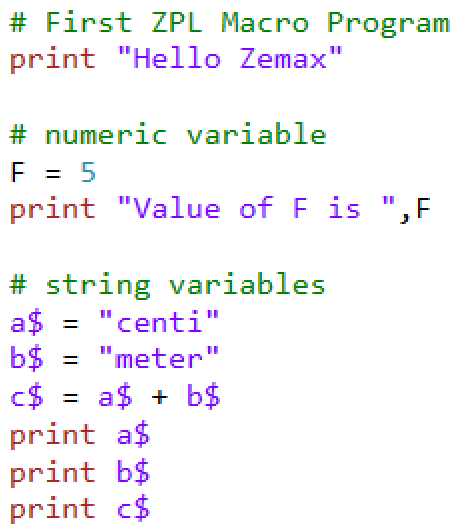
Lectures Notes on Optical Design using Zemax OpticStudio

Lecture 27

Zemax Programming Language



```
# First ZPL Macro Program
print "Hello Zemax"

# numeric variable
F = 5
print "Value of F is ",F

# string variables
a$ = "centi"
b$ = "meter"
c$ = a$ + b$
print a$
print b$
print c$
```

Ahmet Bingül

Gaziantep University
Department of Optical Engineering

May 2025

Content

- Introduction
- Example Simulations using Macro Files

Introduction

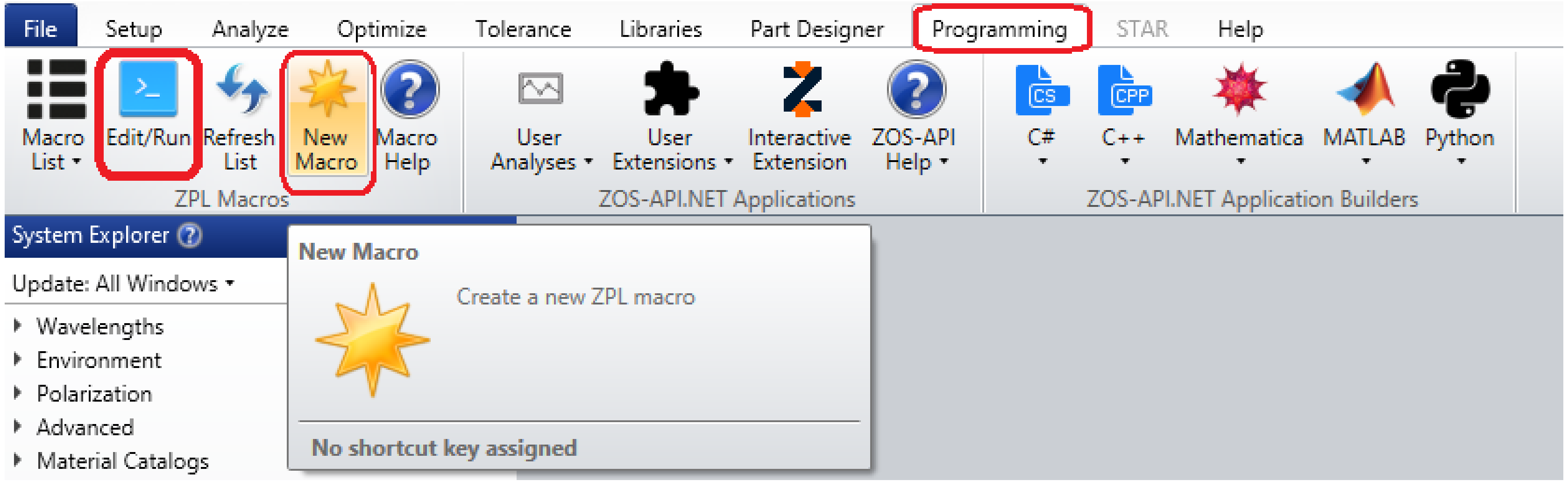
- Zemax is a powerful tool for optical design, but sometimes designers need extra features for special tasks. To help with this, Zemax includes the Zemax Programming Language (ZPL).

- ZPL is a simple programming language, similar to BASIC, that lets users create their own custom functions.

- ZPL file is also known as **Macro** file.

- A detailied guide for ZPL can be in Appendix 4 at the course web page.

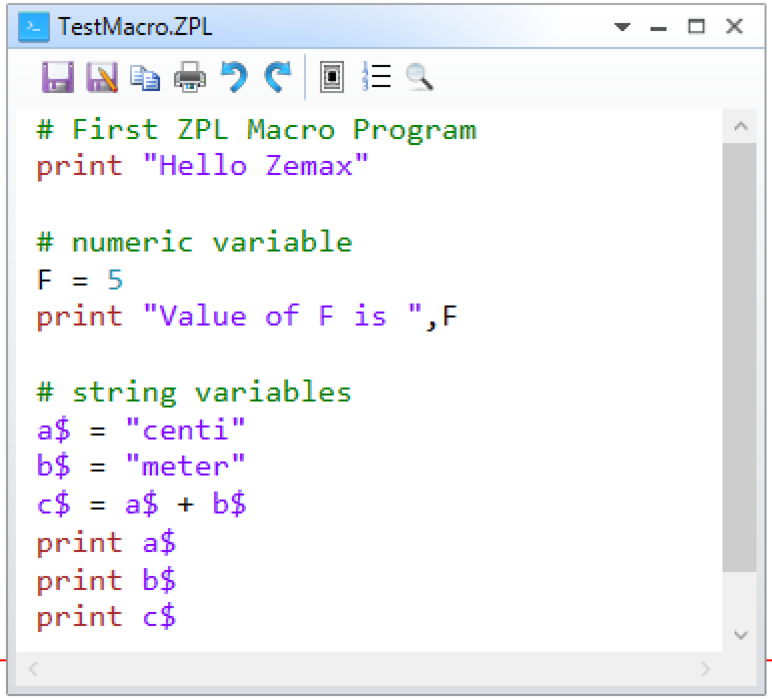
Example 1: How to Edit / Run Macro Files



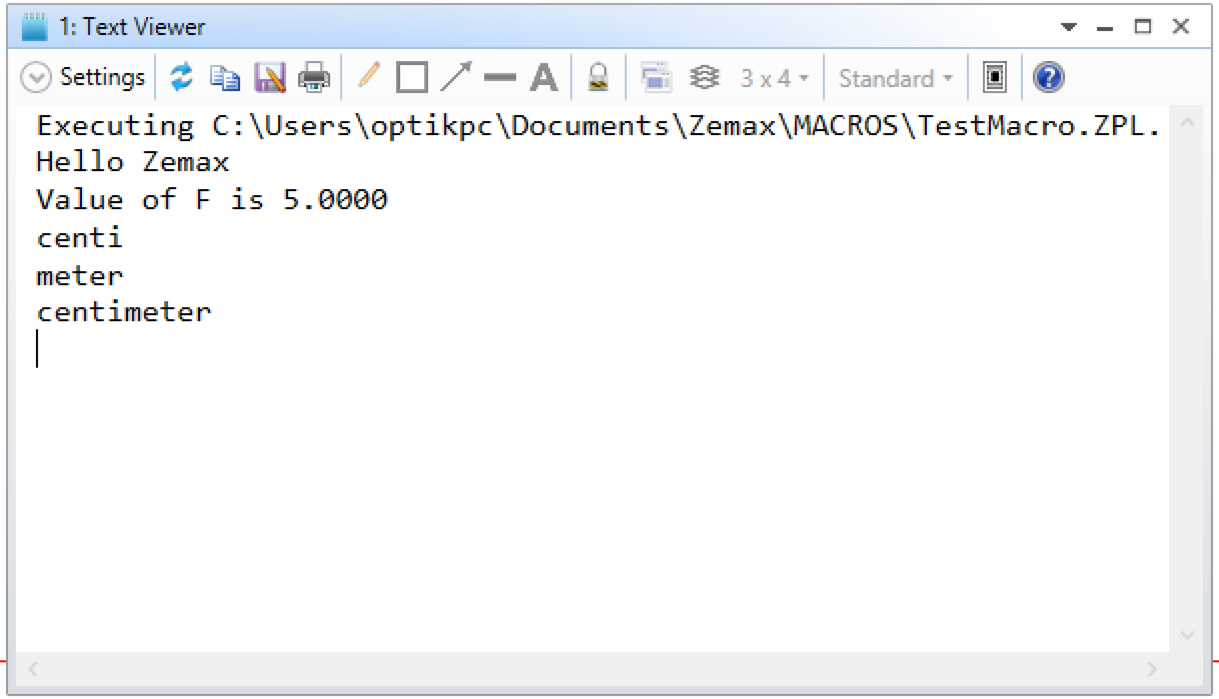
New Macro

Create a new ZPL macro

No shortcut key assigned



```
# First ZPL Macro Program
print "Hello Zemax"

# numeric variable
F = 5
print "Value of F is ",F

# string variables
a$ = "centi"
b$ = "meter"
c$ = a$ + b$
print a$
print b$
print c$
```

