

Přehled důležitých příkazů OSLO

lens_spreadsheet (lse) command

Opens the [surface data spreadsheet](#).

Command Syntax:

```
cmd lens_spreadsheet
  (Spr_access_mode,
   Surface_mode,
   Number_of_surfaces,
   Lens_name)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Spr_access_mode { list = Lse_modes, default (noquery) = "upd"}
1	int	Surface_mode { list = Surface_options, default (noquery) = "rd"}
2	int	Number_of_surfaces { lolim = 0, hilim = pm(msrf), default (onprompt) = 1}
3	char	Lens_name [] { lolim = 0, hilim = 32, default (noquery) = -300}

List Definitions:

```
list Lse_modes
  {"{upd}Update" = [1]0,
   "{new}New"    = [1,2,3]1}
list Surface_options
  {"{cv}Curvature" = 0,
   "{rd}Radius"    = 1}
```

th command

Sets [thickness](#) of Surface, i.e., the distance along the local z-axis to the vertex of the next surface.

Command Syntax:

```
cmd th
  (Surface,
   Thickness)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surface { lolim = 0, hilim = imsnbr}
1	real	Thickness { default (onprompt) = 0.0}

rd command

Sets [radius of curvature](#) of Surface.

Command Syntax:

```
cmd rd
  (Surface,
   Radius)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surface { lolim = 0, hilim = imsnbr}
1	real	Radius { default (onprompt) = 0.0}

ap command

Sets the radius of the [circular aperture](#) for Surface. If the second argument is "chk", the aperture will be marked as a "checked" aperture for [vignetting](#).

Command Syntax:

```
cmd ap
  (Surface,
   Aperture_checking,
   Aperture_radius)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surface { lolim = 0, hilim = imsnbr}
1	int	Aperture_checking { list = Aperture_checking, default (noquery) = "noc"}
2	real	Aperture_radius { lolim = 0.0, default (noquery) = 0.0}

List Definitions:

```
list Aperture_checking
  {"{noc}Not checked" = 0,
   "{chk}Checked"     = 1}
```

atp command

Sets the type of [special aperture](#): elliptical, rectangular, triangle, or quadrangle.

Command Syntax:

```
cmd atp
  (Surface,
   Special_aperture_id,
   Special_aperture_type)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surface { lolim = 0, hilim = imsnbr}
1	char	Special_aperture_id [] { lolim = 1, hilim = 1}
2	int	Special_aperture_type {Region_types, default (onprompt) = "ell"}

List Definitions:

```
list Region_types
  {"{ell}Ellipse"     = 1,
   "{rec}Rectangle"  = 2,
   "{tri}Triangle"   = 3,
   "{qua}Quadrangle" = 4}
```

aac command

Sets the action code for a [special aperture](#): transmit, obstruct, or pass undeviated.

Command Syntax:

```
cmd aac
  (Surface,
   Special_aperture_id,
   Special_aperture_action)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surface { lolim = 0, hilim = imsnbr}

1	char	Special_aperture_id [] { lolim = 1, hilim = 1}
2	int	Special_aperture_action {Region_actions}

List Definitions:

```
list Region_actions
  {"{und}Pass through (hole)" = 1,
   "{obs}Obstruct (absorb)" = 2,
   "{tra}Transmit (refract)" = 4}
```

sur command

Sets [radius of curvature](#), [thickness](#), [glass](#) (must be air, reflector, or a catalog glass), and [aperture radius](#) for Surface.

Command Syntax:

```
cmd sur
  (Surface,
   Radius,
   Thickness,
   Glass,
   Aperture)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surface { lolim = 0, hilim = imsnbr}
1	real	Radius { default (onprompt) = 0.0}
2	real	Thickness { default (onprompt) = 0.0}
3	char	Glass [] { lolim = 0, hilim = 10, default (noquery) = -302}
4	real	Aperture { lolim = 0.0, default (onprompt) = 0.0}

ang command

Sets the (half) [field angle](#), in degrees, for the system. This is the recommended way to define the field of view for an infinite conjugate system.

Command Syntax:

```
cmd ang
  (Half_field_angle)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	real	Half_field_angle { lolim = -180.0, hilim = 180.0, default (noquery) = 0.0000572957795}

ebr command

Sets the [entrance beam radius](#), i.e., the radius of the entering axial beam of light on surface 1.

Command Syntax:

```
cmd ebr  
  (Entering_beam_radius)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	real	Entering_beam_radius { lolim = 1.0e-020, default (onprompt) = 1.0}

air command

Sets the [medium for Surface](#) to air. The [refractive index](#) will be equal to 1.0 for all [wavelengths](#).

Command Syntax:

```
cmd air  
  (Surface)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surface { lolim = 0, hilim = imsnbr}

rfl command

Designates Surface as a [reflector](#) (mirror).

Command Syntax:

```
cmd rfl  
  (Surface)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Glass_name_or_mod { list = Name_or_mod, default (noquery) = "name"}
1	int	Surface { lolim = 0, hilim = imsnbr}
2	char	Glass_name [] { lolim = 1, hilim = 10, default (noquery) = -301}
3	real	Index { lolim = 1.0e-020, optional}

List Definitions:

```
list Name_or_mod
    {"Name"          = 0,
     "{mod}Model"   = 1}
```

cc command

Sets the [conic constant](#) of Surface.

Command Syntax:

```
cmd cc
    (Surface,
     Cc)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surface { lolim = 0, hilim = imsnbr}
1	real	Cc { default (onprompt) = 0.0}

ad command

Sets the fourth-order [polynomial aspheric](#) coefficient.

