Use of an Application Programming Interface (API) to allow non-optical designers to perform specific optical evaluations

Mark C. Sanson
Corning Tropel Corporation, 60 O'Connor Road, Fairport, New York 14450
SansonMC@corning.com

Abstract: A person lacking training in optical design programs may perform tasks using the design program's power. This paper addresses how API, with Windows Component Object Model, allows a person lacking understanding of the operation of the design program to run optical sensitivity routines.

1. Introduction

Commercial optical design programs available today contain hundreds if not thousands of command and analysis options for an optical designer. To run these programs effectively requires not only training, but a fair amount of experience with the software package. However, a person with an understanding of optics, but no training in the use of such programs, may benefit from the use of some of these tools for select tasks. This paper discusses the use of an Application Programming Interface (API) with one of the commercial optical design programs, ORA's Code V[1]. CODEV has designed into its software the ability for Microsoft Windows applications to access some of the tools of the optical design program through Microsoft Windows Component Object Model (COM)[2]. This allows applications like Excel and C++ to drive CODEV.

This paper discusses capabilities the COM interface can bring to optical design programs, and it covers an example of using Excel to run CODEV. API is not new, but more recently the optical design programs make better use of it. A large amount of the power of CODEV is available through the COM ORA has developed. By creating a simple interface with another Windows application, an optical designer can develop a program which allows someone untrained in the use of the optical design software to perform a specific task. This allows the time of the optical designer to be spent on other tasks, while increasing the abilities of someone not skilled in the optical design program. In addition to using the Windows program as the interface, the person can utilize the built in capabilities of the Windows application. The use of Excel to run a sensitivity routine and create charts and tables of the different sensitivities provides an example of this advantage.

2. API and COM

With the use of Microsoft Windows COM, the API that is built into the CODEV optical design program allows another application to start CODEV and run certain commands. The COM allows a client/server relationship to be established where the optical design program is the server (Fig. 1) and the client can be any software using COM, such as Microsoft Excel with Visual Basic for Applications (VBA), MATLAB, or C++. The client can issue several general and standard analysis commands that have been developed in CODEV.

The user can utilize the client program's benefits in conjunction with the power of the optical design program. This feature allows custom user interfaces, graphing of data, statistical analysis, and ease of use for repetitive design tasks. The full capabilities of the design program are available, but COM does not allow the passage of graphics to the client program.

International Optical Design Conference 2006, edited by G. Groot Gregory,
Joseph M. Howard, R. John Koshel, SPIE Vol. 6342, 634205,
© 2006 SPIE-OSA · 0277-786X/06/$15 · doi: 10.1117/12.692191

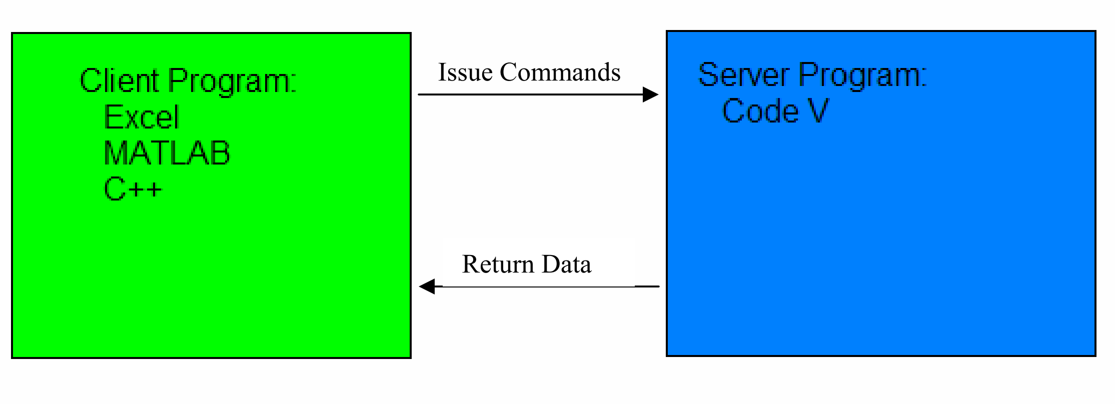
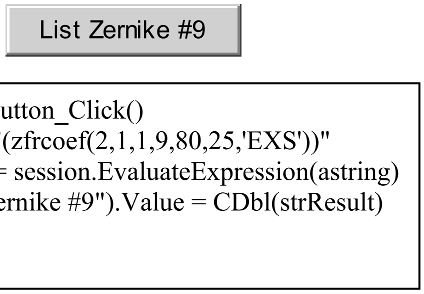


