGTHS w w'** would be:

$$R(v) = \left|r(1-u) + r'u\right|^2 \qquad T(v) = \left|r(1-u) + r'u\right|^2 \qquad u = \frac{(v-w)}{(w'-w)}$$

  The **r**'s and **t**'s in the above equations are the entered complex amplitudes, or the square roots of the real energy coefficients.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Coatings Overview

COLLECTION (ASAP Command)

Calculates the collection PERCENT efficiency or FLUX of the current rayset.

Function

Analyze Ray/Beam Data

Syntax

```
COLLECTION  [ PERCENT  ELLIP  HEIGHT  m  ANGLE  n  a  ]
              FLUX     RECT   AREA       SINE      MAX
                                         GCF
                                         OMEGA
```

Option	Description
m	number of spatial samples (default 100)
n and a	number of direction samples up to the angle **a** in degrees (default 90 or rayset MAX)
HEIGHT	geometric-mean height of the RECTangle or ELLIPse (0 to max)
AREA	Actual area of the RECTangle or ELLIPse (0 to full area)
Functional Coordinates:	
SINE	sine of the angle (0 to 1)
GCF	sine squared of angle (0 to 1)
OMEGA	solid angle (0 to 2 pi)
MAX	maximum rayset

Remarks

- Calculates the collection PERCENT efficiency or FLUX of the current rayset for **m** (default 100) spatial samples and **n** (default 90) direction samples up to the angle **a** in degrees (default 90 or rayset MAX). The spatial limits are determined from the current WINDOW setting and are either its full RECTangle or the inscribed ELLIPse.
- The first choice listed for each literal item is its default if it is left off the end of the command. The result is written to the default distribution file (BRO009.DAT), and can then be processed by the DISPLAY command and its subcommands.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

COLORS (ASAP Command)

Reassigns the color used to display each object.

Function

Setup Plots and Verify System

Syntax

```
COLORS [ i [ j [ k [ l [ m [ m' ... ] ] ] ] ] ]
```

Remarks

- Reassigns the color used to display each object according to the last group assignment or dominant interface property as shown in the following table.

COLOR	INTERFACE
`i`	Group
`j`	Absorbing
`k`	Reflecting
`l`	Transmitting
`m m' ...`	Media 1, Media 2, ...

- ASAP uses a palette of 27 colors to display graphics. The following table summarizes the colors and their associated red/green/blue (RGB) values:

Number	**Color Name**	**RGB Value**
1	Reverse video	(1.00, 1.00, 1.00)
2	Red	(1.00, 0.00, 0.00)
3	Light Blue	(0.00, 0.60, 1.00)
4	Orange	(1.00, 0.60, 0.00)
5	Yellow	(1.00, 1.00, 0.00)
6	Purple	(0.50, 0.00, 1.00)
7	Sea Green	(0.00, 1.00, 0.60)
8	Maroon	(1.00, 0.00, 0.50)
9	Cyan	(0.00, 1.00, 1.00)
10	Chartreuse	(0.60, 1.00, 0.00)
11	Blue	(0.00, 0.00, 1.00)
12	Magenta	(1.00, 0.00, 1.00)
13	Green	(0.00, 1.00, 0.00)
14	Salmon	(0.88, 0.51, 0.45)
15	Olive Drab	(0.45, 0.50, 0.10)
16	Light Brown	(0.90, 0.65, 0.30)
17	Forest Green	(0.30, 0.58, 0.35)
18	Goldenrod	(1.00, 0.85, 0.40)
19	Plum	(0.55, 0.40, 0.55)
20	Tan	(0.87, 0.87, 0.65)
21	Turquoise	(0.25, 0.80, 0.85)
22	Gold	(0.85, 0.74, 0.10)
23	Thistle	(0.83, 0.65, 0.83)
24	Tomato	(0.95, 0.35, 0.20)
25	Wheat	(0.80, 0.70, 0.55)
26	Violet	(0.60, 0.30, 0.90)
27	Background	(0.00, 0.00, 0.00)

- In inverse video mode, only black and white switch; all other colors remain the same.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

...COLORS - sets colors for current graphics commands.

COMBINE (ASAP Command)

Combines the current distribution data file with a previously calculated distribution data file.

Function

Display/Modify Energy Distributions

Syntax

```
COMBINE [ u    ]  [ c   ]
          name      fcn
```

Option	Description
name	file name of the distribution data file (extension is .DIS)
u	logical file number
c	scale factor
fcn	functional combination of two data files

Remarks

- Use for analysis of general wavefronts.
- Adds, subtracts, or multiplies the current distribution data file with a previously calculated distribution data file on a pixel-by-pixel basis.
- The previously calculated data file may be specified by its logical file number **u** (BROxxx.DAT) or by **name**. The file extension for a named distribution data file is *.DIS.
- If **u** is less than one, the original file specified on entry into DISPLAY is used.
- If **c** is not equal to zero, the current data is added to **c** times the given file. If **c** is absent or zero, the two distributions are multiplied together.
- Since the files are combined on a pixel-by-pixel basis, it is important that they have the same number of pixels.
- The functional combination uses a previously defined function to create new data for each pixel. The function parameters passed are _1 (the original distribution pixel data value), and _2 (the combined file pixel data value).
- The record buffer is 10,000 pixels per line.
- Ability to do an arbitrary combination with a $FCN definition.
- To divide two distribution data files, use:

```
$FCN RATIO (_1/_2)
DIS filename1
COMBINE "filename2" RATIO
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

COMPOSITE (Edge Modifier) (ASAP Command)

Combines several edge entities into a single edge entity.

Function

Define/Modify Curvedge Entities

Syntax

```
COMPOSITE [   -n    ] [ GAPS q [ q' ] ]
            i [ i']
```

Option	Description
`-n`	last n edges
`i i'`	edges i through i'
`GAPS q q'`	connection factors

Remarks

- COMPOSITE combines the last **n** edges, edges **i** through **i'**, or all edges defined since the last **EDGE** command into one edge.
- **GAPS** between the endpoints of one edge and the start of the next can be handled as follows.

q	ACTION
-1	merge gap points into one
0	leave gaps open (default)
1	linearly connect gaps

- Specify **q'** for the last-to-first point gap to make a single edge object from a set of discontinuous edges.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

COMPOSITE (Lens Modifier) (ASAP Command)

Combines several lens entities into a single lens entity.

Function

Define/Modify Lens Entities

Syntax

```
COMPOSITE [   -n    ]
            i [ i']
```

Option	Description
`n`	last n lens entities
`i i'`	range of lens entities

Remarks

- Combines the last **n** lenses, lenses **i** through **i'**, or all defined lenses since the last **LENS** command into one lens entity.
- IDEAL or PERFECT lens entities are not allowed in a composite set.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

CONDUIT (ASAP Command)

Sweeps a circle along a planar explicit cubic.

Function

Define/Modify Surfunc Entities

Syntax

```
CONDUIT X x x' r [ y [ a [ a' ] ] ]
        Y y y'     z
        Z z z'     x
```

Remarks

- Creates up to a 6th-order surface formed by sweeping a circle of radius |r| along an explicit cubic planar curve starting at the first axial location and ending at the second.
- If r is positive, the circle is perpendicular to the curve at every point. If it is entered as a negative number, the circle is always perpendicular to the specified axis.
- The end of the curve can be displaced from the axis by the amount given on the sixth entry (default 0).
- The optional last two entries are the starting and ending angles in degrees of the curve relative to the axis. Their defaults are zero and any values entered must be (usually much) less than 90 degrees in magnitude.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

CONIC (ASAP Command)

Creates a quadratic Bezier edge/curve in the plane given at the 2nd and 3rd entries.

Function

Define/Modify Curvedge Entities

Syntax

```
CONIC X x  c y z y2 yz z2   y' z'  y"  z"
                            Y  y' y" [ p ]
                            Z  z' z"
       y  c z x z2 zx x2   z' x' z"  x"
                            Z  z' z" [ p ]
                            X  x' x"
       z  c x y x2 xy y2   x' y' x"  y"
                            X  x' x" [ p ]
                            Y  y' y"
```

Option	Description
Using the first form as an example:	
`X`	coordinate axis
`x`	location along coordinate axis
`c  y  z  y2  yz  z2`	implicit coefficients of this conic curve
`y' z'`	coordinates of start point of the curve
`y" z"`	coordinates of end point of the curve
`Y y' y"`	specifies **y** coordinates of start and stop points, solve for **z' z"**
`Z z' z"`	specifies **z** coordinates of start and stop points, solve for **y' y"**

Remarks

- Entries 4 through 9 are the implicit coefficients of this conic curve. Entries 10 and 11 are the coordinates in this plane of the start point of the segment, and 12 and 13 the stop point.
- Instead of giving a pair of coordinates for each point, only one coordinate value for each point can be given and ASAP solves for the others (smallest root) using the Y y' y", or Z z' z", or X x' x" syntaxes.
- This edge is a single and quadratic curved segment.
- If this is the case, **p** is an optional vertex radius of curvature of a parabola to subtract from the curve.

Example

To define the equivalent of a classical optical conicoid:

```
CURVE
  R=1  !vertex radius of curvature
  K=-1 !conic constant
  H=1  !outer aperture height
  O=0  !inner obscuration height
  CONIC X 0, 0 0 -2*R 1 0 K+1, Y (O) (H)
   SWEEP AXIS 360 Z
  OBJECT
   INTERFACE . . .
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

BEZIER

CONSIDER (ASAP Command)

Selects a set of objects for ray tracing and plotting.

Function

Setup Plots and Verify System

Syntax

```
CONSIDER [ ALL  ]
           NONE
           ONLY    [ i i' ... ]
           EXCEPT
           ADD
           REMOVE
```

Option	Description
`ALL`	consider all known objects
`NONE`	consider no objects
`ONLY`	consider only the objects specified (default is the current GROUP)
`i i' ...`	OBJECT numbers or names to be considered
`EXCEPT`	consider all objects except those specified (default is the current GROUP)
`ADD`	adds the specified objects to the previous CONSIDER command
`REMOVE`	removes the specified objects from the previous CONSIDER command

Remarks

- Provides control over the current set of objects ASAP is to consider in all calculations and output. All objects remain in the database at all times (that is, even when they are temporarily ignored with options other than ALL, they are not deleted).
- A **CONSIDER** command by itself (with no entries) produces a list of the currently considered objects.
- By default all objects in the system database are used.
- The **EXCEPT** option restricts the current object set to all objects except those specified (default is the current GROUP). All objects except those may be drawn or ray traced.
- Particular object numbers (or names with "?" wildcards) can be either excluded with the EXCEPT option or the ONLY ones considered. If no object list is given, the last GROUP is used.
- The **ADD** option adds the specified objects to the previous CONSIDER command.
- The current state (that is, a previous CONSIDER command) can be updated using ADD or **REMOVE** instead of ONLY or EXCEPT, respectively.
- Ray data are initially referenced by OBJECT 0. Before ray tracing, **CONSIDER EXCEPT 0** effectively turns off the ray data just as it would any other object.
- Ray data currently associated with a given object, that is, after a ray trace, are affected in exactly the same way as the object itself.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SELECT
NUMBER

CONTOUR (ASAP Command)

Creates a contour plot of the current distribution data file.

Function

Display/Modify Energy Distributions

Syntax

```
CONTOUR n                [ LOW  ] [ TICS t [ t' ] ] [ VECTOR ] [ 'title' ]
        c c' [ c" ... ]    HIGH     GRID
```

Option	Description
`n`	number of equally spaced contour levels
`c c' c". . .`	fractional contours relative to the full range of the function being plotted or absolute contours (see remarks below)
`LOW`	produce a low resolution greyscale plot
`HIGH`	produce a high resolution color map plot
`t t'`	coordinate TIC mark spacing
`VECTOR`	writes contour plot to 3D vector file
`'title'`	optional title for plot (up to 64 characters)
`GRID`	produces line spacing for vertical and grid horizontal lines

Remarks

- Generates a contour plot with fractional contours **c c' ...** relative to the full range of the function being plotted (0=minimum, 1=maximum).
- If any of the **c**'s are less than zero or greater than one, they are assumed to be absolute.
- Alternatively, **n** equally spaced contours can be specified.
- Also, a **LOW**-resolution greyscale or **HIGH**-resolution color map plot can be produced instead of the line contour plot.
- If the coordinate **TIC** mark or **GRID** line spacings **t t'** (vertical, horizontal) are specified, the plot is slightly reduced in size and drawn with annotated coordinate scales.
- If **VECTOR** is specified, the contour plot is written to the 3-D vector file. As an example, irradiance plots may be viewed with already existing system geometry plots by using the REPLOT command.
- The title is delimited by a single quote (**'**), as shown.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

CORNER (ASAP Command)

Creates an axis-aligned corner of a cube (three mutually perpendicular planes), with the apex at the location given by the third entry.

Function

Define/Modify Surfunc Entities

Syntax

```
CORNER X x HEIGHT d
       Y y HEXAGONAL
       Z z LENGTH
```

Option	Description
HEIGHT	circular corner cube
HEXAGONAL	six-sided corner
LENGTH	axial corner cube

Remarks

- Creates an axis-aligned corner of a cube (three mutually perpendicular planes) with an apex at the location given by th third entry.
- The size of the corner cube can be specified either by a circular **HEIGHT**, **HEXAGONAL** height, or an axial **LENGTH** (maximum cross-section will then be triangular).
- Note that a small hole is created at the apex since the normal to the function becomes undefined there. Therefore, an on-axis ray will never hit the surface(s).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

CUTOFF (ASAP Command)

Sets the conditions controlling ray termination.

Function

Setup Trace

Syntax

```
CUTOFF [ t ] [ n ]
```

Option	Description
t	absolute flux threshold
n	maximum number of total object intersections for any ray

Remarks

- Sets the absolute flux threshold below which ASAP ignores the rays to the decimal number **t** (default 1.E-18).
- The maximum number of total object intersections for any ray can be set to the integer **n** (default 1000).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

HALT

DECOMPOSE (ASAP Command)

Decomposes an existing field into a new set of Gaussian beams.

Function

Create Rays/Beams

Syntax #1

```
DECOMPOSE [ u ]
    n.e
```

Syntax #2

```
DECOMPOSE  [ u ]   POSITION  [ x y z CON ]  [ ADJUST ]
            n.e   -POSITION          DIV
                  +POSITION    a,b,c PLA
```

Syntax #3

```
DECOMPOSE  [ u ]   DIRECTION [ m  [ a  [ a' ]  [ RECT ]  ]  ]
             n.e  -DIRECTION
                  +DIRECTION
```

Option	Description
`x y z`	center of the CONverging or DIVerging spherical reference wavefront
`CON`	specifies a converging spherical reference wavefront
`ADJUST`	produces an accurate decomposition
`DIV`	specifies a diverging spherical reference wavefront
`a,b,c`	direction of the PLAnar reference wavefront
`PLA`	specifies a planar reference wavefront
`m`	number of Gaussian beams created
`a a'`	limiting cone angles for the Gaussian directions (default 90 degrees)
`RECT`	specifies limiting cone angle pattern to be rectangular instead of the elliptical default
`DIRECTION`	decomposition Fourier transform

Remarks for Syntax #1

- The **DECOMPOSE** command reads the complex coherent field stored in the distribution file (number or name with extension). The default is 29 or BRO029.DAT, which is normally created by a FIELD command. The field is then decomposed into a new set of propagating Gaussian beams.
- If the field is a complex vector, that is FRESNEL BOTH is set before the **FIELD** calculation), only the component specified by the last POLARIZATION command is decomposed.
- Negligible beams are not created; that is, their relative flux is below the CUTOFF command floating entry or their relative flux is below HALT.
- For example, to decompose properly a vector field propagating mostly in the Z direction:

```
RAYS 0
POLARIZ X; DECOMPOSE ...
POLARIZ Y; DECOMPOSE ...
```

- The new ray set is automatically added to the current ray set, and a new source with the current WAVELENGTH command is created. If, however, the **DECOMPOSE** command is immediately preceded by a RAYS 0 command, the current ray set is deleted before the new ray set is created.

Remarks for Syntax #2

- The **POSITION**al decomposition creates a straight forward spatial distribution of beams, one per input field pixel (as defined by the last PIXELS command) and is used only when the pixel size is a few wavelengths or larger.

  The direction of each beam in the new grid is adjusted to be normal to the local phase front.

  To make this determination more robust, a **CON**verging or **DIV**erging spherical (centered at **x y z**) or **PLA**nar (with direction **a,b,c**) reference wavefront can also be specified.

  The first setting on the last WIDTH command controls the overlap of beams.

Remarks for Syntax #3

- The **DIRECTION**al decomposition Fourier transforms the field and creates up to **m** (actual integer number or floating point fraction of maximum possible, default 0.1) Gaussian beams whose sum closely approximates the original field distribution. The centers and waists of the beams are all located at the center of the original field distribution. The beams all have the same widths (chosen such that they are uniformly distributed in the far field), but different propagation directions.

  The **a**'s are the limiting cone angles in degrees (default is 90, that is, a full hemisphere) for the Gaussian directions and should be set to the acceptance cone of the subsequent optical system.

  When the **RECT** option is used this cone is rectangular instead of elliptical.

  If you want the number of beams given to just fill the given angular cone, then (assuming **a=a'** and a square WINDOW) the spatial sampling interval of the original field must be:

  $$d = \frac{\sqrt{\frac{m}{c}}\lambda}{N\sin(a)}$$

  where λ is wavelength, N is the size of the FFT used (set by the last FTSIZE command), and **c** is either 4 for **RECT** or π otherwise. Therefore, the **PIXEL** setting for the original field should be the **WINDOW** size divided by *d*.

  The overlap of the beams in the far field is controlled by the first setting on the last **WIDTH** command. The **ADJUST** option takes into account this overlap and produces a more accurate decomposition in most cases.
- The accuracy of the decomposition is related to the number of beams and their maximum far field angle. As a check, it is a good idea to issue a SPOTS DIRECTION and a SPREAD NORMAL command just after decomposing and before continuing to TRACE the beam set to verify the source (beam set) fidelity.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Apodization of Ray Distributions

FIELD

FRESNEL BOTH

POLARIZATION

RAYS 0

Simulating Spatial Filters

WIDTH

WINDOW

DEFORM (ASAP Command)

Adds small deformations to an object.

Function

Create/Modify Objects
Setup Plots and Verify System

Syntax

First Syntax (Short format):

```
DEFORM k [ k' ]
```

Second Syntax (Long format):

```
DEFORM x y z  a [ a' a" ... ] [ AXIS a,b,c ] [ FCN fcn ]
```

Option	Description
`k k'`	explicit surface functions
`x y z`	point on the object's surface
`a a' a" ...`	aspheric deformation coefficients (up to 20)
`AXIS a,b,c`	normal direction of axis
`FCN fcn`	optional macro function

Remarks

- With the short format, one or two general deformation functions defined entirely by the given explicit surface functions (GENERAL EXPLICIT, FITTED EXPLICIT, ZERNIKE, SAMPLED, EXPLICIT or USERSAG) are added to the object surface.
- With the long format, a small user-definable aspheric deformation is added to the previous object. This deformation is rotationally symmetric about the axis defined by either the normal vector that passes through the point (**x y z**) on the object's surface or the given point (**a,b,c**) and **AXIS**.
- The deformation or sag value as a function of perpendicular distance from the aspheric axis (r below) is given by:

$$\text{sag} = \text{fcn}\left(ar + a'r^2 + a''r^3 \ + \ \ldots\right)$$

where **fcn** is the name of an optional macro function (intrinsic, for example, SIN, or user-defined $FCN).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

DIMENSIONS (ASAP Command)

Displays a table of maximum array dimensions for the most important program arrays.

Function

Setup Plots and Verify System

Syntax

```
DIMENSIONS
```

Remarks

- The table information summarizes the storage capabilities of a given version of the program.
- Values listed in the form "number-1" usually indicate a limitation due to decimal encoding (either internally or for text output).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

DIRECTIONAL (ASAP Command)

Produces a polar plot of the current angular distribution data file.

Function

Define/Modify Surfunc Entities
Display/Modify Energy Distributions

Syntax

```
DIRECTIONAL [ UNWRAP ] [RADIANCE] [ 'title' ]
```

Option	Description
`UNWRAP`	flag to create a Cartesian plot of intensity versus angle
`RADIANCE`	flux per projected angle
`'title'`	optional title for plot (up to 64 characters)

Remarks

- Produces a polar plot of the angular energy distribution created by a SPOTS DIRECTION or SPREAD DIRECTION command to an angle space **RADIANCE** or radiant intensity distribution. The polar axis of the spherical angle coordinate system is assumed to be "horizontal" (IES type B photometry).
- By default (that is, without the **RADIANCE** option), it converts the radiance (flux per projected solid angle) as a function of direction cosines generated by these commands to an integrated radiant intensity (flux per solid angle) as a function of angle.
- If any distribution value is negative, the data is assumed to be the common logarithm of energy and is handled accordingly.
- Optionally, the polar plot can be **UNWRAP**ped into a Cartesian plot of intensity versus angle.
- The title is delimited by a single quotation mark (**'**).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

DISPLAY (ASAP Command)

Reads previously created distribution data files so that they can be modified.

Function

Display/Modify Energy Distributions

Syntax

```
DISPLAY [  u     [ MODULUS ] [ s ] ]
           name    PHASE
                   AMPLITUDE
                   WAVEFRONT
                   REAL
                   IMAGINARY
                   ENERGY
                   1ST
                   2ND
                   3RD
                   4TH
                   :
         name.BMP  RED
                   GREEN
                   BLUE
                   MONO
```

Option	Description
`u`	logical unit number of distribution data file
`name`	name of distribution data file, or use _ (DISPLAY _) to prompt for file name from Open File dialog box
`MODULUS, PHASE, AMPLITUDE, WAVEFRONT, REAL, IMAGINARY, ENERGY`	field characteristic of interest

Remarks

- Reads a previously created distribution data file into ASAP where it may be modified and/or examined by DISPLAY subcommands.
- The distribution data on logical unit **u** (default=9) or file **name.DIS** is read into memory and can be modified and/or displayed with the following sets of commands:

```
     DISPLAY "*"
or   DISPLAY _
     DISPLAY "*.bmp"
or   DISPLAY _.BMP
```

- If a BMP bitmap file is given, it is first translated to distribution file format (see description of BMP2DIS utility).
- A **DISPLAY** distribution file can be multi-valued or multi-dimensional.

  If the distribution is multi-valued (for example, the six-component complex vector that can be created by a FIELD command), the first value is extracted by default. Otherwise, a specific component given by the third entry can be extracted for processing.

  If the distribution is three-dimensional, **s** is either an absolute (integer) or fractional slice number (the default is the last two-dimensional slice).
- If more than one set of values is generated in this file, additional parameters are in the **DISPLAY** command to pick out this particular set of values. These are 1st, 2nd, and so on. Without these parameters, only the first matrix of values is seen (therefore, 1st is actually not necessary).
- Commands that modify the distribution data file include:

  NORMALIZE, FORM, FFT, AVERAGE, RADIAL, TRANSPOSE, MODIFY, COMBINE, REDUCE, SECTION, ANGLES, VALUES, OFFSET, FOLD , ABEL, and THRESHOLD.
- Commands that display the distribution data file include: TABLE, RANGE, PLOT3D, ISOMETRIC, DMAP, GRAPH, CONTOUR, DIRECTIONAL, ENCLOSED, MESH, and PICTURE.

- Many analysis commands such as SPREAD, SPOTS, RADIANT, OPDMAP produce scalar distribution data files. These files are stored by default in BRO009.DAT (logical unit number 9).
- For examples of how to use the **DISPLAY** command for reading in data, see the MAP command. Other examples are shown below.

1. Header or Output

```
 DISPLAY
 Opening OLD     distribution file   9: BRO009.DAT
 4-record file header:
  Geometrical Ray SPOTS            127
  Z           3.00000      FLUX / UNIT-AREA
  X                  -2.00000        2.00000             127
  Y                  -2.00000        2.00000             127

 Statistics on  127 by  127 data set:
         FLUX / UNIT-AREA  Location   Y                X
  Minimum   0.0000000      1   1   -1.984252      -1.984252
  Maximum   1008.061      64   1   -0.7450581E-08 -1.984252
  Average   0.5624991     64  64   -0.7450581E-08 -0.7450581E-08

  TOTAL FLUX  =  9.0000
```

2. Entering Files into DISPLAY

```
...(other ASAP commands)...
DISPLAY
```

Reads BRO009.DAT into ASAP for further examination and/or analysis.

```
...(other ASAP commands)...
DISPLAY 25
```

Reads FOR025.DAT into ASAP for further examination and/or analysis.

```
...(other ASAP commands)...
DISPLAY JOE
```

Reads JOE.DIS into ASAP for further examination and/or analysis.

```
FIELD AMPLITUDE 0
DISPLAY
...(other DISPLAY commands)...
RETURN
DISPLAY 29 PHASE
...(other DISPLAY commands)...
```

Calculates the amplitude of the field and saves it in BRO009.DAT. (The complex field is saved by default in BRO029.DAT.) The first time DISPLAY is called, it reads the BRO009.DAT file containing the amplitude. The user can then examine the data as desired. When finished, the user issues a RETURN to get back to ASAP, and then calls DISPLAY again, this time specifying that the phase of the complex field is desired. DISPLAY reads the complex field data from BRO029.DAT and extracts the phase information. The user can then examine the phase data as desired.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

HEADER
REPLICATE
Inputting an Arbitrary Data Distribution

DMAP (ASAP Command)

Creates a character map of the current distribution data file.

Function

Display/Modify Energy Distributions

Syntax

```
DMAP [ n ]
```

Option	Description
`n`	number of distinct levels to be simulated with different characters (default 10)

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

DOMACROS (ASAP Command)

Controls when currently defined macros are executed relative to top-level ASAP commands.

Function

Save or Recover System Data and Control Execution

Syntax

```
DOMACROS  FIRST
          LAST
          NEVER
```

Remarks

- Determines whether your macro commands are checked **FIRST**, **LAST**, or **NEVER** relative to the top-level ASAP commands.
- Use the default option, **LAST**; however, you can use **FIRST** if your macro name conflicts with an ASAP command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

DOME (ASAP Command)

Creates a single refractive element.

Function

Define/Modify Lens Entities

Syntax

```
DOME X x t m [ r [ r' ] ]
     Y y
     Z z
```

Option	Description
t	axial thickness
m	media
r	front radius of curvature
r'	back radius of curvature

Remarks

- Creates a single refractive element of axial thickness **t** and media **m** with front and back radii of curvature **r** and **r'**, respectively.
- The side with the shortest radius with be a complete hemisphere while the other will be truncated at the same plane.
- The default for **r** is **t** (solid hemisphere) while the default for **r'** is r-t (concentric shell).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

DOUBLET (ASAP Command)

Creates a cemented doublet lens.

Function

Define/Modify Lens Entities

Syntax

```
DOUBLET X x t h m m' [ f b r ]
        Y y
        Z z
```

Option	Description
`X or Y or Z`	global coordinate axis
`x or y or z`	location on the global coordinate axis
`t`	overall thickness
`h`	aperture height
`m  m'`	media
`f`	focal length
`b`	overall bending factor
`r`	ratio of the focal length of the first element to the second

Remarks

- A cemented doublet lens is a positive and negative lens in contact.
- This lens entity starts out normal to the defined global coordinate axis (X, Y or Z).
- The **r** is the ratio of the focal length of the first element to the second; for an achromatic doublet, **r** is also the ratio of dispersions (the default if **r** is not given).
- The bending parameter is defined as **(c+c')/(c-c')** or, equivalently, as **(r'+r)/r'-r)**; therefore, **b**=0 implies a plano-convex or plano-concave element and others ASAP computes the curvature automatically shared upon the desired bending parameter.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

DRAWING (ASAP Command)

Plots a four-view drawing of the current system geometry.

Function

Setup Plots and Verify System

Syntax

```
DRAWING [ xyz ] [ DIMENSIONS ] [ NORAYS ]
```

Option	Description
`xyz`	changes the default XYZ coordinate ordering
`DIMENSIONS`	add system dimensions to plot
`NORAYS`	plot the intersection points of the rays on each object

Remarks

- Plots a black-and-white, four-view drawing of the current data in the 3-D vector file (BRO030.DAT or *.VCR). The three nonoblique views are aligned and identically scaled, and optionally, can have simple DIMENSIONing added.
- The default views are

Quadrant	View
Top Left	**X** vs. **Z**
Top Right	**Y** vs. **X**
Bottom Left	**Y** vs. **Z**
Bottom Right	**YX** Oblique

- The optional **xyz** argument is a three-letter permutation of the default XYZ coordinate ordering and can be used to obtain a different set of views.
- The **NORAYS** option suppresses the replotting of the rays themselves, but does plot the intersection points of the rays on each object.
- Takes into account the current CONSIDER settings.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

CONSIDER

DUMP (ASAP Command)

Dumps the currently selected rays to a data distribution file.

Function

Analyze Ray/Beam Data

Syntax

```
DUMP [ name ]
       +
```

Option	Description
name	name of the distribution data file created (default LASTDUMP.DIS)
+	append to the previous DUMP, or after END command reinitialization

Remarks

- Dumps the currently selected rays to the binary distribution file **name** (default LASTDUMP.DIS) or appends ("**+**") to the previous DUMP.
- Only the essential information about each ray is written to the file: position coordinates, direction cosines, flux (size and divergence if XMEMORY MIN is not set).
- Use EMITTING DATA to efficiently read back the file.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

XMEMORY MIN
EMITTING DATA

EDGES/CURVES/ENTITIES (ASAP Command)

Signals ASAP that edge definition commands follow.

Function

Define/Modify Curvedge Entities

Syntax

```
EDGES   [ i ]
CURVES
ENTITIES
```

Option	Description
`i`	starting number for defining EDGES/CURVES

Remarks

- The default value for **i** is one more than the highest edge number previously defined. The **i** is initialized to one at start of program execution.
- EDGE, LENS, and SURFACE data currently reside in the same internal storage locations. Therefore, an EDGE number cannot be the same as an already defined LENSES or SURFACE number.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

ELLIPSE (ASAP Command)

Creates an elliptical edge/curve.

Function

Define/Modify Curvedge Entities

Syntax

```
ELLIPSE X x y z [ n a a' ]
        Y y z x
        Z z x y
```

Option	Description
`X, Y or Z`	specifies the axis of symmetry
`x, y or z`	location along coordinate axis
`y z (z x or x y)`	semimajor widths of the ellipse
`n`	number of points (or segments) on the ellipse (default 16)
`a a'`	angular range (in degrees from first semimajor axis) over which ellipse is defined (default is 0 to 360 degrees)

Remarks

- The default number of points along the edges/curves of the ellipse is 16 or the value specified on a previous ELLIPSE command. Use **-n** if you want it to become the default for future ELLIPSE commands.
- The default angular range over which the ellipse is defined is 0 to 360 degrees.
- The semimajor widths are measured to the points, not to the lines connecting the points.
- If **n**, **a** and **a'** are specified, they become the default settings for most future EDGE commands. They are only actual angles when they are multiples of 90 degrees or the aspect ratio of the figure is unity (that is, the ellipse becomes a circle). In the Z axis case, the effective and actual angles are related by the following formula:

  x TAN(actual) = y TAN(effective)
- This edge is made up of coplanar straight line segments, that is, convex polygons whose vertices lie on a particular curve.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

ELLIPSOID (ASAP Command)

Creates an ellipsoidal surface.

Function

Define/Modify Surfunc Entities

Syntax

```
ELLIPSOID u v w [ x y z ] [ -X ]
                            -Y
                            -Z
                             X
                             Y
                             Z
```

Option	Description
`u v w`	semi-lengths along each axis
`x y z`	center of the ellipsoid
`-X, -Y, -Z, X, Y, Z`	Create only this half of the ellipsoid

Reference Point

At center of surface.

Surface Normal

Along positive coordinate direction.

Autolimiting

Yes

Remarks

- Creates a general ellipsoid with semi-lengths along each axis of (**u,v,w**) and center at (**x,y,z**).
- The normal vector points out away from the center.
- Normally a full closed ellipsoid is created (for example, a complete sphere). However, the additional literal entry can be used instead to specify which half of the ellipsoid is wanted (for example, a hemisphere).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

EMITTING (ASAP Command)

Creates a composite single-source emitter from several EMITTING volume commands.

Function

Create Rays/Beams

Syntax

```
EMITTING
w   DISK    ...
    RECT
    ENTITY
    OBJECT
    CONE
    PYRAMID
    BOX
    SPHEROID
    RAYS
    FILAMENT
    HELIX
    IES
    DATA
w' ...
w" ...
   :
```

Option	Description
`w w' w"`	emitting commands flux weighting factors
`DISK, RECT, ENTITY, OBJECT, CONE, PYRAMID, BOX, SPHEROID, RAYS, FILAMENT, HELIX, IES, DATA`	specific emitting commands

Remarks

- Any of the EMITTING types may be used in a composite (single-source) emitter using the EMITTING format.
- The **w**'s are the flux weighting factors for the individual component emitters.
- Only linear transformation commands are allowed between the individual emitter commands.
- Any USERAPOD commands defined before **EMITTING** is applied to all of the emitting types.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
EMITTING BOX/SPHEROID
EMITTING CONE/PYRAMID
EMITTING DATA
EMITTING DISK/RECTANGLE
EMITTING ENTITY or OBJECT
EMITTING FILAMENT
EMITTING HELIX
EMITTING IES
EMITTING RAYS

EMITTING BOX/SPHEROID (ASAP Command)

Creates a ray set uniformly distributed within a box or spheroid of an emitting volume.

Function

Create Rays/Beams

Syntax

```
EMITTING BOX       x y z u v w n [ X ]
         SPHEROID                  Y
                                   Z
```

Option	Description
x y z	center of emitting volume
u v w	semimajor heights along x, y, and z axes
n	total number of rays to be created
X Y Z	flag to vary radiation non-isotropically

Remarks

- Creates a ray set uniformly distributed within either a **BOX** or a **SPHEROID** centered at (**x,y,z**).
- The total radiated flux of the volume is initialized to 1.0 without **X, . . .** and π/4 with **X, . . .**
- By default the radiation pattern is isotropic. If the optional axis flag is entered, the radiation pattern is a donut or torus centered on that axis; that is, the radiation pattern is a function of the sine of the angle from that axis.
- The overall pattern can be apodized in position and/or direction by the current settings of the USERAPOD ANGLES or USERAPOD BOTH commands.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
EMITTING
EMITTING CONE/PYRAMID
EMITTING DATA
EMITTING DISK/RECTANGLE
EMITTING ENTITY or OBJECT
EMITTING FILAMENT
EMITTING HELIX
EMITTING IES
EMITTING RAYS

EMITTING CONE/PYRAMID (ASAP Command)

Creates a rayset uniformly distributed within a cone or pyramid.

Function

Create Rays/Beams

Syntax

```
EMITTING CONE      X x y z x' y' z' n [ ISO ]
         PYRAMID   Y y z x y' z' x'
                   Z z x y z' x' y'
```

Option	Description
`X, Y or Z`	coordinate axis
`x, y or z`	location of the first face on the coordinate axis
`y z (z x or x y)`	semimajor heights in the given directions at the first face
`x', y' or z'`	location of the second face on the coordinate axis
`y' z' (z' x' or x' y')`	semimajor heights in the given directions at the second face
`n`	total number of rays to be created
`ISO`	emits as an isotropic source

Remarks

- Creates a ray grid distributed within either an elliptical **CONE** or a rectangular **PYRAMID** emitting volume.
- The total radiated flux of the volume is initialized to $\pi/4$ without ISO and 1.0 with **ISO**.
- The number of rays per unit length along the axis is held constant so that the ray density is higher at a small end of the volume. This closely simulates the plasma in a compact arc lamp.
- By default the far-field radiation pattern is a donut or torus; that is, the radiation pattern is a function of the sine of the angle from that axis. The **ISO** option yields an isotropic radiation pattern.
- The overall pattern can be apodized in position and/or direction by the current settings of the USERAPOD ANGLES or USERAPOD BOTH commands.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
EMITTING
EMITTING BOX/SPHEROID
EMITTING DATA
EMITTING DISK/RECTANGLE
EMITTING ENTITY or OBJECT
EMITTING FILAMENT
EMITTING HELIX
EMITTING IES
EMITTING RAYS

EMITTING DATA (ASAP Command)

Creates rays from data in given binary distribution file.

Function

Create Rays/Beams

Syntax

```
EMITTING DATA [ file ] [ n ] [ a [ a' ] [ RECT] ] [ ISO ]
        -DATA    u
        +DATA
```

Option	Description
`file`	name of the distribution data file (default extension is DIS)
`u`	FORTRAN unit number (default 9) of distribution data file (default .DAT extension)
`n`	number of rays created
`a   a'`	half-angles (in degrees) for the elliptical or RECTangular cone of emitted radiation centered on the axis
`RECT`	emit as a rectangular cone
`ISO`	specifies isotropic emission

Remarks

- Creates **n** rays at the current **WAVELENGTH** based on the distribution stored in the distribution data file (default **file.dis** or BRO009.DAT). The default for **n** is the total number of samples in the file.
- The EMITTING DATA command currently supports the following distributions:
  1. General list of rays in full (.*dis) or compressed (*.dmp) DUMP command format. If the current **WAVELENGTH** is zero, it is reset to the shortest one stored in the **DUMP** file header.
  2. Two-dimensional planar spatial distributions (for example, SPOTS POS, SPREAD). Optional half-angles in degrees for the elliptical or RECTangular cone of emitted radiation centered on the positive normal axis are defined by **a a'**. The default for **a'** is a, that is, circular or square light cone. The default for **a** is 90 degrees, that is, a full hemisphere. By default, the surface emits directionally in a Lambertian fashion or **ISO**tropically (the sign on the DATA literal determines into which hemisphere). This pattern can be further apodized in direction by the current USERAPOD DIR settings.
  3. General three-dimensional volume distributions that emit isotropically in direction (unless USERAPOD ANGLES is set).
  4. Volumes with cylindrical symmetry (for example, ABEL INVERSE). Normally, the far-field radiation pattern is a donut, that is, it varies as the sine of the angle from the axis. The **ISO** option yields an isotropic radiation pattern.
- Except for the first form, the rays are distributed spatially in a random fashion using the given data as the probability density function.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
Apodization of Ray Distributions
ABEL INVERSE
EMITTING
EMITTING BOX/SPHEROID
EMITTING CONE/PYRAMID
EMITTING DISK/RECTANGLE
EMITTING ENTITY or OBJECT
EMITTING FILAMENT
EMITTING HELIX
EMITTING IES
EMITTING RAYS
Importing CCD Images

EMITTING DISK/RECTANGLE (ASAP Command)

Creates random emitting surfaces.

Function

Create Rays/Beams

Syntax

```
EMITTING DISK X x y z n [ a [ a' ] [ RECT ] ] [ ISO ]
         RECT Y y z x
              Z z x y
             -X
             -Y
             -Z
```

Option	Description
DISK	emitting surface is an elliptical/circular DISK
[RECT]	elliptical/circular DISK emits as a rectangular cone
RECT	emitting surface is a RECTangular plate
X, Y or Z	coordinate axis
x (y or z)	location of plane on the coordinate axis
y z (z x or x y)	semimajor heights in the given directions
n	total number of rays created
a a'	half-angles (in degrees) for the elliptical or RECTangular cone of emitted radiation centered on the axis
ISO	specifies isotropic emission

Remarks

- Creates a ray set that simulates either an elliptical/circular **DISK** or a **RECT**angular plate emitting surface. The radiation pattern is by default Lambertian or may be made to emit **ISO**tropically.
- The total radiated flux of the surface is initialized to unity without **ISO** and 2.0 with **ISO**.
- The default for **a** is 90 degrees, that is, a full hemisphere.
- The default for **a'** is **a** (circular or square light cone).
- The surface radiates in the positive coordinate direction. To radiate into the negative coordinate direction, enter **EMITTING DISK -X**.
- The overall pattern can be apodized in position and/or direction by the current settings of the USERAPOD command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
Apodization of Ray Distributions
EMITTING
EMITTING BOX/SPHEROID
EMITTING CONE/PYRAMID
EMITTING DATA
EMITTING ENTITY or OBJECT
EMITTING FILAMENT
EMITTING HELIX
EMITTING IES
EMITTING RAYS

EMITTING ENTITY or OBJECT (ASAP Command)

Randomly distributes rays over the faceted surface(s) of SURFACE/EDGE entities or SURFACE/EDGE OBJECTS (entity number k or object k (number or name)).

Function

Create Rays/Beams

Syntax

```
EMITTING ENTITY  k [ n ]       [ ISO  ]
         OBJECT -k  -n   [ d ]   NORM
                    +n
```

Option	Description
k	entity or object number or name
n	number of rays (default 100) per facet
d	distance from the actual smooth surface(s)
NORM	directs rays exactly parallel to local surface normal

Remarks

- If the integer entry **n** is negative, its absolute value is the maximum number of uniformly distributed rays that is created. If **n** is preceded by a plus sign, **n** is the number of rays per facet, and the total number of rays created depends on the entity's intrinsic patching, the object's **FACETS** setting, and this value. In the absence of a leading sign, the previous default was plus but this has been changed to minus; that is, the far more common total rays usage. The default for **n** is still 100.
- The resulting ray position lies very nearly on the actual smooth surface(s) of the entity or for an object, a distance **d** (default is a small fraction of largest limits box dimension) from the actual smooth surface(s). If floating point number **d** is entered as exactly **0.** the ray is tagged as being on the actual object instead of object zero.
- If **NORM** is specified, the ray is directed exactly parallel to the local surface normal. Otherwise, the rays directionally radiate in a random Lambertian or **ISO**tropic fashion from the surface(s).
- The sign of **k** controls the direction of the hemispherical emission relative to the local surface normal; that is, whether the emission is, for example, outward or inward from a sphere. Before apodization, the flux emitted by a facet is equal to its area. Therefore, the total flux in the rays approaches the total exact surface area of the object as the number of facets increases. For example, to simulate the thermal emission from an object of a given temperature:

Example

```
UNITS CM
:
WAVELENGTH1=.4 microns
WAVELENGTH2=.7 microns
TEMPERATURE=300 degrees kelvin
EMISSIVITY=.5 normally same as absorption ratio
SIGMA=5.672E-12 stefan boltzmann constant for cm units
FRACTION=FBI(WAVELENGTH2*TEMPERATURE)-FBI(WAVELENGTH1*TEMPERATURE) for watts
FRACTION=LPW(TEMPERATURE) for lumens
EMITTING
  FRACTION*EMISSIVITY*SIGMA*TEMPERATURE^4  OBJECT ... !no ISO/NORM option
  RETURN
:
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
Apodization of Ray Distributions
EMITTING
EMITTING BOX/SPHEROID

EMITTING CONE/PYRAMID
EMITTING DATA
EMITTING DISK/RECTANGLE
EMITTING FILAMENT
EMITTING HELIX
EMITTING IES
EMITTING RAYS

EMITTING FILAMENT (ASAP Command)

Creates a randomly generated ray set that is uniformly distributed along a predefined or arbitrary curve.

Function

Modify Ray/Beam Data

Syntax

```
EMITTING FILAMENT [ i ] [ fcn t' t" ] n [ r ] [ ISO ]
```

Option	Description
i	specifies the curve
fcn	parametric function name
t'	lower parametric bound
t"	upper parametric bound
n	number of rays
r	cross-sectional, semi-thickness of the wire
ISO	specifies isotropic emission

Remarks

- Creates **n** rays randomly but uniformly distributed either along previously defined CURVE **i** or the arbitrary curve,

  X, Y, Z[,W] = fcn(t) t' < t < t"

  where the function name is defined by a previous $FCN command. The last three or four entries created by its expressions are the three coordinates and an additional flux weighting (apodization) value (default 1).
- With a semi-thickness **r** specified, the emitter becomes an emitting volume source. For example, the commands for a helical filament might be:

  ```
  R=1 radius  T=5 turns  L=5 length
   $FCN EFCN x R*COS(6.2832*T*_) y R*SIN(6.2832*T*_) z L*_ w !! 0<_<1
   EMITTING FILAMENT EFCN 0 1 2000 .1
  ```
- The total radiant flux of the volume is initialized to π/4 without **ISO** and 1.0 with **ISO**.
- The overall pattern can be apodized in position and/or direction by the current settings of the USERAPOD ANGLES command.
- Normally, each segment of the filament emits like a Lambertian cylinder. The **ISO** options force each filament point to radiate isotropically.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
EMITTING
EMITTING BOX/SPHEROID
EMITTING CONE/PYRAMID
EMITTING DATA
EMITTING DISK/RECTANGLE
EMITTING ENTITY or OBJECT
EMITTING HELIX
EMITTING IES
EMITTING RAYS

EMITTING HELIX (ASAP Command)

Creates a volume emitter with randomly generated rays, uniformly distributed along a helical curve.

Function

Create Rays/Beams

Syntax (short format)

```
EMITTING HELIX X x x' t h r n [ ISO ]
               Y y y'
               Z z z'
```

Syntax (long format)

```
EMITTING HELIX  X x y z x' y' z' t r n [ ISO ]
                Y y z x y' z' x'
                Z z x y z' x' y'
```

Option	Description
`X, Y or Z`	coordinate axis
`x x', y y' or z z'`	the initial and final coordinate along the axis (x-x')=length
`t`	number of turns (not necessarily integer)
`h`	outside radius of the helix
`r`	radius (semi-thickness) of the wire
`n`	number of rays
`ISO`	specifies isotropic emission
`x y z, y z x, or z x y`	first elliptical shape (second syntax)
`x' y' z', y' z' x', or z' x' y'`	second elliptical shape (second syntax)

Remarks

- Creates **n** random rays that follow a helical (spiral, spring, coil) curve.
- The **t** is the number of turns (not necessarily an integer).
- The **h** is the outside radius, and **r** the radius of the wire.
- Normally, each segment of the helix emits like a Lambertian cylinder.
- The **ISO** option forces each point to radiate isotropically.
- In addition to a true helix, EMITTING HELIX can also be used for some special cases of interest:

  Torus: set the two axial coordinates to be equal and **t** to one.

  Hollow Cylinder: set **t** to a high number and **r** to **|x'-x|/2t** (**X** case).
- The second syntax (long format) is more flexible since it lets the helix vary from one elliptical shape (unprimed coordinates) to another (primed).
- To get a planar spiral, set the two axial coordinates in the second syntax (long format) to be equal.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
EMITTING
EMITTING BOX/SPHEROID
EMITTING CONE/PYRAMID
EMITTING DATA
EMITTING DISK/RECTANGLE
EMITTING ENTITY or OBJECT
EMITTING FILAMENT
EMITTING IES
EMITTING RAYS

EMITTING IES (ASAP Command)

Creates a ray set that emits according to IES data.

Function

Create Rays/Beams

Syntax

```
EMITTING IES name X x y z n [ a ]
                  Y
                  Z
```

Option	Description
`name`	name of IES data file with extension .IES
`X, Y or Z`	coordinate axis to associate with the file's polar axis
`x y z`	emitter coordinates
`n`	number of rays to create
`a`	azimuthal angle (in degrees)

Remarks

- Creates a ray set that emits according to the data in an IESNA Standard Photometric Data file called **name.ies**. EMIT IES and DISPLAY;IESFILE follow LM-63-1995 standard.
- The coordinate axis corresponds to the file's polar axis (vertical if type A or C photometry, horizontal if B).
- An optional azimuthal angle "a" (in degrees) can also be specified to rotate the angular pattern around the polar axis.
- The USERAPOD settings do not affect EMITTING IES since the equivalent information is found in the IES file.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
Apodization of Ray Distributions
EMITTING
EMITTING BOX/SPHEROID
EMITTING CONE/PYRAMID
EMITTING DATA
EMITTING DISK/RECTANGLE
EMITTING ENTITY or OBJECT
EMITTING FILAMENT
EMITTING HELIX
EMITTING RAYS

EMITTING RAYS (ASAP Command)

Creates an arbitrary set of rays.

Function

Create Rays/Beams

Syntax

```
EMITTING RAYS [ n dx dy dz da db dc ]
x y z  a,b,c  f s d
x' [ y' z' a',b',c' f' s' d' ]
  :
```

Option	Description
`n`	number of random rays per input ray
`dx dy dz`	differential spatial volume over which the rays are created
`da db dc`	differential direction cosine over which the rays are created
`x y z`	global spatial coordinate of an arbitrary ray
`a,b,c`	global direction vectors of an arbitrary ray
`f`	flux of an arbitrary ray
`s d`	size and divergence of an arbitrary ray

Remarks

- Creates an arbitrary set of rays (one line of following input per ray) in terms of their positions, directions, fluxes, sizes, and divergences. Any trailing entries that are omitted take their value from the previous ray.
- Optionally, instead of each of these defined rays, **n** rays randomly distributed within the given half-intervals about each defined ray can be created.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING Source Overview
EMITTING
EMITTING BOX/SPHEROID
EMITTING CONE/PYRAMID
EMITTING DATA
EMITTING DISK/RECTANGLE
EMITTING ENTITY or OBJECT
EMITTING FILAMENT
EMITTING HELIX
EMITTING IES

ENCLOSED (ASAP Command)

Calculates an encircled (ensquared) energy using a square array of points.

Function

Display/Modify Energy Distributions

Syntax

```
ENCLOSED [ i j ] [ 'title' ]
           MAX
```

Option	Description
`i j`	pixel center
`MAX`	maximum point
`'title'`	optional title for plot (up to 64 characters)

Remarks

- Sums the data within a square array of points centered about the pixel (**i j**), the **MAX**imum point, or the centroid and plots the percent enclosed as a function of the size of the square.
- The title is delimited by a single quote **'** , as shown.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

END (ASAP Command)

Terminates program execution.

Function

Save or Recover System Data and Control Execution

Syntax

```
END [ OFF ]
      NOW
```

Remarks

- Immediately terminates the current session of ASAP and does either of two things:
  If in batch mode, returns control to the operating system, or
  If in interactive mode, executes a SYSTEM NEW and RESET
- The optional **OFF** entry temporarily turns off the processing of the END command. In other words, any END command following an END OFF command does not terminate the session but simply switches the input to interactive mode. This is useful for executing input files in interactive mode that also work properly in batch mode. To terminate the session under these conditions, enter an **END NOW** or $EXIT command.

Example

```
...(other ASAP commands entered in batch mode)...
END OFF
```

toggles input to interactive mode

```
...(other ASAP commands entered in interactive mode)...
END NOW
```

terminates ASAP execution

```
...(other ASAP commands entered in interactive mode)...
END
```

terminates ASAP execution

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

$EXIT

ENTITIES (ASAP Command)

Defines mixed geometrical entities without using the EDGE, LENS, or SURFACE commands.

Function

Define/Modify Entities or Single Entity Objects

Syntax

```
ENTITIES [  OBJECTS  ]
    lens  …  [ 'name'  ]
    edge
    surf
      entity modifiers (for example, LOCAL, SWEEP)
    [ object modifiers (for example, INTERFACE, BOUNDS) ]
    lens' …
    edge'
    surf'
     :
```

Remarks

- Begin defining mixed geometrical entities without using the **EDGE/CURVE**, **LENS**, or **SURFACE/FUNCTION** commands.
- Optionally, make each subsequently defined entity an object automatically, without using the OBJECT command. This means that only single entity objects can be defined in this manner (although BOUNDS can still be applied if the bounding entities are defined in the normal manner before the **ENTITIES OBJECTS** command).
- All entity modifiers (for example, LOCAL, SWEEP) must precede any object modifiers (for example, INTERFACE, BOUNDS).

Example

The following valid syntax creates objects automatically as entities are given:

```
ENT OBJECT
PLANE ........... 'BASE'   !! These are just examples of some entities
TUBE ........... 'WIRE1'  !! that automatically become objects.
TUBE............ 'WIRE2'
  REDEFINE COLOR 5         !! This is a modifier for the above object, "WIRE2"
RETURN                     !! This ends the ENT OBJECT list.
```

Note

Do not enter ENTITY OBJECT. The correct syntax is: ENT OBJECT.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EDGE/CURVE ENTITIES
LENS ENTITIES
SURFACES/FUNCTIONS ENTITIES

EXPLICIT (ASAP Command)

Converts the current function to explicit form.

Function

Define/Modify Surfunc Entities

Syntax

```
EXPLICIT [ X ] m [ SVD ]
           Y
          -Z
```

Option	Description
X, Y, or -Z	axis entry
m	degree of the explicit polynomial (maximum 20)
SVD	flag to perform a least squares fit operation

Remarks

- Converts the current surface function to an explicit polynomial of degree m (maximum 20). This may be needed if the function is to be used as a DEFORM on an OBJECT.
- The optional axis entry allows order doubling of certain coordinates to enforce symmetry.
- The conversion is actually done by a Cholesky or SVD least squares fit operation on the original surfaces' mesh points. Therefore, the original surface must have a LOCAL box and should probably have PARAMETERIZE set to -Z.
- The RMS and MAXimum sag error is always displayed upon completion.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT
DEFORM
LOCAL
PARAMETERIZE

EXPLODE (ASAP Command)

Explodes lens conicoids into separate surface objects.

Function

Create/Modify Objects

Syntax

```
EXPLODE [ l ] [ macro ]
         +l
         -l
          ALL
         +ALL
         -ALL
```

Option	Description
l	specifies lens to explode into separate surface objects (default is last)
macro	name of the created macro file (default extension *.MAC)
ALL	apply to all lens entities, not just one at a time

Remarks

- Explodes the **l**th (default last) **LENS** or **ALL** lens entities into separate SURFACE-based objects (or CURVE-based objects if the current macro library is the standard **CADEQUIV.LIB**).
- Creates glass inner edge, Mangin outer edge, and mirror back.
- If a sign precedes the entry, additional objects are created that represent mirror backs and the baffle, mounting, or edge surfaces that connect each coaxial lens surface.
- Use "**-**" for direct sloped cones or "**+**" for right cylinders. The resulting input commands either go into the file **macro.MAC** or are immediately executed from an internal buffer.
- The interfaces for the surfaces are preset as follows:

Refractive	Last primarily transmissive COATING PROPERTY or BARE
Reflective	Last primarily reflective COATING PROPERTY or uUnit reflectivity
Back/Edges	Totally absorbing

Example

```
--- EXPLODE +0.1
 -1-  SURFACE; OPTICAL Z 0.0000000    10.00000    0.0000000     ELLIP  3.000000
 -1-   OBJECT; REDEFINE COLOR  3; INTERFACE COATING BARE BK7      VACUUM_AIR
 *** SPLIT automatically set to 1
 -1-  SURFACE; OPTICAL Z  1.250000     -10.00000    0.0000000   ELLIP  3.000000
 -1-   OBJECT; REDEFINE COLOR  4; INTERFACE COATING BARE F2       BK7
 -1-  SURFACE; TUBE Z 0.7893920  2@0.3000000E+01 0.4606080  2@0.3000000E+01 OUTER
 -1-   OBJECT; REDEFINE COLOR  3
 -1-  SURFACE; OPTICAL Z  1.750000     40.00000     0.0000000   ELLIP  3.000000
 -1-   OBJECT; REDEFINE COLOR  4; INTERFACE COATING BARE BK7      F2
 -1-  SURFACE; TUBE Z  1.862659  2@0.3000000E+01 0.7893920  2@0.3000000E+01 OUTER
 -1-   OBJECT; REDEFINE COLOR  4
 -1-  SURFACE; OPTICAL Z  3.000000    -20.00000    0.0000000   ELLIP   3.000000
 -1-   OBJECT; REDEFINE COLOR  3; INTERFACE COATING BARE VACUUM_AIR       BK7
 -1-  SURFACE; TUBE Z  2.773720  2@0.3000000E+01  1.862659  2@0.3000000E+01 OUTER
 -1-   OBJECT; REDEFINE COLOR  3
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

EXTEND (ASAP Command)

Linearly extends one or both ends.

Syntax

```
EXTEND d [ d' ]
```

Option	Description
`d`	start of edge
`d'`	end of edge

Remarks

- Linearly extend the start and end of the current edge a distance **d** and **d'** respectively.
- The default for **d'** is **d**. Enter a zero distance if you only want to extend the other end.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

EXTREMES (ASAP Command)

Lists the minimum and maximum information regarding position, direction, flux, or optical path length in currently selected ray data.

Function

Analyze Ray/Beam Data

Syntax

```
EXTREMES POSITION  [ k ]
         P#
         DIRECTION
         D#
         FLUX
         LENGTH
```

Option	Description
`k`	reference ray number from which extremes are measured
`POSITION`	position of base ray data
`DIRECTION`	direction of base ray data
`P#`	position of the parabasal ray specified by **#**
`D#`	direction of the parabasal ray specified by **#**
`FLUX`	flux of the base ray data
`LENGTH`	optical path length of the base ray data

Remarks

- Determines the extremes in the ray data specified by the given option (POSITION, DIRECTION, and etc.) and lists the corresponding rays. By default the positions or directions of each base ray are used.
- Any particular parabasal ray may be selected by specifying its number **#** (that is, P0 means base ray position, "D1" first parabasal ray direction, etc.).
- The flux, current object, and optical path length of each extreme ray is listed along with all base coordinate data.
- If **k** is less than zero, the average of the ray data is used.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GET
PUT

FACETS (ASAP Command)

Controls the subdivision of mesh patches.

Function

Create/Modify Objects

Syntax

```
FACETS  n [  n' ]
       -n   -n'
       +n   +n'
```

Option	Description
n	number of facets along the edge or curve (intra-edge)
n'	number of facets between edges or curves (inter-edge)

Remarks

- Sets the number of facet subdivisions in each mesh patch direction.
- The **n** specifies the number of facets along the EDGE or CURVE (intra-edge), and **n'** specifies the number of facets between edges or curves (inter-edge). Greater numerical values of **n** and **n'** result in smoother looking plots, but larger plot files.
- If **n** is unsigned, then this is the number that is always used.
- If **n** is positive, it is the maximum number of subdivisions relative to the corresponding number on a PLOT FACETS or VUFACETS command.
- If **n** is negative, its absolute value is the minimum number of subdivisions per patch.
- Controls the density of rays when applied to a GRID OBJECT or EMITTING OBJECT.
- A PLOT MESH shows the mesh boundaries.
- Note that some edges are discrete, and some are continuous; FACETS only controls how the entity is plotted and not how continuous the entity is.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

FCN (ASAP Command)

Specifies a function to wrap around the function of the previous surface.

Function

Create/Modify Objects
Define/Modify Surfunc Entities

Syntax

```
FCN fcn
```

Option	Description
`fcn`	name of the specified macro function

Remarks

- Specifies a macro function (intrinsic, for example, SIN, or user-defined $FCN) to wrap around the previous surface's base function.

  $$f(X,Y,Z) \rightarrow fcn(f(X,Y,Z))$$

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

FFAD (ASAP Command)

Formats currently selected ray data and plots full-field aberration displays (FFAD).

Function

Analyze Ray/Beam Data

Syntax

```
FFAD [ SPOTS [ s ] ] [ REFERENCE w [k] ]
```

Option	Description
SPOTS	plots spot diagram at each field point **s** spot diagram scale factor
s	spot diagram scale factor
REFERENCE w k	smallest spot identifier

Remarks

- Produces a real Full Field Aberration Display, which is a plot of the 3D best-focused RMS spot ellipses for each field position entered on the SOURCE command. The ellipses are scaled up so that the largest one just touches the spot from an adjacent field position. These plots are useful for identifying aberration nodes on the image surfaces of non-centered or perturbed systems.
- FFAD prints the statistics of the spots, including the best-fit planar and curved surface parameters.
- The corresponding unfocused ray **SPOTS** can also be plotted. The **s** (default=0.3) is an additional scale factor used to make room for the spot patterns since they are always larger than the RMS ellipses.
- The **REFERENCE** option specifies the width **w** (in system units) of a square that is drawn around the smallest (or **k**th) spot.
- FFAD also prints out the statistics of the spots including the best-fit planar and curved surface parameters.

