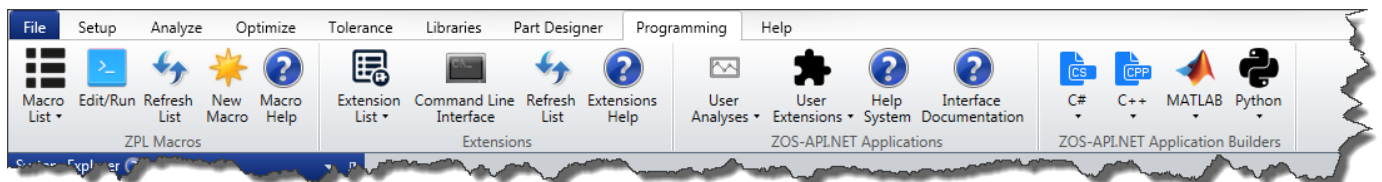


编程选项卡（The Programming Tab）

此菜单用于访问 OpticStudio 的“编程”功能。



ZPL 宏组

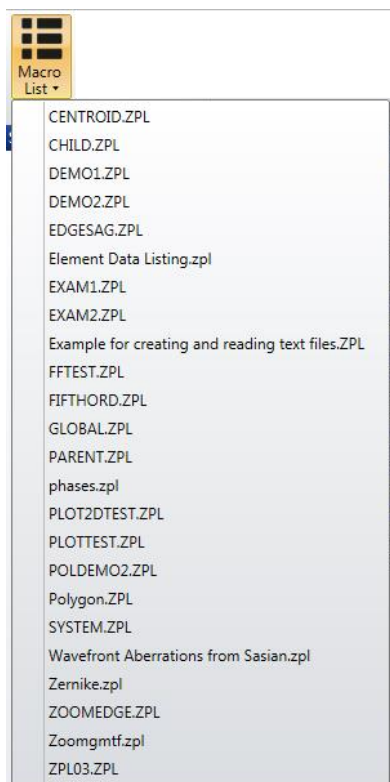
ZPL 宏组是“编程”菜单下的子栏目。



有关更多详情，请参阅[关于 ZPL](#)一节。

宏列表

“宏列表”选项卡位于“编程”菜单的“ZPL 宏”栏目。

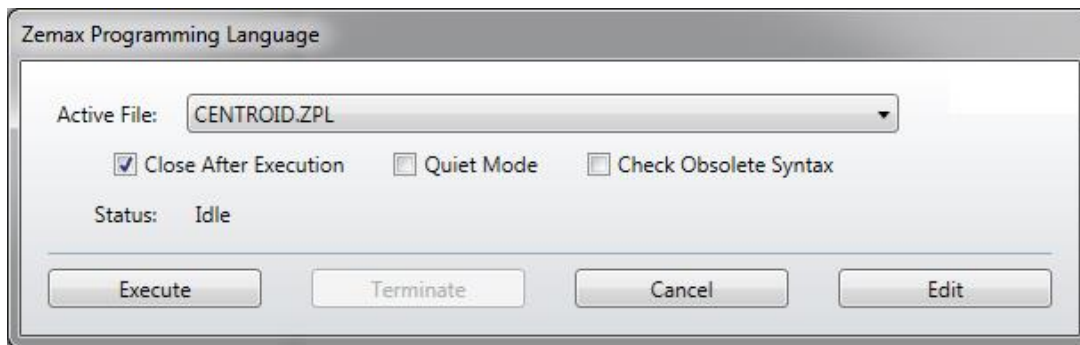


宏列表下拉菜单中显示了当前宏文件夹中所有的 ZPL 宏文件。（参阅“文件夹”）。

单击宏名称，宏将立即执行。

编辑/执行

“编辑/执行”选项卡位于编程菜单中的 ZPL 宏栏目。



此工具可编辑或执行指定的宏。

要执行 ZPL 宏，请在“编程”菜单的“ZPL 宏”栏目中选择“编辑/执行”选项卡，从“当前文件”列表中选择要执行的宏，然后单击“执行”。

Zemax 将开始执行宏。整个宏运行结束后，弹出文本输出窗口，显示“输出”命令结果或者错误提示信息，可使用 CLOSEWINDOW 关键字关闭此输出窗口。

ZPL 控件对话框具有以下选项：

当前文件：可从其下拉菜单列表中选择宏。所有列出的宏都是扩展名以 .ZPL 结尾的文本文件。且文件必须存放在 ZPL 宏文件夹中；请参阅“文件夹”。也可以存放在宏文件夹中的任何子目录下。

执行后关闭：若勾选，则宏运行完之后，ZPL 控制对话框将自动关闭。

安静模式：若勾选，默认的输出文本窗口将不会显示，可在不需要生成图形的宏中使用。

检查过时语法：若勾选，ZEMAX 将检测宏文本中过时的语法。

状态：在执行宏期间，Zemax 将在此区域输出一个状态信息，显示正在运行的宏行数。状态信息每隔 0.25 秒更新一次。

执行：单击开始执行所选择的宏。

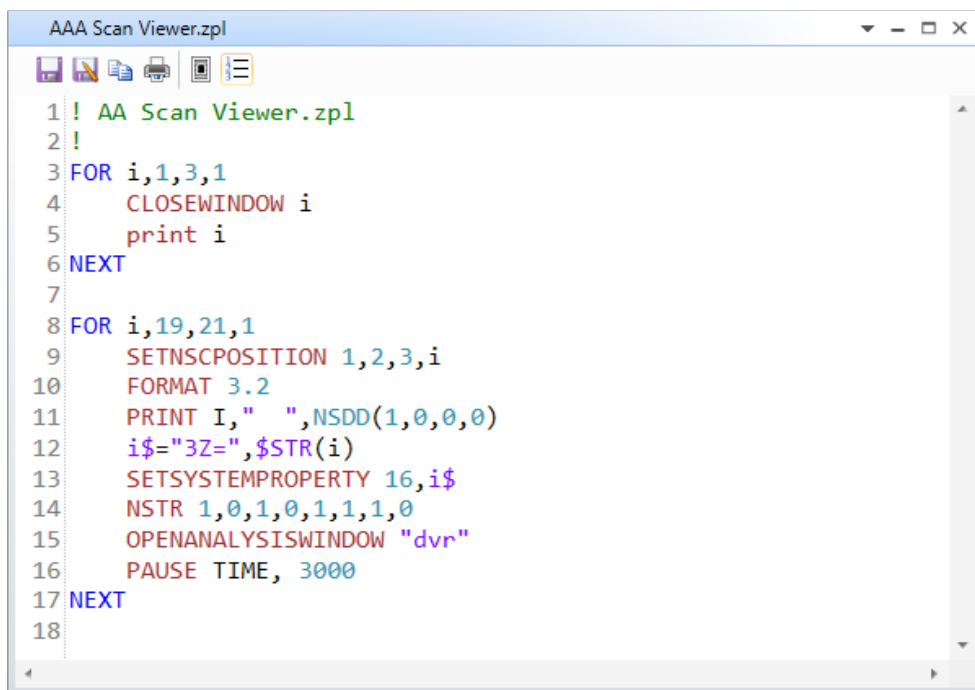
终止：停止执行当前的宏。。

取消：终止当前执行的宏，若当前没有正在执行的宏，则关闭 ZPL 控制对话框。

编辑：调用 OpticStudio 的文本编辑器，可用于修改或重命名宏。在宏文本中，不同 ZPL 语法显示为不同的颜色，如下表所示：

语法分类	颜色
注释	绿 色
字符串	紫 色
数字	浅蓝 色
控制关键字	蓝 色
函数关键字	棕 色
标识符	黑 色

例如，名为 “AAA Scan Viewer” 的示例宏的截屏如下图所示：



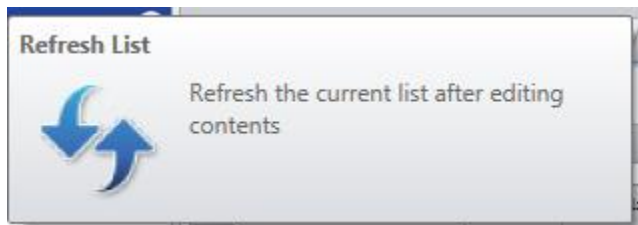
```

1 ! AA Scan Viewer.zpl
2 !
3 FOR i,1,3,1
4     CLOSEWINDOW i
5     print i
6 NEXT
7
8 FOR i,19,21,1
9     SETNSCPOSITION 1,2,3,i
10    FORMAT 3.2
11    PRINT I," ",NSDD(1,0,0,0)
12    i$="3Z=", $STR(i)
13    SETSYSTEMPROPERTY 16,i$
14    NSTR 1,0,1,0,1,1,1,0
15    OPENANALYSISWINDOW "dvr"
16    PAUSE TIME, 3000
17 NEXT
18

```

刷新列表

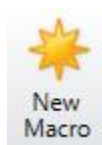
“刷新列表”选项卡位于编程菜单中的 ZPL 宏栏目。



当添加或删除新的宏文件时，可用此工具更新当前宏列表。

新建宏

“新建宏”选项卡位于编程菜单中的 ZPL 宏栏目。



“新建宏”选项卡可打开一个 ZEMAX 文本编辑器来新建一个 ZPL 宏。创建的宏将自动保存为 *.ZPL 格式，并保存在“宏用户数据”文件夹中。

宏帮助

“宏帮助”选项卡位于编程菜单中的 ZPL 宏栏目。



“宏帮助”选项卡打开 HTML OpticStudio 帮助文件中“关于 ZPL”的章节。单击 HTML 窗口中

“关于 ZPL”旁边的“+”展开子目录查看详情。



关于 ZPL

本节介绍如何使用 Zemax 编程语言 (ZPL)。如果在 HTML 对话框中查看“关于 ZPL”，需要在内容选项卡中展开标题来查看相关子目录。

简介

Zemax 编程语言 (ZPL) 是一种专门为使用 Zemax 而设计的宏语言。ZPL 提供了自定义扩展的功能。如果用户要构建一个 ZEMAX 内部没有的特殊计算或者图形显示，则可以编写自己的 ZPL 宏来实现。

ZPL 与 BASIC 编程语言类似，不仅支持部分 BASIC 结构体和关键字，而且还增加了光线追迹的特殊语法函数。ZPL 很容易使用，用户参照本章节的介绍以及实例很快就上手。对于外部编程功能，请参阅“关于扩展 (About Extensions)”章节。

ZPL 是一种强大的编程语言。虽易于使用，但用户需要自己进行错误检查、代码逻辑调试以及编

程实践。因此，对于如何编写宏代码来执行某项详细的计算，ZEMAX 技术支持不能给出编写意见，只能确保所有的 ZPL 函数和关键字正常运行。如果您需要使用 ZEMAX 宏，但又不想自己编程，您可以随时联系 ZEMAX 技术支持，获得满足要求的自定义程序的开发报价。我们在开发宏程序方面有相当丰富的经验，可以很快编写出您所需要的宏程序。

创建 ZPL 宏

ZPL 宏由一系列保存在文本文件中的宏命令组成。宏命令可以是赋值、关键字或注释。详情请参阅“ZEMAX 编程语言”。

要创建 ZPL 宏，最便捷的方法是从自带的宏文件组中选择目标相近的宏进行修改。如果您首次尝试编写宏，您可以参阅本章节末尾的示例文件，或者 ZEMAX 自带的宏文件夹中的宏示例。

可使用任何文本编辑器来创建 ZPL 文件（如记事本编辑器）。ZPL 文件名可自定义，但必须是.ZPL 格式，且文件必须存放在 ZPL 文件夹中，默认是 <data>\Macros。要更改此路径，请参阅“文件夹”。

ZPL 宏的行宽有一定限制，如果出现“行过长 line too long”的错误提示，则需要将一行分成几段更短的行。

ZPL 概述

ZPL 宏由一系列保存在文本文件中的宏命令组成。宏命令可以是赋值、关键字或注释。赋值可以是数字或字符串(文本)数据。赋值和关键字都可以用表达式作为变量，只是语法略有不同，如下文所述。

赋值

赋值的一般语法是

变量 = (表达式)

(表达式) 可以是一个明确的数值（如 5）、已赋值的变量名称或复杂的算术表达式（包含函数、常量和变量）。在所有情况下，等号右侧的表达式进行求值，并将结果赋值给左侧指定的变量。

赋值最简单的形式是表达式为一个固定值，例如：

x = 5

需注意：首先，表达式之前不需要声明变量，也就是说在将固定值 5 赋值给变量 “x” 之前，x 不一定要存在。如果之前 “x” 已被赋予某个值，那么现在将被重新赋值。其次，表达式结尾不需要特殊终止符（如 C 语言中的 “;”）。因此，每条 ZPL 命令必须独占一行。

以下是赋值表达式的示例：

```
x = SQRT(5)
y = SINE(x)
z = SQRT(x+5*(7-x))
```

ZPL 中的 “数值函数” 中定义了许多内置函数，如 SQRT (平方根) 和 SINE (正弦)。注意：ZPL 不区分大小写，SQRT() 和 sqrt() 是同一个函数。

ZPL 中内置了函数 SQRT (平方根) 和 SINE (正弦)。还有很多这样的函数，均在 “数字函数” 中定义。请注意，ZPL 不区分大小写；SQRT() 和 sqrt() 是同一个函数。本文中所有函数和关键字为大写字母，其它内容均为小写。

关于字符串赋值，请参阅 “字符串变量” 一节。

关键字

关键字的一般语法是

```
KEYWORD 变量 1, 变量 2, 变量 3...
```

某些关键字没有变量，某些关键字则有多个变量。变量可以是数值表达式，也可以是字符串常量或字符串变量。某些关键字可混合使用数值和字符串变量。

以关键字 PRINT 为例，PRINT 之后跟随输出项目列表，用逗号隔开。例如，ZPL 宏命令如下所示：

```
x = 3 y = 4
z = SQRT(x * x + y * y) PRINT "The hypotenuse is ",z
```

将输出以下内容：

```
The hypotenuse is 5.0000
```

需注意，ZPL 运算符具有优先级。ZPL 优先级从高到底依次为：括号、函数（如 SQRT）、逻辑运算符（如 ==）、乘除法，最后是加减法。

关键字有很多，均在 “关键字” 一节中进行了详细描述。

注释

在 ZPL 宏中有 3 种方法来添加注释：第 1 是以关键字 **REM** 开始的一行，第 2 是以 “!” 符号开始的一行，第 3 是在行中的任何位置输入 “#” 符号，且 “#” 符号不在字符串内。宏的任何位置均可添加空白行。如下所示为添加注释的 3 个示例：

```
REM this is a remark
!This is also a remark
x = 5 # The # symbol allows comments on the same line as a valid command
```

通过注释，用户可以很轻松地了解和修改宏，并且对宏的正常执行没有任何影响。

创建图形

ZPL 宏中有一些函数用于生成图形，包括 **GRAPHICS**、**GTEXT**、**GLEN- SNAME** 等等。有关更多详情，请参阅“图形”一节。也可以简单地使用关键字 “**PLOT**” 和 “**PLOT2D**” 生成 ZEMAX 2D 图形，详情请参阅 “**PLOT**” 和 “**PLOT2D**”。

数值变量

变量可为未知数值提供临时存储空间，其具体数值将在宏执行时被定义。在您需要一个新的变量时，Zemax 将为您执行大部分工作。例如，如下所示命令：

```
x = 5
```

Zemax 将为新变量 “x” 分配内存，并记录与之相关的值。一旦定义了变量，便可以在后续表达式中使用。使用 ZPL 变量时有以下规则：

变量名不能包含任何 ZPL 逻辑运算符或界定的特殊字符，如 (、)、=、+、-、*、/、!、>、<、^、&、|、#、" 以及空格字符。

变量名不能与关键字或函数重名，比如 **THIC** 或 **RAYX**。由于 ZPL 不区分大小写，因此也不能使用 **Thic** 或 **rayX** 来替换。

每个变量名长度限制在 28 个字符内。

所有 ZPL 数值变量均以 64 位双精度数值存储。

数组变量

数组变量可以是一维或者多维数组，其数值可以是双精度或者整数。不同于数值变量，数组变量必须提前定义才能使用。定义的语法为：

```
DECLARE 名称, 类型, 维数, 维度 1 [, 维度 2 [, 维度 3 [, 维度 4] 等等.]]
```

名称可以是任意如前文所述的合法变量名。类型必须是 **DOUBLE** 或 **INTEGER**，指定该数组变量的类型。整数值“维数”定义了该数组的维数，而非大小，并且必须在 1 到 4 之间，包括 1 和 4。整数“维度 1”、“维度 2”等等定义了该数组的大小。需注意，数组变量从索引 1 开始，因此对于大小为 10 的数组，其有效索引是从 1 到 10。

可以在宏的任何位置定义数组变量，不需要在宏的开始位置声明。可用关键字 **RELEASE** 释放数组变量的相关内存。其语法为：

```
RELEASE 名称
```

关键字 **RELEASE** 命令可写可不写，因为在宏终止后，将自动释放声明变量的相关内存。但是，如果只是在宏执行的某一部分中需要大型数组，则关键字 **RELEASE** 对于节省内存很有用。

数组变量的赋值语法为：

```
名称 (index1, index2, ...) = 值
```

可以使用相同的基本语法来检索数组中存储的值：

```
值 = 名称 (index1, index2, ...)
```

以下示例代码定义了一个二维数组变量，对每个元素进行赋值，并输出数组变量值，然后释放数组的内存：

```
DECLARE Z, DOUBLE, 2, 5, 5
FOR i, 1, 5, 1
  FOR j, 1, 5, 1
    Z(i, j) = i + j
```

```

        NEXT j
NEXT i

FORMAT 8.0 k = 0
FOR i, 1, 5, 1
    FOR j, 1, 5, 1
        PRINT k, i, j, Z(i,j)
        k = k + 1
    NEXT j
NEXT i

RELEASE Z

```

数值运算符

ZPL 宏支持基本数值运算，比如加减乘除。其语法如下所示。

```

x = y + z
x = y - z
x = y * z
x = y / z

```

ZPL 宏所有其它的运算只能通过使用数值函数或数值逻辑运算符来实现，这些将在后续章节中进行描述。

数值逻辑运算符

逻辑运算符用于构造复杂命令，命令结果为 **1** 或 **0**。大部分逻辑运算符采用(左表达式) (运算符) (右表达式)格式，类似于算术表达式，比如 **1 + 2**。不过，非运算符“**!**”除外，非运算符仅使用单个变量，格式是 **!(右表达式)**。逻辑运算符使用时规定：**0** 表示“假”，任何非 **0** 值均表示“真”。如果(右表达式)为 **0** (假)，则非运算符返回 **1** (真)，如果(右表达式)为非 **0** (真)，则非运算符返回 **0** (假)。非运算符常用于 **IF** 命令中，比如：

```
IF !x THEN PRINT "x is zero."
```

IF 命令中的量也可以使用其它逻辑运算符。例如，**IF** 命令可能包含两个条件，这两个条件必须均成立才能执行 **THEN** 命令：

```
IF (x > 1) & (y < 2) THEN PRINT "Both conditions are true."
```

这两个条件通过逻辑“与”表达式（用 & 表示）来关联。需注意，圆括号优先级更高。ZPL 中的数值逻辑运算符如下表所述：

数值逻辑运算符

逻辑运算符	说明
&	与，仅当两个表达式均为非 0 时返回 1。
	或，如果至少一个表达式为非 0，则返回 1。
^	异或，如果仅一个表达式为非 0，则返回 1。
!	非，如果 (右表达式) 为非 0，则返回 0，否则返回 1。
==	等于，如果表达式相等，则返回 1。
>	大于，如果左表达式大于右表达式，则返回 1。
<	小于，如果左表达式小于右表达式，则返回 1。
>=	大于等于，如果左表达式大于等于右表达式，则返回 1。
<=	小于等于，如果左表达式小于等于右表达式，则返回 1。
!=	不等关系，如果两边表达式不相等，则返回 1。

字符串变量

ZPL 支持字符串变量和基本字符串运算。字符串变量最多可以包含 360 个字符。字符串变量不需要声明，可以随时使用赋值命令创建，比如：

```
newstring$ = "Here is the new string"
```