Command Syntax:

```
cmd ad
    (Surface,
     Ad)
```

Argument Definitions:

Arg #	Type	Name / Definition
-------	------	-------------------

0	int	Surface { lolim = 0, hilim = imsnbr}
1	real	Ad { default (onprompt) = 0.0}

cv command

Sets [curvature](#) (i.e., the inverse of the radius of curvature) of Surface.

Command Syntax:

```
cmd cv
  (Surface,
   Curvature)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surface { lolim = 0, hilim = imsnbr}
1	real	Curvature { default (onprompt) = 0.0}

autofocus (auf) command

Adjusts the thickness of the image surface to focus the system.

Click [here](#) for additional help.

Command Syntax:

```
cmd autofocus
  (Autofocus_option)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Autofocus_option { list = Autofocus_options, default (noquery) = "oas"}

List Definitions:

```
list Autofocus_options
  {"{pxl}Paraxial focus" = 100,
   "{oas}Minimum on-axis spot size (monochromatic)" = 101,
   "{asc}Minimum on-axis spot size (polychromatic)" = 102,
   "{fas}Field-averaged spot size (monochromatic)" = 103,
   "{fsc}Field-averaged spot size (polychromatic)" = 104,
   "{oaw}Minimum on-axis RMS OPD (monochromatic)" = 105,
   "{awc}Minimum on-axis RMS OPD (polychromatic)" = 106,
   "{faw}Field-averaged RMS OPD (monochromatic)" = 107,
   "{fwc}Field-averaged RMS OPD (polychromatic)" = 108}
```

ins command

Insert a [new surface](#) at the specified surface number. The inserted surface is initialized to default values: planar, zero thickness, air, solved aperture radius.

Command Syntax:

```
cmd ins
  (Inserted_surf)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Inserted_surf { lolim = 1, hilim = imsnbr, default (onprompt) = cursnbr}

del command

Delete the specified [surface or range of surfaces](#).

Command Syntax:

```
cmd del
  (First_del_surf,
   Last_del_surf)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	First_del_surf { lolim = 1, hilim = imsnbr, default (onprompt) = cursnbr}
1	int	Last_del_surf { lolim = 1, hilim = imsnbr, default (noquery) = 0}

ims command

Sets the image surface number. The default image surface is the highest numbered [surface for which data has been defined](#).

Command Syntax:

```
cmd ims
  (Image_surf)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Image_surf { lolim = 1, hilim = pm(msrf), default (onprompt) = imsnbr}

ast command

Marks Ap_stop_surface as the [aperture stop](#) for the system.

Command Syntax:

```
cmd ast
  (Ap_stop_surface)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Ap_stop_surface { lolim = 1, hilim = imsnbr, default (onprompt) = 1}

var_spreadsheet (vse) command

Opens the [variables spreadsheet editor](#). If Spr_access_mode is "New", existing variables are deleted.

Command Syntax:

```
cmd var_spreadsheet
  (Spr_access_mode)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Spr_access_mode { list = Lse_modes, default (noquery) = "upd"}

List Definitions:

```
list Lse_modes
  {"{upd}Update" = [1]0,
   "{new}New"    = [1,2,3]1}
```

v (vb) command

Adds, deletes, displays, or modifies entry "Variable_nbr" of the [variables set](#).

Command Syntax:

```
cmd v
  (Variable_nbr,
   V_option,
```

Syst_surf_nbr,
 Config_nbr,
 Variable_type,
 Min_bound,
 Max_bound,
 Weight,
 Increment,
 Value)

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Variable_nbr { <u>lolim</u> = 1, <u>hilim</u> = varnbr, <u>default</u> (noquery) = varnbr}
1	int	V_option { <u>list</u> = V_control, <u>default</u> (noquery) = "set"}
2	int	Syst_surf_nbr { <u>lolim</u> = 0, <u>hilim</u> = imsnbr}
3	int	Config_nbr { <u>lolim</u> = 0, <u>default</u> (onprompt) = 0}
4	int	Variable_type { <u>list</u> = Var_types}
5	real	Min_bound { <u>default</u> (noquery) = 0.0}
6	real	Max_bound { <u>default</u> (noquery) = 0.0}
7	real	Weight { <u>default</u> (noquery) = 1.0}
8	real	Increment { <u>default</u> (noquery) = 0.0}
9	real	Value

List Definitions:

```
list V_control
  {"{ins}Insert"      = [2,3,4,5,6,7,8]3,
   "{del}Delete"     = [1]1,
   "{sho}Show"       = [2]2,
   "{min}Minimum"    = [5]5,
   "{max}Maximum"    = [6]6,
   "{wgt}Weight"     = [7]7,
   "{inc}Increment"  = [8]8,
   "{val}Value"      = [9]4,
   "{set}Set"        = [2,3,4,5,6,7,8]0}
```

```
list Var_types
  {"{CV}Curvature (CV)"           = 1,
   "{CVD}Curvature pickup constant (CVD)" = 100,
   "{CVM}Curvature pickup multiplier (CVM)" = 123,
   "{CC}Conic constant (CC)"       = 2,
   "{TH}Thickness (TH)"             = 3,
   "{THD}Thickness pickup constant (THD)" = 101,
   "{THM}Thickness pickup multiplier (THM)" = 122,
   "{AP}Aperture radius (AP)"      = 4,
   "{RN}Refractive index (RN)"     = 5,
   "{DN}Dispersion factor (DN)"    = 6,
   "{CVX}Toric curvature (CVX)"    = 7,
```