Example

```
FFAD SPOTS REFERENCE 0.01 !plot spots with rms ellipses

STATISTICS on best RMS spot sizes from   63 SOURCES:
             X            Y            Z         Overall
Averages 0.21973E-02 0.18023E-02 0.10722E-02 0.30374E-02
Maximums 0.45477E-02 0.32650E-02 0.20836E-02 0.59736E-02
      at     57           57           57           57

Best-fit planar focal surface
                       CENTER 0.4236867E-01 -.5913160E-10 -.2902888E-03
                       NORMAL 0.2065236E-01 -.2884811E-09 0.9997867
Best-fit curved focal surface RADIUS,SHIFT -2.053243     0.8969964E-03
```

- RMS deviations scaled by a factor of 0.65212

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Grids of Sources
SOURCE

FFT (ASAP Command)

Calculates the spatial frequency spectrum of current distribution data.

Function

Display/Modify Energy Distributions

Syntax

```
FFT [ f ] [ AMPLITUDE ] [SIZE [ m ] ] [ 'flabel' ]
      n     PHASE
            MODULUS
            WAVEFRONT
            REAL
            IMAGINARY
            ENERGY
```

Option	Description
f	fraction of grid of transformed data to be used to replace the current distribution file
m	Fourier transform size (see Remarks)
n	number of pixels in the transformed data to be used to replace the current distribution file
AMPLITUDE, PHASE, MODULUS, WAVEFRONT, REAL, IMAGINARY, ENERGY	transform characteristic of interest
flabel	new label for functional data

Remarks

- Replaces the current distribution file with its spatial frequency spectrum. It does this by applying a Fast Fourier Transform to an m-by-m block of data points, where **m** is the current Fourier transform size (2 raised to the **m**th power, or as set by the FTSIZE command) minus one. See DIMENSION output.
- If the original data area is smaller than **m**-by-**m**, the data is centered and any excess is zero filled.
- If the original data area is larger, you must make sure that most of the energy lies in the first **m**-by-**m** block. ASAP then takes the selected part (default MODULUS) of the center **n**-by-**n** or **f*****m**-by-**b*****m** area of the FFT as the current data set.
- If **f** is an integer **n** greater than one, it is the actual number of pixels to use. ASAP calculates the default for **n** to maintain the original data size.
- If **f** or **n**, are a negative number, the inverse FFT operation is performed.
- Applying the FFT command to the diffraction image of a point object generated by a SPREAD or OPDMAP command produces the classical MTF (Modulation Transfer Function) of the system.
- The spacing **S'** between sample points after the FFT is related to the original spacing **S** by:

  $$S' = \left(\left(M+1\right) * S\right)^{-1}$$

  where the maximum pixel value determines the constant in the above expression.

  Use the **flabel** option to relabel the functional data.
- If the Fourier transformed field **U(p,q)** is written in the form:

  $$U\left(p,q\right) = A\left(p,q\right) + iB\left(p,q\right)$$

  where **A(p,q)** is the real part and **B(p,q)** is the imaginary part of the field, then the components of the complex field that can be displayed include those in the following table.

Option	Operation	Physical Significance
AMPLITUDE	sign(real(U)) | U |	signed modulus of field
PHASE	arctan(B / A)	phase in radians
MODULUS	| U |	modulus of the field
WAVEFRONT	arctan(B / A) / 2 p	wavefront in waves
REAL	real(U) or A	real part of field
IMAGINARY	imaginary(U) or B	imaginary part of field
ENERGY	U U*	modulus squared of field (energy

density)

- The phase transfer function (PTF) may be obtained by extracting the phase or wavefront from the FFT operation. The wavefront is usually better choice because it attempts to remove the **2π** phase steps from the wavefront.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
OPDMAP
SPREAD

FIELD (ASAP Command)

Calculates the exact complex field distribution.

Function

Calculate Diffraction/Propagation Effects

Syntax

```
FIELD... MODULUS ...  [  ADD      ]  [ DELTA  [ t ] ]  [ CONTOUR k         ]
         PHASE                        MULTIPLY                        c c' [ c" ... ]
         AMPLITUDE    COUPLE  [ r v [a] ]
         WAVEFRONT
         REAL
         IMAGINARY
         ENERGY
         NONE
```

Option	Description
`AMPLITUDE, PHASE, MODULUS, WAVEFRONT, REAL, IMAGINARY, ENERGY, NONE`	specify the field characteristic of interest
`ADD`	flag for adding the calculated field to the existing optical field file (BRO029.DAT)
`COUPLE`	flag for coupling the calculated field to the existing optical field file (BRO029.DAT)
`DELTA t`	constant phase shift in units of cycle at the current WAVELENGTH
`CONTOUR`	generates an additional contour plot
`k`	use k **equally spaced contours; if** k is less than zero, a "gray scale" plot is produced, in place of the line contour plot.
`c c' c" ...`	**specific contour levels to plot**

Remarks

- If the optical field **U(p,q)** is written in the form:

$$U(p,q) \quad A(p,q)EXP \frac{i2\ W(p,q)}{}$$

where **A(p,q)** is the amplitude function and **W(p,q)** is the wavefront function, then the components of the complex field that can be displayed as shown:

Option	Operation	Physical Significance
`AMPLITUDE`	sign(real(U)) | U |	signed modulus of field
`PHASE`	2 π W(p,q) / λ	phase in radians
`MODULUS`	| U |	modulus of the field
`WAVEFRONT`	W(p,q) / λ	wavefront in waves
`REAL`	real(U)	real part of field
`IMAGINARY`	imaginary(U)	imaginary part of field
`ENERGY`	U U*	modulus squared of field (energy density)
`NONE`	--	none

- The original complex vector field is written to unit 29 (file BRO029.DAT). If the **ADD** option is present, the calculated complex field is added to the previously stored field in the BRO029.DAT file. If the **COUPLE** option is present, the calculated field is coupled to the field previously stored in the file.
- Alternatively, you can directly **COUPLE** to the fundamental mode field of a circular fiber of core radius **r**, normalized frequency

v, and optional gradient index power **a** (for example, 2 for parabolic, absent for step index). If **MULTIPLY** is specified, the result is the product of the two fields. In all three cases, the two fields must be calculated under the same conditions, that is, **WAVELENGTH**, **PIXEL**, and **WINDOW** settings.

- A constant phase shift **t** (in cycles at the current WAVELENGTH) can be added to each beam. This is equivalent to calculating the field at a time other than t=0. If this **DELTA** option is present and a vector field is being calculated (that is, **FRESNEL BOTH** is set), ASAP produces a plot of polarization ellipses, or if **t** is given, arrows representing the relative magnitude and direction of the instantaneous electric field at each sample point.
- The **CONTOUR** option generates an additional contour plot with fractional contours **c c' ...** relative to the full range of the function being plotted. Alternatively, **k** equally spaced contours can be specified. If k is less than zero, a grey scale plot is produced instead of the line contour plot. If **NONE** is not selected, the scalar distribution is written to logical unit 9 and thus can be manipulated, replotted, and named by the **DISPLAY** commands.

Example

```
FIELD ENERGY -1

 Distribution of data within:
  Across or Vertical: Z = -2.50000     to  2.50000      ( 5.00000    )
  Down or Horizontal: Y = -2.50000     to  2.50000      ( 5.00000    )

     Sample plane at: X = -1.00000
 MINIMUM (m) =   0.000000     MAXIMUM (M) =    1.01502
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

FIELDBPM
FIELDSUM
Coherent/Incoherent Analysis
VIOLATION (ASAP Command)
SPREAD and FIELD Differences
SPREAD/FIELD Calculations (Point Sources)
IRRADIANCE
SPREAD
WAVELENGTH

FIELDBPM (ASAP Command)

Field propagation through small geometries and inhomogeneous media.

Function

Calculate Diffraction/Propagation Effects

```
FIELDBPM u   component  [ d     [ n ] ]   [ BC [ i [ i' ] ] ] ...options
        n.e               f  f'               i j    i' j'
```

Option	Description
u	complex distribution file number
n.e	complex distribution file name
f f'	depth coordinate information used to determine the grid
i j, i' j'	one or both directions of the computational window
d	distance in the depth direction
BC	boundary condition

Remarks

- **FIELDBPM** is the second version of the FIELD command that does not use any of the current ray/beam data.
- Takes the field stored in the complex distribution file number **u** or name (with extension) **n.e**, and propagates it a distances **d** (or from **f** to **f'**) in the depth direction by directly solving the partial differential harmonic wave equation (the scalar or semi-vectorial Helmholtz equation).
- The **n** is the number of equally spaced, intermediate propagation locations, at which the field distributions are written to the default binary files (BRO009.DAT and BRO029.DAT). The default is zero; that is, only the final location.
- A unique Finite Difference Beam Propagation Method (FD-BPM) technique is used, which is not only fast and accurate, but automatically determines the best reference refractive index and axial step size for the given media and lateral sampling.
- The field is assumed to start entirely within the current **IMMERSE** medium, but will then interact with any other media it encounters. All media can be complex (**ABSORB**ing), inhomogeneous (**GRIN**), and birefringent (**CRYSTAL** if the optical axis is parallel to one of the global coordinate axes).
- Any object geometry (including surfaces imported from CAD) can act as interfaces between the media, but the actual reflection and transmission coefficients of these objects are ignored.
- Whenever possible, order the media on each object's INTERFACE command so that the first media is before the surface and the second is after (relative to the positive X, Y, or Z depth axis direction). Also, the two INTERFACE media entries must be separated by only a single comma **,** to indicate that there is a preferred order to them. This separator permits nearly all "MEDIA mismatch" errors to be fixed quickly and automatically.
- To prevent spurious "numerical" reflections at the edge of the computational window, a physical (BC option absent) or numerical (BC option alone) absorbing region is implemented by default that involves the outer 1/6 pixels at each edge of the window (that is, a total of 1/3 of the pixels in each lateral direction). Otherwise, the following explicit boundary conditions can be specified with the BC option for one (**i**) or both (**ij**) directions of the computational window (unprimed entries first end of window, primed second):

BC	Description
<-1	Number of pixels in physically absorbing band
-1	Periodic
0	Zero field
1	Totally reflective (zero derivative)
>1	Number of pixels in each numerically absorbing band

- The normally integer **i**s and **j**s can also be entered as floating point fractions of the number of samples in the respective direction.
- The specific BPM algorithm used is capable of operating outside the normal "paraxial" region (small departures from the factored plane wave) because it expands the lateral differential operator into higher order terms. The order number is controlled by the current **ACCURACY** setting and the possibilities are listed (from fastest to slowest) below:

ACCURACY setting	Operator Order	Max Angle from Axis	Comments
LOW	1	10	Fresnel/paraxial/weakly-guided approximation
MEDIUM	2	20	
HIGH	3	40	Wide-angle and/or large refractive index variations

- Besides the obvious current restriction to scalar (non-vector) fields, there is also a basic limitation inherent in the BPM method: it tracks only the "forward" propagating component of the field, and thus ignores back-reflected waves and any flux lost to them. This means that it does a good job of predicting the relative shape and phase of the main field, but not necessarily the absolute magnitude.
- The advantage of the BPM method is that it relatively efficient at propagating wavefields through three-dimensional volumes, versus a more rigorous Time-Domain (TD) method that is essentially four-dimensional and thus orders of magnitude slower for the same size problem.
- The accuracy of the calculation is most directly affected by the lateral sampling used; that is, the number of PIXELS for a given size WINDOW. Many problems require sub-wavelength sampling, which means that we are practically limited to small volumes. A good rule of thumb is to use a sample spacing of one-third of the shortest wavelength (the vacuum wavelength divided by the highest index in the volume). Some problems may allow coarser sampling, while others require finer sampling. The best indicator of accuracy is the two numbers displayed under the "Relative Flux" heading during the propagation. For a non-absorbing (actual and/or numerical) problem, the closer these numbers stay to unity during the entire propagation, the more accurate the final result. To reduce their fluctuation, decrease the lateral sample size (for example, increase the number of PIXELS). Be careful though, the number of longitudinal steps required may increase dramatically along with run times.
- As with virtually all types of numerical simulations, there is a tradeoff between speed and accuracy. BRO recommends that the optimum sampling for a given problem be found first with a fast two-dimensional version (negligible width in second WINDOW direction) before proceeding to the full three-dimensional calculation. For three-dimensional problems with cylindrical symmetry, a fast, radially symmetric calculation can be done by using the following command sequence:

  ```
  WINDOW Y 0 rmax X -tiny tiny
  FIELD ENERGY ... !create 1D starting field
  AXIS Z
  FIELD 29 ENERGY dist ... !propagate a distance
  ```

  where, in this example, Z is the symmetry and propagation axis.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

FIELD

FIELDSUM (ASAP Command)

Sums coherent beams.

Function

Calculate Diffraction/Propagation Effects

Syntax

```
FIELDSUM component [ f [ f' n ] ] ...options
[ l l' m m' z x' y' z' x" y" z" ]
```

Option	Description
`f  f'`	depth coordinate information used to determine the grid
`n`	number of steps
`x, y, z`	origin position vector
`x'x", y'y", z'z"`	two direction increment vectors

Remarks

- **FIELDSUM** is a more general implementation of the (Gaussian) beam summation technique than the SPREAD NORMAL method. It sums ALL coherent beam contributions from the current ray set, including multiple wavelengths.
- Any component of the resulting complex scalar or vector field can be calculated and displayed on an orthogonal or arbitrarily oriented skew planar grid.
- Normally, the grid is determined by the last WINDOW and PIXELS commands, and the third depth coordinate value **f** (or a range from **f** to **f'** in **n** steps).
- If **f** is omitted and the next command starts with a number, the grid information is read from that command. This skew planar grid is specified by an origin position vector (**x,y,z**) and two direction increment vectors, **(x′,y′,z′**) and **(x",y",z"**). The integer ranges of the two grid coordinates are given by **l** to **l′** for the first direction and **m** to **m’** for the second. In other words, the actual grid coordinates in global coordinates are given by:

  $$x(i,j) = x + ix' + jx'' \quad i=l,l' \quad j=m,m'$$
  $$y(i,j) = y + iy' + jy''$$
  $$z(i,j) = z + iz' + jz''$$

  The total number of grid points must not exceed the following:

Version	Scalar	Vector (FRESNEL BOTH)
Basic	736x736/2 = 270848 ~ 520x520	736x736/6 = 90282 ~ 300x300
Pro	2896x2896/2 = 4193408 ~ 2047x2047	2896x2896/6 = 1397802 ~ 1182x1182

- The printed map of the desired field component will be displayed if the **GRAPHICS** unit is active (see $IO command).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

FIELD
FIELDBPBM

FITTED (ASAP Command)

Creates a surface specified by curve fitting an arbitrary set of points.

Function

Define/Modify Surfunc Entities

Syntax (first):

```
FITTED [ X ] [ k ] [ t ] [ SVD [m]] [ FIXTERM [n] ] [ EDGES -n      ] [ VECTOR c ]
         Y                                                   i i' i"
         Z
        -X
        -Y
        -Z
         EXPLICIT
[ coord coord' coord" ]
x y z [ ABS ] x' y' z' x" y" z" ...
        REL
x y z x' y' z' x" y" z" ...
   :
```

Syntax (second):

```
FITTED PLANE       [ EDGES -n       ] [ VECTOR c ] ]
       PARABOLOID           i i' i"
       SPHERE
       ELLIPSOID
[ coord coord' coord" ]
x y z [ ABS ] x' y' z' x" y" z" ...
           REL
x y z x' y' z' x" y" z" ...
   :
```

Option	Description
X, Y, Z, -X, -Y or -Z	axis for order doubled polynomial
k	degree of surface fit (default 2)
t	coefficient threshold (default 1.E-5)
SVD m	singular value decomposition option
FIXTERM n	number of the term to be normalized to 1
EDGES -n	uses points on last n EDGES for fit
EDGES i i' i"	uses points on EDGES i through i' incremented by i" for fit
VECTOR c	puts each data point in the current 3D file as a dot of color **c** (default 1) for later plotting
EXPLICIT	explicit function option
coord coord' coord"	specifies data coordinate order
x y z x' y' z' x" y" z" ...	points for fit
ABS REL	specifies how data is referenced

Reference Point

See Remarks

Surface Normal

Along negative coordinate axis

Autolimiting

See Syntax and Options

Remarks

- Fits a surface of degree **k** (default 2) in a least squares sense to the set of points given on successive commands, the points on the last **n EDGES**, or **EDGES i** through **i'** incremented **i''**.
- Alternatively, the given specific surface type (second syntax) is fitted in a least squares sense to the set of points given on successive commands, the points on the last **n EDGES**, or **EDGES i** through **i'** incremented **i''**.
- The literal entries (**X**, **Y**, **Z**, **-X**, **-Y**, **-Z**) are the same as the ones on the GENERAL command and specify some required symmetry.
- The **t** is an optional threshold between 1.E-8 and 1.E-2 (default 1.E-5) below which the relative contribution of a surface coefficient is assumed to be negligible so that it is reset to zero. If the highest order surface coefficients are all zero, the degree of the surface is reduced accordingly.
- The program saves the maximum extents in each coordinate direction and uses these as the default LOCAL modifier.
- The rarely needed **FIXTERM** option specifies the number of the term to be fixed or normalized to 1. The default is the largest component of the normal to the surface through the first point (if the point lies on the surface) or zero (constant term).
- By default, the **EXPLICIT** solution is a Cholesky decomposition of the normal matrix that can accept any number of data points. One can elect to do a more robust SVD (Singular Value Decomposition) solution, but the maximum number of data points it accepts is limited (see output from the DIMENSION command). If **m** is given, then only every **m**th point is used, which may be necessary to get below this limit.
- When entering points directly, the first point always becomes the reference point of the surfaces and, by default, ASAP assumes that the surface also passes exactly through this point, that is, the constant coefficient of the surface function is set to zero. However, if the literal option is placed after the first three entries, this reference point is not on the surface (constant coefficient nonzero), and all other points entered are either measured **ABS**olutely or **REL**atively from it.
- By default the ordering of a triplet of numbers describing a point is (**x,y,z**). You may specify a different order by listing the coordinate order. For example, to specify data as (**z,x,y**), enter (**Z,X,Y**).
- The VECTOR options puts each data point in the current 3D file as a dot of color **c** (default 1) for later plotting by a REPLOT, DRAW or $VIEW command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GENERAL

DIMENSION

DRAW

LSQFIT for fit control and error monitoring.

REPLOT

$VIEW

FLUX (ASAP Command)

Modifies ray fluxes of currently selected ray data.

Function

Modify Ray/Beam Data

Syntax

```
FLUX  a      [ b  [  i   j ] ]
      TOTAL          SOU m
```

Option	Description
a b	scale factors for ray fluxes
i j	inclusive ray numbers
TOTAL	change the fluxes to produce a total flux **b**
SOU m	source number

Remarks

- Scales the fluxes for the currently SELECTed ray set (or the rays specified by the optional **i** and **j**) according to the following equation:

  new_flux = a + b*old_flux
- The default for **b** is zero.
- Alternatively, the **TOTAL** option specifies that the fluxes be changed to produce a total flux **b**.
- ASAP ignores rays with fluxes less than or equal to the CUTOFF value in any calculation.
- The presence of a BEAMS command changes the way ASAP assigns flux values to the rays. Given a **GRID XXX R r p q s t m n** command where **XXX** is either elliptic or rectangular, the flux per ray is computed as follows:

  BEAMS INCOHERENT GEOMETRIC: (q - p)(t - s) / (mn) = flux / ray

  BEAMS COHERENT DIFFRACT: (q - p)(t - s) / (2mn w^2) = flux / ray
- The GRID POLAR command is too general for this simple calculation to apply since ASAP has to perform all kinds of manipulations (including a least squares fit) to get the fluxes correct.
- By default all rays entered through the RAYSET command have unity flux assigned to them.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

BEAMS (ASAP Command)
CUTOFF
GRID POLAR
RAYSET
SELECT
Spectral Weighting

FMAP (ASAP Command)

Creates a contour slice map of a 3D surface.

Function

Define/Modify Surfunc Entities

Syntax

```
FMAP f [ f' n ]
       [ r x y z ]
```

Option	Description
`f f'`	cutting plane depth coordinate (or range)
`n`	number of steps for cutting plane depth coordinate range
`r`	radius of sphere from which the difference between surface and sphere is calculated
`x y z`	center of sphere (global coordinate)

Remarks

- Can follow any surface definition commands and generates a contour map of one (or more) cuts of the current 3D surface function. The paper coordinates and ranges are given by the last WINDOW command. The resolution is given by the PIXEL command.
- The depth coordinate's cutting plane value is **f** (or **f** to **f'** in **n** steps). The map indicates the different branches, sheets, and regions of the function and its sign so that any problems with signs of normals and unexpected intersections can be graphically verified.
- Optionally, the difference between the surface and a sphere of radius **r** centered at (**x y z**) can be displayed.
- A distribution file of the surface data is also created on logical unit 9 (BRO009.DAT) for later processing by the DISPLAY commands.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
PIXELS
WINDOW
REPEAT

FOCUS (ASAP Command)

Determines the best 3D focal point of currently selected ray data.

Function

Analyze Ray/Beam Data

Syntax

```
FOCUS [ MODE  [ m ] ]  [ MOVE [ r ]  ]
```

Option	Description
MODE m	specifies the method of determining the best 3D focal point for the current ray bundle
MOVE	transfers currently selected ray data to a plane that passes through the best 3D focal point
MOVE r	transfers currently selected ray data to a reference sphere of radius r centered on the best 3D focal point

Remarks

- Determines the best 3D focal point and RMS deviations of the ray data from this point for the current ray bundle using one of the following methods (listed from fastest to slowest):

MODE	METHOD
m<0	Centroid of individual ray foci
m=0 (default)	Least squares closest approach of base rays
m>0	Above plus first "m" parabasal rays

- In general, use CONSIDER to isolate the object of interest before using **FOCUS**. If you do not, **FOCUS** uses ray data from all available objects and gives you the wrong answer.
- If there are several sources, you may need to use SELECT to isolate a particular source to avoid the problem described previously.
- The **MOVE** option calculates best focus and then moves the current ray bundle to the plane that passes through this focal point.
- The SPOTS command can then be used to generate a spot diagram at this best focus.
- **MOVE r** is used to move the ray data to a reference sphere centered on the best focal point and adjust the optical path lengths so that geometric wavefront analyses may be performed. The peak-to-valley and RMS optical path differences are also printed out.

Example

```
--- SELECT ONLY SOURCE 1
    160 ray flags changed
     80 rays now selected

 --- STATS POS

 Current Statistics for Object   0 -
  Total Flux = 13.1172     from     80 rays ( 33.33%)
                      X            Y            Z
        Centroid: 0.185707E-07 -.461274     0.000000
   RMS Deviation: 0.951377     0.961321     0.000000
  Maximum Spread: -1.80000     -1.33873     0.000000
               to  1.80000      2.26127     0.000000

 --- FOCUS
```

```
Least Squares Focus Calculation for     80 Rays:
                    X             Y             Z
 Centroid Point 0.0000000     -3.000000      3.000000
 RMS Deviations 0.3033E-07    0.7508E-07    0.8019E-07
 Mean Direction  0.0000000    -0.6244647     0.7810531

Total Flux =  13.12     RMS Blur Diameter = 0.1832593E-06
Maximum Ray Angle from Mean =  25.1041 degrees, F/ 1.067

--- FOCUS MOVE

Least Squares Focus Calculation for     80 Rays:
                    X             Y             Z
 Centroid Point 0.0000000     -3.000000      3.000000
 RMS Deviations 0.3033E-07    0.7508E-07    0.8019E-07
 Mean Direction  0.0000000    -0.6244647     0.7810531

Total Flux =  13.12     RMS Blur Diameter = 0.1832593E-06
Maximum Ray Angle from Mean =  25.1041 degrees, F/ 1.067

--- STATS POS

Current Statistics for Object   0 -
 Total Flux = 13.1172     from     80 rays ( 33.33%)
                     X            Y            Z
       Centroid: 0.462397E-17 -3.00000      3.00000
   RMS Deviation: 0.547368E-16 0.198808E-15 0.146429E-15
 Maximum Spread: -.146438E-15 -.888178E-15 -.444089E-15
             to 0.285834E-15 0.888178E-15 0.444089E-15

--- FOCUS MOVE -10

Least Squares Focus Calculation for     80 Rays:
                    X             Y             Z
 Centroid Point 0.3239507E-17 -3.000000      3.000000
 RMS Deviations 0.5079E-16    0.8324E-16    0.9947E-16
 Mean Direction  0.0000000    -0.6244647     0.7810531

Total Flux =  13.12     RMS Blur Diameter = 0.2206772E-15
Maximum Ray Angle from Mean =  25.1041 degrees, F/ 1.067

Flux weighted statistics:
P-V Optical Path Difference =   0.280714E-11 waves
   Wavefront Variance (RMS) =   0.412710E-12 waves
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Best Focus Position
CONSIDER
GET
OPDMAP
PUT
SELECT
SPOTS

FOLD (ASAP Command)

Averages the current distribution data about the pixel number of the specified coordinates.

Function

Display/Modify Energy Distributions

Syntax

```
FOLD [ FIRST ]   [ n ]
       SECOND
       BOTH
```

Option	Description
FIRST	average the data about the center of the first coordinate
SECOND	average the data about the center of the second coordinate
BOTH	average the data about the center of both coordinates (default)
n	absolute or fractional pixel number

Remarks

- Averages the current distribution data about the given absolute or fractional pixel number **n** (default is center) of the **FIRST** (vertical), **SECOND** (horizontal) or **BOTH** (default) coordinates.
- Useful for distributions that should be perfectly symmetric about one or both centers but are not due to sampling and/or statistical errors.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

FORM (ASAP Command)

Controls the form of the current distribution data.

Function

Display/Modify Energy Distributions

Syntax

```
FORM [ f ] [ 'flabel' ]
```

Option	Description
`f`	controls the form of the data to be plotted (default=1)
`flabel`	new label for functional data

Remarks

- Either raises the data in the current distribution data file to a power or puts it into log (base 10) space.
- The default value of **f** is one, which does nothing to the data.
- If **f** is greater than zero, the data is raised to that power (that is, **FORM .5** takes the square root of the distribution data) before plotting.
- If less than zero, the common logarithm of the data is plotted with **f** setting a lower cutoff relative to the data maximum.
- If the distribution has any negative values in it, a bias is added to make all values greater than or equal to zero before the requested operation takes place.
- Use 'flabel' to relabel the functional data.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

FRESNEL (ASAP Command)

Adjusts the reflection/transmission coefficients specified on an interface using Fresnel's formulae during a ray trace.

Function

Create/Modify Objects
Setup Trace

Syntax

```
FRESNEL [ TIR ]
        [ OFF ]
          AVE
          S
          P
          BOTH
          k
```

Option	Description
TIR	no variation with angle of incidence up to the critical angle, then TIR
OFF	no variation with angle of incidence
AVE	incoherent average of polarizations (default)
S	S polarization coefficient only
P	P polarization coefficient only
BOTH	both separately
k	controls which Fresnel coefficient to use in the refraction calculation

Remarks

- **FRESNEL** is typically used with a bare coating, specified on the INTERFACE command, and a split level of one or higher. See Combinations of INTERFACE and FRESNEL Settings for other settings.
- **FRESNEL** can be applied globally or on an OBJECT-by-OBJECT basis, as a global modifier. One of the polarization components of a ray will be eliminated at any objects that do not have FRESNEL BOTH set.
- Signals ASAP to take into account the variation of the reflection and transmission coefficients with incidence angle at the object's interface and adjust the flux accordingly. It does this by evaluating the media on either side of the interface and then applying the Fresnel formulae.

CAUTION

- It is not possible to globally apply **FRESNEL BOTH** and then locally (to a specific surface, for example) apply **FRESNEL OFF**. The **FRESNEL OFF** will be ignored.
- **FRESNEL** must be used whenever you want to recalculate the reflection/transmission coefficients for each ray during a ray trace.
- The **k** parameter, or equivalent literal entry, is used to control which Fresnel coefficient to use in the refraction calculation, as shown by the following:

Literal	Integer k	Definition
TIR	-2	Allow TIR rays but no angular flux variation for others
OFF	-1	Use normal incidence properties for all rays (default)
AVE	0	Incoherent average of polarizations
S	1	S polarization coefficient only
P	2	P polarization only
BOTH	3	Both polarizations separately

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Refraction/Reflection Controls

FTSIZE (ASAP Command)

Sets the Fourier Transform size to the default maximum.

Function

Save or Recover System Data and Control Execution

Syntax

```
FTSIZE [ m ]
```

Remarks

- Sets the Fourier Transform size used by future **DECOMPOSE DIR, OPDMAP PSF,** and **FFT** commands to the default maximum recommended (as shown by DIMENSIONS command) or 2 raised to the **m**th power.
- The default size at startup (or after a **RESET** or **END**) is 9; that is, 512 samples.
- The absolute maximum allowed is 16 (65536 samples).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

GAUSSIAN (ASAP Command)

Creates a beam set simulating a coherent astigmatic Gaussian mode.

Function

Create Rays/Beams

Syntax (Long Form):

```
GAUSSIAN X x x' x"  n' n"  m' m"  a' a" [ DEGREES ]
         Y y y' y"                        RADIANS
         Z z z' z"
```

Syntax (Short Form):

```
GAUSSIAN X x x' n a [ DEGREES ]
         Y y y'       RADIANS
         Z z z'
```

Option	Description
`X, Y, or Z`	axis of symmetry
`x, y, or z`	rays/beams starting plane
`x' x", y' y", or z' z"`	first and second waist locations
`n' n"`	number of rays/beams along waist directions
`m' m"`	mode numbers along waist directions
`a' a"`	waist semiwidths (or angular semidivergences)
`DEGREES or RADIANS`	units for **a' a"** (angular semidivergences)

Remarks

- Creates an astigmatic Hermite-Gaussian field distribution propagating in the specified direction and starting at the given plane.
- The **GAUSSIAN** command takes the place of the **GRID** and SOURCE commands since this information is specified directly on the **GAUSSIAN** command.
- The WAVELENGTH must be previously defined, and the field created has total flux identical to a unit peak-irradiance fundamental mode.
- **GAUSSIAN** has two modes of operation:
  1. Specify the general astigmatic properties of the **GAUSSIAN** beam in terms of waist locations and semiwidths (that is, a HeNe laser).
  2. Specify the general astigmatic properties of the **GAUSSIAN** beam in terms of the waist locations and angular semidivergences (that is, a semiconductor-like laser).
- The waist location is determined by the right-hand rule. For example, **x'** is the waist location (or line focus) along the **y** direction, and **x"** is the waist location (or line focus) along the **z** direction.
- When you specify angular semidivergences, you should not have a coincident ray starting plane and waist locations. Start the rays at a distance away from the waist locations and use the MOVE command to move the rays to the actual waist locations.
- The current CLIP POSITION and flux CUTOFF/HALT settings may be used to truncate the rays created with the **GAUSSIAN** command.

When the beam is in the fundamental mode, circularly symmetric, and not astigmatic, then the short form may be used.

Example

```
--- CF=0.8862269
 --- UNITS MM
 --- RAYS 0
       0 rays defined

 --- WAVELENGTH 0.6328 UM
 --- PARABASAL 4; WIDTH 1.6
 --- GAUSSIAN Z 0 0 0 21 21 0 0 0.4*CF 0.6*CF
 OVERALL Gaussian semi-widths   0.354491       0.531736
```

```
         wavefront curvatures   0.000000       0.000000
  Individual beam semi-widths   0.716071E-01   0.107411
                            (    113.159        169.739     waves)
         293 rays created in a GAUSSIAN distribution for a total         293
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW
MOVE

GENERAL (ASAP Command)

Creates a general surface defined by polynomial coefficients.

Function

Define/Modify Surfunc Entities

Syntax

```
GENERAL      [ X ] [ x y z ] [ c c' c" ... ]
COEFFICIENTS   Y
               Z
              -X
              -Y
              -Z
               EXPLICIT
XiYjZk  c    [ c' c" ... ]
   :
```

Option	Description
`X, Y, Z, -X, -Y, or -Z`	axis for order-doubled polynomial
`x y z`	reference point
`c c' c" ...`	polynomial coefficients
`COEFFICIENTS`	synonym for GENERAL
`EXPLICIT`	explicit function option
`XiYjZk`	coefficient term representation

Reference Point

As given

Surface Normal

As defined

Autolimiting

No

Remarks

- For entering the polynomial coefficients directly into ASAP. In effect, all other **SURFACE** definition commands are shorthand for GENERAL, since ASAP saves surface data only in this form. The surface may be written as a 10th-order polynomial in the three Cartesian coordinates as shown in the following equation.

$$\mathrm{f(X,Y,Z)} = \sum_{i=0}^{N}\sum_{j=0}^{i}\sum_{k=0}^{j} c_{ijk}\,(X-x)^{i-j}\left(Y-y\right)^{j-k}(Z-z)^{k} = 0$$

- The **c** entries are the coefficients of the 286-term function defining polynomial. These polynomials are ordered as shown (numbers in Terms column are powers):

Order	Terms
0 Constant	C
1 Linear	X Y Z
2 Quadric	X2 XY XZ Y2 YZ Z2
3 Cubic	X3 X2Y X2Z XY2 XYZ XZ2 Y3 Y2Z YZ2 Z3
4 Quartic	X4 X3Y X3Z X2Y2 X2YZ X2Z2 XY3 XY2Z XYZ2 XZ3 Y4 Y3Z Y2Z2 YZ3 Z4
5 th-order	X5 X4Y X4Z X3Y2 X3YZ X3Z2 X2Y3 X2Y2Z X2YZ2 X2Z3 XY4 XY3Z XY2Z2 XYZ3 XZ4 Y5 Y4Z Y3Z2 Y2Z3 YZ4 Z5

6 th-order	X6 X5Y X5Z X4Y2 X4YZ X4Z2 X3Y3 X3Y2Z X3YZ2 X3Z3 X2Y4 X2Y3Z X2Y2Z2 X2YZ3 X2Z4 XY5 XY4Z XY3Z2 XY2Z3 XYZ4 XZ5 Y6 Y5Z Y4Z2 Y3Z3 Y2Z4 YZ5 Z6
7 th-order	X7 X6Y X6Z X5Y2 X5YZ X5Z2 X4Y3 X4Y2Z X4YZ2 X4Z3 X3Y4 X3Y3Z X3Y2Z2 X3YZ3 X3Z4 X2Y5 X2Y4Z X2Y3Z2 X2Y2Z3 X2YZ4 X2Z5 XY6 XY5Z XY4Z2 XY3Z3 XY2Z4 XYZ5 XZ6 Y7 Y6Z Y5Z2 Y4Z3 Y3Z4 Y2Z5 YZ6 Z7
8 th-order	X8 ...
9 th-order	X9 ...
10 th order	X10 ...

- Any coefficients that are not entered are set to zero.
- The optional coordinate axis is a flag for ASAP to create a surface/function with symmetry about some plane or axis. Substituting one or two of the coordinates in the functional equation with their squares, effectively doubling the order of the polynomial does this.
- The mathematical effect of order doubling is:

Entry	Mathematical Effect	Geometrical Effect
X	$\left(X-x\right)\rightarrow\left(X-x\right)^2$	symmetry about local x plane
-X	$\left(Y-y\right)\rightarrow\left(Y-y\right)^2,\left(Z-z\right)\rightarrow\left(Z-z\right)^2$	symmetry about local x axis

- The alternate **EXPLICIT** entry removes all **Z**-dependent terms from the polynomial except an implied linear **z** coefficient of -1 as shown in the following equation.

$$\mathrm{Z} - \mathrm{z} = \sum_{i=0}^{N}\sum_{j=0}^{i} c_{ijk}\left(X-x\right)^{i-j}\left(Y-y\right)^{j}$$

- This general 2-D polynomial can go as high as the 20th order in **X** and **Y**

Example

A sphere of radius 5 centered at 0,0,2 could be defined in the following ways:

```
GENERAL 0 0 2 -25 0 0 0 1 0 0 1 0 1
GENERAL 0 0 2; C -25; X2 1; Y2 1; Z2 1
GENERAL -Z 0 0 2; C -25; X2 1; Y2 1; Z2 1
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

GET (ASAP Command)

Retrieves ray data and copies it into variables.

Function

Modify or Use Internal Ray/Beam Data as Input

Syntax

```
GET [ k [ k' ] ]
```

Option	Description
`k k'`	number of a given ray or range of rays

Remarks

- Get the current data for all rays, ray **k**, or rays **k** to **k'** and place it into the input registers. This data can then be used in any future commands and can even be modified if the PUT command is used.
- If more than one ray is selected, then registers contain flux-weighted averages (except total flux).
- The register assignments are as follows:

Register	Literal	Ray/Beam Data
A0,B0,C0	X,Y,Z_DIR_B	Absolute X,Y,Z direction cosines of base ray
Ai,Bi,Ci	X,Y,Z_DIR_i	Relative direction vector of **i**th parabasal ray
D0	OPL	Optical path length from start of base ray
E1,E2,E3	X,Y,Z_EPOL	Components of unit polarization vector
F0	FLUX	Total flux in ray/beam
G0	DIVERG	Average divergence angle of beam
H0	HEIGHT	Average height of beam centered on base ray
Ii	PREV_O_i	ith previous split object for ray/beam
J0	SOURCE	Source number from which ray/beam originated
K0	CURR_OBJ	Current object at which ray/beam is located
L0	HITS	Total number of surfaces ray has hit (intersected)
M0	MEDIUM	Medium that ray/beam is in
N0	SPLITS	Number of times ray/beam has been split
N1	LEVELS	Number of times ray/beam has been scattered
P0	POLAR_0	Relative modulus of fundamental beam mode
P1,P2	POLAR_1,2	Relative moduli of polarization components
Q0	NUM_RAYS	Total number of ray/beams
Q1	NSOURCES	Total number of original sources
R0	PARENT	Number of ray from which this ray was split (parent)
S0	SHAPE	Beam shape number (see SHAPES command)
S1	FACTOR	Beam shape factor or number of higher modes
T0	PHASE_0	Relative phase angles of fundamental beam mode

T1,T2	PHASE_1,2	Relative phase angles of polarization components
U0,V0	U,VPARAMB	Parametric coordinates of base ray position
W0	WAVELEN	Wavelength of ray/beam
Wi	WAVLNS_i	Wavelength for ith source
X0,Y0,Z0	X,Y,Z_POS_B	Global X,Y,Z coordinates of base ray
Xi,Yi,Zi	X,Y,Z_POS_i	Relative coordinates of ith parabasal ray

- Some of the variables defined with the GET command relate to the polarization state of a coherent ray:
  1 If U describes the complex scalar wave function of the beam, the addition of a coherent polarization state causes the description to be:

$$\left(a\hat{q} + a'\hat{r}\right)U$$

  2 When the beam is first created,

$$\hat{q},\ \hat{r},\ a \text{ and } a'$$

are determined by the parameter of the current POLARIZ command. During the refraction operation, as the ray propagates, the quantities,

$$\hat{q},\ \hat{r},\ a \text{ and } a'$$

are re-computed so that

$$\hat{q}$$

is the local *s*-polarization direction, and

$$a \text{ and } a'$$

are determined by the *s* and *p* Fresnel equations.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PUT

SECTION

GRAPH (ASAP Command)

Creates 1-D profile(s) of the current distribution data file.

Function

Display/Modify Energy Distributions

Syntax

```
GRAPH [ i [ i'...]] [ APPEND ] [ AXIS [ x [ x' ]]] [ AUTO ] [ w ] 'title' ] [ DISCRETE ]
        f
        MIN
        MAX
[ ' comment …  ]
   :
```

Option	Description
i i' …	plot **i**th line of distribution data
f	plot line that is the floating point fraction **f** (between 0 and 1) of the total number of lines in the distribution data
MAX	plot line that contains MAXimum distribution value (default)
MIN	plot line that contains MINimum distribution value
APPEND	option to append current GRAPH to last GRAPH
AXIS	overwrite abscissa values
x x'	factors for overwriting abscissa values
AUTO	take the vertical function limits from the specified line data
DISCRETE	draws vertical lines (default w=0) or bars (0<w<.5)
'title'	optional title for plot (up to 64 characters)
'comment'	place up to three comments at the bottom of the plot

Remarks

- Produces a fancier and faster series of up to 10 one-dimensional profiles.
- Use TRANSPOSE before **GRAPH** to see reversed profiles or perpendicular to usual ones.
- The particular profiles are specified either as the line that contains the **MAX**imum (the default) or **MIN**imum distribution value, the integer **i**th line, or the floating point fraction **f** (between 0. and 1.) of the total number of lines.
- Up to three 72-character comments following this command can be placed at the bottom of the plot (as with any graphical command, a comment on the command itself gets placed at the top of the plot.
- **APPEND** tells ASAP to add the current profiles to those from the last **GRAPH** command.
- By default, ASAP writes abscissa values of the graph according to the values stored with the distribution data file. This default range of values may be overridden by the **AXIS** option. If no entries follow the AXIS option, the maximum absolute abscissa value is normalized to 1.0. If one entry follows the AXIS option, the abscissa values are divided by **x**. If two entries follow the AXIS option, the abscissa values range from **x** to **x'**.
- GRAPH axes numbering is multiple of three decades.
- The actual numbers being plotted are printed out in filename *.OTR or BRO106.DAT.
- The default operation of **GRAPH** is to plot the line containing the maximum value. At times you may want to plot the centerline through the distribution data (for example, when doing phase transfer function calculations); if so, use **GRAPH .5**.
- By default, the vertical functional limits of the graph are taken from the complete data set. If the **AUTO** option is used, the limits are taken from only the specified line data.
- Up to three 72-character comments can be placed at the bottom of the plot by entering each comment on a separate line following the command.
- For a single dataset only, the **DISCRETE** option draws vertical lines (default w=0) or bars (0<w<.5) from the horizontal (zero) axis to each data point.
- The comments are delimited by a single quote **'** , as shown.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

RANGE (ASAP Command)

GRID DATA (ASAP Command)

Creates a rectangular grid of rays.

Function

Create Rays/Beams

Syntax

```
GRID DATA [ file [ n ] [ RANDOM r ] ]
            u
```

Option	Description
`file`	name of the distribution file (default .DIS extension)
`u`	Logical unit number (default 9) of distribution data file (default .DAT extension)
`n`	specifies the creation of rays at every **n**th location
`RANDOM r`	randomization factor

Remarks

- Creates a rectangular grid of rays based on the coordinates found in the header of the distribution file (default **file.dis** or BRO009.DAT).
- A ray is created at each sample location in the file (or optionally every **n**th location).
- Each ray's flux is proportional to the corresponding data value in the file.
- Each ray position on the grid can be uniformly **RANDOM**ized within a region **r** times the local grid spacing.
- In addition to the usual CLIP POSition, the current flux CUTOFF and HALT settings may also truncate the distribution.
- If the labels in the file do not contain coordinates, then the data is mapped to the current plotting WINDOW at the zero depth plane.
- Used in conjunction with the SOURCE command, **GRID DATA** creates ray data that can be traced in ASAP.
- When in doubt about the exact distribution that is created by **GRID DATA** use the SPOTS POSITION command to display the grid.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW

GRID ELLIPTIC (ASAP Command)

Generates a rectangular grid of rays inside an elliptical aperture.

Function

Create Rays/Beams

Syntax

```
GRID ELLIPTIC X x y y' z z' n n' [ c ] [ RANDOM f ]
              Y y z z' x x'
              Z z x x' y y'
```

Option	Description
`X, Y, or Z`	coordinate axis
`x, y, or z`	location of plane on the coordinate axis

`y y', z z', or x x'`
`z z', x x', or y y'`
	minimum and maximum extent of the plane in the given direction

`n n'`	number of rays in each direction on the plane
`c`	fraction height of a central hole in the grid
`RANDOM f`	randomization factor

Remarks

- Generates a uniform rectangular grid of rays on the given plane, which is then clipped by an inscribed elliptical aperture.
- The flux of each ray is adjusted to give unit irradiance (flux/area) to the entire **GRID**.
- Rays are equally spaced in the two directions only if the extents and the number of rays in each direction are identical.
- The values for **n** and **n'** set the number of rays of the full rectangular grid. The number of rays contained in the inscribed elliptical aperture is usually less than the number in a full rectangular grid.
- The optional **c** entry specifies a fractional height for a central hole in the grid.
- Each position on the grid can be uniformly **RANDOM**ized within a region **f** times the local grid spacing.
- Used in conjunction with the SOURCE command, **GRID ELLIPTIC** creates ray data that can be traced in ASAP.
- When in doubt about the exact distribution that is created by **GRID ELLIPTIC** use the SPOTS POSITION command to display the grid.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID DATA
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW

GRID HEX (ASAP Command)

Generates a hexagonal grid of rays.

Function

Create Rays/Beams

Syntax

```
GRID HEX  X x y z h n [ RANDOM f ]
          Y y z x
          Z z x y
```

Option	Description
X, Y, or Z	coordinate axis
x, y, or z	location of plane on the coordinate axis
y z, z x, or x y	offset of grid
h	height of hexagon
n	number of rays along one side of hexagon
RANDOM f	randomization factor

Remarks

- Creates a hexagonal grid of rays normal to the given direction and centered at the given coordinates.
- The **h** is the height of the hexagon as measured from the center to one of its vertices.
- The **n** is the number of rays along one side of the hexagon.
- Each ray position on the grid can be uniformly **RANDOM**ized within a region **f** times the local grid spacing.
- Used in conjunction with the SOURCE command, **GRID HEX** creates ray data that can be traced in ASAP.
- When in doubt about the exact distribution that is created by **GRID HEX** use the SPOTS POSITION command to display the grid.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID DATA
GRID ELLIPTIC
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW

GRID OBJECT (ASAP Command)

Generates a uniform grid of rays on an object.

Function

Create Rays/Beams

Syntax

```
GRID OBJECT k [ n [ +-X ] ] [ RANDOM f ]
                      Y
                      Z
```

Option	Description
k	object number or name
n	number of rays
+X, -X, +Y, -Y, +Z, or -Z	specifies limit box end
RANDOM f	randomization factor

Remarks

- Generates a uniform grid of rays based on the location and dimensions of the given object **k**.
- If **k** is an EDGE/OBJECT, the ray positions lie at the centers of the object's facets. To control the number of facets on the EDGE/OBJECT use the object modifier FACETS.
- For SURFACE/OBJECTS approximately **n** rays are distributed over the specified LIMITS box end.
- Rays created with **GRID OBJECT** should be **MOVE**d off of the OBJECT to prevent intersection problems with that OBJECT.
- Each ray position on the grid can be uniformly **RANDOM**ized within a region **f** times the local grid spacing.
- Used in conjunction with the SOURCE command, **GRID OBJECT** creates ray data that can be traced in ASAP.
- When in doubt about the exact distribution that is created by **GRID OBJECT** use the SPOTS POSITION command to display the grid.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Apodization of Ray Distributions
GRID DATA
GRID ELLIPTIC
GRID HEX
GRID POLAR
GRID RECT
GRID WINDOW

GRID POLAR (ASAP Command)

Generates a circular grid of rays.

Function

Create Rays/Beams

First Syntax

```
GRID POLAR X x r r' a a' n n' [ w ] [ RANDOM f ]
           Y y
           Z z
```

Second Syntax

```
GRID POLAR X x r' n [ RANDOM f ]
           Y y
           Z z
```

Option	Description
X, Y, or Z	coordinate axis
x, y, or z	location of plane on the coordinate axis
r r'	minimum and maximum extent in the radial direction
a a'	minimum and maximum extent in the angular direction
n	number of rays in the radial direction
n'	number of rays in the first nonzero radial ring
w	weighting factor
RANDOM f	randomization factor

Remarks

- Generates a circular grid of rays centered on the coordinate axis when **w** is not unity.
- The angular arguments **a** and **a'** are entered in degrees.
- The sign of **n** determines how the range entries are to be used:

+n	divide **r** to **r'** into **n** rings
-n	divide **0** to **r'** into **n** rings, eliminate rings inside **r**
+n'	divide **a** to **a'** into **n** sectors (first ring)
-n'	divide 0 to 360 into **n** sectors, eliminate sectors outside **a** to **a'**

- For **r=0** or **n<0**, the rays are distributed radially according to the formula **R(i) = r'(i/n)^w**
- The rays are distributed evenly in the angular direction with the number in each ring chosen so that all rays represent an equal area on the plane. Therefore, when **w**=1 (the default), the grid is equipolar; when **w**=.5, the grid is a standard straight spokes distribution.
- If **r** is equal to zero, there is a ray/beam at the center of the grid.
- The flux of each ray is adjusted to give unit irradiance (flux/area) to the entire grid.
- Each ray position on the grid can be uniformly **RANDOM**ized within a region **f** times the local grid spacing.
- The 2nd syntax is an alternate shorter form is available that creates a full circular (of radius **r**') hexapolar grid (with **n** rings):

```
GRID POLAR X x r' n [ RANDOM f ]
           Y y
           Z z
```

- Only a rotationally symmetric apodization may be entered with USERAPOD.
- Used in conjunction with the SOURCE command, **GRID POLAR** creates ray data that can be traced in ASAP.
- When in doubt about the exact distribution that is created by **GRID POLAR** use the SPOTS POSITION command to display the grid.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID RECT
GRID WINDOW

GRID RECT (ASAP Command)

Generates a rectangular grid of rays inside a rectangular aperture.

Function

Create Rays/Beams

Syntax

```
GRID RECT X x y y' z z' n n' [ c ] [ RANDOM f ]
          Y y z z' x x'
          Z z x x' y y'
```

Option	Description
`X, Y, or Z`	coordinate axis
`x, y, or z`	location of plane on the coordinate axis

`y y', z z', or x x'`
`z z', x x', or y y'`
	minimum and maximum extent of the plane in the given direction

`n n'`	number of rays in each direction on the plane
`c`	fraction height of a central hole in the grid
`RANDOM f`	randomization factor

Remarks

- Generates a uniform rectangular grid of rays.
- The flux of each ray is adjusted to give unit irradiance (flux/area) to the entire **GRID**.
- Rays are equally spaced in the two directions only if the extents and the number of rays in each direction are identical.
- The optional **c** entry specifies a fractional height for a central rectangular hole in the grid.
- Each ray position on the grid can be uniformly **RANDOM**ized within a region **f** times the local grid spacing.
- Used in conjunction with the SOURCE command, **GRID RECT** creates ray data that can be traced in ASAP.
- When in doubt about the exact distribution that is created by **GRID RECT** use the SPOTS POSITION command to display the grid.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID WINDOW

GRID WINDOW (ASAP Command)

Creates a rectangular raster-type ray grid.

Function

Create Rays/Beams

Syntax

```
GRID WINDOW [ f [ m n [ RANDOM r ] ] ]
              MIN
              MAX
              CEN
              OBL
```

Option	Description
`f`	specifies the grid depth plane (default f=0)
`m`	sampling vertical rays
`n`	sampling horizontal rays
`MIN`	grid depth plane is given by the **MIN**imum value
`MAX`	grid depth plane is given by the **MAX**imum value
`CEN`	grid depth plane is given by the **CEN**ter value
`OBL`	grid depth plane is given by the last **OBL**ique value
`RANDOM r`	randomization factor

Remarks

- Creates a rectangular raster-type ray grid using the current WINDOW dimensions and sampling **m** rays vertical by **n** horizontal (or current PIXEL settings).
- The grid depth plane is given by **f** (default 0), the **MIN**imum, **MAX**imum, **CEN**ter, or last **OBL**ique value.
- Each ray position on the grid can be uniformly **RANDOM**ized within a region **r** times the local grid spacing.
- Used in conjunction with the SOURCE command, **GRID WINDOW** creates ray data that can be traced in ASAP.
- When in doubt about the exact distribution that is created by **GRID WINDOW** use the SPOTS POSITION command to display the grid.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT

GROUP (ASAP Command)

Groups a collection of objects or sources as a single unit.

Function

Create/Modify Objects

Syntax

```
GROUP [ i [ i' ... ] ] [ SOURCES j [ j' ... ] ]
       -n                           -n
        0
```

Option	Description
`i i' ...`	object numbers or names
`-n`	specifies last n objects
`SOURCES`	specifies grouping sources
`j j' ...`	source numbers
`SOURCES -n`	specifies last n sources
`0`	specifies last group

Remarks

- Temporarily declares the objects/sources whose number (or object names possibly with '?' wildcards) are entered on it, the last "n' objects/sources, the last group (0), or all objects (no entries) as belonging to the same "group. This single unit can be modified using the SHIFT or ROTATE command.
- The **SOURCES** option can be abbreviated only to SOURCE to avoid confusion with an object name.
- Any linear transformation commands that immediately follow this command are applied to every object and source in the group.
- A few specific examples include:

GROUP	!all objects
GROUP ?	!all objects
GROUP -1000	!all objects
GROUP .?	!objects in same branch as last object

- If **GROUP** is used without any additional arguments, all previously defined objects are grouped.
- If **-n** is entered, only last **n** objects/sources are grouped.
- If **0** is entered, the last group is grouped again.
- If **i, i', ...** are entered, only these objects are grouped.
- If **j, j', ...** are entered, only these sources are grouped.
- The reference point of the group is defaulted to that of the first object/source.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

CONSIDER

HALT (ASAP Command)

Sets the conditions controlling ray termination.

Function

Setup Trace
Create/Modify Objects

Syntax

```
HALT [ n ] [ OFF ] [ c ]
             +-X
               Y
               Z
```

Option	Description
n	maximum number of times a ray can consecutively intersect the same non-LENS object
OFF	turns off the propagation direction
+-X, +-Y, or +-Z	undesirable propagation direction
c	current flux/initial flux threshold ratio

Remarks

- Forces the tracing of certain rays to be halted.
- The **n** must be entered as an integer.
- The literal entry picks out a certain coordinate direction or turns it OFF. If a ray attains a component in this direction after reflection or refraction or the number of consecutive interactions with the same object reaches the integer **n** (default 12), then tracing halts for that particular ray. There is no default for the undesirable propagation direction.
- The default for **c** is 1.E-6 (that is, a ray is stopped when the ratio of current flux to initial flux drops below 1.E-6).
- The **c** must be entered as a decimal number.
- The **HALT** command is more commonly used with the coherent point source commands GAUSSIAN and DECOMPOSE.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

CUTOFF
DECOMPOSE
GAUSSIAN

HARVEY (ASAP Command)

Creates a Harvey (linear shift invariant) scatter model or simple specific model.

Function

Create/Modify Media, Coatings, Scatter Models

First Syntax: isotropic model

```
HARVEY b s [ l [ m [ n [ w ] ] ] ] [ PLOT [ a a' ... ] ]
```

Second Syntax: **anisotropic model**

```
HARVEY X b s l l' [ w ]
       Y
       Z
       :
```

Option	**Description**
b	BSDF at 0.01 radians (0.573 degrees) from specular; if l is used b is the maximum BSDF (at specular).
s	asymptotic fall-off with angle (typically between -1 to -2.5)
l and l'	A-Ao and B-B$_o$ shoulder point in radians (see Remarks)
m, n	2 additional invariance parameters postulated for rough surfaces
w	wavelength (in current or eventual wavelength units)
PLOT	plots the model in log(b-b$_o$) and angle space
a a' ...	user-defined degree specular angles

Remarks

- The out-of-field analog to the SCATTER BSDF command.
- Use with importance area sampling.
- If ***l*** is used,

$$b \quad b_0 \text{ where } b_0 \quad b\ 100\ l^{\ s}$$

Simple Specific (Isotropic) Models:

- If the optional b-b$_o$ shoulder point **l** (in radians) is given, **b** is the maximum BSDF (at specular). These parameters describe a shift-invariant generalized Lorentzian function of B-Bo that normally fits the scatter from smooth surfaces (RMS as well as wavelength) extremely well.
- The **m** and **n** are two additional invariance parameters postulated for rough surfaces; that is, if:

$$BSDF = \left\{\frac{bf\left(B-B_o\right)}{1}\right\}^s \quad f\{x\} = \sqrt{1+x^2}$$

for m=0 and n = 0 (the default), then:

$$g^m BSDF = bf\left\{\frac{B-B_o}{\mathrm{l}g^n}\right\}^s \quad g = \frac{\left(C+C_o\right)}{2}$$

where **B, C** are the sine, cosine of the scatter angle from normal and **Bo, Co** are the sine, cosine of the specular angle. For typical rough surfaces, **m** is approximately 2 and **n** around 1.

- For typical rough surfaces, **m** is approximately 2 and **n** around 1. The **w** is the wavelength (in current or eventual wavelength units) at which this model is defined (or was measured).

Elliptical (Anisotropic) Harvey Model:

- Scattering from anisotropic surfaces is not rotationally symmetric at normal incidence and not necessarily symmetric about the plane of incidence otherwise. Therefore, the orientation of the model on the surface is important and is generally specified by an axis for the second command entry. For syntax information, see ...MODEL... (ASAP Command Argument).
- The **b** is the maximum BSDF (at specular), **s** is the asymptotic fall-off with angle (typically between -1 to -2.5). The **l** and l' are the **A-Ao** and **B-Bo** shoulder points in radians. More precisely,

$$BSDF = bf\left\{\frac{A-A_o}{1},\frac{B-B_o}{1'}\right\}^s \quad f\{x,y\} = \sqrt{1+x^2+y^2}$$

where **A, B** are the scatter direction cosines and **Ao, Bo** are the specular.

Both Models

- The **w** is the wavelength (in current or eventual wavelength units) at which this model is defined (or was measured). The default is the current value from the last WAVELENGTH command. If it is greater than zero, any ray whose wavelength is different automatically has its scatter scaled according to the smooth surface approximation.
- ...PLOT (ASAP Command Argument) plots the model (common base 10 logarithm of the BSDF) for up to seven specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). The current PIXELS setting controls the resolution of these plots in direction cosine space. Also, creates a distribution file **name_angle.dis** for each of these angles.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

BSDF Fit Utility

Mathematical Models for Isotropic-Surface Scattering

MODELS Overview

MODELS

HEADER (ASAP Command)

Redefines the file header for the current data in the display file.

Function

Display/Modify Energy Distributions

Syntax

```
HEADER zlabel z flabel ylabel y y' xlabel x x' 'title'
```

Remarks

Any entries that are not included at the end of the command or any literal underscores "_" do not alter that part of the header. For example, to change only the horizontal label and range, enter:

```
HEADER _ _ _, "New Label" -12 12
```

Note the space between each underscore "_". You must include the spaces or it will not work.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

HELIX (ASAP Command)

Creates a general helical curve.

Function

Define/Modify Curvedge Entities

Syntax

```
HELIX X x y z x' y' z' t [ n ]
      Y y z x y' z' x'
      Z z x y z' x' y'
```

Option	Description
X, Y or Z	coordinate axis
x, y, or z	axial location of the starting plane
x', y', or z'	axial location of the ending plane
y z, z x, or x y	elliptical semi-heights at the starting plane
y' z', z' x', or x' y'	elliptical semi-heights at the ending plane
t	number of turns (not necessarily integer)
n	number of linear segments

Remarks

- If the axial locations of the starting and ending planes are the same, then the helix becomes a simple planar spiral.
- The helix consists of **t** turns (not necessarily integer).
- By default, the helix is perfectly smooth. Optionally, it can consist of **n** linear segments.
- This edge is a combination of straight line or higher-order curved segments.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

HISTORY (ASAP Command)

Recalls ray data from a previously saved binary file for plotting or listing.