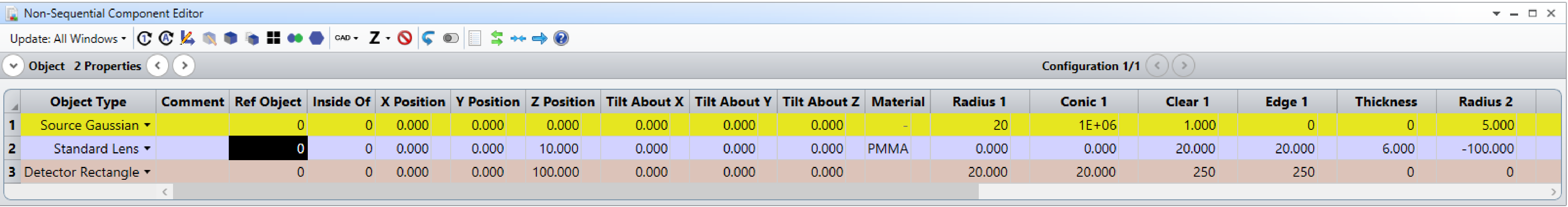


```
Executing C:\Users\optikpc\Documents\Zemax\MACROS\TestMacro.ZPL.
Hello Zemax
Value of F is 5.0000
centi
meter
centimeter
```

Example 2: Ray Tracing and Getting Detector Data

Consider the following system and a merit function:

Source Gaussian	**Standart Lens**	**Detector Rectangle**
Beam Size = 5 mm	Material = PMMA	X-HW = Y-HW = 20
Position = 10 mm	Thickness = 6 mm	X-pix = Y-pix = 250
	Radius2 = -100 mm	
	Edge1=Edge2=Clear1=Clear2=20 mm	



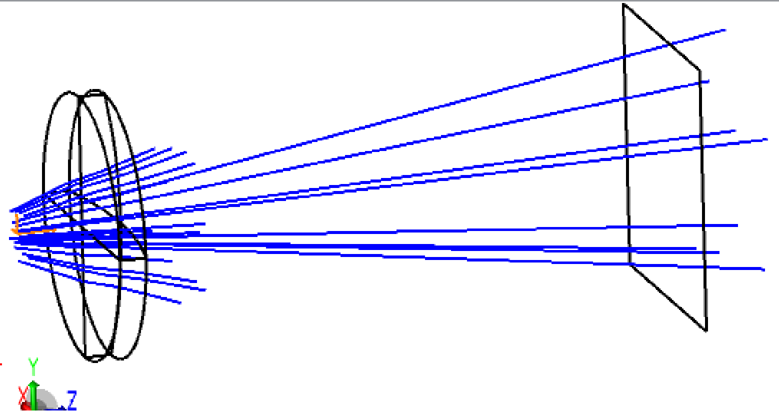
Non-Sequential Component Editor

Update: All Windows

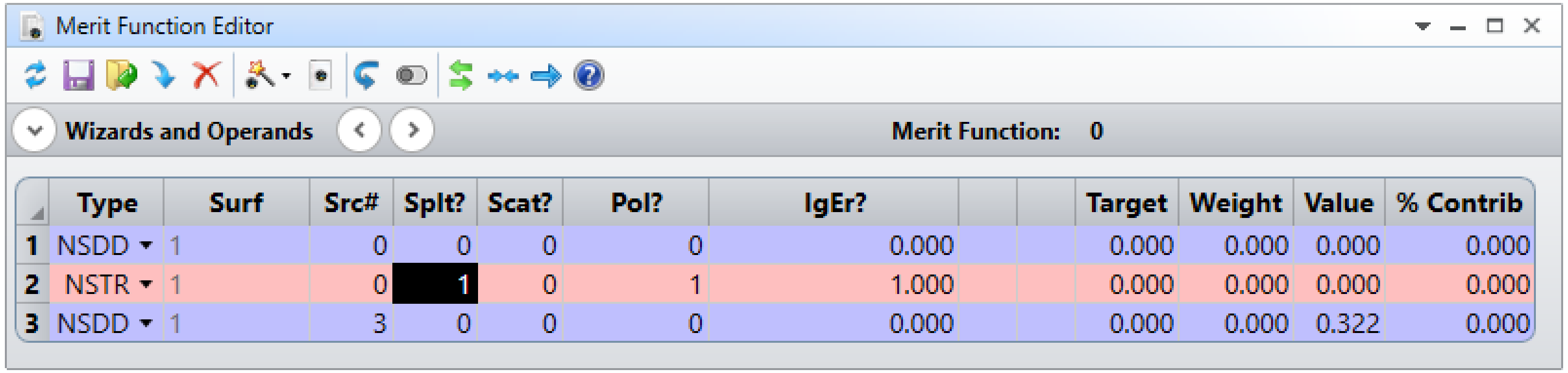
Object 2 Properties

Configuration 1/1

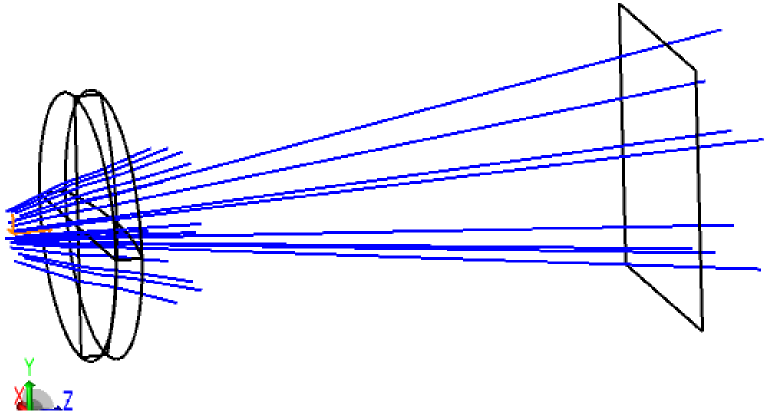
	Object Type	Comment	Ref Object	Inside Of	X Position	Y Position	Z Position	Tilt About X	Tilt About Y	Tilt About Z	Material	Radius 1	Conic 1	Clear 1	Edge 1	Thickness	Radius 2
1	Source Gaussian		0	0	0.000	0.000	0.000	0.000	0.000	0.000	-	20	1E+06	1.000	0	0	5.000
2	Standard Lens		0	0	0.000	0.000	10.000	0.000	0.000	0.000	PMMA	0.000	0.000	20.000	20.000	6.000	-100.000
3	Detector Rectangle		0	0	0.000	0.000	100.000	0.000	0.000	0.000		20.000	20.000	250	250	0	0



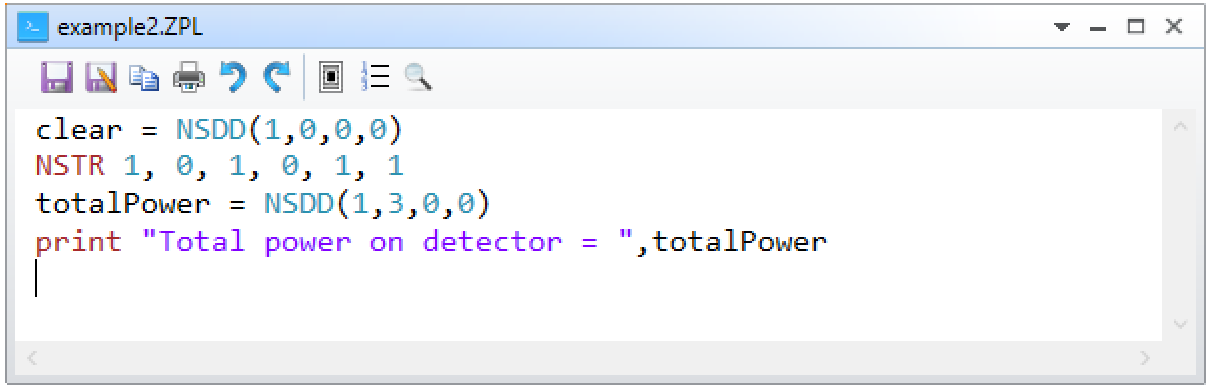
Merit function to read total power on detector:



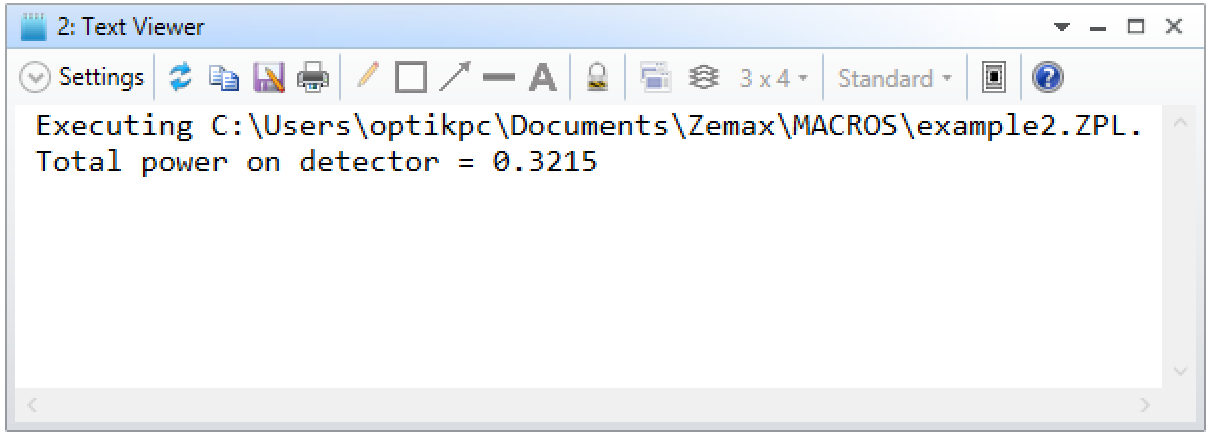
	Type	Surf	Src#	Splt?	Scat?	Pol?	IgEr?			Target	Weight	Value	% Contrib
1	NSDD	1	0	0	0	0	0.000			0.000	0.000	0.000	0.000
2	NSTR	1	0	1	0	1	1.000			0.000	0.000	0.000	0.000
3	NSDD	1	3	0	0	0	0.000			0.000	0.000	0.322	0.000