需注意，字符串变量与数值变量的区别在于，字符串变量的末尾有 \$ 字符。

ZPL 也支持字符串函数，可用于提取文本数据，比如

```
title$ = $LENSNAME()
```

需注意，函数 \$LENSTITLE() 是以 \$ 字符开头。表明此函数返回字符串结果。

字符串运算符

可以使用 “+” 运算符来合并字符串变量。其语法为：

```
C$ = A$ + B$
```

合并运算中可包含字符串常量：

```
total$ = "A$ is " + A$ + " and B$ is " + B$
```

可以在定义命令中使用字符串函数，比如

```
this$ = "Here is the lens title: " + $LENSNAME()
```

字符串变量可以像其他字符串一样用 “PRINT” 输出：

```
PRINT "Here is A$: ", A$
```

需注意，PRINT 函数只能输出单个字符串变量；在输出命令中不能使用合并运算或字符串函数。可先将字符串合并到新的字符串中，然后再输出此新的字符串：

```
A$ = B$ + C$  
PRINT A$
```

或者用逗号进行合并运算：

```
PRINT A$, B$, C$
```

ZPL 不能直接输出字符串函数，比如：

```
PRINT $LENSNAME() !这不正确!!!
```

可先将字符串函数结果赋给某个变量，然后再输出此变量：

```
Z$ = $LENSNAME()  
PRINT Z$
```

字符串逻辑运算符

字符串逻辑运算符与数值逻辑运算符类似。主要区别在于，参与比较的表达式是字符串，而不是数值。ZPL 中定义字符串逻辑运算符如下表所示：

字符串逻辑运算符

逻辑	描述
\$==\$	等于，如果 左边的字符串和右边的字符串相同，则返回 1。
\$>	大于，如果左边的字符串大于右边的字符串，则返回 1。
\$<	小于，如果左边的字符串小于右边的字符串，则返回 1。
\$>=	大于等于，如果左边的字符串大于等于右边的字符串，则返回 1。
\$<=	小于等于，如果左边的字符串小于等于右边的字符串，则返回 1。
\$!=	不等于，如果左边的字符串和右边的字符串不相同，则返回 1。

例如，用 IF 命令对字符串进行逻辑运算，如下所示：

```
A$ = "TEST"  
BS = "TEST"  
IF (A$==$B$) THEN PRINT "Strings are identical."
```

数值函数

数值函数可以作为数值变量赋值命令的右侧表达式，也可以作为关键字中的变量表达式。数值函数可能不需要变量，也可能需要一个或多个变量。所有的函数均返回单个值。某些函数（如表示主波长的 **PWAV()**）返回的值与变量无关，因此不需要提供变量，但圆括号仍然需要的。

下表中列出了所有 ZPL 数值函数。如果函数语法为 **FUNC()**形式，则表示不需要变量。**FUNC(x)** 表示需要一个变量，**FUNC(x,y)** 表示需要两个变量，以此类推。

ZPL 函数列表

函数	变量	返回值
ABSO(x)	数值表达式	表达式的绝对值。
ACOS(x)	数值表达式	反余弦（以弧度为单位）。
APMN(x)	有效的表面编号	最小半径值。对于支撑杆(星形)孔径，表示臂的宽度。对于矩形和椭圆形孔径，表示孔径的 X -半宽。
APMX(x)	有效的表面编号	最大半径值。对于支撑杆(星形)孔径，表示臂的数量。对于矩形和椭圆形孔径，这是孔径的 Y -半宽。
APOI(px, py)	归一化光瞳坐标	光线在指定光瞳坐标上的高斯分布因子。
APXD(x)	有效的表面编号	孔径偏心 (X) 值。
APYD(x)	有效的表面编号	孔径偏心 (Y) 值。
APTP(x)	有效的表面编号	整数代码，用于描述指定表面上的孔径类型。
ASIN(x)	数值表达式	反正弦（以弧度为单位）。
ASPR()	(空)	当前显示图形的纵横比。
ATAN(x)	数值表达式	反正切（以弧度为单位）。

ATYP()	(空)	系统孔径类型代码：0 表示 EPD，1 表示 F/#，2 表示 NA，3 表示光阑尺寸浮动。
AVAL()	(空)	系统孔径值。
CALD(i)	索引	返回调用宏(CALLMACRO)缓冲区中指定索引处的数值。请参阅“从宏中调用宏”
CONF()	(空)	返回当前结构组态的编号，介于 1 和总的多重结构数之间，包括首尾值。
CONI(x)	有效的表面编号	指定表面的圆锥系数。
COSI(x)	以弧度为单位的数值表达式	表达式的余弦。
CURV(x)	有效的表面编号	指定表面的曲率。
EDGE(x)	有效的表面编号	指定表面当前半口径对应的边缘厚度。
EOFF()	(空)	文件结束标志。如果到达文件结尾，则返回 1，否则返回 0。仅在执行了关键字 READ 之后才有效。
ETIM()	(空)	从上个 TIMER 开始所经过的时间（以秒为单位）。
EXPE(x)	数值表达式	e 的表达式次幂。
EXPT(x)	数值表达式	10 的表达式次幂。
FICL(vec#)	矢量编号，在 1 到 4 之间，包括首尾值。	光纤耦合效率。请参阅“使用 FICL() 函数”。
FLDX(x)	有效的视场编号	指定视场的 X 角度或高度。
FLDY(x)	有效的视场编号	指定视场的 Y 角度或高度。
FTYP()	(空)	如果当前视场类型是角度（以度数为单位），则返回 0；如果是物高，则返回 1；如果是近轴像高，则返回 2；如果是实际像高，则返回 3；如果是经纬角，则返回 4。高度单位是镜头单位。

FVAN(x)	有效的视场编号	指定视场的渐晕角。
FVCX(x), FVCY(x)	有效的视场编号	指定视场 x 或 y 方向的渐晕压缩因子。
FVDX(x), FVDY(x)	有效的视场编号	指定视场 x 或 y 方向的渐晕偏心因子。
FWGT(x)	有效的视场编号	指定视场的权重。
GABB(x)	有效的表面编号	指定表面材料的阿贝数。
GAUS(x)	标准偏差	从高斯分布、0 均值及指定的标准偏差中返回一个随机值
GETT(窗口, 行, 列)	要提取的数值的窗口编号、行编号以及列编号。列由空格分隔。	从任何打开的文本窗口中提取数值。可对 Zemax 窗口中显示的任何值进行计算。此函数假定句点 (.) 为小数点分隔符。要使用 Windows 设置中定义的小数点分隔符, 请使用 GETL() 。
GETL(窗口, 行, 列)	要提取数值的窗口编号、行编号以及列编号。列由空格分隔。	从任何打开的文本窗口中提取数值。可对 Zemax 窗口中显示的任何值进行计算。此函数使用 Windows 的“区域和语言”设置中定义的小数点分隔符。也可参阅 GETT() 。
GIND(x)	有效的表面编号	指定表面材料 的 d 光折射率。
GLCA(x)	有效的表面编号	指定表面中心在全局坐标下的 x 矢量
GLCB(x)	有效的表面编号	指定表面中心在全局坐标下的 y 矢量
GLCC(x)	有效的表面编号	指定表面中心在全局坐标下的 z 矢量
GLCM(表面, 项)	“表面”必须是有效的表面编号, “项”是介于 1 到 12 之间的整数。	如果“项”等于 1 到 9, 则返回值为 R11 、 R12 、 R13 、 R21 、 R22 、 R23 、 R31 、 R32 或 R33 。如果“项”等于 10 到 12, 则返回值为全局偏移矢量的 x 、 y 或 z 分量。请参阅“全局坐标参考面”。
GLCX(x)	有效的表面编号	指定表面中心在全局坐标下的 x 坐标。

GLCY(x)	有效的表面编号	指定表面中心在全局坐标下的 y 坐标。
GLCZ(x)	有效的表面编号	指定表面中心在全局坐标下的 z 坐标。
GNUM(A\$)	任何字符串变量名称	如果 A\$是有效的玻璃名称，如 BK7，则 GNUM 返回玻璃库中对应玻璃的编号。该编号可供关键字 “ 设置表面类型 (SETSURFACEPROPERTY)” 使用，用于定义某个表面的玻璃类型。如果 A\$ 没有对应库中的任何玻璃，则 GNUM 返回 0。如果 A\$是 “MIRROR “，则 GNUM 返回 -1。
GPAR(x)	有效的表面编号	指定表面材料的色散系数。
GRIN(s, w, x, y, z)	表面编号#、波长编号# 以及 x、y 和 z 坐标	返回在波长编号 w 下，指定表面 s 上对应 x、y、z 坐标点的折射率。对梯度折射率介质和非梯度折射率介质均有效。
GTEM(代码)	对于复选框值，使用玻璃替代模板的代码为 0，排除数据不全玻璃的代码为 1，标准的代码为 11，首选的代码为 12，过时的代码为 13，特殊的代码为 14，使用相对成本、CR、FR、SR、AR 和 PR 的代码分别为 21-26。对于数值，使用相对成本、CR、FR、SR、AR 和 PR 的代码分别为 31-36。	返回玻璃替代模板中的数据。复选框返回 0 表示 false（未选中），返回任何非 0 值表示 true（选中）。
IMAE(种子)	种子数据值	返回几何图像分析效率。请参阅详述 “几何图像分析”。如果种子为 0，则每次调用 IMAE 时都将使用相同的随机数。如果种子不为 0，则每次调用 IMAE 时都将使用唯一的随机数。
INDX(表面)	有效的表面编号	主波长的折射率。请参阅 ISMS。
INTE(x)	数值表达式	返回不大于变量的最大整数。

ISMS(表面)	有效的表面编号	如果指定表面是第奇数个镜面，或者其之后的非镜面，则返回 1，否则返回 0。
LOGE(x)	正数值表达式	以 e 为底的表达式的对数 ($\ln(x)$)。
LOGT(x)	正数值表达式	以 10 为底的表达式的对数 ($\lg(x)$)。
LOST(代码)	“代码”是 1 或 2，1 可表示由于错误而损耗的能量，2 表示由于阈值而损耗的能量。	在最新 NSTR 非序列光线追迹之后损耗的能量。
LVAL(A\$)	任何字符串变量名称	字符串值。返回字符串 A\$ 的浮点值，采用当前用户语言设置来表示字符串的小数分隔符。另请参阅 SVAL。
MAGN(x,y)	x 和 y 可以是任何实数	计算 x 平方与 y 平方之和的平方根。($\sqrt{x^2 + y^2}$)
MAXF()	(空)	最大径向视场角度（以度为单位）或者径向物高或像高（以镜头单位表示）。返回值取决于视场是按照角度、物高还是像高来定义的。
MAXG()	(空)	当前载入的玻璃数量。
MCON(行,结构,数据)	从多重结构编辑器提取行（操作数编号）、结构编号和需要从多重结构编辑器中提取的数据值。	<p>提取多重结构编辑器中不同结构不同操作数对应的数据值。类似于 MCOP，但 MCOP 具有提取数据的扩展功能。</p> <p>如果行号（row）和结构编号(config)均为 0，数据（data）值为 0、1 和 2 时 MCON 分别返回操作数数量、结构数量和当前结构编号。</p> <p>如果行号（row）介于 1 和总操作数数量之间，结构编号(config)为 0，数据（data）值为 0 到 4 时，MCON 分别返回指定操作数类型，整数 1，整数 2，整数 3 和字符串标志。其中 3 个整数值分别对应不同操作数的不同参数，例如表面和波长编号。</p> <p>如果操作数数据是字符串值（如玻璃名称），则字符串标志是 1，如果操作数数据是数值，则字</p>

		<p>字符串标志 0。</p> <p>如果行号 (row) 介于 1 和总操作数数量之间, 结构编号(config)为有效的结构编号, 数据 (data) 值为 0, 则 MCON 返回对应操作数数值或字符串数据。</p> <p>需注意, 在调用 MCON 之后, MCON 返回的所有字符串数据必须通过 \$buffer 命令来提取。例如, 以下代码示例将把操作数名称存放在字符串 a\$ 中的第一行:</p> <pre>dummy = MCON(1, 0, 0) a\$ = \$buffer()</pre> <p>另请参阅关键字 SETMCOPERAND。</p>
MCOP(行,结构)	多重结构编辑器的行 (操作数编号) 和结构编号。	提取多重结构编辑器中不同结构不同操作数对应的数据值。结构编号 0 表示当前结构。另请参阅关键字 SETMCOPERAND。
MFCN()	(空)	MFCN 更新镜头数据, 使评价函数生效, 更新评价函数, 然后返回当前评价函数值。请参阅 OPER。
NCON()	(空)	返回多重结构的数量。
NFLD()	(空)	返回已定义视场的数量。
NOBJ(表面)	有效的非序列表面编号	返回非序列表面中已定义物体的数量。
NPAR(表面, 物体, 参数)	“表面”是表面编号, “物体”是物体编号, “参数”是要返回值的参数编号。	返回非序列元件编辑器中, 指定物体对应参数的值。请参阅 “SETNSCPARAMETER”。
NPOS(表面, 物体, 代码)	“表面”是表面编号, “物体”是物体编号, “代码” 1 到 6, 分别对应要返回的参考物体的 x、y、z、x 倾斜、y 倾斜和 z 倾斜。	返回非序列元件编辑器中位置列中的值。请参阅 “SETNSCPOSITION”。
NPRO(表面, 物体, 代码)	“表面”是表面编号, “物体”是物体编号, “代码” 1 到 6, 分别对应要返回的参考物体的 x、y、z、x 倾斜、y 倾斜和 z 倾斜。	返回在非序列元件编辑器中定义物体的属性页中

物体, 代码, 面)	体”是物体编号, “代码”的值为关键字 SETNSCPROPERTY 定义的值, “面”是指定物体的面编号。	的数值或字符串值。更多代码值的信息, 请参阅“ SETNSCPROPERTY ”。调用该函数返回字符串数据之后, 可用 \$buffer() 函数提取字符串值。例如, 若要从 NSC 表面 2 中提取物体 5 的注释列, 可使用以下两个命令: dummy = NPRO(2, 5, 1, 0) a\$ = \$buffer() 此时 dummy 值可以忽略, 字符串变量 a\$ 已包含了物体注释。
NSDC (表 面, 物体, 像素, 数据)	“表面”值表示非序列组件的表面编号。物体编号表示所需的探测器。“像素”表示返回指定像素的值。“数据”为 0 表示实部, 1 表示虚部, 2 表示振幅, 3 表示相干强度。 另请参阅“ NSTR ”。	如果物体编号对应的是矩形探测器, 则返回指定像素上的相干强度数据。相干强度数据对所有入射光线进行计算, 包含 4 个数字组成: 实部、虚部、幅度和相干强度。 如果像素编号为 0, 则返回探测器上所有像素数据的总和。
NSDD (表 面, 物体, 像素, 数据)	“表面”值表示非序列组件的表面编号。 物体编号表示所需的探测器。 请参阅右侧的详述内容, 了解有关变量“像素”和“数据”的详细信息。 另请参阅“ NSTR ”。	如果物体编号为 0, 则函数会清除所有探测器数据, 并且返回 0。如果物体编号小于 0, 则函数会清除该物体编号的绝对值定义的探测器数据, 并且返回 0。 如果物体编号对应于矩形探测器、表面探测器或体探测器, 则返回指定像素中的非相干强度数据。 有关变量“像素”和“数据”的完整详述, 请参阅优化操作数“ NSDD ”。该 ZPL 函数中“像素”和“数据”变量的定义与 NSDD 优化操作数相同。 与优化操作数 NSDD 不同的是, 该函数不支持忽略指定#变量值, 并且假定忽略#值始终为 0。
NSDE (表 面, 物体, 像素, 角度, 数据, 波)	“表面”值表示非序列组件的表面编号。 物体编号表示所需的探测器。	如果物体编号为 0, 则函数会清除所有探测器数据, 并且返回 0。如果物体编号小于 0, 则函数会清除该物体编号的绝对值定义的探测器数据, 并

长)	<p>请参阅右侧的详述内容，了解有关变量“像素”、“角度”、“数据”和“波长”的详细信息。</p> <p>另请参阅“NSTR”。</p>	<p>且返回 0。</p> <p>如果物体编号对应于颜色探测器，则返回指定像素上的数据。</p> <p>有关变量“像素”、“角度”、“数据”和“波长”的完整详述，请参阅优化操作“NSDD”。该 ZPL 函数中像素、角度、数据和波长变量的定义与 NSDD 优化操作数相同。</p> <p>与优化操作数 NSDD 不同的是，该函数不支持忽略指定#变量值，并且假定忽略#值始终为 0。</p>
NSDP(表面, 物体, 像素, 数据)	<p>“表面”值表示非序列组件的表面编号。</p> <p>物体编号表示所需的探测器。</p> <p>请参阅右侧的详述内容，了解有关变量“像素”和“数据”的详细信息。</p> <p>另请参阅“NSTR”。</p>	<p>如果物体编号为 0，则函数会清除所有探测器数据，并且返回 0。如果物体编号小于 0，则函数会清除该物体编号的绝对值定义的探测器数据，并且返回 0。</p> <p>如果物体编号对应于极坐标探测器，则返回指定像素上的数据。</p> <p>有关像素和数据变量的完整描述，请参阅优化操作数“NSDP”。该 ZPL 函数中“像素”和“数据”变量的定义与 NSDD 优化操作数相同。</p>
NSUR()	(空)	已定义的表面数量。
NWAV()	(空)	已定义的波长数量。
OBJC(A\$)	任何字符串变量名	含注释的物体。返回注释与字符串 A\$ 内容相同的第一个非序列物体编号，不区分大小写。如果没有 NSC 物体注释与 A\$ 相同，则返回 -1。
OCOD(A\$)	Zemax 优化操作数的任何字符串变量或文本名称。	用作 OPEV 函数的优化操作数代码编号。
ONUM(A\$)	任何字符串变量名	如果字符串 A\$ 是一个有效的优化操作数名称，如 EFFL，则 ONUM 返回操作数的 ID 号。该操作数 ID 号随后可由 SETOPERAND 使用，用于在评价函数编辑器中设置操作数类型。如果 A\$ 不对应任何优化操作数名称，则 ONUM 返回 0。

OPDC()	(空)	光程差。仅在 调用 RAYTRACE 之后有效。如果无法追迹主光线，则 OPDC 不会返回有效数据。 RAYTRACEX 不支持 OPDC 。
OPER(行, 列)	评价函数编辑器的行（操作数编号）和列（数据类型）。	<p>提取来自评价函数编辑器的任何行和列对应的数据。行与操作数编号相同；列为 1 表示类型，2 表示整数 1，3 表示整数 2，4 到 7 表示数据 1 到数据 4，8 表示目标值，9 表示权重，10 表示当前值，11 表示贡献百分比，12 和 13 分别表示数据 5 和数据 6。</p> <p>可使用列编号 2 来获取 BLNK 操作数中的注释，注释随后将置于字符串缓冲区。例如，要获取第 1 行中的注释，可用如下命令：</p> <pre>dummy = OPER(1,2) a\$ = \$buffer()</pre> <p>此时 dummy 值可以忽略，字符串变量 a\$ 已包含了第一行中的注释。</p> <p>OPER 不会更新镜头数据或评价函数，只返回当前数据。另请参阅 OPEV、MFCN 和关键字 SETOPERAND。</p>
OPEV(代 码, 整数 1, 整数 2, 数据 1, 数据 2, 数据 3, 数据 4)	“代码”是优化操作数代码（参阅函数 OCOD ），“整数 1”和“整数 2”以及数据 1 到数据 4 是操作数的定义值。详情请参阅“优化操作数”。	<p>计算的值与任何优化操作数计算的值相同，但不需要将操作数添加到评价函数中。可用于计算已通过优化操作数获得的数值。例如，要通过 EFFL 操作数来计算 EFL，请使用以下代码：</p> <pre>C = OCOD("EFFL") E = OPEV(C, 0, 1, 0, 0, 0, 0)</pre> <p>另请参阅 OPER、OCOD、MFCN 和关键字 SETOPERAND。</p> <p>一些优化操作数（如 OPGT）只有和其他操作数一起使用才有效。OPEV 仅对不取决于先前操作数的操作数有效。另请参阅下面的 OPEW。</p>
OPEW(代 码, 整数 1, 整数 2, 数据 1, 数据 2, 数据 3, 数据 4)	“代码”是优化操作数代码（参阅函数 OCOD ），“整数 1”和“整数 2”以及数据 1 到数据 4 是操作数的定义值。详情请参阅“优化操作数”。	<p>该函数与 OPEV 相似，关键区别是 OPEW 支持两个额外的变量。某些优化操作数最多使用 6 个</p>