Function

Analyze Ray/Beam Data

Syntax

```
HISTORY [ k ] [ PLOT [ m ] ]
                SPOTS
```

Option	Description
k	SAVE file number from the last TRACE command
PLOT	PLOT the reverse trajectories of the current ray set
m	
SPOTS	Plot the SPOTS where the rays struck the objects

Remarks

- If a SAVE file number k exists from the last TRACE, the HISTORY command can be used to list or PLOT the reverse trajectories of the current ray set.
- Alternatively, just the SPOTS where the rays struck the objects can be plotted instead of the trajectory lines.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands

HORN (ASAP Command)

Creates a generalized tube with polynomial profiles and elliptical/rectangular cross-sections.

Function

Define/Modify Surfunc Entities

First Syntax

```
HORN X x r [ r' r" ... ] [ FIT n x [ x'  x"    ... ] ]
     Y y                 [           y [ y' y" ... ] ]
     Z z                 [           z [ z' z" ... ] ]
```

Second Syntax (see **Remarks**)

```
HORN Z z x0 [ x1 x2 ... ] [ FIT n   z0  [ z1  z2 ... ] ]
         y0 [ y1 y2 ... ] [       n' z0' [ z1' z2' ... ] ]
         q0 [ q1 q2 ... ] [       n" z0" [ z1" z2" .. ] ]
```

Option	Description
`X, Y or Z`	axis of symmetry
`x, y or z`	location along coordinate axis
`r r' r"...`	radial coefficients
`FIT`	flag to fit data
`n`	order number for the polynomial

`x x' x"..., y y' y"...,`
`or z z' z"...`
	data points on surface

Reference Point

At location along coordinate axis.

Surface Normal

Radially outward from the axis.

Autolimiting

Requires LOCAL modifier if not doing a **FIT**.

Remarks

- Creates a surface symmetric about the given axis with a radial profile determined by the following polynomial in the radial distance squared:

$$r^2(d) = r + r'd + r''d^2 + \ldots$$

  where **d** represents the particular axial distance coordinate relative to the reference point (and therefore **d** is measured in the local coordinate system).
- The program can also **FIT** the given set of radial and axial positions to an **n**th-order polynomial (**n** less than or equal to 10).
- Rectangular/elliptical horns with different polynomial profiles in the two cross-sections can be created with variations of the following format (only the **Z** axis form is shown for sake of brevity):

```
HORN Z z x0 [ x1 x2 ... ] [ FIT n   z0 [ z1  z2 ... ] ]
         y0 [ y1 y2 ... ] [     n'  z0' [ z1' z2' ... ] ]
         q0 [ q1 q2 ... ] [     n" z0" [ z1" z2" .. ] ]
```

$$X^2 = x0 + x1(Z-z) + x2(Z-z)^2 + \ldots$$

$$Y^2 = y0 + y1(Z-z) + y2(Z-z)^2 + \ldots$$

$$q = q0 + q1(Z-z) + q2(Z-z)^2$$

  The **q** parameter controls whether the cross-section is elliptical, rectangular, or something in between (See the TUBE command for more details). As before, each one of these polynomials can be determined by a least squares **FIT** to a set of values and positions.

- Either way you do it, remember that the coefficients or the fit points are relative to the reference point (the third entry). The following restrictions on the polynomial orders apply:

n	less than or equal to 10
n'	less than or equal to 8
n+n'	less than or equal to 10
n"	less than or equal to 6

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

TUBE

IDEAL (ASAP Command)

Creates an idealized optical element.

Function

Define/Modify Lens Entities

Syntax

```
IDEAL X x h [ t [ h' ] ] [ RANA ] [ fcn ]
      Y y                  RANB
      Z z
[ a b c d [ o ] ]
[ p q r s ]
```

Option	Description
`X, Y, or Z`	specifies axis of symmetry
`x, y, or z`	location along coordinate axis
`h`	input aperture semidiameter (height)
`t`	output distance
`h'`	output aperture semidiameter (height)
`RANA or RANB`	randomize the output rays azimuthally (around the axis)
`fcn`	name of the function
`a b c d`	components of 2x2 ABCD matrix
`o`	object distance from first conicoid
`p q r s`	2x2 complex Jones matrix

Remarks

- Defines an idealized (but unphysical) lens of input height **h**, output distance **t**, and output height **h'**. The input ray vectors are linearly related to the output ray vectors by the 2x2 ABCD matrix given on the next line of input (default is an identity matrix), and whose input polarization state is related to the output polarization states via a Jones matrix.
- Jones matrix behavior is virtually independent of the incidence angle.
- Each matrix is entered on an individual input line. The matrix coefficients are as follows:

Ray Matrix	Jones Matrix
a b	p q
c d	r s

- Common idealized optical systems include:

  Perfect lens with focal length **f: (a,b,c,d)=(1,0,-1/f,1)**

  Afocal system of angular magnification **m**: **(a,b,c,d)=(1/m,0,0,m)**

  Nonlens where output rays are extensions of input rays: **(a,b,c,d)=(1,t,0,1)**
- The input and output media are assumed to be the same isotropic medium. Therefore the determinant of the ray matrix should be unity, that is ad-cb=1. There are no ray aberrations at all conjugates.
- The optional additional line of input (**p q r s**) is the four elements of the 2x2 complex Jones matrix for linearly altering the polarization states of transmitted beams. In general, Jones matrices are really only defined for beams normally incident on the reference plane. Therefore, non-normal incidence beams may have their polarizations (and fluxes) altered in unexpected ways.
- The **RAN** options randomize azimuthally (around the axis) the output rays. **RANA** does just the angle by rotating the output ray about its exit point. **RANB** does both position and direction by rotating about the center of the output aperture. These can be used to simulate some non-imaging effects.
- The fluxes of the rays can be apodized both radially and directionally if a $FCN **fcn** is specified. The function takes as it first argument _1 the normalized (between 0 and 1) radial position of the ray. The optional second argument _2 is the sine of the angle from normal for the ray direction. The valued returned (last expression) is multiplied with the ray flux to get the new ray flux. For example, these two options could be combined to do an efficient simulation of an incoherent multi-mode fiber:

  ```
  NA=.5 !numerical aperture of fiber
  $FCN CUTOFF RECT(_2/(2*NA)) !directional cutoff
  IDEAL Z 0 .1 1000 .1 RANB CUTOFF !1000mm long 100um fiber
  ```
- There are no ray aberrations at all conjugates. However, the wavefronts from this lens can only be perfectly spherical for a single conjugate. The **o** is the object distance from the first surface for this perfect imaging. The default value for **o** is 0; the

object is at infinity (optical path length is not constant for different field points).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PERFECT (ASAP Command)

IESFILE (ASAP Command)

Writes out IES-format file.

Function

Display/Modify Energy Distributions

Syntax

```
IESFILE [ name [ ELLIP w l h ] ] [ PAD [ p ] ] [ 'string' ]
                 RECT
```

Option	Description
name	name of the created file (extension is .IES)
ELLIP or RECT	add the devices ELLIPtical or RECTangular width, length, and height to the file's header
w	width of the elliptical or rectangular device
l	length of the elliptical or rectangular device
h	height of the elliptical or rectangular device
PAD p	flag to pad the distribution with **p** values (default is minimum)
'string'	adds a comment string to the file's header

Remarks

- Writes out a standard IESNA photometric file (**name.IES**) of an angular distribution created by either a RADIANT command (A or C photometry where the polar axis becomes vertical) or a SPOTS DIR and ANGLES combination (B photometry).
- The devices **RECT**angular or **ELLIP**tical width **w**, length **l**, height **h**, and a comment **string** may be added to the file's header.
- If the distribution does not cover the whole sphere, then it can be optionally **PAD**ded with **p** values (default is minimum).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

IMAGE (Curve/edge) (ASAP Command)

Images a curve/edge through a specified LENS entity.

Function

Define/Modify Curvedge Entities

Syntax

```
IMAGE l k k'
```

Option	Description
`k k'`	starting/ending curve/edge points in space
`l`	lens entity

Remarks

- Images the points of the current curve/edge from space **k** to space **k'** of LENS entity **l**.
- The spaces (**k k'**) are numbered from 1 (before the first conicoid) to N+1 (after the last conicoid N).
- IMAGE uses the auxiliary axis technique that steps the image through the centers of curvature of the conicoids (the aperture semi-diameters, conic constants and obscuration ratios of the conicoids do not affect the imaging) and always produces an image even if it is virtual or unphysical. The resulting imaging transform is stigmatic (points go into points) but not necessarily collinear (lines go into lines). Therefore, it is only an approximation since in any real optical system, the image is aberrated (not a perfect point focus).
- IMAGE is most useful for **SCATTER TOWARDS** edges that represent images of important areas.
- The **IMAGE** command used in pre-ASAP 5.1 versions is now named PICTURE.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SCATTER TOWARDS
REPEAT (ASAP Command)

IMAGE (Global Point) (ASAP Command)

Images a global point through a specified LENS entity.

Function

Define/Modify Lens Entities

Syntax

```
IMAGE k x y z k'
```

Option	Description
`k k'`	starting/ending spaces
`x y z`	global point

Remarks

- Images the given point (**x y z**) in space **k** of the current lens into space **k'** and displays the coordinates.
- The spaces are numbered from 1 (before the first conicoid) to N+1 (after the last conicoid N).
- Uses the auxiliary axis technique that steps the image through the centers of curvature of the conicoids (the aperture semi-diameters, conic constants and obscuration ratios of the conicoids do not affect the imaging) and always produces an image even if it is virtual or unphysical. The resulting imaging transform is stigmatic (points go into points) but not necessarily collinear (lines go into lines). Therefore, it is only an approximation since in any real optical system, the image is aberrated (not a perfect point focus).
- The **IMAGE** command used in pre-ASAP 5.1 versions is now named PICTURE.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

IMAGE (Ray Positions) (ASAP Command)

A ray modifier that finds the conjugates of ray coordinates through a LENS (images the current ray positions through a specified LENS entity).

Function

Modify Ray/Beam Data

Syntax

`IMAGE l k k'`

Option	Description
`l`	LENS entity
`k  k'`	starting/ending spaces

Remarks

- Images the current ray positions (not directions) from space **k** to space **k'** of LENS entity **l**. The spaces are numbered from 1 (before the first conicoid) to N+1 (after the last conicoid N).
- **IMAGE** uses the auxiliary axis technique that steps the image through the centers of curvature of the conicoids (the aperture semi-diameters, conic constants and obscuration ratios of the conicoids do not affect the imaging) and always produces an image even if it is virtual or unphysical. The resulting imaging transform is stigmatic (points go into points) but not necessarily collinear (lines go into lines). Therefore, it is only an approximation since in any real optical system, the image is aberrated (not a perfect point focus).
- **IMAGE** is most useful for mapping a grid of rays at an internal stop position of an optical system into its object space (that is, entrance pupil).
- The **IMAGE** command used in pre-ASAP 5.1 versions is now named PICTURE.

Example

Mapping a grid of rays at an internal stop position of an optical system into its object space.

```
LENS 99; ...   !!equivalent imaging train
:
GRID ...       !!covers internal stop
  IMAGE 99 4 1 !!image into entrance pupil
SOURCE DIR ... !!now set directions
  MOVE TO ...  !!out in front of first surface
TRACE
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

IMMERSE (ASAP Command)

Creates rays in MEDIA in other than zero.

Function

Setup Beam Creation

Syntax

```
IMMERSE [ m ]
```

Option	Description
m	media name or number (default=0)

Remarks

- Sets the starting MEDIA (refractive index) for future rays to **m** (name or number, default 0). AIR/VACUUM is available as an index of 1.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

INTERFACE (ASAP Command)

Assigns a reflective, refractive or diffractive interface to objects.

Function

Create/Modify Objects

First Syntax

```
INTERFACE r [  t m m'] …
          COAT k                               coat   coat'
              -k
              +k
```

Second Syntax

```
INTERFACE … DIFFRACT i j [ e j' e' … ]
                           coat   coat'
```

Third Syntax

```
INTERFACE … DIFFRACT i coat j .j' j" …
```

Fourth Syntax

```
INTERFACE … DIFFRACT i… DIFFRACT i' … [ DIFFRACT i" … ]
```

Option	Description
r	reflection coefficient
t	transmission coefficient
m m'	media numbers (or names) on each side of the object
k	coating number (or name)
DIFFRACT	flag to assign a diffractive interface to an object
i	MULTIPLE surface number
j j' ...	diffraction order number(s)
e e' ...	relative efficiencies of the corresponding orders
coat coat'	name of a given coating property

Remarks

- If the object surface is an optical boundary through which rays are to be traced, the optical properties of the interface must be specified using the INTERFACE command after the definition command for that object.
- The **r** is the relative energy (or complex amplitude) reflection coefficient for the surface (or the reflectors in a lens structure).
- The **t** is the transmission coefficient of the surface (or the refractors in a lens structure, their reflectivities are then assumed to be 1-t).
- When explicitly stating numerical **r** and **t** quantities in the INTERFACE command, the COATING parameter should not be used, otherwise ASAP produces the wrong number of rays. For example, instead of entering: INTERFACE COATING 0 1 GLASS GLASS

  enter:

  INTERFACE 0 1 GLASS GLASS

  where "0 1" are examples
- If **t** is nonzero, then **m** and **m'** are the numbers (or names) of the media on each side of the object surface.
- The order in which **m** and **m'** are entered is arbitrary; ASAP determines which to use.
- If either side of the interface is air or vacuum, **m** or **m'** may be set to 0 (zero), **AIR**, or **VACUUM**.
- Alternatively, the **r** and **t** coefficients can be determined from coating number or name **k** (0 for a BARE SUBSTRATE) and the current WAVELENGTH. ASAP uses the reflection and transmission values from the COATING PROPERTIES table or calculates the coefficients from the COATING LAYERS table based on the normal incident form of Fresnel's formulae.
- The signs in front of **k** can be used to instruct ASAP to either propagate only a reflected ray (**-k**: t=0) at that object or to propagate only a transmitted ray (**+k**: r=0) at that object.
- **INTERFACE COATING +k m m'**, where **k**=0 or **BARE_SUBSTRATE**, is the syntax for modeling the polychromatic transmission properties of a bare substrate. The transmission coefficient is calculated from the normal incident form of the

Fresnel formulae and applied to all incident ray angles.

- If an **INTERFACE** command does not follow an object definition, the surface is assumed to be perfectly absorbing and all rays reaching the surface is trapped there.
- **Diffract**ive lines are created by the intersection of the object surface with the different sheets of a MULTIPLE surface **i**, that is, a ruled linear grating is created if **i** is a plane, a zone plate is created if the surface is a cylinder, etc.
- If **i** is positive, the multiple sheet spacing is taken to be the grating spacing in system units. If **i** is negative, the spacing is assumed to be in the same units as the last WAVELENGTH specification. The number of sheets entered on the **MULTIPLE** surface command has no bearing on this application.
- ASAP generates diffracted rays/beams for the diffraction order numbers given by the **j**'s with relative efficiencies given by the corresponding positive **e**'s.
- If an **e** is entered as a negative number or as a name **coat**, then it is a **COATING PROPERTY** possibly containing polychromatic complex amplitudes.
- Multiple exposure holograms can be modeled by using more than one **DIFFRACT** option (the **zero**th-order should only be specified once).
- If an "e" is entered as a negative number or as a name then it is a **COATING PROPERTY** possibly containing polychromatic complex amplitudes.
- A named **COATING MODEL** can be used to specify the angular variation of the diffraction order efficiencies; that is,
- INTERFACE ... DIFFRACT i coat j j' j" ...
- Multiple exposure holograms can be modeled by using more than one DIFFRACT option (the zeroth-order should only be specified once):

  INTERFACE ... DIFFRACT i ... DIFFRACT i'... [DIFFRACT i" ...]

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

COATING MODELS
Interface Overview
COATING LAYERS
COATING PROPERTIES
MULTIPLE
WAVELENGTH
INTERFACE REPEAT
Combinations of INTERFACE and FRESNEL Settings

INTERFACE REPEAT (ASAP Command)

Assigns the interface characteristics of either a specified object or the previous object to the current object.

Function

Create/Modify Objects

Syntax

```
INTERFACE REPEAT [ i ]
```

Option	Description
i	object number

Remarks

- If **i** is zero, all the interface properties are removed.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Interface Overview
INTERFACE

INVERT (ASAP Command)

Reverses the parametric direction of a curve.

Function

Define/Modify Curvedge Entities

Syntax

```
INVERT
```

Remarks

- Reverses the parametric direction of a curve; that is, spatially it remains the same curve but the last point becomes the first and vice-versa.
- **INVERT** is useful when connecting curves into a parametric mesh **OBJECT**.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

OBJECT (ASAP Command)
REPEAT (ASAP Command)

IRRADIANCE (ASAP Command)

Selects the axis that energy or peak flux density is computed relative to a given fixed direction for all sample points.

Function

Calculate Diffraction/Propagation Effects

Syntax

```
IRRADIANCE [ OFF ]
             X
             Y
             Z
```

Option	Description
`OFF`	compute irradiance relative to local propagation direction
`X Y or Z`	compute irradiance relative to given fixed direction

Remarks

- Causes any future **SPREAD** or **FIELD** commands (until it is turned OFF) to calculate the irradiance (flux per unit area) relative to a given fixed direction for all sample points.
- If no entry is given, the normal to the sample grid is used.
- If the **IRRADIANCE** command is turned **OFF**, by default, the **SPREAD** and **FIELD** commands calculate the energy or peak flux density relative to the local propagation direction at each sample point.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Coherent/Incoherent Analysis
IRRADIANCE (ASAP Command)
VIOLATION (ASAP Command)
SPREAD (ASAP Command)
FIELD (ASAP Command)

ISOMETRIC (ASAP Command)

Creates an isometric view of the current distribution data file.

Function

Display/Modify Energy Distributions

Syntax

```
ISOMETRIC [ s ] [ n ] [ 'title' ]
```

Option	Description
`s`	used to readjust the vertical scale
`n`	plot every **n**th line (default is n=1)
`'title'`	optional title for plot (up to 64 characters)

Remarks

- Produces a true isometric view of the data and no 1-D cross-sections.
- ISOMETRIC is similar to the PLOT3D command.
- The floating point entry **s** can be used to readjust the vertical scale.
- The integer entry can be used to plot every **n**th line instead of the default of every line (n=1).
- The title is delimited by a single quote **'** , as shown.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

LAMBERTIAN (ASAP Command)

Creates a Lambertian scatter model.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
LAMBERTIAN t [ PLOT [ a a' ... ] ]
```

Option	Description
`t`	total hemispherical diffuse scatter ratio (less than 1)
`PLOT`	plots the model in log(b-b_o) and angle space
`a a' ...`	user-defined degree specular angles

Remarks

- Use with importance area sampling.
- The **PLOT** option plots the model (common base 10 logarithm of the BSDF) for up to seven specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). The current PIXELS setting controls the resolution of these plots in direction cosine space.
- The BSDF is independent of wavelength, incidence direction, and scatter direction.
- Creates a distribution file **name_angle.dis** for each of these angles.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MODELS Overview

MODELS

LENSES ENTITIES (ASAP Command)

Signals ASAP that lens definition commands follow.

Function

Define/Modify Lens Entities

Syntax

```
LENSES  [ i ]
ENTITIES
```

Option	Description
i	starting number for defining LENSES

Remarks

- The default value for **i** is one more than the highest lens number previously defined. The **i** is initialized to one at start of program execution.
- EDGE, LENS, and SURFACE data currently reside in the same internal storage locations. Therefore, a LENS number cannot be the same as an already defined EDGE or SURFACE number.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

LEVEL (ASAP Command)

Controls the number of times a scattered ray may be rescattered.

Function

Create/Modify Objects

Syntax

```
LEVEL [ n ] [ OFF ] [ c ]
              ALL
```

Option	Description
n	maximum number of scattering levels allowed for random diffuse rays
OFF	turns off generation of random diffuse rays
ALL	generates scattered rays for each specular child ray
c	diffuse ray relative flux threshold

Remarks

- **LEVEL** can be applied on an OBJECT by OBJECT basis. In effect, this is now an OBJECT modifier, similar to the INTERFACE command.
- Controls the level of scattered ray splitting in much the same way the **SPLIT** command controls the level of specular ray splitting.
- The parent rays (a ray originally created by the **GRID** or RAYSET commands) are allowed to generate random diffuse scatter rays at any object with a SCATTER RANDOM interface. The **LEVEL** command controls the splitting of the children scatter rays (rays that have been split off of a parent ray). Therefore **LEVEL 1** tells ASAP to split the parent rays, but the children scatter rays are not allowed to split. **LEVEL 2** allows the parent rays and the children scatter rays to split as often as necessary, but the grandchildren scatter rays are not allowed to split.
- Deterministic near-specular and back-scattered rays (created at interfaces with SCATTER RMS or SCATTER BSDF) are never rescattered, so LEVEL should be set to 1 if only these two scatter mechanisms are of interest.
- If the fractional scattered energy (relative to the incident ray) drops below **c** (defaulted to 1E-12), the scattered ray is not generated.
- Normally, scattered rays are not generated for any specular child rays, only the parent. If **ALL** is specified, scattered rays are generated for each specular child ray (including multiple diffraction orders, extraordinary rays, etc.), so that "bi-directional" scatter can occur at a partially reflecting/transmitting interface.
- Sets the default refraction/reflection controls for all objects or may be applied only to a specific object.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Refraction/Reflection Controls
GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW
RAYSET
SCATTER RANDOM
SCATTER RMS or SCATTER BSDF
SPLIT

LIGHTS (ASAP Command)

Sets up optional light sources for the **RENDER** command.

Function

Setup Plots and Verify System

Syntax

```
LIGHTS  e a,b,c [ e' a',b',c' [ e" ... ] ]
       -e x y z  -e' x  y  z   -e"
```

Option	Description
`e, e', ...`	relative irradiance of the collimated beams
`a,b,c a',b',c' ...`	direction of propagation of the collimated beams
`-e, -e', ...`	isotropic point sources
`x y z, x' y' z', ...`	location of the point sources

Remarks

- Sets up optional light sources (up to 10) for the RENDER command.
- If **e** is positive, then it is the relative irradiance of the collimated beam and **a,b,c** are its direction of propagation.
- If **e** is negative, the light is an isotropic point source at (**x y z**).
- The addition of these light sources can significantly increase the time to render a scene because ASAP must check for obscurations along the path of the light sources.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

RENDER

LIMITS (ASAP Command)

Assigns a global bounding box to the object.

Function

Create/Modify Objects

First Syntax (Long format):

```
LIMITS x x' [ y y' z z' ] [ X [ f ] ]
                            Y
                            Z
                            SHORTEST
                            LONGEST
                            NORMAL
```

Second Syntax (Short format):

```
LIMITS [ OFF     ]
         REPEAT i  [ X [ f ] ]
         STATS       Y
         AXIS        Z
        -X  x        SHORTEST
        -Y  y        LONGEST
        -Z  z        NORMAL
        +X  x'       OFF
        +Y  y'
        +Z  z'
         EXPAND r
                X r
                Y
                Z
```

Option	Description
`x x'`	minimum and maximum extent in the global x direction
`y y'`	minimum and maximum extent in the global y direction
`z z'`	minimum and maximum extent in the global z direction
`X, Y, or Z`	coordinate direction
`SHORTEST`	coordinate axis determined by shortest limit box dimension
`LONGEST`	coordinate axis determined by longest limit box dimension
`NORMAL`	coordinate axis determined by coordinate direction nearest to the surface normal
`OFF`	makes limits a rectangular box nearest surface normal
`f`	fractional height of inner boundary (default=0)
`LIMITS OFF`	temporarily turn off future limits checking
`REPEAT i`	copies/repeats the limits box from object **i**
`STATS`	resets the limits box according to the previous **TRACE STATS** command
`AXIS`	resets the limits axis
`-X x, -Y y, -Z z ...`	specifies the specific bounding box side
`EXPAND`	scales the limits box
`r`	relative scale factor

Remarks

- The minimum and maximum extents are defined in global coordinates.

- If the coordinate direction is not entered, the object is constrained by a rectangular box whose endpoints are given by the minimum and maximum extent.
- If the coordinate direction is entered, the object is constrained by a cylinder with a constant elliptical cross-section in planes perpendicular to the given coordinate.
- The optional inner boundary of fractional height **f** (default is zero) may be used to put a proportional hole in the object in the coordinate direction. If **f** is negative, the proportional hole is rectangular.
- The **LIMITS** command can be used for very simple bounding of an object. For more sophisticated surface boundaries, **BOUND** surfaces can also be employed.
- The short format can be used to: turn **OFF** future limits checking temporarily, set the limits on the current object to those of a previous object **I**. Use the ranges of ray intersections found from the previous TRACE STATS command, or reset the limits AXIS, or just one of the six limits values.
- **EXPAND** can be used to enlarge (or shrink) by a relative amount **r** the entire limits box or just in one direction, that is, **EXPAND -.1** shrinks the entire box by 10 percent and **EXPAND X .1** enlarges the limit box by 10 percent in the X direction.
- Any linear transformation commands that follow an object definition are applied to the object surface and all boundary surfaces, that is, the entire object. The coordinate ranges entered with the **LIMITS** command are also adjusted so that the object lies entirely in a new 3-D orthogonal box. Therefore, the shape of the object can change with a **ROTATE** or **SKEW** if only the **LIMITS** are used to bound it. To avoid this, use a **LOCAL** box on the base **SURFACE** of the object.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

BOUNDS
LOCAL
ROTATE
SKEW

LINE (ASAP Command)

Creates a line edge.

Function

Define/Modify Curvedge Entities for DASHED line

Syntax

```
LINE x y z x' y' z' [ n ] [ DASHED ]
```

Option	Description
x y z	line starting coordinates
x' y' z'	line ending coordinates
n	number of connected segments (default is 1 or value specified on previous LINE command)
DASHED	disconnects every other segment of the line (open)

Remarks

- Creates a straight line from the first point to the second point that is equally divided into **n** connected segments.
- The default is 1 or value specified on the previous LINE command. Use **-n** if you want it to become the default for future LINE commands.
- If the DASHED option is used, then every other segment of the line is open (not connected).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

LIST (ASAP Command)

Lists currently selected ray data by the second literal entry for the current ray set.

Function

Analyze Ray/Beam Data

Syntax

```
LIST POSITION  [ k ]
     DIRECTION
     SOURCES
```

Option	Description
POSITION	lists ray positions, fluxes, current objects, and optical path lengths
k	ray number whose coordinates and optical path length are subtracted from each ray before they are listed
DIRECTION	lists ray directions, fluxes, current objects, and optical path lengths
SOURCES	lists wavelength and ray numbers for each source

Remarks

- Lists the specified ray data (POSITION, DIRECTION or SOURCES). By default, the positions or directions of each base ray are used.
- All rays are listed. To select a subset of rays to be listed, the CONSIDER and SELECT commands should be used.
- The SOURCES option lists the wavelength and range of ray numbers for each currently defined source.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

AXIS allows ray data to be listed in local and/or cylindrical ray trace coordinates.
CONSIDER
GET
LIST ELLIPSE
LIST INTEGER
LIST (Parabasal Ray Data)
LIST RAYS
PUT
SELECT

LIST (Parabasal Ray Data) (ASAP Command)

Lists currently selected base and parabasal ray data.

Function

Analyze Ray/Beam Data

Syntax

```
LIST P#
     D#
```

Remarks

- Creates a list of the ray data specified by the given option for the current ray set.
- Any particular parabasal ray may be selected by specifying its number (**#)**. For example, P0 indicates base ray position; D1 indicates first parabasal ray direction, and so on.
- If an S is used for **#** (**PS** or **DS**), the differences between the base ray and all the parabasal rays are also listed.
- The flux, size, and optical path length of each beam are listed along with the coordinate data.
- The CONSIDER and SELECT commands can be used to restrict the list to rays that are currently on certain objects.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GET
LIST
LIST ELLIPSE
LIST INTEGER
LIST RAYS
PUT

LIST ELLIPSE (ASAP Command)

Lists polarization parameters of currently selected ray data.

Function

Analyze Ray/Beam Data

Syntax

```
LIST ELLIPSE
```

Remarks

- If FRESNEL BOTH is set, prints the major axis orientation (in global coordinates), the ellipticity, and handedness of the polarization ellipse.
- The major axis orientation is printed out in global coordinates. The ellipticity ratio is the ratio of the minor to major axis of the polarization ellipses. A purely linear state has a ratio of zero, and a purely circular state has a ratio of 1. The sign of the ellipticity determines the handedness of the polarization ellipse.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

FRESNEL BOTH
LIST
LIST INTEGER
LIST (Parabasal Ray Data)
LIST RAYS
Polarization Ray Tracing task
POLARIZ
PLOT POLARIZ
GET
PUT

LIST INTEGER (ASAP Command)

Lists ray database information.

Function

Analyze Ray/Beam Data

Syntax

```
LIST INTEGER
```

Remarks

- Creates a list containing the current medium, split level, source number, number of objects, current object and previous starting objects for each ray in the VIRTUAL.PGS file.
- To reduce the amount of output, ASAP only lists the break points where the integer data changes from one ray to the next.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

LIST
LIST ELLIPSE
LIST (Parabasal Ray Data)
LIST RAYS

LIST RAYS (ASAP Command)

Lists ray data in **EMITTING RAYS** format.

Function

Analyze Ray/Beam Data

Syntax

```
LIST RAYS
```

Remarks

- **LIST**s the position, direction, flux, size, and divergence of each ray/beam in a format compatible with the input to the EMITTING RAYS command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

LIST
LIST ELLIPSE
LIST INTEGER
LIST (Parabasal Ray Data)

LOCAL (ASAP Command)

Assigns a local bounding box to the surface/function.

Function

Define/Modify Surfunc Entities

Syntax (short form):

```
LOCAL x x' [ y y' z z' ] [ X [ f ] ]
                           Y
                           Z
                           SHORTEST
                           LONGEST
                           NORMAL
```

Syntax (long form):

```
LOCAL [ OFF      ]
        REPEAT i  [ X [ f ] ]
        STATS       Y
        AXIS        Z
       -X  x        SHORTEST
       -Y  y        LONGEST
       -Z  z        NORMAL
       +X  x'       OFF
       +Y  y'
       +Z  z'
        EXPAND r
               X r
               Y
               Z
```

Option	Description
x x', y y', or z z'	minimum and maximum extents in the given direction
X, Y or Z	coordinate axis
f	fractional inner boundary height
SHORTEST	coordinate axis determined by shortest limit box dimension
LONGEST	coordinate axis determined by longest limit box dimension
NORMAL	coordinate axis determined by coordinate direction
OFF	makes local box a rectangular box nearest surface normal
REPEAT i	copies/repeats the local box from surface i
STATS	resets the local box according to the previous TRACE STATS command
AXIS	resets the local box axis
-X, -Y, -Z, +X, +Y, or +Z	specifies the specific bounding box side
r	relative scale factor

Remarks

- Assigns a local bounding box to the previous surface/function.
- The long form of the command allows you to specifically override or reset any quantity of the local box. For additional functionality use the long form of the command.
- The minimum and maximum extents are defined in local coordinates.
- If the coordinate direction is not entered, the surface is constrained by a rectangular box whose endpoints are given by the minimum and maximum extents.
- If the coordinate direction is entered, the surface is constrained by a cylinder with a constant elliptical cross-section in planes

perpendicular to the given coordinate.

- The **SHORTEST** limit box dimension, the **LONGEST**, or the coordinate direction nearest to the surface **NORMAL** can also determine the axis.
- An optional inner boundary of fractional height **f** [default is zero] can be used to put a proportional hole in the surface in the given coordinate direction. If **f** is negative, the proportional hole is rectangular.
- **LOCAL OFF** temporarily turns off future limits checking.
- The **REPEAT** option to sets the limits on the current surface to those of a previous surface **i**.
- The **STATS** option sets the ranges of ray intersections found from the previous TRACE STATS command (assuming an AXIS LOCAL command is in effect).
- The long form can also be used to reset the limits **AXIS** or just one of the six limits values.
- The **EXPAND** option can be used to enlarge (or shrink) by a relative amount **r** the entire limits box or just in one direction, for example, **EXPAND -.1** shrinks the entire box by 10%.
- After a **LOCAL** command, any linear transformation commands is applied to the surface's local-to-global transformation matrix and not to the surface coefficients themselves.
- The **LOCAL** command can be used for very simple bounding of a surface/function. For more sophisticated boundaries, BOUND surfaces can also be employed on the object.
- Certain surfaces/functions are self-bounding (ELLIPSOID and TORUS, for example) if you use the entire surface/function. To bound surfaces that don't fall into this category, use the **LOCAL** command. The **LOCAL** command creates a rectangular or cylindrical box around the surface effectively defining its physical dimension. The **LOCAL** box is centered on the surface's reference point. The reference points of surfaces are as follows:

Surface	**Reference Point**
PLANE	at intersection of surface and coordinate axis
OPTICAL	at intersection of surface and coordinate axis
ELLIPSOID	at center of ellipsoid
TORUS	at center of torus
TUBE	dependent on sign of symmetric axis qualifier
	if unsigned:midpoint of tube
	+X, +Y, +Z:positive end of tube
	-X, -Y, -Z:negative end of tube

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

LSQFIT (ASAP Command)

Controls the singular value decomposition least squares fitting algorithm.

Function

Save or Recover System Data and Control Execution

Syntax

```
LSQFIT [ t ]  [ LIST ] [ NORM ]
                         OFF
```

Option	Description
t	singular value threshold (default 1.E-5)
LIST	option to print fit parameters
NORM	normalizes variables prior to fit
OFF	does not normalize variables prior to fit

Remarks

- LSQFIT controls the SVD algorithm used by any of the commands that do a least squares fit.
- Any singular values that are "t" (default 3.E-5) times less than the maximum are removed.
- If the **LIST** option is present, then the singular values, variable values, and fit errors are printed.
- Sometimes an improved fit is obtained if the variables are first **NORM**alized. **OFF** sets both these options back to their default states.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

FITTED

HORN

POLYNOMIAL/TRINOMIAL/BINOMIAL. . .FITTED

REVOLUTION

MANGIN (ASAP Command)

Creates a Mangin mirror.

Function

Define/Modify Lens Entities

Syntax

```
MANGIN  X x t h m [ RD r  r ]
        Y y         CV c c'
        Z z         FL f b  [ a     ]
                         a    APLANAT
```

Option	Description
X Y or Z	global coordinate axis
x y or z	location on the global coordinate axis
t	lens thickness
h	aperture height
m	internal medium (number or name)
f	focal length
b	bending parameter
c c'	curvatures of the two surfaces
r r'	radii of curvature of the two surfaces
APLANAT	bending factor and one conic constant are automatically calculated
a	conjugate factor

Remarks

- A Mangin mirror is a lens with a reflective second conicoid; rays are refracted twice at the first conicoid.
- The format of this command is identical to that of the SINGLET, except that the second surface is reflective; that is, rays are refracted twice at the first surface. The following remarks are the same for the SINGLET.
- This lens entity starts out normal to the defined global coordinate axis (X, Y or Z).
- **RD** is used to specify radii of curvature (**r r'**), **CV** is used to specify curvatures (**c c'**), and **FL** is used to specify focal length **f** and bending parameter **b**.
- The bending parameter **b** is defined as (**c**+**c'**)/(**c**-**c'**) or, equivalently, as (**r'**+**r**)/(**r'**-**r**); therefore, **b**=0 implies a biconvex or biconcave element; **b**=-1 implies a plano-convex or plano-concave element; and **b**=1 implies a convex-plano or concave-plano element.
- **a** is an optional conjugate factor; that is, one plus the object-to-image magnification divided by one minus the magnification **(0=one-to-one imaging, 1=infinite object distance, -1=infinite image distance)**
- If the **APLANAT** option is used, the bending factor is automatically determined for the given **a** so that third-order coma is also eliminated (assuming the thin lens approximation applies).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

MAP (ASAP Command)

Produces a depth map of the currently defined object surfaces.

Function

Setup Plots and Verify System

Syntax

```
MAP [ name ] [ DEPTH [ d' [ d"] ] ]  [ SLOPES ]
```

Option	Description
`name`	name of the distribution data file (default extension .DIS)
`d' d"`	range of the third coordinate (the one different than the two specified on the WINDOW command)
`SLOPES`	flag to compute surface derivative data; see example below for optional parameters
`DEPTH`	value of map depth

Remarks

- Produces a depth map of the current object surfaces using the last WINDOW and PIXEL settings.
- The map is saved to distribution data file **name**.dis (default BRO009.DAT) for later processing and can optionally contain not only the depth value but also the two **SLOPES** at each scan point. This file may be examined using the DISPLAY command.
- If the actual range of depth values (`d'` to d" are not given, they are automatically calculated from the current object geometry (and reversed if a DEPTH option is present, but without any numerical entries).
- If **SLOPES** is present, ASAP computes and saves the slopes (derivatives) of the surface at each scan point as well. See the examples below.

Examples for MAP and DISPLAY

To map the surface of an object and save it to a file, specify a WINDOW and PIXEL resolution, and then enter:

MAP REF_MAP SLOPES

Data is now in the file, REF_MAP. However, SLOPES creates three sets of data for each pixel in the window specified: one matrix of Height data, one matrix of the slopes with respect to the first window coordinate, and one matrix of slopes for the second window coordinate. The DISPLAY command with the file name REF_MAP alone would bring in only the first matrix. To read in a particular matrix for viewing or saving to a file, enter one or all of the following:

DISPLAY REF_MAP 1ST !! Reads in the height values for the entire pixel matrix

DISPLAY REF_MAP 2ND !! Reads in the first window coordinate slope values for all pixels (for example, X values for WIN X Y)

DISPLAY REF_MAP 3RD !! Reads in the second window coordinate values for all pixels (for example, Y values for WIN X Y)

After each of the above lines, data may be reviewed in the Command Output window, or saved to a separate file using the WRITE command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
PIXEL
WINDOW

MATRIX (ASAP Command)

Specifies the transformation matrix that operates on the surface general transformation matrix.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities
Define/Modify Lens Entities
Create/Modify Objects
Modify Ray/Beam Data

Syntax (first):

```
MATRIX
x  a  b  c
y  a' b' c'
z  a" b" c" [ LIST ]
```

Syntax (second):

```
MATRIX [ PREVIOUS ]  [ LIST ]
         INVERSE
         k
```

Option	Description
x y z	translation vector
a b c, a' b' c', a" b" c"	rotation submatrix
LIST	decodes transformation matrix into single operation (if possible) and prints
PREVIOUS	uses previously defined transformation matrix
INVERSE	uses inverse of previously defined transformation matrix

Remarks

- Specify a 3x4 (first dummy row excluded) transformation matrix directly.
- **MATRIX PREVIOUS** or **INVERSE** allows you to reuse the transformation matrix from the previous entity, its inverse, or the matrix of SURFACE entity **k** (use zero for inverse of the SURFACE entity currently being transformed).
- When used with REPEAT, group MATRIX with these commands: ROTATE; SHIFT; SCALE; SKEW; PLACE; ALIGN; XEQ.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

MEDIA (ASAP Command)

Creates a single refractive material.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
MEDIA [ m ]
n [ n' n" ... ] [ options ... ] [ 'name' ]
   catalog_glass
   …
     :
```

Option	Description
m	starting media number
n n' n" ...	real (or complex) refractive indices
options...	ABSORB, CRYSTAL, GRIN, or SCATTER
name	descriptive name that can be assigned to this medium (only the first 16 characters after the comment delimiter are stored).

Remarks

- Starting with medium **m**, define media with real (or complex) refractive indices **n** or from its literal designation in a glass catalog (for example, SCHOTT_BK7). Optionally, separate indices can be entered (or computed) that correspond to the wavelengths entered on the last multiple WAVELENGTHS command.
- The default value of **m** is one more than the largest medium number defined and is set to 1 at the start of program execution.
- Linear interpolation is performed, if necessary, to compute a refractive index at a particular wavelength. The interpolated values are then used to calculate absorption coefficients (for complex indices).
- Each medium can have a different interpolation WAVELENGTH.
- More than one of the options (ABSORB, CRYSTAL, GRIN, or SCATTER) may be used in a single MEDIA command. The parameters associated with these options are discussed in the related command topics (see list of commands below, under See Also).
- When specifying more than one of the options (ABSORB, CRYSTAL, GRIN, or SCATTTER), make the step size and the maximum number of steps the same for each option.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Media Database Overview
MEDIA ABSORB
MEDIA CRYSTAL
MEDIA GRIN
MEDIA SCATTER

MEDIA ABSORB (ASAP Command)

Creates an absorbing refractive material.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
MEDIA [ m ]
n [ n' n" ... ] ABSORB a [ j q t [ l ] [ 'name' ]
     :
```

Option	Description
m	starting media number
n n' n" ...	real (or complex) refractive indices
a	absorption (or gain) coefficient (inverse length units); positive for an absorbing medium or negative for a gain (lasing) medium
j	SURFACE designation for a inhomogeneous ABSORB function
q	exponent of inhomogeneous ABSORB function
t	step length to be used during ray trace in an inhomogeneous medium
l	maximum number of ray steps in medium; default is 1000
name	descriptive name that can be assigned to this medium (only the first 16 characters after the comment delimiter are stored).

Remarks

- **ABSORB** identifies the medium as a general absorbing one.
- Refractive indices for dispersive materials must be entered in the order indicated by the previous WAVELENGTHS command.
- If only real refractive index data are entered and a non-zero value is NOT given for **a**, the absorption is set to zero.
- If complex refractive index data are entered and **a** is NOT specified, ASAP calculates **a** (the absorption coefficient) from the wavelength and the imaginary part of the complex refractive index η as follows:

  given $\eta = n + ik$, then $a = 4 \Pi k / \lambda$

  The meaning of the absorption coefficient **a** is given by

  $$I = I_o e^{-aL}$$

  In this equation, **I** is the intensity at a depth **L** in the medium, when the intensity on entering the medium is

  I_o
- To handle inhomogeneous absorption or gain, assign the medium a GENERAL polynomial function in the global coordinates (**X,Y,Z**). The magnitude of **j** is the SURFACE designation for this function. The absorption coefficient at each point in the medium is then calculated as follows:

  given $\eta = n + ik$, then $a = 4 \Pi k / \lambda$
- Linear interpolation is performed, if necessary, to compute a refractive index at a particular wavelength. The interpolated values are then used to calculate absorption coefficients (for complex indices).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Media Overview

MEDIA

MEDIA CRYSTAL

MEDIA GRIN
MEDIA SCATTER

MEDIA CRYSTAL (ASAP Command)

Defines birefringent media.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
MEDIA [ m ]
n  [ n' n" . . . ] CRYSTAL a,b,c [ m' ] [ 'name' ]
      :
```

Option	Description
m	starting media number
n n' n" ...	real (or complex) refractive indices
a,b,c	optical axis direction
m'	ordinary index number or name
name	descriptive name that can be assigned to this medium (only the first 16 characters after the comment delimiter are stored).

Remarks

- The ordinary indices are specified on a previous medium **m'** (default is last media). For example, a calcite crystal with its optical axis initially aligned with the Y-direction is defined as follows:

```
MEDIA
1.66 !!ordinary indices for following extraordinary
1.49 CRYSTAL 0 1 0 'CALCITE'
```

- The above example is expressed in the following way when the ordinary index is not defined on the immediately previous medium:

```
MEDIA
1.66 'ORDINARY' !!ORDINARY INDICES FOR FOLLOWING EXTRAORDINARY
1.60 'GLASS'
1.49 CRYSTAL 0 1 0 ORDINARY 'CALCITE'
```

- If SPLIT is one or higher, ASAP automatically generates an ordinary and extraordinary ray/beam at each crystal interface.
- If you linearly transform an object assigned a birefringent MEDIA, the linear transformation is applied also to the optical axis direction of the medium. Any other object using this medium is therefore affected.
- Refractive indices for dispersive materials must be entered in the order indicated by the previous WAVELENGTHS command.
- Linear interpolation is performed, if necessary, to compute a refractive index at a particular wavelength.
- Accurately handles interface reflectivity/transmissivity values in the most common cases (for example, waveplates). However, the approximate (but fast) algorithm used can produce erroneous values for more exotic situations. (The algorithm will be updated in a future release of ASAP to address this.)

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Anisotropic Polarizing Elements
Media Overview
MEDIA
MEDIA ABSORB
MEDIA GRIN
MEDIA SCATTER

MEDIA GRIN (ASAP Command)

Creates a GRIN refractive material.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
MEDIA [ m ]
n  [ n' n" . . . ]  GRIN k p t [ l ] [ 'name' ]
                :
```

Option	Description
m	starting media number
n	real (or complex) refractive index
a	absorption (or gain) coefficient (inverse length units); positive for an absorbing medium or negative for a gain (lasing) medium
k	GRIN function number
p	exponent of GRIN function
t	step length to be used during ray trace
l	maximum number of ray steps in medium; default is 1000
name	descriptive name that can be assigned to this medium (only the first 16 characters after the comment delimiter are stored).

Remarks

- **GRIN** specifies that the medium consists of Gradient index (GRIN) materials. The square of the refractive index is given by:

  $$n^2 \quad ;x,y,z \quad n^2 \quad f^p \quad X,Y,Z$$

  where

  $$f^p \quad X,Y,Z$$

  is a general polynomial function and

  $$n^2$$

  is the refractive index entered on the MEDIA command. The technique for modeling dispersive gradients is to define a separate GRIN function for each WAVELENGTH and to update the object data between individual ray traces.
- The **k** refers to the SURFACE/FUNCTION that defines the index variation.
- If the constant coefficient of the function

  $$f^p \quad x,y,z$$

- is unity, the refractive indices entered after the MEDIA command corresponds to those at the function's reference point.
- The **t** is the step length in systems units used by ASAP while tracing a ray in this inhomogeneous medium.
- The **GRIN** function is defined in global coordinates. This is of little consequence if the gradients are radial (there is no position dependence along the axis), but if the desired gradient is axial, you may have to SHIFT the gradient to the correct global coordinates to align the gradient with the object.
- The function can be ARRAYed.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRIN Overview
Media Overview
MEDIA ABSORB
MEDIA CRYSTAL

MEDIA SCATTER (ASAP Command)

Assigns to a medium a GENERAL polynomial or USERFUNC function in the global coordinates X, Y, Z.

Syntax

```
MEDIA   [ m ]
n  [ n' n" . . . ] SCATTER m k e t [ l ]  ]  [  'name' ]
              :
```

Option	Description
m	refers to a VOLUME scattering MODEL
k	its magnitude is the SURFACE designation for this function
t	step length used for tracing a ray when tracing a ray in this inhomogeneous medium

Remarks

- This inhomogeneous Monte-Carlo scattering option can be handled by assigning to the medium a GENERAL polynomial or USERFUNC function in the global coordinates X, Y, Z.
- The magnitude of **k** is the SURFACE designation for this function.
- The scattering at each point in the medium is then multiplied by:

$$f^{e}\left(\left[w;\right]X,Y,Z\right)$$

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Media Database Overview
MEDIA
MEDIA ABSORB
MEDIA CRYSTAL
MEDIA GRIN

MESH (ASAP Command)

Creates a 3-D representation of the distribution in the system vector (*.VCR) file.

Function

Display/Modify Energy Distributions

Syntax

```
MESH [ i  [ j ]  ] [ LOW  ]
                     HIGH
```

Option	Description
i	line across specification
j	line down specification

Remarks

- Writes a three-dimensional representation (wireframe or **LOW** / **HIGH**-resolution shaded "surfaces") of the distribution into the 3-D system vector file so that it can be REPLOTted with other geometrical entities.
- If given, only every **i**th line across or **j**th line down is written (defaults are 1).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
REPLOT

MINIMIZE (ASAP Command)

Minimizes the maximum RMS spot size or the average of the previously specified fields, weighted by the "**w**"s.

Function

Define/Modify Lens Entities

Syntax

```
MINIMIZE [ w w'...] [ control c ] [ control c' ] ...
```

Remarks

Control	Default	Description
DIST	1 (100%)	Maximum allowed fractional spot-centroid distortion
GLTH	0	Minimum center and/or edge glass thickness
	infinite	Maximum center and/or edge glass thickness
TLEN	infinite	Maximum total system length (first to last conicoid)
	1	Optional starting conicoid
	last	Optional ending conicoid
SDIA	infinite	Maximum unvignetted semi-diameter of a conicoid set
	1	Optional starting conicoid
	last	Optional ending conicoid
BKFD	0	Minimum back focal distance (last conicoid to focus)
BKWD	-Infinite	Minimum back working distance
UBAR	1	Maximum magnitude of final chief ray slope

If the image quality of a design does not get better during an optimization, it is usually due to conflicting constraint violations or constraint violations not affected by the specified variables.

- Normal optimization controls:

Control	Default	Description
MULT	1 (local)	Number of multiple trial solutions (large for global)
SEED	987654321	Random (large odd integer) or quasi-random (zero) seed
TARG	1.E-9	Take first solution that drops below this RMS spot size

- Advanced optimization controls:

Control	Default	Description
TOLR	.0001	Fractional change in merit function at local minima
DELV	.001-1	Maximum allowable randomization of normalized variables
ITER	10	Maximum allowable randomization of normalized variables

These advanced controls are automatically determined by the **MULT** value and the number of variables. In rare circumstances, they may have to be set explicitly.

- The underlying design engine actually has a comprehensive "pickup" (a conicoid variable is constrained to follow one at a previous conicoid), and "paraxial solve" (a conicoid variable is determined from paraxial ray-trace requirements) capability. These are currently hidden but are used in the following circumstances:

1. The focal length of the lens is always held by a paraxial marginal ray solve for the curvature of the last conicoid and, sometimes, a previous one coupled to it (see #3 and #5).

2. The nominal focal plane position is found from a paraxial marginal ray solve for the back focal distance that results in a zero ray height.

3. If the starting design is symmetric and only the conicoid parameters for the front half are varied, the lens will automatically remain symmetric during optimization.

4. If more than one element references the same glass and only the first instance of that glass is varied, all other instances follow automatically.

5. If an element is actually a reflector, where the surface before and after it are geometrically the same, and even though, internally, two conicoids are used to represent the duplicate surface, the second instance will always replicate any changes in the first. A Mangin back-surface mirror and a Maksutov telescope (where the secondary is an aluminized spot on the back of the meniscus corrector) are common examples of this condition.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Optimizing a Centered Imaging System

MIRROR (ASAP Command)

Creates a simple mirror.

Function

Define/Modify Lens Entities

Syntax

```
MIRROR  X x h [ f k o ]
        Y y
        Z z
```

Option	Description
X Y or Z	global coordinate axis
x y or z	location on the global coordinate axis
h	aperture height of mirror
f	focal length (zero for a flat)
k	conic constant (for example, 0=sphere, -1=parabola)
o	central hole ratio

Remarks

- The focal length of a concave mirror is positive; the focal length of a convex mirror is negative.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

MISSED (ASAP Command)

Sets how a missed ray is plotted.

Function

Setup Trace

Syntax

```
MISSED [ OFF   ]  [ d ]
         ARROW
         LINE
```

Option	Description
OFF	does not draw missed ray
d	distance for extending rays in 3D
ARROW	draws arrow
LINE	draws line

Remarks

- Determines what type of vector is drawn during ray plotting to indicate a missed ray. Normally, the missed rays are extended only to the edge of the 2D plotting window.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

MODELS (ASAP Command)

Creates a scattering model.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
MODELS [ i [ PLOT [ a a'...] ] ]
```

Option	Description
`i`	beginning model number
`PLOT`	plots the model in log (b-bo) and angle space
`a a' ...`	user-defined degree specular angles

Remarks

- Defines a specific scatter model from a general set of BSDF scattering models.
- Begin defining scatter **MODELS** at number **i** (the default model number is one more than the highest defined).
- Similar to the MEDIA and the COATING commands, it constructs a database for use in the construction of an OBJECT.
- Creates a distribution file **name_angle.dis** for each of these angles.
- Both the **r** and **t** values on the INTERFACE command must be non-zero.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

...PLOT (ASAP Command Argument)
PARTICLES (ASAP Command) - VOLUME option
BSDF Fit Utility
RMS
COATING PROPERTIES
MODELS Overview
NONLINEAR

MODIFY (ASAP Command)

Alters current distribution data.

Function

Display/Modify Energy Distributions

Syntax

```
MODIFY [ m m' n n' [ a b ] ] [ 'flabel' ]
                     fcn
[ m m' n n' [ a b ] ]
              fcn
   :
```

Option	Description
m m'	integer pixel ranges in the across direction
n n'	integer pixel ranges in the down direction
a b	scale factors
fcn	internally or user-defined function
'flabel'	optional text to relabel the functional data

Remarks

- Modifies the data region specified by the two integer pixel ranges, **m** to **m'** (across) and **n** to **n'** (down). The data in that region is replaced by a **a** plus **b** times the data value, that is:

$$f'(i,j) = a + b\, f(i,j) \qquad i = m, m' \quad j = n, n'$$

- Alternatively, an intrinsic or extrinsic function may be used to modify the data according to the equation:

$$f'(i,j) = fcn(f(i,j)) \qquad i = m, n' \quad j = n, n'$$

- The function name **fcn** is either internal or user-defined via $FCN.
- More complex modifications can be done with multiple commands.
- Use the **'flabel'** option to relabel the functional data.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
REDUCE

MOVE (ASAP Command)

Moves rays along their propagation direction to a new location.

Function

Modify Ray/Beam Data

Syntax

```
MOVE BY d
     TO X d
        Y
        Z
        OPL
```

Option	Description
BY	relative move
TO	absolute move
d	distance along a given coordinate direction
OPL	optical path length

Remarks

- Moves the ray data to a new reference position either **BY** an amount **d** (relative move) or **TO** a position **d** (absolute move) along the individual ray directions, given coordinate direction, or Optical Path Length (OPL).
- During the **MOVE** operation, the rays are projected along their direction vectors. The rays do **not** intersect any **OBJECTS** along the way.
- Commonly used for virtual ray tracing from a buried entrance pupil to object space so that the beam may then be traced through the system.
- Use to observe the effects of defocus on a beam.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MOVE PARABASALS
MOVE TO FOCI
MOVE TO PLANE
MOVE TO POINT
MOVE TO SPHERE

MOVE PARABASALS (ASAP Command)

Moves the parabasal rays for each ray/beam to its base ray plane.

Function

Modify Ray/Beam Data

Syntax

```
MOVE PARABASALS
```

Remarks

- Moves parabasals to a plane perpendicular to the base point. It does not in any way change the beam characteristics, but does affect the PLOT BEAMS display since, by default, the parabasals lie on the current object surface.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PARABASAL
MOVE
MOVE TO FOCI
MOVE TO PLANE
MOVE TO POINT
MOVE TO SPHERE
DISPLAY

MOVE TO FOCI (ASAP Command)

Moves rays or beams along the propagation direction to a focus with the given parabasal rays and the base ray.

Function

Modify Ray/Beam Data

Syntax

```
MOVE TO FOCI [ n n' n" ... ]
```

Option	Description
`n n' n" ...`	specified parabasal rays

Remarks

- Moves each ray/beam to the centroid of the intersections of the given parabasal rays with the base ray.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MOVE
MOVE PARABASALS
MOVE TO PLANE
MOVE TO POINT
MOVE TO SPHERE

MOVE TO PLANE (ASAP Command)

Moves rays along their propagation direction to a specified reference plane.

Function

Modify Ray/Beam Data

Syntax

```
MOVE TO PLANE [ x y z ] [ a,b,c ]
```

Option	Description
`x y z`	global coordinate of a reference plane
`a,b,c`	global direction vector of the surface normal of the plane

Remarks

- Transfers the current ray/beam set to a plane with normal (**a,b,c**) through the point (**x,y,z**).
- The default point is the average of the current base ray coordinates.
- The default for the normal to the plane is the average of the current base ray directions.
- The peak-to-valley optical path difference on this surface is also printed so that normally it represents the reference plane of an afocal system.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MOVE
MOVE TO FOCI
MOVE PARABASALS
MOVE TO POINT
MOVE TO SPHERE

MOVE TO POINT (ASAP Command)

Moves rays along their trajectories to their closest approach to the given point (**x y z**).

Function

Modify Ray/Beam Data

Syntax

```
MOVE TO POINT x y z
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MOVE
MOVE PARABASALS
MOVE TO FOCI
MOVE TO PLANE
MOVE TO SPHERE

MOVE TO SPHERE (ASAP Command)

Moves rays along their propagation direction to a specified different sphere.

Function

Modify Ray/Beam Data

Syntax

```
MOVE TO SPHERE r [ x y z ]
```

Option	Description
`r`	radius of reference sphere
`x y z`	coordinate of center of reference sphere

Remarks

- Transfers the current ray/beam set to the surface of a sphere of radius **r** centered on the point (**x,y,z**) or a PLANE with normal (a,b,c) through point (x y z) by using the variation of the MOVE command.
- The default point is the average of the current base ray coordinates.
- The peak-to-valley optical path difference on this surface is also printed so that normally it represents the reference sphere of a focusing system.
- If none of the SPHERE data is given, a best-fit spherical wavefront is used.
- The default for the normal to the PLANE is the average of the current base ray directions.
- The default point is the average of the current base ray coordinates.
- The Peak-to-Valley and RMS optical path difference on this surface are also printed so that normally it represents either the reference sphere of a focussing system or the reference plane of an afocal system.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MOVE
MOVE PARABASALS
MOVE TO FOCI
MOVE TO PLANE
MOVE TO POINT

MULTIPLE (ASAP Command)

Converts the surface into multiple sheets.

Function

Define/Modify Surfunc Entities
Create/Modify Objects

Syntax

```
MULTIPLE n f'      [ EXPONENT p ]
         d x y z
```

Option	Description
n	number of sheets to be generated
f'	additive constant to the original function
d	distance between original and first sheets
x y z	an arbitrary point on the original surface
EXPONENT p	exponent to which sheet number is raised

Remarks

- The surface is converted into multiple parallel surfaces. In other words, the equation of the surfaces becomes:

$$\mathrm{f}(\mathrm{X},\mathrm{Y},\mathrm{Z}) = j^{p} f' \quad j = 0, n$$

- The zeroth sheet is the original surface.
- Alternatively, ASAP can calculate **f'** such that the distance from a point (**x,y,z**) on the original surface to the first sheet is **d**.
- The exponent **p** is defaulted to 1, but can be used, for example, to get evenly spaced cylinders or spheres (p=2).
- If **MULTIPLE** is being used to define a diffraction grating, then the value of **n** is irrelevant.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

NONLINEAR (ASAP Command)

Creates a scatter model based on the combination of the Phong and Harvey models.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax (isotropic):

First Syntax (polynomial coefficients):

```
NONLINEAR a1 b1 c1 d1 e1 [ a2' b2' c2' d2' e2' [ a3" b3" c3" d3" e3" ...] ] ] [ PLOT [ a a'
...] ]
```

Syntax (anisotropic):

```
NONLINEAR X p q a b c d e [ p' q' a' b' c' d' e' ... ] ]
          Y
          Z
          :
```

Syntax (fitting BSDF values):

```
NONLINEAR FIT [ n       ] [ options ]
          FRAC   e e' ...
  data…
  :
```

Option	Description
a b c d e ...	BSDF equation coefficients
PLOT	plots the model in log(b-b$_o$) and angle space
a a' ...	user-defined degree specular angles
FIT	FIT the BSDF values to the model
n (or optionally e e' ...)	**n** is the number of 5-parameter terms (default 5); the **e**'s are the starting guesses for the exponents of each term
ANGLES	specifies spherical angle coordinates
LOG	specifies common logarithmic BSDF values

Remarks

The following remarks apply to both isotropic and anisotropic versions unless otherwise noted.