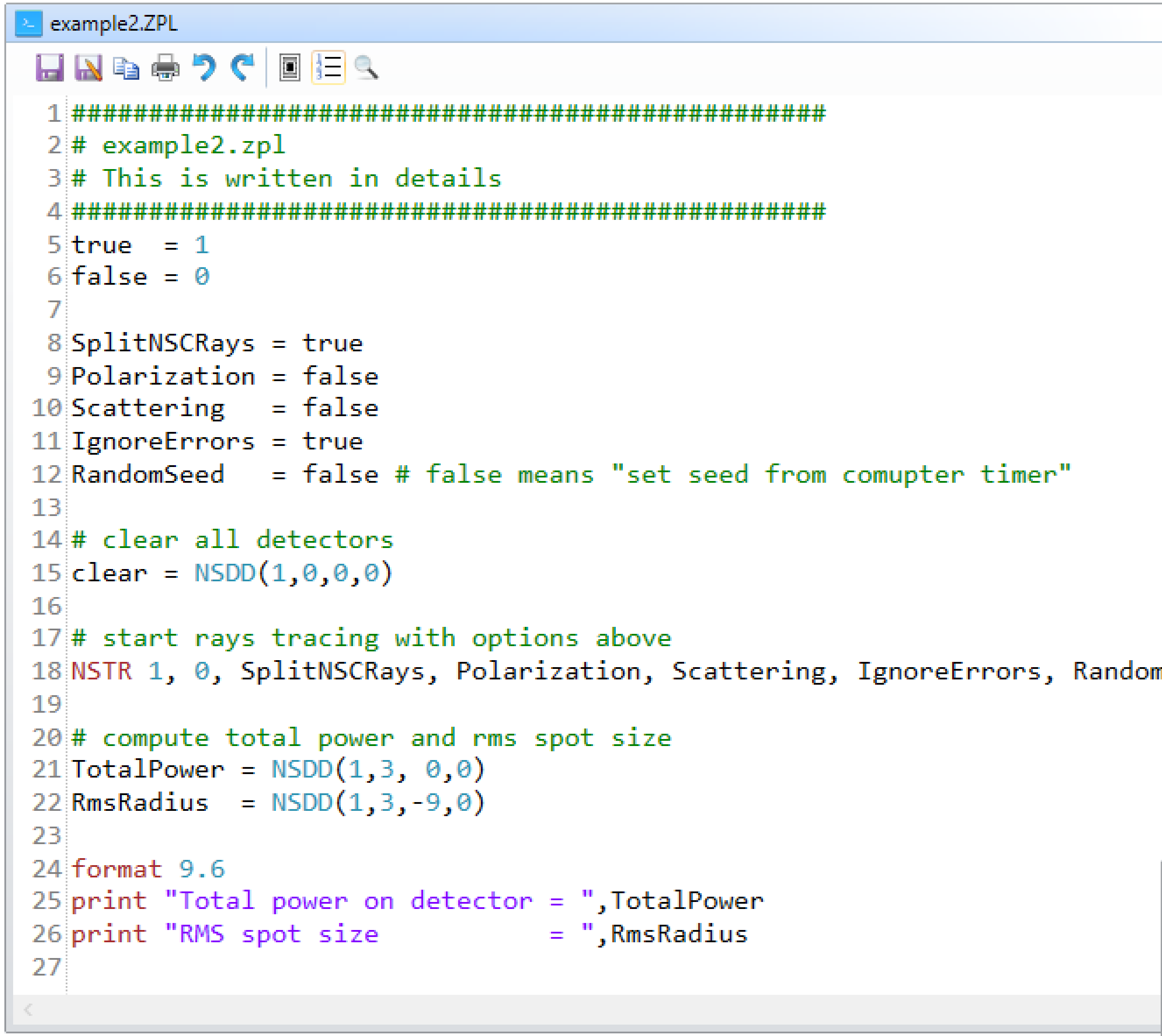
The same operation can be done using ZPL macro as follows:



```
clear = NSDD(1,0,0,0)
NSTR 1, 0, 1, 0, 1, 1
totalPower = NSDD(1,3,0,0)
print "Total power on detector = ",totalPower
```

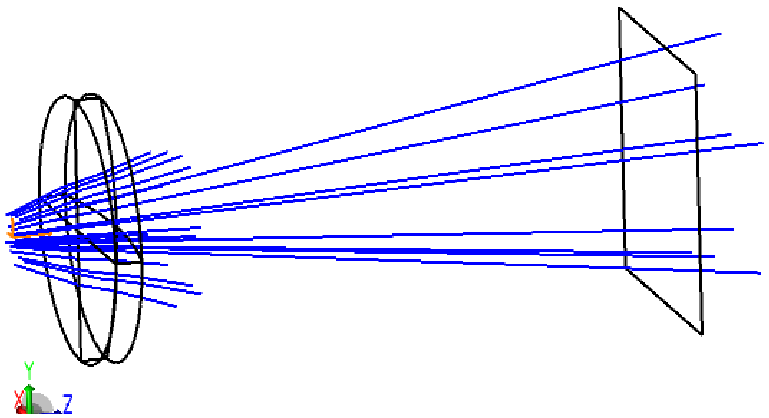


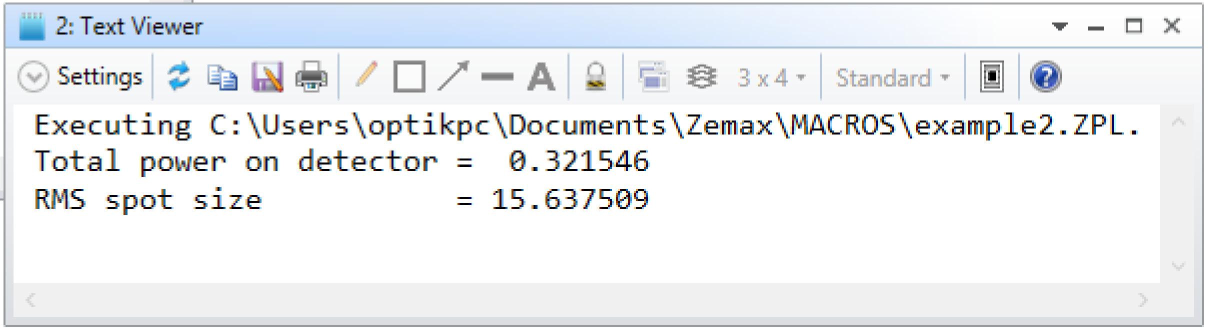
```
Executing C:\Users\optikpc\Documents\Zemax\MACROS\example2.ZPL.
Total power on detector = 0.3215
```



example2.ZPL

```
##################################################
# example2.zpl
# This is written in details
##################################################
true  = 1
false = 0

SplitNSCRays = true
Polarization = false
Scattering   = false
IgnoreErrors = true
RandomSeed   = false # false means "set seed from comupter timer"

# clear all detectors
clear = NSDD(1,0,0,0)

# start rays tracing with options above
NSTR 1, 0, SplitNSCRays, Polarization, Scattering, IgnoreErrors, RandomSeed

# compute total power and rms spot size
TotalPower = NSDD(1,3, 0,0)
RmsRadius  = NSDD(1,3,-9,0)

format 9.6
print "Total power on detector = ",TotalPower
print "RMS spot size           = ",RmsRadius
```





2: Text Viewer

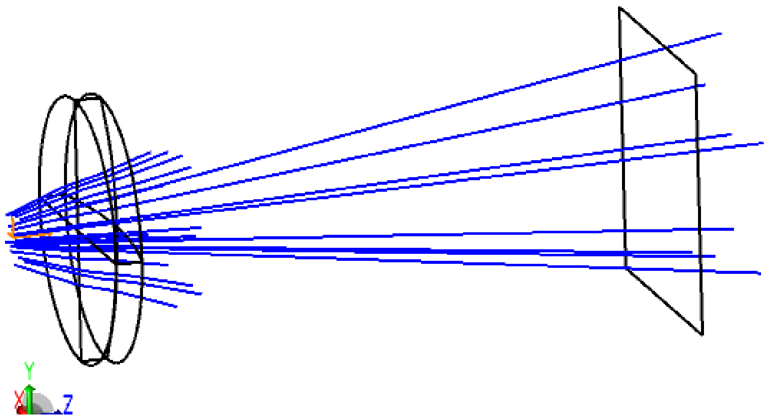
```
Executing C:\Users\optikpc\Documents\Zemax\MACROS\example2.ZPL.
Total power on detector =  0.321546
RMS spot size           = 15.637509
```