数据 1, 数据 2, 数据 3, 数据 4, 数据 5, 数据 6)	1”和“整数 2”以及数据 1 到数据 6 是优化操作数的定义值。详情请参阅“优化操作数”。	数据值，而非 4 个。
OPTH(x)	有效的表面编号	沿光线到指定表面的光程长度。不同于 RAYT 和 RAYO，OPTH 考虑衍射表面的相位（如光栅、全息图和二元光学面）。仅在调用 RAYTRACE 之后有效。如果无法追迹主光线，则 OPTH 不会返回有效数据。RAYTRACEX 不支持 OPTH。
PARM(n,s)	有效参数编号和表面编号	表面“s”的参数“n”。
PIXX(A\$)	一个包含 BMP、JPG 或 PNG 图像文件完整路径的字符串变量。	图形中 X 轴方向的像素数量。
PIXY(A\$)	一个包含 BMP、JPG 或 PNG 图像文件完整路径的字符串变量。	图形中 Y 轴方向的像素数量。
PMOD()	(空)	如果近轴面模式关闭，则返回 0，否则返回 1。
POWR(x,y)	x 和 y 是任何数字	计算 x 的 y 次幂的绝对值。 (x^y)
PVHX()	(空)	ZPLM 优化操作数中的“数据 1”参数。
PVHY()	(空)	ZPLM 优化操作数中的“数据 2”参数。
PVPX()	(空)	ZPLM 优化操作数中的“数据 3”参数。
PVPY()	(空)	ZPLM 优化操作数中的“数据 4”参数。
PWAV()	(空)	主波长编号。
RADI(x)	有效的表面编号	指定表面的曲率半径。如果表面半径无穷大，则 RADI 返回 0.0。必须考虑这种可能性，以防止出现除数为 0 的错误。
RAGL(x)	有效的表面编号	指定表面出射光线的全局 X 轴方向余弦。仅在调

		用 RAYTRACE 或 RAYTRACEX 之后有效。
RAGM(x)	有效的表面编号	指定表面出射光线的全局 Y 轴方向余弦。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。
RAGN(x)	有效的表面编号	指定表面出射光线的全局 Z 轴方向余弦。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。
RAGX(x)	有效的表面编号	光线交点的全局 X 坐标。仅在 RAYTRACE 或 RAYTRACEX 调用之后有效。
RAGY(x)	有效的表面编号	光线交点的全局 Y 坐标。仅在 RAYTRACE 或 RAYTRACEX 调用之后有效。
RAGZ(x)	有效的表面编号	光线交点的全局 Z 坐标。仅在 RAYTRACE 或 RAYTRACEX 调用之后有效。
RAND(x)	正数值表达式	在 0 和表达式之间均匀分布的随机浮点数。一般在调用关键字 RANDOMIZE 之后使用，以改变随机数种子。如果正在使用 NSC 光线追迹操作数（如 NSTR、NSRA 等等），则需要使用关键字 RANDOMIZE 重新设定随机数生成器的种子。
RANX(x)	有效的表面编号	指定表面法线的 X 轴方向余弦。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。
RANY(x)	有效的表面编号	指定表面法线的 Y 轴方向余弦。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。
RANZ(x)	有效的表面编号	指定表面法线的 Z 轴方向余弦。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。
RAYE()	(空)	光线追迹错误标记，如果没有错误，则返回 0。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。有关详细信息，请参阅关键字 RAYTRACE。
RAYL(x)	有效的表面编号	指定表面出射光线的 X 轴方向余弦。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。

RAYM(x)	有效的表面编号	指定表面出射光线的 Y 轴方向余弦。仅在 调用 RAYTRACE 或 RAYTRACEX 之后有效。
RAYN(x)	有效的表面编号	指定表面出射光线的 Z 轴方向余弦。仅在 调用 RAYTRACE 或 RAYTRACEX 之后有效。
RAYO(x)	有效的表面编号	光线从前一个表面到指定表面的光程长度。光程长度等于路径长度乘以折射率，路径长或折射率都可能为负数。对于非序列表面内的光线，RAYO 返回光线通过的所有物体的路径长度乘以折射率的总和。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。另请参阅 OPTH 和 RAYT。
RAYT(x)	有效的表面编号	从前一个表面到指定表面的光线路径长度。光线路径长度可能是负数。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。另请参阅 OPTH 和 RAYO。对于非序列表面内的光线，RAYT 返回通过所有物体的路径长度之和。另请参阅 RAYO。
RAYV()	(空)	如果光线没有渐晕，则返回 0，否则返回渐晕表面编号。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。
RAYX(x)	有效的表面编号	光线交点的 X 坐标。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。
RAYY(x)	有效的表面编号	光线交点的 Y 坐标。仅在调用 RAYTRACE 或 RAYTRACEX 之后有效。
RAYZ(x)	有效的表面编号	光线交点的 Z 坐标。仅在 调用 RAYTRACE 或 RAYTRACEX 之后有效。
RELI(f)	视场编号	指定视场位置的相对照度 (RI)。
SAGG(x,y,z)	x、y 是编号为 z 的表面上对应的点坐标（以镜头单位表示）	计算表面上指定点的矢高（以镜头单位表示）。

SCOM(A\$, B\$)	任何两个字符串变量名称	如果字符串相同，则 SCOM 返回 0。如果 A\$ 小于 B\$，则 SCOM 返回一个负数，否则返回一个正数。
SDIA(x)	有效的表面编号	指定表面的半口径。
SIGN(x)	数值表达式	如果变量小于 0，则返回 -1，如果变量为 0，则返回 0，如果变量大于 0，则返回 +1。
SINE(x)	以弧度为单位的数值表达式	表达式的正弦。
SLEN(A\$)	任何字符串变量名称	字符串变量 A\$ 中字符的数量
SOLM(操作数, 结构, 代码)	“操作数”是多重结构操作数编号。“结构”是结构编号。“代码”为 0 表示状态，1 表示跟随解中定义的多重结构编号，2 表示跟随解中定义的操作数行，3 表示跟随解中定义的缩放因子，4 表示跟随解中定义的偏移量，5 表示操作数的当前数值或字符串值，6 表示 ZPL 宏求解中的宏名称，7 表示玻璃库名称。求解状态类型不一定使用所有代码。	返回指定多重结构操作数、指定结构对应数值的求解类型。如果“代码”为 0（表示状态），则 SOLM 返回 0，表示对应数值求解类型为固定值，返回 1 表示变量，返回 2 表示跟随，返回 3 表示热跟随，返回 4 表示 ZPL 宏，返回 5 表示玻璃替换求解类型。在调用该函数返回字符串数据之后，可使用 \$buffer() 函数提取字符串值。有关求解数据和求解参数的更多信息，请参阅“求解”。
SOLV(表面, 代码, 参数)	“表面”为表面编号。代码 0 表示曲率，1 表示厚度，2 表示玻璃，3 表示圆锥系数，4 表示半口径，5 表示 TCE（热膨胀系数）。表面参数数据的代码是 100 加上参数编号。表面附加数据的代码是 300 加上附加数据编号。“参数”是 0 到 4 之间的整数，包括 0 和 4。	返回指定表面参数求解类型中的数据。如果“参数”为 0，则返回代表求解类型的整数(返回 0 表示固定值，1 表示变量，2 表示跟随，3 表示主光线，4 表示 ZPL 宏)。如果“参数”为 1 到 3，则返回对应求解类型的相关参数。如果“参数”为 4，则返回跟随参数列。(如：跟随求解类型，参数为 1-4 时，分别返回跟随表面、缩放因子、偏移和跟随参数列)。在调用该函数返回字符串数据之后，可使用 \$buffer() 函数提取字符串值。有关求解数据和求解参数的更多信息，请参阅“求解”。

SOSO(代码)	代码为 0，返回当前执行的宏求解的表面编号；代码为 1，返回物体编号。	返回当前正在执行的求解宏中的表面或物体编号。如果当前宏中没有求解，则返回无意义的数值。
SPRO(表面, 代码)	“表面”为表面编号。“代码”为关键字 SETSURFACEPROPERTY 中定义的代码值。仅支持整数代码值，不支持助记符。	返回镜头数据编辑器中定义的表面属性页或编辑器的数值或字符串值。有关“代码”值的信息，请参阅“SETSURFACEPROPERTY、SURP”。在调用该函数返回字符串数据之后，可使用 \$buffer() 函数提取字符串值。例如，要提取表面 4 的注释列，请使用以下两个命令： dummy = SPRO(4, 1) a\$ = \$buffer() 此时 dummy 值可以忽略，字符串变量 a\$ 已包含了表面 4 的注释。SPRO 不能使用 SETSURFACEPROPERTY 中“数据 2”变量定义的属性值，若需要，请使用 SPRX。”数据 2”值的属性包括参数数据、附加数据、膜层缩放、折射率偏移值和状态。
SPRX(表面, 代码, 数据 2)	“表面”为表面编号。“代码”和“数据 2”的值为关键字 SETSURFACEPROPERTY 中定义的值。仅支持整数代码值，不支持助记符。	SPRX 类似于 SPRO，区别在于 SPRX 具有一个额外的变量。通过 SETSURFACEPROPERTY “数据 2”变量设置的参数才需要使用 SPRX。请参阅“SETSURFACEPROPERTY、SURP”
SQRT(x)	正数值表达式	表达式的平方根。 $(\sqrt{x^2})$
STDD() - 废弃	此函数已废弃，请改为使用 SRPO。	
SURC(A\$)	任何字符串变量名称	含注释的表面。返回注释与字符串 A\$ 匹配的的第一个表面编号。字符不分大小写。如果没有任何表面注释与 A\$匹配，则返回 -1。
SVAL(A\$)	任何字符串变量名称	字符串值。返回字符串 A\$ 的浮点值，假定句点(.)是字符串中的小数点分隔符。另请参阅

		LVAL。
SYPR(代码)	“代码”值为关键字 SETSYSTEMPROPERTY 中定义的代码值。仅支持整数代码值，不支持助记符。	<p>返回相关系统数据的数值或字符串值。有关“代码”值的信息，请参阅“SETSYSTEMPROPERTY、SYSP”。在调用该函数返回字符串数据之后，可使用 \$buffer() 函数提取字符串值。例如，要提取镜头标题，请使用以下两个命令：</p> <pre>dummy = SYPR(16) a\$ = \$buffer()</pre> <p>此时 dummy 值可以忽略，字符串变量 a\$ 已包含了镜头标题。SYPR 不能使用 SETSURFACEPROPERTY 中“数据 2”变量定义的属性值。</p>
TANG(x)	以弧度为单位的数值表达式	表达式的正切。
函数	变量	返回值
TMAS()	(空)	从表面 1 到像面中透镜的总质量（以克为单位）
THIC(x)	有效的表面编号	指定表面的厚度。
TOLV(操作数, 列)	整数“操作数”是公差操作数编号，“列”是公差数据编辑器中的列编号。请参阅右侧详细描述。	<p>返回公差数据编辑器中的数值或字符串值。如果“列”是 1、2 或 3，则返回该操作数类型使用的整数 1、整数 2 或整数 3。如果“列”是 4 或 5，则返回公差的最小值或最大值。如果“列”为 6，则返回操作数属性页中的“在反向公差时不调整”标志（1 表示勾选，0 表示未勾选）。如果“列”为 90，则返回对应公差操作数的当前值。如果“列”值为 0 或 99，则返回操作数名称或操作数注释，并将其存放于缓冲字符串中。例如，要提取操作数 7 的操作数名称，请使用以下两个命令：</p> <pre>dummy = TOLV(7, 0) a\$ = \$buffer()</pre> <p>此时 dummy 值可以忽略，字符串变量 a\$ 已包含</p>

		了操作数名称。另请参阅执行类似功能的字符串函数 \$TOLOPERAND 和 \$TOLCOMMENT。
UNIT()	(空)	如果当前单位类型分别是毫米、厘米、英寸或米，则分别返回 0、1、2 或 3。
VEC1(x)	正下标值	返回数组变量中指定下标对应的值。
VEC2(x)	正下标值	返回数组变量中指定下标对应的值。
VEC3(x)	正下标值	返回数组变量中指定下标对应的值。
VEC4(x)	正下标值	返回数组变量中指定下标对应的值。
VERS()	(空)	返回 Zemax 的版本。版本的格式为 yymmdd。例如，2005 年 6 月 15 日发行的版本为 050615。
WAVL(x)	有效的波长编号	以微米为单位的波长。
WINL()	(空)	返回由关键字 OPENANALYSISWINDOW 最近打开的分析窗口的编号。
WINN()	(空)	返回所有打开分析窗口的编号。
WWGT(x)	有效的波长编号	波长权重。
XMIN()	(空)	图形窗口中的最小 X 坐标。
XMAX()	(空)	图形窗口中的最大 X 坐标。
函数	变量	返回值
YMIN()	(空)	图形窗口中的最小 Y 坐标。
YMAX()	(空)	图形窗口中的最大 Y 坐标。

使用 FICL() 函数

FICL 函数计算光纤耦合。FICL 函数使用非常多的变量，可先将这些变量放置在一个矢量数组中，

然后再调用此函数。矢量数组必须是已经定义四个 VEC 数组之一。矢量数组中存储值的定义方式如下：

- 0 = 采样 (1 表示 32 x 32 · 2 表示 64 x 32 · 以此类推)
- 1 = 波长#
- 2 = 视场#
- 3 = 忽略光源标志 (0 表示 假 · 1 表示 真)
- 4 = 源光纤 x 方向数值孔径 (NA)
- 5 = 接收光纤 x 方向数值孔径 (NA)
- 6 = 源光纤 x 角度 (度)
- 7 = 源光纤 y 角度 (度)
- 8 = 接收光纤沿 x 倾斜 (度)
- 9 = 接收光纤沿 y 倾斜 (度)
- 10 = 接收光纤 x 偏心
- 11 = 接收光纤 y 偏心
- 12 = 接收光纤 z 偏心
- 13 =源光纤对准主光线标志 (0 表示假 · 1 表示真)
- 14 =接收光纤对准主光线标志 (0 表示假 · 1 表示真)
- 15 = 使用偏振标志 (0 表示假 · 1 表示真)
- 16 = 源光纤 y 方向数值孔径 (NA) (如果为 0 · 则 NAy = NAx)
- 17 = 接收光纤 y 方向数值孔径 (NA) (如果为 0 · 则 NAy = NAx)
- 18 = 使用惠更斯衍射积分标志 (0 表示假 · 1 表示真)

光纤耦合通过调用 FICL(n) 来计算，其中 n 是包含变量列表的矢量编号。

字符串函数

下表列出了可用的字符串函数。

ZPL 字符串函数

函数	描述
----	----

\$BUFFER()	返回镜头缓冲区中的当前字符串。可从各种 ZPL 关键字和函数中提取字符串数据。
\$CALLSTR(i)	返回 关键字 CALLMACRO 的字符串缓冲区中位于索引 i 的字符串。请参阅“从宏中调用宏”。
\$COAT(i)	返回第 i 个表面的膜层名称。
\$COATINGPATH()	返回膜层文件的路径名称。
\$COMMENT(i)	返回第 i 个表面的注释字符串。
\$DATAPATH()	返回数据文件的路径名称。
\$DATE()	返回当前日期和时间字符串。日期和时间的格式由“配置选项”的“常规”栏目中的“日期/时间”设置来定义的。
\$EXTENSIONPATH()	返回 Zemax 扩展的路径名称。
\$FILENAME()	返回当前镜头文件名，但不包含路径。
\$FILEPATH()	返回当前镜头文件名，包含完整路径。
\$GETSTRING(A\$, n)	返回字符串 A\$ 中的第 n 个子串，字符串 A\$ 中使用空格作为分隔符。例如，如果 A\$ = “one two three”，则 \$GETSTRING(A\$, 2) 返回 “two”。
\$GETSTRINGC(A\$, n)	返回字符串 A\$ 的第 n 个子串，字符串 A\$ 中使用逗号作为分隔符。例如，如果 A\$ = “one,two,three”，则 \$GETSTRING(A\$, 2) 返回 “two”。
\$GLASS(i)	返回表面编号为 i 的玻璃名称。
\$GLASSCATALOG(i)	返回当前镜头的第 i 个载入的玻璃库名称。如果 i 小于 1，则返回包含已载入所有玻璃库名称的单个字符串，以空格分隔。
\$GLASSPATH()	返回玻璃库文件的路径名称。
\$LEFTSTRING(A\$, n)	返回字符串 A\$ 中最左侧的 n 个字符。如果 A\$ 中的字符数少于 n 个，则剩余的字符位置用空格填充。这样可以定义固定长度的字符

	串。
\$LENSNAME()	返回系统选项中定义的镜头标题。
\$MACROPATH()	返回宏文件的路径名称。
\$NOTE(line#)	<p>返回系统选项中定义的注释信息。由于注释可能很长，因此 \$NOTE 以行为单位返回注释中的字符。在出现换行符（回车）或者当一行中连续字符总数超过 100 时，一行结束，以首先满足条件者为准。 line# 表示将返回注释中的哪一行。</p> <p>例如，\$NOTE(1) 返回注释中的第一行（直至第一个换行符）。如果找到第一个换行符之前连续字符数超过 100 个，则第 1 行为前 100 个字符，第 2 行为截止到第一个换行符的其余字符，或者是接下来的 100 个字符，以首先满足条件者为准。</p> <p>注释中的最大字符数量前限制为 4000（但由于空格限制，在系统选项的“标题/注释”栏目中可能无法编辑如此多的字符）。如果指定行的注释中没有任何已定义的字符，则 \$NOTE 返回空字符串。</p> <p>函数 SLEN 可用于确定 \$NOTE 返回的实际字符数，使用的语法如下：</p> <p>A\$ = \$NOTE(1) N = SLEN(A\$)</p> <p>数据 N 为字符串 A\$ 中的字符数（整数）。</p>
\$OBJECTPATH()	返回 NSC 物体文件的路径名称。
\$PATHNAME()	仅返回当前镜头文件的路径名称。这可用于确定存储当前镜头文件的文件夹。
\$PROGRAMPATH()	返回程序文件的路径名称。
\$QUOTE()	返回双引号字符 (")。
\$RIGHTSTRING(A\$, n)	返回字符串 A\$ 中最右侧的 n 个字符。如果 A\$ 中的字符数少于 n 个，则剩余的字符位置用空格填充。这样可以定义固定长度的字符串。
\$STR(表达式)	返回一个格式由关键字 FORMAT 定义的字符串，数值表达式可以是任何等式，包括常量、变量和函数的组合。请参阅用于将字符串转换为数值的函数 SVAL(A\$) 。

\$TAB()	返回制表符 (\\t)。
\$TEMPFILENAME()	返回临时文件的名称，包含完整路径，适合于临时存储文本或二进制数据。请参阅关键字 GETTEXTFILE。
\$TOLCOMMENT(操作数)	返回指定公差操作数的注释。
\$TOLOPERAND(操作数)	返回指定公差操作数的操作数名称。
\$UNITS()	根据当前镜头单位，返回 MM、CM、IN 或 M。

关键字

关键字可以为程序流发出指令、生成输出以及执行某些关键任务，比如光线追迹和修改镜头规格。
以下各节中详细描述了每个关键字的功能。

APMN、APMX、APTP、APXD、APYD

这些命令已被废弃。请参阅“SETSURFACEPROPERTY、SURP”。

ATYP、AVAL

这些命令已被废弃。请参阅“SETSYSTEMPROPERTY、SYSP”。

BEEP

发出蜂鸣声。

语法：

BEEP

详述：

此命令可用于在计算完成或需要输入时提醒用户。

BROWSE

显示一个窗口，提示用户选择或输入文件名，并将文件路径存放在一个字符串变量中。

语法:

```
BROWSE "提示", 变量, 模式  
BROWSE "提示", 变量, 模式, "扩展"
```

详述:

变量可以是任何有效的字符串变量名，模式变量为一个整数，可以是 0 或 1。当模式变量为 0 时，表示“加载模式”，提示用户输入一个现有文件；当模式变量为 1 时，表示“保存模式”，提示用户输入另存文件名。字符串“扩展”可以用来过滤其他格式文件（如“.CFG”）。

需注意：**BROWSE** 函数允许用户选择一个文件并将文件路径存放在一个字符串变量中，但不会自动打开或保存文件。

例如:

```
! Load  
BROWSE "File:", A$, 0, ".txt"  
OPEN A$  
PRINT "Reading file:"  
PRINT A$ Zemax, LLC OpticStudio 17 Help Files The Programming Tab • 2076  
  
READ x  
READ y  
READ z  
CLOSE  
! Save  
BROWSE "Save Merit Function as: ", B$, 1, ".mf"  
SAVEMERIT B$  
PRINT "Saving file as: ", B$
```