- Generalizes the combination of the Harvey (sharp peak) and Phong (broad peak) models and as such is applicable to both smooth and rough surfaces.
- Isotropic: The command is defined by the following relatively simple formula:

  $$\mathrm{BSDF} = \sum_{i=1}^{N} \left\{ a_i (U+W) + b_i V + c_i \sqrt{(1-U)(1-W)} + d_i \right\}^{e_i}$$

  where from the previous definition of the isotropic-surface direction cosine variables:

  $$U + W = \sin^2 \mathrm{F} + \sin^2 \mathrm{F}_0$$

  $$V = \sin\mathrm{F} \sin\mathrm{F}_0 \cos\mathrm{R}$$

  $$\sqrt{(1 - \mathrm{U})(1-W)} = \cos\mathrm{F} \cos\mathrm{F}_0$$

  F = Scatter angle from normal

  Fo = Specular angle from normal

  R = Angle around normal
- Anisotropic: The command assumes the surface anisotropy is aligned with the local Alpha or Beta direction and is defined by the following relatively simple formula:

$$\text{BSDF}=\sum_{i=1}^{N}\left\{p_i\left(A^2+Ao^2\right)+q_i\left(B^2+Bo^2\right)+a_iAo+b_iBBo+c_iCCo+d_i\right\}^{e_i}$$

where from the previous definition of the anisotropic-surface direction cosine variables:

$$c^2=1-A^2-B^2=\cos^2 F \qquad \text{F = Scatter angle from normal}$$

$$\text{Co}^2=1-Ao^2-Bo^2=\cos^2 Fo \quad \text{Fo = Specular angle from normal}$$

- If the quantity in braces {} is less than zero, then the term is set to zero.
- The **e** exponents do not have to be integer or positive.
- Note that the resulting BSDF is guaranteed to have all the correct positivity, symmetry, and reciprocity properties.
- Scattering from anisotropic surfaces is not rotationally symmetric at normal incidence and not necessarily symmetric about the plane of incidence otherwise. Therefore, the orientation of the model on the surface is important and is generally specified by an axis for the second command entry. For syntax information, see ...MODEL... (ASAP Command Argument).
- Isotropic: The total number of parameters must be less than or equal to 285, that is, **N** less than or equal to 57 quintuples (five or less terms is usually sufficient for most surfaces). The specific cases of this model are:

Forward Harvey	a>>1	b=-2a	c=0	d>0	e<0
Retro Harvey	a>>1	b=2a	c=0	d>0	e<0
Lambertian	a=0	b=0	c=0	d>0	e=1
Forward Phong	a=0	b=c	c>0	d=0	e>1
Retro Phong	a=0	b= -c	c>0	d=0	e>1

- Anisotropic: The total number of parameters must be less than or equal to 40 terms (five or less terms are usually sufficient for most surfaces).

  NOTE: When the **p**s equal the **q**s and the **a**s equal the **b**s for each term, this model reduces to the isotropic-surface version.
- Isotropic: The user can optionally fit BSDF values to the above model by using the **NONLINEAR FIT** command where **n** is the number of five-parameter terms (default 3) or optionally the **e**s are the starting guesses for the exponents of each term.
- Anisotropic: The user can optionally fit BSDF values to the above model by entering the data on successive commands:

```
NONLINEAR FIT [ n ] [ options ... ]
          FRAC e e' ...
data ...
:
```

- Anisotropic: **n** is the number of 7-parameter terms (default 5) or optionally the "e"s are the starting guesses for the exponents of each term. If the given data does not cover most of the input and output hemispheres, then the fitted model can do unexpected things in the missing regions; for example, have a TIS greater than one.
- Since the **NONLINEAR** model is not defined in logarithm space (like the POLYNOMIAL model), the **FIT** may have a tough time accurately reproducing any BSDF with a high dynamic range. Optionally, the **FRAC**tional error at each data point can be used instead of the absolute error. This has about the same effect as fitting in logarithm space. In either case, the fit is done using an iterative non-linear damped least-squares algorithm. Therefore, it converges to one of the local minima and not necessarily the best one. An off-line, time-consuming, global optimization technique (for example, simulated annealing) could be used if the data only needs to be fit once.
- The **PLOT** option plots the model (common base 10 logarithm of the BSDF) for up to seven specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). The current PIXELS setting controls the resolution of these plots in direction cosine space, and it creates a distribution file **name_angle.dis** for each of these angles.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MODELS Overview
MODELS
HARVEY
LAMBERTIAN
USERBSDF

NORMALIZE (ASAP Command)

Renormalizes the current distribution data.

Function

Display/Modify Energy Distributions

Syntax

```
NORMALIZE [  d  [ c [ c' ]  ]  ]  [ 'flabel' ]
            MAX  MAX
```

Option	Description
d	data renormalization value
MAX	maximum distribution data value
c c'	coordinate normalization factors
MAX	maximum coordinate value
flabel	new label for functional data

Remarks

- Renormalizes distribution data (divides functional values) by the value of the last **NORMALIZE** command (default).
- Alternatively, the data may be normalized by the given value **d** or the **MAX**imum value in the distribution data.
- The coordinate ranges can also be normalized in a similar manner by the optional third and fourth entries (**c** and **c'**). The range of the vertical axis scales up by 1/c, and the range of the horizontal axis scales up by 1/c'. If only **c** is given, both axes scale up by 1/c.
- Use the **flabel** option to relabel the functional data.
- Different scale factors can be used for the two coordinate ranges.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

NUMBERS (ASAP Command)

Displays current valid media, surfaces, edges, lenses, and object members.

Function

Setup Plots and Verify System

Syntax

```
NUMBERS [ NAMES  ]
          SUMMARY
```

Option	Description
NAMES	lists a table of numbers and names for all media, coatings and objects
SUMMARY	lists a short summary of storage usage

Remarks

- Displays either a short SUMMARY of storage usage or a table of numbers and NAMES of the current valid media, coatings, surfaces, edges, lenses, and object members.
- Unused numbers appear as blank areas in the tables.
- Temporarily disabled objects (via the CONSIDER command) are shown as negative numbers.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

OBJECT (ASAP Command)

Defines an object based upon previously defined entities.

Function

Create/Modify Objects

Syntax (define a simple object using explicit entities)**:**

```
OBJECT [ j ] [ 'name' ]
OBJECTS [ j ]
  i [ i' ] [ 'name' ]
   :
```

Syntax (use multiple surfaces or edges to create a solid or mesh object):

```
OBJECT;  i   [ q  i' q' ... ]
        -n q [ q' ... ] q"
```

Option	Description
j	starting object number
name	name of the object
i i' ...	entity numbers
q q' q" ...	connection factors
n	last n entities

Remarks

- The first syntax defines a new OBJECT using either the last entity or begins defining (or redefining) OBJECTS starting with number **j**.
- The default value for **j** is one more than the largest object number defined and is set to one at the start of program execution.
- For the first syntax, the **i** is the surface, edge, or lens number that defines the geometry of the object. Simple mesh objects can be formed by one or (if **i'** is also given) two edges. The edges used to define a simple mesh object must have the same number of points. If the second edge is not entered, ASAP defines the object as a planar surface bounded by the dimensions of the first edge.
- The **name** is a descriptive designator that the user can assign to the object.
- For the second syntax, if the **i**'s refer to EDGES, a two-dimensional mesh is formed from these edges with the edge-to-edge connection specified by the **q**'s (See the POINTS commands for definitions but note that the inter-curve Bezier degree cannot be greater than 2). The edges in the mesh should be similar, that is have the same number of points and point-to-point connection factors.
- Alternatively, with the second syntax, the last **n** edges may be used to form a mesh with **q** the connection factor for odd edges, **q'** the even, and **q"** the last to first. The edges in the mesh should be similar; that is, have the same number of segments.
- For the second syntax, if the **i**'s refers to SURFACES, then a solid is formed by bounding each surface with the others. The signs of the **q**'s determine the proper side of each surface (See the BOUNDS command). A zero **q** means the surface is not to be used to bound the others in the object. Note that each surface's LOCAL box (if defined) always clips it.
- The default INTERFACE for surfaces and edges is totally absorbing. The default INTERFACE for a lens object is that refractive conicoids are 100 percent transmitting and reflective conicoids are 100 percent reflecting; the lens itself is surrounded by air.
- The FRESNEL, SPLIT, and LEVEL commands can be different for every object.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EDGES
POINTS (3-D)
POINTS (2D)
SURFACES

OBLIQUE (ASAP Command)

Toggles future graphical output between normal (orthogonal) projection and oblique (nonorthogonal) projection.

Function

Setup Plots and Verify System

Syntax

```
OBLIQUE [  f  ] [ a a' ]
          OFF
```

Option	Description
f	third depth coordinate pivot point
a	rotation angle (in degrees) about the vertical axis
a'	rotation angle (in degrees) about the horizontal axis
OFF	oblique on/off toggle

Remarks

- When active, all future plots are an oblique type projection pivoted about **f**.
- If autoscaling is in effect, **f** is automatically determined by ASAP; otherwise **f**=0.
- The normal orthogonal projections are restored by **OBLIQUE OFF** (the default at program startup).
- The defaults for **a** and **a'** are 45 and 30 degrees, respectively.
- Setting both **a** and **a'** equal to zero effectively turns **OBLIQUE** off.
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X** or **WINDOW Z -Y** to display the system in a right-handed coordinate system.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
FIELD
FMAP
MAP
OPDMAP
PIXELS
PLOT
PROFILES
RADIANT
REPLOT
SPOTS
SPREAD
WINDOW

OFFSET (ASAP Command)

Shifts the origin of the current distribution data.

Function

Display/Modify Energy Distributions

Syntax

```
OFFSET [ v [ h ] ]
```

Option	Description
`v h`	coordinates of the new origin

Remarks

- Shift the origin of the distribution to the given actual coordinates.
- Default is to make the new origin the center of the data set.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

OPDMAP (ASAP Command)

Creates a geometric wavefront map of the currently selected ray data.

Function

Analyze Ray/Beam Data

Syntax

```
OPDMAP [ name ]  [ d ]  [ PSF  [ f ] ]
```

Option	Description
`name`	name of the distribution file (default extension *.dis)
`d`	specifies the sampling performed (see Remarks below)
`PSF`	flag for determination of point spread function

Remarks

- Produces a geometric wavefront map of the ray data as contained within the plotting window specified by the last WINDOW command. The wavefront map is stored in **name.dis** or the default file BRO009.DAT. The data can then be manipulated and plotted using the DISPLAY commands.
- The geometric wavefront map is determined by interpolating the optical path length data (of the first source encountered in the current ray set). In the process, the average OPD value (piston) is removed. You may remove focus and tilt errors by preceding OPDMAP with a FOCUS MOVE or **MOVE TO** command.
- If **d** is entered as zero, ASAP generates a fast discrete map of the wavefront with no interpolation.
- Otherwise, ASAP uses a very general (and somewhat slower) linear interpolation (no extrapolation) technique that considers only a small set of ray points within a distance **d** (default is 3 times smallest ray separation) of each grid point.
- With the PSF option, ASAP forms the complex geometrical pupil function from the interpolated OPD map and stores its Fourier Transform (that is, the coherent PSF) on unit 29 (BRO029.DAT). The **f** controls how much of the PSF to return. If it is less than one, it is a fractional energy (relative to the maximum) cutoff. If greater than one, it is the maximum distance in Airy units (default 5). This PSF is normalized relative to an ideal circular aperture of equivalent area. A **DISPLAY 29 ENERGY** command can then be used to manipulate and plot this far-field distribution.
- **OPDMAP** should not be used at focus or within a caustic; the calculation is not relevant there. Instead, use the FOCUS MOVE or MOVE TO SPHERE commands to move the ray data to an appropriate reference sphere (located well away from focus, the exit pupil is a traditional location) before issuing the **OPDMAP** command.
- The ...CLIP command argument can be used to specify an object (**i**) or edge number (**j**) whose bounds and limits clips the distribution. If **i** is not given, it is defaulted to the current object number of the first valid ray. If **j** is negative the interior of the closed edge is used, if **j** is positive, the exterior of the closed edge is used.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

...CLIP
DISPLAY
MOVE TO FOCI
MOVE TO PLANE
MOVE TO POINT
WINDOW

OPTICAL (ASAP Command)

Creates aspheric surfaces normal to the axis of symmetry.

Function

Define/Modify Surfunc Entities

Syntax

```
OPTICAL X x r[`p ][ c d e f g h i j k l ] [EXPAND] [aperture ... ]
        Y y
        Z z
[ t r[`p ]  [ c d e f g h i j k l ] [EXPAND]]
```

Option	Description
X, Y or Z	axis of symmetry
x, y or z	location along coordinate axis
r	vertex radius of curvature
p	vertex radius of curvature of a parabolic term; optionally subtracted from base surface
c	conic constant
d, e, ..., l	4th, 6th, ..., 20th-order aspheric deformation coefficients
EXPAND	flag to expand conic into aspheric terms
aperture	ELLIPSE, RECTANGLE, or HEXAGONAL
t	relative distance of second optical surface

Reference Point

At intersection of surface and coordinate axis.

Surface Normal

Along positive coordinate direction.

Remarks

- Creates a classical optical surface normal to the axis of symmetry at a value given by the third entry. This is the command of choice for making simple conic or aspheric surfaces in ASAP.
- The second entry designates the axis of symmetry (either **X**, **Y**, or **Z**) for the surface.
- The vertex radius of curvature **r** is negative if the center of curvature is on the negative side of the surface. A zero or very large **r** corresponds to a planar flat.
- OPTICAL surfaces have trouble faceting when the slope at the boundary becomes infinite, such as the edge of a hemisphere. Using an ELLIPSE aperture that is slightly smaller than the boundary radius minimizes these problems.

  Example: A hemispherical dome with radius 100 is defined by:

  SURFACE; OPTICAL axis Z z 0 radius 100 conic 0 ELLIPSE 99.999
- **c** is the conic constant (for example, 0 is a sphere, -1 is a parabola, and so on.).
- **d** is the 4th-order aspheric deformation coefficient, **e** is the 6th-order, and so on up to **l** the 20th.
- This surface/function is stored in order doubled mode.
- ASAP models the surface function exactly up to the 10th-order, or if the base surface is parabolic (conic constant = -1) up to the 20th-order.
- If the base surface is not parabolic and exceeds the 10th-order, the surface is approximated by a truncated power series. If **a** is specified, a table of sag points for user verification is printed to verify accuracy.
- The sag of the surface as a function of the radial coordinate ρ is given by the following equation:

$$\mathrm{sag}(\rho) = \frac{\rho^2 / r}{1 + \sqrt{1 - (1+c)\left(\rho / r\right)^2}} + d\rho^4 + e\rho^6 + f\rho^8 + g\rho^{10} + h\rho^{12} \ldots l\rho^{20}$$

- At the vertex of the surface the normal vector points along the positive coordinate direction.
- If the highest order aspheric entry is the literal **EXPAND**, the conic is expanded into aspheric components up to that order.

- A second line immediately following the OPTICAL command can be used to specify an optional second surface a distance **t** from the first surface. This inclusion permits modeling of the front and back surfaces of a singlet lens with a single entity.
- This surface can extend to infinity unless a LOCAL command follows, or a trailing aperture option of the following form is specified:

```
ELLIPSE   a [ a' [ o [ s [ s' ] ] ] ]
RECTANGLE
HEXOGONAL a [ o [ s [ s' ] ] ]
```

- **a a'** are the heights in the other two transverse directions.
- For the **HEXAGONAL** form, **a** is the center-to-vertex distance (maximum height).
- **o** is an optional central hole ratio.
- **s s'** are the transverse coordinates of the center of the aperture.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

OVAL (ASAP Command)

Creates a polygonal edge.

Function

Define/Modify Curvedge Entities

Syntax

```
OVAL X x y z q [ n a a' ]
     Y y z x
     Z z x y
```

Option	Description
X, Y or Z	specifies the axis of symmetry
x, y or z	location along coordinate axis
y z (z x or x y)	semimajor widths of the oval
q	parameter controlling type of closed curve
n	number of points (or segments) on the oval (default 16)
a a'	angular range (in degrees from first semimajor axis) over which the oval is defined (default is 0 to 360 degrees)

Remarks

- Defines a polygonal edge that can continuously vary between an ellipse (q=0) and a rectangle (q=1).
- The semimajor widths are measured to the points, not to the lines connecting the points.
- If **n**, **a** and **a'** are specified, they become the default settings for most future EDGE commands.
- This edge is made up of coplanar straight line segments, that is, convex polygons whose vertices lie on a particular curve.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

ELLIPSE for descriptions of other parameters.

PARABASAL (ASAP Command)

Sets the number of parabasal rays.

Function

Setup Beam Creation

Syntax

```
PARABASAL n [ d  [ d' ] ] [ CLIP ]
[ g  h  u  v  ]
[ g' h' u' v' ]
[ g" h" u" v" ]
        :
```

Option	Description
n	number of parabasal rays
d d'	semidivergences of each geometric incoherent beam (radians)
g h u v, g' h' u' v' ...	see Remarks
CLIP	flag to check for parabasal ray clipping

Remarks

- Sets the number of parabasal rays to be traced around each base ray and should precede any ray definition commands (**GRID,** RAYSET and others).
- The parabasal rays follow the same path through the system as the primary ray and are never clipped by a boundary unless the primary ray is also clipped.
- The **n** usually takes on values of 0, 2, 4 or 8.
- If **n** equals four or eight and a WAVELENGTH has been specified, the parabasal rays are used to define general Gaussian beams centered on each base ray.
- The **d** and **d'** set the defaults for the semidivergences of each geometrical incoherent beam in radians. This value is only used if **n**=0 or 4, and the current **WAVELENGTH** is zero, that is, a geometrical optics radiant beam.
- The **CLIP** option signals ASAP to check for clipping of the parabasal rays by boundaries and reduce the beam flux accordingly. The parabasal rays are not actually clipped from the beam, however.
- Parabasal rays are numbered according to the following scheme (where W**i** refers to a waist ray displaced in the positive **i**th direction and D**i** refers to a divergence ray propagating in the positive **i**th direction):

Parabasal Ray Number

n	1	2	3	4	5	6	7	8
1	Wx							
2	Wx	Wy						
3	Wx	Wy	Dx					
4	Wx	Wy	Dx	Dy				
5	Wx	Wy	Dx	Dy	-Wx			
6	Wx	Wy	Dx	Dy	-Wx	-Wy		
7	Wx	Wy	Dx	Dy	-Wx	-Wy	-Dx	
8	Wx	Wy	Dx	Dy	-Wx	-Wy	-Dx	-Dy

- The divergence angle, ***u*,** of the parabasal rays is calculated from the equation:

  $$\tan(u) = \lambda/(4h)$$

  where λ is the WAVELENGTH of the beam and h is the waist semidiameter of the Gaussian beam. This definition was chosen so that the width of the Gaussian beam corresponds to

  e^(-pi /4) or roughly the 45 percent amplitude point.
- Normally, depending upon the properties of the beams being created, the transverse positions and direction of the parabasal rays are automatically set by ASAP. However, an override table can be entered, one line per parabasal ray. The **g**'s and **h**'s are the relative heights in the two orthogonal directions. The **u**'s and **v**'s are the orthogonal convergences/divergences angles in radians.
- The WIDTHS command modifies the default parabasal ray settings, scaling the width of the parabasal rays.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MOVE PARABASALS
GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW
Parabasal Rays
RAYSET
WAVELENGTH
WIDTH

PARAMETERIZE (ASAP Command)

Sets the parameterization axis for meshing the current surface.

Function

Define/Modify Surfunc Entities

Syntax

```
PARAMETERIZE -X
             -Y
             -Z
             +X
             +Y
             +Z
```

Remarks

- PLOT/FACETS command can be used on virtually all SURFACE/OBJECTS to produce a wire-frame plot of the OBJECT. In some situations, the inherent parameterization of the polynomial fails. The PARAMETERIZE command re-parameterizes the plot.
- Sets the parameterization of the current surface to be either parallel (-) or perpendicular (+) to the given local axis.
- Tells ASAP how to mesh the surface (for PLOT SURFACE, PLOT MESH, PLOT FACETS, VUFACETS, and GRID OBJECT or ENTITY, EMIT OBJECT or ENTITY commands), but requires that the surface have a LOCAL box defined either implicitly or explicitly. Since this is automatically set for most surfaces, it is rarely necessary to override with the PARAMETERIZE command.
- If you do override the command, remember that ASAP cannot mesh (facet) the surface if the LOCAL box clips the surface along a parameterization direction. However, do not set the LOCAL box too large in this direction since this can also prevent the meshing or faceting of the surface.
- This command and whether a surface can be meshed (facetted) has absolutely no effect on tracing rays/beams to or through the surface.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT
EMIT OBJECT or ENTITY
GRID OBJECT or ENTITY
LOCAL
PLOT SURFACE
PLOT MESH
PLOT FACETS
VUFACETS

PARTICLES (ASAP Command)

Creates a particulate scatter model.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
PARTICLES g q[`p] f
VOLUME    MIE m a a' f  [  fcn n c c'  …  ]
          r    s
```

Option	Description
g	directional factor between -1 (perfect backscatter) to 1 (perfect forward scatter) exclusive
q \`p	scattering and absorption efficiency per particle (typically near or less than one)
MIE	flag for exact MIE calculation
m	MEDIA number or name
a a'	particle radii (in wavelength units)
f	overall fractional area obscuration at a normal incidence
r	intrinsic surface reflectivity
s	RMS surface roughness

Remarks

- The surface/volume scatters as if it had a uniform random distribution of particles on/in it. The three-entry short form is based on the simple Henyey-Greenstein model (see option **g**).
- A **g** value of zero corresponds to isotropic scatter.
- The **\`p** allows for optional entry of absorption efficiency per particle. If used, it should immediately follow the scatter efficiency value **q** with only a backward tick to separate the two numbers. Either number is typically near or less than one. Entering **q \`0** is the same as entering **q** by itself.

  Note: This entry may look like a complex number, but it is not treated mathematically as such. It is only a convenient method to allow entry of the optional value for **p**.
- The **f** option is equivalent to the number of particles-per-unit-area times the average particle cross-sectional area. Therefore, it is usually a small number much less than one.
- For particles in a **VOLUME**, **f** is the overall fractional area obscuration per unit length, and is equivalent to the number of particles per-unit-volume times the average particle cross-sectional area. Whether the scattering is from actual real particles or something else, only the product of **q** and **f** is important, since it is just equal to the standard "extinction" coefficient.
- Whether the scattering is from actual real particles or something else, only the product of **q+p** and **f** really matters since this is just equal to the standard "extinction" coefficient (**q** times **f** is inversely proportional to the mean free path length).
- Two other models for single sphere scattering are available:
  1. An exact time-consuming **MIE** calculation with particle refractive indices specified by **MEDIA m**.
  2. A fast approximation for large (radii much greater than a wavelength), opaque (indices much greater than one), rough (white Gaussian statistics) spheres with intrinsic surface reflectivity **r** and RMS surface roughness **s** (relative to the sphere radius).

  In both cases, the particle radii lie between **a a'** (in wavelength units) corresponding to the one-over-e-squared-points of a default Gaussian normal-size distribution.
- **PLOT** creates a plot of the model (common base 10 logarithm of the BSDF) for up to seven specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). The current PIXELS setting controls the resolution of these plots in direction cosine space. **PLOT** does not apply to **VOLUME**, where the concept of BSDF, direction cosines, and angle of incidence are meaningless.
- Creates a distribution file **name_angle.dis** for each of these angles.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.
- Optionally, any size distribution function **fcn** can be defined via a previous $FCN command. The main "_" argument of the function is a normalized size between the integration limits -1 to 1; that is, the main "_" argument is equal to

$$\frac{r-\frac{1}{2}(a+a')}{\frac{1}{2}(a'-a)}$$

where **r** is the actual size (again in wavelength units). The **n** is the number of integration samples to use in the given size range (enter zero for a minimal default). The additional (and optional) **c c′ .."** command parameters are passed in the "_1 _2 ..." registers (up to 66). For example, an equivalent to the default distribution would be simply defined as follows:

```
$FCN E2GAUS EXP(-2*_^2)
```

Also, if the distribution function is known in terms of actual size **R**, define it in terms of the normalized radius "_"; that is,

```
$FCN PSDIST R=_*(_2-_1)/2+(_1+_2)/2 ...
MODEL
PARTICLE MIE m a a' f PSDIST a a' ...
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MODELS Overview
MODELS
MEDIA

PATCHES (ASAP Command)

Points represent Bezier surface patches.

Function

Define/Modify Curvedge Entities

Syntax

```
PATCHES k l [ m n [ SEPARATE ] ]
```

Option	Description
`k l`	Bezier degrees used for surface mesh
`m n`	rational patches
`SEPARATE`	use separate rational patches (default is joined)

Remarks

- Make current curve/edge points a surface mesh of Bezier degree **k** by **l** (both not to exceed 20) with **m** by **n** continuously joined or **SEPARATE** rational patches.
- The total number of points must be exactly equal to:

  (1+k*m)*(1+l*n) continuous
  (1+k)*(1+l)*m*n SEPARATE
- The connection **q** factors for each point is interpreted as actual rational Bezier weights **w**.
- Separate patches are defined sequentially from first to last.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

PATHS (ASAP Command)

Produces a table of summarizing propagation paths of currently defined rays.

Function

Analyze Ray/Beam Data

Syntax

```
PATHS [ p ] [ n ] [ AVERAGE ] [ m ] [ SHORT ]
                    PEAK              MEDIUM
                    TOTAL             LONG
                    OBJECT
```

Option	Description
p	flux sorting threshold
n	flag to not print paths containing less than n rays
AVERAGE	sorts paths by AVERAGE irradiance
PEAK	sorts paths by PEAK irradiance
TOTAL	sorts paths by TOTAL flux (default)
OBJECT	sorts paths by OBJECT flux
m	path contribution
SHORT	specifies that the length of the printed output is SHORT (all integer object numbers)
MEDIUM	specifies that the length of the printed output is MEDIUM (current and previous objects integer) (default)
LONG	specifies that the length of the printed output is LONG (all decimal object numbers)

Remarks

- Prints out a table of ray paths according to current, previous, and initial objects.
- If **p** is greater than zero, ASAP sorts the paths by **AVERAGE** irradiance, **PEAK** irradiance, **TOTAL** flux (default), or **OBJECT** flux and prints out only those paths whose percentage contribution is greater than the decimal number **p**.
- Any paths containing less than **n** rays are also not printed.
- The value of the integer **m** determines whether the above flux-related quantity for each distinct path is the minimum (m = -1), sum (m = 0, default), or maximum (m = +1) of the individual contributions.
- The length of the printed output can be **SHORT** (all integer object numbers), the default **MEDIUM** (current and previous objects integer) or **LONG** (all decimal object numbers).
- In the list of **split/scatter** objects, a negative number indicates scattered ray generation. Except for **SHORT** output, the three digits following the decimal point can be used to determine more precisely the mechanism that generated that ray at that object. The code numbers and their meanings are as follows:

1st Digit	Definition	2^{nd}/3rd Digits	Definition
0	Random Scatter	0 0	Hemispherical
0	Random Scatter	# #	TOWARDS number
1	Ordinary Reflection	# #	DIFFRACT number
2	Ordinary Transmission	# #	DIFFRACT number
3	Deterministic Scatter	0 0	Retro-reflected
3	Deterministic Scatter	0 1	Near Reflected
3	Deterministic Scatter	0 2	Near Transmitted
4	Extraordinary Reflection	# #	DIFFRACT number
5	Extraordinary Transmission	##	DIFFRACT number
6	Volume Scatter	# #	MEDIA number

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SCATTER RANDOM/MODEL
SCATTER RMS/BSDF
SPLIT

PERFECT (ASAP Command)

A "perfect" (but realistic) lens.

Function

Define/Modify Lens Entities

Syntax

```
PERFECT X x f h [ t [ h' ] ]
        Y y
        Z z
```

Option	Description
f	focal length
h	input height
t	output distance
h'	output height

Remarks

- A "perfect" (but realistic) lens of focal length **f** (signed), input height **h** (less than magnitude of **f**), output distance **t** (default 0), and output height **h'** (default f*tan(asin(h/f))).
- The output ray vectors are determined from the input ray vectors by the solutions to the eikonal (characteristic) function for perfect imaging of an object plane at infinity (image plane at back focal plane) and no spherical aberration of the principal points.
- Unlike the IDEAL lens, there will be blurring ray aberrations at all other conjugates.
- Two back-to-back PERFECT lenses, with a small collimated space between, them can be used to perfectly reimage a finite-distance object to a finite-distance plane, with a magnification equal to the ratio of their focal lengths.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

PHYSICAL (ASAP Command)

A comprehensive physical reflection model that is "good" even for rough surfaces at grazing incidence.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
PHYSICAL s l [ r  ] [ GAUSSIAN ]
               coat
```

Option	Description
s	RMS roughness
l	autocorrelation length
r	normal incidence ideal reflectivity

Remarks

- Defined in terms of the statistical properties of the surface height variation; that is, RMS roughness **s** and autocorrelation length **l** in WAVELENGTH UNITS for either a fractal or GAUSSIAN PSDF.
- Requires the normal incidence ideal reflectivity **r** or polychromatic COATING name (default is the particular objects specular reflectivity).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

PICTURE (ASAP Command)

Produces a "grey"-scale picture of the current distribution data.

Function

Display/Modify Energy Distributions

Syntax

```
PICTURE
```

Remarks

- Opens the **Data Viewer** window containing the currently defined distribution data (or indirectly opens it using the procedure defined with the **IMAGER** switch or environment variable).
- Also can indirectly produce a "grey"-scale picture of the current distribution data, using the procedure defined with the IMAGER switch or environment variable.
- When the **PICTURE** command is entered, ASAP copies the current DISPLAY distribution data to a file DISPLAYxx.TMP and then creates a Data Viewer window with which you can interrogate the distribution data. With each **PICTURE** command, **xx** is incremented, and a new file is created.
- ASAP deletes the DISPLAYxx.TMP files when the Data Viewer window is closed. If you want to save this display file, use WRITE.
- The **PICTURE** command was formerly the **IMAGE** command in pre-ASAP 5.1 versions.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

PIXELS (ASAP Command)

Sets the resolution for graphics and calculations.

Function

Setup Plots and Verify System

Syntax

```
PIXELS n [ r ] [ ON  ] [ FILL ]
                 OFF
```

Option	Description
n	sets the number of equally-spaced pixels along the vertical axis of the plotting window for this plot only (default=39)
r	ratio of pixels per system unit along the horizontal axis to pixels per system unit along the vertical axis
ON or OFF	flag to turn the plotting of a pixel sampling box on or off (the default)
FILL	fills the plotting window with the plot

Remarks

- Sets the resolution of the various plotting functions by dividing the window into a given number of pixels.
- Also sets the number of rays used to create a profile, and therefore, determines the resolution of system plots.
- If the window is square, the plotting window is divided into a grid **n** pixels vertical by **n*r** horizontal.
- The default value of **n** is 39. (The DIMENSION command may be used to determine the maximum number of pixels allowed in your version of ASAP.)
- The r is the desired aspect ration of the pixels; that is, the ratio of the vertical width to the horizontal (default 1).
- The number of horizontal pixels, **m**, is calculated from the window parameters (a, a', d, d'), n, and r, as follows:

$$a'' = \frac{(a' - a)}{n} \qquad d'' = \frac{a''}{r} \qquad m = \frac{(d' - d)}{d''} = nr\frac{(d' - d)}{(a' - a)}$$

- By default, ASAP autoscales so that the aspect ratio is maintained. It is possible to distort the aspect ratio by making **r** negative.
- Certain graphics commands, PROFILE in particular, can plot a dotted box to highlight the current window setting. ON/OFF toggles this feature; the default is OFF.
- The **FILL** option allows you to plot physically distorted views where the entity completely fills the plotting window in both directions.
- Also controls the number of data pixels generated by the SPREAD (and possibly FIELD) commands (within the current WINDOW) and therefore determines the record structure of the data files they create and use.
- Command argument, ...PIXELS sets the number of pixels for the current plot only.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DIMENSION
FIELD
FMAP
MAP
OPDMAP
...PIXELS
PROFILE
RADIANT
SPOTS
SPREAD
WINDOW

PLACE (Global Coordinate) (ASAP Command)

Specifies the absolute placement of an entity in global coordinates.

Function

Create/Modify Objects

Syntax

```
PLACE [ x y z ] [ LIST ]
        X x
        Y y
        Z z
```

Option	Description
`X, Y or Z`	places axis
`x, y or z`	position in global coordinates
`LIST`	decodes 4-by-4 transformation matrix into simple operations (if possible) and prints

Remarks

- Translates the entity TO the given point (x,y,z). Therefore, TRACE performs an absolute shift.
- If you enter PLACE without any numeric arguments, ASAP moves the entity's reference point to the global origin.
- The LIST option causes the resulting 4x4 transformation matrix to be printed and decoded into simple operations if possible.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PLACE (Relative to Entity)
PLACE (CURve/edge)

PLACE (CURve/edge) (ASAP Command)

Places the entity at the global position that corresponds to parametric coordinate **u** of CURve/edge **k**.

Function

Define/Modify Curvedge Entities

Syntax

```
PLACE CUR k u
```

Option	Description
`k`	entity number
`u`	parametric coordinate

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

PLACE (Global Coordinate)

PLACE (Relative to Entity)

PLACE (Relative to Entity) (ASAP Command)

Places the current entity relative to previously defined entities.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities
Define/Modify Lens Entities
Modify Ray/Beam Data

Syntax

```
PLACE AT  k [ d k'   ] [ LIST ]
      SUR     x y z
      EDG
      LEN
      OBJ
      RAY
```

Option	Description
k k'	entity numbers
d	fractional distance between the two entities (default 0)
x y z	coordinates of the second point in space
AT	refers to the same entity type as the entity being transformed
SUR	specifies SURFACE entity
EDG	specifies EDGE entity
LEN	specifies LENS entity
OBJ	specifies OBJECT entity
RAY	specifies RAY entity
LIST	decodes the 4-by-4 transformation matrix into simple operations (if possible) and prints

Remarks

- Translates the entity TO the point (x,y,z). This form of PLACE determines the point (x,y,z) from the fractional distance d (default 0) from a given entity k to another point or entity k'.
- **PLACE** (Relative to Entity) is an alternate form of the PLACE (Global Coordinate) command.
- **PLACE AT k** tells ASAP to move the current entity such that its reference point is at the same coordinates as the reference point of entity **k**.
- **PLACE AT k 0.5 k'** tells ASAP to move the current entity such that its reference point is located at **0.5** times the distance between entities **k** and **k'**
- **PLACE SUR k 0.75 x y z** tells ASAP to move the current entity such that its reference point is located at **0.75** times the distance between the reference point of surface **k** and point (**x, y, z**).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PLACE (Global Coordinate)
PLACE (CURve/edge)

PLANE (ASAP Command)

Creates a plane normal to the axis of symmetry, at the value given by the third entry.

Function

Define/Modify Surfunc Entities

Syntax

```
PLANE X x  [ aperture ... ]
      Y y
      Z z
```

Option	Description
`X, Y or Z`	axis of symmetry
`x, y or z`	location along coordinate axis
`aperture`	ELLIPSE, RECTANGLE, or HEXAGONAL

Reference Point

At intersection of surface and coordinate axis

Surface Normal

Along positive coordinate direction

Remarks

- Creates a planar surface.
- The normal vector points along the positive coordinate direction.
- This surface can extend to infinity unless a LOCAL command follows, or a trailing aperture option of the following form is specified:

```
ELLIPSE   a [ a' [ o [ s [ s' ] ] ] ]
RECTANGLE
HEXAGONAL a [ o [ s [ s' ] ] ]
```

- For the **HEXAGONAL** form, **a** is the center-to-vertex distance (maximum height).
- **o** is an optional central hole ratio.
- **s s'** are the transverse coordinates of the center of the aperture.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PLANE NORMAL
PLANE POINTS

PLANE NORMAL (ASAP Command)

Creates a plane specified by its surface normal and a point.

Function

Define/Modify Surfunc Entities

Syntax

```
PLANE NORMAL a,b,c  x y z
```

Option	Description
a,b,c	unit normal vector in global coordinates
x y z	point on surface of plane

Reference Point

At specified point

Surface Normal

In specified direction

Autolimiting

No, requires LOCAL or LIMITS modifiers.

Remarks

- Creates a plane surface containing the point (**x y z**) and with unit normal vector (**a,b,c**).
- The normal vector always points into the space on the positive side of the surface where **f(x y z)>0**. As a protective measure, ASAP always renormalizes the vector length to 1.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

General Input Techniques

PLANE POINTS (ASAP Command)

Creates a plane specified by three points.

Function

Define/Modify Surfunc Entities

Syntax

```
PLANE POINTS x y z x' y' z' x" y" z"
```

Option	Description
`x y z x' y' z' x" y" z"`	global coordinates of the three points on the surface

Reference Point

At second specified point

Surface Normal

By right-hand rule

Autolimiting

No, requires LOCAL or LIMITS modifiers.

Remarks

- Creates a planar surface that contains the three points (**x y z**), (**x' y' z'**), and (**x" y" z"**).
- The positive side of the plane is found by applying the right-hand rule to the three points in the order they are entered.
- The second point becomes the surface's reference point.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

PLOT BEAMS (ASAP Command)

Plots extent of each Gaussian beam within the currently specified graphics window.

Function

Analyze Ray/Beam Data

Syntax

```
PLOT BEAMS  [ 'title' ]
```

Option	Description
'title'	optional title for plot (up to 64 characters)

Remarks

- Plots either existing parabasal ray data or internally generated symmetric Gaussian ray data within the window specified by the WINDOW command. The ½-amplitude ellipses are plotted.
- May be executed at any time after the beam is constructed. Use the **WINDOW** command to set the plotting window.
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- Use the SELECT and CONSIDER commands to restrict either the ray set or the objects.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR (*.vcr) file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
PARABASAL
WIDTHS
WINDOW

PLOT CURVES (ASAP Command)

Plots a wire-frame outline of curves, with the control points displayed, in the currently specified graphics window.

Function

Setup Plots and Verify System

Syntax

```
PLOT CURVES [ -n ] [ 'title' ]
            [  k [ k' [ k" ] ] ]
```

Option	Description
`n`	plot last n curves
`k k' k"`	range of edges (k to k') to be plotted in k" steps
`'title'`	optional title for plot (up to 64 characters)

Remarks

- By default, plots all currently defined curves whether or not they are assigned to objects.
- If **n** is entered, the last **n** curves are plotted.
- If **k** is positive and no other entries are present, only the **k**th curves is plotted.
- If **k** and **k'** are present, curves **k** through **k'** are plotted.
- If **k**, **k'**, and **k"** are all present, ASAP plots curves **k** through **k'** in steps of **k"** (that is, **PLOT CURVES 1 5 2** tells ASAP to plot curves 1, 3, and 5).
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
WINDOW

PLOT EDGES (ASAP Command)

Plots a wire-frame outline of edges in the currently specified graphics window.

Function

Setup Plots and Verify System

Syntax

```
PLOT EDGES [ -n ] [ 'title' ]
           [  k [ k' [ k" ] ] ]
```

Option	Description
n	plot last n edges
k k' k"	range of edges (k to k') to be plotted in k" steps
'title'	optional title for plot (up to 64 characters)

Remarks

- By default, plots all currently defined edges whether or not they are assigned to objects.
- If **n** is entered, the last **n** edges are plotted.
- If **k** is positive and no other entries are present, only the **k**th edge is plotted.
- If **k** and **k'** are present, edges **k** through **k'** are plotted.
- If **k**, **k'**, and **k"** are all present, ASAP plots edges **k** through **k'** in steps of **k"** (that is, **PLOT EDGES 1 5 2** tells ASAP to plot edges 1, 3, and 5).
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
WINDOW

PLOT ENTITIES (ASAP Command)

Plots surface, edge, and lens entities in the currently specified graphics window.

Function

Setup Plots and Verify System

Syntax

```
PLOT ENTITIES [ -n ] [ 'title' ]
              [  k [ k' [ k" ] ] ]
```

Option	Description
`n`	plot last n entities
`k k' k"`	range of edges (k to k') to be plotted in k" steps
`'title'`	optional title for plot (up to 64 characters)

Remarks

- By default, plots all currently defined surface, edge, and lens entities whether or not they are assigned to objects.
- If **n** is entered, the last **n** entities are plotted.
- If **k** is entered and no other entries are present, only the **k**th entity is plotted.
- If **k** and **k'** are entered, entities **k** through **k'** are plotted.
- If **k**, **k'**, and **k"** are all present, ASAP plots entities **k** through **k'** in steps of **k"** (that is, **PLOT ENTITIES 1 5 2** tells ASAP to plot entities 1, 3, and 5).
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
WINDOW

PLOT FACETS (ASAP Command)

Plots parametric mesh representation of objects in the currently specified graphics window.

Function

Setup Plots and Verify System

Syntax

```
PLOT FACETS  [ n n' [ m ] ] [ REMOVE    ] [ 'title' ]
                        HIDE [ d ]
```

Option	Description
n n'	minimum or maximum number of subdivisions in the two parametric directions for each patch of an object
m	maximum number of facets per object
REMOVE	removes nonessential facet edges
HIDE	produces hidden-line perspective view
d	viewing distance relative to the maximum lateral size (default: d=2)
'title'	optional title for plot (up to 64 characters)

Remarks

- Produces a parametric mesh representation of all currently defined objects in the current plot window.
- For **n n',** also see the FACETS command.
- If possible, the maximum number of total facets on any object is kept below **m** (the default is 1000; use 0 to turn off).
- Each facet is trimmed by any BOUNDS and/or LIMITS command into, at most, an eight-sided convex polygon. Therefore, it may be necessary to increase the facet density to see the effects of fine-scale trimming.
- **REMOVE** can be used to remove all facet edges from the plot except for the main patch edges and any trimming edges. The plot this option produces is just as accurate but is less crowded.
- For **d** the default is **d=**2, which simulates a typical 35 mm camera lens. Use a **d** of 1000 to produce a nearly parallel projection.
- Currently, the following restrictions apply to the **HIDE** option:
  1. There is a limit to the total number of facets that can be processed, so you may only get a partial plot.
  2. All objects are plotted in the same color.
  3. The plot is always autoscaled to fill the window (or scaled window, that is, **WINDOW 1.1** command before **PLOT** command).
  4. The **VECTOR** unit is restarted so previous 3-D plot data is lost.
  5. Unlike other plots, an oblique view is always true (right-handed). Therefore, OBLIQUE may not produce the expected view.
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-oriented coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-oriented coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
BOUNDS
FACETS
LIMITS
OBLIQUE
WINDOW

PLOT LENSES (ASAP Command)

Plots lenses in the currently specified graphics window.

Function

Setup Plots and Verify System

Syntax

```
PLOT LENSES [ -n ] [ 'title' ]
            [  k [ k' [ k" ] ] ]
```

Option	Description
`n`	plot last n lenses
`k k' k"`	range of edges (k to k') to be plotted in k" steps
`'title'`	optional title for plot (up to 64 characters)

Remarks

- By default, plots all lenses in the current database, whether or not they are assigned to objects.
- If **n** is negative, the last **n** lenses are plotted.
- If **k** is positive and no other entries are present, only the **k**th lens is plotted.
- If **k** and **k'** are present, lenses **k** through **k'** are plotted.
- If **k**, **k'**, and **k"** are all present, ASAP plots lenses **k** through **k'** in steps of **k"** (that is, **PLOT LENSES 1 5 2** tells ASAP to plot lenses 1, 3, and 5).
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
WINDOW

PLOT LIMITS (ASAP Command)

Plots a wire-frame box denoting the extent of the specified objects limits box in the currently specified graphics window.

Function

Setup Plots and Verify System

Syntax

```
PLOT LIMITS [ 'title' ]
```

Option	Description
`'title'`	optional title for plot (up to 64 characters)

Remarks

- By default, plots the limit boxes of all currently defined objects. Objects not CONSIDERed are not drawn.
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands

CONSIDER

WINDOW

PLOT LOCAL (ASAP Command)

Plots local boxes of surface entities.

Function

Setup Plots and Verify System

Syntax

```
PLOT LOCALS  [ -n ] [ 'title' ]
             [  k [ k' [ k" ] ] ]
```

Option	Description
n	plot the local boxes of last n surfaces
k k' k"	range of edges (k to k') to be plotted in k" steps
'title'	optional title for plot (up to 64 characters)

Remarks

- By default, plots the local boxes of all currently defined surfaces whether or not they are assigned to objects.
- If **n** is entered, the local boxes of the last **n** surfaces are plotted.
- If **k** is entered and no other entries are present, only the local box of the **k**th surface is plotted.
- If **k** and **k'** are entered, the local boxes of surfaces **k** through **k'** are plotted.
- If **k**, **k'**, and **k"** are all present, ASAP plots the local boxes of surfaces **k** through **k'** in steps of **k"** (that is, **PLOT LOCAL 1 5 2** tells ASAP to plot the local boxes of surfaces 1, 3, and 5).
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
WINDOW

PLOT MESHES (ASAP Command)

Plots all currently specified objects using a wireframe mesh.

Function

Setup Plots and Verify System

Syntax

```
PLOT MESHES [ m ] [ 'title' ]
```

Option	Description
`m`	interpolation curves between segments
`'title'`	optional title for plot (up to 64 characters)

Remarks

- Plots a wireframe mesh for each currently specified **OBJECT**.
- The **m** is the number of interpolation curves between segments of an object mesh (wireframe). It can be 0, 1 (the default) or 2.
- BOUNDS or LIMITS do not trim plotted meshes. However, any **BOUND** edges for an object are also shown.
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
BOUNDS
LIMITS
WINDOW

PLOT POLARIZATION (ASAP Command)

Plots polarization state ellipses.

Function

Analyze Ray/Beam Data

Syntax

```
PLOT …POLARIZATION [ e ]  [ 'title' ] …
```

Option	Description
e	size of the ellipses is proportional to the ray/beam's flux raised to the "e" power.
'title'	optional title for plot (up to 64 characters)

Remarks

CAUTION: the SOURCE propagation direction must be initialized before running PLOT POLARIZATION, or the command does not plot anything.

- The polarization state ellipses for each ray/beam are plotted. The size of the ellipses is proportional to the ray/beam's flux raised to the **e** (default 1) power.
- Arrows are drawn to show the handedness of the polarization state. In the case of linear polarization, arrows are drawn on both ends of the degenerate ellipse. The size of the tip of the arrow used to show the handedness of the polarization is determined by the overall size of the polarization ellipse. At times it might be necessary to use the ARROW command to scale the tip for better visibility.
- May be executed at any time after the beam is constructed.
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
ARROW
SEGMENTS
SOURCE
POLARIZ
WINDOW

PLOT RAYS (ASAP Command)

Plots vectors along the direction of each ray contained within the currently specified graphics window.

Function

Analyze Ray/Beam Data

Syntax

```
PLOT RAYS d [ i ] [ 'title' ]
```

Option	Description
`d`	scale factor for ray plot vector
`i`	plot only every **i**th ray
'`title`'	optional title for plot (up to 64 characters)

Remarks

- Plots a line from the current position of each ray along its direction vector. The actual distance plotted is **d** times the flux of the ray divided by the maximum ray flux.
- If **d** is entered as a negative number, then only the endpoint of this vector is plotted. This structure is useful for producing 3-D far-field intensity distribution point clouds when **d** is much larger than any inter-ray distance. The default for **d** is approximately 1/10 the window size. If **d** is entered as a negative number, only the end point of this vector is plotted. This option is useful for producing 3D far-field intensity distribution point clouds when **d** is much larger than any inter-ray distance.
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
WINDOW

PLOT SURFACES (ASAP Command)

Plots surfaces in the currently specified graphics window.

Function

Trace Ray/Beams

Syntax

```
PLOT SURFACES [ -n ] [ 'title' ]
              [  k [ k' [ k" ] ] ]
```

Option	Description
`n`	plot last n surfaces
`k k' k"`	range of edges (k to k') to be plotted in k" steps
`'title'`	optional title for plot (up to 64 characters)

Remarks

- By default, plots all currently defined surfaces whether or not they are assigned to objects.
- If **n** is entered, the last **n** surfaces are plotted.
- If **k** is entered and no other entries are present, only the **k**th surface is plotted.
- If **k** and **k'** are entered, surfaces **k** through **k'** are plotted.
- If **k**, **k'**, and **k"** are all present, ASAP plots surfaces **k** through **k'** in steps of **k"** (that is, **PLOT SURFACES 1 5 2** tells ASAP to plot surfaces 1, 3, and 5).
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- In addition to the 2-D plot that is generated, the full 3-D data is written to the VECTOR file (default logical unit 30).
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands
WINDOW

PLOT3D (ASAP Command)

Creates a 3D representation of the current distribution data file with 1D cross-sections.

Function

Display/Modify Energy Distributions

Syntax

```
PLOT3D [ s ] [ 'title' ]
```

Option	Description
`s`	used to readjust the vertical scale
`'title'`	optional title for plot (up to 64 characters)

Remarks

- Produces a 3D hidden line representation of the data and a set of 1D cross-sections on the same plot.
- The **s** can be used to readjust the vertical scale of the plot.
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
ISOMETRIC
Plotting Commands

POINTS (2D) (ASAP Command)

Creates a generalized planar edge.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities

Syntax

```
POINTS X x  y z  q [ y' z' q' ... ]
       Y y  z x      z' x'
       Z z  x y      x' y'
  :
```

Option	Description
X, Y or Z	specifies coordinate axis
x, y or z	specifies location along coordinate axis
y z (z x or x y), y' z' ...	coordinates of point in orthogonal axes
q q' ...	connection parameters

Reference Point

Origin of the coordinates

Autolimiting

Yes

Remarks

- Edges consist of coplanar straight line and/or higher-order curved segments.
- Initially, the reference point is the specified axial location and, therefore, not necessarily on the edge.
- Directly inputs the coordinates of a collection of planar points.
- The **q** parameters control how each edge point is connected to the next, as described in the following table**.**

q PARAMETER	EDGE POINT CONNECTION
q=0	Not connected to the next point (open)
q=1	Connected by a straight line to the next point
q=2	Connected by a quadratic rational Bezier (conic) curve to the point after next. The next point is the intermediate control vertex. The next **q** is the intermediate weighting factor and is always positive**:** 0 straight line <1 ellipse 1 parabola >1 hyperbola
q=-2	Connected by an elliptical arc to the point after next. The next point is the center of the parent ellipse. The next **q** is the angular subtense (in degrees) of the arc and must be less than 180 degrees.
q=n	Connected by an **n**th (up to 10th) degree rational Bezier curve. The intermediate **n**-1 points are control vertices with given (positive) weight factors.

- A nonzero **q** for the last point tells ASAP to connect this point to the first point making a closed curve. However, if the first **q** entry is a literal, the second point entered is the first point on the edge. All other points are measured either ABSolutely or RELatively from the reference point.
- The first point entered is always the reference point for the edge and is normally the first point on the actual edge.
- The **POINT** command, analogous to the GENERAL and SEQUENCE commands, is the most general implementation of ASAP edges. Although the **POINT** command is useful for modeling simple systems, especially when used with the SWEEP command, its primary purpose is to serve as an input form for the complicated output of the IGES to ASAP translator.
- Edge objects ray trace slower than surface or lens objects because of the parametric nature of the edge objects. This edge is a combination of coplanar straight line and higher-order curved segments.
- See POINTS (3D) for further details (excluding reference point).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

POINTS (3D) (ASAP Command)

Creates a generalized edge.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities

Syntax

```
POINTS x y z   q      [ x' y' z' q'... ]
               ABS
               REL
[ x" y" z" q" ... ]
  :
```

Option	Description
x y z [x' y' z' ...]	coordinates of points
q [q' ...]	connection parameters
ABS	measure all points **ABS**olutely from the reference point
REL	measure all points **REL**atively from the reference point

Reference Point

Origin of the coordinates

Remarks

- Use to input the coordinates of edge points directly.
- The **q** parameters control how each edge point is connected to the next, as described in the following table**.**

q PARAMETER	EDGE POINT CONNECTION
q=0	Not connected to the next point (open)
q=1	Connected by a straight line to the next point
q=2	Connected by a quadratic rational Bezier (conic) curve to the point after next. The next point is the intermediate control vertex. The next **q** is the intermediate weighting factor and is always positive**:** 0 straight line <1 ellipse 1 parabola >1 hyperbola
q=-2	Connected by an elliptical arc to the point after next. The next point is the center of the parent ellipse. The next **q** is the angular subtense (in degrees) of the arc and must be less than 180 degrees.
q=n	Connected by an **n**th (up to 20th) degree rational Bezier curve. The intermediate **n**-1 points are control vertices with given (positive) weight factors.

- A nonzero **q** for the last point tells ASAP to connect this point to the first point making a closed curve. However, if the first **q** entry is a literal, the second point entered is the first point on the edge. All other points are measured either **ABS**olutely or **REL**atively from the reference point.
- The first point entered is always the reference point for the edge and is normally the first point on the actual edge.
- If **ABS** or **REL** are entered (instead of the first **q**), the second point entered is the first point on the edge, and it (and all other points) is measured either **ABS**olutely or **REL**atively from the reference point.
- The **POINT** command, analogous to the GENERAL and SEQUENCE commands, is the most general implementation of ASAP edges. Although the **POINT** command is useful for modeling simple systems, especially when used with the SWEEP command, its primary purpose is to serve as an input form for the complicated output of the ASAP/IGES translator.
- Edge objects ray trace slower than surface or lens objects because of the parametric nature of the edge objects.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

POINTS (2D)
REPEAT

POLARIZ (ASAP Command)

Sets the polarization properties for future ray creation.

Function

Setup Beam Creation

Syntax

```
POLARIZ X [ a a' ]
        Y
        Z
        OFF
```

Option	Description
X, Y, or Z	coordinate axis
a a'	complex amplitudes
OFF	flag to turn polarization direction off

Remarks

- Sets the polarization direction and optionally the complex amplitudes of the two orthogonal polarization components of rays, for future ray creation (RAYSET and GRID) and analyses (SPREAD and FIELD).
- Initializes the complex coefficients **a** and **a'** for the two orthogonal components.
- **POLARIZ** selects the ***E*** field direction in the FIELD command. The ***E*** field is parallel to the specified coordinate axis.
- Must precede the GRID, RAYSET, FIELD and SPREAD commands as it is a physical property of the rays.
- The FRESNEL BOTH command should be used in conjunction with POLARIZ to configure the system geometry to include polarization effects.
- If the complex optical field *U* is written in the form:

  $$U = \left(aq + a'r\right)U\left(q,r\right)$$

  where **q,r** are orthogonal unit vectors, then the coordinate axis specified on the **POLARIZ** command is parallel to the **q** direction, and **a** is the complex amplitude of the **q** component of the field. The **a'** is the amplitude of the **r** component of the field. The actual coordinate axis of the **r** direction is determined from the right-hand rule operating on the **q** direction and the direction of ray propagation or, conversely, the direction of propagation given by **q**x**r.**

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GET
GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW
LIST ELLIPSE
Polarization Effects of a Sapphire Dome task
Polarization Ray Tracing task
PLOT POLARIZ
PUT

POLYNOMIAL/TRINOMIAL/BINOMIAL (ASAP Command)

Creates a scatter model based on user-supplied data or polynomial coefficients.

Function

Create/Modify Media, Coatings, Scatter Models

First Syntax (polynomial coefficients):

```
POLYNOMIAL n m [ l [ l' [ d ] ] ] [ PLOT [ a a' ... ] ]
TRINOMIAL
BINOMIAL
  c [ c' c" ... ]
```

Second Syntax (data fitting):

```
POLYNOMIAL n m [ l [ l' [ d ] ] ]  FIT [ k ] [ options…]
TRINOMIAL                          SVD
BINOMIAL
  data…
     :
```

Option	Description
n m l l'	summing indices
d	logarithmic coefficient
c c' c" ...	polynomial coefficients
PLOT	plots the model in log($b-b_o$) and angle space
FIT	FIT the BSDF values to the polynomial model
k	use every **k**th value in the FIT (default is to use every BSDF value entered)
SVD	using the singular value decomposition (SVD) algorithm, FIT the BSDF values to the polynomial model
f [f' ...]	either actual BSDF values or the common LOG of the BSDF

Remarks

- Models scattering described by a general polynomial of three symmetry variables. In this formalism, scattering from an isotropic surface must be symmetric with regard to the plane of incidence and surface normal. Symmetry properties are guaranteed if the BSDF is only a function of the following variables:

$$U = \alpha^2 + \beta^2 < 1, V = \beta\beta_0, W = \beta_0^2 < 1$$

- The α and β are the direction cosine coordinates of scatter direction; the $(0,\beta_o)$ is the specular direction. Note that the distance squared (in direction cosines) from specular is given by.

$$T = \alpha^2 + (\beta - \beta_0)^2 = U - 2V + W < 4$$

- The scattering is modeled by a general polynomial of 2 or 3 symmetry variables with coefficients **c ...** (up to 286); that is,

POLYNOMIAL

$$\log(\mathrm{BSDF}) = \sum_{k=0}^{n} \left\{ \sum_{i=0}^{m[-k]} \sum_{j=0}^{i} c_{ijk} \left(U^i W^j + U^j W^i \right) + \sum_{i=l'}^{l} c_{ik} \log\left(1 + d^i T\right) \right\} \frac{V^k}{2}$$

TRINOMIAL

$$\log(\mathrm{BSDF}) = \sum_{k=0}^{n} \left\{ \sum_{i=0}^{m[-k]} \sum_{j=0}^{i} c_{ijk} U^{i-j} V^{j} + \frac{1}{2} \sum_{i=l'}^{l} c_{ik} \log\left(1 + d^{i} T\right) \right\} W^{k}$$

BINOMIAL

$$\log(\mathrm{BSDF}) = \sum_{k=0}^{n} \left\{ \sum_{i=0}^{m[-k]} c_{ik} \left(\sqrt{T}\right)^{i} + \frac{1}{2} \sum_{i=l'}^{l} c_{ik} \log\left(1 + d^{i} T\right) \right\} V^{k}$$

- If **m** is entered as a negative number, then the upper limit of the second sum becomes $|m| - k$. The coefficients are entered in the order in which they appear in the previous equations for the given **n m l l'**.
- Note that if **d** is zero, then all the Lorentzian (log specular) terms vanish.
- If **FIT** or **SVD** is entered, ASAP fits the user-supplied BSDF data to the specified polynomial form. Under the **SVD** option, a limited number of data points can be fitted. (See DIMENSIONS command output.)
- By default, every BSDF value entered (above 1.E-9, -9 LOG) is used in the fit. Optionally, only every **k**th value can be used.
- The restricted variables used in the **BINOMIAL** form and the term symmetry in the **POLYNOMIAL** form assure that the resulting BSDF exactly obeys reciprocity. The **TRINOMIAL** form is not by definition reciprocal but during a **FIT** it replicates each input point by interchanging the specular and scatter directions. This in effect tries to force the resulting BSDF to obey reciprocity.
- If the peak of the BSDF remains nearly centered on all specular directions (as is the case for relatively smooth or glossy surfaces) or only in-plane data is available, then try the following Lorentzian-only fit:

```
POLYNOMIAL n 0 14 -4 2. FIT
TRINOMIAL
BINOMIAL
```

  where **n** is less than or equal to the number of non-normal incident directions in the data. This generates a model with 20(**n**+1) coefficients. Confirm that there are at least as many BSDF values (a few times more is better).
- Another more general technique is to use $ITER to do a "regressive" fit. For example:

```
$ITER N 1 5 -5, M 1 5 -5, E
 { MODEL 1
   BINOMIAL INT[N] INT[M] SVD
   :
   data
   :
   RETURN
   $GRAB 'RMS=' E }
```

- The **PLOT** option plots the model (common base 10 logarithm of the BSDF) for up to seven specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). The current PIXELS setting controls the resolution of these plots in direction cosine space.
- Creates a distribution file **name_angle.dis** for each of these angles.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.
- Use with importance area sampling.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MODELS Overview

MODELS

PRINT (ASAP Command)

Prints out specific database information.

Function

Setup Plots and Verify System

Syntax

```
PRINT     [  i  ]
SPRINT    SURFACE   [ i ]
QPRINT    EDGE
          LENS
          ALL
          MEDIA
          COATING
          MODEL
          OBJECT
```

Option	Description
PRINT	prints out complete information for the specified database
SPRINT	excludes the main geometry data (that is, coefficients, conicoids, points) from the print out
QPRINT	only one line per entity is printed out
i	entity number
SURFACE, EDGE, LENS, ALL, MEDIA, COATING, MODEL, OBJECT	specific database

Remarks

- Prints database information.
- If the entity number is omitted, all the information in the given database is printed.

Examples

1. Output for a Surface/Object

```
--- MEDIA
 ---   4.03 'GERMANIUM'
 --- COATING PROPERTIES
 ---   0 1  'TRANS'
 --- SURFACE
 ---   PLANE Y -5 ELLIPSE 0.5
 --- OBJECT 'LENS_SURF_1
 ---   INTERFACE COATING TRANS AIR GERMANIUM
 --- SURFACE
 ---   OPTICAL Y -4.7 -1 -16.241 ELLIPSE 0.5
 --- OBJECT 'LENS_SURF_2
 ---   INTERFACE COATING TRANS AIR GERMANIUM
 --- PRINT 2

 SURFUNC 2: Degr= 2x2(OPTICAL) X 0.000000     Y -4.70000     Z 0.000000
  Branch Test sign and direction  1.  0.000000    1.00000      0.000000
  Local Ave Radius -1.0000   Normal   0.0000000   1.0000000    0.0000000
  Local Box X -.5000  0.5000    Y -.7879E-1 0.5000E-3 Z -.5000    0.5000
             width= 1.000          width=0.7929E-1       width= 1.000
```

```
    Cylindrical along Y axis with central ratio   0.000
  This entity used by objects   2
  Polynomial Coefficients:
  x2    0.500000    y   1.00000      z2    0.500000      y2    -7.62050

 OBJECT   2: LENS_SURF_2
  Physical Surface    2 (OPTICAL )
        Box X -.5000   0.5000  Y -4.779    -4.700    Z -.5000    0.5000
             width= 1.000          width=0.7929E-1       width= 1.000
  Coating  1 at 0.0000     wavelength
            Transmission 1.000000 Media   0   1

 MEDIUM  Index/Absorb  FUNC:  exponent   steplength  maxnum
          1.000000       0  1.00000     0.100000E+10  1000  VACUUM_AIR
    1     4.030000       0  1.00000     0.100000E+10  1000  GERMANIUM

                        0.0000
 COATING Name           R  1  T
  1 TRANS             0.000 1.000
```

KEY TO OUTPUT:

SURFUNC = Entity identifier & data base location
Degr = Order of the polynomial surface/function equation
X, Y, Z = Coordinates of reference point
Local Ave = Base radius of curvature
Normal = Surface/function normal direction vector in direction cosines
Local Box = Extents & widths of the LOCAL box in local coordinates
Polynomial = Polynomial coefficients of surface/function
OBJECT = Object data base location & name
Physical = Entity number used by the object
Box = Extents & widths of the LIMIT box in global coordinates
Coating = Coating number & wavelength used by the INTERFACE command
Trans = R & T coefficient values & media surrounding the interface

2. Output for Edge Objects

```
--- COATING PROPERTIES
---   0.05 0 'REFL'
 --- EDGES
 ---   RECTANGLE Y 4 1 .5
 ---   RECTANGLE Y 12 3 1.5
 ---   OBJECT
 ---     0.1 0.2 'FACETED_TUBE'
 ---   INTERFACE COATING REFL
 --- PRINT 1

 CURVEDG  1: Pnts=  4  (RECTANGL) X 0.000000     Y  4.00000     Z 0.000000
  This entity used by objects   1
  Points and Connection Factors:
   x          y          z         q         x          y          z         q
-0.500000  0.000000  1.000000   1.00000  0.500000  0.000000  1.000000 1.00000
 0.500000  0.000000 -1.000000   1.00000 -0.500000  0.000000 -1.000000 1.00000
  Sweep distance & direction   0.000   0.000000        1.00000       0.000000

 CURVEDG    2: Pnts=   4  (RECTANGL) X 0.000000     Y  12.0000     Z 0.000000
```

```
  This entity used by objects   1
  Points and Connection Factors:
    x         y         z         q         x         y         z         q
 -1.500000  0.000000  3.000000   1.00000  1.500000  0.000000  3.000000 1.00000
  1.500000  0.000000 -3.000000   1.00000 -1.500000  0.000000 -3.000000 1.00000
  Sweep distance & direction   0.000   0.000000        1.00000       0.000000

 OBJECT   1: FACETED_TUBE
  Physical Surface    2 (RECTANGL) to    1 (RECTANGL)
        Box X -1.580     1.580    Y  3.920     12.08    Z -3.080     3.080
             width= 3.160          width= 8.160          width= 6.160
  Coating  1 at 0.0000     wavelength
  Interface Reflectivity 0.050000

                       0.0000
 COATING Name           R  1  T
  1 REFL              0.050 0.000
```

KEY TO OUTPUT:

CURVEDG = Entity identifier & data base location

x, y, z = Coordinate of curve/edge points

q   = Previous point to next point connection factor

3. Output for Lens Objects

```
--- UNITS IN
--- WAVELENGTHS 486 587 656 NM
--- LENSES
---   DOUBLET Z 0 1.5 2 SCHOTT_BK7 SCHOTT_F2 10 1 36.6/64.2
---   OBJECT 'DOUBLET'
--- PRINT 1

 LENS   1: Srfs=   3  (DOUBLET ) X 0.000000     Y 0.000000     Z 0.000000
  This entity used by objects   1
  Conicoid Data:
      X            Y            Z            A            B            C
 0.000000     0.000000     0.000000     0.000000     0.000000      1.00000
   APT= 2.000    CRV=0.230632    CON=  0.0000  OBS=.0000 MED= 1 SCHOTT_BK7
 0.000000     0.000000      1.35000     0.000000     0.000000      1.00000
   APT= 2.000    CRV=-.205460   CON=  0.0000   OBS=.0000 MED= 2 SCHOTT_F2
 0.000000     0.000000      1.50000     0.000000     0.000000      1.00000
   APT= 2.000    CRV=0.000000   CON=  0.0000   OBS=.0000 MED= 0 VACUUM_AIR

 OBJECT   1: DOUBLET
  Physical Surface    1 (DOUBLET )
        Box X -2.040     2.040    Y -2.040     2.040    Z -.4000E-1  1.540
             width= 4.080          width= 4.080          width= 1.580
  Interface Reflectivity 1.000000
            Transmission 1.000000 Media   0   0

 MEDIUM  Index/Absorb  FUNC:  exponent   steplength  maxnum
          1.000000       0  1.00000     0.100000E+10  1000  VACUUM_AIR
    1     1.522387       0  1.00000     0.100000E+10  1000  SCHOTT_BK7
    2     1.632104       0  1.00000     0.100000E+10  1000  SCHOTT_F2
```

KEY TO OUTPUT:

LENS = Entity identifier & data base location

Srfs = Number of conicoids of lens

X, Y, Z= Coordinates of reference point

Conicoid Data:

X, Y, Z= Coordinates of first conicoid

A, B, C= Conicoid normal direction vector in direction cosines

APT= Conicoid aperture semi-diameter

CRV= Conicoid base curvature

CON= Conicoid conic constant

OBS= Conicoid obscuration ratio

MED= Media after conicoid

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SURFACES

EDGES

LENSES

MEDIA

COATING

MODEL

OBJECT

PROFILES (ASAP Command)

Plots slices through objects contained within the specified graphics window.

Function

Setup Plots and Verify System

Syntax

```
PROFILES [ f f' n ] [ NOOPTIMIZE ] [ QUICK ] [ 'title' ]
```

Option	Description
`f f'`	range of the third depth coordinate
`n`	number of the third depth coordinate
`NOOPTIMIZE`	generates the original raw plot commands
`QUICK`	turns off exact boundary clipping to speed up the plot
`'title'`	optional title for plot (up to 64 characters)

Remarks

- Basic system plot command for ASAP that is valid for all types of objects.
- **PROFILES** draws the system by tracing grids of rays through the plotting volume. The number of rays (and, hence, the spatial resolution) is determined from the PIXELS command. At the intersection of a ray and an object, a tick mark is drawn at an angle that corresponds to the slope of the object at that point. A complete system plot is therefore built up from these tick marks. All objects are considered to be transparent during the generation of a profile(s).
- Two common commands for producing pictures are:

  PROFILE d d' n for plotting slices of each object by planes parallel to the window and between the depth coordinates d and d' (best resolution with the PIXELS command and number of slices n).

  PLOT FACETS n n' for plotting a 3-dimensional mesh (set resolution by entering values for n and n').
- The **n** is the number, and f to f' is the range of the third depth coordinate (the one different from the other two specified on the last WINDOW command) values for which slices through the current set of objects are plotted.
- If **n** is negative, then **f f'** are ignored on objects with LOCAL/LIMITS, and **n** equally spaced slices are made across the individual local/limits ranges of those objects. If **n** is positive, then **n** equally spaced slices are made across the plotting volume. An object appears on the plot **only** if a slice intersects the object. In most cases, it is best to use a negative value for **n**.
- The default for n is -1 when OBLIQUE is off and -5 when it is on.
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X**, or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- Normally ASAP tries to connect any short vectors generated by a **PROFILE** command. This slows down the immediate generation of the graphics, but produces a cleaner plot file that won't overwork a pen plotter. Use the **NOOPTIMIZE** option to generate the original raw plot commands.
- The title is delimited by a single quote **'** , as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands (includes standard plot command options)

PUT (ASAP Command)

Copies variable data into the current ray data.

Function

Modify or Use Internal Ray/Beam Data as Input

Syntax

```
PUT k
```

Option	Description
k	number of a given ray

Remarks

- Puts the current values in the input registers back into the storage locations for ray **k**.
- The input register assignments are as follows:

Register	Literal	Ray/Beam Data
A0,B0,C0	X,Y,Z_DIR_B	Absolute X,Y,Z direction cosines of base ray
Ai,Bi,Ci	X,Y,Z_DIR_i	Relative direction vector of **i**th parabasal ray
D0	OPL	Optical path length from start of base ray
E1,E2,E3	X,Y,Z_EPOL	Components of unit polarization vector
F0	FLUX	Total flux in ray/beam
G0	DIVERG	Average divergence angle of beam
H0	HEIGHT	Average height of beam centered on base ray
Li	PREV_O_i	**i**th previous split object for ray/beam
J0	SOURCE	Source number from which ray/beam originated
K0	CURR_OBJ	Current object at which ray/beam is located
L0	HITS	Total number of surfaces ray has hit (intersected)
M0	MEDIUM	Medium that ray/beam is in
N0	SPLITS	Number of times ray/beam has been split
N1	LEVELS	Number of times ray/beam has been scattered
P0	POLAR_0	Relative modulus of fundamental beam mode
P1,P2	POLAR_1,2	Relative moduli of polarization components
Q0	NUM_RAYS	Total number of ray/beams
Q1	NSOURCES	Total number of original sources
R0	PARENT	Number of ray from which this ray was split (parent)
S0	SHAPE	Beam shape number (see SHAPES command)
S1	FACTOR	Beam shape factor or number of higher modes
T0	PHASE_0	Relative phase angles of fundamental beam mode
T1,T2	PHASE_1,2	Relative phase angles of polarization components
U0,V0	U,VPARAMB	Parametric coordinates of base ray position
W0	WAVELEN	Wavelength of ray/beam
Wi	WAVLNS_i	Wavelength for **i**th source
X0,Y0,Z0	X,Y,Z_POS_B	Global X,Y,Z coordinates of base ray
Xi,Yi,Zi	X,Y,Z_POS_i	Relative coordinates of **i**th parabasal ray

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

```
GET
SECTION
SHAPES
```

RACETRACK (ASAP Command)

Creates a racetrack or rectangular edge with smooth (continuous) rounded corners.

Function

Define/Modify Curvedge Entities

Syntax

```
RACETRACK X x y z y' [ z'] [ SPLIT ]
          Y y z x z'   x'
          Z z x y x'   y'
```

Option	Description
`X, Y or Z`	coordinate axis
`x, y or z`	location along coordinate axis
`y z, z x, or x y`	overall semimajor widths

`y' [ z' ], z' [ x' ],`
`or x' [ y' ]`
	semimajor widths of the elliptical corners

`SPLIT`	flag to split straight-aways into two equal line segments

Remarks

- RACETRACK creates a racetrack or rectangular edge with straight sides and smooth (continuous), rounded corners.
- If the overall semiwidths are equal to the corner semiwidths, the racetrack has no straight-aways; it is a pure ellipse.
- If the corner semiwidths are zero, the racetrack has no elliptical turns; it is a pure rectangle.
- For compatibility with previous versions of ASAP, the straight-aways can be **split** into two equal line segments.
- This edge is a combination of coplanar straight line and higher-order curved segments.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

RADIAL (ASAP Command)

Rotationally averages the current distribution data, and optionally computes the encircled energy.

Function

Display/Modify Energy Distributions

Syntax

```
RADIAL [ i j ]  [ FUNCTION  [ REPLACE ] ] [ 'title' ]
         MAX      INTEGRAL  [ p ]
                  BOTH
```

Option	Description
`i j`	rotationally average the distribution data about the pixel (**i j**)
`MAX`	rotationally average the distribution data about the maximum point
`FUNCTION`	plot out a cross-section of the radially averaged function
`INTEGRAL`	plot out the cross-section of the encircled energy
`BOTH`	plot the cross-sections of the radially averaged function and the encircled energy
`REPLACE`	replace the current distribution data with the radially averaged data
`p`	print the radius of **p** percent encircled integral
'`title`'	optional title for plot (up to 64 characters)

Remarks

- Rotationally averages the distribution data about the centroid (default) and then replaces the current distribution data.
- **RADIAL i j** causes ASAP to rotationally average the distribution data about the pixel (**i j**).
- **RADIAL MAX** causes ASAP to rotationally average the distribution data about the maximum point.
- In all cases, the resulting distribution is radially symmetric about the given point.
- **FUNCTION** plots out a cross-section of the radially averaged function, **but does not replace the current distribution data**.
- **INTEGRAL** plots out the cross-section of the encircled energy, **but does not replace the current distribution data.**
- **BOTH** plots the cross-sections of the radially averaged function and the encircled energy, **but does not replace the current distribution data**.
- **RADIAL BOTH** and **RADIAL INTEGRAL** should not be used after the **ANGLES** command when averaging directional distributions. **ANGLES** remapping will distort the coordinate system, making the radial average invalid.
- **REPLACE** replaces the current distribution data with the radially averaged data.
- If **p** is specified, the radius of **p** percent encircled integral is printed.
- **RADIAL** should only be used when rotational symmetry is expected; in most cases use the AVERAGE command, which assumes no symmetry.
- The title is delimited by a single quote **'** , as shown.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

RADIANT (ASAP Command)

Calculates the far-field incoherent intensity or radiance of the currently selected ray data.

Function

Analyze Ray/Beam Data

Syntax

```
RADIANT [ X  [ f f' n a a' n' ] ] [ MAP  [ s ] ]  [ LOW ]
          Y                         AREA            HIGH
          Z
         -X
         -Y
         -Z
```

Option	Description
X, Y, or Z	specifies polar axis of the coordinate system (default Z)
f f'	zenith/latitude angle ranges
n	number of zenith/latitude subdivisions
a a'	azimuth/longitudinal angle ranges
n'	number of azimuth/longitudinal subdivisions
MAP	radiant intensity (flux per solid angle) calculation option; produces polar plots of the radiant intensity as a function of spatial position
AREA	radiance (flux per solid angle per projected area) calculation option; produces a distribution file containing the radiance (flux/solid angle/projected unit area)
s	polar diagram rescaling factor

Remarks

- Normally calculates in spherical coordinates the far-field incoherent radiant intensity (flux per solid angle) pattern of the current ray set.
- The **f f'** is the range in degrees for the angles from the polar axis (zenith/latitude angles). The defaults are 0 to 180 degrees. The **n** is the number of subdivisions of this angular range (default 36; that is, five-degree increments).
- The **a a'** is the range in degrees for the angles about the axis (azimuth/longitude angles). The defaults are -180 to 180 degrees. The n' is the number of subdivisions of this angular range (default 36, that is, 10 degree increments).
- The pattern is written to the standard distribution file BRO009.DAT and can therefore processed further by the DISPLAY commands (for example, DIRECTION or MESH produces a 2-D or 3-D representation of the radiation pattern, which may be viewed with REPLOT).
- **MAP** or **AREA** also uses the current PIXELS and WINDOW settings to calculate the full four-dimensional pattern, that is, a function of not only the two angles but also the two spatial coordinates. MAP outputs radiant intensity (flux per solid angle) while AREA outputs radiance (flux per solid angle per projected area).
- The 3-D polar-diagrams for each point on the spatial grid is written to the current VECTOR file (use REPLOT or $VIEW to see this data) as either simple wireframes or LOW/HIGH-resolution, shaded "surfaces".
- The **s** is an optional rescaling factor for altering the sizes of the polar diagrams relative to the spatial grid spacing.
- If the angular sampling is 1x1, the radiant intensity or radiance map for this solid angle is written to the standard distribution file, BRO009.DAT.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DIRECTION
DISPLAY
DIRECTION
PIXELS
MESH

PIXELS
REPLOT
WINDOW

RANGE (ASAP Command)

Adjusts the vertical plotting scale of current distribution data.

Function

Display/Modify Energy Distributions

Syntax

```
RANGE [ d [ d' ] ]
```

Option	Description
d	minimum values to be plotted
d'	maximum values to be plotted

Remarks

- Expands or recalculates the minimum **d** and/or maximum **d'** values to be plotted by the GRAPH, PLOT3D, ISOMETRIC and CONTOUR and MESH commands.
- Plots different data at the same fixed scale.
- Makes the plotting range only larger than the actual data range.
- Use the THRESHOLD command to "clamp" the data values to a smaller range.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

RAY (ASAP Command)

Defines and traces a single ray.

Function

Trace Ray/Beams

Syntax

```
RAY x y z a,b,c [h] [DIR] [PLOT [d]] [SEARCH] [KEEP] [LENS [n]]
```

Option	Description
`x y z`	starting coordinates of ray
`a,b,c`	initial direction vector of ray
`h`	cross-sectional, semidiameter (default 1)
`DIR`	prints ray coordinates and direction cosines at each intersection
`PLOT`	plots ray and any PARABASALS in current graphics WINDOW
`d`	ray depth coordinate
`SEARCH`	ray intersection logic option
`KEEP`	adds ray to current ray storage after ray tracing; otherwise deletes
`LENS n`	creates a lens entity **n** from the sequence of surfaces encountered by the parent ray

Remarks

- Traces a single ray/beam through the entire system starting in the IMMERSE medium at the point (x y z), with a direction given by the vector (a,b,c) and cross-sectional semi-diameter h (default 1).
- The program always lists the ray coordinates at each surface intersected. The **DIR** option adds the ray directions to the listing.
- The **PLOT** option plots the ray and any PARABASALS in the current graphics WINDOW. If you enter RAY interactively and the ray is being overlaid on a previous system graphics screen, you may trace more rays by marking the starting point (depth x, y or z) and second point (depth d) with the graphics cursor. Press the right mouse button (or the Enter key) to terminate this mode.
- The **d** is the depth of each ray starting coordinate within the plotting window.
- The **SEARCH** option resets the allowable object intersections for future rays according to the sequence of objects encountered during this single ray trace.
- Unless the **KEEP** option is used, any rays traced with RAY are deleted from storage after they have been traced.
- The sequence of surfaces encountered by the parent ray can be optionally used to create LENS entity **n** (default is one more than current highest entity number).

  Some commands, including RAY, require the specification of a direction vector. The following format can be used for this input entry:

`RAY`	Direction of vector between positions of the two entities.

Examples

1. General Format

```
Ray numberi
Object X Y ZSize        Flux
 U  V
object name
object numberx yzbeam size          flux before intersection
u v
   Nrml m naverage curvature  incident angle Inc
   Dirabcbeam divergence    flux after intersection
```

Notes on "u" and "v" parameters:

The "u" and "v" are not printed for a SURFACE;OBJECT unless the SURFACE is a SURFACE; ARRAY;OBJECT. Then the "u" is the "n"th array element of the OBJECT and the "v" is the "n'"th array element of the OBJECT that the ray intersects.

The "u" indicates the conicoid number for the LENS;OBJECT the ray intersects.

The "u" and "v" for an EDGE;OBJECT parametrically indicate the mesh number and the fractional coordinate within the mesh that the ray intersects.

2. Plot Options

```
--- WIN Y Z; WIN 1.5
 --- PROFILE 'RAY COMMAND RAY TRACE AND PROFILE' OVER
  Across or Vertical: Y = -7.50000     to  7.50000      ( 15.0000    )
  Down or Horizontal: Z = -21.3806     to  16.2821      ( 37.6627    )



   -1 Cuts for X=0.000000     to 0.000000
 --- RAY 2 2 -20 0 0 1 PLOT 2 SEARCH
  Across or Vertical: Y = -7.50000     to  7.50000      ( 15.0000    )
  Down or Horizontal: Z = -21.3806     to  16.2821      ( 37.6627    )



 Ray number      1
 Object       X             Y             Z          Size          Flux
    0    2.000000      2.000000     -20.00000     1.000000      1.000000
    2    2.000000      2.000000     -.1000000     1.000000      1.000000
    1   0.5005005     0.5005005     -15.02002    0.2502502     0.8999999
    3   -.1916530E-06 -.1916530E-06  10.00000    0.5674274E-06 0.8099998
          Total OPL =  60.00000000381213
  Effective Focal Length =  100.020

  Across or Vertical: Y = -7.50000     to  7.50000      ( 15.0000    )
  Down or Horizontal: Z = -21.3806     to  16.2821      ( 37.6627    )



 Ray number      2
 Object       X             Y             Z          Size          Flux
    0    2.000000     -4.188794     -10.49998    1.000000      1.000000
    2    2.000000     -2.989614     -.1617224    1.000000      1.000000
    1   0.4851324      1.061682     -15.05435    0.2309581     0.8999999
 ***Ray      2 missed after object   1 - SECONDARY_MIRROR
  Effective Focal Length =  74.5145

 --- SEARCH LIST
 Object From  To  By   Description
    0    2    2    1
    1    1  300    1  SECONDARY_MIRROR
    2    1    1    1  PRIMARY_MIRROR
    3    1  300    1  DETECTOR
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

...OVERLAY (ASAP Command)

RAYS (ASAP Command)

Sets the total number of rays in the virtual memory paging.

Function

Setup Plots and Verify System

Syntax

```
RAYS [ n ]
```

Option	Description
n	total number of rays in storage

Remarks

- A shortened version of the RAYSET command that resets the total number of rays in storage to **n**. The default is the number of rays in the current virtual ray file; that is, the rays from a previous run can be restored with RAYS. The following two commands restart an analysis from the last time ASAP was run within the current directory:

```
SYSTEM FROM     !!restores system from the LASTEXEC.SYS file
RAYS            !!restores rays/beams from the VIRTUAL.PGS file
```

- Reinitialize ray storage by entering **RAYS 0**.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SYSTEM

RAYSET (ASAP Command)

Specifies an arbitrary set of rays.

Function

Create Rays/Beams

Syntax

When rays originate in the X plane:

```
RAYSET X x
y [ z f y' z' k [ a a' a" ... ] ] [ NOSPLIT ]
     :            s
```

When rays originate in the Y plane:

```
RAYSET Y y
z [ x f z' x' k [ a a' a" ... ] [ NOSPLIT ]
     :            s
```

When rays originate in the Z plane:

```
RAYSET Z z
x [ y f x' y' k [ a a' a" ... ] [ NOSPLIT ]
     :            s
```

Option	Description
X, Y, or Z	plane in which rays originate
x, y or z	location of plane from which rays originate
y z, z x, or x y	coordinates of ray on that plane
f	flux assigned to the ray
y' z', z' x', or x' y'	45 percent amplitude semidiameters of the parabasal ray waist
k	beam shape and coherence parameter
a	complex amplitude of TEMoo beam mode
a' a" ...	complex amplitudes of higher order beam modes
s	optional beam shape parameter
NOSPLIT	flag to never split this ray, regardless of what the **SPLIT** command indicates

Remarks

- Object zero is used to refer to the ray data.
- Any number of ray coordinates may follow the **RAYSET** command.
- **RAYSET** always adds rays to the current ray storage until a TRACE is performed.
- The **f** is the flux (energy per unit time) to be assigned to each corresponding ray (default=1, unity).
- The primed coordinates are the relative positions of the 2 orthogonal parabasal rays to be traced if the number of parabasal rays has been previously set by a PARABASAL command.
- The **k** sets the beam shape and coherence of the ray (See the SHAPE command for a definition of **s** and further information).
- If k=0, the beam is a coherent set of Hermite-Gaussian modes and **a a' a". . .** are the complex amplitude coefficients for each mode. The modes are ordered as follows: 00 10 01 20 11 02 30 21 12 03 40 31 22 13 04 50
- The default amplitude for the fundamental mode (TEMoo) is 1 (unity). The amplitudes of the higher order beam modes default to zero.
- When higher- order beam modes are desired, both the number of higher-order modes and their amplitudes must be entered. The number of higher order beam modes is given by **n**.
- The **NOSPLIT** option tells ASAP to never split this ray, regardless of what the SPLIT or LEVEL commands indicate.
- The RAYS command (short form for the RAYSET command) resets the number of rays in storage to **n**. The **RAYSET** and **GRID** commands always add their rays to the ones already in storage until a TRACE is performed.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Apodization of Ray Distributions
GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW

RECTANGLE (ASAP Command)

Creates a rectangular edge/curve.

Function

Define/Modify Curvedge Entities

Syntax

```
RECTANGLE X x y z [ n a a' ]
          Y y z x
          Z z x y
```

Option	Description
X, Y or Z	specifies the axis of symmetry
x, y or z	location along coordinate axis
y z (z x or x y)	semimajor widths of the rectangle
n	number of points (or segments) on the rectangle (default 4)
a a'	angular range (in degrees from first semimajor axis) over which the rectangle is defined (default is 0 to 360 degrees)

Remarks

- Defines a rectangular polygon. See ELLIPSE for description of parameters.
- When defining a closed rectangle (0 to 360 degrees), **n** should be a multiple of 4 so that there are edge points at each corner.
- The default value for **n** is 4 or the value specified on a previous **RECTANGLE** command. Use **-n** if you want it to become the default for future **RECTANGLE** commands.
- If **n**, **a** and **a'** are specified, they become the default settings for most future EDGE commands.
- This edge is made up of coplanar straight line segments, that is, convex polygons whose vertices lie on a particular curve.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

REDEFINE (ASAP Command)

Adds options to an object.

Function

Create/Modify Objects

Syntax

```
REDEFINE [ SURFACE j ] [ THICKNESS t ] [ COLOR k ] [ 'name' ]
           NORMAL
```

Option	Description
SURFACE	transfers to alternate surface
j	surface number
NORMAL	uses alternate surface normal
THICKEN	flag to thicken the object surface(s)
t	distance by which object surface(s) are thickened
COLOR k	assigns a specific color (1 to 26) to the object
'name'	descriptive name to be assigned to this object

Remarks

- The **SURFACE** option forces ASAP to move any intersection points on a non-lens object to a nearby continuous surface **j** after first intersecting the original object surface. The **j** is also then used to calculate the normal for refraction and/or reflection.
- The **NORMAL** option allows the user to designate a continuous surface number **j** to be used for calculating just the normal to the object that is different than the actual object normal.
- The **SURFACE** and **NORMAL** options are useful for modeling Fresnel optics and more accurately tracing rays/beams through discrete faceted approximations to objects.
- The **THICKNESS** option is used to thicken the object surface(s), that is, effectively turn it into two surfaces spaced a small distance **t** apart. The sign of **t** determines in which direction along the surface normal the second surface is located.
- The **COLOR** option assigns a specific color (1 to 26) to the object that is used in all subsequent graphical displays unless overridden by the global COLOR command.
- Although **REDEFINE** may be used with objects based upon lenses with its full options, the most common usage is for **COLOR** remapping of a lens object.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

COLORS

REDUCE (ASAP Command)

Selects a subset of the current distribution data and rejects remaining data.

Function

Display/Modify Energy Distributions

Syntax

```
REDUCE [ s  [ s' ]  ]
        -s
        +s
         m"  [ n" ]
         m  m'  m" [ n n' n" ]
```

Option	Description
s	select out a square region of points (either absolute number or fractional) at the center
-s	select out a square region of points (either absolute number or fractional) at the lower corner
+s	select out a square region of points (either absolute number or fractional) at the upper corner
m m'	integer pixel ranges in the across direction
n n'	integer pixel ranges in the down direction
m"	increments in the across direction
n"	increments in the down direction

Remarks

- Reduces the number of data points by selecting out a subset of the current data, that is, the new data points are given by:

$$f(i,j) : i = m \text{ to } m' \text{ by } m'',\ j = n \text{ to } n' \text{ by } n''$$

- The defaults for the increments **m"** and **n"** are 1.
- The default for **n** is **m** and **n'** is **m'**.
- A single entry **s** selects out a square region of points (either absolute number or fractional) at the center, lower corner (**-s**), or upper corner (**+s**).
- If no entries are given, the outer minimum value is cropped off.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

RENDER (ASAP Command)

Creates rendered graphics of the system geometry and ray trace.

Function

Setup Plots and Verify System

Syntax

```
RENDER [   d   ] [ DEPTH [ d' [ d"] ] ] [ RAYS [ s ] ] [MODEL [ m ] ]
       x y z
```

Option	Description
d	distance of the observation point in front of the scene
x y z	coordinates (global) of the observation point
DEPTH d' d"	cuts in the depth direction at the hither (near) and yon (far) coordinate planes d' and d"
RAYS	flag to add a ray trace to the rendering
s	scale factor for the width of the rays
MODEL	any reflective SCATTER MODELS assigned to objects are used in rendering
m	defaults objects with no SCATTER MODELS to given MODEL m

Remarks

- Creates a 3-D shaded view (rendering) of the system geometry as seen through the current plot window, and as determined by the WINDOW, PIXELS, and LIGHT commands. **RENDER** works for all types of entities with complex bounds and obscurations.
- The observation (eye/camera) point is either given absolutely as (**x,y,z**) or as a distance **d** (default 10 times maximum scene span) in front of the scene.
- The scene is always illuminated by a light source of unit irradiance emanating from the observation point (for example, a camera-mounted flash bulb). Therefore, unless a second **LIGHT** source is specified, no shadowing is visible.
- Normally, only the outside of the objects is visible. However, the outside can be cut away in the **DEPTH** direction at the hither (near) and yon (far) coordinate planes **d'** and **d".**
- The rendered plot is written to the distribution data file BRO009.DAT. See the DISPLAY command for more information about reading and displaying the data distribution file. A somewhat crude representation of the scene is written to the 2-D plot device during the rendering process.
- RAYS can be added to the rendering if SAVE was turned on before the last TRACE command. Only those portions of the ray paths not hidden by any part of any object (whether it is transmissive or not) is visible.
- The addition of rays can significantly increase the time to render a scene; therefore, the number of rays should be kept to a minimum. The **s** is an optional scale factor for the width of the rays (default is 1, that is one pixel). If **s** is entered as a negative number, each ray segment is rendered as a smooth cylinder (that is, anti-aliasing).
- Normally all object surfaces are considered to be perfect 100% Lambertian diffuse reflectors. With the **MODEL** option, any reflective **SCATTER MODELS** assigned to objects are used in the rendering. Objects with no SCATTER MODELS assigned to them will either default to 100% Lambertian or the given MODEL **m**.
- Each pixel point is a single floating point value where the integer part is a color number and the fractional part its intensity (shade). A standard COLORS command argument can be used to set the background color to something different from the default black.
- Previously, a separate utility program (such as DIS2PS or DIS2VGA) was required to display this file on a device capable of representing many intensities of the colors. This utility can still be run automatically if it is specified with the IMAGER switch or environment variable.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

RENORM (ASAP Command)

Renormalizes coefficients of surface/function polynomials.

Function

Define/Modify Surfunc Entities

Syntax

```
RENORM [ c ]
       [ term c ]
       [ MIN  c ]
       [ MAX  c ]
       [ FUNC c [ x y z ] ]
```

Option	Description
c	normalization constant
term, MIN, or MAX	normalizes the specified coefficients to be exactly equal to c
x y z	point at which function value is to be set

Remarks

- Renormalizes the coefficients of the previous surface function.
- If it is actually a surface (function is 0), the renormalization does not affect the location of the surface, but may change the direction of the normal to the surface or avoid numerical overflow problems.
- If no additional entries are made, the coefficients are normalized such that the largest and smallest ones are equally spaced in magnitude about 1.
- If a single numerical value is entered, all the coefficients are divided by **c**.
- If a **term** is specified (that is, **XiYjZk** or the **MIN/MAX**), then the coefficients are normalized so that the particular coefficient becomes exactly equal to **c**.
- The last form allows you to set the value of the **FUNC** to **c** at the point (**x,y,z**) (default is the reference point).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

REPEAT (ASAP Command)

Repeats previously defined entity data.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities
Define/Modify Lens Entities

Syntax

```
SURFACE or EDGE or LENS
REPEAT [ i ]
```

Option	Description
`i`	previously defined entity number

Remarks

- Copies data from a previously defined entity into the current entity location.
- The default for **i** is 0.1 or the last entity defined.
- After repeating entity data the current entity data may be changed by various entity modifiers, that is, linear transformations.
- If the entity is a SURFACE, then all the coefficients are transferred in their final form, which include any linear transformations applied to the original surface.
- If the entity is an EDGE, then all edge points are transferred in their final form, which include any linear transformations applied to the original edge.
- If the entity is a LENS, then all conicoid data are transferred in their final form, which include any linear transformations applied to the original lens.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

REPLICATE (ASAP Command)

Replicates (copies) a distribution.

Function

Display/Modify Energy Distributions

Syntax

```
REPLICATE [ n ]
```

Option	Description
`n`	|n| times in horizontal (second) direction

Remarks

- Copies a distribution |**n**| times in horizontal (second) axis.
- If **n** is negative, alternate reversing (mirroring) the distribution for each replicant.
- The default for **n** is one.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

REPLOT (ASAP Command)

Plots data stored in the 3D vector file within the currently defined graphics window.

Function

Setup Plots and Verify System

Syntax

```
REPLOT [ NORAYS ] [ OPTIMIZE ] [ 'title' ]
```

Option	Description
`NORAYS`	plot only intersections of rays with objects
`OPTIMIZE`	optimize plot vectors
'`title`'	optional title for plot (up to 64 characters)

Remarks

- Replots within the currently defined graphics WINDOW all the 3-D vector data found in the BRO030.DAT (*.VCR) file. Vector data outside of this window is not plotted.
- Useful for zooming in on particular areas by changing the **WINDOW** and **REPLOT**ting.
- The **NORAYS** option suppresses the replotting of the rays themselves but does plot the intersection points of the rays on each object.
- The **OPTIMIZE** option forces ASAP to try and connect up any short vectors generated by a previous PROFILE command. This slows down the immediate generation of the graphics but produces a cleaner plot file that won't drive a pen plotter crazy.
- A graphical cursor (that is, crosshairs) appears at the end of plotting. You can then position this cursor and press the specified key (or mouse button) to perform the following operations:

Key	Mouse button	Operation
?		Display 3D system coordinates of cursor position in output window
#		Display number of object nearest to cursor
space	left mouse button	Mark the lower left-hand corner of a new window
enter	right mouse button	Opens menu; select **End Replot** to terminate

- The **#** operation also replaces the current GROUP with all objects so-tagged (a GROUP and/or CONSIDER command can then be used to operate on these selected objects).
- The **?** operation also puts the coordinates in the output buffer for access by $GRAB.
- For the last operation, place the cursor at the upper-right corner of the desired window. After striking another space key (or left mouse button), ASAP draws a box on the current plot to indicate the extent of the new window and then beeps. Pressing **Enter** or clicking the right mouse button causes ASAP to clear the screen and to begin replotting the data in the new window. This zooming-in process can be repeated indefinitely. If the two corners of the new window are entered as the same point, then ASAP zooms back to include the entire plot.
- The **$IO VECTOR REWIND** command may be used to rewind (and thereby reinitialize) the BRO030.DAT file so that new data may be written into the file.
- **WINDOW X Z**, **WINDOW Y X**, and **WINDOW Z Y** display the system in a left-handed coordinate system. Use **WINDOW X -Z**, **WINDOW Y -X** or **WINDOW Z -Y** to display the system in a right-handed coordinate system.
- The title is delimited by a single quote (**'**), as shown.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands (includes standard plot command options)
CONSIDER
CONTOUR
$IO VECTOR REWIND

PLOT
PROFILE
WINDOW

RESET (ASAP Command)

Reinitializes all control settings to those at program startup.

Function

Save or Recover System Data and Control Execution

Syntax

```
RESET
```

Remarks

- Also resets all ray data with the equivalent of the RAYS 0 command.
- Does not affect system description (use SYSTEM NEW to reset this).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SHOW ALL

RETURN (ASAP Command)

Returns to ASAP top command level (that is, the `ASAP>` prompt).

Function

Save or Recover System Data and Control Execution

Syntax

```
RETURN
```

Remarks

- Alternatively, RETURN can be used to terminate the graphics mode (that is, OVERLAY) and return you to text command level.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

REVERSE (ASAP Command)

Reverses the propagation direction of all currently defined rays.

Function

Modify Ray/Beam Data

Syntax

```
REVERSE [ OPL ]
```

Option	Description
`OPL`	signals ASAP to also start subtracting optical path lengths instead of adding them for ALL rays

Remarks

- Reverses the directions of the currently selected rays.
- Useful for doing virtual ray traces.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MOVE

REVOLUTION (ASAP Command)

Creates a surface specified by spinning a 2D curve around a given axis.

Function

Define/Modify Surfunc Entities

Syntax

(numbers represent powers):

```
REVOLUTION X x c0 z1 x1 [z2 zx x2 [z3... [z4... [z5... ]]]] [DECENTER d]
           Y y c0 x1 y1 [x2 xy y2 [x3... [x4... [x5... ]]]]
           Z z c0 y1 z1 [y2 yz z2 [y3... [y4... [y5... ]]]]
```

Option	Description
`X, Y or Z`	axis of symmetry
`x, y or z`	location along coordinate axis
`c0 z1 x1..., c0 x1 y1..., or c0 y1 z1...`	coefficients of 2D curve
`DECENTER d`	optional decentering

Reference Point

At location along coordinate axis.

Surface Normal

Radially outward from the axis.

Autolimiting

No; requires LOCAL or LIMITS modifiers.

Remarks

- Takes the general 2D curve defined relative to the given coordinate by the ascending coefficients (up to 21, that is, fifth), optionally decenters it a distance **d**, and then rotates it about the given axis to form a 3D surface of up to twice the order of the curve.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

REVOLUTION (Fitted) (ASAP Command)

Creates a surface by spinning a 2-D curve fit to data points.

Function

Define/Modify Surfunc Entities

Syntax

```
REVOLUTION X 1ST z x z' x' ... [ VECTOR [c ] ]
           Y 2ND x y x' y'
           Z 3RD y z y' z'
             4TH
             5TH
             FIT
```

Option	Description
X, Y or Z	axis of symmetry
1ST 2ND ...	type of curve fit
z x z' x' ..., x y x' y' ..., or y z y' z' ...	coordinate pairs (up to 1000)

Reference Point

Axial location of the point furthest from the axis

Surface Normal

Radially outward from the axis

Autolimiting

Yes

Remarks

- Determines the coefficients of a 1ST, 2ND, 3RD, 4TH or 5th-order curve by fitting in a least squares sense to a series of points (up to 5000 coordinate pairs).
- The axial location of the point furthest from the axis becomes the surface's reference point.
- The threshold on the last LSQFIT command is used to determine if any of the resulting coefficients can be considered negligible (that is, zero).
- The VECTOR options puts each data point in the current 3D file as a dot of color "c" (default 1) for later plotting by a REPLOT, DRAW or $VIEW command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

LSQFIT

RIGHT (ASAP Command)

Creates a simple prism that deviates the rays by 90 degrees.

Function

Define/Modify Lens Entities

Syntax

```
RIGHT  X x h m [ Y,Z ]
PENTA  Y y       Z,X
       Z z       X,Y
```

Option	Description
X, Y, or Z	global coordinate axis
x, y, or z	location on the global coordinate axis
h	aperture height
m	media (name or number)
Y,Z or Z,X or X,Y	output coordinate direction

Remarks

- RIGHT and PENTA create the specified 90-degree deviation prism with aperture hight **h'**. and medium **m**, and output coordinate direction given by the last entry. RIGHT creates a right-angle prism; whereas, PENTA creates a five-sided prism.
- The output coordinate direction establishes the orientation of the prism by specifying the output direction of a ray that originally entered the prism propagating along the input coordinate axis.
- All faces are circular.
- The best way to model any prism in ASAP is to write a macro that creates a faceted object from two EDGES (see supplied UTIL.LIB).
- This lens entity starts out normal to the defined global coordinate axis (X, Y or Z).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

RMS (ASAP Command)

Scatter model given surface variations for RMS, fall-off, and autocorrelation.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
RMS r l s
```

Option	Description
r	RMS height variation of the surface in WAVELENGTH units
l	autocorrelation length of the surface variations in WAVELENGTH units
s	asymptotic fall-off with angle from specular (0=Lambertian, otherwise typically between -1 and -2.5)

Remarks

- The B-Bo shoulder point is the ray's wavelength divided by **l**.
- The **s** is the asymptotic fall-off with angle from specular (0=Lambertian, otherwise typically between -1 and -2.5). This is an approximate (violates reciprocity) theoretical model primarily for smooth surfaces (**r** much less than a wavelength and **l** much greater than a wavelength), but it otherwise behaves well (although it may not represent any actual rough surface).
- The BSDF varies with wavelength, incidence direction, scatter direction, and the specific object's INTERFACE properties. It also automatically conserves energy (within standard statistical variations) if a ROUGHNESS **r l** command is also assigned to the object.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MODELS

Scatter Models Overview

ROTATE (ASAP Command)

Rotates an entity about a point.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities
Define/Modify Lens Entities
Create/Modify Objects
Modify Ray/Beam Data

First Syntax:

```
ROTATE X d [ y z ] [ LIST ]
       Y d   z x
       Z d   x y
```

Second Syntax:

```
ROTATE d ABOUT a,b,c [ x y z ] [ LIST ]
```

Option	Description
X, Y or Z	rotation axis
d	rotation angle measured in degrees
y z (z x or x y)	displacement from coordinate axis
LIST	decodes transformation matrix into simple operations (if possible) and prints
a,b,c	arbitrarily oriented axis direction
x y z	point about which entity is rotated

Remarks

- Rotates the entity through an angle **d** about an axis displaced from the coordinate axis.
- The sign of the angle is determined by the right-hand rule where the positive value results when the thumb points along the positive direction of the rotation axis.
- The center of rotation is defaulted to the entity's reference point.
- To rotate **d** degrees about an arbitrary vector (**a,b,c**) passing through a point (**x,y,z**), use the second syntax.
- Group **ROTATE** with these commands: MATRIX; SHIFT; SCALE; SKEW; PLACE; ALIGN; XEQ

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

ROUGHNESS (ASAP Command)

Assigns a roughness to the entity of an object, and conserves energy between specular and scattered beams.

Function

Create/Modify Objects

Syntax

```
ROUGHNESS [ r [ l ] ] [ MODEL  i             [ FRACTION [ f ] ] ]
                        RANDOM h [ s [ m ] ]
```

Option	Description
`r`	RMS micro-roughness
`l`	autocorrelation length
`MODEL I`	specifies BRDF scattering model **i**
`RANDOM`	flag for height and slope variations
`h`	RMS random height variation
`s`	RMS normal deviations
`m`	probability distribution for normal

Remarks

- Assigns macroscopic, random height, and normal deviations to the object entities.
- **ROUGHNESS r** assigns an RMS micro-roughness to object surface(s) in wavelength units; use it to transfer energy out of the specularly reflected/refracted beams (and usually into any scatter beams) to conserve energy between specular and scattered beams. It does not affect the optical path lengths or directions of the specular beams.
- If the autocorrelation length **l** (also in **WAVELENGTH** units) is given, the effects of shadowing at high angles of incidence are included.
- For a **FRACTION f** (default 1 or given **MODEL TIS**) of the incident rays, **RANDOM** assigns a macroscopic height variation **h** (defaulted to 0) is a Gaussian distribution with an RMS value entered in WAVELENGTH units and slope error **s** in radians (both defaulted to 0) to the surface(s) of the object. The random height variations affect only the position of a point on the object and, therefore, the optical path lengths of any reflected or refracted beams, while the slope errors affect only the normal (and thus the beam directions).
- The **RANDOM** normal deviations **s**, entered as an RMS value in radians, affect only the normal and, therefore, the beam directions. The maximum allowable normal deviation is 0.2 radians (ASAP/Pro). If **s** exceeds about one fifth (0.2 radians), then unexpected raytrace results may occur at the surface, that is, wrong side warnings may be generated because, for example, a ray may randomly reflect into the surface.
- For near normal incidence rays, if the RMS slope error exceeds about one fifth (.2), unexpected raytrace results may occur at the surface; for example, "wrong side" warnings, because a ray may randomly reflect or refract into the surface. For near grazing incidence rays, the RMS slope may have to be much smaller than this to avoid these raytrace errors. Both errors are, by default, generated according to an approximately Gaussian-normal distribution (type 2 below). However, the slope error distribution function can be any one of the following:

m	Slope distribution	(Maximum/RMS)^2	Equivalent
-3	Two deltas	1	RAN(-15)
-2	Lambertian		
-1	Ramp	2	RAN(-1)
0	Uniform	3	RAN(0)
1	Triangular	6	RAN(2)
2	Gaussian-like	9	RAN(3)
3	Cosine	5	RAN(1)
4	Near-Gaussian	15	RAN(5)
5	Gaussian	2 In (2^32)	RAN(15)

- To be more precise, if the randomly unperturbed normal points along the **z** axis, the components of the randomly perturbed normal are:

$$\left[s\ \cos(u),\ \mathrm{s}\ \sin(u), \sqrt{1-\mathrm{s}^2}\right];\ -1\ < s\ < 1 \text{ and } -\pi\ < u\ < \pi$$

where **u** is a uniformly distributed random number, and **s** is a random number determined by the given slope distribution (RMS and probability function).

- Note the maximum absolute slope error is always limited to one; that is, 45 degrees.
- Alternatively, the surface slopes may be randomized such that the shape of the normal incidence reflected pattern matches that of the BRDF specified by the scattering **MODEL i**.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

BSDF Fit Utility

OBJECT

ROUNDED (ASAP Command)

Creates a rounded-corner, rectangular edge/curve.

Function

Define/Modify Curvedge Entities

Syntax

```
ROUNDED X x y z r [ n a a' ]
        Y y z x
        Z z x y
```

Option	Description
`X, Y or Z`	specifies the axis of symmetry
`x, y or z`	location along coordinate axis
`y z, z x, or x y`	semimajor widths of the oval
`r`	radius of the rounded corners
`n`	number of points (or segments) on the oval (default 16)
`a a'`	angular range (in degrees from first semimajor axis) over which the oval is defined (default is 0 to 360 degrees)

Remarks

- The default number of points along the edges/curves of the rectangle is 16.
- The default angular range over which the rectangle is defined is 0 to 360 degrees.
- If **n**, **a** and **a'** are specified, they become the default settings for most future EDGE commands.
- This edge is made up of coplanar straight line segments, that is, convex polygons whose vertices lie on a particular curve.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

ELLIPSE for description of other parameters

SAG (ASAP Command)

Deforms or sags an edge onto a surface.

Function

Define/Modify Curvedge Entities

Syntax

```
SAG [ X ] i  [ ABS ]
          Y    REL
          Z  [ CV c [ c'] ] [ CC k [ k'] ] [ AD d [ d'] ] [ VP p [ p'] ]
               RD r   r'
               SPHERE r [ q q' [ p p' ] ]
```

Option	Description
X Y Z	sag direction
i	**i**th previously defined surface/function
ABS	sag the edge ABSsolutely (the default) for all points
REL	sag the edge RELatively for any interior control points
CV	curvature sag flag
c c'	vertex curvature
CC	conic constant sag flag
k k'	conic constant
AD	fourth-order deformation sag flag
d d'	fourth-order deformation coefficients
VP	vertex offset sag flag
p p'	vertex offset
RD	radius sag flag
r r'	vertex radius
k k'	conic constant
SPHERE	sphere sag flag
r	distance from the edge's reference point (or the given vertex point p p') to center of sphere
q q'	maps remaining two coordinate directions to the surface
p p'	vertex offset

Remarks

- Deforms the last EDGE specification by sagging the points on the edge.
- The actual sag can be specified in three different ways:

1. The edge can be sagged to the previously defined ith SURFACE. This can be ABSsolutely (the default) for all points or RELatively for any interior control points. The latter retains the shapes of individual segments of the edge.
2. The sag can be specified independently via a standard optical sag formula. The **c**, **r**, **d**, **k** and **p** specify the vertex curvature, vertex radius, fourth-order deformation coefficient, conic constant, and vertex offset respectively, of the first semimajor axis component. The primed quantities refer to the second axis and are defaulted to the same values as the first. The default for the conic constant **k** is -1, that is, a parabola.
3. All the edge points can be exactly sagged to a **SPHERE**. The center of the sphere is located a distance **r** from the edge's reference point (or the given vertex point **p p'**) along the given coordinate axis. The mapping of the other two coordinate directions to the surface of the sphere is controlled by the **q** and **q'** parameters. A value of zero performs a latitude mapping; a value of one performs a longitude mapping. The parameters can be varied continuously between these two cases.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT (ASAP Command)

SAMPLED (ASAP Command)

Creates an explicit surface interpolated from sampled data.

Function

Define/Modify Surfunc Entities

First Syntax (use an existing data distribution file):

```
SAMPLED [ name ]
```

Second Syntax (create the file directly):

```
SAMPLED name X x y y' z z' [ SLOPES ]
             Y y z z' x x'   DIFFS
             Z z x x' y y'
d [ s' s" ] d' ...
   :
```

Option	Description
`name`	distribution file name of sampled data
`X, Y or Z`	specifies axis of symmetry
`x, y or z`	location along axis
`y y', z z', or x x'`	maximum data extents in specified direction
`z z', x x', or y y'`	
`d d' ...`	deformation values
`SLOPES`	specifies slope data
`s' s"`	slope values in direction of maximum data extents
`DIFFS`	estimates slopes based on finite differences from finite deformations only

Reference Point

If read from a distribution file, **name.dis** file (default BRO009.DAT), point is at (**0**,**0**) depth coordinate.

If created directly, point is at the intersection of reference plane and coordinate axis.

Surface Normal

Along positive coordinate direction

Autolimiting

Yes, see Remarks

Remarks

- Creates an explicit surface defined as follows:
  1. By reading deformation samples found in the binary distribution file **name.dis** (default BRO009.DAT) external file.
  2. By entering the sample points following the command description.
- The deformation and slopes for ray tracing, if specified in the distribution file, are linearly interpolated between data points. Up to 1000 data points are allowed in the across direction and an unlimited number in the down direction.
- The file header of the distribution file should contain directional labels (**X**, **Y**, and **Z**) and data ranges that determine the orientation and size of the surface in its local coordinate system. If not specified, ASAP assumes a **z(x,y)** surface located at the origin.
- The distribution file can be created by an external user program, a $ITER command, or by a MAP command.
- If the sample points are entered directly in ASAP, then (using **X** as an example) **name.dis** is the name of the distribution file created from the sample points, **X** is the axis of symmetry, and **x** is the location of the reference plane along the axis. The **y,y'** and **z,z'** are the maximum extents of the sample points. The following lines specify the deformation values that cover the given area uniformly up to (and including) the given extents.
- If the **SLOPES** option is included, each sample point has a deformation value and the slopes in the two other directions associated with it.
- Uses smooth positional interpolation with slopes and ACCURACY HIGH.
- If **DIFFS** is specified, the slopes are estimated from the deformation values by finite differences.

- The **SAMPLE** entity is autolimiting, but only in a rectangular sense because of the data file structure. If an elliptical aperture is desired, you must use a LOCAL or LIMITS modifier.

Example

For defining an approximation to a shallow conical surface:

```
$ITER  X -1. 1. -11  Y -1. 1. -11  Z !negative sample numbers required
  Z=.1*SQRT(X^2+Y^2)                 !z proportional to radial value
SURFACE; SAMPLED ITER     !default file name when $ITER not in a macro
  LOCAL AXIS Z    !convert box from default rectangular to cylindrical
  SHIFT Z ...     !move it into global position
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

$ITER

MAP

LIMITS

LOCAL

SAMPLED (Cylindrical)

SAMPLED (Cylindrical) (ASAP Command)

Creates a deformed cylindrical surface interpolated from sampled data.

Function

Define/Modify Surfunc Entities

Syntax

```
SAMPLED name RX r x x' a a' [ SLOPES ]
             RY   y y'        DIFFS
             RZ   z z'
d [ s' s" ] d' ...
  :
```

Option	Description
RX, RY, or RZ	specifies axis of reference cylinder
r	radius (semidiameter) of reference cylinder
x x', y y', or z z'	initial and final axial extents of reference cylinder
a a'	initial and final angular extents of reference cylinder
d d' ...	deformation values
SLOPE	specifies slope data
s' s" ...	slope values in axial and angular directions
DIFFS	estimates slopes based on finite differences from finite deformations only

Reference Point

At the intersection of the reference cylinder and coordinate axis.

Surface Normal

Radially outward (cylindrical symmetry)

Autolimiting

Yes

Remarks

- This variation of the SAMPLED surface may be used to model sample points as a departure from a cylinder instead of a plane.
- The third and fourth entries define the reference cylinder (that is, its axis and radius). The fifth and sixth entries define its extent in the axial direction. The seventh and eighth define the angular (degrees) extent around the axis. The command is then followed by the lines of deformation values that cover the given area uniformly up to (and including) the given extents.
- If the sample points are entered directly in ASAP, using **RX** as an example, **name.dis** is the name of the distribution file created from the sample points. **RX** is the axis of the reference cylinder, **x x'** is the initial and final axial extents of the reference cylinder, and **a a'** is the initial and final angular extents of the reference cylinder. The lines that follow specify the deformation values that cover the given area uniformly up to (and including) the given extents.
- If the **SLOPES** option is included, then each sample point has a radial deformation value and the slopes in the axial and angular directions associated with it.
- The angular slope has units of length per radian.
- If **DIFFS** is specified, then the slopes are estimated from the deformation values by finite differences.
- The deformation values for a cylindrical sampled surface must cover the area of surface uniformly up to and including the given extents.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SAMPLED

SAVE (ASAP Command)

Writes future ray trace data to a file for post-processing.

Function

Save or Recover System Data and Control Execution

Syntax

```
SAVE [ k [ name ]]
       OFF
```

Option	Description
k	unformatted binary file number (or extension)
name	unformatted binary file name
OFF	turns off selection of saving ray trace data (data is not saved)

Remarks

- Directs ASAP to save an unformatted binary file number information about every ray intersection found during any future ray traces.
- If **k** is present, it becomes the file extension; otherwise, "his" is the assumed extension, and is automatically incremented after each **TRACE** command, unless another **SAVE** command is used to override it or turn it **OFF**.
- The default name part of the file is "rayinter" or it is taken from the last **SYSTEM** or **$IO** command. Otherwise, it can be directly specified with the additional "name" entry.
- The next **HISTORY** or **RENDER…RAYS** command uses the file.
- After program completion, it can be processed by another program such as STRAP. Each of the fixed length records in the file corresponds to a one-ray intersection and contains the following six or 12 words (24 or 48 bytes) of information. The first word (0) is a signed integer, all others are floating point:

Words	Content
0	Previous record number
1	Object number (negative if starting or split object)
2	Incident power/flux of ray (negative if scattered ray)
3 to 5	Global X,Y,Z coordinates of ray on object
6	Cross-sectional size (height on surface) of beam centered on ray
7	Average radius of curvature of beam wavefront
8 to 10	X,Y,Z components of unit normal to object surface
11	Square root of the cosine of the incidence angle from normal

-
- The short six-word record is used for a **SAVE** command with no additional entries or if XMEMORY is set to **MIN**. The long 12-word record is the minimum amount of information needed by the STRAP module for calculating stray light paths.
- Currently, SAVE takes precedence over the TRACE PLOT command to prevent excessive disk thrashing.