Example 3: More Settings

In this macro example,

we’ll use the same setup (as in Example2)

Source Gaussian -> Std.Lens -> Detector Rectangle



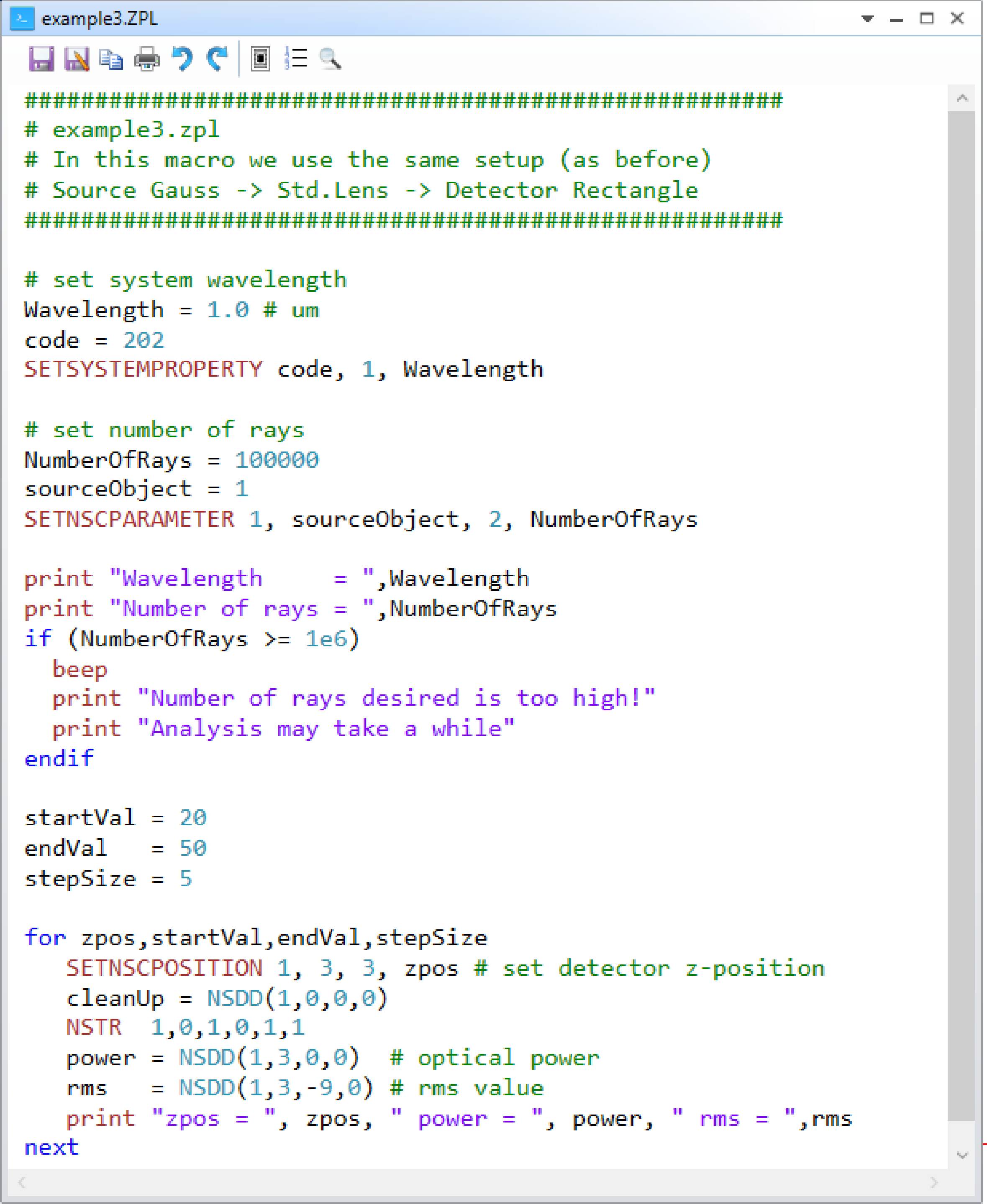
we'll see three keywords:

```
SETSYSTEMPROPERTY code, value1, value2
SETNSCPARAMETER   surface, object, parameter, value
SETNSCPOSITION    surface, object, code, value
```

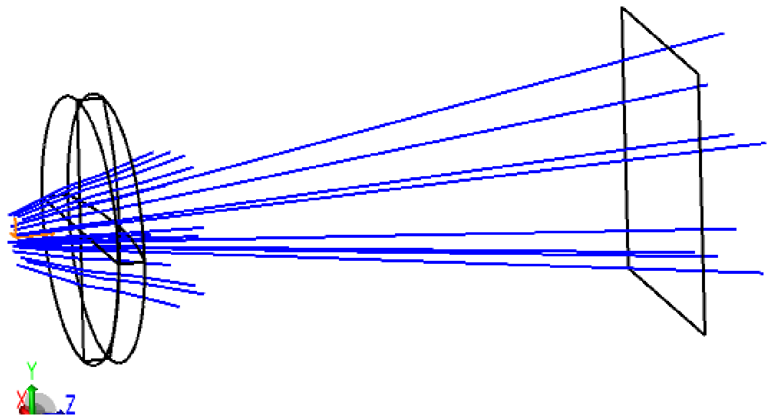
and two control statements:

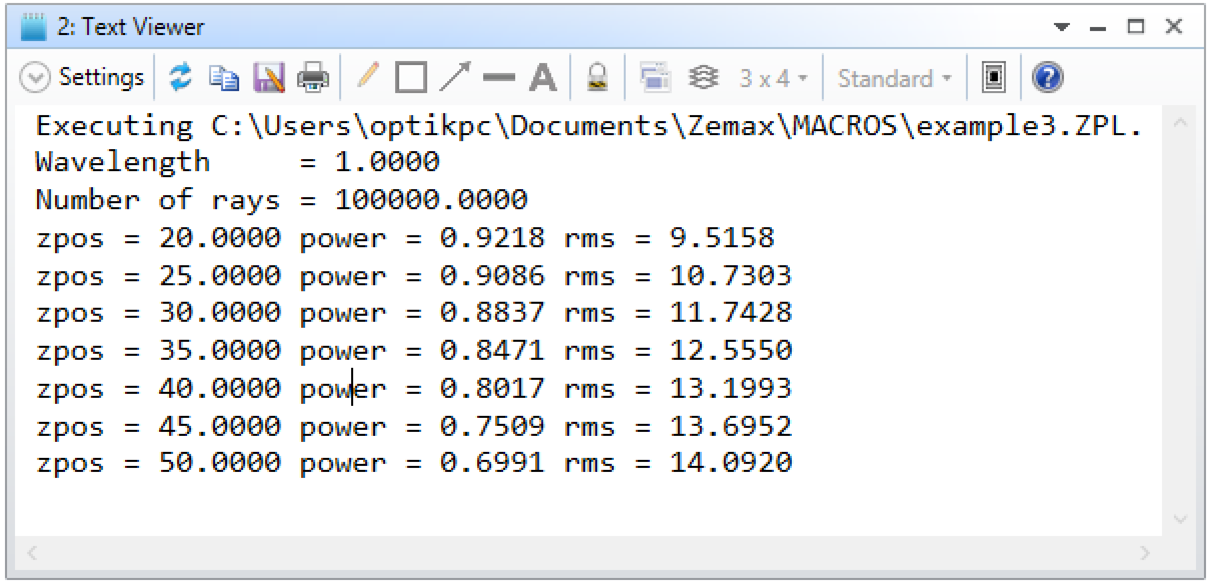
```
IF (expression)
   (commands)
ELSE
   (commands)
ENDIF
```

```
FOR variable, start_value, stop_value, increment
  (commands)
NEXT
```



```
########################################################
# example3.zpl
# In this macro we use the same setup (as before)
# Source Gauss -> Std.Lens -> Detector Rectangle
########################################################

# set system wavelength
Wavelength = 1.0 # um
code = 202
SETSYSTEMPROPERTY code, 1, Wavelength

# set number of rays
NumberOfRays = 100000
sourceObject = 1
SETNSCPARAMETER 1, sourceObject, 2, NumberOfRays

print "Wavelength      = ",Wavelength
print "Number of rays = ",NumberOfRays
if (NumberOfRays >= 1e6)
  beep
  print "Number of rays desired is too high!"
  print "Analysis may take a while"
endif

startVal = 20
endVal   = 50
stepSize = 5

for zpos,startVal,endVal,stepSize
   SETNSCPOSITION 1, 3, 3, zpos # set detector z-position
   cleanUp = NSDD(1,0,0,0)
   NSTR  1,0,1,0,1,1
   power = NSDD(1,3,0,0)  # optical power
   rms   = NSDD(1,3,-9,0) # rms value
   print "zpos = ", zpos, " power = ", power, " rms = ",rms
next
```





```
Executing C:\Users\optikpc\Documents\Zemax\MACROS\example3.ZPL.
Wavelength     = 1.0000
Number of rays = 100000.0000
zpos = 20.0000 power = 0.9218 rms = 9.5158
zpos = 25.0000 power = 0.9086 rms = 10.7303
zpos = 30.0000 power = 0.8837 rms = 11.7428
zpos = 35.0000 power = 0.8471 rms = 12.5550
zpos = 40.0000 power = 0.8017 rms = 13.1993
zpos = 45.0000 power = 0.7509 rms = 13.6952
zpos = 50.0000 power = 0.6991 rms = 14.0920
```