相关关键字: INPUT

CALLMACRO

调用另一个 ZPL 宏。

语法:

```
CALLMACRO filename.zpl  
CALLMACRO name$
```

详述:

此命令用于调用另一个 ZPL 宏。文件名是不包含路径的完整宏名称，且文件必须存放在 ZPL 宏文件的文件夹中。当 CALLMACR 宏返回时，被调用宏的任何文本输出都将复制到 CALLMACR 宏的文本输出窗口。请参阅“从宏中调用宏”。

CALLSETDBL

在主宏的缓冲区中设置双精度数值。

语法：

CALLSETDBL 索引, 数值

详述：

此命令用于在主宏的缓冲区中设置数值，索引可为 0 到 50 之间的任何值。可以使用 CALD 函数来检索数值。请参阅“从宏中调用宏”。

CALLSETSTR

在主宏的缓冲区中设置字符串值。

语法：

CALLSETSTR 索引, 字符串文本

详述：

此命令用于在主宏的缓冲区中设置字符串值。索引可为 0 到 50 之间的任何值。随后可以使用 \$CALLSTRING 函数来检索字符串值。请参阅“从宏中调用宏”。

COAT

该命令已被废弃。请参阅“SETSURFACEPROPERTY、SURP”。

CLOSE

关闭之前由 OPEN 命令打开的文件。

语法：

CLOSE

详述：

请参阅 OPEN 的描述。

CLOSEWINDOW

关闭默认输出窗口。此关键字还可用于关闭任何已打开的分析窗口。

语法：

```
CLOSEWINDOW  
CLOSEWINDOW n
```

详述：

CLOSEWINDOW (无变量 “n”) 以安静模式执行 ZPL 宏。如果在宏的任何行中包含了关键字 CLOSEWINDOW，则宏执行结束后，不会显示文本输出窗口。CLOSEWINDOW 对宏执行没有其它影响。

CLOSEWINDOW (有变量 “n”) 将关闭编号 n 的分析窗口。

COLOR

设置用于图形线条和文本函数的线条颜色。

语法：

```
COLOR n
```

详述：

n 必须是对应于所需颜色的整数。图形模式中所有后续线条和文本命令都将使用新的线条颜色。颜色 0 表示黑色，其它颜色的定义请参阅“颜色 1-12，颜色 13-24”。除黑色以外，还有 24 种颜色可用。

COMMAND

执行 shell 命令。

语法：

```
COMMAND 可执行文件, 变量
```

详述：

“可执行文件”和“变量”可以是字符串或字符串变量。“可执行文件”为即将运行的宏文件的路径名称。“变量”是可选的，并且包含“可执行文件”中的任何命令行。

一旦命令启动，可执行文件便会在自身的线程中运行。ZPL 宏或 Zemax 不需要等待可执行命令结束任务。COMMAND 不会返回任何错误代码，并且 Zemax 无法知晓 shell 命令是否成功启动。

COMMENT

该命令已被废弃。请参阅“SETSURFACEPROPERTY、SURP”。

CONI

该命令已被废弃。请参阅“SETSURFACEPROPERTY、SURP”。

CONVERTFILEFORMAT

将文本文件从 ANSI 格式转换为 Unicode 格式，或者从 Unicode 转换为 ANSI 格式。

语法：

CONVERTFILEFORMAT 文件名, 编码

详述：

“文件名”是需要转换的文本文件名称。“编码”为 1 表示将文件从 Unicode 格式转换为 ANSI 格式，“编码”为 2 表示从 ANSI 格式转换为 Unicode 格式。需注意，如果 Unicode 字符没有对应的 ANSI 字符，那么将 Unicode 文件转换为 ANSI 格式时可能会导致数据丢失。

CONVERTIMAGETOGRID

CONVERTIMAGETOGRID 将单色图像文件（.BMP、.JPG、.TIFF、.PNG 等）转换为可用于网格相位面型的 .DAT 文件。对于 RGB 图像，在转换中使用 R 通道，即在宏中执行“将 BMP 图像转换为 OpticStudio DAT 格式”命令。

语法：

CONVERTIMAGETOGRID 文件名, 增量 x , 增量 y , 单位标记, 偏移 x , 偏移 y

详述：

“文件名”是需要转换文件的完整路径名称。“增量x”和“增量y”表示在指定单位下的像元宽度和高度。“单位标记”为0表示毫米，1表示厘米，2表示英寸，3表示米。“偏移x”和“偏移y”可用于设置图像偏心。

示例:

```
A$ = "C:\pictures\opticstudio.bmp"  
CONVERTIMAGETOGRID A$, 0.1, 0.1, 0, 0, 0
```

COPYFILE

COPYFILE 表示复制文件。

语法:

```
COPYFILE 源文件名, 新文件名
```

详述:

该关键字需要两个文件名，文件名为引号中的字符串表达式或字符串变量。将“源文件名”对应的源文件复制到“新文件名”对应的新文件。如果“新文件名”已经存在，则系统会在不发出警告的情况下覆盖此文件。

示例:

```
COPYFILE "C:\source.dat", "C:\copy.dat"
```

相关关键字:

DELETEFILE

RENAMEFILE

CURV

该命令已被废弃。请参阅“SETSURFACEPROPERTY、SURP”。

DECLARE

请参阅“数组变量”。

DEFAULTMERIT

生成默认评价函数。

语法：

DEFAULTMERIT 类型，数据，参考，方法，环，臂，网格，删除，轴向，径向，起始，x 权重，总权重，光瞳遮光比

详述：

此关键字可在评价函数编辑器中生成默认评价函数。将删除现有的默认评价函数。有关更多详情，请参阅“修改评价函数”（“Modifying the merit function”）。相关变量定义如下所示：

类型：0 表示 RMS，1 表示 PTV。

数据：0 表示波前，1 表示光斑半径，2 表示光斑 X，3 表示光斑 Y，4 表示光斑 $x + y$ 。

参考：0 表示质心，1 表示主光线，2 表示无参考。

方法：1 表示高斯积分，2 表示矩形阵列。

环：环的数量（仅限高斯积分）。

臂：臂的数量（仅限高斯积分）。臂的数量必须是偶数，并且不得小于 6。

网格：网格尺寸。用一个整数表示，比如 8 表示 8×8 网格。n 必须是偶数且不小于 4。

删除：0 表示不删除渐晕光线，1 表示删除渐晕光线。

轴向：-1 表示自动，当系统为轴对称系统时才使用自动对称。1 表示假设轴对称，0 表示不假设轴对称。

径向：1 表示忽略垂轴色差，否则为 0。

起始：定义添加默认评价函数的起始行。-1 表示自动，将会在现有 DMFS 操作数之后添加默认评价函数。也可自定义起始操作数编号，起始行之前的操作数都将保留。

x 权重, 总权重：评价函数的 x 方向权重和总权重。只有当“数据”变量为“spot $x + y$ ”时才使用 x 权重值。

pup_obsc：光瞳遮阑比例。

DELETE

从镜头数据编辑器表格中删除某个表面。

语法：

```
DELETE n
```

详述：

n 必须为整数结果的表达式。另请参阅 INSERT。

示例：

```
DELETE 5  
DELETE i+2*j
```

DELETECONFIG

DELETECONFIG 删除多重结构编辑器中的现有结构。

语法：

```
DELETECONFIG 结构
```

详述：

“结构”必须为整数结果的表达式，且必须大于 0 小于等于总多重结构数量。另请参阅 INSERTCONFIG 和 DELETENCO。

DELETEFILE

DELETEFILE 用于删除文件。

语法：

```
DELETEFILE 文件名
```

详述：

该关键字需要一个文件名，文件名定义为引号中的字符串表达式或字符串变量。

示例：

```
DELETEFILE XFILE$
```

相关关键字：

COPYFILE RENAMEFILE

DELETEMCO

DELETEMCO 删除多重结构编辑器中的现有操作数。

语法：

```
DELETEMCO 行
```

详述：

“行”必须为整数结果的表达式，且必须大于 0 小于等于必当前多重结构编辑器中操作数的总数量。另请参阅 INSERTMCO 和 DELETECONFIG。

DELETEMFO

DELETEMFO 删除评价函数编辑器中的现有操作数。

语法：

```
DELETEMFO 行  
DELETEMFO ALL
```

详述：

“行”必须为整数结果的表达式，且必须大于 0 小于等于必当前评价函数编辑器中操作数的总数量。如果使用了变量 ALL，则将删除评价函数中所有的操作数。另请参阅 INSERTMFO。

DELETEOBJECT

删除现有 NSC 物体。

语法：

```
DELETEOBJECT 表面, 物体
```

详述：

“表面”必须为整数结果的表达式，并且指定的表面必须为非序列组件表面。如果 ZEMAX 当前的程序模式为非序列，则“表面”值 设为 1。“物体”的值必须为整数，且大于等于 1 小于等于当前物体数加 1。DELETEOBJECT 将删除指定物体，并根据需要对其它物体重新进行编号。另请参阅 INSERTOBJECT 和 SETNSCPROPERTY。

示例：

```
DELETEOBJECT 1, 1
```

DELETETOL

DELETETOL 删除公差数据编辑器中的现有操作数。

语法：

```
DELETETOL 行
```

详述：

“行”必须为整数结果的表达式，且必须大于 0 小于等于必当前公差数据编辑器中操作数的总数量。另请参阅 INSERTTOL 和 SETTOL。

EDVA

该命令已被废弃。请参阅“SETSURFACEPROPERTY、SURP”。

END

请参阅 GOSUB。

EXPORTBMP

将图形窗口导出为 BMP 文件。

语法：

```
EXPORTBMP 窗口编号, 文件名, 延迟
```

详述：

“窗口编号”可为整数或整数结果的表达式。表示需要保存到文件的图形窗口编号。Zemax 以窗口打开的顺序对窗口进行编号，从 1 开始。若用户关闭某些窗口，则 ZEMAX 会从列表中删除已关闭的窗口，但不会对余下的窗口重新编号。重新打开的任何窗口将会使用最小的可用窗口编号。

“文件名”是包含路径的完整文件名称，不含扩展名。Zemax 会自动添加扩展名 BMP。“延迟”

变量是可选的，表示指定延迟时间（以毫秒为单位）。对于某些复杂图形，导出图形或截屏的完成需要一定的延迟。如果导出的 BMP 文件看上去不完整，请尝试将“延迟”值设置为 500 到 2500 毫秒。另请参阅 EXPORTJPG。

示例：

```
EXPORTBMP 1, "C:\TEMP\MYBMPPFILE"  
EXPORTBMP k, A$
```

EXPORTCAD

将镜头数据导出为 IGES、STEP、SAT 或 STL 格式文件，以导入到 CAD 软件中。

语法：

```
EXPORTCAD 文件名
```

详述：

有关 CAD 文件导出的完整描述，请参阅导出“CAD 文件”，位于“文件菜单>导出组>CAD 文件”。

“文件名”可以是字符串（比如“C:\DATA\FILE.IGS”）或者是字符串变量（比如 MYFILENAME\$）。

导出 CAD 文件需要定义其它参数数据。这些数据存放在 VEC1 变量中，如下所示：

VEC1(1)：文件类型。用 0 表示 IGES，1 表示 STEP，2 表示 SAT，3 表示 STL。

VEC1(2)：样条点的数量（某些实体类型需要设置）。设为 16、32、64、128、256 或 512。

VEC1(3)：要导出的第一个表面。在 NSC 模式下，表示要导出的第一个物体。

VEC1(4)：要导出的最后一个表面。在 NSC 模式下，表示要导出的最后一个物体。

VEC1(5)：放置光线数据的图层。

VEC1(6)：放置镜头数据的图层。

VEC1(7)：使用 1 表示可导出虚拟表面，否则使用 0。VEC1(8)：使用 1 表示可作为实体导出表面，否则使用 0。

VEC1(9)：光线模式。0 表示 XY，1 表示 X，2 表示 Y，3 表示环，4 表示列表，5 表示无模式，6 表示网格，7 表示实体光束。

VEC1(10)：光线数量。

VEC1(11)：波长编号。0 表示导出全部波长光线。

VEC1(12)：视场编号。0 表示导出全部视场光线。

VEC1(13)：使用 1 表示删除渐晕光线，否则使用 0。VEC1(14)：虚表面厚度，以镜头单位表示。

VEC1(15)：使用 1 表示导出 NSC 分裂光线，否则使用 0。VEC1(16)：使用 1 表示导出 NSC 散射光线，否则使用 0。

VEC1(17)：使用 1 表示追迹 NSC 光线时使用偏振，否则使用 0。如果已选择光线分裂，则自动选择偏振。

VEC1(18)：0 表示当前结构组态，1 到 n 表示指定结构组态（其中 n 是多重结构的总数量），n + 1 表示“ All By File”，n + 2 表示“ All By Layer”，n + 3 表示“ All At Once”。有关导出设置的更详情，请参阅“导出 CAD 文件”。

VEC1(19)：公差设置。0 表示 1.0E-4, 1 表示 1.0E-5, 2 表示 1.0E-6, 3 表示 1.0E-7。需注意：如果输入其他值，公差设置默认为 1.0E-4。

VEC1(20)、VEC1(21)：在混合序列/非序列模式下（即包含一个非序列组件表面），VEC1(20)、VEC1(21)的值表示指定导出非序列物体的范围。VEC1(20)表示要导出的第一个物体，VEC1(21)表示要导出的最后一个物体。如果这两个值均为 0 或者超出范围，则导出非序列组件表面中包含的所有物体。

示例：

```
VEC1(1) = 0
VEC1(2) = 32
VEC1(3) = 1
etc...
VEC1(18) = 0
EXPORTCAD "C:\TEMP\MYCADFILE.IGS"
```

EXPORTJPG

将任何图形窗口导出为 JPG 文件。

语法：

```
EXPORTJPG 窗口编号, 文件名, 延迟
```

详述：

请参阅 EXPORTBMP。

示例：

```
EXPORTJPG 3, "C:\TEMP\MYJPGFILE"
```

```
EXPORTJPG k, FILENAME$, 500
```

FINDFILE

用于查找文件名称。

语法：

```
FINDFILE TEMPNAME$, FILTER$
```

详述：

此关键字需要两个字符串变量，一个表示将当前文件名存储在此字符串变量中，另一个表示“过滤”字符串。过滤字符串指定所需文件类型的路径名称和通配符。请参阅以下示例。**FINDFILE** 用于列出某个文件夹中指定格式的所有文件，或者用于分析大量相似的镜头文件。在列出所有文件之后，如果需要将 **FINDFILE** 重置回列表中的第一个文件，则可以使用其他“过滤”字符串调用一次 **FINDFILE**，然后再使用之前的“过滤”字符串重新调用 **FINDFILE**。每次使用新的“过滤”字符串调用 **FINDFILE** 时，会返回文件夹中指定类型的第一个文件。

示例：

```
FILTER$ = "C:\Zemax\*.ZMX"
PRINT "Listing of all Zemax files in ", FILTER$
FINDFILE TEMPFILE$, FILTER$
LABEL 1
if (SLEN(TEMPFILE$))
    PRINT TEMPFILE$
    FINDFILE TEMPFILE$, FILTER$
    GOTO 1
ENDIF
PRINT "No more files."
```

FLDX, FLDY, FWGT, FVDX, FVDY, FVCX, FVCY, FVAN

这些命令已被废弃。请参阅“SETSYSTEMPROPERTY、SYSP”。

FOR, NEXT

关键字 **FOR** 和 **NEXT** 定义了一个循环语句，指定循环中执行的次数。

语法：

```
FOR 变量, 起始值, 终止值, 增量...
NEXT
```

详述：

关键字 **FOR** 标志着循环命令的开始。**FOR** 需要指定一个充当计数器的变量（不需要是整数）、变量的起始值、终止值和增量。关键字 **NEXT** 标志着循环命令的结束。**FOR-NEXT** 循环可以嵌套。**FOR** 命令和 **NEXT** 命令的数量必须相同。

FOR 命令一旦执行，其起始值、终止值和增量便由对应表达式计算出来并保存。即使在后续程序中，对应表达式中的某些变量发生了变化，也不会重新计算起始值、终止值和增量。只有在 **FOR** 循环开始时定义的值才有效。

如果起始值和终止值相同，则循环仅执行一次。如果起始值小于终止值，则循环继续，直到变量大于终止值。如果起始值大于终止值，则循环继续，直到变量小于终止值。

示例：

```
FOR i, 1, 10, 1
    a = 0.2*i
    PRINT a
NEXT

j = 5 k = 0
FOR i, j, j + 5, 2
    k = i + j + k
NEXT
```

FORMAT

指定后续 **PRINT** 和 **\$STR** 命令的数值精度格式。

语法：

```
FORMAT m.n
FORMAT m.n EXP
FORMAT m [INT]
FORMAT "C_format_string" [LIT]
```

详述：

整数 **m** 和 **n** 由小数点分隔。**m** 和 **n** 的值必须为明确值，不能使用变量。**m** 表示要输出的字符总数，包含空格符。**n** 表示小数点后显示的位数。例如，**FORMAT 8.4** 表示后续 **PRINT** 命令输出 8 个字符，并显示 4 位小数。**FORMAT .5** 表示后续 **PRINT** 根据需要输出字符数，并显示 5 位小数。**FORMAT** 仅影响 **PRINT** 的数值输出。如果某个数值太大，超过指定 **m** 位数，那么后续 **PRINT** 会忽略 **FORMAT**

命令的 **m** 部分。

m.n 之后的可选关键字 **EXP** 表示使用指数计数法。

可选关键字 **INT** 表示数据以整数格式输出，且显示为指定 **m** 位数。

可选关键字 **LIT** (文字) 表示根据指定 **C** 语言格式输出数据。可在 **C** 语言的任何编程参考中找到 **C** 格式规范。

示例：

宏：

```
X = 5
FORMAT 5.3
PRINT "FORMAT 5.3      :", X
FORMAT 12.2 EXP
PRINT "FORMAT 12.2 EXP :", X
FORMAT 6 INT
PRINT "FORMAT 6 INT   :", X
FORMAT "%#06i" LIT
PRINT "FORMAT %#06i LIT :", X
```

输出：

```
FORMAT 5.3      :5.000
FORMAT 12.2 EXP :5.00E+000
FORMAT 6 INT    :5
FORMAT %#06i LIT :000005
```

FTYP

该命令已被废弃。请参阅“**SETSYSTEMPROPERTY**、**SYSP**”。

GCRS

该命令已被废弃。请参阅“**SETSYSTEMPROPERTY**、**SYSP**”。

GDATE

GDATE 表示用户自定义图形窗口时，在文字框中镜头名称下面显示当前日期。

语法：

```
GDATE
```


详述：

GDATE 主要用于使 ZPL 宏生成的图形类似于其它 Zemax 图形。

示例：

请参阅“图形”。

GETEXTRADATA

获取镜头数据编辑器中载入的附加数据值。并且将附加数据存储在某一个矢量数组变量中（VEC1、VEC2、VEC3 或 VEC4 中的任何一个）。

语法：

GETEXTRADATA 矢量表达式，表面表达式

详述：

数据存储在某指定的 VECn 数组变量中。例如，GETEXTRADATA 1, 5 表示表面 5 的附加数据存储于 VEC1 中。数据用以下格式存储，其中每行的第一个数字表示数组中的位置：

0：数组变量中数据的数量

1：第一个附加数据值

n：第 n 个附加数据值

有关附加数据值的描述，请参阅“附加数据”。

GETGLASSDATA

获取当前玻璃库中任何玻璃的数据。并且将玻璃数据存储在某一个矢量数组变量中（VEC1、VEC2、VEC3 或 VEC4 中的任何一个）。

语法：

GETGLASSDATA 矢量表达式，玻璃编号

详述：

数据存储在某指定的 VECn 数组变量中。例如，GETGLASSDATA 1, 32 表示将玻璃编号 32 的数据存储于 VEC1 中。GNUM 函数表示返回玻璃编号。数据用以下格式存储，其中每行的第一个数字表示数组中的位置：

0：矢量数组中数据的数量

1：玻璃的色散系数求解公式编号，如下所示：

1 = Schott · 2 = Sellmeier 1 · 3 = Herzberger · 4 = Sellmeier 2

5 = Conrady · 6 = Sellmeier 3 · 7 = Handbook 1 · 8 = Handbook 2

9 = Sellmeier 4 · 10 = Extended · 11 = Sellmeier 5 · 12 = Extended 2

(ZEMAX 玻璃库选项卡中色散公式下拉菜单中还有一个 Extended 3，是不是得有个 13= Extended 3)