```

" {CCX} Biconic xz conic const (CCX)" = 8,
" {AD} Fourth-order aspheric coef (AD)" = 9,
" {AE} Sixth-order aspheric coef (AE)" = 10,
" {AF} Eighth-order aspheric coef (AF)" = 11,
" {AG} Tenth-order aspheric coef (AG)" = 12,
" {DCX} X decentration (DCX)" = 13,
" {DCY} Y decentration (DCY)" = 14,
" {DCZ} Z decentration (DCZ)" = 15,
" {TLA} Alpha tilt angle (TLA)" = 16,
" {TLB} Beta tilt angle (TLB)" = 17,
" {TLC} Gamma tilt angle (TLC)" = 18,
" {TOX} Tilt vertex offset in x (TOX)" = 115,
" {TOY} Tilt vertex offset in y (TOY)" = 116,
" {TOZ} Tilt vertex offset in z (TOZ)" = 117,
" {GSP} Grating spacing (GSP)" = 19,
" {AS_} Polynomial aspheric coefficient (AS_)" = 20,
" {DF_} Diffractive surface coefficient (DF_)" = 21,
" {ZR_} Zernike phase surf coefficient (ZR_)" = 22,
" {UDF_} User diffr surf coefficient (UDF_)" = 109,
" {DWV} Design wavelength (DWV)" = 23,
" {ASA_} ISO aspheric A coefficient (ASA_)" = 24,
" {ASB_} ISO aspheric B coefficient (ASB_)" = 25,
" {ASC_} ISO aspheric C coefficient (ASC_)" = 26,
" {ASD_} ISO aspheric D coefficient (ASD_)" = 27,
" {CNS} ISO cone slope (CNS)" = 28,
" {CNX} ISO asymmetric cone x coef (CNX)" = 29,
" {CNY} ISO asymmetric cone y coef (CNY)" = 30,
" {SS_} Radial Spline zone slope (SS_)" = 31,
" {ABCZ_} Asymmetric Spline sag (ABCZ_)" = 121,
" {HX1} HOE pt 1 x coordinate (HX1)" = 32,
" {HY1} HOE pt 1 y coordinate (HY1)" = 33,
" {HZ1} HOE pt 1 z coordinate (HZ1)" = 34,
" {HX2} HOE pt 2 x coordinate (HX2)" = 35,
" {HY2} HOE pt 2 y coordinate (HY2)" = 36,
" {HZ2} HOE pt 2 z coordinate (HZ2)" = 37,
" {HWV} HOE construction wavelength (HWV)" = 38,
" {NR1} Second-order radial GRIN coef (NR1)" = 39,
" {NR2} Fourth-order radial GRIN coef (NR2)" = 40,
" {NR3} Sixth-order radial GRIN coef (NR3)" = 41,
" {NR4} Eighth-order radial GRIN coef (NR4)" = 42,
" {NZ1} First-order axial GRIN coef (NZ1)" = 43,
" {NZ2} Second-order axial GRIN coef (NZ2)" = 44,
" {NZ3} Third-order axial GRIN coef (NZ3)" = 45,
" {NZ4} Fourth-order axial GRIN coef (NZ4)" = 46,
" {VR1} Second-order radial GRIN V-number (VR1)" = 47,
" {VR2} Fourth-order radial GRIN V-number (VR2)" = 48,
" {VR3} Sixth-order radial GRIN V-number (VR3)" = 49,
" {VR4} Eighth-order radial GRIN V-number (VR4)" = 50,
" {VZ1} First-order axial GRIN V-number (VZ1)" = 51,
" {VZ2} Second-order axial GRIN V-number (VZ2)" = 52,
" {VZ3} Third-order axial GRIN V-number (VZ3)" = 53,
" {VZ4} Fourth-order axial GRIN V-number (VZ4)" = 54,
" {PR1} Second-order radial grin partial disp (PR1)" = 55,
" {PR2} Fourth-order radial grin partial disp (PR2)" = 56,
" {PR3} Sixth-order radial grin partial disp (PR3)" = 57,
" {PR4} Eighth-order radial grin partial disp (PR4)" = 58,
" {PZ1} First-order axial GRIN partial disp (PZ1)" = 59,
" {PZ2} Second-order axial GRIN partial disp (PZ2)" = 60,
" {PZ3} Third-order axial GRIN partial disp (PZ3)" = 61,
" {PZ4} Fourth-order axial GRIN partial disp (PZ4)" = 62,
" {GOZ} Gradium blank offset (GOZ)" = 63,
" {GMZ} Gradium blank thickness (GMZ)" = 64,
" {GNZ_} Gradium profile coef (GNZ_)" = 108,
" {NRX} Elliptical GRIN x coef (NRX)" = 74,
" {NRY} Elliptical GRIN y coef (NRY)" = 75,
" {SGC} Spherical GRIN center distance (SGC)" = 76,
" {SVA} Sinusoidal GRIN amplitude (SVA)" = 77,
" {SVP} Sinusoidal GRIN period (SVP)" = 78,
" {SVF} Sinusoidal GRIN phase (SVF)" = 79,

```