Fig. 1. Client/server relationship with CODEV's COM

3. Sensitivity Application in Excel

In an optical design program, it is fairly straightforward to gather information on manufacturing sensitivities. Often this data is taken to an application like Excel to graph relevant data. Observing a graph of the sensitivities may ease identification of sensitive parts of the lens for different types of perturbations and aberrations. Any time this sensitivity analysis is re-computed, there is time involved with moving the data into Excel again and possibly reworking some of the charts. This work must be done by someone trained in the use of the design program as well as Excel. This additional work is the motivation to use Excel, the end program, as the interface for a sensitivity analysis. With the development of the Excel interface the user can run CODEV without the design program being seen, and the sensitivity can be rerun without needing to manually import the data to Excel.

Using Excel's Visual Basic for Applications, an session of CODEV can be started, but it will remain hidden from the user. The COM allows certain functions to be run with the design program. These functions allow the user to analyze ray trace data, MTF data, wavefront data, Zernike data, etc. However the full benefit of the optical design program is the ability to enter any command that would normally be entered in the programs command line. Creating the visual basic code that issues the commands to CODEV that the trained user would know allows an individual not proficient in CODEV to run the program. Instead of issuing a command by typing "eva (zfrcoef(2,1,1,9,80,25,'EXS'))", the operator of Excel can press a simple button with code linked to it.

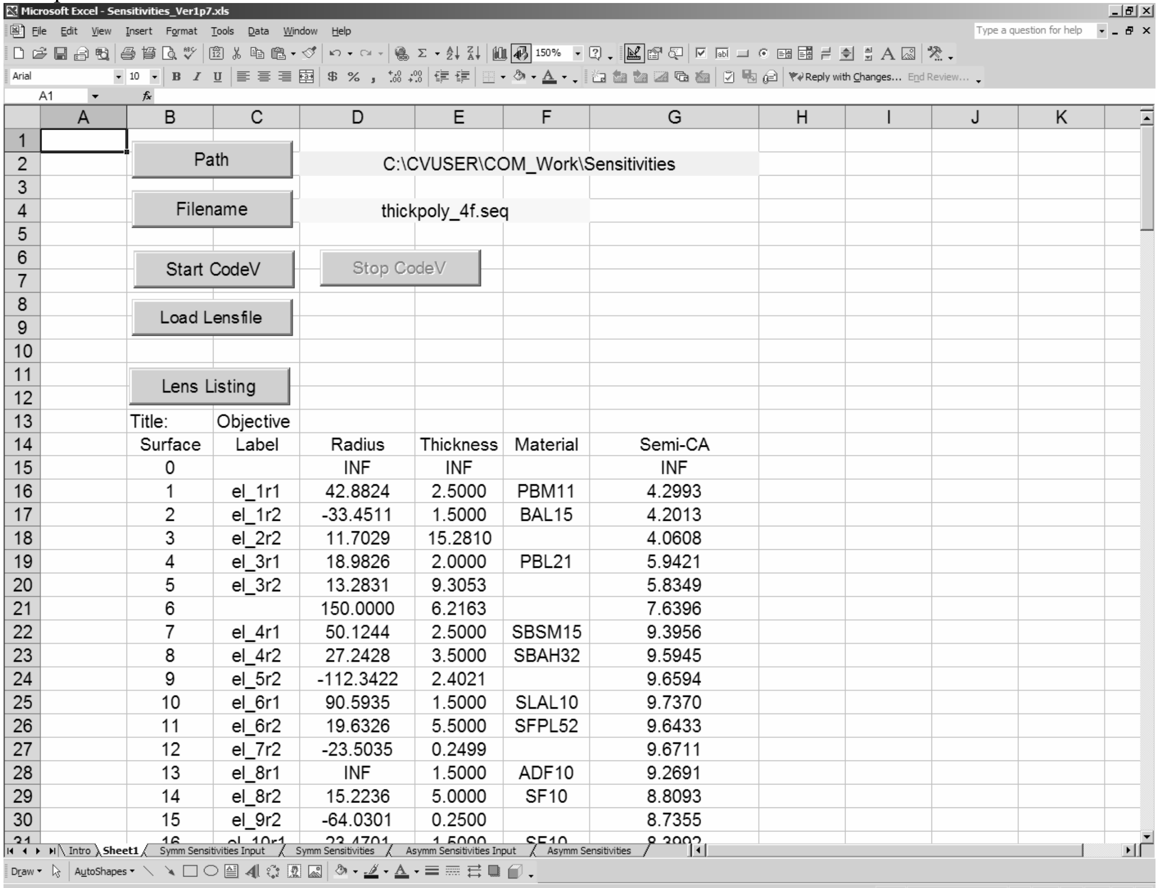


```
Private Sub Button_Click()
    astring = "(zfrcoef(2,1,1,9,80,25,'EXS'))"
    strResult = session.EvaluateExpression(astring)
    Range("Zernike #9").Value = CDbl(strResult)
End Sub
```