2：参考温度，以摄氏度为单位

3：折射率 Nd

4：阿贝数 vd

5：热膨胀系数：-30 到 +70 c

6：热膨胀系数：+20 到 300 c

7：密度：g/cm³

8：局部色散相对于法线的偏差值 DPgF

9：波长 λ 最小值

10：波长 λ 最大值

11-16：色散系数 A0-A5 (取决于公式)

17-22：热分析模型中所采用的热系数 (热色散系数)

23-(22 + #waves)：内透射系数 α (/mm) 的值， $T=\exp(\alpha \cdot \text{path})$ 。波长 1 的 α 存储在 23 中，波长 2 的 α 存储在 24 中，以此类推，直到系统使用的波长数量。

(23 + #waves) - (32 + #waves)：色散系数 A0-A9 (取决于公式)

相关函数：

GNUM

GETLSF

计算几何边缘和线响应函数，类似于“几何线/边缘扩散”函数。

语法：

GETLSF 波长，视场，采样，矢量，最大半径，使用偏振

详述：

“波长”指定用于计算的波长编号，“波长”为 0 时表示多色计算。“视场”必须为 1 到最大视场数之间的整数，表示要计算的视场编号。“采样”可为 1 (32 x 32)、2 (64 x 64)、3 (128 x 128) 等... 最大为 2048 x 2048。“矢量”为 1 到 4 之间的整数，指定数据存储的矢量数组(VEC1,VEC2,VEC3 或

VEC4)。“最大半径”表示边缘和线扩散函数的最大径向坐标。“最大半径”的值给出的是整个数据范围的半宽，0 表示默认半宽。“使用偏振”变量为 1 表示使用偏振，0 表示未使用偏振。如果给定的任何变量超出有效范围，则会使用最近的值替代。

返回的数据存储在指定的矢量数组变量中。矢量位置 0 到 3 分别表示点“N”的数量、起始 x 坐标（半宽的负数）、坐标间隔和偏移量。偏移量表示矢量中第一个边缘或线扩散数据。从偏移量开始，第一个 N 值是子午 LSF 响应。第二个 N 值是弧矢 LSF 响应，第三个 N 值是子午 ERF 值，第四个 N 值是弧矢 ERF 值。

如果当前矢量大小不足，Zemax 将自动增加矢量的大小，以按照 SETVECSIZE 中描述的方式保存 LSF 数据。

示例：

```
!Macro computes and prints the LSF and ERF for polychromatic light at field 1.
!
!Syntax is GETLSF wave, field, samp, vector, maxradius, usepol
!
GETLSF 0, 1, 3, 1, 0, 0
N_BINS = vec1(0)
STARTX = vec1(1)
DELTAX = vec1(2)
OFFSET = vec1(3)
FORMAT 15.0
PRINT "Number of Bins    = ", N_BINS
FORMAT 15.3 EXP
PRINT "Starting Coordinate = ", STARTX
PRINT "Delta Coordinate  = ", DELTAX
FORMAT 15.0
PRINT "Offset            = ", OFFSET
OFF1 = OFFSET
OFF2 = OFF1 + N_BINS
OFF3 = OFF2 + N_BINS
OFF4 = OFF3 + N_BINS
MAXI = N_BINS-1
FORMAT 16.3 EXP
PRINT
PRINT "          X          TLSF          SLSF          TERF          SERF"
PRINT
FOR i, 0, MAXI, 1
    PRINT STARTX + DELTAX*i,
    PRINT vec1(OFF1 + i),
    PRINT vec1(OFF2 + i),
    PRINT vec1(OFF3 + i),
    PRINT vec1(OFF4 + i)
NEXT i
```

GETMTF

计算当前镜头文件的子午和弧矢 MTF、实部、虚部、相位或方波响应数据，并将数据存储于四个矢量数组之一（VEC1、VEC2、VEC3 或 VEC4）。

语法：

GETMTF 频率，波长，视场，采样，矢量，类型

详述：

“频率”是以 MTF 单位表示的空间频率(参阅“MTF 单位”)。如果频率小于 0 或者大于截止频率，则 GETMTF 返回 0。“波长”指定用于计算的波长编号，“波长”为 0 时表示多色计算。“视场”必须为 1 到最大视场数之间的整数，表示要计算的视场编号。“采样”可为 1 (32 x 32)、2 (64 x 64)、3 (128 x 128) 等... 最大为 2048 x 2048。“矢量”变量必须为介于 1 到 4 之间的整数，指定数据存储的矢量数组(VEC1,VEC2,VEC3 或 VEC4)。变量“类型”表示数据类型：1 表示 MTF、2 表示实部、3 表示虚部、4 表示弧度单位的相位、5 表示方波 MTF。如果给定的任何变量超出有效范围，则会使用最近的值替代。使用 FFT MTF 方法来计算 MTF、实部和虚部等数据值（参阅“FFT MTF”）。返回的数据存储在指定的矢量数组变量中，存储格式如下所示：

矢量位置 0：子午响应；矢量位置 1：弧矢响应。

示例：

```
!This macro computes the T & S response at 30 lp/mm
! for the currently loaded lens, polychromatic,
! at the maximum defined field,
! and a 32x32 grid density (sampling = 1).
!Data will be placed in vector 1.
!This is all it takes to get the data:
GETMTF 30, 0, NFLD(), 1, 1, 1
PRINT "Tangential response:", vec1(0)
PRINT "Sagittal response :", vec1(1)
```

GETNSCMTF 在非序列模式下，根据矩形探测器上的点列图，计算 X 和 Y 方向的几何 MTF。语法：
GETNSCMTF 频率，表面，物体
详述：“频率”是以 MTF 单位表示的空间频率(参阅“MTF 单位”)。如果频率小于 0，则 GETNSCMTF 使用其绝对值。“表面”允许在序列/非序列模式下使用此变量，对于纯非序列模式下的系统，始终将“表面”值设为 1。“物体”表示非序列中的物体编号，指定需要计算 MTF 的矩形探测器，且只能是矩形探测器。计算时采用几何 MTF 方法，即在矩形探测器上对点列图进行傅里叶变换。

返回的数据存储在指定的矢量数组变量中，存储格式如下所示：

矢量位置 0：X 方向 MTF；矢量位置 1：Y 方向 MTF。

```
示例: ! This macro computes the X & Y direction NSC MTF at 50 lp/mm
! at object 10 - Detector Rectangular
PRINT "Resetting detectors..."
y = NSDD(0, 0, 0, 0)
PRINT "Tracing rays..."
NSTR 1, 0, 0, 0, 0, 1, 0
PRINT "Calculating MTF X and Y..."
GETNSCMTF 50.0, 1, 10
PRINT "mtf X = " + $STR(vec1(0))
```

PRINT "mtf Y = " + \$STR(vec1(1))

GETPSF

使用 FFT 算法计算衍射点扩散函数 (PSF)，并将数据存储于指定的矢量数组中（VEC1、VEC2、VEC3 或 VEC4）。

语法:

```
GETPSF wave, field, sampling, vector, unnormalized, phaseflag, imagedelta
```

详述:

wavelength 指定用于计算的波长编号，wavelength 为 0 时表示多色计算。field 必须为 1 到最大视场数之间的整数，表示要计算的视场编号。sampling 可为 1 (32 x 32)、2 (64 x 64)、3 (128 x 128) 等... 最大为 2048 x 2048。vector 为 1 到 4 之间的整数，指定数据存储的矢量数组(VEC1,VEC2,VEC3 或 VEC4)。unnormalized 值为 0 或 1，为 0 表示返回的所有数据归一化；否则为 1。phaseflag 可为 0 或 1，为 0 表示返回强度数据，为 1 表示返回角度数据。imagedelta 表示 PSF 点之间的间隔（以微米为单位），为 0 表示采用默认间距。波长必须是单色光才能计算相位数据。如果给定的任何变量超出有效范围，则会使用最近的值替代。

返回的数据存储在指定的矢量数组变量中，存储格式如下所示：

矢量位置 0：矢量数组中 PSF 数据点的总数。通常，此数值为 $4 \times n \times n$ ，其中 n 是采样尺寸（32、64 等）。例如，如果采样密度是 2，则光瞳采样将为 64 x 64，此时数组中有 128 x 128（即 16,384）个值。每个数值需要 8 个字节，总共 131 kb。如果采样密度为 1024，则数组至少需要 8 Mb 存储数据，还需要 64 Mb 或更多空间来计算 PSF。矢量位置 0 返回其他值表示出现不同错误。返回 0 表示终止计算。返回 -1，表示矢量数组的大小不足以容纳所有数据，可使用 SETVECSIZE 来增大数组尺寸。返回 -2，表示没有足够系统 RAM 来计算 PSF 数据。返回 -3，表示计算 PSF 时出现常规错误。

矢量位置 1 到 $4*n*n$ 返回 PSF 强度数据。前 $2n$ 个值是第一行，从左到右依次为 $-x$ 到 $+x$ ，接下来每 $2n$ 个值为一行，从 $-y$ 到 $+y$ 行。

矢量位置 $4*n*n+1$ 返回数据值之间的间隔（以微米为单位）。

示例:

```
!This macro computes the PSF
! for the currently loaded lens, polychromatic,
! at the first field,
! and a 32x32 grid density (sampling = 1),
! data will be placed in vector 1,
! normalized to 1,
! no phase data,
! default image delta.

SETVECSIZE 4500
GETPSF 0, 1, 1, 1, 0, 0, 0
np = vec1(0)
IF (np == 0)
    PRINT "PSF Computation aborted."
    GOTO 1
ENDIF
IF (np == -1)
    PRINT "SETVECSIZE too small for PSF data."
    GOTO 1
ENDIF
IF (np == -2)
    PRINT "Not enough system RAM for PSF data."
    GOTO 1
ENDIF
PRINT "There are ", np, " data points, spaced ", vec1(np+1), " micrometers apart".
LABEL 1
```

GETSYSTEMDATA

获取大部分系统数据，比如有效焦距、工作 F/#、切趾因子以及与面型无关的其它数据。返回的数据存储在指定的矢量数组变量中(VEC1,VEC2,VEC3 或 VEC4)。

语法:

```
GETSYSTEMDATA vector_expression
```

详述:

数据存储在指定的 VECn 数组变量中。例如，如果执行命令 GETSYSTEMDATA 1，则系统数据存储于矢量 VEC1 中。另请参阅 SETSYSTEMPROPERTY。

数据用以下格式存储，其中每行的第一个数字表示数组中的位置:

- 0: 指定矢量数组中系统数据的数量。
- 1: 孔径值
- 2: 切趾因子
- 3: 切趾类型 (0: 无, 1: 高斯, 2: 正切)
- 4: “折射率与环境匹配”设置 (1 表示勾选, 0 表示未勾选)
- 5: 以摄氏度表示的温度 (仅当“使用环境数据”为 true 时才有效)
- 6: 以 ATM 表示的压强 (仅当“使用环境数据”为 true 时才有效)
- 7: 有效焦距
- 8: 像方空间 $F/\#$
- 9: 物方空间数值孔径
- 10: 工作 $F/\#$
- 11: 入瞳直径
- 12: 入瞳位置
- 13: 出瞳直径
- 14: 出瞳位置
- 15: 近轴像高
- 16: 近轴放大率
- 17: 角放大率
- 18: 系统总长
- 19: 光线瞄准 (0 表示关闭, 1 表示近轴, 2 表示实际)
- 20: X 光瞳漂移
- 21: Y 光瞳漂移
- 22: Z 光瞳漂移
- 23: 光阑面编号
- 24: 全局坐标参考面编号
- 25: 远心物空间 (0 表示关闭, 1 表示开启)
- 26: 多重结构数量
- 27: 多重结构操作数的数量
- 28: 评价函数操作数的数量
- 29: 公差操作数的数量
- 30: 无焦像方空间 (0 表示关闭, 1 表示开启)
- 31: X 光瞳压缩
- 32: Y 光瞳压缩
- 33: 系统的轴对称性 (0 表示轴对称, 1 表示非轴对称)

GETTEXTFILE(keywords)

根据 OpticStudio 中分析窗口的文本创建新的文本文档。

语法:

```
GETTEXTFILE textfilename, type, settingsfilename, flag
```

详述:

textfilename 可以是引号内的文件名或者字符串变量名, 包括要创建文件的完整路径、名称和扩

展名。字符串函数 \$TEMPFILENAME 可用于定义合适的临时文件名。

type 变量是一个 3 字符的字符串代码，表示要执行的分析窗口类型。如果需要不同分析窗口对应的字符串代码的完整列表，请参阅[编程菜单](Programming Tab)的[字符串代码](String Codes)栏目。如果没有提供分析窗口的类型或者无法识别类型，则进行标准光线追迹。

settingsfilename 变量可以是引号内的设置文件名或者字符串变量名，OpticStudio 执行 GETTEXTFILE 命令时采用已保存的设置进行文本计算，设置类型具体取决于 flag 参数的数值。

如果 flag 的值为 0，那么将使用默认设置。如果当前镜头文件修改了某些分析窗口的默认设置并保存了.cfg 格式文件，则将使用当前镜头文件定义的设置，这些设置都存储于 “lensfilename.cfg” 格式文件中。如果当前镜头没有.cfg 格式文件，则将使用存储在 zemax\configs 文件夹中的 “Zemax.CFG” 文件定义的设置。如果没有任何预先设置，则使用 OpticStudio 的“出厂”默认设置。

如果 flag 的值为 1，那么使用设置文件中的有效设置进行计算并生成文件。如果设置文件中的数据无效，则使用默认设置进行计算并生成文件。只有通过 OpticStudio 保存的设置文件才有效，用户可将其重命名。例如，在点列图的设置对话框中，点击[保存](Save)将生成 “SPT.CFG” 文件。此时将其重命名为 “MySPT.CFG”，然后可在执行 GetTextFile 命令时将 settingsfilename 变量设为 “MySPT.CFG” 的完整路径名。

如果 flag 的值为 2，那么将会显示设置文件中的有效设置对话框，用户可在对话框中对当前设置进行修改，GETTEXTFILE 根据新的设置生成文件。设置对话框使用当前镜头数据编辑器中的数据计算。

不管 flag 的值是多少，只要 settingsfilename 使用了有效的文件名，那么设置都将写入到设置文件中，并覆盖文件中的所有数据。要修改现有设置文件中定义的设置，请使用关键字 MODIFYSETTINGS。

GETTEXTFILE 仅支持文本文件，不支持图形文件。

另请参阅 OPEN、CLOSE、READ、READNEXT、READSTRING 和 MODIFYSETTINGS。

示例：

```
!Macro sample to generate cardinal points in a file.
!Get a temporary file name
A$ = $TEMPFILENAME()
!Compute the data and place in the temp file
GETTEXTFILE A$, Car
!Open the new file and print it out
OPEN A$
LABEL 1
READSTRING B$
IF (EOFF()) THEN GOTO 2
PRINT B$
GOTO 1
LABEL 2
CLOSE
```


[使用 GetTextFile 获取分析数据](Getting Analysis Data Using GetTextFile)

扩展程序经常被忽略的一个功能是：能够通过单次调用 `zclient`，从 `OpticStudio` 所支持的分析类型来创建文本和图形数据文件。例如，要生成点列图列表的文本，可使用以下命令

```
PostRequestMessage("GetTextFile","C:\\\\OUTPUT.TXT\\",Spt, , 0", szBuffer);
```

点列图数据将放置在指定文件夹下文本文件“OUTPUT.TXT”中。有关更多信息，请参阅 `GetTextFile`。

[字符串代码](String Codes)

字符串代码是关键字“GETTEXTFILE”和“OPENANALYSISWINDOW”以及扩展“GetTextFile”中使用的 3 字符的变量。

字符串代码在下方表格中列出。请注意，代码区分大小写。

code	说明
[文件代码](File Codes)	
“New”	NEW_FILE
“Ope”	OPEN_FILE
“Sav”	SAVE_FILE
“Sas”	SAVE_AS
“Bac”	BACKUP_TO_ARCHIVE_FILE
“Res”	RESTORE_FROM_ARCHIVE_FILE
“Ins”	INSERT_LENS

“Prf”	PREFERENCES
“Ext”	EXIT
“Upd”	UPDATE
“Upa”	UPDATE_ALL
“Gen”	GENERAL_LENS_DATA
“Fie”	FIELD_DATA
“Wav”	WAVELENGTH_DATA
“Nxc”	NEXT_CONFIGURATION
“Lac”	LAST_CONFIGURATION
“LDE”	LENS_DATA_EDITOR
“MFE”	MERIT_FUNCTION_EDITOR
“MCE”	MULTI_CONFIG_EDITOR
“TDE”	TOLERANCE_DATA_EDITOR
“EDE”	EXTRA_DATA_EDITOR
“NCE”	NON_SEQUENTIAL_EDITOR
“Und”	UNDO
“Red”	REDO
[布局代码](Layout Codes)	
“Lay”	2D_LAYOUT
“L3d”	3D_LAYOUT
“Lwf”	WIREFRAME_LAYOUT
“Lsm”	SOLID_MODEL_LAYOUT
“Lsh”	SHADED_MODEL_LAYOUT
“Ele”	ZEMAX_ELEMENT_DRAWING
“ISO”	ISO_ELEMENT_DRAWING

“L3n”	NSC_3D_LAYOUT
“LSn”	NSC_SHADED_MODEL_LAYOUT
“Obv”	NSC_OBJECT_VIEWER
“Pvr”	CAD_PART_VIEWER
[光扇代码](Fan Codes)	
“Ray”	RAY_FAN
“Opd”	OPD_FAN
“Pab”	PUPIL_ABERRATION_FAN
[点列图代码](Spots Codes)	
“Spt”	SPOT_DIAGRAM
“Stf”	THROUGH_FOCUS_SPOT
“Sff”	FULL_FIELD_SPOT
“Sma”	SPOT_MATRIX
“Smc”	SPOT_MATRIX_CONFIG
[MTF 代码](MTF Codes)	
“Mtf”	MODULATION_TF
“Tfm”	THROUGH_FOCUS_MTF
“Smf”	SURFACE_MTF
“Mth”	MTF_VS_FIELD
“Fmm”	FFT_MTF_MAP
“Gmm”	GEOMETRIC_MTF_MAP
“Hmf”	HUYGENS_MTF
“Hsm”	HUYGENS_SURFACE_MTF
“Hmh”	HUYGENS_MTF_VS_FIELD
“Gtf”	GEOMETRIC_MTF

“Tfg”	THROUGH_FOCUS_GTF
“Htf”	HUYGENS_THROUGH_FOCUS_MTF
“Gvf”	GEOMETRIC_MTF_VS_FIELD
[PSF 代码](PSF Codes)	
“Fps”	FFT_PSF
“Pcs”	PSF_CROSS_SECTION
“Lsf”	FFT_LINE/EDGE_SPREAD
“Hps”	HUYGENS_PSF
“Hcs”	HUYGENS_PSF_CROSS_SECTION
[波前代码](Wavefront Codes)	
“Wfm”	WAVEFRONT_MAP
“Int”	INTERFEROGRAM
“Foa”	FOUCAULT_ANALYSIS
[表面矢高和相位代码](Surface Sag and Phase Codes)	
“Srs”	SURFACE_SAG
“Srp”	SURFACE_PHASE
“Scv”	SURFACE_CURVATURE
“Ssc”	SURFACE_SAG_CROSS
“Spc”	SURFACE_PHASE_CROSS
“Scc”	SURFACE_CURVATURE_CROSS
[RMS 代码](RMS Codes)	
“Rms”	RMS_VS_FIELD
“Rmw”	RMS_VS_WAVELENGTH
“Rmf”	RMS_VS_FOCUS
“Rfm”	RMS_FIELD_MAP

[圈入能量代码](Encircles Energy Codes)	
“Enc”	DIFF_ENCIRCLED_ENERGY
“Gee”	GEOM_ENCIRCLED_ENERGY
“Lin”	LINE/EDGE_SPREAD
“Xse”	EXTENDED_SOURCE_ENCIRCLED
“Rel”	RELATIVE_ILLUMINATION
“Ils”	ILLUMINATION_SCAN
“Ilf”	ILLUMINATION_SURFACE
“Foo”	FOOTPRINT_ANALYSIS
“Uni”	UNIVERSAL_PLOT_1D
“Un2”	UNIVERSAL_PLOT_2D
“Fcd”	FIELD_CURV/DISTORTION
“Grd”	GRID_DISTORTION
“Lon”	LONGITUDINAL_ABERRATION
“Lat”	LATERAL_COLOR
“Sim”	IMAGE_SIMULATION
“Ima”	GEOMETRIC_IMAGE_ANALYSIS
“Ibm”	GEOMETRIC_BITMAP_IMAGE_ANALYSIS
“Lsa”	LIGHT_SOURCE_ANALYSIS
[其它杂项代码](Other Miscellaneous Codes)	
“ABg”	ABG_DATA_CATALOG
“Agm”	ATHERMAL_GLASS_MAP
“Bfv”	BEAM_FILE_VIEWER
“C31”	SRC_COLOR_CHART_1931
“C76”	SRC_COLOR_CHART_1976