```

" {TAS} Tapered GRIN slope (TAS)" = 80,
" {TAO} Tapered GRIN offset (TAO)" = 81,
" {GDX} GRIN axis x decentration (GDX)" = 102,
" {GDY} GRIN axis y decentration (GDY)" = 103,
" {GDZ} GRIN axis z decentration (GDZ)" = 104,
" {GTA} GRIN axis alpha tilt angle (GTA)" = 105,
" {GTB} GRIN axis beta tilt angle (GTB)" = 106,
" {GTC} GRIN axis gamma tilt angle (GTC)" = 107,
" {UG1} User GRIN coef 1 (UG1)" = 65,
" {UG2} User GRIN coef 2 (UG2)" = 66,
" {UG3} User GRIN coef 3 (UG3)" = 67,
" {UG4} User GRIN coef 4 (UG4)" = 68,
" {UG5} User GRIN coef 5 (UG5)" = 69,
" {UG6} User GRIN coef 6 (UG6)" = 70,
" {UG7} User GRIN coef 7 (UG7)" = 71,
" {UG8} User GRIN coef 8 (UG8)" = 72,
" {UGR_} User GRIN coefficient (UGR_)" = 73,
" {UT_} User surface coefficient (UT_)" = 82,
" {EI_} Eikonal surface coefficient (EI_)" = 83,
" {PFP} Perfect lens power (PFP)" = 84,
" {PFM} Perfect lens magnification (PFM)" = 85,
" {FCV} Fresnel surface base curvature (FCV)" = 86,
" {FCC} Fresnel surface base conic cnst (FCC)" = 87,
" {JAA} Jones matrix A amplitude (JAA)" = 88,
" {JPA} Jones matrix A phase (JPA)" = 89,
" {JAB} Jones matrix B amplitude (JAB)" = 90,
" {JPB} Jones matrix B phase (JPB)" = 91,
" {JAC} Jones matrix C amplitude (JAC)" = 92,
" {JPC} Jones matrix C phase (JPC)" = 93,
" {JAD} Jones matrix D amplitude (JAD)" = 94,
" {JPD} Jones matrix D phase (JPD)" = 95,
" {LTH_} Coating layer thickness (LTH_)" = 97,
" {LIN_} Coating layer refractive index (LIN_)" = 98,
" {LEC_} Coating layer extinction coef (LEC_)" = 99,
" {CAK} Crystal axis x direction number (CAK)" = 118,
" {CAL} Crystal axis y direction number (CAL)" = 119,
" {CAM} Crystal axis z direction number (CAM)" = 120,
" {ITS} Interferometric deformation scale factor (ITS)" = 110,
" {ITR} Interferometric deformation radius (ITR)" = 111,
" {ITA} Interferometric deformation rotation angle (ITA)" = 112,
" {ITX} Interferometric deformation x center (ITX)" = 113,
" {ITY} Interferometric deformation y center (ITY)" = 114}

```

Optimization Variables

"[vse](#)" command ("[var](#)" for output)

"Variables" button from the "Surface Data" spreadsheet

Type	Description	OSLO Premium only
cv	curvature	
cc	conic constant	
th	thickness	
ap	aperture radius	
rn	refractive index (model glass only)	

dn	dispersion factor (model glass only)	
cvx	toric curvature	
ccx	xz azimuth conic constant	•
fcv	Fresnel surface base curvature	
fcc	Fresnel surface base conic constant	
ad	4th order aspheric coefficient	
ae	6th order aspheric coefficient	
af	8th order aspheric coefficient	
ag	10th order aspheric coefficient	
dcx	x-decentration	
dcy	y-decentration	
dcz	z-decentration	
tla	tilt around x-axis	
tlb	tilt around y-axis	
tlc	tilt around z-axis	
gsp	grating spacing	
pfp	perfect lens power (1/focal length)	
pfm	perfect lens magnification	
asi	aspheric surface coefficient i	
asai	ISO aspheric surface coefficient i	
asbi	ISO aspheric surface coefficient i	•
asci	ISO aspheric surface coefficient i	•
asdi	ISO aspheric surface coefficient i	•
cns	ISO cone surface coefficient	
cnx	ISO asymmetric cone x coefficient	•
cny	ISO asymmetric cone y coefficient	•
ssi	spline surface slope at radial zone i	
dfi	diffractive surface coefficient i	
dvw	diffractive surface design wavelength	
zri	Zernike phase surface coefficient i	
hx1	hologram source point 1 x-coordinate	
hy1	hologram source point 1 y-coordinate	
hz1	hologram source point 1 z-coordinate	
hx2	hologram source point 2 x-coordinate	
hy2	hologram source point 2 y-coordinate	
hz2	hologram source point 2 z-coordinate	

hwv	hologram construction wavelength	
nr1	GRIN medium r2 or r ² coefficient	
nr2	GRIN medium r4 or r ⁴ coefficient	
nr3	GRIN medium r6 or r ⁶ coefficient	
nr4	GRIN medium r8 or r ⁸ coefficient	
nz1	GRIN medium z coefficient	
nz2	GRIN medium z ² coefficient	
nz3	GRIN medium z ³ coefficient	
nz4	GRIN medium z ⁴ coefficient	
vr1	GRIN r2 or r ² v-number	
vr2	GRIN r4 or r ⁴ v-number	
vr3	GRIN r6 or r ⁶ v-number	
vr4	GRIN r8 or r ⁸ v-number	
vz1	GRIN z v-number	
vz2	GRIN z ² v-number	
vz3	GRIN z ³ v-number	
vz4	GRIN z ⁴ v-number	
pr1	GRIN r2 or r ² partial dispersion	
pr2	GRIN r4 or r ⁴ partial dispersion	
pr3	GRIN r6 or r ⁶ partial dispersion	
pr4	GRIN r8 or r ⁸ partial dispersion	
pz1	GRIN z partial dispersion	
pz2	GRIN z ² partial dispersion	
pz3	GRIN z ³ partial dispersion	
pz4	GRIN z ⁴ partial dispersion	
goz	GRADIUM axial offset into blank	
nrx	elliptical GRIN x-coefficient	•
nry	elliptical GRIN y-coefficient	•
sgc	spherical GRIN center of symmetry	•
sva	sinusoidal GRIN amplitude	•
svp	sinusoidal GRIN period	•
svf	sinusoidal GRIN phase	•
tas	tapered GRIN slope	•
tao	tapered GRIN offset	•
ugri	user GRIN coefficient i	•
uti	user raytrace surface coefficient i	•

eii	eikonal surface coefficient i	•
jaa	amplitude of polarization element JA	•
jpa	phase of polarization element JA	•
jab	amplitude of polarization element JB	•
jpb	phase of polarization element JB	•
jac	amplitude of polarization element JC	•
jpc	phase of polarization element JC	•
jad	amplitude of polarization element JD	•
jpd	phase of polarization element JD	•

operands (ope) command

Lists the [operands set](#) or enters the [command-line operands editor](#).

Command Syntax:

```
cmd operands
  (Ope_option,
   First_item_nbr,
   Last_item_nbr)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Ope_option { list = Ope_options, default (noquery) = "prt"}
1	int	First_item_nbr { lolim = 0, default (noquery) = 0}
2	int	Last_item_nbr { lolim = 0, default (noquery) = 0}

List Definitions:

```
list Ope_options
  {"New"
   "{upd}Update"
   "All"
   "{lis}List"
   "{prt}Print"
   "{erf}Error Function" = 5}
   = [1],
   = [10],
   = 3,
   = 4,
   = -1,
```

o (op) command

Adds, deletes, displays, or modifies entry "Operand_nbr" of the [operands set](#) according to the value of the "O_option" argument:

- "Insert" inserts a new operand
- "Delete" deletes the specified operand
- "Show" displays the operand
- "Edit" allows the operand definition to be edited on the command line
- "Mode" sets the mode (minimize or constraint)
- "Weight" sets the weight (minimize-mode)
- "Target" sets the target (constraint-mode)
- "Name" sets the operand name
- "Set" changes the operand definition

OSLO Premium/OSLO Standard

The arguments "Con mode" and "Target", and the options "Mode" and "Target" are available in OSLO Premium and OSLO Standard.

Command Syntax:

```
cmd o  
  (Operand_nbr,  
   O_option,  
   Operand_defn,  
   Con_mode,  
   Weight,  
   Target,  
   Operand_name)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Operand_nbr { lolim = 1, hilim = oprnbr, default (noquery) = oprnbr}
1	int	O_option { list = O_control, default (noquery) = "set"}
2	char	Operand_defn [] { hilim = 46}
3	int	Con_mode { list = Con_min, default (noquery) = "min"}
4	real	Weight { default (noquery) = 1.0}
5	real	Target { default (noquery) = 0.0}
6	char	Operand_name [] { hilim = 10, default (noquery) = ""}

List Definitions:

```
list O_control
```

```

    {"ins}Insert" = 3,
    {"del}Delete" = [1],
    {"sho}Show" = [2],
    {"edt}Edit" = [9],
    {"mod}Mode" = [3]11,
    {"wgt}Weight" = [4]7,
    {"tar}Target" = [5]31,
    {"nam}Name" = [6]10,
    {"set}Set" = 0}
list Con_min
    {"@cor" = [5,6]1,
    {"con}Constraint" = [5,6]1,
    {"min}Minimize" = [4,6]0}

```

opt_oprdccl (opoc) command

generic operands: opoc("opcb_abs")

Sets the name of the command (SCP, CCL, or DLL) that will be executed to compute the values of [user-defined operands](#).

See [optimization conditions](#).

Command Syntax:

```

cmd opt_oprdccl
    (Operating_condition_value,
    Optim_operand_ccl_command)

```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Operating_condition_value { list = Op_con_val, default (noquery) = "set"}
1	char	Optim_operand_ccl_command [] { lolim = 0, hilim = 23, default (onprompt) = -310}

List Definitions:

```

list Op_con_val
    {"set}Set to new value" = [1]0,
    {"def}Use default" = [1]1}

```

iterate (ite) command

Performs Nbr_iters iterations of [Damped Least Squares optimization](#). If "Recalculate_derivate_increments" is "yes", the derivative increments for [variables](#) with currently non-negative increments will be recalculated before beginning the iteration. If "Reset_variable_damping_factors" is "yes", the variable damping will be reset to 1.0 for variables with currently non-negative damping factors.

Click [here](#) for additional help.

Command Syntax:

```
cmd iterate
  (Ite_option,
   Nbr_iters,
   Printing_option,
   Recalculate_derivative_increments,
   Reset_variable_damping_factors)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Ite_option { list = Ite_options, default (noquery) = "ful"}
1	int	Nbr_iters { lolim = 1, default (noquery) = 10}
2	int	Printing_option { list = Printing_options, default (noquery) = "std"}
3	int	Recalculate_derivative_increments { list = Yes_no, default (noquery) = "y"}
4	int	Reset_variable_damping_factors { list = Yes_no, default (noquery) = "y"}