Fig. 2. Sample Command Button and VBA code behind it to allow Zernike #9 to be listed

The button and underlying code shown in fig. 2 would calculate Zernike #9 and the value would then be placed in a named cell on the worksheet.

The sample Excel workbook contains several worksheets. One worksheet allows the lens to be loaded and listed (fig. 3), two worksheets allow the user to choose which symmetric and asymmetric perturbations to run, and finally two worksheets list the results of the perturbations in a table and create several charts. A simple interface on the first worksheet allows the user to load the lens to have a sensitivity routine performed on it. Buttons contain the code that will allow commands to be sent to the server optical program and/or information to be gathered from the server optical program. After loading the lens, the user has the ability to list the lens with VBA code automatically formatting the prescription. Different tasks can be arranged in different workbooks, as has been done in the sensitivity example. The symmetric perturbations (radii, thickness, indices, etc.) are on one sheet and the asymmetric perturbations (decenters, tilts, etc.) are on another. The user does not need to know any CODEV commands to be able to run the basic sensitivities. The application automatically generates tables and charts of the different perturbations.



Path: C:\CVUSER\COM_Work\Sensitivities

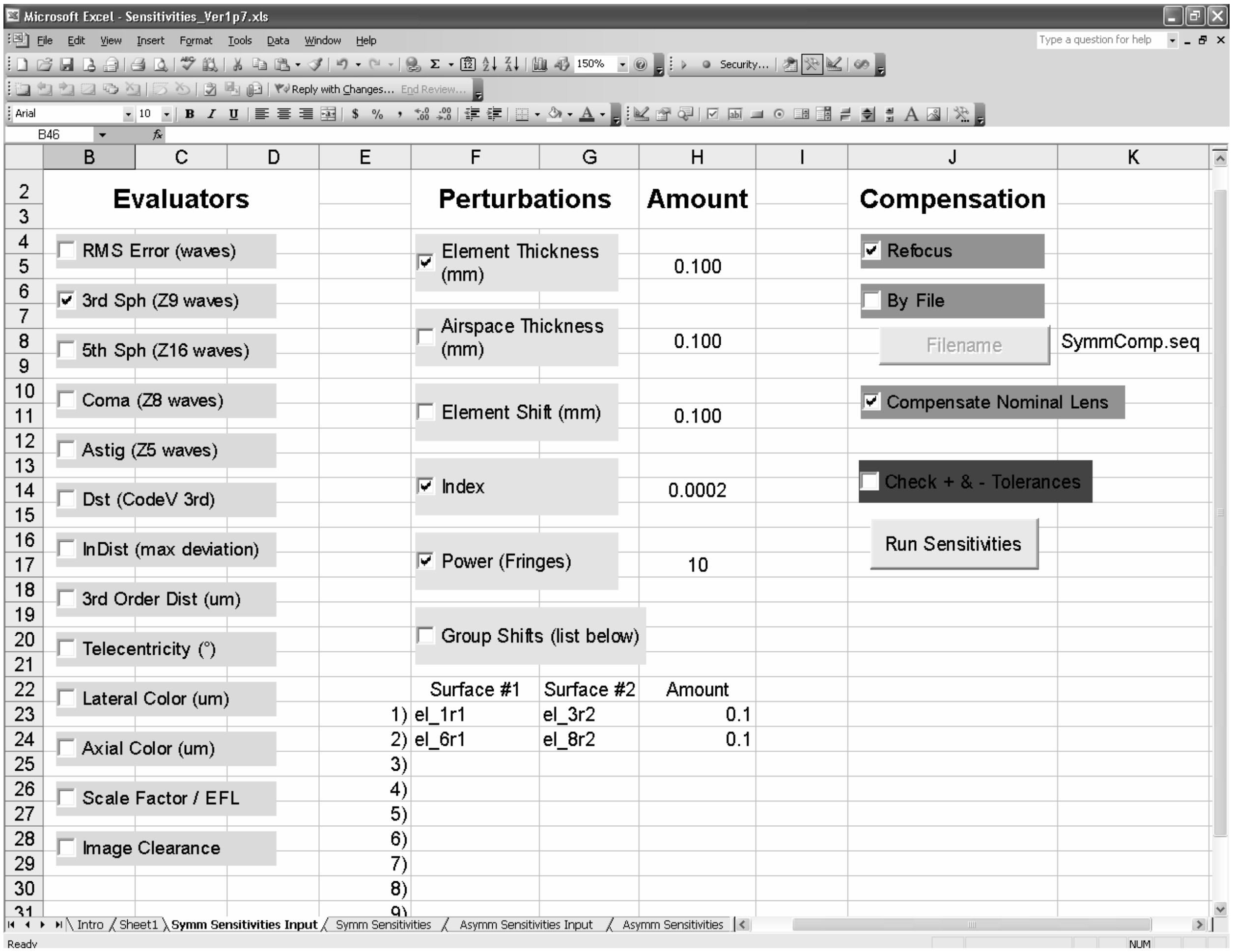
Filename: thickpoly_4f.seq

Title: Objective

Surface	Label	Radius	Thickness	Material	Semi-CA
0		INF	INF		INF
1	el_1r1	42.8824	2.5000	PBM11	4.2993
2	el_1r2	-33.4511	1.5000	BAL15	4.2013
3	el_2r2	11.7029	15.2810		4.0608
4	el_3r1	18.9826	2.0000	PBL21	5.9421
5	el_3r2	13.2831	9.3053		5.8349
6		150.0000	6.2163		7.6396