“Caa”	COATING_ABS_VS_ANGLE
“Car”	CARDINAL_POINTS
“Cas”	COAT_ALL_SURFACES
“Caw”	COATING_ABS_VS_WAVELENGTH
“Cca”	CONVERT_TO_CIRCULAR_APERTURES
“Cda”	COATING_DIA_VS_ANGLE
“Cdw”	COATING_DIA_VS_WAVELENGTH
“Cfa”	CONVERT_TO_FLOATING_APERTURES
“Cfm”	CONVERT_TO_MAXIMUM_APERTURES
“Cfo”	CONVERT_FORMAT
“Cfs”	CHROMATIC_FOCAL_SHIFT
“Cgl”	CONVERT_GLOBAL_TO_LOCAL
“Chk”	SYSTEM_CHECK
“Clg”	CONVERT_LOCAL_TO_GLOBAL
“Cls”	COATING_LIST
“Cna”	COATING_RET_VS_ANGLE
“Cng”	CONVERT_TO_NSC_GROUP
“Cnw”	COATING_RET_VS_WAVELENGTH
“Coa”	CONVERT_ASHERE
“Con”	CONJUGATE_SURFACE_ANALYSIS
“Cpa”	COATING_PHASE_VS_ANGLE
“Cpw”	COATING_PHASE_VS_WAVELENGTH
“Cra”	COATING_REFL_VS_ANGLE
“Crw”	COATING_REFL_VS_WAVELENGTH
“Csd”	NSC_CONCATENATE_SPECTRAL_SOURCE_FILES

“Csf”	NSC_CONVERT_TO_SPECTRAL_SOURCE_FILE
“Cta”	COATING_TRAN_VS_ANGLE
“Ctw”	COATING_TRAN_VS_WAVELENGTH
“Dbl”	DOUBLE_PASS
“Dim”	PARTIALLY_COHERENT_IMAGE_ANALYSIS
“Dip”	BIOCULAR_DIPVERGENCE/CONVERGENCE
“Dis”	DISPERSION_PLOT
“Drs”	NSC_DOWNLOAD_RADIANT_SOURCE
“Dvl”	DISPERSION_VS_WAVELENGTH_PLOT
“Dvr”	NSC_DETECTOR_VIEWER;
“Eca”	EXPLODE_CAD_ASSEMBLY
“Ecp”	EXPLODE_CREO_PARAMETRIC_ASSEMBLY
“Ect”	EDIT_COATING
“Eec”	EXPORT_ENCRYPTED_COATING
“Eia”	EXPLODE_INVENTOR_ASSEMBLY
“Esa”	EXPLODE_SOLIDWORKS_ASSEMBLY
“Fba”	FIND_BEST_ASPHERE
“Fcl”	FIBER_COUPLING
“Fld”	ADD_FOLD_MIRROR
“Flx”	DEL_FOLD_MIRROR
“Fov”	BIOCULAR_FIELD_OF_VIEW_ANALYSIS
“Fvw”	NSC_ZRD_FLUX_VS_WAVELENGTH
“Gbp”	PARAX_GAUSSIAN_BEAM
“Gbs”	SKEW_GAUSSIAN_BEAM
“Gcp”	GLASS_COMPARE

“Gft”	GLASS_FIT
“Gho”	GHOST_FOCUS
“Gip”	GRIN_PROFILE
“Gla”	GLASS_CATALOG
“Glb”	GLOBAL_OPTIMIZATION
“Gmf”	GENERATE_MAT_FILE
“Gmp”	GLASS_MAP
“Gpr”	GRADIUM_PROFILE
“Grs”	NSC_GENERATE_RADIANT_SOURCE
“Gst”	GLASS_SUBSTITUTION_TEMPLATE
“Ham”	HAMMER_OPTIMIZATION
“Hlp”	HELP
“Igs”	EXPORT_IGES_FILE
“Iht”	ANGLE_VS_IMAGEHEIGHT
“Imv”	IMA/BIM_FILE_VIEWER
“Jbv”	JPG/BMP_FILE_VIEWER
“Len”	LENS_SEARCH
“Lok”	LOCK_ALL_WINDOWS
“Ltr”	NSC_LIGHTNING_TRACE
“Mfl”	MERIT_FUNCTION_LIST
“Mfo”	MAKE_FOCAL
“Opt”	OPTIMIZATION
“Pal”	POWER_FIELD_MAP
“Pat”	NSC_ZRDA_PATH_ANALYSIS
“Pci”	PARTIALLY_COHERENT_IMAGE_ANALYSIS

“Per”	PERFORMANCE_TEST
“Pha”	POL_PHASE_ABERRATION
“Pmp”	POL_PUPIL_MAP
“Pol”	POL_RAY_TRACE
“Pop”	PHYSICAL_OPTICS_PROPAGATION
“Ppm”	POWER_PUPIL_MAP
“Pre”	PRESCRIPTION_DATA
“Ptf”	POL_TRANSMISSION_FAN
“Pzd”	PLAYBACK_ZRD_ON_DETECTORS
“Qad”	QUICK_ADJUST
“Qfo”	QUICK_FOCUS
“Raa”	REMOVE_ALL_APERTURES
“Rcf”	RELOAD_COATING
“Rda”	NSC_REVERSE_RADIANCE_ANALYSIS
“Rdb”	RAY_DATABASE
“Rdw”	NSC_ROADWAY_LIGHTING_ANALYSIS
“Rev”	REVERSE_ELEMENTS
“Rg4”	NEW_REPORT_GRAPHIC_4
“Rg6”	NEW_REPORT_GRAPHIC_6
“Rml”	REFRESH_MACRO_LIST
“Rtc”	RAY_TRACE_CONTROL
“Rtr”	RAY_TRACE
“Rva”	REMOVE_VARIABLES
“Rxl”	REFRESH_EXTENSIONS_LIST
“Sag”	SAG_TABLE

“Sca”	SCALE_LENS
“Sdi”	SEIDEL_DIAGRAM
“Sdv”	SRC_DIRECTIVITY
“Sei”	SEIDEL_COEFFICIENTS
“Sfv”	SCATTER_FUNCTION_VIEWER
“Sld”	SLIDER
“SIm”	STOCK_LENS_MATCHING
“Spj”	SRC_PROJECTION
“Spo”	SRC_POLAR
“Spv”	SCATTER_POLAR_PLOT
“Srv”	SRC_RSM_VIEWER
“Ssg”	SYSTEM_SUMMARY_GRAPHIC
“Ssp”	SRC_SPECTRUM
“Sti”	NSC_CONVERT_SDF_TO_IES
“Sur”	SURFACE_DATA
“Sys”	SYSTEM_DATA
“Tde”	TILT/DECENTER_ELEMENTS
“TIs”	TOLERANCE_LIST
“Tol”	TOLERANCING
“Tpf”	TEST_PLATE_FIT
“Tpl”	TEST_PLATE_LISTS
“Tra”	POL_TRANSMISSION
“Trw”	TRANSMISSION_VS_WAVELENGTH
“Tsm”	TOLERANCE_SUMMARY
“Unl”	UNLOCK_ALL_WINDOWS

“Vig”	VIGNETTING_PLOT
“Vop”	VISUAL_OPTIMIZER
“Vra”	VARIABLE_RAD
“Vth”	VARIABLE_THI
“Xdi”	EXTENDED_DIFFRACTION_IMAGE_ANALYSIS
“Xis”	EXPORT_IGES/STEP/SAT/STL_FILE
“Yni”	YNI_CONTRIBUTIONS
“Yyb”	Y-YBAR
“Zat”	ZERNIKE_ANNULAR_TERMS
“Zbb”	EXPORT_ZEMAX_BLACK_BOX_DATA
“Zex”	ZEMAX_EXTENSIONS
“Zfr”	ZERNIKE_FRINGE_TERMS
“Zpd”	NSC_ZEMAX_PART_DESIGNER
“Zpl”	EDIT/RUN_ZPL_MACROS
“Zst”	ZERNIKE_STANDARD_TERMS
“Zvf”	ZERNIKE_COEFFICIENTS_VS_FIELD

GETVARDATA

获取当前所有优化变量的数量、类型和数值。并将数据存储于某个矢量数组变量中（VEC1、VEC2、VEC3 或 VEC4）。

语法：

```
GETVARDATA vector
```

详述：

数据存储于指定的数组变量 VECn 中。例如，如果执行了命令 GETVARDATA 1，那么数据将存储于 VEC1 中。数据用以下格式存储，其中每行的第一个数字表示数组中的位置：

- 0: n, 变量的数量
- 1: 第一个变量的类型代码
- 2: 第一个变量的表面编号
- 3: 第一个变量的参数编号
- 4: 第一个变量的物体编号
- 5: 第一个变量的数值
- 5*q-4: 第 q 个变量的类型代码
- 5*q-3: 第 q 个变量的表面编号
- 5*q-2: 第 q 个变量的参数编号
- 5*q-1: 第 q 个变量的物体编号
- 5*q: 第 q 个变量的数值
- etc...

整数 q 的值从 1 到 n，其中 n 是变量数量。如果 n 为 0，则无有效数据返回。数组位置 0：n 值始终有效。变量的类型代码如下表所述。对于不同类型的变量，表面编号、参数编号和物体编号可能没有意义。

GETVARDATA 的类型和 ID 代码

变量类型	类型代码	表面	参数	物体
曲率	1	表面#	-	-
厚度	2	表面#	-	-
圆锥系数	3	表面#	-	-
折射率 Nd	4	表面#	-	-
阿贝数 Vd	5	表面#	-	-

部分色散 $\Delta P_{g,F}$	6	表面#	-	-
TCE	7	表面#	-	-
参数值	8	表面#	参数#	-
附加数据值	9	表面#	附加数据#	-
多重结构操作数值	10	操作数#	结构#	-
非序列物体位置 X	11	表面#	-	物体#
非序列物体位置 Y	12	表面#	-	物体#
非序列物体位置 Z	13	表面#	-	物体#
非序列物体 X 倾斜	14	表面#	-	物体#
非序列物体 Y 倾斜	15	表面#	-	物体#
非序列物体 Z 倾斜	16	表面#	-	物体#
非序列物体参数	17	表面#	参数#	物体#

GETZERNIKE

计算当前载入的镜头文件的 泽尼克边缘、标准或环形系数，并将其存储于某个矢量数组变量中（VEC1、VEC2、VEC3 或 VEC4）。

语法：

```
GETZERNIKE maxorder, wave, field, sampling, vector, zerntype, epsilon, reference
```

详述：

对于 泽尼克边缘系数而言，变量 maxorder 为 1 到 37 之间的任何数值，对于 标准系数或环形系数，变量 maxorder 为 1 到 231 之间的任何数值（请参阅下文有关 zerntype 的详述）。wavelength 和 field 是分别表示波长和视场编号对应的整数值。sampling 的值指定用于拟合系数的网格尺寸。sampling 可以是 1 (32 x 32)、2 (64 x 64) 等，最大为 2048 x 2048。vector 变量必须为介于 1 到 4 之间的整数，指定将数据存储在哪个矢量数组中。zerntype 为 0 表示 边缘泽尼克项，1 表示标准泽尼克项，2 表示 环形泽尼克项。关于不同类型泽尼克项的更多描述，请参阅[泽尼克边缘系数](Zernike Fringe Coefficients), [泽尼克标准系数](Zernike Standard Coefficients), 及[泽尼克环形系

数](Zernike Annular Coefficients)。对于泽尼克环形系数，“epsilon”表示环形比率；对于其它泽尼克类型，将忽略此值。要使 OPD 参考主光线，reference 值应为 0 或者忽略；reference 值为 1 表示参考面顶点。如果给定的任何变量超出有效范围，则会使用最近的值替代。

数据用以下格式返回在其中某个矢量数组中：矢量位置 1：以波长为单位的 PV 值；矢量位置 2：OPD 为 0 时的 RMS 值（以波长为单位）（此值没有实际意义，仅供参考）；矢量位置 3：主光线的 RMS 值（以波长为单位）；矢量位置 4：图像质心的 RMS 值（以波长为单位）（这是与图像质量相关的最有实际意义的数值）；矢量位置 5：以波长为单位的方差值；矢量位置 6：斯特列尔比；矢量位置 7：RMS 拟合误差（以波长为单位）；矢量位置 8：任意点的最大拟合误差（以波长为单位）；其余的矢量位置包含实际的 Zernike 系数数据。例如，Zernike 项数 1 位于矢量位置 9，Zernike 项数 2 位于矢量位置 10，以此类推。

示例：

```
!This macro computes the first 37 Zernike Fringe coefficients
! for the currently loaded lens, at wave 1, field 1
! and a 32x32 grid density (sampling = 1).The coefficients
! will be placed in vector 1.First get the data:
GETZERNIKE 37,1,1,1,1,0
!Now print it out:
FORMAT 16.6
PRINT "Peak to Valley   : ", vec1(1)
PRINT "RMS to chief     : ", vec1(3)
PRINT "RMS to centroid : ", vec1(4)
PRINT "Variance        : ", vec1(5)
PRINT "Strehl ratio     : ", vec1(6)
PRINT "RMS Fit Error    : ", vec1(7)
PRINT "Maximum Fit Error : ", vec1(8)
i = 1
label 1
    FORMAT 2.0
    PRINT "Zernike #", i, " = ",
    FORMAT 16.6
    PRINT vec1(8+i)
    i = i + 1
if (i < 38) THEN GOTO 1
PRINT "All Done!"
```

GLAS

该命令已被废弃。请参阅“SETSURFACEPROPERTY、SURP”。

GLASSTEMPLATE

GLASSTEMPLATE 设置玻璃替代模板上的数据。

语法:

```
GLASSTEMPLATE code, data
```

详述:

对于复选框 `code` 的值，使用玻璃替代模板的代码为 **0**，排除数据不全的玻璃的代码为 **1**，标准的代码为 **11**，首选的代码为 **12**，废弃的代码为 **13**，特殊的代码为 **14**，使用相对成本、**CR**、**FR**、**SR**、**AR** 和 **PR** 的代码分别为 **21-26**。如果 `data` 的值为 **0**，则将取消选中此复选框。任何非 **0**`data` 值都将选中此复选框。

若要修改使用相对成本、**CR**、**FR**、**SR**、**AR** 和 **PR** 对应的数值，则 `code` 的值分别为 **31-36**，`data` 的值可根据用户需求输入合适值。

相关函数:

GTEM

GLENSNAME

GLENSNAME 在用户自定义的图形屏幕上的文本框左上角显示当前镜头名称。

语法:

```
GLENSNAME
```

详述:

GLENSNAME 主要用于使用户自定义的图形类似于其它 OpticStudio 图形。

示例:

请参阅 GRAPHICS 一节。

GLOBALTOLOCAL(keywords)

将坐标变换从全局转换为局部坐标系参考。

语法:

```
GLOBALTOLOCAL surf1, surf2, direction
```

详述:

GLOBALTOLOCAL 可以将所有相邻坐标变换面合并成一个单一的坐标变换面。`surf1` 和 `surf2` 定义表面范围。`direction` 为 **0** 表示坐标正向，为 **1** 表示坐标反向。请参阅[全局到局部] (Global To Local)。

另请参阅 LOCALTOGLOBAL。

GOSUB、SUB、RETURN 和 END

这四个关键字同时使用，用于在 ZPL 宏文件中定义和调用子程序。每个关键字都有特殊用途。**GOSUB** 用于将程序流定位到某个子程序。**SUB** 用于定义子程序名称，并指定子程序主体的开始。**RETURN** 表示在调用了最新的 **GOSUB** 后，宏继续执行。**END** 用于终止宏。

语法：

有关语法示例，请参阅[示例] (Example)一节。

详述：

每个 ZPL 宏文件的子程序不能超过 100 个。每个子程序都必须用 **RETURN** 命令来终止。子宏体中可能不止一个 **RETURN** 命令。如果定义了子程序，那么至少使用一个 **END** 命令来指示主宏体的结束，主宏体必须在文件的顶部。

在一个 ZPL 宏中使用的嵌套级别不能超过 100。例如，如果子程序 **ABC** 调用子程序 **XYZ**，则嵌套级别为 2。如果子程序 **XYZ** 又调用子程序 **DEF**，则嵌套级别为 3。

ZPL 中的所有变量均为全局变量。任何一个在子程序中使用或定义的变量，也都会存在于主宏中。

示例：

```
x = 1 y = 2
GOSUB add
print "the sum of ", x, " and ", y, " is ", z
END

SUB add
z = x + y
RETURN
```

GOTO

通常情况下，宏中的每行命令依次执行。**GOTO** 可用于定义在宏中的任意点重新开始执行。**GOTO** 始终与 **LABEL** 命令一起使用。

语法：

```
GOTO label_number
GOTO text_label
```

详述：

在宏中必须有一个包含对应 **label_number** 或 **text_label** 的 **LABEL** 命令，否则宏将出现错误。

示例：

```
LABEL 1
```



```
x = RAND(10)
if x <= 5 THEN GOTO 1
PRINT " X is greater than 5 "
```

GRAPHICS(keywords)

使用图形标题对应的标尺线创建一个标准 OpticStudio 图形框架。

有关创建图形的简单方法，请参阅“PLOT”和“PLOT2D”。

语法:

```
GRAPHICS
GRAPHICS NOFRAME
GRAPHICS OFF
```

详述:

如果单独定义了 GRAPHICS，那么将产生一个标准的 OpticStudio 图形窗口。如果提供了可选的变量 NOFRAME，那么在标准图形窗口中不会显示标题栏。所有后续图形命令都会发送到这个新创建的窗口中。GRAPHICS OFF 将关闭现已打开的图形窗口，然后显示这个关闭的窗口。

示例:

```
graphics
xmx = xmax()
xmn = xmin()
ymx = ymax()
ymn = ymin()
xwidth = xmx-xmn
ywidth = ymx-ymn
xleft = xmn + (.1 * xwidth)
xrigh = xmn + (.9 * xwidth)
ytopp = ymn + (.1 * ywidth)
ybott = ymn + (.7 * ywidth)
line xleft,ytopp,xrigh,ytopp
line xrigh,ytopp,xrigh,ybott
line xrigh,ybott,xleft,ybott
line xleft,ybott,xleft,ytopp
gtitle " the rain in spain falls mainly on the plain"
glensname
gdate
gtext xmx/2,ymx/2,0, " start this text in the center."
gtextcent ymx*.05, " center this text near the top."
gtext xmx*.05,ymx*.75,90, " place me vertically near left edge."
gtext xmx*.15,ymx*.68,30, " orient me at 30 degrees."
graphics off
```

相关关键字:

LINE、GITITLE、GLENSNAME、GDATE、GTEXTCENT、GTEXT、PIXEL、PLOT

GTEXT

GTEXT 表示使用自定义文本来标记图形。

语法:

```
GTEXT x, y, angle, user_text
GTEXT x, y, angle, A$
```

详述:

坐标 **x** 和 **y** 指定要显示文本字符串 `user_text` 的左边缘。`user_text` 可以用引号内的常量字符串，也可以是字符串变量。`angle` 指定自定义文本相对于图形框架的旋转角度，默认为 **0 度**（水平）。另请参阅 **SETTEXTSIZE**。

示例:

请参阅[图形] (GRAPHICS)。

GTEXTCENT

GTEXTCENT 表示使用用户自定义文本标记图形中心。

语法:

```
GTEXTCENT y, user_text
```

详述:

坐标 **y** 指的是文本字符串 `user_text` 的垂直位置。另请参阅 **SETTEXTSIZE**。

示例:

请参阅[图形] (GRAPHICS)。

GTITLE

GTITLE 类似于 GTEXT，唯一区别是需要指定文本，且文本将显示在图形标题栏的中央。GTITLE 用于使 ZPL 图形函数生成的图形类似于 OpticStudio 标准图形。

语法:

```
GTITLE user_title
```

示例:

请参阅[图形] (GRAPHICS)。

HAMMER(keywords)

调用锤形优化算法，使用当前评价函数来优化当前镜头。

语法:

```
HAMMER
HAMMER                                     number_of_cycles
HAMMER number_of_cycles, algorithm
```

详述:

如果未提供变量，则锤形优化将使用阻尼最小二乘法运行 1 个周期。如果提供了变量，则变量必须为 1 到 99 之间的整数，并且锤形优化算法将运行指定的周期数。**algorithm** 变量为 0 时表示使用阻尼最小二乘法，为 1 时表示使用正交下降法。有关更多信息，请参阅[执行优化](Performing an optimization)。

相关函数:

MFCN

示例:

```
PRINT "Starting merit function:", MFCN()
HAMMER 3
PRINT "Ending merit function :", MFCN()
```

IF-THEN-ELSE-ENDIF

IF 提供了有条件的宏执行和分支。

语法:

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

或

```
IF (expression) THEN (command)
```

详述:

IF-ELSE-ENDIF 结构用于有条件地执行 IF 后面的命令或者 ELSE 后面的命令，但两者不能同时执行。如果 **expression** 的值为 0，则被视为假（**false**），否则被视为真（**true**）。**expression** 可以是任何有效的 ZPL 表达式（由函数、变量、操作数和常量组成）。IF 命令必须与 ENDIF 成对使用，但 ELSE 为可选。IF-ENDIF 对可以嵌套到任何级别。

IF-THEN 结构便于对单个指令有条件的执行。如果指定了 THEN 关键字，则 IF 命令终止，并且不需要 ENDIF。IF-THEN 结构中不支持 ELSE 关键字。

示例:

```
x = 1
y = 2
if (x < y)
    PRINT "x is less than y"
ELSE
    if (x == y) THEN PRINT "x equals y"
    if (x > y) THEN PRINT "x is greater than y"
ENDIF
```

IMA

计算几何图像分析功能并将结果保存到 **BIM** 文件中。有关 **BIM**（二进制图像）格式的描述，请参阅“**BIM 格式**”。有关几何图像分析功能的描述，请参阅“几何图像分析”。

语法:

```
IMA outfilename,infilename
```

详述:

此关键字需要指定输出的 **BIM** 文件名称，可选的输入 **IMA** 或 **BIM** 文件名称。如果未提供 outfilename 的扩展名，则附加扩展名 **BIM**。必须在 infilename 中提供扩展名。如果文件名中使用了空格或其它特殊字符，则文件名必须加引号。outfilename 和 infilename 都必须存放在 <data>\<images> 文件夹中。文件名不能包含路径。

几何图像分析功能的参数设置采用的是当前镜头之前保存过的设置。若要调整这些参数设置，请打开[几何图像分析](Geometric Image Analysis)窗口并选择适当的设置，然后按[保存](Save)。在[显示为](Show As)参数下拉菜单中不能选择[点列图](spot diagram)。后续对 IMA 的调用都会使用保存的设置。输出文件名和输入源文件例外，输出文件名为 IMA 关键字之后的第一个变量，输入源文件为 IMA 关键字之后的第二个可选变量。

示例:

```
IMA "output.BIM", "LETTERF.IMA"
```

相关关键字:

IMASHOW、IMASUM

IMAGECOMBINE

合并两个图形文件，并将合并的图形写入到新文件中。

语法:

```
IMAGECOMBINE source1$, source2$, destination$, mode$
```

详述:

此关键字将打开两个源图形文件，将其合并为单个图像，然后将结果写入到目标文件中。文件可为 BMP、JPG 或 PNG 格式。基于所提供的扩展名，不同的格式之间可以自动转化。如果源图像的行数相同，则可以从左到右合并。如果源图像的列数相同，则可以从上到下合并。使用的合并方式取决于 mode\$ 值，mode\$ 值可由“LEFTRIGHT”或“TOPBOTTOM”来定义。

示例:

```
S1$ = "C:\TEMP\SOURCE1.JPG" S2$ = "C:\TEMP\SOURCE2.JPG"  
D$ = "C:\TEMP\DESTINATION.JPG" M$ = "LEFTRIGHT"  
IMAGECOMBINE S1$, S2$, D$, M$
```

相关函数:

PIXX, PIXY

相关关键字:

IMAGEEXTRACT

IMAGEEXTRACT

提取图形文件的矩形区域，并将此区域写入到新文件中。

语法:

```
IMAGEEXTRACT source$, destination$, startx, starty, width, height
```

详述:

此关键字打开源图形，提取文件的矩形区域，然后将提取的区域写入到目标文件中。文件可为 BMP、JPG 或 PNG 格式。基于所提供的扩展名，不同的格式之间可以自动转化。startx 和 starty 值分别表示与左侧和顶部的偏移量（以像素为单位）。顶部行和左侧列都从编号 1 开始。Width 和 height 值表示要提取的矩形区域的大小（以像素为单位）

示例:

```
S$ = "C:\TEMP\SOURCE.JPG"  
D$ = "C:\TEMP\DESTINATION.JPG"  
IMAGEEXTRACT S$, D$, 1, 1, 20, 20
```

相关函数:

PIXX, PIXY

相关关键字:

IMAGECOMBINE

IMASHOW

在视图窗口中显示一个 IMA 或 BIM 文件。有关 IMA 和 BIM 文件格式的描述，请参阅[IMA 格式](The IMA format)和[BIM 格式](The BIM format)”。

语法:

```
IMASHOW filename.ima
```

详述:

此关键字需要指定 IMA 或 BIM 文件的名称。必须包含扩展名。如果文件名中使用了空格或其它特殊字符，则文件名必须加引号。文件必须位于 <data>\<images> 文件夹中。该命令将打开新窗口来显示文件。

示例:

```
IMASHOW "LETTERF.IMA"
```

相关关键字:

IMA、IMASUM

IMASUM

对 BIM 文件中的强度求和。尽管此函数的名称为“IMASUM”，但目前只能对 BIM 文件求和，不能对 IMA 文件求和。这个限制是由 IMA 文件格式中所支持的灰阶数量较低而导致的。有关 IMA 和 BIM 文件格式的描述，请参阅“IMA 格式”和“BIM 格式”。

语法:

```
IMASUM filename1, filename2, outfilename
```

详述:

此关键字需要指定三个 BIM 文件名（文件名可相同）。三个文件都必须是 BIM 格式。如果文件名中使用了空格或其它特殊字符，则文件名必须加引号。文件必须位于 <data>\<images> 文件夹中，也是输出文件的位置。两个源文件必须具有相同数量的像素和色彩通道，否则将出现错误信息。

示例:

```
IMASUM "A.BIM", "B.BIM", "sum of A and B.BIM"
```

相关关键字:

IMA, IMASHOW

IMPORTEXTRADATA(keywords)

通过一个文件将数据导入到镜头数据编辑器中。

语法:

```
IMPORTEXTRADATA surface, filename
```

详述:

surface 必须是使用附加数据值的有效表面编号。filename 应包含完整路径。关于有效数据格式，请参阅序列模式中“栅格矢高”表面的“导入栅格数据”。

INPUT

INPUT 在宏执行时，提示用户输入数值或文本。

语法:

```
INPUT          "Prompt"          String"          ,          variable
INPUT          "Prompt"          String"          ,          variable
INPUT          "Prompt"          String"          ,          string_variable$
INPUT string_variable$
```

详述:

variable 可以是任何有效变量名称。如果变量名是字符串变量，则表示输入的内容是文字字符串，否则表示输入的内容是数值。如果没有指定 Prompt String，则 INPUT 命令将使用“?”进行提示。提示框一直显示在屏幕上，直到用户用键盘输入相关数值或字符串。Prompt String 的长度不能超过 200 个字符。

示例:

```
INPUT "Enter value for x:", x
PRINT "X = ", x
INPUT "Enter a value for A$:", A$
PRINT A$
```

INSERT

INSERT 可在镜头数据编辑器中插入新表面。

语法:

```
INSERT surf
```

详述:

surf 值必须为整数表达式。另请参阅“DELETE”和“SURFTYPE”。

示例:

```
INSERT 5  
INSERT i+2*j
```

INSERTCONFIG

INSERTCONFIG 可在多重结构编辑器中插入新的结构组态。

语法:

```
INSERTCONFIG config
```

详述:

config 可为整数值或整数表达式，其结果必须大于 0 且小于等于当前结构数量加 1。另请参阅 DELETECONFIG 和 INSERTMCO。

示例:

```
INSERTCONFIG 4
```

INSERTMCO(keywords)

INSERTMCO 可在多重结构编辑器中插入新的多重结构操作数。

语法:

```
INSERTMCO row
```

详述:

row 可为整数值或整数表达式，其结果必须大于 0 且小于等于当前操作数的数量加 1。另请参阅 DELETETMCO 和 INSERTCONFIG。

示例:

```
INSERTMCO 7
```

INSERTMFO(keywords)

INSERTMFO 可在评价函数编辑器中插入一个新的评价函数操作数。

语法:

```
INSERTMFO row
```

详述:

row 可为整数值或整数表达式，其结果必须大于 0 且小于等于当前操作数的数量加 1。另请参阅

DELETEMFO 和 SETOPERAND。

示例:

```
INSERTMFO 23
```

INSERTOBJECT(keywords)

插入一个新的 NSC 物体。

语法:

```
INSERTOBJECT surf, object
```

详述:

surf 值必须为整数表达式，在混合序列/非序列模式下，surf 的值必须为非序列组件表面的编号。在非序列模式下，surf 的值设为 1。object 的值必须为介于 1 和当前物体数加 1 之间的整数，包括首尾值。新物体将按照指定的位置插入，OpticStudio 会根据要求重新对其它物体进行编号。另请参阅 DELETEOBJECT 和 SETN- SCPROPERTY。

示例:

```
INSERTOBJECT 1, 1
```

INSERTTOL

INSERTTOL 可在公差数据编辑器中插入一个新的公差操作数。

语法:

```
INSERTTOL row
```

详述:

row 可为整数值或整数表达式，其结果必须大于 0 且小于等于当前操作数的数量加 1。另请参阅 DELETETOL 和 SETTOL。

示例:

```
INSERTTOL 3
```

LABEL

LABEL 可为 GOTO 命令提供目标，更多详情，请参阅“GOTO”。

语法:

```
LABEL label_number  
LABEL text_label
```

详述:

label_number 必须是大于 0 的整数，如 1 或 7。如果使用了 **text_label**，则 **text_label** 不得包含空格或者用作分隔符的其它特殊字符。**LABEL** 自身对程序流没有任何影响。在一个宏中，使用的 **LABEL** 命令不能超过 300 个。

示例:

```
LABEL 7  
LABEL startover
```

LINE(keywords)

LINE 是用于图形显示的直线绘制函数。

语法:

```
LINE oldx, oldy, newx, newy
```

详述:

LINE 将求出四个表达式的值，并绘制出连接定义点的直线。坐标指的是当前图形结构的坐标，坐标必须限定在 **XMIN**、**YMIN**、**XMAX** 和 **YMAX** 定义的边界内。**LINE** 可将整数坐标值准确绘制出来，也支持实数坐标变量，将其设为最接近的等效整数值。**LINE** 仅在图形模式下有效。绘制直线的颜色由当前画笔颜色决定，当前画笔颜色可通过关键字 **COLOR** 来指定。

示例:

请参阅[图形] (GRAPHICS)。

LOADARCHIVE

打开现有 Zemax 存档文件 (*.ZAR)。

语法:

```
LOADARCHIVE filename, extractpath
```

详述:

filename 必须包含存档文件的名称，包括文件扩展名。如果 **filename** 不包含存档文件的完整路径，则会使用镜头的默认文件夹。如果未定义 **extractpath**，则假定提取路径与 **filename** 路径相同。随后将打开存档文件中定义的镜头和 **Session** 文件。请参阅[恢复存档文件] (Restore From Archive File)。

LOADCATALOG

为当前加载的镜头重载玻璃库和膜层库。

语法:

```
LOADCATALOG
```

详述:

载入镜头后，将自动载入任何相关的玻璃库和数据文件，包括 COATING.DAT 文件（如果尚未载入）。但是，如果数据库目录已被修改（可能是由 ZPL 宏自身修改的），那么需要用关键字 LOADCATALOG 来重新载入这些库。若在打开当前 ZEMAX Session 文件之后修改了 COATING.DAT 或玻璃 AGF 库文件，则不需要使用此关键字。

LOADDETECTOR(keywords)

将保存在某个文件中的数据加载到 NSC 矩形探测器、颜色探测器、极坐标探测器或体探测器物体中。

语法:

```
LOADDETECTOR surf, object, filename 详述:
```

此关键字需要指定 surf 和 object 的数值表达式以及 filename。在混合序列/非序列模式下，surf 是非序列组件的表面编号；在非序列模式下，surf 的值设为 1。object 是探测器物体的编号。filename 可包含完整路径，如果没有提供路径，则会使用当前镜头的路径。对于矩形探测器、颜色探测器、极坐标探测器和体探测器物体，扩展名应分别为 DDR、DDC、DDP 或 DDV。

相关关键字:

```
SAVEDETECTOR
```

LOADLENS

加载一个新的镜头文件，替换当前内存中的镜头文件。

语法:

```
LOADLENS filename, appendflag, session
```

详述:

LOADLENS 将加载一个镜头文件。如果 filename 包含完整路径（如 C:\MYDIR\MYLENS.ZMX），则将加载指定文件。如果 filename 不含完整路径，则将使用镜头的默认文件夹（参阅[文件夹] (Folders)）。

如果 appendflag 为 0 或者不存在，则 **LOADLENS** 将直接加载指定文件。如果 appendflag 大于 0，则将文件附加到当前镜头数据中，从 appendflag 值指定的表面开始。仅在序列模式下，将一个系统附加到另一个系统时，才会使用 appendflag。目前不支持附加非序列系统。

如果 session 标记不为 0，则将加载镜头的任何关联 session 文件，并且所有窗口都将更新，否则将忽略镜头 session 文件。

示例：

```
LOADLENS "COOKE.ZMX"
```

相关关键字：

SAVELENS

LOADMERIT(keywords)

加载评价函数文件，替换当前镜头中的评价函数。

语法：

```
LOADMERIT                                     "filename"  
LOADMERIT file$
```

详述：

LOADMERIT 将通过 .MF 文件加载一个新的评价函数。如果 filename 包含完整路径（如 C:\MYDIR\MYLENS.MF），则将直接加载指定文件。如果 filename 不含完整路径，则将使用默认路径文件夹<data>\MeritFunction（参阅[文件夹] (Folders) (“Folders”)）。此关键字还可以使用相同语法从 ZMX 文件中加载评价函数。另请参阅 SAVEMERIT。

LOADTOLERANCE(keywords)

加载公差文件，替换当前镜头中的公差。

语法：

```
LOADTOLERANCE "filename"  
LOADTOLERANCE file$
```

详述：

LOADTOLERANC 将加载新的公差文件。如果 filename 包含完整路径（如 C:\MYDIR\MYLENS.TOL），则将加载指定文件。如果 filename 不含完整路径，则将使用默认路径文件夹<data>\Tolerance（参阅[文件夹] (Folders) (“Folders”)）。另请参阅 SAVETOLERANCE。

LOCALTOGLOBAL(keywords)

将坐标变换从局部参考转换为全局参考。

语法:

```
LOCALTOGLOBAL surf1, surf2, reference
```

详述:

LOCALTOGLOBAL 可将所有坐标变换修改为 3 组坐标变换，提供了一种在全局坐标中定位表面的方法。surf1 和 surf2 定义表面范围。reference 指定参考表面编号，必须位于 surf1 之前。请参阅 [局部到全局坐标系] (Local To Global)。另请参阅 GLOBALTOLOCAL。

LOCKWINDOW

锁定任何一个或所有打开的窗口。

语法:

```
LOCKWINDOW winnum
```

详述:

请参阅 [图形窗口操作数] (Graphic windows operations)。如果 winnum 为 0，则将锁定所有已打开的窗口。如果 winnum 为 -1，则在宏结束后，将锁定当前执行的窗口。

示例:

```
LOCKWINDOW 7
```

相关关键字:

UNLOCKWINDOW

MAKEFACETLIST

生成一个 ASCII 列表，列出任何可导入 CAD 文件的多面描述的顶点。

语法:

```
MAKEFACETLIST infile, outfile
```

详述:

infile 可以是字符串或字符串变量，包含 CAD 格式文件的完整路径。outfile 可以是字符串或字符串变量，包含输出文件的完整路径。输出文件是一个文本列表。在物体的多面描述中，每个三角形都有 18 行。前 6 行表示第一个三角形顶点 1 的数据，格式如下：

- x 坐标
- y 坐标
- z 坐标
- x 法向矢量
- y 法向矢量
- z 法向矢量

后续 6 行的值表示顶点 2，再接下来的 6 行的值表示顶点 3。18 个值共同定义第一个三角形。对下一个三角形重复此模式，每个三角形通过 18 个值来表示，直到写入所有三角形。

MAKEFOLDER

为文件创建一个文件夹。

语法:

```
MAKEFOLDER "C:\TEMP\TEST_FOLDER"  
MAKEFOLDER path$
```

详述:

将创建一个新的文件夹。如果文件夹已经存在，则忽略此关键字。如果路径无效，则出现错误信息提示，并终止宏执行。

MODIFYSETTINGS(keywords)

修改 GETTEXTFILE 使用的设置文件中的数据。

语法:

```
MODIFYSETTINGS settingsfilename, type, value, windowtype
```

详述:

settingsfilename 可以是引号内的字符串或者字符串变量名称，并且包括要修改文件的完整路径、名称和扩展名。

变量 type 表示文本助记符，指定要修改文件中的哪些设置。下表中列出了不同变量 type 的助记符。由于有很多不同类型的分析窗口，并且每个窗口具有很多不同设置，因此 type 列表无法包含所有可能的设置。如果需要添加更多 type 值；请联系技术支持以获取帮助。

变量 windowtype 是一个 3 字符的字符串代码，指定要在特定镜头设置文件中查找的分析类型。如需字符串代码的完整列表，请参阅[编程菜单](Programing Tab)的[字符串代码](String Code)栏目。如果省略 windowtype，则将修改与变量 type 匹配的第一个设置；请注意，某些 type 可能匹配多个不同分析窗口设置；如果指定分析窗口中没有任何设置与给定 type 匹配，则不会对文件进行任何更改。

MODIFYSETTINGS 支持的类型代码

功能	有效的类型代码																								
2D 视图(2DLayout)	LAY_RAYS: 光线数量。																								
探测器查看器 (Detector Viewer)	<div>DVW_SURFACE: 表面编号。用 1 表示非序列模式。</div> <div>DVW_DETECTOR: 探测器编号。</div> <div>DVW_SHOW: “显示为”(“Show As”)设置。“Show As”取决于所显示的窗口类型(图形或文本)和探测器类型(矩形探测器、颜色探测器等等)。DVW-SHOW 的值对应于探测器窗口的 UI 中所需项目的位置:</div> <table><thead><tr><th>值</th><th>图形窗口 颜色探测器</th><th>文本窗口 极坐标探测器</th></tr></thead><tbody><tr><td>0</td><td>灰度图</td><td>完整数据列表</td></tr><tr><td>1</td><td>反灰度图</td><td>方位角截面数据</td></tr><tr><td>2</td><td>伪彩色图</td><td></td></tr><tr><td>3</td><td>反伪彩色图</td><td></td></tr><tr><td>4</td><td>真彩色图</td><td></td></tr><tr><td>5</td><td>行截面图</td><td></td></tr><tr><td>6</td><td>列截面图</td><td></td></tr></tbody></table> <div>DVW_ROWCOL: 截面图的行数或列数。</div> <div>DVW_ZPLANE: 体探测器的 Z 平面编号。</div> <div>DVW_SCALE: 比例模式。用 0 表示线性, 1 表示 Log -5, 2 表示 Log -10, 3 表示 Log -15。</div> <div>DVW_SMOOTHING: 整数平滑值。</div> <div>DVW_DATA: “显示数据”(“Show Data”)设置。类似于 DVW_SHOW,</div>	值	图形窗口 颜色探测器	文本窗口 极坐标探测器	0	灰度图	完整数据列表	1	反灰度图	方位角截面数据	2	伪彩色图		3	反伪彩色图		4	真彩色图		5	行截面图		6	列截面图	
值	图形窗口 颜色探测器	文本窗口 极坐标探测器																							
0	灰度图	完整数据列表																							
1	反灰度图	方位角截面数据																							
2	伪彩色图																								
3	反伪彩色图																								
4	真彩色图																								
5	行截面图																								
6	列截面图																								