List Definitions:

```
list Ite_options
  {"ful"}Full" = [1,2,3,4]3,
  {"std"}Standard" = [1,3,4]4,
  {"con"}Constraints" = [1,3,4]1}
list Printing_options
  {"min"}Minimal" = 0,
  {"std"}Standard" = 1,
  {"ful"}Full" = 2}
list Yes_no
  {"y"}Yes" = 1,
  {"n"}No" = 0}
list Yes_no
  {"y"}Yes" = 1,
  {"n"}No" = 0}
```

paraxial_trace (pxt) command

Displays the ray height, refracted angle, and angle of incidence for the paraxial axial and chief rays. The default is to trace the rays in the y-z plane. If the rays are traced in the x-z plane, the aperture and field may be scaled from the values defined by the entrance beam radius/object numerical aperture and field angle/object height.

Click [here](#) for additional help.

Command Syntax:

```
cmd paraxial_trace
  (Parax_trace_surf_range,
   First_surf_nbr,
   Last_surf_nbr,
   Paraxial_meridian,
```

```
Fractional_xz_aperture,  
Fractional_xz_field,  
Evaluation_wavelength)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Parax_trace_surf_range { list = Parax_srf_option, default (noquery) = "srf"}
1	int	First_surf_nbr { lolim = 0, hilim = imsnbr, default (noquery) = 0}
2	int	Last_surf_nbr { lolim = 0, hilim = imsnbr, default (noquery) = 0}
3	int	Paraxial_meridian { list = Parax_merid, default (noquery) = "y"}
4	real	Fractional_xz_aperture { lolim = 0.0, default (noquery) = 1.0}
5	real	Fractional_xz_field { lolim = 0.0, default (noquery) = 1.0}
6	int	Evaluation_wavelength { lolim = 0, hilim = numw, default (noquery) = curwav}

List Definitions:

```
list Parax_srf_option  
  {"Srf" = [1,2,3,4,5,6]0,  
   "All" = [3,4,5,6]1}  
list Parax_merid  
  {"{y}YZ plane" = [6]0,  
   "{x}XZ plane" = [4,5,6]1}
```

paraxial_constants (pxc) command

Displays the paraxial constants for the system. The values may be computed in either the y-z (default) or x-z meridian. If the x-z meridian is chosen, the aperture and field may be scaled from the values defined by the entrance beam radius/object numerical aperture and field angle/object height. For focal systems, the quantities displayed are the effective focal length, working f-number, numerical aperture, paraxial (Lagrange) invariant, transverse magnification, Gaussian image height, and Petzval radius. For afocal systems, the quantities displayed are the angular magnification, eye relief, paraxial invariant, and Petzval radius.

Click [here](#) for additional help.

Command Syntax:

```
cmd paraxial_constants  
  (Paraxial_constants_meridian,  
   Fractional_xz_aperture,  
   Fractional_xz_field,  
   Evaluation_wavelength)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Paraxial_constants_meridian { list = Parax_cons_merid, default (noquery) = "y"}
1	real	Fractional_xz_aperture { lolim = 0.0, default (noquery) = 1.0}
2	real	Fractional_xz_field { lolim = 0.0, default (noquery) = 1.0}
3	int	Evaluation_wavelength { lolim = 0, hilim = numw, default (noquery) = curwav}

List Definitions:

```
list Parax_cons_merid
    {"y}YZ plane" = [3]0,
    {"x}XZ plane" = [1,2,3]1}
```

chromatic_abers (chr) command

Displays the paraxial chromatic aberrations: primary and secondary axial and lateral chromatic aberration. The primary chromatic aberrations are computed using wavelengths 2 and 3, while the secondary chromatic aberrations are computed using wavelengths 2 and 1. The meaning of the coefficients is determined by the aberration mode (amo) general condition.

Click [here](#) for additional help.

Command Syntax:

```
cmd chromatic_abers
    (Surf_range,
     First_surf_nbr,
     Last_surf_nbr)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Surf_range { list = Srf_option, default (noquery) = "srf"}
1	int	First_surf_nbr { lolim = 0, hilim = imsnbr, default (noquery) = 0}
2	int	Last_surf_nbr { lolim = 0, hilim = imsnbr, default (noquery) = 0}

List Definitions:

```
list Srf_option
    {"Srf" = [1,2]0,
    {"All" = [1]}
```

seidel_abers (sei) command

Displays the third-order (Seidel) image aberrations: spherical aberration, coma, astigmatism, Petzval curvature, and distortion. The meaning of the coefficients is determined by the aberration mode (amo) general operating condition.

Click [here](#) for additional help.

Command Syntax:

```
cmd seidel_abers
  (Aber_surf_range,
   First_surf_nbr,
   Last_surf_nbr,
   Evaluation_wavelength)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Aber_surf_range { list = Aber_srf_option, default (noquery) = "srf"}
1	int	First_surf_nbr { lolim = 0, hilim = imsnbr, default (noquery) = 0}
2	int	Last_surf_nbr { lolim = 0, hilim = imsnbr, default (noquery) = 0}
3	int	Evaluation_wavelength { lolim = 0, hilim = numw, default (noquery) = curwav}

List Definitions:

```
list Aber_srf_option
  {"Srf" = [1,2,3]0,
   "All" = [3]1}
```

fifth_order_abers (fif) command

Displays the fifth-order image aberrations in Seidel format, i.e., fifth-order spherical aberration, coma, astigmatism, Petzval curvature, and distortion. Also displayed is seventh-order spherical aberration. The meaning of the coefficients is determined by the aberration mode (amo) general operating condition.