7	el_4r1	50.1244	2.5000	SBSM15	9.3956
8	el_4r2	27.2428	3.5000	SBAH32	9.5945
9	el_5r2	-112.3422	2.4021		9.6594
10	el_6r1	90.5935	1.5000	SLAL10	9.7370
11	el_6r2	19.6326	5.5000	SFPL52	9.6433
12	el_7r2	-23.5035	0.2499		9.6711
13	el_8r1	INF	1.5000	ADF10	9.2691
14	el_8r2	15.2236	5.0000	SF10	8.8093
15	el_9r2	-64.0301	0.2500		8.7355
16	el_10r1	23.4701	1.5000	SF10	8.3992

Fig. 3. Sample screen shot displaying the simple interface

The desired sensitivities and evaluators are selected by the user with check boxes (fig 4). Each perturbation is individually performed, and the change in performance criteria is computed. The type of perturbations and performance criteria available to the user has been pre-determined, and coded into the sensitivity routine. The content is based upon advice from the optical designer. Writing the code for all necessary options allows the user to realize the maximum benefit from the program in terms of the number and diversity of systems for which the user may utilize the program.


Microsoft Excel - Sensitivities_Ver1p7.xls
File Edit View Insert Format Tools Data Window Help
Type a question for help
150%
Security...
Reply with Changes... End Review...
Arial 10
B46
Intro Sheet1 Symm Sensitivities Input Symm Sensitivities Asymm Sensitivities Input Asymm Sensitivities
Ready
NUM

Evaluators

☐ RMS Error (waves)
☑ 3rd Sph (Z9 waves)
☐ 5th Sph (Z16 waves)
☐ Coma (Z8 waves)
☐ Astig (Z5 waves)
☐ Dst (CodeV 3rd)
☐ InDist (max deviation)
☐ 3rd Order Dist (um)
☐ Telecentricity (°)
☐ Lateral Color (um)
☐ Axial Color (um)
☐ Scale Factor / EFL
☐ Image Clearance

Perturbations	Amount
☑ Element Thickness (mm)	0.100
☐ Airspace Thickness (mm)	0.100
☐ Element Shift (mm)	0.100
☑ Index	0.0002
☑ Power (Fringes)	10
☐ Group Shifts (list below)	