Example 4: Getting Detector Data

In this macro example,
we’ll use the same setup (as in Example2)

* NSDD operand to get detector data as follows:

```
NSDD(1,Det,Pix,Data)
```

Det = detector number (for our case 3)

Pix = pixel number (from 0, 1, 2, …, #number of pixels)

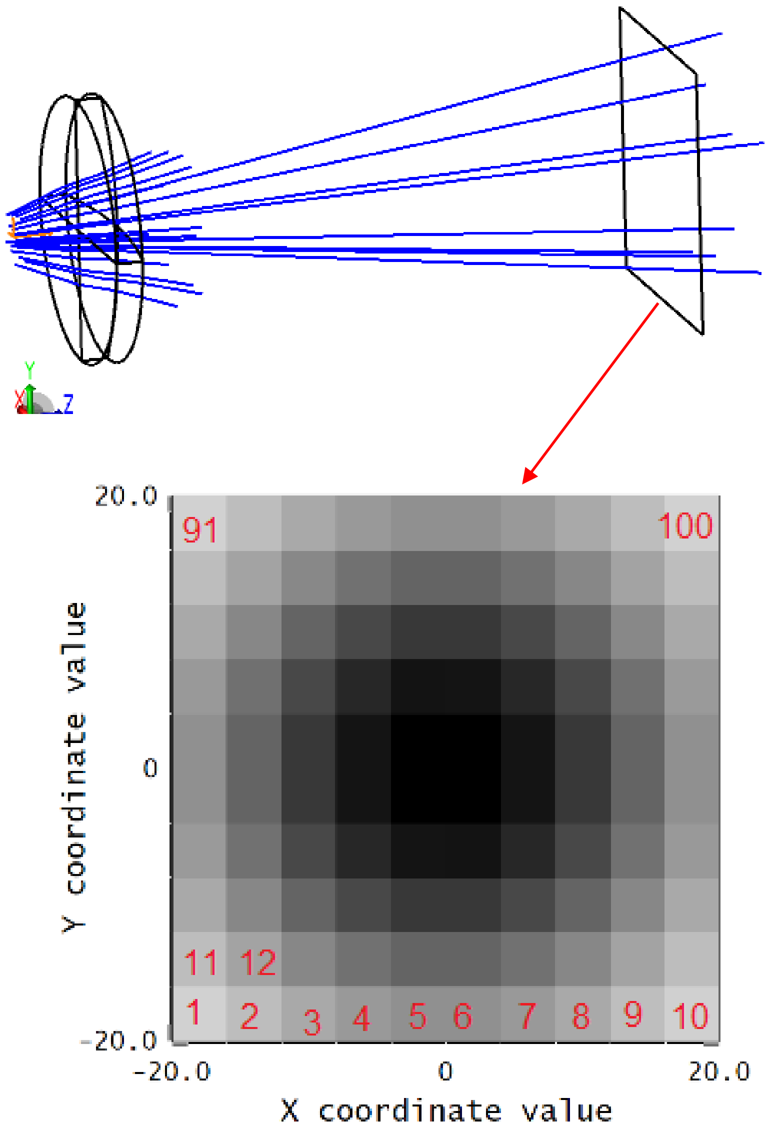
Data = 0 power (flux), Data = 1 power/area (irradiance), . . .

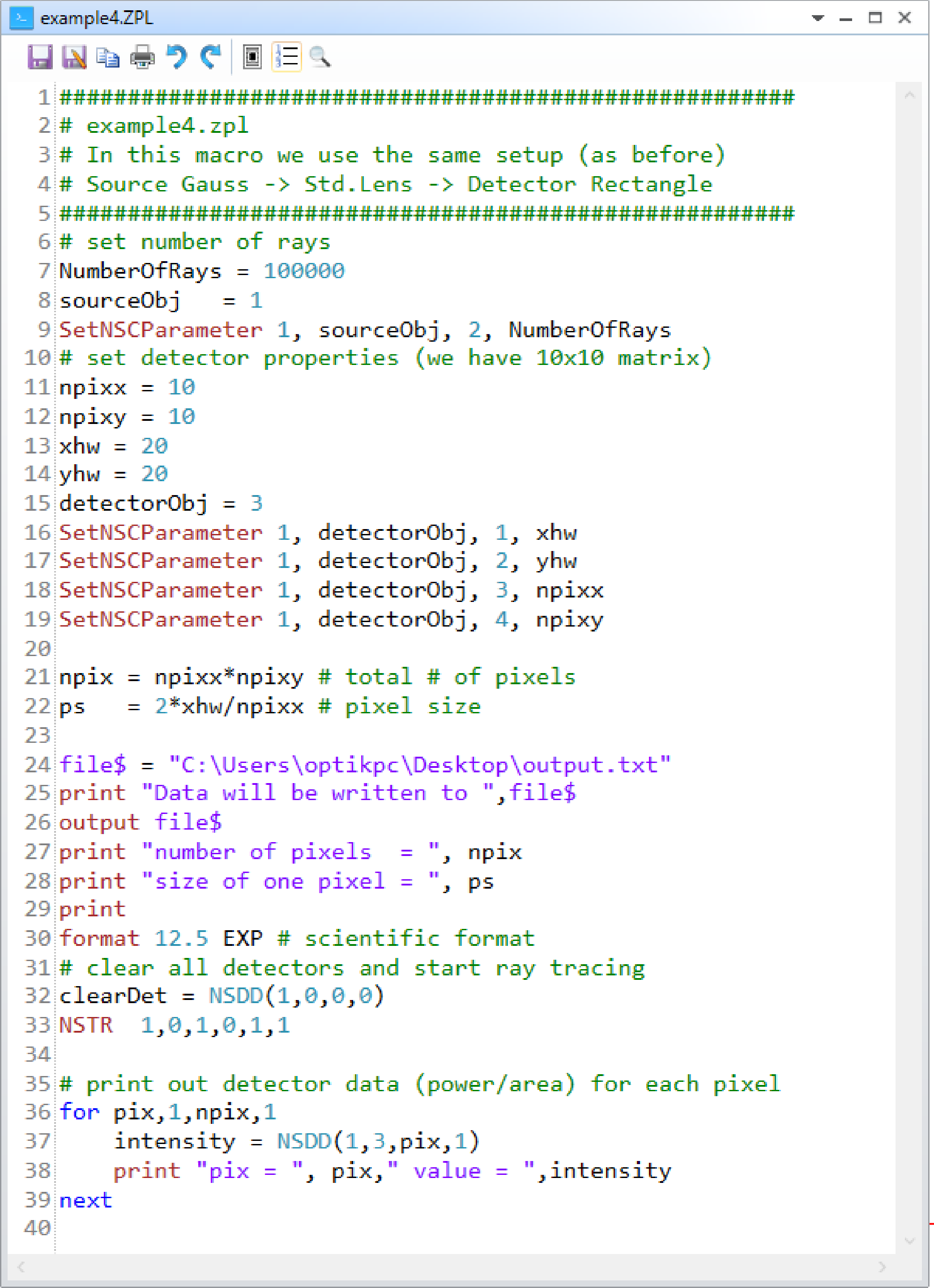
* PRINT can not only output messages to the display,
but also output messages to files using keyword OUTPUT.