Example

The following code is a sample program for reading a SAVE file and listing the results:

```
PROGRAM DUMPS
*
* dumps basic SAVE file information to text
*
PARAMETER (LNRECD=1) !Digital/Compaq Fortran default
! PARAMETER (LNRECD=4) !otherwise
DIMENSION RADATA(11)
```

```
! REAL NUMREC !ASAP versions 6.0 or earlier
CHARACTER*100 FIL
LOGICAL EX
* ask for file
WRITE(*,*) 'Enter file name:'
READ(*,'(A)') FIL
* determine record length from OS
LENREC=0
INQUIRE(FILE=FIL,EXIST=EX,RECL=LENREC)
IF (.NOT.EX) STOP 'File does not exist'
IF (LENREC.EQ.0) THEN
* or from "zeroth" record
OPEN(1,FILE=FIL,ACCESS='DIRECT',FORM='UNFORMATTED',
& RECL=LNRECD,STATUS='OLD')
READ(1,REC=1,IOSTAT=IOS) LENREC
CLOSE(1)
IF (IOS.NE.0) STOP 'Cannot read first record'
IF (MOD(LENREC,256).NE.247) STOP 'Not a BRO binary file'
LENREC=LENREC/256
ENDIF
* check record length and open file
LENREC=LENREC/4
IF ((LENREC.NE.6).AND.(LENREC.NE.12)) STOP 'Wrong record size'
OPEN(1,FILE=FIL,ACCESS='DIRECT',FORM='UNFORMATTED',
& RECL=LENREC*LNRECD,STATUS='OLD')
* get number of records
READ(1,REC=2,IOSTAT=IOS) NUMREC
IF (IOS.NE.0) STOP 'Error reading second record'
N=NUMREC+1
L=LENREC-1
* loop through records
DO J=2,N
READ(1,REC=J,IOSTAT=IOS) NUMREC,(RADATA(I),I=1,L)
IF (IOS.NE.0) STOP 'Error reading all the records'
WRITE(*,*) NUMREC,(RADATA(I),I=1,L)
ENDDO !J
CLOSE(1)
END
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

$IO
SYSTEM
TRACE PLOT

SAWTOOTH (ASAP Command)

Creates a sawtooth pattern edge in the plane.

Function

Define/Modify Curvedge Entities

Syntax

```
SAWTOOTH X x y z  y' z' n [ w ]
         Y y z x  z' x'
         Z z x y  x' y'
```

Option	Description
X, Y or Z	coordinate axis of plane
x, y or z	location of plane on coordinate axis
y z, z x, or x y	first tip (or control point)
y' z', z' x', or x' y'	end of the first tooth
n	number of teeth
w	positive Bezier weight factor used to get rounded conic teeth

Remarks

- The edge starts at the origin in the plane.
- Instead of sharp pointed teeth, a positive Bezier weight factor **w** can be used to get rounded conic teeth (See the POINTS command for more information).
- SAWTOOTH Bezier weights are positive real numbers ranging from 0 to 1000 (any larger number has little effect and can lead to numeric errors).
- This edge is a combination of coplanar straight line and higher-order curved segments.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

SCALE (ASAP Command)

Specifies arbitrary scaling of an entity.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities
Define/Modify Lens Entities
Create/Modify Objects
Modify Ray/Beam Data

Syntax

```
SCALE a [ b c ] [ LIST ]
      X a [ x ]
      Y b   y
      Z c   z
```

Option	Description
a b c	Scales the entity by the uniform factor "a", or nonuniformly by separate coordinate factors (a,b,c)
x y z	optional point (x,y,z)
LIST	decodes transformation matrix into simple operations (if possible) and prints

Remark

- Scales the entity by the uniform factor "a", or nonuniformly by separate coordinate factors (a,b,c), about the optional point (x,y,z). This point is defaulted to the entity's reference point.
- ASAP scales relative to the original centroid as the reference point. Simple statistical noise can therefore create large variations in x, y, z positioning. When scaling sources centered about 0, 0, 0, it is best to specify x, y, z as 0, 0, 0 to fix their position.
- The default for (**a,b,c**) is 1 (no scaling).
- When scaling rays anamorphically (unequal scaling in one or more directions), the angular direction cosines of each ray is inversely scaled. An isotropic or Lambertian source do not remain isotropic or Lambertian after anamorphic scaling.
- Group SCALE with these commands: MATRIX; ROTATE; SHIFT; SKEW; PLACE; ALIGN; XEQ

CAUTION

Do NOT use a nonuniform SCALE command to resize raysets created using; for example, EMIT DATA. If you do, the angular distribution of the rayset is altered (see above). BRO recommends using the NORMALIZE command to rescale the axes of the distribution data file before creating rays.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT
SCALE FROM (ASAP Command)
EMITTING DATA (ASAP Command)
NORMALIZE (ASAP Command)

SCALE FROM (ASAP Command)

Scales from the given length units to the current system UNITS.

Function

Modify Ray/Beam Data

Syntax

```
SCALE FROM units [ LIST ]
```

Option	Description
`units`	given length units
`LIST`	decodes transformation matrix into simple operations (if possible) and prints

Remark

- Scales from the given length **units** to the current system UNITS.
- For allowable **units** strings See the UNITS command.
- If the system UNITS are not set, they are set to **units** and no scaling takes place.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT
SCALE
UNITS

SCATTER RANDOM/MODEL/TOWARDS (ASAP Command)

Assigns a scattering interface to an object and specifies importance area or direction sampling for scattered rays.

Function

Create/Modify Objects

First Syntax

```
SCATTER [ MODEL  m       ] [ RANDOM [ r' n  ] ] [ ABS r t ]
          MODELS m m'                             REL
```

Second Syntax

```
SCATTER MODEL  m
        MODELS m m'                               REL
        [ TOWARDS [ EDGE ] i    [ n' f t a" [ REL] ]
                           POINT
                           X
                           Y
                           Z
                          -X
                          -Y
                          -Z
                           SPEC
                           REFL
                           TRAN
        [ TOWARDS … ]
           :
```

Option	Description
MODEL m	assigns scatter model m to the interface
MODEL m m'	assigns reflective **m** and transmissive **m'** scatter models to the interface
RANDOM	specifies Lambertian scatter input
r'	total hemispherical reflectivity
n	number of randomly distributed scatter rays to be generated (default value is 1)
ABS or REL	specifies absolute or relative scatter
r t	separately scales the magnitude of the reflective/transmissive components; required with **INTERFACE** and **LEVEL ALL** for 2-way scattering; see **TOWARDS** syntax
TOWARDS	specifies an importance edge or direction towards which rays are scattered; required for specifying a **SCATTER MODEL m**
EDGE	optional edge comment
i	importance area **EDGE** number
X, Y, Z, -X, -Y, or -Z	importance coordinate direction
SPEC	importance specular direction
n'	number of random rays scattered towards importance area or direction
f	fractional upper importance area or direction band
t	fractional lower importance area or direction band
a"	scattered ray divergence half-angle threshold
REL	specifies cone angle importance area or direction

Remarks

- Assigns incoherent random diffuse scattering properties to an interface and may be used to simulate Lambertian scatter at the interface (equally bright in all directions).

- If reflective **m** and transmissive **m'** scatter models are different, specify both.
- The **RANDOM** option may be used with the **MODEL** option to generate a seed ray; that is, a ray propagating in a direction other than the importance sampling edge, to keep the ray trace going. All rays generated by **RANDOM** have the same flux.
- To produce scattered rays, **SCATTER MODEL m** must be followed by the **TOWARDS** modifier.
- More diffuse scatter rays directed **TOWARDS** importance areas can be specified with additional lines of input (up to 10 per object).
- The **RANDOM** option does not require the modifier **TOWARDS**, but BRO recommends it since it is more efficient to scatter **TOWARDS** an importance sampled edge.
- Using the MODELS command requires non-zero specular reflection and transmission coefficients on the INTERFACE command. If the surface has only transmissive and reflective scatter, use INTERFACE 1E-15 1E-15 to eliminate the specular rays while still allowing two-way scatter.
- While the **LEVEL** command is in effect, the **RANDOM** option causes ASAP to generate a set of randomly distributed diffuse scatter rays for each incoming ray on the object.
- The **r'** is the additional total hemispherical diffuse reflectivity of the scattering surface. The **r'**=1 corresponds to white Lambertian.
- The **n** is the number of rays to be scattered into the hemisphere centered on the ray intersection point.
- No scattered rays are generated unless the LEVEL ALL command has been turned on.
- If both reflective and transmissive (or multiple diffraction orders) scatter rays are generated (See the **LEVEL ALL** command), the specified scatter can be separately scaled in magnitude for the two components by the given **r** and **t** factors. The scatter specified can also be either **ABSOLUTE** (referenced to incident power) or **REL**ative to the specular power. The default is ABS 1 1 (except for **SCATTER RMS**, which is REL 1 1).
- The **TOWARDS** modifier specifies an importance edge or direction towards which rays are scattered.
- If an **EDGE** or a number follows the **TOWARDS** option, the absolute value of **i** entries is the numbers of the edges used as importance areas. If **i** is entered as a positive number, radiation is scattered towards the designated real area; if **i** is negative, radiation is scattered away from the designated virtual area.
- The entry **n'** is the number of randomly directed rays scattered into the importance area or importance direction by each **TOWARDS** command. In most cases the value of **n'** should be set to a value larger than

$$N_0 = \frac{\Gamma}{\pi a''^2}$$

where Γ is the projected solid angle subtended by the importance area or direction. The default value for a" is 0.1 radians when PARABASAL is otherwise greater than zero and $\pi/2$ radians. See the last technical note for the definition of Γ and a discussion of how the flux assigned to each ray depends on **n'**, Γ, and **a"**.

- The **f** and **t** are the fractional lower and upper bounds (default 0 to 1) of the importance area relative to the defining edge. ASAP does not take into account importance areas whose solid angles overlap to maintain conservation of energy.
- Instead of defining an importance area with a polygon (edge), an importance direction can specified as lying between the surface normal and the following:

Entry	**Description**
X,Y,Z,-X,-Y,-Z	Global coordinate axis direction
REFL,TRAN,SPEC	Reflected, transmitted, or either specular direction
POINT I	Direction from ray point to entity "i" reference point

In these cases, **f** is the cone half-angle in radians of the solid angle centered on the direction. The **t** is a parameter that varies the scattering direction from the surface normal (t=0) to the specified direction (t=1, default). A value of t=-1 corresponds to the retroREFLection.

- The parameter **a"** is the half angle of a scattered ray divergence threshold. It should be entered with units of radians. Its default value is $\pi/2$ radians for PARABASAL 0 and 0.1 otherwise. If the **REL** option is used, **a"** is unitless and the divergence threshold angle is equal to **a"** times the half angle subtended by the importance area or direction. See the next remark for a discussion of how the divergence angle threshold affects the flux assigned to the scattered rays generated by each **TOWARDS** command.
- The flux assigned to each scattered ray depends on the projected solid angle subtended by the importance area. The projected solid angle is defined as

$$\Gamma = \int_{\Omega} \sin(\theta)\cos(\theta)\,d\theta d\phi$$

where

θ and φ

are polar and azimuthal angles with respect to the surface normal, and symbolizes integration over the solid angle of the importance area. If

Ω

is a full hemisphere, then

$$\Gamma = \Pi$$

Define a parameter

$$N_0$$

as

$$N_0 = \frac{\Gamma}{\pi a''^2}$$

where **a''** is the user specified divergence angle threshold that is entered on the **TOWARDS** command (the default value for **a''** is 0.1 radians when PARABASAL is otherwise greater than zero and $\pi/2$ radians).
When scattering towards an importance edge, if the number of scattered rays (**n'**) entered on the **TOWARDS** command is greater than N_o
then the flux assigned to each scattered ray is

$$F_s = F_o \rho(\theta_i, \phi_i; \theta_s, \phi_s) \frac{\mathrm{A}\cos(\theta_e)\cos(\theta_s)}{n' R^2}$$

where F_s

is the flux of the scattered ray, F_o is the flux of the incident ray,

$$\rho(\theta_i, \phi_i, \theta_s, \phi_s)$$

is the BSDF of the surface that is appropriate to the incident

$$(\theta_i, \phi_i)$$

and scatter

$$(\theta_s, \phi_s)$$

directions (which are defined with respect to the surface normal) of each ray, **A** is the area of the importance edge, θ_e
is the angle between the scattered ray and the normal to the importance edge, and **R** is the distance from the surface to the importance edge as measured along the path traveled by the scattered ray. If the number of scattered rays is less than N_o
all scattered rays are assigned the same flux, and this flux is equal to

$$F_s = F_o \rho(\theta_i, \phi_i; \theta_a, \phi_a) \frac{\Gamma}{n'}$$

where

$$(\theta_a, \phi_a)$$

is the direction to the center of the importance area.

- When scattering towards an importance direction (from example, with **TOWARDS SPEC**), each scattered ray is assigned a flux

$$F_s = F_o \rho(\theta_i, \phi_i; \theta_s, \phi_s) \frac{\Gamma}{n'}$$

when the number of rays is greater than N_o
If the number of scattered rays is set to a value greater than *n*, the **TOWARDS SPEC** command may be used to scatter into a full hemisphere. The TIS is correct if enough rays are used (substantially more than N_o may be needed if high accuracy is required or if the BSDF is sharply peaked or varies rapidly with direction). In most cases, the number of scattered rays is **n**, which should be set to a value that is larger than **n** to obtain accurate results.

Example of Scatter Interface and the Power Calculation

```
--- PRINT 1
```

```
SURFUNC    2: Degr=   2  (ELLIPSOI) X 0.000000     Y 0.000000     Z 0.000000
 Local Box X -1.010     1.010    Y -1.010     1.010    Z -1.010     1.010
            width= 2.020          width= 2.020          width= 2.020
 This entity used by objects   1
 Polynomial Coefficients:
 C -1.000000000      x2     1.00000      y2     1.00000      z2     1.00000

SURFUNC    3: Degr=   2  (ELLIPSOI) X 0.000000     Y 0.000000     Z  1.00000
 Local Box X -.2020E-1 0.2020E-1 Y -.2020E-1 0.2020E-1 Z -.2020E-1 0.2020E-1
            width=0.4040E-1       width=0.4040E-1       width=0.4040E-1
 This entity used by objects   1
 Polynomial Coefficients:
 C -1.000000000      x2     2500.00      y2     2500.00      z2     2500.00

OBJECT   1: INTEGRATING_SPHERE
 Physical Surface    2 (ELLIPSOI)
       Box X -1.030     1.030    Y -1.030     1.030    Z -1.030     1.030
            width= 2.060          width= 2.060          width= 2.060
 Bounding Entities    3
   Random (MonteCarlo) Scatter:
     Full Hemispherical    1 0.88000     0.10000
     Important Area   1      1 0.00000      1.0000     0.50000

--- TRACE GRAPH; CONSIDER ONLY DETECTOR
      3667 ray-objects encountered  at    121 per second
     11001 order-intersections done at    364 per second

--- WINDOW Y -1 1 X -1 1; SPOTS POS ATTRI 0

Distribution of data within:
 Across or Vertical: Y = -1.00000     to  1.00000      ( 2.00000    )
 Down or Horizontal: X = -1.00000     to  1.00000      ( 2.00000    )

Opening NEW     distribution file   9: BRO009.DAT
  Rays  Flux
   625 0.691418E-03
MINIMUM (m) =   0.000000     MAXIMUM (M) =   0.401746

--- $REG IPOW 'IDEAL INTEGRATING SPHERE CALCULATED POWER'
IDEAL INTEGRATING SPHERE CALCULATED POWER .733333333333334E-03
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

LEVEL
SCATTER RMS/SCATTER BSDF
SCATTER REPEAT

SCATTER REPEAT (ASAP Command)

Assigns the scatter characteristics of a given object to the current object.

Function

Create/Modify Objects

Syntax

```
SCATTER REPEAT [ i ]
```

Option	Description
`i`	object number

Remarks

- Assigns the scatter characteristics of object **i** (or the previous object) to the current object.
- If "i" is zero, then all the scatter properties are removed.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SCATTER RANDOM/MODEL
SCATTER RMS/SCATTER BSDF

SCATTER RMS/BSDF (ASAP Command)

Assigns a near-specular scattering interface to an object.

Function

Create/Modify Objects

Syntax

```
SCATTER [ RMS b [ s a ]
          BSDF
```

Option	Description
`RMS or BSDF`	specifies incoherent near-specular scatter input
`b`	either the RMS microroughness of the surface in wavelength units, or actual BSDF in inverse steradians at 0.573 degrees from specular
`s`	asymptotic falloff of surfaces PSDF (also the BSDF) in the direction cosine space, typically a number between -1 and -2
`a`	back scatter divergence angle

Remarks

- Assigns incoherent, near-specular scattering properties to an object, thereby simulating the type of scatter commonly seen on a smooth optical surface.
- Since the **SCATTER RMS** command requires the reflection and transmission coefficients, an INTERFACE command must be executed first.
- If scattered ray/beam generation has been turned on by the LEVEL command, ASAP generates a diverging near-specular beam at any interface with a non-zero **b** value.
- If the cone half-angle in radians **a** is non-zero, ASAP generates a back-scattered beam that propagates back along the incoming direction of the ray with a half-angle divergence of **a** radians in addition to the forward-scattered beam. The scatter in this direction is calculated using the **b s** or **m** parameters.
- If **a** is negative, only the back-scattered beam is generated at the interface; the forward-scattered beam is suppressed. The scatter in this direction is calculated using the **b s** or **m** parameters.

CAUTION

Do NOT use these commands to assign a Harvey model to an object; use MODEL, HARVEY, and SCATTER MODEL for this.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

- HARVEY
- INTERFACE
- LEVEL
- MODEL
- SCATTER RANDOM/MODEL
- SCATTER REPEAT
- SPREAD NORMAL

SEARCH (ASAP Command)

Sets the local and global object intersection strategy.

Function

Setup Trace

Syntax

```
SEARCH [ LIST ]
       [ ALL        FORWARD  ]
         SEQUENTIAL BACKWARD
i         j         j'         j"
      :
```

Option	Description
`LIST`	lists SEARCH settings
`ALL`	all objects are candidates for intersection
`SEQUENTIAL`	searches objects sequentially
`FORWARD`	searches objects forward
`BACKWARD`	searches objects backward
`i j j' j"`	on object i, searches objects j through j' in steps of j"

Remarks

- Controls the way in which the rays are traced through the system by specifying the set of objects to be considered as the possible nearest intersections for rays leaving an object.
- Additional entries on the **SEARCH** command itself apply to all objects.
- For any **ALL** input, the current object itself is considered for the next intersection with the rays.
- The range of possible objects can be selectively set by additional commands; the first entry **i** is the number for the particular object followed by a pair of entries.
- **SEARCH** must be executed after all objects are defined, or ASAP issues an error message.
- **SEARCH** sets the object intersection strategy for the objects defined at the time the command is executed. If additional objects are added to a system model, another **SEARCH** command must be entered.
- The **LIST** option prints the current search settings for each object.
- With **m** being the largest possible object number, the various options for ray intersection after **OBJECT i** are tabulated according to the following table.

Input	Object Range		Increment
`j j' j"`	j	j'	j"
`ALL`	1	m	1
`ALL FORWARD`	i	m	1
`ALL BACKWARD`	i	1	-1
`SEQUENTIAL`	i-1	i+1	2
`SEQUENTIAL FORWARD`	i+1	i+1	
`SEQUENTIAL BACKWARD`	i-1	i-1	

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

ALLOWED

SECTION (ASAP Command)

Prints or transfers the current distribution data to/from variables.

Function

Display/Modify Energy Distributions

Syntax

```
SECTION [ m m' [ n n' ] ] [ GET ]
                            PUT
```

Option	Description
m m' n n'	range of pixels to be printed or transferred
GET	flag to transfer distribution data into variables
PUT	flag to put variables into distribution data

Remarks

- Prints a table of current distribution data or transfers the current distribution data to or from an array of variables.

Print, **GET** into registers or **PUT** from registers the section of data:

f(i,j): i=m to m', j=n to n'

- If the **GET** or **PUT** options are used, then the first data value is associated with register A0.
- If m'-m > 8 and n'-n > 24, then the maximum allowable intrinsic register section is transferred:

$$\begin{array}{c} A0 \Leftrightarrow f(m,n) \ldots A9 \Leftrightarrow f(m+9,n) \\ \vdots \qquad\qquad\qquad \vdots \\ Z0 \Leftrightarrow f(m,n+25) \ldots Z9 \Leftrightarrow f(m+9,n+25) \end{array}$$

- Historical note: the SECTION command was the TABLE command in pre-ASAP 5.1 versions.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
GET
PUT

SEED (ASAP Command)

Initializes the seed for the random number generator.

Function

Setup Trace

Syntax

```
SEED [ n ] [ QUASI ]
             OFF
```

Option	Description
n	seed number (default =2000000001)
QUASI	use a quasi random number sequence
OFF	cease using a quasi random number sequence

Remarks

- Initializes the seed value for the random number generator used by the ROUGHNESS RANDOM, GRID RANDOM, SCATTER RANDOM/TOWARDS and EMITTER commands.
- The **n** should be some large odd integer and is set to 2000000001 at program startup.
- If a zero is entered for **n**, the QUASI number generator is no longer random and returns a fixed value of 0.5 every time.
- For the EMITTERS, a **QUASI** random number sequence can be used instead. In this case, only the least significant digits of **n** are used to select between 50 possible sequences.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EMITTING BOX/SPHEROID
EMITTING CONE/PYRAMID
EMITTING DATA
EMITTING DISK/RECTANGLE
EMITTING ENTITY or OBJECT
EMITTING FILAMENT
EMITTING HELIX
EMITTING IES
EMITTING RAYS
GRID DATA RANDOM
GRID ELLIPTIC RANDOM
GRID HEX RANDOM
GRID OBJECT RANDOM
GRID POLAR RANDOM
GRID RECT RANDOM
GRID WINDOW RANDOM
ROUGHNESS RANDOM
SCATTER RANDOM

SEGMENTS (ASAP Command)

Controls number of segments plotted per arc.

Function

Setup Plots and Verify System

Syntax

```
SEGMENTS [ n ]
```

Option	Description
n	number of straight line segments used to approximate a 45 degree arc

Remarks

- Controls the number of straight-line segments used to draw conic arcs.
- The default number of line segments is 3. The minimum and maximum numbers are 1 and 45, respectively.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PLOT EDGES
PLOT FACETS

SELECT (ASAP Command)

Isolates a ray set for further analysis.

Function

Modify Ray/Beam Data

Syntax

```
SELECT [ ALL ]
         ONLY    [ entry entry' [ AND entry entry' ...
         EXCEPT                   OR
```

Option	Description
ALL	selects all defined rays
ONLY	selects only the specified rays
EXCEPT	selects all rays except the specified rays
AND OR	logical operators
entry entry'	see Remarks

Remarks

- Gives the user control over the current set of rays ASAP is to consider in all calculations and output (similar to the CONSIDER command for objects).
- No entries or the **ALL** option forces consideration of all known rays.
- Particular ray sets are selected by entering pairs of entries separated by a logical operator (AND/OR) according to the following table:

entry	entry'	
i	**j**	Ray number is between **i** through **j** inclusive
OBJECT	**n**	Ray comes from previous object **n** (name or number)
OBJECT	**-n**	Ray was scattered from object **n**
OBJECT	**+n**	Ray was split from object **n**
SOURCE	**k**	Ray originated from source number **k**
SOURCE	**-k**	Scattered ray from source number **k**
MEDIA	**m**	Ray is in MEDIA **m** (name or number
MEDIA	**-m**	Scattered ray is in MEDIA m
GENERATION	**n**	Ray was split and scattered **n** times
GENERATION	**-n**	Ray was scattered **n** times
GENERATION	**+n**	Ray was split **n** times
EVERY	**n**	Ray number modulo **n** equals one.
HITS	**n**	Ray has hit objects **n** times
	-[n]	Ray has hit objects **n** times and has not yet refracted/reflected with last object
	+[n]	Ray has hit objects **n** times and has refracted/reflected with last object
PATH	**l**	Ray belongs to **l**th path from last PATHS command
PATH	**0**	Ray belongs to a path not listed by last PATHS command

w	**W**	Ray has wavelength greater than **w**
W	**w**	Ray has wavelength less than **w**
f	**F**	Ray has flux greater than **f**
F	**f**	Ray has flux less than **f**
d	**L**	Ray has optical path length greater than **d**
L	**d**	Ray has optical path length less than **d**
r	**R**	Ray has AXIS radial coordinates greater than **r**
R	**r'**	Ray has AXIS radial coordinates less than **r'**
t	**T**	Ray has AXIS angular coordinates greater than t degrees
T	**t'**	AXIS angular coordinates less than t' degrees
x	**X**	Ray has X coordinates greater than **x**
X	**x'**	Ray has X coordinates less than **x'**
y	**Y**	Ray has Y coordinates greater than **y**
Y	**y'**	Ray has Y coordinates less than **y'**
z	**Z**	Ray has Z coordinates greater than **z**
Z	**z'**	Ray has Z coordinates less than **z'**
a	**A**	Ray has X direction cosines greater than a
A	**a'**	Ray has X direction cosines less than **a'**
b	**B**	Ray has Y direction cosines greater than **b**
B	**b'**	Ray has Y direction cosines less than **b'**
c	**C**	Ray has Z direction cosines greater than **c**
C	**c'**	Ray has Z direction cosines less than **c'**

- The sign of the entry for HITS determines whether or not the ray has interacted with the last object. A minus sign indicates the ray has not yet refracted/reflected. A plus sign or no sign indicates the ray has already refracted/reflected.
- All tests, including logical operations, are applied to each ray individually, one at a time. The ray must satisfy all specified conditions to be selected. The final number of rays selected is displayed after all rays have been tested.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SUBSET

SEQUENCE (ASAP Command)

Creates a lens composed of a sequence of conicoid surfaces.

Function

Define/Modify Lens Entities

Syntax

```
SEQUENCE [ CURV ]  [ HEIGHT ]
           RADI      DIAM
x y z a,b,c h [ r [`p'  k ['s]  o m ]
d h [ r [ `p ] k [ `s ] o m ]
  :
```

Option	Description
CURV	specifies a vertex curvature
RADI	specifies a vertex radius
x y z	global coordinates of the vertex (center) of the conicoid (long format)
a,b,c	normal vector to the conicoid at the vertex (long format)
h	aperture HEIGHT (semidiameter) or DIAMeter
r	either the vertex CURVature or RADIus
k	conic constant (0=sphere, -1=parabola, and so on)
o	central obscuration (or hole) ratio
m	number or name of medium that follows the conicoid (use -1 or REFL for a reflector)
d	distance from previous conicoid
p	first optional paired entry **p** is either a vertex CURVature or RADIus of a paraboloid that is subtracted from the conicoid

Remarks

- Creates a lens directly as a sequence of conicoid surfaces (one line of input per surface).
- Enter the lens directly as a sequence of conicoids (one line of input per surface).
- Two formats are available: a long and short. In either case, the last five entries are the same. The first conicoid of a **SEQUENCE** must be entered in the long format. The following conicoids may be entered in either the long format (**x y z a,b,c**) or short format (**d**) as appropriate.
- If the curvature is non-zero (that is, not flat), then **k** is the conic constant (for example, 0=sphere, -1=parabola). Otherwise, it is the 4th-order aspheric coefficient.
- For a reflecting surface, **m** should be **-1** or **REFL**.
- To get an aspheric "flat" (that is, corrector plate), set **p** equal to **r**. The fourth-order and higher aspheric terms are then identical to those of the series-expanded conicoid times **s**.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

LENSES

SHAPE (ASAP Command)

Sets the beam shape of the individual rays of the currently selected rayset.

Function

Modify Ray/Beam Data

Syntax

```
SHAPE k    [ s ]
      beam
      0    [ n a ] [ a' a" ... ]
      MODE
```

Option	Description
k	beam shape parameter
beam	beam shape name corresponding to k
s	optional parameter required by certain shape parameters
0 (or MODE)	specifies higher-order Hermite-Gaussian beam modes
n	number of higher order Hermite-Gaussian beam modes
a	complex amplitude of TEMoo beam mode
a' a" ...	complex amplitudes of higher order beam modes

Remarks

- Resets the beam shape to the literal **beam** or integer **k** for the currently SELECTed ray set.
- ASAP currently allows the following beam shapes (the shape is either an amplitude profile for coherent beams or a flux density profile for incoherent beams).

Beam	k	Beam Shape
MODE	0	Complex Hermite-Gaussian beam (default if WAVEL>0)
ELIP	1	Uniform ellipse (default when WAVEL=0), central hole ratio s
GAUS	2	External real Gaussian
RECT	3	Uniform parallelogram with central hole ratio **s**
BELL	4	Bell-shaped (cosine squared) approximation to Gaussian
LRNZ	5	General Lorentzian (inverse **s** asymptotic falloff)
SINC	6	Sinc (sinx/x) with Fourier central hole ratio **s**
SECH	7	Sech (hyperbolic secant)
SOMB	8	Sombrero or Airy disk function with Fourier central hole **s**
FIBR	9	Fundamental mode of circular fiber at normalized frequency **s**
SLAB	10	Fundamental mode of symmetric waveguide at normalized frequency **s**

- The sign of **k** (or **beam**) determines whether the beam is incoherently added (intensity summing) or coherently added (complex field summing) to other beams of similar designation.

k < 1	Coherent beam
k > 0	Incoherent beam

- The optional entry **s** is an arbitrary factor that is passed to the USHAPE routine.
- For MODE or **k** equal to 0, **n** is the number of higher-order Hermite-Gaussian beam modes and **a a' a"...** are the corresponding complex amplitudes. The modes are ordered in the following manner: 00 10 01 20 11 02 30 21 12 03 40 31 22 13 04 50 ...

CAUTION

When using **SHAPE** to alter the beam shape, please be aware that **it is possible to create a physically unreal beam.**

- The default amplitude for the TEMoo is 1 (unity). The amplitudes of the higher order beam modes are defaulted to zero.
- When higher order beam modes are desired, both the number of higher order modes and their amplitudes must be entered. The number of higher order beam modes is given by **n**.
- Ray tracing has no affect upon beam shape; it is a property of the beam and is not altered by propagation. Consequently, it may be reset at any time and may be parametrically varied if desired.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

RAYSET
SELECT
WAVELENGTH

SHIFT (ASAP Command)

Specifies a relative shift of an entity along the coordinate axes.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities
Define/Modify Lens Entities
Create/Modify Objects
Modify Ray/Beam Data

First Syntax:

```
SHIFT [ x y z ]  [ LIST ]
        X x
        Y y
        Z z
```

Second Syntax:

```
SHIFT d ALONG a,b,c  [ LIST ]
```

Option	Description
`X, Y, Z`	shifts axis
`x, y, z`	relative shift along coordinate axis
`d`	distance
`a b c`	direction vector
`LIST`	decodes transformation matrix into simple operations (if possible) and prints

Remarks

- Translates the entity BY the given distances (x,y,z). The default distances move the entity's reference point to the global origin.
- To shift a distance **d** along a given direction (**a,b,c**) use the second syntax.
- When used with REPEAT, group SHIFT with these commands: MATRIX; ROTATE; SCALE; SKEW; PLACE; ALIGN; XEQ

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

SHOW (ASAP Command)

Displays current settings.

Function

Setup Plots and Verify System

Syntax

```
SHOW [ ALL ]
```

Remarks

- Prints out the current status of some or **ALL** of those commands that set flags or data used by other commands. These commands include:

  ACCURACY
  ARROWS
  AXIS
  BEAMS
  BILATERAL
  CLIP DIRECTION/POSITION
  CUTOFF
  FRESNEL
  HALT
  IRRADIANCE
  LEVEL
  LIGHT
  LSQFIT
  MISSED
  OBLIQUE
  PARABASAL
  PIXELS
  POLARIZATION
  SAVE
  SEED
  SEGMENTS
  SPLIT
  TITLE
  UNITS
  UPDATE
  WARNINGS
  WAVELENGTH
  WIDTHS
  WINDOW
  XMEMORY

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

SINGLET (ASAP Command)

Creates a simple singlet lens.

Function

Define/Modify Lens Entities

Syntax

```
SINGLET X x t h m [ RD r r']
        Y y         CV c c'
        Z z         FL f b  [ a       ]
                         a    APLANAT
```

Option	Description
X or Y or Z	global coordinate axis
x or y or z	location on the global coordinate axis
t	lens thickness
h	aperture height
m	internal medium (number or name)
r r'	radii of curvature of the two surfaces
c c'	curvatures of the two surfaces
f	focal length
b	bending parameter
APLANAT	produces zero third-order spherical aberration and coma
a	conjugate factor

Remarks

- This lens entity starts out normal to the defined global coordinate axis (X, Y or Z).
- **RD** is used to specify radii of curvature (**r r'**), **CV** is used to specify curvatures (**c c'**), and **FL** is used to specify focal length **f** and bending parameter **b**.
- The bending parameter **b** is defined as (**c**+**c'**)/(**c**-**c'**) or, equivalently, as (**r'**+**r**)/(**r'**-**r**); therefore, **b**=0 implies a biconvex or biconcave element; **b**=-1 implies a plano-convex or plano-concave element; and **b**=1 implies a convex-plano or concave-plano element.
- **a** is an optional conjugate factor; that is, one plus the object-to-image magnification divided by one minus the magnification **(0=one-to-one imaging, 1=infinite object distance, -1=infinite image distance)**
- If the **APLANAT** option is used, the bending factor is automatically determined for the given **a** so that third-order coma is also eliminated (assuming the thin lens approximation applies).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DOUBLET

SKEW (ASAP Command)

Specifies an arbitrary skewing of an entity.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities
Define/Modify Lens Entities
Create/Modify Objects
Modify Ray/Beam Data

Syntax

```
SKEW X a Y [ c ] [ LIST ]
     Z
     Y a Z
     X
     Z a X
     Y
```

Option	Description
`X Y Z`	skew direction (first column of X,Y,Z)
`a`	angle measured in degrees from the second direction (second column of X,Y,Z)
`c`	corresponding coordinate of the skew center in second direction
`LIST`	decodes transformation matrix into simple operations (if possible) and prints

Remarks

- Skews the entity in the first direction by an angle **a** measured in degrees from the second direction.
- The corresponding coordinate of the skew center is optionally given by **c**. The skew center is defaulted to the entity's reference coordinate.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

SMOOTH (ASAP Command)

Quadratically (or cubically) smoothes the current curve.

Function

Define/Modify Curvedge Entities

Syntax

```
SMOOTH [ 2 ]
         3
```

Option	Description
2	quadratically smooth the current curve (default)
3	cublicly smooth the current curve

Remarks

- Quadratically (2, the default) or cubicly (3) smoothes the current curve assuming it is piecewise linear, that is one created by the ELLIPSE, ROUNDED, or OVAL commands.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

ELLIPSE
ROUNDED
OVAL (ASAP Command)
REPEAT (ASAP Command)

SOLID (ASAP Command)

Makes the previous surface a solid.

Function

Define/Modify Surfunc Entities

Syntax

```
SOLID PLUS  [ NOHOLE ]
      MINUS
      OFF
```

Option	Description
PLUS	uses positive side of surface/function
MINUS	uses negative side of surface/function
NOHOLE	excludes the interior hole of the LOCAL box, if one exists
OFF	turns the SOLID option off

Remarks

- Make the current surface a SOLID formed from the interior of its LOCAL box and the given side of the function.
- Currently, the SOLID command only affects the BOUNDS command.
- If the volume needed can be represented by just a LOCAL box, then use the above command with a dummy degree zero surface, for example,

```
SURFACE n
  GENERAL x y z +1
    LOCAL ...
    SOLID PLUS
:
OBJECT
  :
  BOUNDS +-n
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

BOUNDS
LOCAL
REPEAT

SOURCE (ASAP Command)

Specifies sources for the rays/beams.

Function

Create Rays/Beams

Syntax

```
SOURCE POSITION  x y z [ x,y,z' x,y,z" ... ]
       FOCUS
       LINE
       DIRECTION a,b,c   a,b,c' a,b,c" ...   [ RANDOM f ]
       GRID  [ ONE ]
```

Option	Description
POSITION	defines the angular properties of a diverging beam
FOCUS	defines the angular properties of a focusing beam
LINE	defines the ray propagation for an ellipsoidal or line source
x y z	global coordinate of point from which rays appear to emanate (POSITION) or converge (FOCUS)
x,y,z' x,y,z"...	additional source points
DIRECTIOn	defines the angular properties of a collimated beam
a,b,c	direction vector
a,b,c' a,b,c" ...	additional source direction vectors
RANDOM f	randomizes the directions within an angle **f** times the divergence angle
GRID	sets up a grid of sources
ONE	treat the entire grid as one source

Remarks

- The **SOURCE** command is used after the GRID and RAYSET commands to initialize the directions and optical path lengths of the rays.
- The vector (a,b,c) specifies a direction assigned to all the rays, that is, a collimated beam.
- It is not necessary to normalize the direction vector; ASAP does it automatically.
- With the **LINE** option, each pair of coordinates locates the foci of an ellipsoidal wavefront. If the foci are sufficiently separated, one effectively gets a line (or cylindrical) source.
- The optical path length of the first ray is always chosen to be zero.
- More than one source location or direction can be entered on SOURCE.
- With the **DIRECTION** option, the directions can be **RANDOM**ized within an angle **f** times the divergence angle given on the PARABASAL command.
- DIRECTION can be used with the first parameter as the AXIS (X Y Z), and with the second parameter as the angle in degrees made to that axis. For example, SOURCE DIR Z 50 makes rays go at 50 degrees to the Z axis.
- The **GRID** option uses the coordinates defined in the last GRID command (GRID DATA, GRID ELLIPTIC, GRID HEX, GRID OBJECT, GRID POLAR, or GRID RECT, GRID WINDOW) as the **SOURCE** coordinates and not as starting ray coordinates.
- If the **GRID** option is used with the **DIRECTION** option, the grid coordinates are interpreted as direction cosines. Therefore, the sum of the squares of the grid coordinates should not exceed one; if they do, they are eliminated from the grid. The sign of the third direction is taken from the 4th entry on the GRID command.
- By default each entry in the source grid is treated as a separate source. The entire source grid may be treated as a single source by using the **ONE** option. (It may be necessary to treat all sources as a single source if the total number of individual sources exceeds the maximum number of sources permitted in a given installation of ASAP.)
- Multiple **SOURCE**s refer to the same grid of rays. Each is a different source, but all sources initially are referenced as OBJECT 0.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Apodization of Ray Distributions
FFAD
SOURCE WAVEFUNC

SOURCE WAVEFUNC (ASAP Command)

Specifies the ray propagation direction using a wavefront function.

Function

Create Rays/Beams

Syntax

```
SOURCE WAVEFUNC k [ p k' p' ... ]
                X
                Y
                Z
                N
```

Option	**Description**
k	previously defined surface/function
p	power to which surface/function is raised (default is 1)
k' p' ...	specifies additional sources
X, Y, or Z	direction along which ray positions are moved to the wavefront surface **k**
N	moves ray positions along surface normal to the wavefront surface **k**

Remarks

- The wavefront function or Eikonal is specified by the previously defined SURFACE FUNCTION **k** raised to the **p**th power (default 1) and the ray **GRID**.
- The normalized gradient of the function (times the sign of **k**) becomes the ray's direction, the normalized value its optical path length.
- Alternatively, the rays generated by a **GRID** command are moved along the given direction (**X**, **Y**, **Z**) or along the surface normal (**N**) to the actual wavefront surface **k**.
- Multiple source wavefronts can be created with additional pairs of entries and refer to the same **GRID** of rays. Each is a different source, but all SOURCEs initially are referenced as OBJECT 0.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW
RAYSET
SURFACE FUNCTIONS

SPECTRUM (ASAP Command)

Simplifies flux weighting of rays as a function of wavelength - for spectral apodization.

Function

Setup Beam Creation

Syntax

```
SPECTRUM [ OFF              ]
           FCN  fcn  [ s ]
           VISUAL
           SCOTOPIC
           THERMAL  t
           PHOTONS  t
           w w' w" ...
```

Option	Description
`OFF`	turn off spectral apodization
`fcn`	name of the function
`s`	scale to some other peak value
`VISUAL`	Standard Luminosity Curve (photopic eye in bright light)
`SCOTOPIC`	Eye response in dim light
`THERMAL`	Power output of a blackbody at temperature **t** (degrees K)
`PHOTONS`	Photons/second of a blackbody at temperature **t** (degrees K)
`w w' w" ...`	table of weights

Remarks

- Spectrally apodizes individual monochromatic sources that constitute a polychromatic source.
- Sets up or turns OFF spectral apodization (flux weighting as a function of wavelength) for future ray/beam creation.
- Four standard (VISUAL, SCOTOPIC, THERMAL and PHOTONS) spectral curves are provided.
- Two user-definable curves are also available, either a $FCN called **fcn** or a table of weights **w w' ...** (one for each wavelength specified on the last WAVELENGTHS command).
- The four standard spectral curves are by default normalized to unity at their maximum.

Example

Creating a polychromatic source with visual weighting example

```
UNITS ...
SPECTRUM VISIBLE
$DO 400 700 25
  { WAVELENGTH ? NM
        :
  ray creation commands (for example, EMIT)
        :
  }
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

WAVELENGTHS

SPLINE, EXPLICIT 2D (ASAP Command)

Explicit 2-D curvature (G2) continuous, piecewise cubic spline curve in the given plane (default 0).

Function

Define/Modify Curvedge Entities

Syntax

```
SPLINE X [ x ]
       Y   y
       Z   z
h  s   d   [ c ]
h' s' [ d']
[ h" s" [ d"] ]
:
```

Remarks

- Defined in terms of subsequent heights (next coordinate), slopes, optional distances (remaining coordinate), and starting curvature **c** (inverse radius).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SPLINE, GENERAL 3D

SPLINE, GENERAL 3D (ASAP Command)

General 3D piecewise, cubic, spline curve with given reference point (default origin), and defined in terms of subsequent relative points and tangent directions (and optional inverse radii of CURVatures).

Function

Define/Modify Curvedge Entities

Syntax

```
SPLINE [ x y z ]
         x  y  z  a, b, c
         x' y' z' a',b',c'
       [ x" y" z" a",b" c" ] ]
            :
```

Remarks

- If a tangent direction is not provided for any point, one is calculated by finite differences using the adjacent points. Therefore, this command can also be used as an (often better) alternative to POINTS followed by SMOOTH **3** (although it does create a curve with three times as many Bezier points).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SPLINE, EXPLICIT 2D

SPLIT (ASAP Command)

Controls the number of times a ray may be split into specular components.

Function

Create/Modify Objects
Setup Trace

Syntax

```
SPLIT [ n ] [ MONTECARLO ] [ 'c' ]
              TRAN
              REFL
              NORM
              OFF
```

Option	Description
MONTECARLO	no additional specular and/or scatter rays are created
n	maximum number of times a ray can be split into specular components
OFF	resets n to zero
TRAN, REFL, or NORM	split trace operation selection; NORM is the default
c	fractional energy cutoff for ray splitting

Remarks

- **SPLIT** can be applied on an OBJECT by OBJECT basis. In effect, this is now an OBJECT modifier, similar to the **INTERFACE** command.
- Controls the splitting of the children rays, which are rays that have been split off a parent ray. Therefore, **SPLIT 1** tells ASAP to split the parent rays, but the children rays are not allowed to split. **SPLIT 2** allows the parent rays and the children rays to split as often as necessary, but the grandchildren rays are not allowed to split, and so on. The parent ray is a ray originally created by the **GRID** or RAYSET commands.
- Since the total number of rays to be traced can become quite large, **SPLIT** should be used with some attention to the possible consequences of generating a large number of split rays. For a typical application such as ghost image analysis, **n**=2 is sufficient.
- Even though ASAP has virtual ray storage, you are practically limited by your total free disk storage and long run times. Therefore, the default deterministic splitting should be used with these restrictions in mind.
- Splitting may occur at interfaces with nonzero reflection and transmission coefficients as well as on diffraction gratings with multiple orders. When ASAP encounters a nonzero reflection and transmission coefficient on an INTERFACE command, it automatically sets **SPLIT** to 1.
- Normally, if the transmission coefficient is equal to or greater than the reflection coefficient, the transmitted ray is the first ray propagated. If the reflection coefficient is greater than the transmission coefficient, the reflected ray is the first ray propagated. The split trace option selection allows the most energetic ray at **NORM**al incidence (the default), the **TRAN**smitted, or **REFL**ected component to be traced first, while the other component is traced later.
- Total internal reflection is considered a ray error unless **SPLIT** is turned on.
- With the **MONTECARLO** option, no additional specular and/or scattered rays are actually created, the new parent is probabilistically selected from the possible child rays. Otherwise, for a modest number of initial rays and a refractive/scatter object, the total number of rays eventually traced can be quite large.
- For a typical application such as ghost image analysis, a value of 2 for **n** is sufficient.
- If the fractional energy in the split-off ray/beam is below **c** (default 1.E-6), splitting does not occur, that is, only the main component is propagated.
- To control the level of scattered component splitting, use the parallel command **LEVEL**. Sets the default refraction/reflection controls for all objects or may be applied only to a specific object.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID DATA
GRID ELLIPTIC
GRID HEX

GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW
LEVEL
RAYSET

SPOTS (ASAP Command)

Creates a geometric spot diagram for the currently selected ray data.

Function

Analyze Ray/Beam Data

Syntax

```
SPOTS POSITION  [ u ] [ ATTRIBUTE i ] [ OBJECT ] [ NUMBER [s] ] [ EVERY n ] [ 'title' ]
      Pc          ADD
      DIRECTION
      Dc
```

Option	Description
POSITION	spot diagram of positional ray data
DIRECTION	spot diagram of directional ray data
u	FORTRAN unit number for distribution data file
Pc	spot diagram of positional ray data for the base ray and/or particular parabasal rays (see Remarks)
Dc	spot diagram of directional ray data for the base ray and/or particular parabasal rays (see Remarks)
ADD	adds flux data to existing distribution data file
ATTRIBUTE i	output format control (see Remarks)
OBJECT	output color control
NUMBER	draw the rays number on the plot
s	optional scale factor for the character size
EVERY n	plot only every **n**th ray instead of all the currently selected ones
'title'	optional title for plot (up to 64 characters)

Remarks

- For the current ray set, **SPOTS** produces a spot diagram of the ray data of the specified type, in coordinate directions specified by the last WINDOW command. By default the positions or directions of each base ray are used.
- **SPOTS** may be calculated at any time to examine the ray distribution. ASAP projects all **SPOTS** into the plotting window regardless of third coordinate location.
- The **WINDOW** settings, if not specified, are automatically adjusted for either position or direction cosine data. The **SPOTS** calculation is faster if the **WINDOW** is preset as opposed to ASAP autoscaling the **WINDOW**.
- Use the CONSIDER and SELECT commands to isolate the object(s) and sources whose ray distributions are to be examined.
- The default distribution file is BRO009.DAT. The flux data for each pixel written to logical unit **u** in unformatted direct access binary records whose length is the same as the current PIXELS setting. If **u** is negative, then any previous data on the specified unit is overwritten. Otherwise, the flux data is added pixel by pixel to what is already in the file. The default value of **u** is -9. **ADD** is equivalent to +9. This file can be manipulated, plotted, and named with the DISPLAY command and its associated subcommands.
- For the **Pc** and **Dc** options, the **c** entry can be used to select the following specific data:

c	Coordinates Plotted
	Current base ray (default)
S	All the current parabasals
0	Initial base ray
1	First parabasal only
2	Second parabasal only
:	:

- Normally, ASAP produces a distribution file of accumulated ray fluxes at each pixel location. The **ATTRIBUTE** option (other than **SPOTS ATTRIBUTE 0**) instructs ASAP not to create a distribution file and to print only a symbol where each ray strikes, as shown by the following:

ATTRIBUTE	Symbol
-1	Connecting line
0	No plot generated
1	Plus sign
2	X cross
3	Up triangle
4	Down triangle
5	Square
6	Diamond

- The spot diagram data also is placed into the BRO030.DAT (or filename.vcr) file, but only if the **ATTRIBUTE i** is less than one.
- With the **OBJECT** option, the color associated with the ray's current object, instead of the ray's source, is used for plotting the **SPOT**. In addition, the distribution file contains object number information and not flux information.
- Optionally, the ray's **NUMBER** can also be drawn on the plot. The **s** is an optional scale factor for the character size to be used.
- Also, only **EVERY n**th ray can be plotted instead of all the currently selected ones (which are still summed in the distribution file).
- The title is delimited by a single quote **'** , as shown.
- The ...CLIP command argument can be used to specify an object (**i**) or edge number (**j**) whose bounds and limits clips the distribution. If **i** is not given, it is defaulted to the current object number of the first valid ray. If **j** is negative the interior of the closed edge is used, if **j** is positive, the exterior of the closed edge is used.
- The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands (includes standard plot command options)
...CLIP
CONSIDER
DISPLAY
PIXELS
SELECT
WINDOW

SPREAD (ASAP Command)

Calculates the exact coherent/incoherent energy distribution.

Function

Calculate Diffraction/Propagation Effects

Syntax

```
SPREAD DIRECTION [ u ] [ DOWN d d' ] [ COLENGTH l ]
       POSITION   ADD
       APPROX
       NORMAL
```

Option	Description
`DIRECTION, POSITION, APPROX, or NORMAL`	calculation to use (see Remarks)
`u`	logical unit number for the distribution data file
`DOWN d d'`	restricts the calculation to a fractional range given by **d** to **d'**
`COLENGTH l`	incoherently sums any beam whose optical path length is different from that of the first valid beam by the coherence length **l**
`ADD`	ADD the calculated distribution to the existing BRO009.DAT file

Remarks

- Calculates the coherent/incoherent energy distribution for the current beam set in the directions given by the last WINDOW command.
- The particular calculation is selected by the **DIRECTION**, **POSITION**, **APPROX** or **NORMAL** entry (listed from fastest to slowest):

Option	Functional Coordinates	Optical Method	Assumed Coherence	Units
`DIRECTION`	d-cosines	rough geom	incoherent	Flux/Ster
`POSITION`	positions	rough geom	incoherent	Flux/Area
`APPROX`	positions	rough wave	coh/incoh	Flux/Area
`NORMAL`	positions	exact wave	coh/incoh	Flux/Area

- For the **SPREAD** calculation to take place, a WAVELENGTH (and possibly a PARABASAL) command must be entered before any ray definitions.
- ASAP performs this calculation with four or eight parabasal rays. To accurately model astigmatic effects in the Gaussian beams, four or eight parabasal rays are required.
- The WINDOW and PIXELS commands affect the sampling used. **WINDOW** sets up the area in space over which the calculation is to be performed. **PIXEL** defines the number of resolution points that are to be contained within the current **WINDOW** setting.
- **DOWN** may be used to select only a piece of the **WINDOW** setting in the down (horizontal) direction. This piece is determined from the **d** and **d'** entries. For example, for **d**=.25 and **d'**=.75, the central 50 percent of the window is selected. However, if **d=d'**=.5, a 1-D profile of the energy pattern along the centerline of the current WINDOW is calculated.
- The maximum resolution (number of pixels) across the window is dependent on your version of the software. There is no limit to the number of pixels down the window, but the distribution data file is truncated past the maximum number of pixels.
- The default distribution file is BRO009.DAT. This file can be manipulated, plotted, and named with the DISPLAY command and its associated subcommands.
- If the **ADD** option is present, the calculated distribution is added to the existing BRO009.DAT file.
- **SPREAD** automatically sums over multiple wavelengths incoherently and only uses beams located in the same medium.
- If **SPREAD APPROX** is selected or the number of **PARABASAL** rays is 0, only symmetric beams are used, that is, individual astigmatic effects are ignored.
- **SPREAD NORMAL** is exact only if at least four **PARABASAL** rays were created and/or traced. If there are multiple wavelengths in the current ray set, ASAP automatically cycles through each distinct wavelength and incoherently (intensity) sums it with any other wavelengths.
- **SPREAD NORMAL** does *not* calculate polarization effects.

- **SPREAD NORMAL** is nominally equivalent to FIELD ENERGY.
- The **COLENGTH** option may be used to incoherently sum a subset of beams that would normally be coherently summed. Any beam whose optical path length is different from that of the first valid beam by the coherence length **l** is incoherently summed. The default for **l** is a very large number; beams with SHAPE factors that are less than or equal to zero are coherently summed.
- The ...CLIP command argument can be used to specify an object **i** or edge number **j** whose bounds and limits clip the distribution. If **i** is not given, it is defaulted to the current object number of the first valid ray. If **j** is negative the interior of the closed edge is used, if **j** is positive, the exterior of the closed edge is used.
- The flux data for each pixel is written to logical unit **u** in unformatted direct access binary records whose length is the current **PIXEL** setting. If **u** is negative, then any previous data on the specified unit is overwritten. Otherwise, the flux data is added pixel by pixel to what is already in the file. The default value of **u** is -9.
- **ADD** is equivalent to +9. This file can be manipulated, plotted, and named with the DISPLAY command and its associated subcommands.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Coherent/Incoherent Analysis

VIOLATION (ASAP Command)

SPREAD and FIELD Command Differences

SPREAD/FIELD Calculations (Point Sources)

FIELD

IRRADIANCE

STATS (ASAP Command)

Lists statistics of the currently selected ray data.

Function

Create/Modify Objects
Analyze Ray/Beam Data

Syntax

```
STATS [ ALL ]
        POSITION
        P#
        DIRECTION
        D#
```

Option	Description
ALL	object summary for all considered objects
POSITION	statistical analysis of positional ray data
DIRECTION	statistical analysis of directional ray data
P#	statistical analysis of positional ray data of the base ray or a particular parabasal ray
D#	statistical analysis of directional ray data of the base ray or a particular parabasal ray

Remarks

- By default produces a one line per object summary of the flux and number of rays on each object (**ALL** considered objects, even those with no rays and zero flux on them).
- If a second entry is specified, it also produces a statistical analysis of the **POSITION**al or **DIRECTION**al ray data including centroids, full variances (RMS deviations), and maximum spreads in each coordinate direction on each object.
- By default the positions or directions of each base ray is used.
- Any particular parabasal ray may be selected by specifying its number **#**, for example, **P0** means base ray position, **D1** first parabasal ray direction, etc.
- The actual objects analyzed may be controlled by the CONSIDER command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

EXTREMES
GET
PUT
SELECT

STATS TRACE (ASAP Command)

Prints statistics of a previous TRACE.

Function

Trace Ray/Beams

Syntax

```
STATS TRACE [ MEDIA ] [ MODELS ] [ OBJECTS ]
```

Remarks

- If the previous TRACE had a STATS or ACCUM option on it, selectively print the given statistics.
- For **OBJECTS** (the default), only those currently CONSIDERed are printed.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

STORE (ASAP Command)

Stores the current state of the lens in the ASAP internal database, or optionally in the following file types: ASAP input , macro, CODE V SEQuence, Zemax, or OSLO.

Function

Define/Modify Lens Entities

Syntax

```
STORE [ name.INR ] [ name.MAC ] [ name.SEQ ] [ name.ZMX ] [ name.LEN ]
```

Remarks

- Virtually all commercial lens-design programs can read this simple version of an Optical Research Associates CODE V file. The possible exception is the private glass catalog, which can be avoided if you use standard catalog glasses like Schott.
- The stand-alone ASAP files automatically create lens edges, do a basic plot with ray traces, and produce spot diagrams across the image surface.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Optimizing a Centered Lens

SUBSET (ASAP Command)

Selects a subset of the current ray set and rejects remaining ray data.

Function

Modify Ray/Beam Data

Syntax

```
SUBSET [ RESET [ m ] ]
```

Remarks

- Collapses ray storage to include only those rays determined by the last CONSIDER, SELECT, and CUTOFF commands. All other ray information is lost.
- Useful when a lot of rays have been traced and only a small subset is analyzed extensively.

  Optionally **RESET** the following information:

RESET Information	RESET value
Current object	0
Previous objects	0
Number of hits	0
Specular splits	0
Scatter levels	0
Parent ray	itself
Optical Path Length	0 for ray 1
Current medium	**m** if specified

- Useful for preparing the output from one optical system (for example, a complex source) to be the input for a completely different one.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

CONSIDER
SELECT
CUTOFF

SUM (ASAP Command)

Creates a composite scatter model.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
SUM [ X ] i i' [ i" ... ]
      Y
      Z
      :
```

Option	Description
`i  i'  i" ...`	previously defined model numbers

Remarks

- Adds the BSDF's of the given list of previously defined models to form a composite model.
- ASAP does not support other VANES and non-VANES models. However, it is fully recursive so that one SUM model can reference another.
- Only one anisotropy orientation is used for all the models, either the one specified or the one for the first anisotropic model listed.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MODELS
VANES

SWEEP (ASAP Command)

Specifies the optional sweeping of a curve into a surface.

Function

Define/Modify Curvedge Entities

First Syntax:

```
SWEEP DIR  d [ a,b,c ]
      TO   x y z
      POS  f [ x y z ]
      AXIS t   a,b,c  [ x y z ]
```

Second Syntax:

```
SWEEP [ OFF ]
        d
        f
        t
```

Option	Description
DIR POS AXIS	sweep type (see Remarks)
d	actual sweep distance
a b c	sweep direction or axis
TO	sweep first point on edge TO
x y z	sweep point
f	fractional sweep distance
t	sweep angle
OFF	turns off sweep

Remarks

- If **DIR**ection, sweep an actual distance **d** along direction (**a,b,c**) (default is normal to best fit plane through edge points) or sweep the first point on edge **TO** (**x y z**).
- If **POS**ition, sweep edge a fractional distance **f** towards point (**x y z**) (default is centroid of edge points).
- If **AXIS**, sweep the edge an angle **t** in degrees around the axis specified by the given direction and point.
- This information is used to define the surface of a single edge object or a BOUNDS edge.
- The SWEEP can be turned OFF or modified with the second syntax.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT (ASAP Command)

SURFACES/FUNCTIONS/ENTITIES (ASAP Command)

Signals ASAP that surface definition commands follow.

Function

Define/Modify Surfunc Entities

Syntax

```
SURFACES  [ i ]
FUNCTIONS
ENTITIES
```

Option	Description
i	starting number for defining SURFACES/FUNCTIONS

Remarks

- The default value for **i** is one more than the highest surface number previously defined. The **i** is initialized to one at start of program execution.
- EDGE, LENS, and SURFACE data currently reside in the same internal storage locations. Therefore, a SURFACE number cannot be the same as an already defined EDGE or LENS number.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

SYSTEM (ASAP Command)

Stores and retrieves system information.

Function

Save or Recover System Data and Control Execution

Syntax

```
SYSTEM NEW
       TO   [ file ]
       FROM
```

Option	Description
NEW	reinitializes the data storage
FROM	reads system from file
TO	saves system to file
file	name of unformatted binary file (up to eight characters)

Remarks

- Initializes ASAP at the start of execution. To input a new system, you must reinitialize the data storage using **SYSTEM NEW**, or ASAP appends the new data to the old system data.
- Saving the current system data in a binary format is accomplished with the **SYSTEM TO** file command.

CAUTION

ASAP stores only the system data (in a binary format, not a text format). Ray data and associated parameters are not saved with the system.

- Previously stored system files may be reloaded into ASAP with the **SYSTEM FROM** file command.
- The file extension for system files is *.SYS.
- After an END command and just before exiting, ASAP writes out a system data file named LASTEXEC.SYS. This file may be read into ASAP during the next session by typing **SYSTEM FROM** without any additional arguments.
- A **Select File** dialog box containing all of the **SYSTEM** files in the current directory may be obtained by issuing **SYSTEM FROM "*"** or **SYSTEM FROM_**.

NOTE

BRO recommends that you AVOID using **SYSTEM** files for archival purposes, since their binary format normally change between releases.

Data the **SYSTEM** command saves:

Number of surfaces, edges, lenses, number of media, number of objects, number of coatings

Objects' names

Commands used to define each surface, edge, lens

Degree of each surface polynomial, number of points in each edge, number of conicoids in each lens

Number of aspheric terms (using the DEFORM command) used on each object

Reference point of each surface, edge, lens

Surface coefficients/edge points/lens parameters

Multiple and array surface data

Test points

LOCAL data (including transformation matrices and others)

Refractive index and absorption data

GRIN data

Names of MEDIA

COATING prescription data

OBJECT status (from CONSIDER command)

LIMITS Main data

BOUNDed surface data

Real/complex reflection/transmission coefficients

INTERFACE media numbers

Diffractive/holographic optic INTERFACE data

Object SEARCH strategy data
INTERFACE scatter data (RMS heights, slopes, BSDFs, RMS roughnesses, important areas, backscatter data and others)
Current WAVELENGTH, UNITs, and wavelengths used to define dispersive MEDIA
$FCNs needed to create geometry

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

RESET
RAYSET

TABLE (ASAP Command)

Creates a table of the current data.

Function

Display/Modify Energy Distributions

Syntax

```
TABLE [ m n ] [ l ] [ SKIP [ s ] ] [ PLOT [ c ] ]
```

Option	Description
`m  n`	table range (default 10x10)
`l`	number of characters (default 7, maximum 13)
`SKIP s`	leaves table entries with value **s** blank
`PLOT`	flag to plot the table
`c`	character rescaling factor (default 1)

Remarks

- Produces an **m** by **n** (default 10x10) table of the current data with **l** characters (default 7, maximum 13) for each number.
- If the **SKIP** option is used, table entries with the same value as **s** (default is the data's minimum value) are left blank.
- The table can also be **PLOT**ted where **c** is a character rescaling factor (default 1). This plotted table can be OVERLAYed with other similar plots such as PLOT BEAMS, PLOT POLARIZATION, SPOTS, FIELD DELTA, or CONTOUR.

Example

Overlaying a plotted table with other similar plots:

```
DISPLAY
  CONTOUR 10 TICS OVERLAY
  TABLE PLOT
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

TELESCOPE (ASAP Command)

Creates a one- or two-mirror telescope.