Click [here](#) for additional help.

Command Syntax:

```
cmd fifth_order_abers
  (Aber_surf_range,
   First_surf_nbr,
   Last_surf_nbr,
   Evaluation_wavelength)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Aber_surf_range { list = Aber_srf_option, default (noquery) = "srf"}
1	int	First_surf_nbr { lolim = 0, hilim = imsnbr, default (noquery) = 0}
2	int	Last_surf_nbr { lolim = 0, hilim = imsnbr, default (noquery) = 0}
3	int	Evaluation_wavelength { lolim = 0, hilim = numw, default (noquery) = curwav}

List Definitions:

```
list Aber_srf_option
  {"Srf" = [1,2,3]0,
   "All" = [3]1}
```

plot_spot_diagram (pls) command

Plots the [spot diagram](#) for the [current object point](#). The ray intersections may be plotted as "spots" or as symbols, with a different symbol for each [wavelength](#). The geometric RMS spot sizes and the diffraction limit are displayed in the plot.

Command Syntax:

```
cmd plot_spot_diagram
  (Pls_option,
   Spot_or_sym,
   Focus,
   Spot_diagram_scale)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Pls_option { list = Pls_options, default (noquery) = "cen"}
1	int	Spot_or_sym { list = Pls_symbols, default (noquery) = "spt"}
2	real	Focus { default (noquery) = 0.0}
3	real	Spot_diagram_scale { lolim = 0.0, default (noquery) = 0.0}

List Definitions:

```
list Pls_options
  {"{cen}Centroid"      = 0,
   "{ref}Reference ray" = 1,
   "{xy}Absolute x and y" = 2}
list Pls_symbols
  {"{spt}Spot"      = 0,
   "{sym}Symbols"  = 1}
```

spd_show_airy (sdsa) command

Designates that the equivalent Airy disk is to be drawn on [spot diagram displays](#).

Command Syntax:

```
cmd spd_show_airy
  (Operating_condition_value,
   Show_airy_disk)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Operating_condition_value { list = Op_con_val, default (noquery) = "set"}
1	char	Show_airy_disk { list = On_off, default (onprompt) = off}

List Definitions:

```
list Op_con_val
  {"{set}Set to new value" = [1]0,
   "{def}Use default"      = [1]1}
list On_off
  {"On" = 1,
   "Off" = 0}
```

ssbuf_reset (sbr) command

Resets the [Spreadsheet Buffer](#) for the active text window. Reset_index specifies the row of the current Spreadsheet Buffer that is to become row number 1 after the reset (if 0 or -1 is given for this argument, it is equivalent to 1). Rows_to_clear indicates the number of rows (starting with row Reset_index) that are to be filled with the value Value_for_clear (which defaults to 0.0).

See [Spreadsheet Buffer](#) for additional help.

Command Syntax:

```
cmd ssbuf_reset
  (Reset_index,
   Rows_to_clear,
   Value_for_clear)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Reset_index { lolim = -1999, hilim = 1999, default (noquery) = 0}
1	int	Rows_to_clear { lolim = 0, hilim = 1999, default (noquery) = 1999}

2	real	Value_for_clear { default (noquery) = 0.0}
---	-------------	---

ssb function

Returns the value from row Row_index, column Col_index of the [Spreadsheet Buffer](#) of the active text window.

See [Spreadsheet Buffer](#) for additional help.

Command Syntax:

```
real ssb
  (Row_index,
   Col_index)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Row_index { lolim = 1, hilim = 1999, default (noquery) = 1}
1	int	Col_index { lolim = 1, hilim = 10, default (noquery) = 1}

sbrow function

Returns the row number of the next row to receive numeric output in the [Spreadsheet Buffer](#) of the active text window. Reset by [Ssbuff_reset](#) command and by [Textwin_reset](#) command.

See [Spreadsheet Buffer](#) for additional help.

Command Syntax:

```
int sbrow
  (void)
```

graphwin_open (gwo) command

Opens the graphics window indicated by Graphics_window_number (default = 0, indicating the next available graphics window). If the specified window is already open, it is brought to the foreground. Window_width and Window_height are the initial width and height of the area the window occupies, in pixels. Window_X and Window_Y are the coordinates of the position of the upper left corner of the window, in pixels.

See [Graphic Window Management Functions](#) for additional help.

Command Syntax:

```
cmd graphwin_open
  (Graphics_window_number,
   Window_width,
   Window_height,
   Window_x,
   Window_y)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Graphics_window_number { lolim = 0, hilim = 32, default (noquery) = 0}
1	int	Window_width { lolim = 0, default (noquery) = 0}
2	int	Window_height { lolim = 0, default (noquery) = 0}
3	int	Window_x { lolim = 0, default (noquery) = 0}
4	int	Window_y { lolim = 0, default (noquery) = 0}

graphwin_close (gwc) command

Closes the graphics window indicated by Graphics_window_to_close (default = 0, indicating the active graphics window). Note that if only one graphics window is open, it cannot be closed.

See [Graphic Window Management Functions](#) for additional help.