OUTPUT SCREEN

OUTPUT filename$

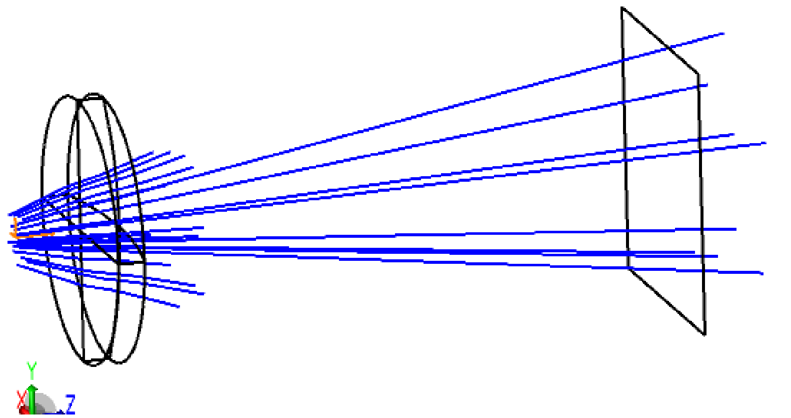
OUTPUT filename$, APPEND

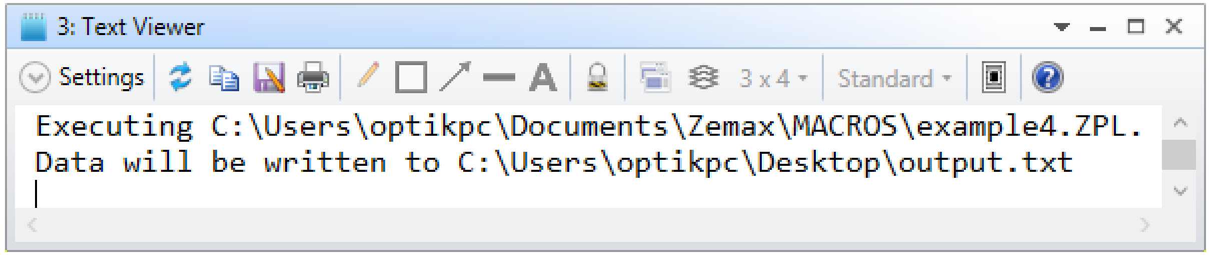




example4.ZPL

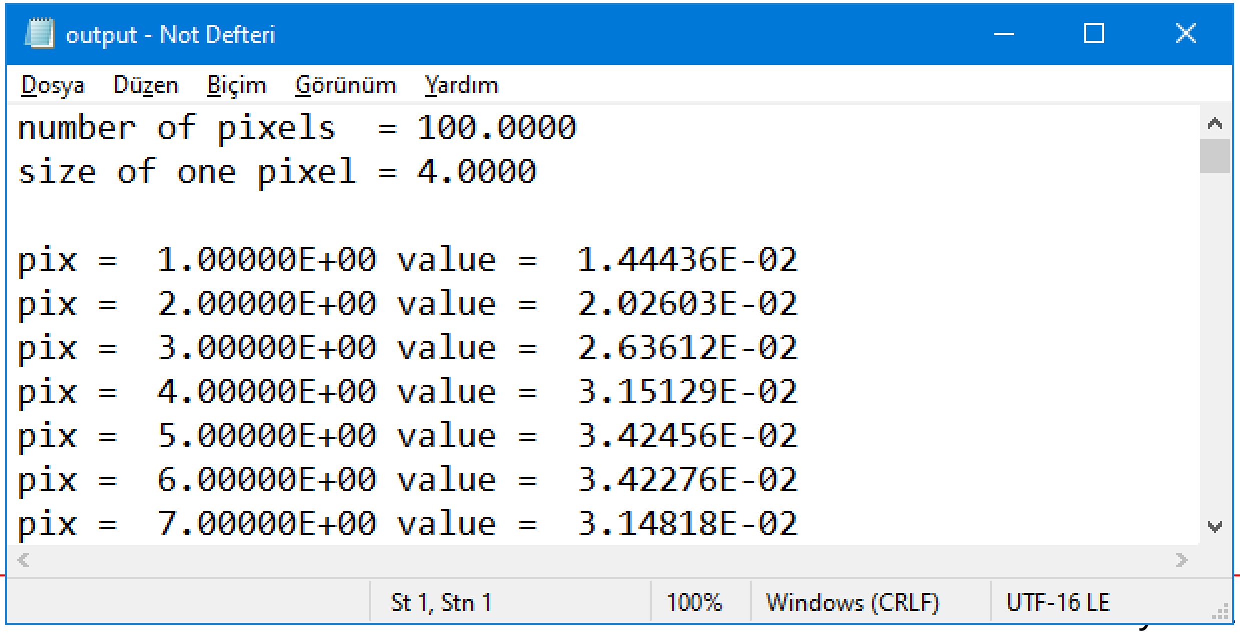
```
#######################################################
# example4.zpl
# In this macro we use the same setup (as before)
# Source Gauss -> Std.Lens -> Detector Rectangle
#######################################################
# set number of rays
NumberOfRays = 100000
sourceObj    = 1
SetNSCParameter 1, sourceObj, 2, NumberOfRays
# set detector properties (we have 10x10 matrix)
npixx = 10
npixy = 10
xhw = 20
yhw = 20
detectorObj = 3
SetNSCParameter 1, detectorObj, 1, xhw
SetNSCParameter 1, detectorObj, 2, yhw
SetNSCParameter 1, detectorObj, 3, npixx
SetNSCParameter 1, detectorObj, 4, npixy

npix = npixx*npixy # total # of pixels
ps   = 2*xhw/npixx # pixel size

file$ = "C:\Users\optikpc\Desktop\output.txt"
print "Data will be written to ",file$
output file$
print "number of pixels  = ", npix
print "size of one pixel = ", ps
print
format 12.5 EXP # scientific format
# clear all detectors and start ray tracing
clearDet = NSDD(1,0,0,0)
NSTR  1,0,1,0,1,1

# print out detector data (power/area) for each pixel
for pix,1,npix,1
    intensity = NSDD(1,3,pix,1)
    print "pix = ", pix," value = ",intensity
next

```





3: Text Viewer

```
Executing C:\Users\optikpc\Documents\Zemax\MACROS\example4.ZPL.
Data will be written to C:\Users\optikpc\Desktop\output.txt
```



output - Not Defteri

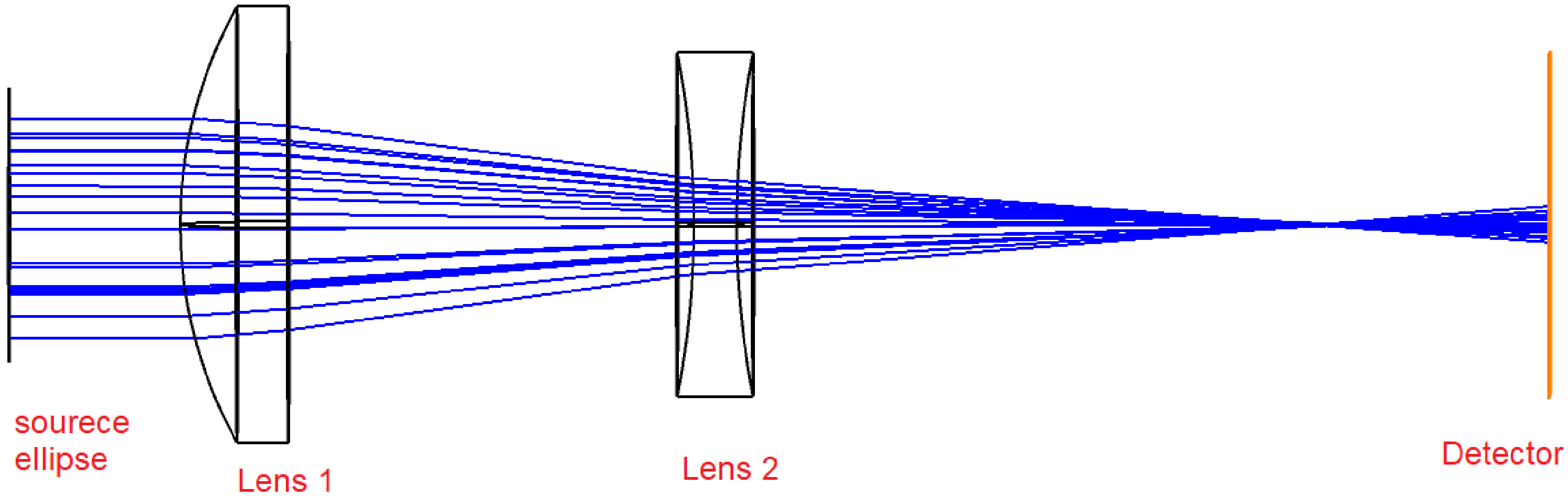
```
number of pixels  = 100.0000
size of one pixel = 4.0000

pix =  1.00000E+00 value =  1.44436E-02
pix =  2.00000E+00 value =  2.02603E-02
pix =  3.00000E+00 value =  2.63612E-02
pix =  4.00000E+00 value =  3.15129E-02
pix =  5.00000E+00 value =  3.42456E-02
pix =  6.00000E+00 value =  3.42276E-02
pix =  7.00000E+00 value =  3.14818E-02
```