Function

Define/Modify Lens Entities

Syntax

```
TELESCOPE X x h FL1 f [ SEP s FL2 d ] [ STOP p [ m ] ] [ FOV a [ l ] ]
          Y y   FL      MAG   BWD
          Z z
```

Option	Description	Sign
X or Y or Z	global coordinate axis	
x or y or z	location on the global coordinate axis	
h	entrance pupil height	
FL1 f	Focal length of primary mirror	+
FL	Overall focal length of telescope	+C,-G *
SEP s	Separation distance between mirrors	+
MAG	Secondary magnification factor	+C,-G *
FL2 d	Focal length of the secondary mirror	-C/+G *
BWD	Position of final image relative to primary	+/-
STOP p	Distance from real object space aperture stop to primary	+
m	media number or name for a corrector plate at stop	+
FOV a	Semi-field angle in degrees	+
l	media number or name for a field flattening lens	+

* C = two-mirror Cassegrain configuration, G = two –mirror Gregorian configuration

Remarks

- Creates a one or two mirror telescope with or without a corrector plate and/or field flattener.
- The primary mirror is located at the plane, given the defined global coordinate axis and the location on the global coordinate axis.
- The final design is always corrected for third-order spherical aberration, and as many other aberrations as possible.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

TEST (ASAP Command)

Selects a particular branch of a surface.

Function

Define/Modify Surfunc Entities

Syntax

```
TEST  OFF
     -POINT   x y z
     +POINT
     -DIRECTION  a,b,c
     +DIRECTION
     -AXIS  a,b,c [ x y z ]
     +AXIS
```

Option	Description
OFF	turns OFF previous or default test
+/-POINT	tests POINT at (x,y,z)
+/-DIRECTION	tests DIRECTION at (a,b,c)
+/-AXIS	tests AXIS vector (a,b,c) through point (x,y,z)

Remarks

- When a surface of order two or greater has multiple branches, usually you are only interested in one of them. This problem is equivalent to the alternate surface intersection problem seen in conventional optical design software. Use TEST to select a given branch relative to a test point or direction.
- TEST initiates the following action: at each intersection with the surface, ASAP takes the **DIRECTION** or the vector from the **POINT** to the intersection and dots it with the gradient of the **SURFACE** at the point of intersection. The **AXIS** syntax is useful for surfaces of cylindrical symmetry. ASAP draws a vector perpendicular to the **AXIS** to a point on the curve. In all cases, the intersection is valid only if the result has the indicated sign. Only one entry (OFF/POINT/DIRECTION/AXIS) can be used at a time with TEST.
- Certain commands (OPTICAL, AXICONIC and others) internally generate a test in an attempt to select the appropriate branch. In the event the logic fails, it is usually necessary to issue a new **TEST** command to get the desired surface**.**

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

AXICONIC
OPTICAL
REPEAT

TEXTFILE (ASAP Command)

Writes the current data to a user-definable text file.

Function

Display/Modify Energy Distributions

Syntax

```
TEXTFILE name
   :
  header lines
   :
  DATA [ k ] [ s ] [ 'format' ]
   :
  trailer lines
   :
  EOF
```

Option	Description
name	name of the text file (default extension is .txt)
k	number of columns to write per line
s	scale factor (default=1; no scaling)
format	Fortran FORMAT specification (default G15.7)
EOF	flag to close the file

Remarks

- Writes the current data to a user-definable text file called **name** (default extension is **.txt**).
  The following registers are immediately loaded and can be used in the header and trailer definitions:

Register	Literal Value	Numeric Value	Description
T	title	0	Data set title
F	flabel	0	Function label
Z	LABELZ	zval	3rd coordinate value
L1	labelx	0	1st coordinate label
L2	labely	0	2nd coordinate label
N1	NUMX	numx	Number of 1st samples
N2	NUMY	numy	Number of 2nd samples
A1	XMIN	xmin	Minimum 1st coordinate
A2	YMIN	ymin	Minimum 2nd coordinate
B1	XMAX	xmax	Maximum 1st coordinate
B2	YMAX	ymax	Maximum 2nd coordinate
D1	DELX	delx	1st coordinate spacing
D2	DELY	dely	2nd coordinate spacing
F1	FMIN	fmin	Minimum function value
F2	FMAX	fmax	Maximum function value

- The actual DATA can be written in virtually any format.
- The **s** is an optional scale factor that is multiplied with each data value before writing (default is 1, no scaling).
- An actual Fortran FORMAT specification (**format**) can be used for each number (default G15.7).
- The **k** is the number of columns to write per line (default is number of columns in data). Some values of **k** have special meaning:

k	output
-1	f
	:
-2	x f
	:
-3	x y f
	:
-4	x y z f
	:
NUMX+1	x x' ...
	y f ...
	y' f' ...
	:

- EOF closes the text file.
- Alternatively, a **RETURN** can be used to return to DISPLAY mode without closing the file. Another **TEXTFILE** command without a file name entry continues writing to the original file. This way multiple distribution data sets can be written to the same file.
- TEXTFILE in the DISPLAY command can output the distribution data file into a matrix format, for direct importing to a spreadsheet program. In general, the output format is user-modifiable.

The simplest use of TEXTFILE is to dump the data in straight matrix/array or row/column format for importing into spreadsheet software; that is, **TEXTFILE name; DATA; EOF**

Example

A sophisticated example of TEXTFILE is a macro that takes the current distribution in DISPLAY and converts it to a PostScript image file:

```
PSFILE { 1 Convert current distribution to a PostScript image file
 TEXTFILE #1.PS
  R=ABS((YMAX-YMIN)/(XMAX-XMIN)) I=(700/R)<560 J=(560*R)<700
  '%!PS-Adobe'
  '%%BoundingBox: 0 580 0 770'
  'gsave'
  "/picstr" (NUMX) "string def"
  20 20 "translate" (I) (J) "scale"
  (NUMX) (NUMY) 8
  "[" (NUMX) 0 0 -(NUMY) 0 (NUMY) "]"
  '{ currentfile picstr readhexstring pop }'
  'image'
  DATA 255/FMAX 'Z2.2'
  'showpage grestore'
  EOF
}
PostScript file name>
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

THRESHOLD (ASAP Command)

Resets all data values.

Function

Display/Modify Energy Distributions

Syntax

```
THRESHOLD a b [ m n ]
```

Option	Description
`a b`	reset all data values less than b to a
`m n`	reset all data values greater than m to n

Remarks

- Reset all data values less than **b** to **a**.
- Optionally, reset all data values greater than **m** to **n**.
- Each of these value entries can be specified either directly, as the MIN, AVE or MAX value, or as a percentage of the range from minimum to maximum (for example, 33.3%).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

TITLE (ASAP Command)

Specifies default title for future graphics.

Function

Setup Plots and Verify System

Syntax

```
TITLE [ userid ] [ 'default_title' ]
```

Option	Description
`userid`	display a userid literal in the upper-left corner of each graphics screen (up to 8 characters)
'`default_title`'	default title for graphics created with subsequent commands

Remarks

- Up to an eight-character **USERID** literal can be displayed in the upper-left corner of each graphics screen. If a title is not supplied on a graphics generating command (that is, a comment string on a PROFILE or PLOT), the **'default_title'** specified by this command is used.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

TORUS (ASAP Command)

Creates a toroidal or doughnut surface.

Function

Define/Modify Surfunc Entities

Syntax

```
TORUS X x h r [ r' [ q ] ]
      Y y
      Z z
```

Option	Description
X, Y, or Z	axis of symmetry
x, y, or z	location along coordinate axis
h	height to center of torus ring
r r'	semimajor lengths of the cross-sectional ellipse/rectangle at height h
q	curve type control parameter (default is q=0, an ellipse)

Reference Point

At the specified coordinate point (not on the surface of the torus).

Surface Normal

Outward from the surface.

Autolimiting

Yes

Remarks

- The **q** parameter controls the shape of the cross-section in the following way:

q=0	elliptical (default)
0<**q**<1	rounded rectangle
q=1	rectangular

- **r** and **r'** are the (perpendicular to the axis and parallel to the axis, respectively) semi-major widths.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

TRACE (ASAP Command)

Traces all currently selected rays in storage.

Function

Trace Ray/Beams

Syntax

```
TRACE [j j']  [STEP [k]]   [LIST [n]]  [GRAPH [n       ]] [STATS [m] ]  [KEEP]
                            DIR         PLOT       [d]     ACCUM
```

Option	Description
`j  j'`	starting/stopping objects (defaults are 0 0, from all objects to all possible objects)
`STEP [ k ]`	number of stepped object intersections (default is -1)
`LIST [ n ]`	prints every **n**th ray object-by-object ray position data
`DIR [ n ]`	prints every **n**th ray object-by-object ray position and direction data
`PLOT [ n ]`	plots every **n**th ray as it is being traced
`TRACE PLOT COLOR [ n ]`	plots a trace in a color of your choice
`GRAPH [ n ]`	graphical means of monitoring ray trace, **n** is the maximum ray number magnitude (default is 6, that is, 1 million rays)
`STATS [ m ]`	prints ray statistics (base and up to the first **m** parabasals) for each model or object encountered during ray trace; handles only absorption (not gain media)
`KEEP`	keeps previously traced rays
`d`	distance from zero plane (see Remarks)
`ACCUM`	keeps track of ray statistics (base and up to the first **m** parabasals where the default is all)

Remarks

- Uses the ray data in the virtual.pgs file (which is constructed when you use the **GRID**, **RAYSET**, **EMITTING**, and other ray initialization commands), and traces only those rays currently selected through the objects currently considered. During the ray trace, each ray is accessed, traced, and the final status of the ray is copied back to the virtual.pgs file. Upon completion of the ray trace, you can issue analysis commands such as **SPOTS**, **OPDMAP** and others, which in turn access the virtual.pgs file for current ray data. Rays traced via the **TRACE** command are therefore permanent rays.
- With **TRACE** only, no output is typically generated except for possible ray warning messages. **TRACE** also lists warning summary for suppressed child rays.
- Optionally, only those rays whose starting depth coordinate lies within a distance **d** of the zero plane are **PLOT**ted.
- Once traced, rays cannot be retraced unless they are reinitialized.
- By default, the first ray definition command (for example, GRID, EMIT) executed after tracing automatically resets the ray pointer back to zero (RAYS 0) which flushes any rays. To override this behavior, use the **KEEP** option on the TRACE command.
- Sign of **j'** or **k** determines if a refraction/reflection calculation is performed at the stopping object (negative **j'** or **k** turns this calculation off). In other words, a **+j'** or **+k** performs the optical transformation on the ray at the **OBJECT**; **-j'** or **-k** does not.
- Alternatively, the rays can be STEPped **k** objects intersection. The sign of this entry determines whether one steps through the object (**+k**) or up to the object (**-k**).
- The **PLOT** option plots the rays as they are being traced and also store their vectors on the 3-D graphics file future plotting.
- The progress of the ray trace can be monitored **GRAPH**ically on a logarithmic curve. The **n** in this case is the maximum ray number magnitude (default is six, that is, 1 million rays).
- **STATS** or **ACCUM** tracks ray statistics (base and up to the first **m** parabasals, where the default is all) for each GRIN/absorbing media, scatter model, and object encountered during the trace. **STATS** prints the statistics on completion of the trace. If scattered rays are involved, they are listed separately, with minus signs preceding the objects. If **ACCUM** is used, the same information is displayed when you issue the **STATS TRACE** command at a later time. Absorption inside media is also calculated.
- **TRACE LIST/DIR** produces the same output as the RAY command (less the effective focal length).
- A typical use of the **TRACE PLOT** option is to plot a ray trace over a graphic; that is, PROFILE OVERLAY**; TRACE PLOT.**

The ...OVERLAY command argument tells ASAP not to issue an end of plot so that the next plot created is overlaid with the current plot. This is useful for combining system plots with ray trace plots (assuming that the WINDOW is the same for all plots), multiple spot diagrams, and so on.

- WARNINGS limit ends only current **TRACE**, not entire program.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Plotting Commands (includes standard plot command options)

STATS TRACE (ASAP Command)

TRANSPOSE (ASAP Command)

Rotates the current distribution data by 90 degrees.

Function

Display/Modify Energy Distributions

Syntax

```
TRANSPOSE [ i ]
```

Option	Description
`i`	line to transpose distribution data array about (default is about diagonal)

Remarks

- Transposes distribution data array about the **i**th line (default is about diagonal) so that future PLOT3D and ISOMETRIC commands give a different view of the function.
- Use before the GRAPH command to see reversed profiles or perpendicular to usual ones.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY

TREE (ASAP Command)

Displays object name tree.

Function

Setup Plots and Verify System

Syntax

```
TREE [ ENTITIES ] [ i ] [ 'delimiter' ]
```

Option	Description
`ENTITIES`	displays the base entities and BOUNDS entities for each object
`i`	indent space to use when displaying object name tree (default 2)
`'delimiter'`	character delimiter separating each object name tree level (default is a period)

Remarks

- Displays object name tree indented by **i** (default 2) at each level, assuming they are separated by the single character **delimiter** (default is a period (**.**)).
- The **ENTITIES** option also displays the base entities and signed BOUNDS entities for each object.
- The corresponding object (and entity) numbers are shown inside angle brackets **< >**.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

BRANCH and TREE Commands (ASAP Example)
BRANCH

TUBE (ASAP Command)

Creates a tube with rectangular/elliptical cross-sections.

Function

Define/Modify Surfunc Entities

First Syntax:

```
TUBE X x y z x' y' z'  [ q [  q' ] ]
     Y y z x y' z' x'    INNER
     Z z x y z' x' y'    OUTER
```

Second Syntax:

```
TUBE X x y z [ ANGLES b c [ q ] ]
     Y y z x        c a
     Z z x y        a b
```

Option	Description
`X, Y or Z`	axis of symmetry
`x, y or z`	location along coordinate axis of first closed curve
`y z, z x, or x y`	semimajor widths of first closed curve
`x', y' or z'`	location along coordinate axis of second closed curve
`y' z', z' x', or x' y'`	semimajor widths of second closed curve
`q q'`	parameters that control the type of closed curve at the two ends (see Remarks)
`INNER/OUTER`	see Remarks
`b c, c a, or a b`	angles (in degrees) that the two orthogonal projections of the sides of the tube make with the symmetry axis

Reference Point

If **+X**, **+Y**, **+Z**, reference point is at positive end of the tube

If **-X**, **-Y**, **-Z**, reference point is at negative end of the tube

If **X**, **Y**, **Z**, reference point is in the middle of the tube

Surface Normal

Radially outward

Autolimiting

First syntax: Yes   Second syntax: No

Remarks

- Creates a general tube-like surface symmetric about the given axis with rectangular or elliptical cross-sections.
- The parameters **q** and **q'** control the type of closed curves at the two ends of the tube in the following way:

q=0	elliptical
0<**q**<1	rounded rectangle
q=1	rectangular

  Therefore, it is possible to have a fifth-order tube whose cross-section varies linearly from an ellipse on one end to a rectangle on the other.
- The default is **q=q'**=0, that is, elliptical cross-sections.
- If the tube is fourth (q=q') or fifth-order, there is several branches to the surface that may or may not have to be excluded by the user.
- If the two ends of the tube are of different size, instead of a sloped cone, the two ends can be joined by an **INNER** and **OUTER** cylinder and an annular plane at one end.
- TUBE has limitations when processing surfaces with squared corners or tapering at one end. When using the ROUGHNESS command, these circumstances cause leaks or generate wrong-side errors.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

UNITS (ASAP Command)

Defines the system units.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
UNITS [ length ] [ 'flux' ]
```

Option	Description
`length`	length units of the system geometry
`'flux'`	label for the flux units

Remarks

- Takes an optional flux label that is printed on intensity, irradiance and radiance plots in DISPLAY.
- Sets the length units of the system geometry to one of the following lengths:

IN	or INCHES
MM	or MILLIMETERS
YD	or YARDS
FT	or FEET
MI	or MILES
M	or METERS
KM	or KILOMETERS
MIL	or MILS
UM	or MICRONS
CM	or CENTIMETERS
UIN	or MICROINCHES

- The labeling of flux units can be set to the given string (up to 11 arbitrary characters). Some typical ones would be WATTS, LUMENS or PHOTONS/SEC.
- The **UNITS** command is used in conjunction with the WAVELENGTH and CAD commands to associate units with the optical system.
- Also specifies system units in conjunction with a WAVELENGTH UNITS command to properly scale the optical path difference.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

CAD
WAVELENGTH

UPDATE (ASAP Command)

Controls the entity storage updating.

Function

Define/Modify Entities or Single Entity Objects

Syntax

```
UPDATE [ OFF ]
```

Option	Description
`OFF`	overwrites data in storage

Remarks

- Controls how basic geometrical entity data (that is, **SURFACE** coefficients, **LENS** conicoids, **EDGE** points) is updated in storage.
- Normally when an already defined entity is updated at a later time, the new data overwrites the old. This overwrite presents no problems as long as the amount of new data is less than or equal to the amount of old data. However if the new data occupies more storage space, then the data for the next entity can become corrupted.
- The **UPDATE** command tells ASAP to place all future entity data at the end of storage. The old data is not deleted, but it is never used again.
- To return to the default overwrite mode, use the **OFF** option.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

USERAPOD (ASAP Command)

Specifies a beam irradiance or intensity apodization.

Function

Setup Beam Creation
Modify Ray/Beam Data

First Syntax:

```
USERAPOD  POS [ fcn ] [ c c' c" ... ] [ 'string' ]
APODIZE   DIR   OFF
          OFF
```

Second Syntax:

```
USERAPOD POS c c'
APODIZE  DIR
[ e  p  [ f  ] [ PLOT ] ]
                 PROD
[ e' p' [ f' ] ]
    :
```

Option	Description
POS or DIR	type of apodization
fcn	name of the apodization function to be applied to the set of rays
c c' c". . .	up to 50 coefficients used to define apodization function
OFF	turn the currently defined apodization function off
e f	measured energy flux
p	spatial or direction cosine coordinate
PLOT	creates a default distribution file (BRO009.DAT)
PROD	flag to use the product of the two orthogonal profiles instead of the weighted sum

Remarks

- Users can define their own **POS**itional and/or **DIR**ectional apodization function to be applied to a set of rays or beams created in a **GRID** or **EMITTING** surface command. This is done by either using the given $FCN function, **fcn** or rewriting the Fortran function, **USERAPOD** and relinking the program.
- Turns on or **OFF** the use of the currently defined apodization function by the program.
- **APODIZE** operates immediately on all the currently selected or considered rays and therefore, can be used before or after a TRACE.
- **APODIZE** is a ray modifier that apodizes ray distributions of all currently selected and/or considered rays before or after a ray trace. It is less constrictive than the **USERAPOD** command. It may be used to angularly apodize rays through a filter or at a detector.
- **USERAPOD** is only applied to rays when they are created. Therefore, it should be placed prior to the ray definition commands for which it is applicable. Once entered, however, it modifies all subsequent ray data until turned **OFF**.
- **USERAPOD POS** only applies to the following ray creation commands: GRID HEX, GRID POLAR, GRID RECT, GRID WINDOW, SOURCE POS GRID, SOURCE DIR GRID, and EMITTING DISK/RECTANGLE
- **USERAPOD DIR** only applies to the following ray creation commands: SOURCE DIR GRID, EMITTING DISK/RECTANGLE, EMITTING ENTITY or OBJECT, and EMITTING DATA plane.
- If **fcn** is specified, the in-plane, 2-D **POS**itional or **DIR**ectional coordinates are passed in the _1 and _2 variables. The 50 coefficients **c c'...** are passed in the _3 _4 ... _52 variables. The last entry in the $FCN definition of **fcn** is used as the apodization flux scaling factor. For example, to define and use a Gaussian apodization (identical to the default one discussed below):

```
$FCN APOD _5*GAUS[_1/_3]*GAUS[_2/_4]/(_3*_4)
  :
USERAPOD POS APOD ...
  :
    GRID ...
    SOURCE ...
```

- Otherwise, up to 50 coefficients (**c c' ...**) and the **'string'** can be passed to the **USERAPOD/APODIZE** command where the default apodizing function is a Gaussian apodization of the form (using the Z axis as the direction of propagation):

  $$T(x,y) = \frac{c''}{(cc')} \text{EXP}\left\{ -\pi \left[\left(\frac{x}{c} \right)^2 + \left(\frac{y}{c'} \right)^2 \right] \right\}$$

  Therefore, **c** and **c'** are the semi-major widths of the Gaussian envelope. The **c''** is the total flux in the beam, assuming it is not appreciably truncated by the finite size of the grid or emitting surface and that the default POS irradiance or DIR radiance distribution is unity.
- Only a rotationally symmetric apodization may be used with the GRID POLAR.
- The **x** and **y** in the **T(x,y)** equation become direction cosines if the **DIR** option is used.
- With the second syntax, a table of measured energy flux in the two orthogonal grid directions **e,f** versus **POS**ition or **DIR**ection cosine **p** can be entered to apodize the grid.
- The value of **p** is entered in ascending order. ASAP linearly interpolates to obtain intermediate flux values.
- The **c** and **c'** values are used to scale **p** into the two coordinate values at the GRID or EMITTER plane. If no scaling is desired, enter 1 1 for **c** and **c'**.
- Using the **PLOT** option with the second syntax creates a default distribution file (BRO009.DAT) for future plotting by the DISPLAY command (the current PIXELS setting controls the resolution).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

USERAPOD ANGLES

USERAPOD BOTH

USERAPOD ANGLES (ASAP Command)

Specifies spatial or 3D angular apodization of volume emitters.

Function

Setup Beam Creation

First Syntax:

```
USERAPOD ANGLES X [ fcn ] [ c c' c" ... ] [ 'string' ]
APODIZE         Y
                Z
                OFF
```

Second Syntax:

```
USERAPOD ANGLES X [ f f' t t' ] [ i t" s ]
APODIZE         Y  -1   'filename'
                Z
```

Option	Description
X Y Z	specifies spherical coordinate system axis
fcn	name of the apodization function
c c' c". . .	up to 50 coefficients used to define the apodization function
f f'	zenith angle limits in degrees
t t'	azimuth angle limits in degrees
i	number used to name the interpolation data file
-1 'filename'	see Remarks
t"	azimuth angle data offset
s	data scale factor

Remarks

- Apodize as a function of spherical angles for the given axis the 3D radiation patterns of volume emitters created with the EMITTING commands.
- **APODIZE** operates immediately on all the currently selected or considered rays and therefore, can be used before or after a TRACE.
- **USERAPOD** is only applied to rays when they are created. Therefore, it should be placed prior to the ray definition commands for which it is applicable. Once entered, however, it modifies all subsequent ray data until turned **OFF**.
- **USERAPOD ANGLES** only applies to the following ray creation commands: EMITTING ENTITY or OBJECT, EMITTING DATA volume, EMITTING BOX/SPHEROID, EMITTING CONE/PYRAMID, EMITTING FILAMENT, EMITTING HELIX, and EMITTING RAYS.
- The volume emitters are apodized according to the predefined $FCN function or the user-supplied Fortran in the function **USERANGS**. If **fcn** is specified, then the zenith angle from the axis (0 to pi radians) and the azimuth angle around the axis (-pi to pi radians) are passed in the _1 _2 and the 50 coefficients **c c'...** in _3 _4 ... _52.
- Otherwise, the optional coefficients (**c c' c" ...**)and **'string'** are passed to USERANGS. The default definition for USERANGS is a spherical angle clipping and/or interpolation (see second syntax).
- The **f f'** are the limits of the zenith angle (in degrees) measured from the axis (default 0 to 180).
- The **t t'** are the limits of the azimuth angle measured around this axis (default 0 to 360).
- The **i** is an optional number (non-zero) of a file that contains an interpolation table in spherical coordinates. If **i** is negative, the complete case-sensitive name of the file is taken from the comment string. If **i** is positive and less than 80, then the file's name must be usap3d and its extension is the character whose ASCII value is i+48 (for example, for i=2, the file is usap3d.2). Otherwise, the file name is the number itself and the extension is ua3 (that is, for i=9007, the file is 9007.ua3).
- The **t"** is an azimuthal angle data offset and **s** a data scale factor for the data in the file.
- Data contained in the file **USAP3D.i** is used to apodize the emitting volumes. The first two numbers on the first row of the file are the number of azimuth and zenith angles, respectively. The first column is the zenith angles. The first row is the azimuth angles. The subsequent data is the flux-weighted value at the particular angles. ASAP linearly interpolates to obtain the flux between data points.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

USERAPOD

USERAPOD BOTH

USERAPOD BOTH (ASAP Command)

Apodize in both position and/or direction the full 3D radiation patterns of volume emitters.

Function

Setup Beam Creation

First Syntax:

```
USERAPOD BOTH [ fcn ] [ c c' c" ... ] [ 'string' ]
APODIZE  OFF
```

Second Syntax:

```
USERAPOD BOTH k f f' t t' [ i t" s  ]
APODIZE  X   -i  'filename'
         Y
         Z
```

Option	Description
fcn	name of the apodization function
c c' c". . .	up to 50 coefficients used to define the apodization function
k	specifies axis for spherical coordinate system
X Y Z	specifies spherical coordinate system axis
f f'	zenith angle limits in degrees
t t'	azimuth angle limits in degrees
i	number used to name the interpolation data file
-i 'filename'	see Remarks
t"	azimuth angle data offset
s	data scale factor

Remarks

- Apodize, in both position and/or direction, the full 3-D radiation patterns of volume emitters created with the EMITTING commands.
- **APODIZE** operates immediately on all the currently selected or considered rays and therefore, can be used before or after a TRACE.
- **USERAPOD** is only applied to rays when they are created. Therefore, you must place it prior to the ray definition commands for which it is applicable. Once entered, however, it modifies all subsequent ray data until turned **OFF**.
- **USERAPOD BOTH** only applies to the following ray creation commands: EMITTING ENTITY or OBJECT, EMITTING BOX/SPHEROID, EMITTING CONE/PYRAMID, EMITTING FILAMENT, EMITTING HELIX, and EMITTING RAYS.
- The volume emitters are apodized according to the predefined $FCN function or the user-supplied Fortran in the function **USERBOTH**. If **fcn** is specified, then the global positional coordinates are passed in the _1 _2 _3 registers, the unit direction vector in _4 _5 _6, and the 50 coefficients **c c' ...** in _7 _8 ... _56. Otherwise, the optional coefficients (**c c' c" ...**)and **'string'** are passed to USERBOTH.
- The default definition for USERBOTH is the same as USERANGS, a spherical angle clipping and/or interpolation (see second syntax).
- Option **k** is an axis specification (1=x, 2=y, 3=z) for a spherical coordinate system.
- Option **f f'** are the limits of the zenith angle (in degrees) measured from this axis (default 0 to 180).
- Option **t t'** are the limits of the azimuth angle measured around this axis (default 0 to 360).
- Option **i** is an optional number (non-zero) of a file that contains an interpolation table in spherical coordinates. If **i** is negative, the complete case-sensitive name of the file is taken from the comment string. If **i** is positive and less than 80, then the name of the file must be **usap3d** and its extension is the character whose ASCII value is i+48 (that is, for i=2, the file is usap3d.2). Otherwise, the file name is the number itself and the extension is ua3 (that is, for i=9007, the file is 9007.ua3).
- The **t"** is an azimuthal angle data offset and **s** a data scale factor for the data in the file.
- Data contained in the file **USAP3D.i** is used to apodize the emitting volumes. The first two numbers on the first row of the file are the numbers of azimuth and zenith angles, respectively. The first column is the zenith angles. The first row is the azimuth angles. The subsequent data is the flux-weighted value at the particular angles. ASAP linearly interpolates to obtain the flux between data points.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

USERAPOD
USERAPOD ANGLES

USERBSDF (ASAP Command)

Specifies an isotropic or anisotropic scatter model.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax: Isotropic-surface Scatter Model (user definable)

```
USERBSDF [ fcn ] c c' c" ... [ PLOT [ a a' ... ] ]
```

Option	Description
`fcn`	FCN function
`c c' c" ...`	coefficients
`PLOT`	plots the model in log(b-b$_o$) and angle space
`a a' ...`	user-defined degree specular angles

Syntax: Anisotropic-surface Scatter Model

```
USERBSDF X [ fcn ] c c' c'' …
         Y
         Z
   :
```

Remarks

Isotropic-surface Scatter Model

- Creates, along with a $FCN definition, a user-defined BSDF function.
- Up to 63 coefficients **c** can be used by the predefined $FCN function or up to 286 coefficients can be passed to the USERBSDF Fortran function.
- If **fcn** is specified, the incident vacuum wavelength and the three isotropic-surface symmetry variables (**U,V,W**) are passed in the variables **_0, _1, _2, _3** registers, the 63 input coefficients **c c' …** in **_4_*5* …66**. The last entry on the **$FCN** definition is returned as the actual BSDF value. If the BSDF is only a function of these variables, scattering from an isotropic surface is symmetric with regard to the plane of incidence and surface normal.

  The last entry on the $FCN definition is returned as the actual BSDF value.
- For example, a BSDF that varies as the cosine of the angle from specular raised to some arbitrary power (Phong-like model) could be implemented with the following code:

  $\text{\$FCN FONG C} = \_2+\sqrt{(1-\_1)*(1-\_3)}$,! cosine of angle from specular

  $$\_4*\left((C>0)^{-5}\right)+\_6$$

  MODEL; USERBSDF FONG .3 10 .1
- Alternatively, the simplest BSDF model with constant TIS that is Lambertian at normal incidence but becomes more and more specular as it approaches grazing incidence:

  $\text{\$FCN ROUGH} = \frac{4*(1+\_2)}{3.1416}$! _4 will be the TIS

  MODEL; USERBSDF ROUGH .9 !90% TIS

  Both these examples as defined obey reciprocity (same results if _1 and _3 are interchanged).
- Another example of USERBSDF is the GUERAPV standard model:

  $\text{\$FCN GUERAPV C}=\sqrt{1-\_1}\ \text{A=ACOS}\ \left(\_2+C*\sqrt{1-\_3}\right)$,

  A1=_4 B1=_5 C1=_6 D1=_7 A2=_8 B2=_9 C2=_10 D2=_11 D=_12,

  $$\left(\frac{\text{A1}*\text{EXP}(\text{-B1*A})*\text{COS}(C1*A+D1)+A2*\text{EXP}(-B2*A)*\text{COS}(C2*A+D2)}{\frac{C+D}{3.1416}}\right)$$

  MODEL; USERBSDF GUERAPV ...
- This is an unphysical model that does not obey reciprocity, cannot have the proper directional dependency for smooth surfaces, and could even become negative.
- Enter a negative index for particles on a reflecting surface.

- Besides a $FCN definition, users can program the USERBSDF routine to return the BSDF as a function of the wavelength and isotropic-surface scattering variables. The default definition for USERBSDF is the crude APART model for scattering from a random collection of homogeneous spherical particles where:

  c = particle radius divided by wavelength

  c' = fractional area obscured by particles

  c" = real refractive index of particle (default=100; that is, opaque)

Anisotropic-Surface Scatter Models

- Scattering from anisotropic surfaces is not rotationally symmetric at normal incidence and not necessarily symmetric about the plane of incidence otherwise. Therefore, the orientation of the model on the surface is important and is generally specified by an axis for the second command entry. For syntax information, see ...MODEL... (ASAP Command Argument).
- The predefined **$FCN** function can use up to 62 coefficients, or up to 286 coefficients can be passed to the USERANIS Fortran function. If **fcn** is specified, the following registers are passed:

Registers	**Data**
_0	Incident vacuum wavelength
_1	Scatter Alpha direction cosine
_2	Scatter Beta direction cosine
_3	Specular Alpha direction cosine
_4	Specular Beta direction cosine
_5 ... _66	c c' c" ... input entries

  The last entry on the $FCN definition is returned as the actual BSDF value.

- The default definition for **USERANIS** is a simple elliptical Gaussian centered on the specular direction:

  c = Peak BSDF value

  c' = Width in local Alpha direction

  c" = Width in local Beta direction

Both Isotropic and Anisotropic Models:

- The **PLOT** option plots the model (common base 10 logarithm of the BSDF) for up to seven specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). The current PIXELS setting controls the resolution of these plots in direction cosine space. Creates a distribution file **name_angle.dis** for each of these angles.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Mathematical Models for Isotropic-Surface Scattering

MODELS Overview

MODELS

PARTICLES MIE

USERCURVE (ASAP Command)

Creates a piecewise linear curve (or bilinear surface) from a $FCN function.

Function

Define/Modify Curvedge Entities

Syntax

```
USERCURVE fcn u u' m [ v v' n ]
                      SPLINE
                      CURV
```

Option	Description
fcn	user-defined function
u u'	parameter range
m	number of segments
v v'	second parameter range
n	number of segments (second parameter)

Remarks

- Normally creates a piecewise linear curve (or bilinear surface) from a $FCN function **fcn**.
- The last three expressions of the $FCN definition are the X, Y, and Z coordinates of a point as function of the parameter register **_** or **_1** (and optionally **_2** for a surface).
- For a curve, the parameter ranges from **u** to **u'** in **m** segments.
- For a surface, the second parameter ranges from **v** to **v'** in **n** segments. See the example below.

Example

A wiggly circle definition:

```
PI=ACOS(-1)
$FCN SQUIGGLE A=2*PI*_ R=1+.1*SIN(10*A),
              x R*COS(A) y R*SIN(A) z 0
EDGE
  USERCURVE SQUIGGLE 0 1 100
    SMOOTH
```

or a bilinear approximation to a "sphere" surface

```
$FCN SPHERE R=1 P=_1*PI/180 A=_2*PI/180,
            x R*COS(A)*SIN(P) y R*SIN(A)*SIN(P) z R*COS(P)
EDGE
  USERCURVE SPHERE 0 180 6 0 360 6
```

Optionally, the derivatives of the $FCN function can be used to smooth the curve into a curvature (G2) continuous, piecewise cubic, non-rational SPLINE or rational CURVe.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

USERFUNC (ASAP Command)

Creates a surface specified by a user-defined function.

Function

Define/Modify Surfunc Entities

Syntax

```
USERFUNC x y z [ fcn ] [ c c' c" ... ]
```

Option	Description
`x y z`	global coordinates of reference point
`fcn`	user-defined function
`c c' c" ...`	coefficients to function

Reference Point

As specified

Surface Normal

As specified

Autolimiting

No, requires LOCAL or LIMITS modifiers.

Remarks

- Specifies a user-programmable function with reference point (**x,y,z**) and double-precision coefficients. Its value and gradient at any point can be user-coded in the $FCN named **fcn** or the Fortran function **USERFUNC**. If the function is continuous in both value and gradient everywhere in space, there are no restrictions on the use of this function in ASAP except possibly the application of non-orthogonal transformations to it, that is, SKEW or non-isotropic SCALE.
- If the **fcn** is specified, the current WAVELENGTH (or the associated ray/beam's wavelength) is passed in the **0** register, the local (**x,y,z**) coordinates are passed in the **_1 _2 _3** registers and up to 63 coefficients **c c' ...** in **_4 _5. . ._66**.
- If four or more values are returned, the last four entries of the executed function must be the functional value and its gradient vector. If only one value is returned, it is the functional value, and the gradient will be calculated numerically using finite differences. For example, a sphere of radius 10 centered about the reference point would be done as follows.

  ```
  $FCN SPH _1^2+_2^2+_3^2-_4^2 2*_1 2*_2 2*_3
    :
  SURFACE; USERFUNC .5 0 -1 SPH 10
  ```

  Otherwise, the default **USERFUNC** is an aspheric conicoid with vertex at **x,y,z** and normal direction **c,c',c"**. The fourth and fifth coefficients are the vertex curvature (inverse radius) and the conic constant (0=sphere, -1=parabola), respectively. The remaining coefficients are for a straight polynomial in an offset radial distance including odd terms; that is, the seventh coefficient is the linear term, the eighth is the quadratic, and so on. For example:

  ```
  USERFUNC x y z 0 0 1 c k h d d' d". . .
  ```

  $$f\left(X,Y,Z\right)=\left(Z-z\right)-\frac{cR^2}{1+\sqrt{1-\left(1+k\right)c^2R^2}}-dr-d'r^2-d''r^3-...$$

  $$R^2=\left(X-x\right)^2+\left(Y-y\right)^2$$

  $$r=R-h$$

- An exception has been implemented when the curvature **c** is zero, that is,

  $$f\left(X,Y,Z\right)=k\left(Z-z\right)-dr-d'r^2-d''r^3-...$$

  Default routine handles isotropic SCALE transformation.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

USERSAG (ASAP Command)

Creates a surface with a user-defined radial or toric profile.

Function

Define/Modify Surfunc Entities

Syntax

```
USERSAG  X x  [ fcn ]  c c' ...  [ TORIC r  [ k ]  ]  [ aperture ]
         Y y
         Z z
```

Option	Description
X, Y or Z	axis of symmetry
x, y or z	location along axis
fcn	user-defined function
c c' ...	coefficients (up to 66)
TORIC r	flag to create a TORIC surface of radius **r**
aperture	ELLIPSE, RECTANGLE, or HEXAGONAL

Remarks

- Creates at the given plane, a surface with a user-programmable radial or TORIC (of radius **r**) profile specified by a sag formula.
- The second entry in the syntax designates the axis of symmetry (either **X**, **Y**, or **Z**) for the surface.
- If **fcn** is specified, then the local transverse height coordinate is passed in the **_0** (or **_**) register and up to 66 coefficients **c c' ...** in registers **_1 _2 ... _66**. The last expression of the executed function must be the sag value at that height.
- If **k** is specified, the profile is, instead, shifted along a conic curve with vertex radius of curvature **r** and conic constant **k** (0 = circular, -1 = parabolic). Specifying a **k** of zero is not the same as an absent **k**.
- Otherwise, the default is defined in the **USERSAG** Fortran routine and is a conic plus aspheric terms. The fourth and fifth entries of the syntax are the vertex curvature (inverse radius) and the conic constant (0=sphere, -1=parabola), respectively. The remaining coefficients are for a straight polynomial including odd terms; that is, the sixth entry is the linear term, the seventh is the quadratic, and so on.
- For the Z-axis case,

  ```
  USERSAG Z z c k d d' d" ...
  ```

  $$Z-z=\frac{ch^2}{1+\sqrt{1-(1+k)c^2h^2}}+dh+d'h^2+d''h^3+\ldots h^2=X^2+Y^2$$

  This surface can extend to infinity unless a LOCAL command follows, or a trailing aperture option of the following form is specified:

  ```
  ELLIPSE   a [ a' [ o [ s [ s' ] ] ] ]
  RECTANGLE
  HEXAGONAL a [ o [ s [ s' ] ] ]
  ```

  a a' are the heights in the other two transverse directions.

  For the **HEXAGONAL** form, **a** is the center-to-vertex distance (maximum height).

  o is an optional central hole ratio.

  s s' are the transverse coordinates of the center of the aperture.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

USERSURF (ASAP Command)

Creates a user-defined surface with a user-defined surface intersection.

Function

Define/Modify Surfunc Entities

Syntax

```
USERSURF x y z c [ c' c" ... ]
```

Option	Description
`x y z`	reference point
`c c' c" ...`	double precision coefficients

Reference Point

As specified

Surface Normal

As specified

Autolimiting

Depends on definition

Remarks

- A user-programmable surface with reference point (**x,y,z**) and double precision coefficients whose intersection with an arbitrary ray can be coded by the user in the Fortran function, **USERSURF**.
- The following restrictions apply to this surface:

  A ray is not allowed to intersect the surface more than once consecutively.

  The FCN, BEND, FMAP, MULTIPLE, ARRAY, and TEST commands cannot be used in conjunction with this command.

  Non-orthogonal transformations are not allowed (SCALE or SKEW).

  They cannot be used later as BOUND, GRIN, or alternate surfaces/normals.
- The default definition is a conicoid with vertex at **x,y,z** and normal direction **c**,**c'**,**c"**. The fourth and fifth coefficients are the vertex curvature (inverse radius) and the conic constant (0=sphere, -1=parabola), respectively.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

UVSPACE (ASAP Command)

Tags current curve to be defined in parametric-space of an object.

Function

Define/Modify Curvedge Entities

Syntax

```
UVSPACE [ OFF ]
          i
```

Option	Description
i	object number/name
OFF	return the curve definition to the Z=0 plane in global three-dimensional space (for example, so that it can be PLOTted).

Remarks

- Assume current edge is defined in UV-space parametric space of object **i**. Currently, UVSPACE is only useful if it is used as a closed BOUNDS on the object.
- Alternatively, any planar edge can be directly defined in parametric space by substituting **UV** for the second (axis) and third (location) entries on its command. For example,

  ```
  POINTS Z z ...
  ```

  becomes

  ```
  POINTS UV ...
  ```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

VALUES (ASAP Command)

Lists the functional values of the current distribution data file.

Function

Display/Modify Energy Distributions

Syntax

```
VALUES v h [ reg ] [ v' h' [ reg' ] ... ]
```

Option	Description
`v h, v' h', ...`	coordinate values (v for vertical, h for horizontal)
`reg, reg', ...`	register names

Remarks

- Lists the functional values corresponding to the exact pair of actual coordinates given.
- ASAP linearly interpolates, if necessary, to obtain functional values.
- The functional values can be optionally placed in the given register names.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
SECTION

VANES (ASAP Command)

Creates an anisotropic scatter model that simulates the scatter from a vane structure.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
VANES X a s d t [ t' [ c [ w ]]] [ EDGES r [ t" ]] [ DIFFRACT [ h n ]] [ PLOT [ a a' ... ] ]
      Y
      Z
:
```

Option	Description
X	specifies symmetric axis
a	vane angle from the surface (0 to 180).
s	vane separation
d	vane depth
t t'	vane TIS (sides and bottom)
c	cylindrical radius of curvature of the vaned surface (default=0; infinite)
w	vane cavity width (default=0; infinite)
EDGES r	vane tips edge with radius
t"	Lambertian vane edge TIS
DIFFRACT	specifies vane tip diffraction
h	diffracting edge serration angle
n	number of diffracting vertices per unit length
PLOT	plots the model in log(b-b$_o$) and angle space
a a' ...	user-defined degree specular angles

Remarks

- The surface scatters as if it were the locus of vane tips parallel to the local "Alpha" direction.
- The vane cavity surfaces are assumed to be Lambertian with TIS values of **t** and **t'** for the sides and bottom, respectively.
- The vane tips may have rounded EDGES of radius **r** and a Lambertian TIS of **t"**.
- The vane tips may also diffract energy as if they are optionally serrated with **h**, the half-angle in degrees of each vertex, and **n** the number of vertices per unit length (that is, two over the serration period).
- Scattering from anisotropic surfaces is not rotationally symmetric at normal incidence and not necessarily symmetric about the plane of incidence otherwise. Therefore, the orientation of the model on the surface is important and is generally specified by an axis for the second command entry. For syntax information, see ...MODEL... (ASAP Command Argument).
- The **PLOT** option plots the model (common base 10 logarithm of the BSDF) for up to 7 specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). The current PIXELS setting controls the resolution of these plots in direction cosine space. Also, creates a distribution file **name_angle.dis** for each of these angles.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.
- Please note that this model is an approximation that currently ignores the scatter between the sides and bottom of the cavity. Therefore, under certain circumstances it may underestimate the effective BRDF.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MODELS Overview
MODELS

VARIABLES (ASAP Command)

Declares which lens construction parameters are to be varied during optimization.

Function

Define/Modify Lens Entities

Syntax

```
VARIABLES [ OFF ] [ LIST ] [ TH [ i i'...] ] ...
            CV
            CC
            CP
            AS
            BN
            GL
```

Option	Description
TH	thicknesses
CV	curvatures
CC	conic constants
CP	vertex curvature
AS	aspheric surfaces
BN	bending
GL	glass

Remarks

- The basic parameters are the THicknesses, CurVatures, and Conic Constants on any set of conicoids listed after each variable. If the conicoid number is positive, the parameter is free to vary.
- If entered as a negative number, that particular parameter is subtracted from the variable list (that is, it is frozen at its current value). If no conicoid set is given, all but the last conicoid is allowed to vary.
- These commands are cumulative until another ABERRATION command or an OFF option resets all the parameters to their default frozen state. The new variable set can also be LISTed, where asterisks indicate each active one.
- The sag of higher-order aspheric surfaces (like corrector plates) is given by **AS** (default 1) times the difference between the conic defined by **CV**/**CC**, and a parabola with vertex curvature **CP**.
- BeNding is a composite variable that changes the curvature of the given conicoid(s) while keeping constant the difference in curvatures between it and the next conicoid in the lens. This approximately fixes the power (and focal length) of the two adjacent conicoids, while allowing their aberration contributions to vary. It works with both glass elements and air spaces.
- In global optimization mode, the optimum **GL**ass (or combination of GLasses) can be found from all the currently defined MEDIA.
- Since there can be potentially six variables per conicoid, an optimization can use up to 720 variables. A large number of variables can quickly become impractical even for a local optimization, because runtimes go approximately as its cube.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

VCAVITY (ASAP Command)

Creates a scatter model as a random collection of v-cavities.

Function

Create/Modify Media, Coatings, Scatter Models

Syntax

```
VCAVITY s  r  r'  [ PLOT [ a a' ... ] ]
       -k -h
```

Option	Description
s	RMS slope (in radians)
r	intrinsic normal-incidence specular reflectivity of the material
r'	Lambertian diffuse reflectivity
k	COATING (in vacuum/air)
h	diffuse from the RMS height roughness (in wavelength units).
PLOT	plots the model in log(b-b_o) and angle space
a, a', ...	user-defined degree specular angles

Remarks

- Scatter from a rough surface geometrically modeled as a random collection of v-cavities.
- Includes the effects of shadowing and multiple reflections within a cavity.
- The slopes of the cavities are assumed to follow a Gaussian-normal probability distribution with zero mean and RMS slope **s** (in radians).
- The intrinsic normal-incidence specular reflectivity of the material is defined by **r** and the corresponding Lambertian diffuse reflectivity is defined by **r'** (that is, the sum of these reflectivities should be equal to one minus the material absorption).
- For polychromatic effects, the specular reflectivity can be determined from COATING **k** (in vacuum/air) and the diffuse from the RMS height roughness **h** (in wavelength units).
- The **PLOT** option plots the model (common base 10 logarithm of the BSDF) for up to 7 specular angles in ascending order (default 0, 15, 30, 45, 60, 75, 89.9 degrees). The current PIXELS setting controls the resolution of these plots in direction cosine space.
- Creates a distribution file **name_angle.dis** for each of these angles.
- The ...MINMAX command argument may be used to set the minimum and maximum values of the BSDF for this specific model.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

MODELS

VIEW (ASAP Command)

Make all future plots a true perspective view.

Function

Setup Plots and Verify System

Syntax

```
VIEW [ X ] [ h ] [ CENTER x y z ] [ EYE x' y' z' ]
       Y                            DIR a,b,c
       Z
```

- Make all future plots a true perspective view instead of using the current WINDOW (and OBLIQUE) settings. The view is completely determined by an up coordinate axis, a vertical full height **h** in system units, the global center of the view **x y z**, and eye point **x' y' z'** (or **DIR**ection **a,b,c** equivalent to an eye point a distance 1000***h** away).
- The current view can be modified with the following forms:

```
VIEW [ OFF ] [ ZOOM f ] [ DOLLY d ] [ ORBIT o ] [ FOV a ]
      0
```

- The current perspective view can be modified a variety of ways. Turn perspective viewing **OFF**; that is, return to normal WINDOW (and OBLIQUE) settings.

  The form "VIEW 0" calculates the closest perspective view to the current WINDOW (and OBLIQUE) settings assuming an eye distance that is twice the vertical height (closely approximates a standard 35mm camera lens).

  ZOOM divides the current vertical height by a factor **f** (>1 zooms in, <1 zooms out).

  DOLLY changes the current height and eye distance by the signed fraction **d** (positive moves toward the center point, negative away).

  ORBIT rotates the eye point an angle **o** degrees about the up coordinate axis.

  FOV changes the eye distance to yield a full field-of-view equal to **a** degrees (enter zero for parallel projection).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

VIOLATION (ASAP Command)

Controls the handling of paraxial departure and positivity violations during future SPREAD and FIELD commands.

Function

Calculate Diffraction/Propagation Effects

Syntax

```
VIOLATION g [ g' [ p [ p' [s [s' ] ] ] ] ]
```

Option	Default	Description
g	0.1 wave	Paraxial invariant departure before message issued
g'	1 wave	Paraxial invariant departure before calculation terminated
p	0.01	Positivity violation before message issued
p'	1	Positivity violation before message issued
s	.1	Stability violation before message issued
s'	10	Stability violation before calculation terminated

Remarks

Setting these thresholds higher than the defaults may allow a calculation to finish, but the results should be viewed as suspect.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

SPREAD
FIELD

VOXELS (ASAP Command)

Accumulates on the next TRACE, in a 3D array of volume elements (voxels), the flux-per-unit volume or irradiance on an area.

First Syntax

```
VOXELS FLUENCE  [ d d ' [ n ] ]
       ABSORBED   x x' y y' z z' [ n n' n" ]
       X
       Y
       Z
      -X
      -Y
      -Z
      +X
      +Y
      +Z
```

Second Syntax

```
VOXELS [  OFF        ]
          READ [ u ]
          file
```

Remarks for First Syntax

- During the next **TRACE**, accumulate in a 3D array of volume elements (voxels), the flux-per-unit volume (total absolute **FLUENCE** or **ABSORBED** in media) or irradiance on an area perpendicular to the given coordinate axis.
- Only accumulates irradiance from rays going in the direction specified by the sign of the axis (no sign is equivalent to a bi-directional irradiance).
- The volume is either the enclosing box of the currently considered objects, determined by the current **WINDOW** settings and the specified depth range, or given directly in terms of a range for each of the global coordinates.
- The number of samples per coordinate is then either taken from the current **PIXELS** setting and a depth number or specified directly. If any of the **n**'s are entered as zero, the PIXELS setting at the time of the TRACE is used.
- The total number of voxels is limited to approximately 81 cubed in ASAP/Basic, or 203 cubed in the ASAP/Pro version.
- After the completion of the next TRACE, the results are written to a 3D version of the standard binary distribution file (BRO009.DAT), and any 2D slice can then be processed by **DISPLAY** and its subcommands.

Remarks for Second Syntax

- Turn ON or OFF the volume flux tracking during the next **TRACE**. Optionally, read in a previous 3D binary distribution given by number **u** (default 9), or name "**file**" (default DIS extension). The results of the next TRACE are added to these values, and the total is written to the standard distribution file (BRO009.DAT).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

TRACE (ASAP Command)

VUFACETS (ASAP Command)

Shortcut for faceting and viewing current objects.

Function

Setup Plots and Verify System

Syntax

```
VUFACETS [ n n' [ m ] ] [ LIST ]
```

Option	Description
n n'	faceting numbers
m	maximum number of facets per object

Remarks

- VUFACETS is a shortcut for the $IO VECTOR REWIND, PLOT FACETS [n n'], and $VIEW command sequence. However, it temporarily suppresses any 2-D plotting to minimize computation time and disk space.
- Replots the current accumulated 3-D VECTOR graphics taking into account the current CONSIDER settings.
- If possible, the maximum number of total facets on any object is kept below **m** (the default is 1000; use 0 to turn off).
- Remembers the faceting numbers **n n' m"** from the previous VUFACETS command (an END command resets them to the default 1).
- A $VIEW will list the facets in the Command Output window.
- The numbers of facets created for each object can also be **LIST**ed (a $VIEW is not automatically issued in this case).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

WARNINGS (ASAP Command)

Controls output of parent ray warning messages.

Function

Setup Trace

Syntax

```
WARNINGS [ n ] [ LIST ]
```

Option	Description
n	threshold number of ray warnings
LIST	lists all future warning messages

Remarks

- Sets a state based on the number of ray warnings.
- If **n** is zero or greater and more than **n** ray warnings occur, the raytrace is terminated. The default for **n** is a negative number, which means there is no limit on the number of warnings.
- If the **LIST** option is present, all warning messages are printed during future ray traces.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

LIST
SELECT

WAVELENGTH(S) (ASAP Command)

Defines the wavelength(s) for different MEDIA and COATING definitions.

Function

Create/Modify Media, Coatings, Scatter Models
Setup Beam Creation

Syntax

```
WAVELENGTH w [ w' w" ... [  name  ] ]
                            UNITS u
```

Option	Description
`w w' w" ...`	wavelength(s) (up to a maximum of 15)
`name`	wavelength units designation
`u`	multiplier to convert wavelength into system units

Remarks

- Sets the wavelength for any future calculations to **w** (normally in system units).
- WAVELENGTH(S) must precede any ray definitions to properly set up the Gaussian beams used to calculate the diffracted field.
- Specifies wavelengths used in media, coatings, and other databases. Multiple wavelengths in either ascending or descending order should be entered for any **MEDIA** or **COATING** commands that have multiple refractive indices or properties so as to identify the wavelength for each entry.
- **WAVELENGTH**s as measured in vacuum should be entered on WAVELENGTHS.
- The **UNITS** option with a scale factor **u** (default=1) may be used to convert **w** into system units. The argument **u** is almost always a number much less than unity.
- Alternatively, if a basic UNITS command has already be issued, then any of the following **name**'s can be used to calculate **u** automatically:

  A or ANGSTROMS

  NM or NANOMETERS

  UM or MICRONS

  MM or MILLIMETERS

  CM or CENTIMETERS

  M or METERS
- Some input parameters have units that are the same as **w** (See the INTERFACE command) so that it is sometimes more convenient for these parameters to have different units than the physical system
- In ASAP Pro, the maximum number of WAVELENGTHs for interpolation is 25.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

INTERFACE
SPECTRUM
MEDIA
COATING LAYERS
COATING MODELS
COATING PROPERTIES

WEDGE (ASAP Command)

Creates a wedge of glass.

Function

Define/Modify Lens Entities

Syntax

```
WEDGE X x h m a [ t ]
      Y y
      Z z
```

Option	Description
X or Y or Z	global coordinate axis
x or y or z	location on the global coordinate axis
h	aperture height
m	medium (number or name)
a	angle between the two faces
t	optional thickness (default is h)

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

WIDTHS (ASAP Command)

Modifies the default parabasal ray settings, scaling the width of the parabasal rays.

Function

Setup Beam Creation

Syntax

```
WIDTHS w [ h ] [ EDGE ]
```

Option	Description
`w`	Gaussian beam overlap scale factor
`h`	parabasal ray height scale factor
`EDGE`	option that improves the sharpness of a beam pattern's edge

Remarks

- The **w** is the factor by which ASAP scales the actual width of each beam. When w=1 (the default), a **GRID** command creates a set of beams the just touch each other, that is, unit packing density. When w<1, there are gaps between adjacent beams and when w>1 they overlap. The latter case is especially useful for overlapping a Gaussian beam set so that the overall beam shape is much smoother. For this application, a value of 1.6 is usually sufficient and a value of 2 is more than enough.
- The **h** is the ratio of the parabasal ray heights to the half-max half-width of the beam they define. The default is 1 but it can be set smaller if the exact parabasal rays depart too much from the first-order approximation on which the beam calculations are based. The **h** option should be used with caution because it may contribute to sampling errors.
- The **EDGE** option causes ASAP to add narrow odd-order Hermite-Gaussian beams at the edge of a GRID RECT or GRID POLAR of beams to preserve the sharpness of the overall beam pattern.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

GRID DATA
GRID ELLIPTIC
GRID HEX
GRID OBJECT
GRID POLAR
GRID RECT
GRID WINDOW
PARABASAL

WINDOW (ASAP Command)

Sets the plotting window for graphical output.

Function

Setup Plots and Verify System

Syntax for Setting a Window**:**

```
WINDOW [ X [ a a'  ]  Y  [ d    d' ] ]
              OBJ  i   Z    OBJ  j
         Y             Z
         X
         Z             X
         Y
```

Syntax for Modifying a Window:

```
WINDOW [ ACROSS     ] m p   [ ACROSS    m' p'  ]
         VERTICAL       CEN   VERTICAL     CEN
         DOWN                 DOWN
         HORIZONTAL           HORIZONTAL
         BOTH
```

Option	Description
a a'	window limits along the vertical axis of the graphics device
d d'	window limits along the horizontal axis of the graphics device
OBJ i, OBJ j	sets window to the LIMITS range on the given OBJect numbers
X Y Z	coordinate axis
m m'	magnification factors
p p'	relative pan factors
ACROSS	window direction to be modified
DOWN	window direction to be modified
VERTICAL	window direction to be modified
HORIZONTAL	window direction to be modified
CEN	automatically centers the window on the current data

Remarks

- Since ASAP has no preferred set of axes, use WINDOW to define the plane of your plotting window. For example, WINDOW YZ, or WINDOW y - 11 z 0 3.
- Objects/data inside the window are plotted; objects/data outside this window are not plotted.
- **WINDOW** does not set the third coordinate (the one different than the two entered on the WINDOW command).
- If either **a=a'** or **d=d'**, ASAP attempts to autoscale in that direction (the word attempt is used because **WINDOW** always keeps the aspect ratio correctly unless overridden with a PIXEL command).
- If both sets of numerical entries are omitted, ASAP autoscales in both directions.
- Autoscaling is performed on the basis of the current object set. Objects that have been considered invalid do not influence the autoscaling, nor are they drawn. Autoscaling is also performed on ray trace data (spot diagrams, and so on.).
- Window settings are active until replaced by another **WINDOW** command. A previous window setting may be scaled by **WINDOW f**, where **f** is a scale factor.
- If **a'<a** or **d'<d**, the plot is flipped about that axis. It is possible to flip a plot about an axis by prefixing the coordinate indicator with a minus sign.
- The window size may also be set to the **LIMITS** range on the OBJECT number or name given.
- The current (or next autoscaled) window can be modified by specifying a window magnification factor **m** and/or relative pan factor **p** for one particular direction or both.
- **ACROSS=VERTICAL** and **DOWN=HORIZONTAL** depending on whether one is referring to the printer or plot output.
- As an alternative to the numeric pan factor, ASAP can automatically calculate a pan factor that results in **CEN**tering the particular data being plotted within the window.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

DISPLAY
FIELD
FMAP
MAP
OPDMAP
PIXEL
PLOT
PROFILES
RADIANT
REPLOT
SPOTS
SPREAD

WRITE (ASAP Command)

Writes the current distribution data file to disk file.

Function

Display/Modify Energy Distributions

Syntax

```
WRITE [  u    ]          [ 'title' ]
         name   [ DIS ]
                  DIN
```

Option	Description
`u`	write current data to Fortran unit number **u**
`name`	name of the binary and/or text file
`DIS`	flag to only create a binary file
`DIN`	flag to only create a text file
'`title`'	comment string (up to 24 characters) that overrides the title record in the resulting file

Remarks

- Writes the current data to the current file, logical unit number **u**, binary file **name.dis**, or text file **name.din** (default is both). An optional comment string can be used to override the 'title' record in the resulting file.
- With **DIN**, ASAP limits the column output per line to 10, writes a comma to the end of the line, and continues any further numbers on the next line.
- The title is delimited by a single quote **'** , as shown.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

TEXTFILE (for a more sophisticated text output method)

XEQ (ASAP Command)

Specifies immediate application of current linear transformation.

Function

Define/Modify Curvedge Entities
Define/Modify Surfunc Entities
Define/Modify Lens Entities
Create/Modify Objects
Modify Ray/Beam Data

Syntax

```
XEQ [ LIST ]
```

Option	Description
`LIST`	decodes transformation matrix into simple operation (if possible) and prints

Remarks

- Forces ASAP to immediately apply the current transformation matrix to the entity instead of waiting for a non-transformation command to signal the end to the transformation command set. The transformation matrix is then reinitialized to the identity matrix and ASAP begins looking for a new set of transformation commands.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

REPEAT

XMEMORY (ASAP Command)

Controls the amount of ray data stored in the virtual memory paging file (default VIRTUAL.PGS), or entirely in RAM for a given maximum of **m** rays..

Function

Setup Beam Creation

Syntax