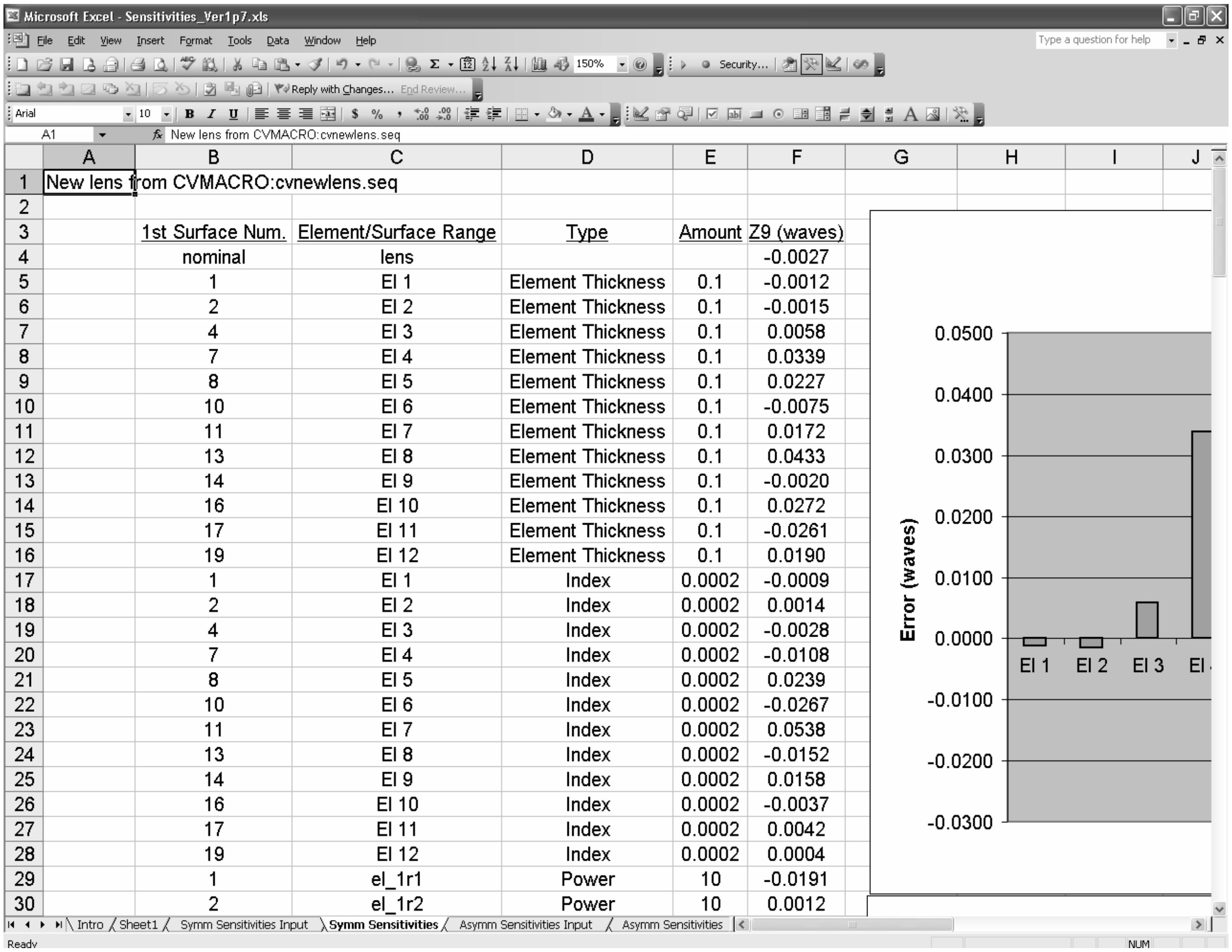
	Surface #1	Surface #2	Amount
1)	el_1r1	el_3r2	0.1
2)	el_6r1	el_8r2	0.1
3)			
4)			
5)			
6)			
7)			
8)			
9)			

Compensation

☑ Refocus
☐ By File
Filename SymmComp.seq
☑ Compensate Nominal Lens
☐ Check + & - Tolerances
Run Sensitivities

Fig. 4 Sample screen shot displaying the symmetric perturbation worksheet

The optical design program performs all of the perturbations and analysis behind the scenes. The session of the optical design program that is run does not appear on the desktop. The design program then sends the sensitivity data back to the client program. Excel formats and lists the data (Fig. 5). Then individual graphs for each perturbation type are automatically generated (Fig. 6). The graphs can be enhanced with VBA by adding the perturbation type and amount. Repetitive tasks are done automatically with VBA, such as formatting the decimal precision, underlining headings, graphing, and highlighting. With some extra VBA programming the customization of the charts can occur and be unique based upon data supplied from the optical design program.