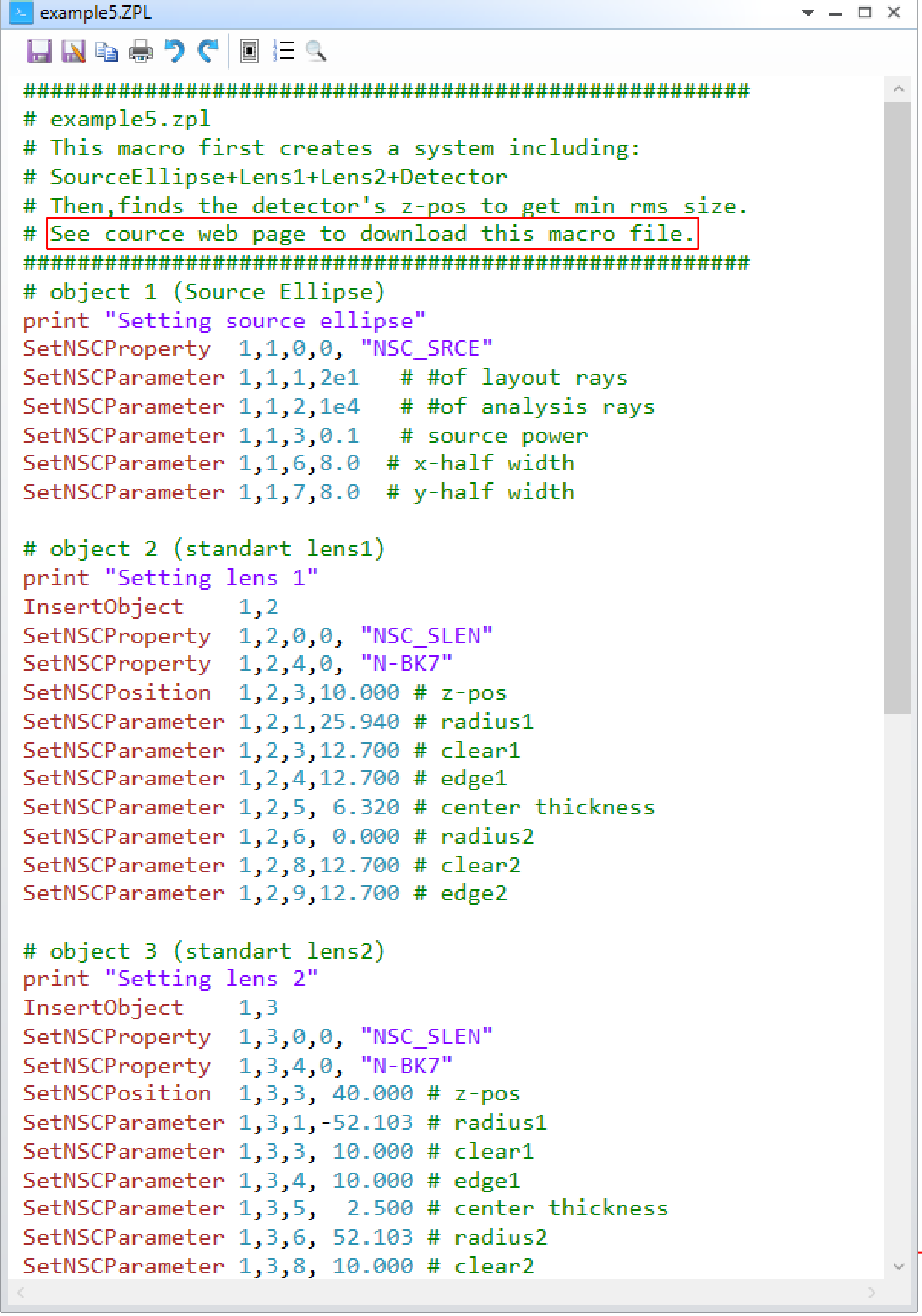
Example 5: Setup Optical Sys. in Macro

In this macro example, we’ll setup a system using a macro file.

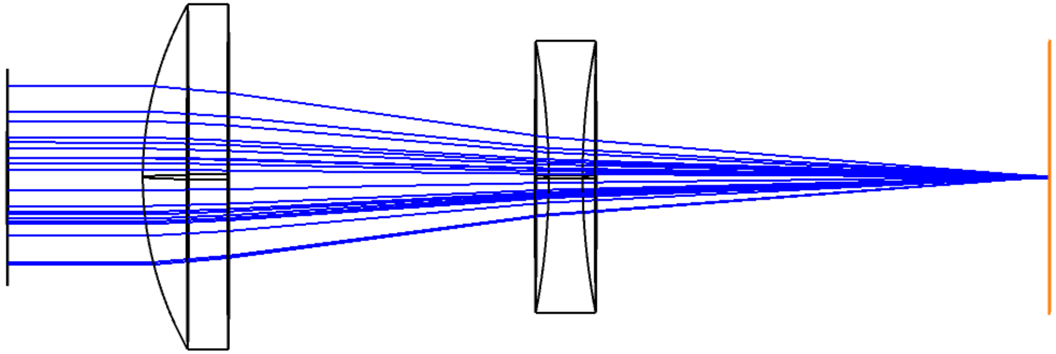
We will use the following keywords (surface=1 for all NSC objects):

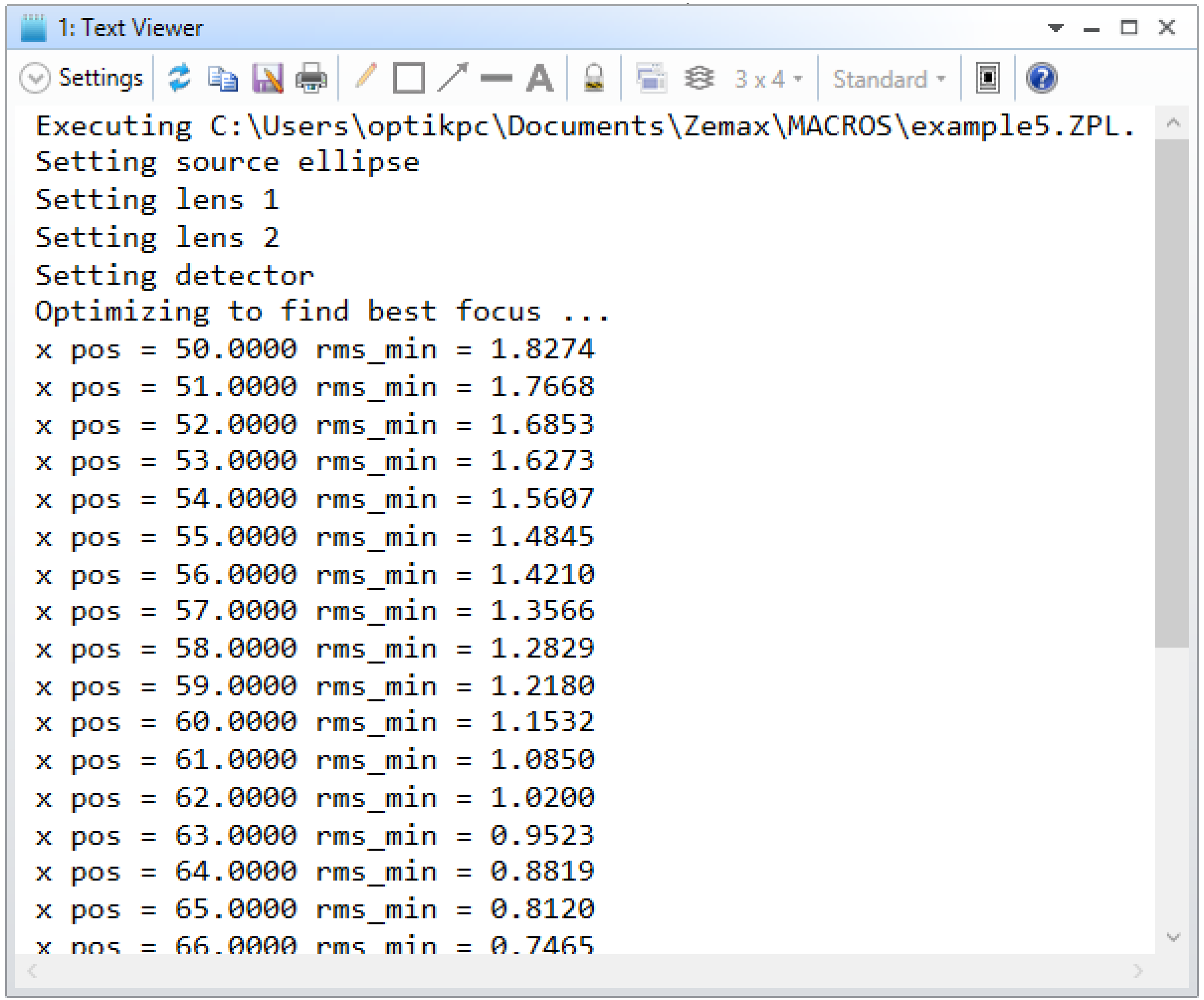
```
INSERTOBJECT      surface, object
SETNSCPROPERTY    surface, object, code, face, value
SETNSCPARAMETER   surface, object, parameter, value
SETNSCPOSITION    surface, object, code, value
SETSYSTEMPROPERTY code, value1, value2
```