```
XMEMORY [ OFF  ] [ 'pagefile' ]
          MIN       m
          NORM
          FULL
```

Option	Description
OFF, NORM	Imaging (incoherent) No parabasals and higher-order beam modes (default)
MIN	Illumination (incoherent) No polarization, path lengths, paths, and beam sizes; also PLOT BEAMS/POLARIZ, SPREAD, FIELD, OPDMAP disabled
FULL	Imaging (coherent) Automatically turned on by PARABASAL command
'pagefile'	name of the pagefile (default VIRTUAL.PGS)
m	allocates RAM for virtual.pgs, up to a maximum of **m** rays

Remarks

- XMEMORY can reduce the file/RAM size (that is, allow more rays to be traced for a given file/RAM size), and speed up run time.
- Not all ASAP ray traces and associated analysis require the same amount of ray information. For example, coherent sources of radiation, such as lasers and monochromatic plane waves, required much more information per ray, per analysis, than incoherent sources, such as coiled filament light bulbs. Since ray data is read and written to disk, it is sometimes beneficial to reduce the amount of information accessed from the disk, thereby reducing disk I/O time and space.
- The default mode is **XMEMORY NORM**; that is, the additional data is not saved. This feature is automatically turned to **FULL** when the PARABASAL or **XMEMORY FULL** commands are executed.
- Use **XMEMORY** options to reduce the VIRTUAL.PGS file size considerably (that is, allow more rays to be traced for a given file size) and thereby speed up execution.

XMEMORY setting	Words (Bytes) per ray	Comments
FULL	128 (512)	Automatically turned on by PARABASAL command
NORM,OFF	48 (192)	no parabasals and higher-order beam modes (default)
MIN	16 (64)	no polarization, path lengths, paths, and beam sizes; also LIST ELLIP, PLOT BEAMS/POLARIZ, SPREAD, FIELD, OPDMAP disabled

- Use **XMEMORY MIN** when modeling illumination system sources. **XMEMORY MIN** is recommended because of the large number of rays required in these analyses.
- A VIRTUAL.PGS file generated by a ASAP run with **XMEMORY MIN/NORM** cannot be read by a second ASAP run with **XMEMORY** turned **FULL** and vice versa; the VIRTUAL.PGS files are incompatible.
- By default, the virtual.pgs file is written to the hard drive, in the current ASAP working directory. However, if the optional parameter **m** is specified, space is allocated in RAM to write virtual.pgs, up to a maximum of **m** rays. This can improve the speed of the ray trace.

CAUTION

XMEMORY should be set before defining rays. If you run **XMEMORY** during an ASAP analysis, ASAP initializes (deletes) the current VIRTUAL.PGS file.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

PARABASAL
GET
PUT

ZERNIKE (ASAP Command)

Creates a surface specified by an explicit Zernike polynomial.

Function

Define/Modify Surfunc Entities

Syntax

```
ZERNIKE X x c c' c" ...  [aperture ]
        Y y
        Z z
```

Option	Description
`X, Y, or Z`	specifies axis of symmetry
`x, y, or z`	location along axis of symmetry
`c c' c" c'" ...`	University of Arizona FRINGE Zernike coefficients
`aperture`	ELLIPSE, RECTANGLE, or HEXAGONAL

Reference Point

At intersection of the surface and coordinate axis.

Surface Normal

Along negative coordinate axis.

Remarks

- Creates an explicit surface by transforming the University of Arizona FRINGE Zernike coefficient set **c c' ...** to a polynomial.
- The second entry designates the axis of symmetry (either **X**, **Y**, or **Z**) for the surface.
- The zeroth coefficient **c** is the constant term (piston). The 36th or last coefficient is the 12th-order radial term (spherical). The first nine terms are related to the classical third-order optical aberrations:

Term	Dependency	Name
c	1	Piston
c'	$\cos$	x tilt
c"	$\sin$	y tilt
c'"	$2\;{}^{2}\;1$	defocus
:	${}^{2}\cos 2$	x astigmatism
	${}^{2}\sin 2$	y astigmatism
	$3\;{}^{2}\;2\;\cos$	x coma
	$3\;{}^{2}\;2\;\sin$	y coma
	$6\rho^4-6\rho^2+1$	spherical

- ρ is the distance from the coordinate axis.
- It may be necessary to SCALE the surface to its proper dimensions, since these coefficients are usually defined relative to a unit circle area. The **SCALE** command automatically scales the physical dimension of the surface.
- If the sag of the surface is to remain constant over the new physical dimension, then one of the **SCALE** parameters must be 1. For example, to **SCALE** by a factor of 2 in the **X** and **Y** dimensions while maintaining the same sag as over the unit circle, use **SCALE 2 2 1**.
- This surface can extend to infinity unless a LOCAL command follows, or a trailing aperture option of the following form is specified:

```
ELLIPSE   a [ a' [ o [ s [ s' ] ] ] ]
RECTANGLE
HEXAGONAL a [ o [ s [ s' ] ] ]
```

- **a a'** are the heights in the other two transverse directions.
- For the **HEXAGONAL** form, **a** is the center-to-vertex distance (maximum height).
- **o** is an optional central hole ratio.
- **s s'** are the transverse coordinates of the center of the aperture.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

LIMITS

ASAP Macro Language

The input to ASAP is structured along the concept of fundamental commands strung together to perform complex analysis tasks. To that end, the commands have been designed for efficiency, generality, and power.

However, to extract the full power of ASAP from its relatively simple command set, you may frequently need to access the program at an even more basic level where you, in a sense, are programming ASAP. For example, with the macro language you can replace often-used input commands with a single macro command followed by alphanumeric arguments.

The macro language itself is unique and does not try to emulate another programming language. Its command set has evolved (and continues to evolve) to cover a variety of engineering needs.

ASAP includes two types of macros:

User-defined macros, which are typically a series of ASAP commands strung together to perform a specific task, such as source creation.

Internally defined macros that give you advanced programming capabilities.

See Also

Creating a Macro Library File
Calling a Macro
Mathematical Operators
Variables
Internally Named Variables
Externally Named Variables
Variable Substitution
Increasing Macro Command Execution Speed
User-Defined Macros

Calling a Macro

A macro command is unformatted like all other commands. There can be only one user macro command per input record and it should be the last command on the record. The first entry of a macro command is either a dollar sign "$" or ampersand "&" followed immediately by the macro name. The name can be of any length but only the first 32 characters are significant. The dollar sign "$" causes the program to print the commands in the macro definition after the variables have been replaced by their corresponding arguments. Only the macro command itself is printed when the ampersand "&" is used.

The remaining entries on the macro command are the optional arguments to be substituted for the variables in the macro definition. If an argument is not present on the macro command that is referenced in the macro definition, then the program prompts the user for the argument in interactive mode or the variable is replaced with a null entry.

Example

```
Macro name prefixes (echo,no echo of macro records): $ &
```

When the previous example is called, it would be expanded as follows:

```
    :
    $NAME 10.02
    -1- SOURCE .01 10.02
    -1- H 1.01 C 2.01
    :
    &NAME
    Enter collector object number: 5.01
    :
```

Calling a Macro from a Library Other than the Current

Normally, the program searches for a macro definition in the currently loaded library (see $IO LIB command). However, this can be overridden on a case-by-case basis using the following syntax:

```
                --- library$name ...
                           &
```

where

"library" is the name of the library file (a .LIB extension will be added) and "name" is the desired macro in that file. The program looks for the library file first in the current working directory (see $PATH), then in the same directory as the currently loaded library ($IO LIB), and finally in the directory specified by the FILES switch or environment variable.

See Also

ASAP Macro Language

Defining a Macro from the Input Stream

Instead of defining macros in the library file, you can create individual macro files directly from the input stream. There are two formats, a short or long, respectively:

Short Syntax

```
name { [ n ] [ description ]
:
definition records
:
   }
```

Long Syntax

```
$$name m [ n ] ...
$& label
&$
&&
   :
  m records in macro definition or until "label" literal record reached
   :
[label]
```

Remarks

- Long format macro definitions can be created from the input stream by prefixing the name with two macro prefix characters and entering on successive records the m records in the definition.
- An alternative to specifying the integer number of records m in advance is to use a record starting with the given literal (for example, "_") after the last definition record (and before the n argument prompt definitions). The first prefix character determines whether the macro is defined and executed ($), or just defined (&). As usual the second character determines whether the individual records in the macro are displayed ($) or not (&).

Prefix	Action
$$	Define, execute, and echo
$&	Define and execute
&$	Define and echo
&&	Just define

For example, to simply define a macro NAME using a meaningful label:

&&NAME EOM

:

EOM

- As opposed to $GO and $ERR, "label" in this context must be in the currently acceptable case (see $CASE). The above syntax can also be in short format as follows:

NAME {

:

}

Internally Defined Macros

The internally defined macros are contained within the parser (and therefore are always available). These macros give the program looping, input/output redirection, conditional (IF) processing, branching, and other advanced capabilities. Internally defined macros may be used freely in both user-defined macros and general input commands and may appear more than once in a given record.

See Also

ASAP Macro Language

Variables

ASAP has 1768 variable locations (memory cell locations or registers) available for intermediate arithmetic variable assignment. This collective group is referred to as variables, which are divided into two groups: internally named and externally named variables.

See Also

Internally Named Variables

ASAP Macro Language

Internally Named Variables

ASAP has 286 internally named variable locations. They are designated by the letters A through Z9 (A, B, C,…, Z, A0, B0, C0,…, Z0, A1 B1, C1,…, A9, B9, C9,…, Z9).

Three pieces of information are associated with each of these variables:

- variable location (A…Z9)
- variable name (literal up to 16 characters)
- number designation (double precision)

At program startup, the information assigned to all variable locations is zeroed and blanked. You may assign or retrieve information from any of these variable locations in the following way (the variable location designation **R** is used as an example):

Assign a number to R	R=number
Assign a variable name (literal) to R	R="literal"
Recall number from R	(R)
Recall literal from R	R"
Recall number assigned to a variable name	(literal)

You may assign both a variable name and a number to a variable location, and use either the variable location designation or the variable name to recall the number.

Tip

The variable name may be used as a literal assignment, without a numerical value, to a variable location. The variable location designation may then be used to access the literal.

See Also

Externally Named Variables

ASAP Macro Language

Externally Named Variables

Externally named variables are those that allow the user to select a variable name. ASAP then allocates a variable location for it. These variables may be accessed only by the variable name. You can not directly address the particular variable location assigned to that variable.

ASAP has approximately 1500 externally named variables.

Only two pieces of information are associated with each of these variables:

- variable name (literal up to 32 characters)
- number (double precision)

At program start-up, these variables are zeroed and blanked.

You may assign or retrieve information from any of these variables in the following way (the variable name XYZ is used as an example):

Store number **XYZ**:	**XYZ=**number
Recall number from **XYZ**:	**(XYZ)**

Remarks

- No blank spaces are permitted in a variable assignment.
- The variable assignments and arithmetic calculations all take place in the background (within the parser). The calculations or results are transparent to ASAP, unless they are used as arguments to ASAP commands.
- Be careful when using internally named variables if you are also using the GET and PUT commands to access ray data from VIRTUAL.PGS. GET and PUT transfer ray data into and out of specific internal variable locations, potentially overwriting variables you may have intended for another purpose.

See Also

Internally Named Variables

ASAP Macro Language

Variable Substitution

A common use of the variables is simple substitution of a variable for a command argument. In this case, you assign a value to a variable and proceed to use that variable throughout the input. Changes to the model may be accomplished by simply changing the value of the variable.

You do not have to delimit individual variables in an operational string. However, parentheses "**(name)**" or "**name.**" are required when you want to use the numerical content of an isolated variable.

You can modify and use the contents of a variable.

See Also

ASAP Macro Language

Increasing Macro Command Execution Speed

- Do not put too many definitions in one macro file. If possible, break up a large macro file into a number of smaller files, and then use the $IO **LIBRARY** command to switch between them. This technique speeds things up by reducing search time for a particular macro.
- Order your macro definitions within the library file from smallest to largest, or most used to least used.
- Put several commands on each macro definition record to keep the total number of records in the macro file down to a minimum.
- If possible, use the internal variables to pass information instead of the macro arguments. A macro that is called with no arguments runs much faster than one with arguments.

See Also

ASAP Macro Language

User-Defined Macros

User-defined macros can be defined and used within an ASAP input file (*.INR) or defined and placed within a macro library for later use. In either case the syntax is the same.

Syntax

The two formats are short or long, respectively:

Short format:

```
name { [ n ] [ description ]
   :
   definition records
   :
}
```

Long format:

```
$$name  m  [ n ] ...
   $&    label
   &$
   &&
  :
    m records in macro definition or until "label" literal record
reached
    :
[label]
```

Long format macro definitions can be created from the input stream by prefixing the name with two macro prefix characters and entering on successive records the **m** records in the definition.

An alternative to specifying the integer number of records **m** in advance is to use a record starting with the given literal (for example, "_") after the last definition record (and before the **n** argument prompt definitions). The first prefix character determines whether the macro is defined and executed ($), or just defined (&).

The second character determines whether the individual records in the macro are displayed ($) or not (&).

Syntax

```
Prefix      Action
------+------------------------
 $$   Define, execute, and echo
 $&   Define and execute
 &$   Define and echo
 &&   Just define
```

For example, to simply define a macro NAME using a meaningful label:

```
&&NAME EOM
   :
EOM
```

NOTE

As opposed to $GO and $ERR, **label** must be in the currently acceptable case (see $CASE). Actually, the above syntax can now also be in short format as follows:

```
NAME {
   :
}
```

Remarks

- *First record*: The first record of the macro definition contains the macro name and, optionally, the number of argument prompt arguments. This record is read in an unformatted manner starting in column 1. The numerical entries are separated by blanks.
- *Macro names*: must start with a letter, and can be the same as a standard ASAP command without causing conflicts. (In this

latter case, first enter a **DOMACROS FIRST** command to establish the search order for identifying the commands.) However, do not name your macro the same as any of the predefined macros, such as **$ASK, $IO, $SCR**, or a conflict will occur.

- *Argument prompt records:* optional, but are strongly recommended as a means of documenting variables.
- *Macros defined in an *.INR file*: automatically stored in separate files called **name.mac**. The **'comments'** are replaced in the *.MAC file with a **!comments** header for documentation purposes.
- *Macros stored in a macro library file* (libname.**lib**): any number of blank lines or comment lines (starting with an exclamation point "**!**" or asterisk "*****") may be inserted before the start of each definition to enhance readability of the macro library file. (Macros defined in a library file are not available until the macro library is attached to ASAP. Use the $IO command to do this. For example, to attach a macro library named UTILITY.LIB, you would enter **$IO LIBRARY UTILITY.LIB**; from that point on, all of the macros in this library are available.)
- *Macro commands*: are unformatted, like all other commands. There can be only one user macro command per input record, and it should be the last command on the record.
- *Printing commands*: prefixing the macro name with a dollar sign "**$**" causes ASAP to print the commands in the macro definition after the variables have been replaced by their corresponding arguments. Only the macro command itself is printed when the ampersand "**&**" is used as the prefix.
- *Referencing other user-defined macros*: User-defined macros may reference other user-defined macros up to nine levels. Predefined macros do not change the level of macro nesting.
- *Passing variables*: Up to 99 variables may be passed to a macro. Variables may include literals as well as numbers. Within the macro, variables are referred to by a pound sign “**#**” followed by a one- or two-digit integer between 1 and 99, corresponding to the variable’s order when the macro is called. The “**#0**” indicates the number of arguments actually present.
- *Variables within a macro* are replaced character for character with the input. An expanded macro record cannot exceed 128 characters.
- *Arguments*: If an argument is not present on the macro command that is referenced in the macro definition, then ASAP prompts you for the argument (interactive mode), or the variable is replaced with a null entry (batch mode).

See Also

DOMACROS (ASAP Command)

ASAP Macro Language

CAD Library

Creating a Macro Library File

Macro definitions are stored in a library on logical unit 24 by default. Definitions may be created either directly by an ASCII editor or during the decoding of the input by the program.

The first record of a macro definition contains the following information:

- up to a 32-character name of the macro,
- the number of definition records that follow (or up to an 8-character label that marks the end of the definition records),
- and possibly the number of argument prompt records after that.

The record is read in an unformatted manner, with the name starting in column 1 and the numerical entries separated by blanks. Therefore, any characters past the last numerical entry can be used for documentation purposes (that is, a description of the purpose or arguments of the macro).

The macro name must start with a letter and can be the same as a standard program command without causing conflict (for an exception see section on internal predefined macros). The definition records that follow contain the commands that make up the definition. User macro definitions can reference other user macro commands up to 99 levels deep. The internal predefined macros do not change the level of macro nesting. Up to 99 variables may be used in the definition. These variables are represented by a pound sign "#" followed by a 1- or 2-digit integer between 1 and 99 inclusive.

When the macro is called during input decoding, the characters of the corresponding argument of the macro command replace these two or three characters. Therefore, macro variables may represent literal and numeric quantities of any length. However, an expanded macro record cannot exceed 344 characters.

- Default macro library file logical unit: 24
- Format of macro initialization record: name m [n]
  label
  {
- Macro arguments (string substitution): #1 through #99
- Number of arguments actually present: #0
- Maximum level of macro call nesting: 99

An example of a macro definition is:

```
NAME 2 1 2 definition + 1 argument prompt records
SOURCE .01 #1
H 1.01 C 2.01
Enter collector object number:
ANALYZE 10 next definition with no argument prompts
:
```

Any number of blank lines or comment lines (starting with an exclamation point "!" or asterisk "*") may be inserted before the start of each definition to enhance readability of the macro library file.

See Also

ASAP Macro Language
Sharing Library Files

Sharing Library Files

In environments where multiple engineers work together, you may want to share ASAP library files (*.LIB). If the PC's running ASAP are connected via a LAN, two methods in which LIB files can be shared enable you to modify a LIB file so that other staff can use the information.

Method 1

From a performance standpoint this method is not the best choice, but it does demonstrate that files can live almost anywhere. The **$IO LIBRARY** command can specify a drive and subdirectory to where a library file is located:

```
$IO LIBRARY "u:\jmiller\asap_lib\SOURCES.lib"
```

- Note that double quotes are used to delimit the long path name. The quotes are necessary because of the special characters typical of a DOS path, which ASAP would otherwise interpret incorrectly.
- The performance issue is that when a macro from a library is called, this file is read in order (from the top, down) to locate the requested macro. Since the drive is a network drive, this takes more time than if the file is located on the local PC.
- Depending on your LAN environment you may not be able to have multiple people accessing the LIB file simultaneously. Also, it is important to remember to close the library file at the end of the INR file with the **$IO LIBRARY CLOSE** command.

Method 2

The goal is for you to have access to the latest version of the library files each time you run your INR files. Within the ASAP scripting language, the **$SYS** command runs a DOS-level command. Prior to opening the LIB file, the main running INR file can copy the LIB file from the LAN to the local PC.

```
$SYS "COPY U:\USR\JMILLER\ASAP_LIB\SOURCES.LIB *.*"
$IO LIB "SOURCES.lib"
```

When the INR is run, a DOS box appears briefly.

A variation is the "Default Setup File" option.

- Open the ASAP Preferences dialog from the Settings menu. Click the "Default Setup File" tab. For example, create a new file in an ASAP editor window called SETUP.INR. Place the following command in this file:

```
$SYS "COPY U:\USR\JMILLER\ASAP_LIB\SOURCES.LIB C:\ASAP_LIB\*.*"
```

- Save this SETUP.INR file before you run another INR file.
- Open the ASAP Preferences dialog from the Settings menu. Click "Default Setup File" tab. You can either browse to the location of the SETUP.INR file you just created, or type the drive, directory, and SETUP.INR in the space provided. Click OK to close the dialog. ASAP runs this file when you start ASAP as well as any time you click on the END toolbar button or an **END** command is issued. You can then use the **$IO LIBRARY** command as usual:

```
$IO LIB "c:\asap_lib\SOURCES.lib"
```

- Virtually any DOS command can be used in with **$SYS**. Even a .BAT file, which gives you another level of flexibility and power.

See Also

Creating a Macro Library File

$ABORT (ASAP Macro)

From any macro/loop level, returns to the command prompt as soon as possible and displays the optional "message".

Syntax

```
ABORT [ 'message' ]
```

Option	Description
`message`	optional message to display

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$ARGS (ASAP Macro)

Controls argument prompting in user-defined macros.

Syntax

```
$ARGS [ ALL ]
        NONE
        USER
        SCR
```

Remarks

The SCR option displays a dialog box for prompts with arguments. After pressing Enter after each entry, click the OK or Cancel button as appropriate to exit the dialog.

Interactive prompts are displayed for either **ALL**, **NONE** or only the ones defined by the **USER** in the macro definition. The default in the absence of any **$ARGS** command is **ALL**. If no argument is given on the command, the state before the last **$ARGS** command is restored.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$ASK (ASAP Macro)

Assigns a value to a variable through a prompting dialog box.

Syntax

```
&ASK [ register [ register' ... ] ] [ 'prompt' ]
$ASK
```

Remarks

- Macro prompts you for new value(s) for the variable(s) by displaying a prompting dialog box. This dialog box has a user-specified prompt string across the top and an edit box along the bottom, into which the data is entered. For improved usability, enter a prompt string on Macro.
- The values may be either numeric or literal.
- In the event there is more than one variable, the dialog box contains a long edit box into which all of the values are entered. ASAP does not create a separate dialog box or prompting edit box for each variable.
- If the **$** prefix is used, the current value(s) for the variable(s) is also displayed.
- If there are no variable arguments, ASAP displays the prompt dialog box and then waits for you to click on OK.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$BEEP (ASAP Macro)

Causes computer to emit a beep.

Syntax

```
$BEEP [ n [ s ] ]
```

Remarks

Macro causes the terminal or console to beep **n** times (default 1) with **s** seconds (default 0.00) between each beep.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$CASE (ASAP Macro)

Specifies case sensitivity.

Syntax

```
$CASE [ UPPER ]
        LOWER
        BOTH
```

Remarks

- Macro instructs the parser to recognize only **UPPER**- or **LOWER**-case letters or **BOTH**.
- The case of the prompt displayed below the ASAP Command Input window reflects the setting of the **$CASE** macro.
- If no argument is given on the command, the state before the last **$CASE** command is restored.
- Letters opposite in case to that set by the $CASE command can also be used as imbedded comments since the program treats them as blanks. Trailing comments can be entered after an exclamation point "!" since this character signals the program to stop decoding input from the command.
- Opposite case letters: treated as blanks
- Last entry, stop parsing: exclamation point !

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$CLEAR (ASAP Macro)

Clears the output buffer so that the scope of future $GRAB commands is limited to only the output that follows.

Syntax

```
$CLEAR
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$COPY (ASAP Macro)

Does a raw byte-for-byte copy of the source file to a destination file.

Syntax

```
$COPY source destination
&COPY   s        d
$MOVE
&MOVE
```

Remarks

- Does a raw byte-for-byte copy of the source file (name **source** or number **s**) to a destination file (name **destination** or number **d**).
- If the destination file does not exist, it is created. Otherwise, **$COPY** overwrites it while **&COPY** appends to it.
- The command works with all file types (text and binary), but it usually is not recommended for appending with one or both files if they are binary**.**
- The **MOVE** versions behave identically to **COPY** except that the source file is deleted afterwards.

Examples

```
$COPY MODE.DAT 29           !copy mode.dat to
                            bro029.dat
&COPY NEW.LIB UTIL.LIB      !add new library to
                            util.lib
$COPY 30 SAVE.VCR           !copy current 3D vectors
                            to save.vcr
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$DATIM (ASAP Macro)

Toggles printout of date and time.

Syntax

```
$DATIM [ ON  [ ON ] ]
         OFF   OFF
```

Remarks

Macro turns **ON** or **OFF** any date and/or clock time outputs. When turned **OFF**, the system date and/or time routines return blank strings.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$DBG (ASAP Macro)

Creates a debug listing of parser output.

Syntax

```
$DBG  [ ... ]
      [ ... ]
```

Remarks

- Macro displays how the parser decoded a line of input into entries.
- The start, length, type, and value of each entry on the line is listed.
- If the line of input is on the **$DBG** macro itself, the line is **not** passed to ASAP.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$DISP (ASAP Macro)

Immediately displays the given binary distribution file.

Syntax

```
$DISP [ ON  ]
        OFF
        file
```

Remarks

- Immediately displays (via the IMAGER switch or environment variable) the given binary distribution "file" (default BRO009.DAT or "file.DIS").
- Otherwise, turns ON or OFF the automatic displaying of such files immediately after their creation.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$DO (ASAP Macro)

Loops through input record(s).

Syntax

```
$DO [ i [ j [ k ] ] ]
&DO
   ... [ ? ] ...
         |
```

Remarks

- Macro reads and runs the next input record(s) as many times as necessary to satisfy the starting, ending, and increment parameters.
- The integer increment **k** defaults to either +1 or -1 depending upon the values of **i** and **j**.
- The sign of **k** is always set to be the same as the sign of **j-i**, and so the loop is always run at least once.
- For **$DO** the question mark "**?**" is the loop counter; for **&DO** the vertical bar "**|**" is used instead.
- The loop is terminated only when the counter reaches or exceeds the ending value.
- Only one level of looping per macro nesting level is permitted; for multiple looping it is necessary to jump to another macro level.
- An input error, a $GO or a $LEAVE command in the record forces a premature exit from the loop.
- It is not necessary to reference the loop counter in the record(s).
- **By default only the next record is repeated***;* if all of the commands to be looped cannot fit on the next record, you may put the commands into a macro and then loop over the macro. Alternatively, you may enclose multiple looping records between curly braces.
- You cannot implement nested $DO loops directly. However, you can define a macro with a $DO loop, and the macro can contain a $DO loop.

Example

```
!! TO CONSTRUCT A NESTED LOOP AND UPDATE
!! GRAPHICS TITLE WITH INCREMENT VALUES
!! [date]
MACC { 1
$DO 2 3
TITLE 'BTB=#1 BCB=?'
 SHOW TITLE
}
$DO 0 2
{ &MACC LIT[?]
  }
SURFACE
 PLANE Z 0 RECT 1 1
OBJ
WIN Y X; PRO
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$ECHO (ASAP Macro)

Controls display of input commands.

Syntax

```
$ECHO [ ALL ]
        NONE
```

Remarks

- Macro echoes (redisplays) **ALL** input commands or **NONE**. Macro overrides the control of echoing specified on the call to a user macro.
- A **$ECHO** command by itself (no argument) restores this default mode.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$EDIT (ASAP Macro)

Calls user-specified editor from within ASAP.

Syntax

```
$EDIT filename.ext

$EDIT macroname

$EDIT [ 'string' ]
```

Remarks

- When editing an entire file, BRO recommends reinitializing ASAP before the new file is loaded (or you will end up with a composite of old and new data). You can initialize ASAP with the following commands: SYSTEM NEW, RESET, RAYS 0.
- **$EDIT filename.ext** edits the given text file specification, which must contain the period ".". If the file is already open, it is first closed, then edited, reopened. ASAP prompts you before positioning the cursor at the end of the file.
- **$EDIT macroname** edits the specified macro, extracting it from the current library if necessary. After exiting the Editor, ASAP asks if you want to replace the old macro definition with the new one, and if you want to run the macro immediately.
- **$EDIT 'string'** edits the current input file (as specified on the last **$IO** command) or only the part delimited by blank lines that contains the given **'string'**. After exiting the editor, you are asked if you want to replace the old part of the input file with the new one, and if you want to run all of the input file, only the edited part, or none of it. Note that no application initialization is automatically done before re-running the input file. You must do this with the appropriate application commands.
- Select any editor available on your system to do online editing by specifying the command line with the **EDITOR** switch or environmental variable. Otherwise,

OFF	Suppresses screen graphics
ASK	Prompts at end of each screen for plotting and/or saving
EACH	Plots each screen without prompting
NORM	Restores default (screen graphics and no prompting)

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$ERR (ASAP Macro)

Branches on an input error.

Syntax

```
$ERR label
     :
label
     :
```

Remarks

- Macro sets a flag such that when an input error occurs, ASAP skips records until one starting with the '**label**' string is found. Normal input processing resumes on the next record, and the flag is reset.
- The **label** must start in column 1 of the input record.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$EVAL (ASAP Macro)

Evaluates a given $FCN function.

Syntax

```
$EVAL    vary a b n [ vary' a' b' n' ...]  func [ func' ...] [ 'title' ]
    [ m ]         d               d'
```

Remarks

- The first format evaluates the given $FCN functions while changing the internal register **vary** from **a** to **b** in **n** steps (up to 10000). Actually,

$$
\begin{aligned}
&\text{if } n > 0\\
&\text{vary} = a + (i - .5)\frac{b - a}{n} \qquad i = 1, n\\
&\text{or if } n < 0\\
&\text{vary} \;=\; a + (i\text{-}1)\frac{(b - a)}{(|n| - 1)} \quad i = 1, |n|
\end{aligned}
$$

- Up to 40 variables are permitted. The resulting values in the function **func** (or the sum of the squares, if more than one given) at the first three levels are continually written to a BRO binary distribution file called EVAL.DIS or macro.dis (with optional **title**) for later processing. One additional evaluation is done with the registers reset to their values at which the discrete value of "func" was minimum.
- The second format interates the next input record either **m** times while changing the **vary**s randomly, or to approach the actual minimum of the sum of the squared **func**s (up to 125). If **m** is specified the **d**s are the probability distribution types; that is,

$$
\text{vary} = a\left(1 - x\right) + bx \qquad x = \frac{(1 + \text{RAN}[d])}{2}
$$

  Otherwise, if **m** is not specified, **d**s are fractional derivative increments, relative to the ranges **a b** that are used to build a change matrix, which is solved by a SVD technique. Double-sided derivatives are computed to approximate a damping factor from the non-linearity predicted by the homogeneous second derivatives. Therefore, the required number of evaluations is 2*(variables+1); that is,

$$
\begin{aligned}
&\text{initial state}\\
&\text{vary - d}\left(\frac{b - a}{2}\right)d < 1\\
&\text{vary} + d\left(\frac{b - a}{2}\right)\\
&\quad\vdots \qquad d' < 1\\
&\quad\vdots\\
&\quad\text{predicted min}
\end{aligned}
$$

- The number of **func**s should be greater than (>) or equal to (=) the number of **vary**s for it to find a unique solution. For nonlinear problems, successive **$EVAL** commands may be needed to reach the precise minimum.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$EXIT (ASAP Macro)

Starts immediate program termination.

Syntax

```
$EXIT
```

Remarks

- **$EXIT** is supported only in batch mode, not in the user interface.
- Macro immediately terminates program and returns control to the operating system. If the program is in interactive mode, you are asked to verify your decision to terminate program. The **$EXIT** macro bypasses the normal ASAP execution shell program. As a result, it **does not** rename certain files from their BRO0XX.DAT designation.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$EXP (ASAP Macro)

Switches the expression precedence.

Syntax

```
$EXP [ OLD ]
       NEW
```

Remarks

- Instructs the parser to recognize the **OLD** left-to-right or the **NEW** operator precedence expression syntax; the default is **NEW**.
- When **$EXP** is set to **OLD**, consecutive operations are always evaluated from left to right with no operator precedence until a delimiter terminates the expression. Nested parentheses or brackets can be used when necessary.
- The **$EXP OLD** should precede any arithmetic operations or variable designations defined using the **OLD** left-to-right precedence expression syntax. Users with .INR files and macros created prior to version 4.0 who do not want to update the file syntax should include this macro in their .INR files.
- To reset ASAP to the default, BRO recommends that users also include a **$EXP NEW** command at the end of a file, immediately before the **END** command.
- If you have an INR file created prior to version 4.0, BRO recommends you include **$EXP NEW** in your defsetup.inr file so that all new files run properly. The defsetup.inr file runs at startup.
- **$EXP** may be used at any time to switch between expression syntax parsers.
- If no argument is given, the state before the last **$EXP** command is restored.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Preferences dialog for definition of Default Setup File (defsetup.inr)

$FAST (ASAP Macro)

Performs fast reading of numeric arrays.

Syntax

```
$FAST n   [ m ]   [ s ]   [ COMPLEX ]
                 label
  . . .    n or 2n numbers
      :
   m records
 [ label ]
```

Remarks

- Macro reads directly from the current input file **m** records (default 1 or until **label** is reached, or until read error) of **n** real or COMPLEX entries each.
- ASAP uses the flexible, list-directed input facility of Fortran and, therefore:
  1. ASAP does not do the extensive parsing it normally does; that is, only numbers separated by a comma and/or blanks are accepted.
  2. This input is never echoed to the output device.
  3. If reading directly from a file and not indirectly from a macro or loop, ASAP attempts to read **n** times **m** numbers, with line breaks required only after every **n**th number; that is, additional line breaks can be inserted so that lines are not limited to the normal 344 characters.
- Each number of the array optionally may be multiplied by a scale factor **s** (default 1).
- Optionally, ASAP can also assume each successive pair of numbers forms a COMPLEX (real, imaginary) entry**.** In this case, rule 3 above should be read with **2n** substituted for **n.**
- **FAST** may be used in user-defined macros.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$FCN (ASAP Macro)

Defines new in-line math functions.

Syntax

```
$FCN [ name [ e e' e" ... ] ]
```

Remarks

- Macro defines a new in-line math function with up to a 32-character **name**. The new function may be used in future arithmetic expressions just like the intrinsic functions **SIN, COS, SQRT,** and so on.
- The function is defined by the expressions entered after the name. Up to 60 of these functions can be defined at any one time. The result of the last expression is the function value returned during execution.
- The dummy argument of the function is represented in the defining expressions as the base argument register "**_**". For example:

  $FCN SECH 2/(EXP(_)+EXP(-_))
- If no expressions are given, the function, **name** is deleted from internal storage. A **$FCN** command with no other entries deletes all current user functions.
- These user-definable functions are recursive; that is, you may include one in the argument of another. Also, any registers/variables defined in the function before they are used are automatically assumed to be local to that instance of the function, and therefore do not conflict with any external (for example, global) ones of the same name.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

$EVAL (ASAP Macro)

$FF (ASAP Macro)

Inserts a form feed into the output stream.

Syntax

```
$FF [ text ]
&FF   k
```

Remarks

- If **$** prefix is used, a page advances or form feed is sent to the current output unit. Any text present on this command (up to 62 characters) is also printed at the top of the page along with the current date and time.
- The integer entry form outputs only a form feed to logical unit **k** instead.
- If the **&FF** syntax is used, only the text (up to 80 characters) that is preceded and followed by single blank lines is written.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$GO (ASAP Macro)

Branches to other records or skips over input records.

Syntax

```
 $GO  m
     +m
     -m

$GO label
     :
label
     :
```

Remarks

- Macro allows you to branch or skip records unconditionally.
- If **m** is unsigned, the next record to be run is macro record **m**.
- If **m** is signed, the next record to be run is the next record plus or minus **m**.
- If the target record goes beyond the end of the macro definition, the macro run is terminated after the current record is processed.
- If a ‘**label**‘ is entered, ASAP skips over input records until it finds a record with the ‘**label**‘ starting in column 1. Normal input processing begins on the next record.
- You must first rewind the file to branch back to a previous ‘**label**‘ in the input file.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$GRAB (ASAP Macro)

Grabs isolated numbers or literals from the output buffer.

Syntax

```
$GRAB 'string' [ i j ] [ reg [ reg '...]]
&GRAB
```

Remarks

- The macro $GRAB (numbers) or &GRAB (literals) (blank delimited strings) searches backwards through the output buffer for the given '**string**' (delimited like a comment.) It then displays or assigns to the given register the **j**th number or word after the string, or the **j**th number or word of the **i**th relative record.
- Successive numbers (or words) may also be assigned to additional registers.
- A nonfatal warning is issued, and the variable is set to zero if a number is not found.
- The default values are: **i**=0 (found record) and **j**=1 (next number or word).
- If you want to specify **j** explicitly, also enter **i**.
- Literals must be stored in one of the 286 direct registers designated by the letters A through Z by themselves or followed by the numbers 0 through 9.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$GUI (ASAP Macro)

Sends the given command strings to the GUI front-end. The **$** form sends the strings exactly as entered. The **&** form parses any expressions and replaces their results with string equivalents.

The following syntax are used with Remote Start, a feature of ASAP/Optical, to specify the directory in which Remote will start.

Syntax

```
$GUI command  [ command' ... ]
&GUI
```

The following section applies to some Windows® and ASAP commands.

Where ComputerName is used in the following commands, these four forms of ComputerName can be used:

```
computer
\\computer
computer.domain.com
IP address, for example, 125.0.156.120
```

$GUI RemoteStart \\ComputerName [workdir]

Starts a remote kernel session on the named computer. An output tab for this named computer is added to the Command Input window. When this tab is selected, all commands typed from the Command Input window are directed to its kernel. In the following example the remote computer name is “tut1”.

Example:

```
$GUI RemoteStart \\tut1 "C:\foo"
```

If the optional working directory is specified, this directory is used for the remote session. The default working directory is located in Documents and Settings\All Users\Application Data. Once the working directory is set for a remote session, it cannot be changed. This restriction limits potential conflict with other remote kernels running on one machine.

$GUI RemoteEnd \\ComputerName

Ends a previously started kernel session on the named computer. Its Command Input window tab is deleted and some final status from the kernel is posted to the local computer's Command Input window. In the example, the remote kernel is ended on the computer named, “tut1”.

Example:

```
$GUI RemoteEnd \\tut1
```

$GUI RemoteCommand \\ComputerName Command

Issues an ASAP command to the remote computer. Commands are placed in a queue on the remote session. In this example, the remote ASAP seed is set to a value different from the default value with the SEED command, using the QUASI option.

Example:

```
$GUI RemoteCommand \\tut1 SEED 987654321 QUASI
```

$GUI RemoteSendFile \\ComputerName "Filename"

Sends a file(s) from the local computer to the named remote computer. If this command is used from a remote computer, and also a period is used in place of a computer name, the file(s) are transferred from the remote computer to the local computer. Transferring files form one remote session to another remote session is not possible. Full wildcard use is supported, although it cannot recursively act upon sub-directories. Files sent from a remote computer to the local computer are placed in a sub-directory with the same name as the remote computer.

Example:

```
$GUI RemoteSendFile \\tut1"*.lib"
$GUI RemoteSendFile . "*.dis"
```

$GUI RemoteGetFile \\ComputerName "Filename"

Gets a file(s) from the named remote computer and places it in a directory with the same name as the remote computer. If this command is used from a remote computer, and a period is also used in place of a computer name, the file(s) is transferred from the local computer to the remote computer. Transferring files form one remote session to another remote session is not possible. Full wildcard use is supported, although it cannot recursively act upon sub-directories.

Example:

```
$GUI RemoteGetFile \\ComputerName "greatstuff.dis"
$GUI RemoteGetFile . "*.inr"
```

$GUI DisplayRange fMin fMax

Changes the minimum and maximum range for the last opened Display Viewer window. No error checking is done to ensure that the minimum is less than the maximum.

Example:

```
$GUI DisplayRange -5.25 10.25
```

$GUI RemoteBusy \\ComputerName regname

Tests whether the kernel session on the named remote computer is busy executing a command, or if it is idle, waiting for input. The result, 1 or 0, is placed in the register regname. The regname register must have been previously declared.

Example**:**

```
$GUI RemoteBusy \\tut1 ISBUSY1
```

$GUI RemoteSetPriority \\ComputerName Low, Normal, High, Realtime

Changes the process priority for the kernel session on the named computer while it is busy executing commands. A priority of Low ensures that the kernel will use only CPU cycles when other processes are idle. A priority of Normal puts the kernel session on the same level as all other processes. A High priority causes the kernel session to have a slightly higher average amount of CPU time as compared to all other processes. A Realtime priority causes the kernel session to become the most important task for the operating system. All other processes basically get only idle time processing, which, due to the nature of the kernel, is almost never. Setting the priority to Low is the most typical use, so that a remote computer is not burdened with extensive CPU cycles, which a kernel session may require when running in a normal priority. Normal Windows usage can therefore be expected by the average user.

Example:

```
$GUI RemoteSetPriority \\tut1 Low
```

$GUI RemoteSetCPU \\ComputerName cpu

Sets the ideal CPU on which the kernel will run on the named remote computer. This setting does not guarantee that the process will run only on the selected CPU, but this CPU will be the preference. CPUs are numbered starting with zero.

Example:

```
$GUI RemoteSetCPU 1
```

$GUI Plot Link

Forces all plot windows that are created by the kernel to become links that can be clicked in the Command Output window.

$GUI Plot Off

Disables all dynamic plot creation by the kernel.

$GUI Plot On

Restores the dynamic plot creation ability of the kernel.

$GUI Echo Off

Turns off the echoing of information from the kernel to the GUI.

$GUI Echo On

Restores the echoing of information from the kernel to the GUI.

$GUI ClearConsole

Clears text from the Command Output window.

$GUI ClearWarnings

Sets to zero the Warnings counter on the status bar.

$GUI ClearErrors

Sets to zero the Errors counter on the status bar.

$GUI ClearCounters

Sets to zero both the Errors and and Warning counters on the status bar.

$GUI ClearAll

Performs both the ClearConsole and ClearCounter options.

See Also

ASAP Remote

$HELP (ASAP Macro)

Displays information on all or one command or a listing of the MACROS in the current library.

Syntax

```
$HELP [ command ]
```

Remarks

- If no entry is given, an abbreviated list of all commands is displayed.
- If only one command (up to two literals) is specified, a complete one or two paragraph explanation of the command is shown.
- The &HELP form displays only the format of the command with no accompanying text.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$IF (ASAP Macro)

Creates conditional processing of a block of records or block structures.

Syntax

```
$IF a rel b [ log a' rel' b' log' ... ] [ n  ]
```

Remarks

- Macro provides a means to optionally run a block of **n** records depending upon the result of some combination of relational/logical operations.
- Valid relational and logical operators include those outlined below:

relational operators	
EQ	equal
LT	less than
GT	greater than
NE	not equal
GE	greater than or equal
LE	less than or equal
logical operators	
AND	logical and
OR	logical or
EQV	equal in logical value
NEQV	not equal in logical value
XOR	exclusive or (same as **NEQV**)

- Normally the relational operators compare two floating-point (real) entries or two full 24-character literals (including trailing blanks). However, if any upper-case, alphanumeric character is appended to the operator name, then truncated forms of the two entries are compared (that is, the integer equivalents of the two numerics or the literals truncated to the smaller of the two).
- If the overall **$IF** result is true, the next **n** input records are processed, including any commands that follow the **$IF** on its record. Otherwise, they are skipped over and processing resumes on the next record.
- The default value of **n** is 1; that is, only the next record is processed if the expression is true. However if **n** is not specified and more commands follow the **$IF** on its record, then **n** is set to zero.
- The **THEN** keyword is used to trigger processing of conditional block structures. These structures may be nested up to 10 levels. **NOTE***:* The control words *must* be uniquely and identically indented at each nesting level.

Example of Truncated Comparisons

`1.75 EQI 1.`	is true	`1.75 EQ 1`	is false
`YES EQS Y`	is true	`YES EQ Y`	is false

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

$IF THEN (ASAP Macro)

$IF THEN (ASAP Macro)

Use the THEN keyword at the end of a $IF statement to trigger processing of conditional block structures.

Syntax

```
 $IF a rel b [ log a’ rel’ b’ log’ ... ] THEN
:
 [ $ELSEIF ... THEN ]
:
 [ $ELSE ]
:
 $END[IF]
```

Remarks

These structures can be nested up to 99 deep. However, the control words MUST be uniquely and identically indented at each level of nesting. This mandatory indentation significantly speeds processing and is also a good standard programming practice. The following is an example of an IF structure inside a DO loop block.

Example

```
$DO 1 3
  { $IF ?\2 EQ 0 THEN
       A?=?
    $ELSE
       B?=?
    $ENDIF }
```

Remarks

- Notice the required vertical alignment of the $IF,$ELSE,$ENDIF commands.
- Using a $GO to jump into a block may cause unexpected results.
- Only a forward $GO to a label can be safely used to jump out of a block.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$IO (ASAP Macro)

Controls I/O redirection.

Syntax

```
$IO iotype n           iotype' ...
           literal
           file [ n ]
$IO [ n ]
```

Remarks

- Macro redirects the ASCII I/O specified by the **iotype** to another **file** or logical unit number **n**.
- The significance and defaults are described in the following table.

Input/Output Type	*N*	*Default Name (*.DAT)*	*Extension*	*Description*
INPUT	**1**	BRO001	INR	Default command input
LIBRARY	**24**	BRO024	LIB	Macro library (new precedence expressions)
MACRO	**24**	BRO 024	MAC	Macro library (old left-right expressions)
OUTPUT	**6**	BRO 006	OUT	Default nongraphical text output
GRAPHICS	**6**	BRO 006	OUT	Character graphics output (off by default)
USER	**7**	BRO 007	USR	Archive of interactively entered commands
*	**8**	BRO 008		virtual.pgs
*	**9**	BRO 009		Default distribution data file
PLOT	**20**	BRO 020	PLR	Default 2-D plot instruction output
LIBRARY	**24**	BRO 024	LIB	Macro library command input (new expression parser)
*	**29**	BRO 029		Complex optical field data from **FIELD**
VECTOR	**30**	BRO 030	VCR	Default 3D vector instruction output
FILE	**-**			Arbitrary file (user must specify **n**)*

** I/O cannot be redirected*

- An **iotype** entry without any trailing arguments simply closes that unit; to reopen it, type **$IO iotype n** where **n** is the file unit number.
- If no entries are present, **$IO** toggles the program input between your console and the current input unit assignment. This allows you to insert interactive breakpoints in the input file.
- When directing **OUTPUT** to a file, it is by default also echoed to the Command Output window. If the word **ONLY** is placed after the file name instead of the unit number, nothing is sent to the console. For example,

```
$IO OUTPUT file n      !both to file and console
$IO OUTPUT file BOTH   !same
$IO OUTPUT file ONLY   !just to file
```

- A prefix character for each number or a literal determines which of the following operations are performed before the unit is used:

Literal	*Numeric Form*	*Operation*
APPEND	**+n**	Unit **n** is positioned at the end of the file

CANCEL	**0**	Output to **iotype** (**GRAPHICS, PLOT, VECTOR**) is suppressed
CHANGE	**n**	Current position of unit **n** is not changed
CLOSE	**None**	Unit associated with **iotype** is closed
DELETE	**0n**	Unit **n** is closed and the associated file is deleted
REWIND	**-n**	Unit **n** is positioned at the beginning of the file

Example

To understand the relationship between unit number and file name, consider the following example.

$IO OUTPUT JOE 2

is equivalent to the Fortran expression:

OPEN (unit=**2**, file="**JOE.OUT**",. . .).

In this special case, text is directed to the file JOE.OUT and to the screen. Without the unit number, text is directed only to the file.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$ITER (ASAP Macro)

Iterates input records and a set of variables.

First Syntax

```
$ITER      vary a b n [ vary' a' b' n' ... ] [func
[ func' ... ] ] [ 'title']
      [ m ]        d              d'
   ...
```

Second Syntax

```
$ITER vary a b n [ vary '
...] func [ func' ... ]
  { ...
       :
        ... }
```

Remarks, 1st syntax

- The first format of **$ITER** iterates the next input record, usually a macro, or brace delimited block, while changing the internal register **vary** from **a** to **b** in **n** steps (up to 10000):

$$\text{if } n>0$$

$$vary = a + (i-.5)\frac{(b-a)}{n} \qquad i = 1, n$$

$$\text{or if } n<0$$

$$vary = a + (i-1)\frac{(b-a)}{(|n|-1)} \qquad i=1,|n|$$

- Up to 40 variables (levels of iteration) are permitted so that the next input record is run "1+**n***n'***. . .** times.
- If specified, the resulting values in the register **func**, or the sum of the squares if more than one is given, at the first three levels are continually written to a BRO binary distribution file called **iter.dis** or **macro.dis** (with optional **title**) for later processing. (The last iteration resets the registers to their values at which the discrete value of **func** was minimum.)
- The second format iterates the next input record either **m** times while changing the **vary**s randomly, or in order to approach the actual minimum of the sum of the squared **func**'s (up to 125). If **m** is specified, the **d**s are the probability distribution types That is,

```
vary = a*(1-x) + b*x      x=(1+RAN[d])/2
```

Otherwise if **m** is not specified, the **d**s are fractional derivative increments relative to the ranges **a b**, which are used to build a change matrix that will be solved by a SVD technique. Double-sided derivatives are computed to approximate a damping factor from the non-linearity predicted by the homogeneous second derivatives. Therefore, the required number of iterations is 2*(variables+1); that is,

```
initial state
vary - d*(b-a)/2d < 1
vary + d*(b-a)/2
     :d'< 1
     :
predicted min
```

- The number of **func**s should be greater than (>) or equal to (=) the number of **vary**'s to find a unique solution. In the case of nonlinear problems, successive **$ITER** commands may be required to reach the precise minimum.
- Multiple record iteration loops must be enclosed in braces; that is, the next record after the **ITER** and the first record of the block must start with an open brace **{** and the last must end with a closed brace **}**.

Remarks, econd syntax

- Use **&ITER** instead of **$ITER** to automatically cancel output during loop execution and restore it when completed.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

$ITER Overview

$LEAVE (ASAP Macro)

Causes premature exit from a loop, macro, or input file.

Syntax

```
$LEAVE [ n ]
```

Remarks

- Macro forces an immediate exit from the current **DO** loop, macro, or input file in that order.
- The **n** is the number of construct levels to exit (default is 1). If the above loop was also inside a macro and you wanted the test to exit not only the loop but also the macro, then set "n" to 2.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$LOC (ASAP Macro)

Makes the given registers/variables names local to the current macro. In other words, on exit from the macro, the values of these registers will be automatically restored to their values previous to the $LOC command.

Syntax

```
$LOC reg [ reg’ reg" ... ]
```

Example

```
MY_MACRO {
$LOC A,B,C
A=#1 B=#2 C=#3
:
}
```

Therefore, MY_MACRO will not (permanently) change the values of the registers A, B, and C when it is called.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$MENU (ASAP Macro)

Displays a menu of currently defined macros for selection.

Syntax

```
$MENU [ m ] [ 'title' ]
    &MENU
```

Remarks

- Macro displays a menu of macros present (and their descriptions, if present in the macro definition) in the currently attached macro library, from which a selection may be made.
- **$MENU** displays a scrolling list box from which a selection may be made by highlighting the name of the desired macro and then clicking OK. The *m* argument has no effect on **$MENU**.
- **&MENU** displays the menu in a graphics text window. Select the desired macro by placing the graphics cursor (crosshair) over the macro and pressing the left mouse button. (In the current release of ASAP, the menu window is not automatically deleted after the selection is made.)
- The menu entries are displayed horizontally in columns by default. The *m* is the maximum number of columns to use (default 6). For example, entering a 1 would force the menu entries into one vertical column.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$NEXT (ASAP Macro)

Immediately begins execution of the next $DO or $ITER iteration by forcing a branch to the top of the current loop block.

Syntax

```
$NEXT
```

Example

```
$DO 1 10
{ :
This block is executed every iteration
:
$IF ? LE 5; $NEXT
:
This block is executed only for iterations 6 through 10
:
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$PAGE (ASAP Macro)

Pauses text output.

Syntax

```
$PAGE [ n   ]
        OFF
```

Remarks

- Macro causes future text output to be paused every **n** lines (default 12); that is., it waits for you to press Enter to continue.
- Pausing can be turned OFF with this command or by typing a caret “^” at a pause.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$PATH (ASAP Macro)

Sets the default directory "path" for future file openings.

Syntax

```
$PATH [ "path" ]
```

Remarks

- If no path is given, then reverts to the previous one.
- "path" must include the closing directory character (\, / or] depending on the operating system).
- The default files (see $IO command) are closed in the old directory, and then reopened in the new one.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$PLOT (ASAP Macro)

Used to control screen graphics and plots.

Syntax

```
$PLOT [ OFF   ]
        ASK
        EACH
        NORM
```

Option	Description
no argument	Immediately processes all the plots in the current 2-D plot file using the operating system command defined with the PLOTTER switch or environment variable.
OFF	Suppresses screen graphics
ASK	Prompts at end of each screen for plotting and/or saving
EACH	Plots each screen without prompting
NORM	Restores default (screen graphics and no prompting)

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$RAN (ASAP Macro)

Resets the random number seed for the "~" operator to the integer "i" (default 2000000001).

Syntax

```
$RAN [ i ] [DEC]
      BEST
```

Remarks

- The default uniform random number generator (upon which all others are based) is identical to that used on the old DEC (Digital Equipment Corporation) systems. One can switch to the BEST possible generator based on the same algorithm but using different numerical constants and thus a different seed sequence.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$READ (ASAP Macro)

Start reading future records from the beginning of the given file "name" (with default extension .IN?).

Syntax

```
$READ name
```

Remarks

- Temporarily reads input from a given file.
- If end-of-file is reached, return to calling macro or file automatically.
- $READ commands, in addition to other commands that read data lists, can be nested directly or indirectly.
- File names are limited to 24 alphanumeric characters unless the name is enclosed in double quotation marks ("name").

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$REG (ASAP Macro)

Displays the contents of registers and associated variables.

Syntax

```
$REG [ register register' ... ] [ 'text' ]
&REG
```

Remarks

- Macro displays the contents of a given register or set of registers with optional **'text'**.
- If no registers/variables are entered, all of the registers/variables whose contents have changed since program startup are displayed.
- **$REG** prints out the internal ASAP register name in addition to the variable name(s) and value(s). **&REG** prints only the variable name(s) and value(s).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Registers for Storing Arithmetic Results

$SCR (ASAP Macro)

Defines a user-programmable screen template.

Syntax

```
$SCR [ name ] [ at [ aw [ ar [ au ] ] ] ]  [
'title' ]
            m
```

Screen definition file extension	SCR
Maximum screen size	79 x 24 characters
Register field start delimiter	****
Register field end delimiter	**\ ?**
Literal field designator	**"**
Integer field designator	(nothing at end of register)
Floating point designator	**.**
Boolean field designator	**:**

Remarks

- Macro may be used to define a screen template from which any group of variables may be displayed or modified by editing their assigned fields.
- The template may be read from a file called **name.scr** or from the **m** records that follow.
- In the former case and instead of the normal end-of-file, a line with only a closed brace "}" in column one can be used to terminate the template definition and signal the beginning of up to 1000 lines of 79 character-wide help text.
- A colon, period, double quote or nothing at the end of the register name determines the register data type; Boolean, floating point, literal or integer, respectively. If floating, a single-digit number to the right of the period determines the number of digits to the right of the decimal point that will be displayed. The overall size of the field, and the register value displayed in it, are controlled by the position of the second delimiter.
- If there are no editable fields in the template (no question marks "?"), the result is written directly as simple text to the current output device.

Screen definition file extension:	SCR
Maximum editable screen size:	79x24 characters
Maximum output template size:	344x24 characters
Register field start delimiter:	****
Register field end delimiter	**\ ?**
Literal field designator	**"**
Integer field designator:	
Floating point designator:	**.**
# digits right of decimal point:	**.#**
Boolean field designator:	**:**

- Within the screen template, each variable field is delimited by either two backslashes (\\ ) for a display-only field, or a backslash and a question mark (**\?**) for modifiable fields. The size of the field and the variable value displayed in it are controlled by the position of the second delimiter.
- The display attributes of the background and fields can be programmed according to the following table:

Entry	Default	Meaning
at	7	Background text
aw	7	Write only fields
ar	0	Read fields before
au	0	After update

- The Tab and Shift-Tab keys are used to move the cursor within the dialog.
- Press **Enter** after each entry, and click **OK** or **Cancel** as appropriate to exit the dialog.
- The screen is displayed in the upper-right corner.
- To display the CANCEL button, enter **SCR_CANCEL=1** before the $SCR command. If SCR_CANCEL equals 0, the button does not display.
- If you need text or display-only fields, enter **DUMMY : ?** at the end of the .SCR file to display the window.
- If you click **OK**, SCR_CANCEL becomes 0.
- If you previously clicked **OK**, reset SCR_CANCEL to 1 before using it each time. Otherwise, the next $SCR will not display a CANCEL button.

- After the $SCR line, enter:

  $IF (SCR_CANCEL) EQ 1; $GO usercancel !! or a similar jump to skip the code that you do not want to execute!!
- If **OK** was previously clicked, reset SCR_CANCEL to 1 before using it each time. Otherwise, the next $SCR will not have a CANCEL button.
- When the screen template displays, the **OK** and **RESTORE** buttons are visible. Click **OK** if you want the INR file to continue from the point the $SCR command was issued. If you entered values but want to return to the original settings, click **RESTORE**. This action restores the default values that were in place when you issued the $SCR command.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

See Also

Screen (SCR) Editor

$SHOW (ASAP Macro)

Displays the status and states of several internally defined macros.

Syntax

```
$SHOW
&SHOW
```

Remarks

- Macro shows the current states of internal macro commands.
- The **&** version also lists the internal stack code (operators/operands) for any $FCN definitions.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$STO/$RCL/&STO (ASAP Macro)

Stores or recalls variable data to or from a file.

Syntax

```
$RCL [ file ]
$STO          [ reg reg' ]
&STO
```

Remarks

- Stores or recalls the range of given registers (or the entire register set) to or from a binary($) or text(&) "file") The default file is LASTVALS.REG.
- **$RCL** reads in either a binary or text file. If it does not recognize the requested format, it tries the other.
- **&STO** writes the .REG file in an ASCII format.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$SYS (ASAP Macro)

Runs a system command or opens an operating system window.

Syntax

```
$SYS [ command line ]
```

Remarks

- Macro, if entered without a command line argument, has ASAP open a DOS window through which you can enter DOS commands. This window remains open and visible on the screen until it is closed with an $EXIT command.
- If this command is entered with a command line, a DOS window is opened, the command is run, the window is closed immediately, and control returns to ASAP. Given the short time the window is visible on the screen, this syntax is best for commands that do not require much user interaction, such as the **DEL** command under DOS.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$TIC (ASAP Macro)

Displays elapsed and CPU time.

Syntax

```
$TIC
&TIC
```

Remarks

- **$TIC** macro displays the number of CPU seconds that have elapsed since the last **$TIC** or **&TIC** macro was processed.
- **&TIC** macro displays nothing and resets the timer.
- Normally the time "units" are allowed to float, but can be fixed for this and all future outputs to MICroseconds, MILliseconds, SEConds, MINutes, HOUrs, DAYs, WEEks or YEArs.
- Use OFF to go back to the default floating units. The display resolution is always 1/100 of the selected units and is limited to less 10,000 total units.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$UNVAR (ASAP Macro)

Determines the type of message to be issued for future uninitialized (used before set) variables/registers.

Syntax

```
$UNVAR [ NONE ]
         WARN
         ERROR
```

Remarks

The default is WARN, that is, a message but not a fatal ERROR. If no entry is given, the state before the last $UNVAR command is restored.

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$VIEW (ASAP Macro)

Immediately processes the system geometry in the current 3D vector file (*.vcr) using the operating system command defined with the VIEWER switch or environment variable.

Syntax

```
$VIEW
&VIEW [ file ]
```

Remarks

If a 3D vector **file** name is given, it is processed by either a new instance of the 3D Viewer (**$**), or it is added to the contents of the current 3D Viewer (**&**).

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html

$WAIT (ASAP Macro)

Causes the program to wait "s" seconds (default is 5 seconds) before continuing. An optional "message" can also be displayed.

Syntax

```
$WAIT [ s ] [ ’message’ ]
```

Examples

See the Index of Example Scripts in <install directory>\projects\examples\examples_scripts.html