	A	B	C	D	E	F
1	New lens from CVMACRO:cvnewlens.seq					
2						
3		1st Surface Num.	Element/Surface Range	Type	Amount	Z9 (waves)
4		nominal	lens			-0.0027
5		1	El 1	Element Thickness	0.1	-0.0012
6		2	El 2	Element Thickness	0.1	-0.0015
7		4	El 3	Element Thickness	0.1	0.0058
8		7	El 4	Element Thickness	0.1	0.0339
9		8	El 5	Element Thickness	0.1	0.0227
10		10	El 6	Element Thickness	0.1	-0.0075
11		11	El 7	Element Thickness	0.1	0.0172
12		13	El 8	Element Thickness	0.1	0.0433
13		14	El 9	Element Thickness	0.1	-0.0020
14		16	El 10	Element Thickness	0.1	0.0272
15		17	El 11	Element Thickness	0.1	-0.0261
16		19	El 12	Element Thickness	0.1	0.0190
17		1	El 1	Index	0.0002	-0.0009
18		2	El 2	Index	0.0002	0.0014
19		4	El 3	Index	0.0002	-0.0028
20		7	El 4	Index	0.0002	-0.0108
21		8	El 5	Index	0.0002	0.0239
22		10	El 6	Index	0.0002	-0.0267
23		11	El 7	Index	0.0002	0.0538
24		13	El 8	Index	0.0002	-0.0152
25		14	El 9	Index	0.0002	0.0158
26		16	El 10	Index	0.0002	-0.0037
27		17	El 11	Index	0.0002	0.0042
28		19	El 12	Index	0.0002	0.0004
29		1	el_1r1	Power	10	-0.0191
30		2	el_1r2	Power	10	0.0012

Fig. 5 Sample screen shot displaying the symmetric sensitivities worksheet

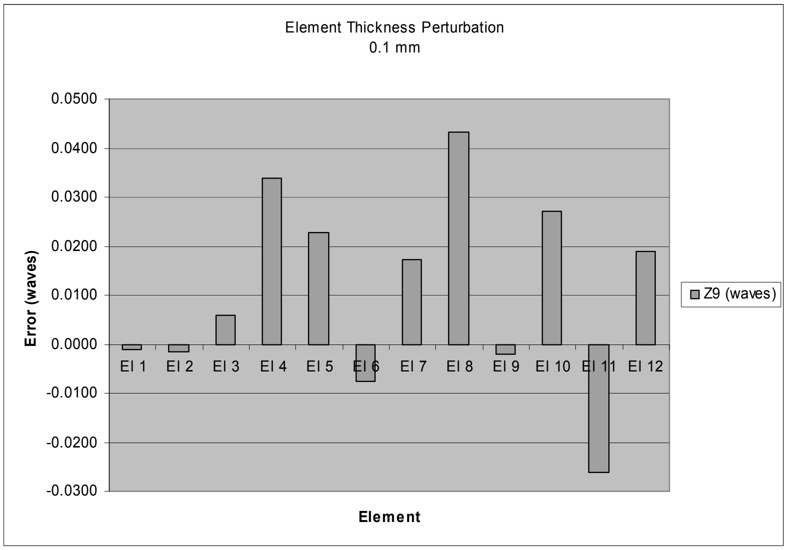


Fig. 6 Sample graph that is automatically generated from sensitivity data

4. Summary

The ability to use an optical design program as a server for other programs increases the number of people capable of performing optical routines. Secondary programs which support COM can be integrated with design code to perform repetitive specialized analysis routines, data collection, and graphical display of information. Some of these tasks may be tasks that the optical program cannot perform alone. The client program even may drive the optical program in its tasks, based upon analysis of data gathered from the program. A custom task-specific user interface allows all of these advantages. An API that performs an optical sensitivity analysis has been demonstrated.

5. References

[1] CODEV is a registered trademark of Optical Research Associates
[2] Windows is a registered trademark of Microsoft Corporation