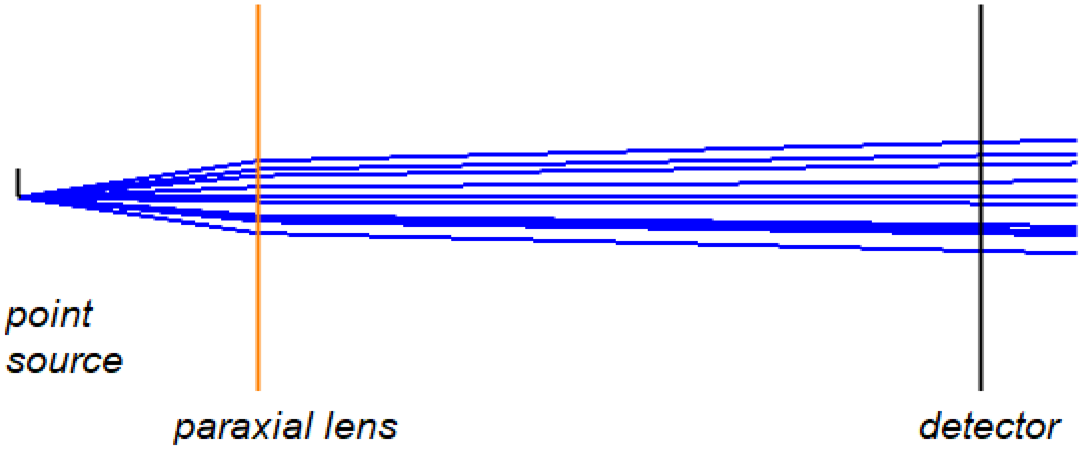
```
###################################################################
# example5.zpl
# This macro first creates a system including:
# SourceEllipse+Lens1+Lens2+Detector
# Then,finds the detector's z-pos to get min rms size.
# See cource web page to download this macro file.
###################################################################
# object 1 (Source Ellipse)
print "Setting source ellipse"
SetNSCProperty  1,1,0,0, "NSC_SRCE"
SetNSCParameter 1,1,1,2e1   # #of layout rays
SetNSCParameter 1,1,2,1e4   # #of analysis rays
SetNSCParameter 1,1,3,0.1   # source power
SetNSCParameter 1,1,6,8.0  # x-half width
SetNSCParameter 1,1,7,8.0  # y-half width

# object 2 (standart lens1)
print "Setting lens 1"
InsertObject    1,2
SetNSCProperty  1,2,0,0, "NSC_SLEN"
SetNSCProperty  1,2,4,0, "N-BK7"
SetNSCPosition  1,2,3,10.000 # z-pos
SetNSCParameter 1,2,1,25.940 # radius1
SetNSCParameter 1,2,3,12.700 # clear1
SetNSCParameter 1,2,4,12.700 # edge1
SetNSCParameter 1,2,5, 6.320 # center thickness
SetNSCParameter 1,2,6, 0.000 # radius2
SetNSCParameter 1,2,8,12.700 # clear2
SetNSCParameter 1,2,9,12.700 # edge2

# object 3 (standart lens2)
print "Setting lens 2"
InsertObject    1,3
SetNSCProperty  1,3,0,0, "NSC_SLEN"
SetNSCProperty  1,3,4,0, "N-BK7"
SetNSCPosition  1,3,3, 40.000 # z-pos
SetNSCParameter 1,3,1,-52.103 # radius1
SetNSCParameter 1,3,3, 10.000 # clear1
SetNSCParameter 1,3,4, 10.000 # edge1
SetNSCParameter 1,3,5,  2.500 # center thickness
SetNSCParameter 1,3,6, 52.103 # radius2
SetNSCParameter 1,3,8, 10.000 # clear2
```

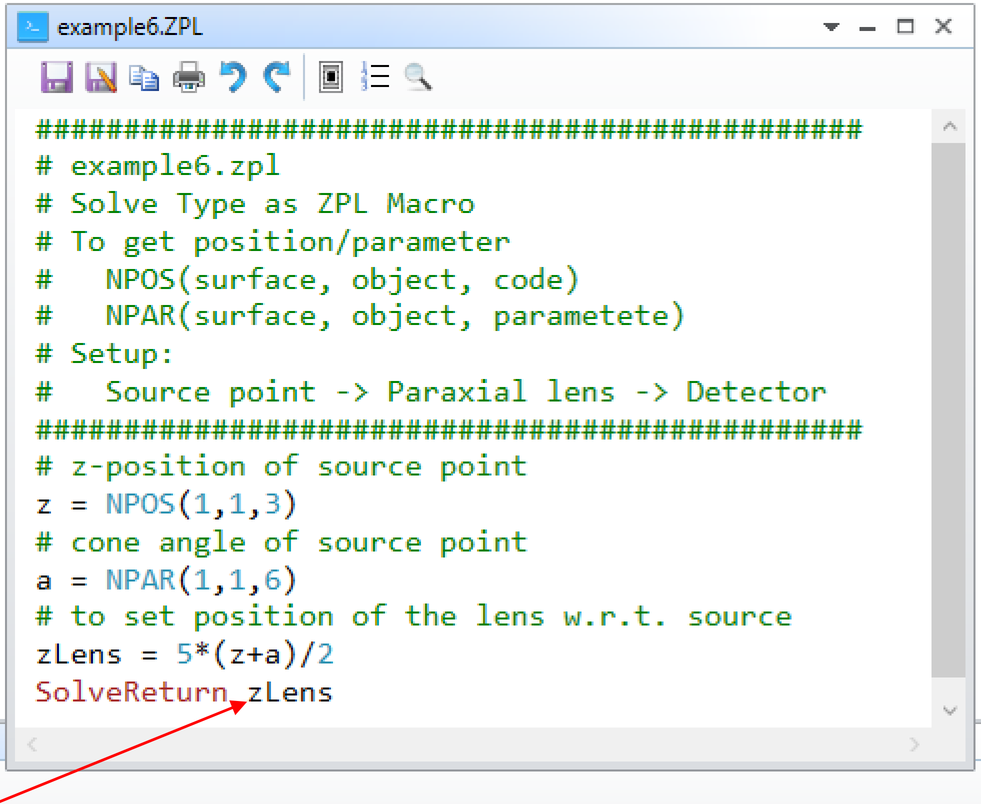




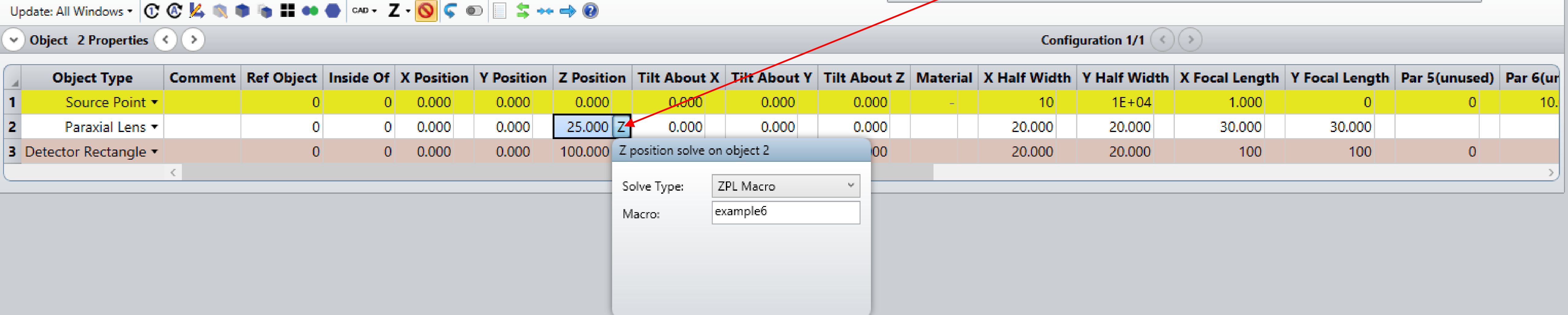
```
Executing C:\Users\optikpc\Documents\Zemax\MACROS\example5.ZPL.
Setting source ellipse
Setting lens 1
Setting lens 2
Setting detector
Optimizing to find best focus ...
x pos = 50.0000 rms_min = 1.8274
x pos = 51.0000 rms_min = 1.7668
x pos = 52.0000 rms_min = 1.6853
x pos = 53.0000 rms_min = 1.6273
x pos = 54.0000 rms_min = 1.5607
x pos = 55.0000 rms_min = 1.4845
x pos = 56.0000 rms_min = 1.4210
x pos = 57.0000 rms_min = 1.3566
x pos = 58.0000 rms_min = 1.2829
x pos = 59.0000 rms_min = 1.2180
x pos = 60.0000 rms_min = 1.1532
x pos = 61.0000 rms_min = 1.0850
x pos = 62.0000 rms_min = 1.0200
x pos = 63.0000 rms_min = 0.9523
x pos = 64.0000 rms_min = 0.8819
x pos = 65.0000 rms_min = 0.8120
x pos = 66.0000 rms_min = 0.7465
```

Example 6: Solve Type as ZPL Macro




example6.ZPL

```
##############################################
# example6.zpl
# Solve Type as ZPL Macro
# To get position/parameter
#   NPOS(surface, object, code)
#   NPAR(surface, object, parametete)
# Setup:
#   Source point -> Paraxial lens -> Detector
##############################################
# z-position of source point
z = NPOS(1,1,3)
# cone angle of source point
a = NPAR(1,1,6)
# to set position of the lens w.r.t. source
zLens = 5*(z+a)/2
SolveReturn zLens
```


Non-Sequential Component Editor

Update: All Windows

Object 2 Properties

Configuration 1/1

	Object Type	Comment	Ref Object	Inside Of	X Position	Y Position	Z Position	Tilt About X	Tilt About Y	Tilt About Z	Material	X Half Width	Y Half Width	X Focal Length	Y Focal Length	Par 5(unused)	Par 6(ur
1	Source Point		0	0	0.000	0.000	0.000	0.000	0.000	0.000	-	10	1E+04	1.000	0	0	10.
2	Paraxial Lens		0	0	0.000	0.000	25.000 Z	0.000	0.000	0.000		20.000	20.000	30.000	30.000		
3	Detector Rectangle		0	0	0.000	0.000	100.000			00		20.000	20.000	100	100	0	

Z position solve on object 2

Solve Type: ZPL Macro

Macro: example6