Command Syntax:

```
cmd graphwin_close
  (Graphics_window_to_close)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Graphics_window_to_close { lolim = 0, hilim = 32, default (noquery) = 0}

graphwin_reset (gwr) command

Clears the active graphics window.

See [Graphic Window Management Functions](#) for additional help.

Command Syntax:

```
cmd graphwin_reset
  (Standard_aspect_ratio,
   Preserve_callback_command)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	real	Standard_aspect_ratio { lolim = 0.0, default (noquery) = 0.0}
1	int	Preserve_callback_command { list = Yes_no, default (noquery) = "n"}

List Definitions:

```
list Yes_no
  {"{y}Yes" = 1,
   "{n}No" = 0}
```

graphwin_print (gwp) command

This command prints the [current graphics window](#). If Printer_or_file is "Printer", output is sent to the printer; otherwise, it is sent to a graphics file. If Show_print_dialog is "Yes", the standard Print dialog box (for printer output) or File dialog box (for file output) is displayed. If Show_print_dialog is "No", the window is printed to the printer or to the graphics file whose name is given by Plot_file_name.

Command Syntax:

```
cmd graphwin_print
  (Printer_or_file,
   Show_print_dialog,
   Plot_file_name)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Printer_or_file { list = Printer_or_file_list, default (noquery) = "printer"}
1	int	Show_print_dialog { list = Print_dialog_options, default (noquery) = "yes"}
2	char	Plot_file_name [] { lolim = 0, hilim = 260, default (onprompt) = pm(*)}

List Definitions:

```
list Printer_or_file_list
  {"Printer" = [1]0,
   "File" = 1}
list Print_dialog_options
  {"Yes" = [1]1,
   "No" = 0}
```

graphwin_update (gwu) command

Invokes the command string assigned to graphics window Graphics_window_number by the Graphwin_execute command.

See [Graphic Window Management Functions](#) for additional help.

Command Syntax:

```
cmd graphwin_update
  (Graphics_window_number)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Graphics_window_number { lolim = 0, hilim = 32, default (noquery) = 0}

graphwin_reset (gwr) command

Clears the active graphics window.

See [Graphic Window Management Functions](#) for additional help.

Command Syntax:

```
cmd graphwin_reset
  (Standard_aspect_ratio,
  Preserve_callback_command)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	real	Standard_aspect_ratio { lolim = 0.0, default (noquery) = 0.0}
1	int	Preserve_callback_command { list = Yes_no, default (noquery) = "n"}

List Definitions:

```
list Yes_no
  {"{y}Yes" = 1,
  "{n}No" = 0}
```

moveto (gmv) command

Moves the current pen position in the active graphics window to the position (X_Coordinate,Y_Coordinate).

See [Moveto/Moverel](#) for additional help.

Command Syntax:

```
cmd moveto
  (X_coordinate,
   Y_coordinate)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	real	X_coordinate
1	real	Y_coordinate

lineto (glt) command

Draws a line in the active graphics window from the current pen position to the position (X_Coordinate,Y_Coordinate) using the current pen.

See [Lineto/Linerel](#) for additional help.

Command Syntax:

```
cmd lineto
  (X_coordinate,
   Y_coordinate)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	real	X_coordinate
1	real	Y_coordinate

polylineto (gpl) command

Draws a series of lines in the current pen color connecting the points specified in the X_Coordinates and Y_Coordinates arrays. The position of each point is specified relative to the origin of the viewport. Number_of_points is the number of points in the arrays (note that the arrays begin with element 0, so the array indices go from 0 to Number_of_points - 1). Fill specifies the color used to fill a polygon: 0 (the default) means no fill, while nonzero specifies the pen number to be used to fill the polygon. Note that if a polygon is to be filled, it must be closed; that is, point (Number_of_points - 1) should equal point 0.

See [Polylineto/Polylinerel](#) for additional help.

Command Syntax:

```
cmd polylineto
  (X_coordinates,
   Y_coordinates,
   Number_of_points,
   Fill)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	real	X_coordinates []
1	real	Y_coordinates []
2	int	Number_of_points { lolim = 2}
3	int	Fill { lolim = 0, default (noquery) = 0}

symbol (gsy) command

Draws the symbol specified by Symbol_number at the current pen position in the active graphics window. Symbol_size specifies the size in units of 10 stroke widths (default = 1, corresponding to 10 stroke units).

See [Symbol](#) for additional help.

Command Syntax:

```
cmd symbol  
  (Symbol_number,  
   Symbol_size)
```

Argument Definitions:

Arg #	Type	Name / Definition
0	int	Symbol_number { lolim = 0, hilim = 31}
1	real	Symbol_size { lolim = 0.0, hilim = 16.0, default (noquery) = 0.0}

polypoint (gpp) command

Draws an array of points using the current pen color in the active graphics window at the coordinates specified by the X_coordinates and Y_coordinates arrays. The position of each point is specified relative to the origin of the viewport. Point_count is the number of points in the arrays (note that the arrays begin with element 0, so the array indices go from 0 to Point_count - 1).

See [Polypoint](#) for additional help.

Command Syntax:

```
cmd polypoint  
  (X_coordinates,  
   Y_coordinates,
```

Point_count)

Argument Definitions:

Arg #	Type	Name / Definition
0	real	X_coordinates []
1	real	Y_coordinates []
2	int	Point_count { lolim = 1}

label (lab) command

Draws a string in the active graphics window. This command takes a [Printf\(\)](#)-style argument list: a [format string](#), plus any number of arguments, which are formatted into a string that is drawn at the current pen position, using the current pen color and label size.

See [Label](#) for additional help.

Command Syntax:

```
cmd label  
  (vlist)
```