	<p>“Show Data”取决于其他设置，例如显示窗口类型、探测器类型及光源单位。 DVW_DATA 的值对应于探测器窗口的 UI 中所需项目的位置，例如：</p> <table><tr><td>值</td><td>图形窗口 矩形探测器 光源单位： Lumens</td><td>文本窗口 总结所有</td></tr><tr><td>0</td><td>非相干辐照度</td><td>非相干辐照度</td></tr><tr><td>1</td><td>相干辐照度</td><td>相干辐照度</td></tr><tr><td>2</td><td>相干相位</td><td></td></tr><tr><td>3</td><td>辐射强度</td><td></td></tr><tr><td>4</td><td>辐亮度（位置空间）</td><td></td></tr><tr><td>5</td><td>辐亮度（角度空间）</td><td></td></tr></table> <p>DVW_ZRD： 光线数据库名称，或者为空（表示无名称）。 DVW_FILTER： 过滤字符串。 DVW_MAXPLOT： 最大绘图比例。 DVW_MINPLOT： 最小绘图比例。 DVW_OUTPUTFILE： 输出文件名称。</p>	值	图形窗口 矩形探测器 光源单位： Lumens	文本窗口 总结所有	0	非相干辐照度	非相干辐照度	1	相干辐照度	相干辐照度	2	相干相位		3	辐射强度		4	辐亮度（位置空间）		5	辐亮度（角度空间）	
值	图形窗口 矩形探测器 光源单位： Lumens	文本窗口 总结所有																				
0	非相干辐照度	非相干辐照度																				
1	相干辐照度	相干辐照度																				
2	相干相位																					
3	辐射强度																					
4	辐亮度（位置空间）																					
5	辐亮度（角度空间）																					
扩展衍射图像 分析(Extended Diffraction Image Analysis)	<p>EXD_DISPLAYSIZE： 显示尺寸。 EXD_FIELD： 视场编号。 EXD_FILESIZE： 文件大小。 EXD_WAVE： 波长编号。</p>																					
扩展光源圈入能量 (Extended Source Encircled Energy)	<p>XSE_FIELD： 视场编号 (1, .., n) XSE_FIELDSIZE： 视场尺寸(Field Size) XSE_WAVE： 波长 (0 表示所有波长, 1, ...n)</p>																					

	<p>XSE_KRAYS: 光线 x 1000</p> <p>XSE_IMANAME: *.IMA 文件名</p> <p>XSE_SURFACE: 表面编号</p> <p>XSE_MAXRAD: 最大辐射距离</p> <p>XSE_TYPE: 数据类型(1 表示“圈入能量”，2 表示“仅 X”，3 表示“仅 Y”等等)</p> <p>XSE_REF: 参考(0 表示“主光线”，1 表示“质心”，2 表示“顶点”)</p> <p>XSE_POLARIZATION: 使用偏振标志(0 表示关闭，1 表示开启)</p> <p>XSE_REMVIGNET: 移除渐晕因子标志</p> <p>XSE_MULDIFFLI: 乘以衍射极限标志</p>
<p>FFT PSF 线/边缘扩散</p> <p>(FFT Line/Edge Spread)</p>	<p>LSF_COHERENT: 0 表示非相干，1 表示相干</p> <p>LSF_TYPE: 用 0-9 分别表示 X-线性、Y-线性、X-对数、Y-对数、X-相位、Y-相位、X-实部、Y-实部、X-虚部或 Y-虚部。</p> <p>LSF_SAMP: 采样，用 1 表示 32 x 32，用 2 表示 64 x 64，以此类推。</p> <p>LSF_SPREAD: 用 0 表示线，1 表示边缘。</p> <p>LSF_WAVE: 波长编号，用 0 表示多色(仅限非相干)</p> <p>LSF_FIELD: 视场编号。</p> <p>LSF_POLARIZATION: 用 0 表示无偏振，用 1 表示偏振。</p> <p>LSF_PLOTSIZE: 绘图比例。</p>
FFT PSF	<p>PSF_TYPE: 用 0 到 4 分别表示线性、对数、实部或虚部。</p> <p>PSF_SAMP: 采样，用 1 表示 32 x 32，用 2 表示 64 x 64，以此类推。</p> <p>PSF_WAVE: 波长编号，用 0 表示多色。</p> <p>PSF_FIELD: 视场编号。</p> <p>PSF_SURFACE: 表面编号，用 0 表示像面。</p> <p>PSF_POLARIZATION: 用 0 表示无偏振，用 1 表示偏振。</p> <p>PSF_NORMALIZE: 用 0 表示非归一化，1 表示归一化。</p> <p>PSF_IMAGEDELTA: 像面采样间距(单位为微米)。</p>

FFT PSF 截面图 (FFT PSF Cross Section)	<p>PSF_TYPE: 用 0-9 分别表示 X-线性、Y-线性、X-对数、Y-对数、X-相位、Y-相位、X-实部、Y-实部、X-虚部或 Y-虚部。</p> <p>PSF_ROW: 行数（如果进行 X 扫描）或列数（如果进行 Y 扫描）。用 0 表示中心。</p> <p>PSF_SAMP: 采样，用 1 表示 32 x 32，用 2 表示 64 x 64，以此类推。</p> <p>PSF_WAVE: 波长编号，用 0 表示多色。</p> <p>PSF_FIELD: 视场编号。</p> <p>PSF_POLARIZATION: 用 0 表示无偏振，用 1 表示偏振。</p> <p>PSF_NORMALIZE: 用 0 表示非归一化，1 表示归一化。</p> <p>PSF_PLOTSCALE: 绘图比例。</p>
光迹图 (Footprint Diagram)	<p>FOO_RAYDENSITY: 光线密度。用 0 表示圆环，1 表示 10，2 表示 15，3 表示 20，以此类推。</p> <p>FOO_SURFACE: 表面编号。</p> <p>FOO_FIELD: 视场编号。</p> <p>FOO_WAVELENGTH: 波长编号。</p> <p>FOO_DELETEVIGNETTED: 删除渐晕，用 0 表示否，1 表示是。</p>
几何位图图像 分析 (Geometric Bitmap Image Analysis)	<p>GBM_FIELDSIZE: Y 视场大小。</p> <p>GBM_RAYS: 每个像素的光线数。</p> <p>GBM_XPIX: X 像素数。</p> <p>GBM_YPIX: Y 像素数。</p> <p>GBM_XSIZ: X 像元大小。</p> <p>GBM_YSIZ: Y 像元大小。</p> <p>GBM_INPUT: 输入文件名</p> <p>GBM_OUTPUT: 输出文件名</p> <p>GBM_SURFACE: 表面编号</p> <p>GBM_ROTATION: 翻转设置</p>
几何图像分析	<p>IMA_FIELD: 视场尺寸。</p>

(Geometric Image Analysis)	<p>IMA_IMAGESIZE: 像面尺寸。</p> <p>IMA_IMANAME: 图像文件名。</p> <p>IMA_KRAYS: 光线数 x 1000。</p> <p>IMA_NA: 数值孔径。</p> <p>IMA_OUTNAME: 输出文件名称。</p> <p>IMA_SURFACE: 表面编号。</p> <p>IMA_PIXELS: 像素数。</p>
离焦 FFT MTF (FFT Through Focus MTF)	<p>TFM_SAMP: 采样。用 1 表示 32x32, 2 表示 64x64 等。</p> <p>TFM_DELTAFOC: 离焦范围。</p> <p>TFM_FREQ: 空间频率。</p> <p>TFM_STEPS: 焦平面移动步长。</p> <p>TFM_WAVE: 波长编号。0 表示所有波长。</p> <p>TFM_FIELD: 视场编号。0 表示所有视场。</p> <p>TFM_TYPE: 数据类型。用 0 表示调制, 1 表示实部, 2 表示虚部, 3 表示相位, 4 表示方波。</p> <p>TFM_POLAR: 使用偏振。用 0 表示否, 1 表示是。</p> <p>TFM_DASH: 使用虚线。用 0 表示否, 1 表示是。</p>
惠更斯 MTF (Huygens MTF)	<p>HMF_PUPILSAMP: 光瞳采样。用 1 表示 32x32, 2 表示 64x64 等。</p> <p>HMF_IMAGESAMP: 像面采样。用 1 表示 32x32, 2 表示 64x64 等。</p> <p>HMF_IMAGEDELTA: 像面采样间距 (单位为微米)。</p> <p>HMF_CONFIG: 结构编号。用 0 表示全部, 用 1 表示当前。</p> <p>HMF_WAVE: 波长编号。用 0 表示多色。</p> <p>HMF_FIELD: 视场编号。0 表示所有视场。</p> <p>HMF_TYPE: 数据类型。当前仅支持调制 (0)。</p> <p>HMF_MAXF: 最大空间频率。</p> <p>HMF_POLAR: 使用偏振。用 0 表示否, 1 表示是。</p> <p>HMF_DASH: 使用虚线。用 0 表示否, 1 表示是。</p>

离焦惠更斯 MTF (Huygens Through Focus MTF)	<p>HTF_PUPILSAMP: 光瞳采样。用 1 表示 32x32, 2 表示 64x64 等。</p> <p>HTF_IMAGESAMP: 像面采样。用 1 表示 32x32, 2 表示 64x64 等。</p> <p>HTF_IMAGEDELTA: 像面采样间距 (单位为微米)。</p> <p>HTF_CONFIG: 结构编号。用 0 表示全部, 用 1 表示当前。</p> <p>HTF_FREQ: 空间频率。</p> <p>HTF_WAVE: 波长编号。0 表示所有波长。</p> <p>HTF_FIELD: 视场编号。0 表示所有视场。</p> <p>HTF_TYPE: 数据类型。当前仅支持调制 (0)。</p> <p>HTF_DELTAFOC: 离焦范围。</p> <p>HTF_STEPS: 焦平面移动步长。</p> <p>HTF_POLAR: 使用偏振。用 0 表示否, 1 表示是。</p> <p>HTF_DASH: 使用虚线。用 0 表示否, 1 表示是。</p>
惠更斯 MTF vs. 视场 (Huygens MTF vs. Field)	<p>HMH_SAMP: 采样。用 1 表示 32x32, 2 表示 64x64 等。</p> <p>HMH_SCANTYPE: 视场扫描类型。用 0 表示 +Y, 1 表示 +X, 以此类推。</p> <p>HMH_WAVE: 波长。0 表示所有波长。</p> <p>HMH_FIELDDENSITY: 视场密度。</p> <p>HMH_FREQ1: 空间频率 1。</p> <p>HMH_FREQ2: 空间频率 2。</p> <p>HMH_FREQ3: 空间频率 3。</p> <p>HMH_FREQ4: 空间频率 4。</p> <p>HMH_FREQ5: 空间频率 5。</p> <p>HMH_FREQ6: 空间频率 6。</p> <p>HMH_POLAR: 使用偏振。用 0 表示否, 1 表示是。</p> <p>HMH_DASH: 使用虚线。用 0 表示否, 1 表示是。</p> <p>HMH_REMOVEVIGNETTING: 移除渐晕因子。用 0 表示否, 1 表示是。</p>
惠更斯 PSF (Huygens PSF)	<p>HPS_PUPILSAMP: 光瞳采样, 用 1 表示 32 x 32, 用 2 表示 64 x 64, 以此类推。</p>

	<p>HPS_IMAGESAMP: 像面采样, 用 1 表示 32 x 32, 用 2 表示 64 x64, 以此类推。</p> <p>HPS_WAVE: 波长编号, 用 0 表示多色。</p> <p>HPS_FIELD: 视场编号。</p> <p>HPS_IMAGEDELTA: 像面采样间距 (单位为微米)。</p> <p>HPS_TYPE: 数据类型。用 0 到 8 分别表示线性、Log -1、Log-2、Log -3、Log -4、Log-5、实部、虚部或相位。</p>
惠更斯 PSF 截面图 (Huygens PSF Cross Section)	<p>HPC_PUPILSAMP: 光瞳采样, 用 1 表示 32 x 32, 用 2 表示 64 x 64, 以此类推。</p> <p>HPC_IMAGESAMP: 像面采样, 用 1 表示 32 x 32, 用 2 表示 64 x 64, 以此类推。</p> <p>HPC_WAVE: 波长编号, 用 0 表示多色。</p> <p>HPC_FIELD: 视场编号。</p> <p>HPC_IMAGEDELTA: 像面采样间距 (单位为微米)。</p> <p>HPC_TYPE: 数据类型。用 0 到 9 分别表示 X 线性、X 对数、Y 线性、Y 对数、X 实部、Y 实部、X 虚部、Y 虚部、X 相位或 Y 相位。</p>
照度 XY 扫描 (Illumination XY Scan)	<p>ILL_SOURCE: 光源大小。</p> <p>ILL_SMOOTH: 采用的平滑值。</p> <p>ILL_DETSIZE: 探测器大小。</p> <p>ILL_SURFACE: 表面编号。</p>
图像模拟 (Image Simulation)	<p>ISM_INPUTFILE: 输入文件名。指定的文件名应不含路径。</p> <p>ISM_FIELDHEIGHT: Y 视场高度。</p> <p>ISM_OVERSAMPLING: 过采样值。用 0 表示无, 1 表示 2X, 2 表示 4x, 以此类推。</p> <p>ISM_GUARDBAND: 安全宽度值。用 0 表示无, 1 表示 2X, 2 表示 4x, 以此类推。</p> <p>ISM_FLIP: 翻转源位图。用 0 表示无, 1 表示上下翻转 (TB), 2 表示 左右翻转 (LR), 3 表示 TB&LR。</p> <p>ISM_ROTATE: 旋转源位图:用 0 表示无, 1 表示 90 度, 2 表示 180 度, 3</p>

	<p>表示 270 度。</p> <p>ISM_WAVE: 波长。用 0 表示 RGB, 1 表示 1+2+3, 2 表示波长编号 1, 3 表示波长编号 2, 以此类推。</p> <p>ISM_FIELD: 视场编号。</p> <p>ISM_PSAMP: 光瞳采样: 用 1 表示 32x32, 2 表示 64x64 等。</p> <p>ISM_ISAMP: 像面采样: 用 1 表示 32x32, 2 表示 64x64 等。</p> <p>ISM_PSFY, ISM_PSFY: PSF 栅格点的数量。</p> <p>ISM_ABERRATIONS: 像差: 用 0 表示无, 1 表示几何, 2 表示衍射。</p> <p>ISM_POLARIZATION: 使用偏振: 用 0 表示否, 1 表示是。</p> <p>ISM_FIXEDAPERTURES: 使用固定孔径: 用 0 表示否, 1 表示是。</p> <p>ISM_USERI: 使用相对照度: 用 0 表示否, 1 表示是。</p> <p>ISM_SHOWAS: “显示为”: 用 0 表示仿真图, 用 1 表示光源位图, 用 2 表示 PSF 网格。</p> <p>ISM_REFERENCE: 参考: 用 0 表示主光线、1 表示顶点、2 表示初级主光线。</p> <p>ISM_SUPPRESS: 压缩框架: 用 0 表示否, 1 表示是。</p> <p>ISM_PIXELSIZE: 像元尺寸: 用 0 表示默认大小 (以镜头单位表示)。</p> <p>ISM_XSIZE, ISM_YSIZE: X 像素数, Y 像素数。用 0 表示默认像素数。</p> <p>ISM_FLIPIMAGE: 像面翻转: 用 0 表示无, 1 表示上下翻转 (TB) 等等。</p> <p>ISM_OUTPUTFILE: 输出文件名或者空字符串 (表示无输出文件)。</p>
MTF - FFT	<p>MTF_SAMP: 光瞳采样, 用 1 表示 32, 用 2 表示 64, 以此类推。</p> <p>MTF_WAVE: 波长编号, 用 0 表示全部。</p> <p>MTF_FIELD: 视场编号, 用 0 表示全部。</p> <p>MTF_TYPE: 用 0 表示调制, 1 表示实部, 2 表示虚部, 3 表示相位虚部, 4 表示方波。</p> <p>MTF_SURF: 表面编号, 用 0 表示像面。</p> <p>MTF_MAXF: 最大频率, 用 0 表示默认值。</p> <p>MTF_SDLI: 显示衍射极限, 0 表示否, 1 表示是。</p>

	<p>MTF_POLAR: 使用偏振, 0 表示否, 1 表示是。</p> <p>MTF_DASH: 使用虚线, 0 表示否, 1 表示是。</p>
<p>非序列物体查看器 (NSC Object Viewer)</p>	<p>SHA_ROT X: x 旋转 (以度为单位)。</p> <p>SHA_ROT Y: y 旋转 (以度为单位)。</p> <p>SHA_ROT Z: z 旋转 (以度为单位)。</p>
<p>NSC 实体模型 (NSC Shaded Model)</p>	<p>SHA_ROT X: x 旋转 (以度为单位)。</p> <p>SHA_ROT Y: y 旋转 (以度为单位)。</p> <p>SHA_ROT Z: z 旋转 (以度为单位)。</p>
<p>部分相干图像 分析 (Partially Coherent Image Analysis)</p>	<p>PCI_FIELD: 视场编号。</p> <p>PCI_FILESIZE: 文件大小。</p> <p>PCI_WAVE: 波长编号。</p> <p>PCI_RESAMPLE: 重新采样像面设置, 0 表示否, 1 表示是。</p> <p>PCI_RSN X: 重新采样数量 x</p> <p>PCI_RSN Y: 重新采样数量 y</p> <p>PCI_RSD C X: 重新采样 x 偏心</p> <p>PCI_RSD C Y: 重新采样 y 偏心</p> <p>PCI_RSD L X: 重新采样增量 x</p> <p>PCI_RSD L Y: 重新采样增量 y</p>
<p>偏振光瞳图 (Polarization Pupil Map)</p>	<p>PPM_SAMP: 采样, 用 0 表示 3x3, 1 表示 5x5, 2 表示 7x7, 以此类推。</p> <p>PPM_FIELD: 视场编号。</p> <p>PPM_WAVE: 波长编号。</p> <p>PPM_SURFACE: 表面编号。</p> <p>PPM_J X: Jx 振幅。</p> <p>PPM_J Y: Jy 振幅。</p> <p>PPM_P X: Px 相位。</p> <p>PPM_P Y: Py 相位。</p>

	PPM_ADDCONFIG: 干涉相加结构字符串。 PPM_SUBCONFIGS: 干涉相减结构字符串。
物理光学传播 - 常规标签 (Physical Optics Propagation - eneral Tab)	POP_END: 终止面。 POP_FIELD: 视场编号。 POP_START: 起始面。 POP_WAVE: 波长编号。
物理光学传播-I 光束定义标签 (Physical Optics Propagation - Beam Definition Tab)	POP_AUTO: 模拟按下“自动”按钮, 可根据采样和其它设置来自动选择合适的 X 和 Y 光束宽度。 POP_BEAMTYPE: 选择光束类型。用 0 表示高斯束腰, 1 表示高斯角, 2 表示高斯尺寸 + 角, 3 表示平顶, 4 表示文件, 5 表示 DLL, 6 表示多模。 POP_PARAMn: 设置光束参数 n, 例如, 使用 POP_PARAM3 以设置参数 3。 POP_PEAKIRRAD: 按照度峰值进行归一化设置。 POP_POWER: 按光束总功率进行归一化设置。 POP_SAMPX: X 方向采样, 用 1 表示 32, 2 表示 64, 以此类推。 POP_SAMPY: Y 方向采样, 用 1 表示 32, 2 表示 64, 以此类推。 POP_SOURCEFILE: 如果起始光束是由 ZBF 文件、DLL 或多模文件定义, 则 POP_SOURCEFILE 为文件名。 POP_WIDEX: X 方向宽度。POP_WIDEY: Y 方向宽度。
物理光学传播 - 光纤数据标签 (Physical Optics Propagation - Fiber Data Tab)	POP_COMPUTE: 使用 1 表示打开计算光纤耦合效率, 0 表示关闭。 POP_FIBERFILE: 文件名 (如果光纤模式是由 ZBF 或 DLL 定义)。 POP_FIBERTYPE: 使用与上文的 POP_BEAMTYPE 相同的值, 但尚不支持多模。 POP_FPARAMn: 设置光纤参数 n, 例如, 使用 POP_PARAM3 以设置光纤参数 3。 POP_IGNOREPOL: 用 1 可忽略偏振, 0 可考虑偏振。 POP_POSITION: 光纤位置设置。用 0 表示主光线、1 表示面顶点。 POP_TILTX: X 倾斜。

	POP_TILTY: Y 倾斜。
相对照度 (Relative Illumination)	REL_RAYDENSITY: 光线密度。 REL_FIELDDENSITY: 视场密度。 REL_WAVE: 波长编号, 用 0 表示全部。 REL_POLAR: 使用偏振: 用 1 表示使用偏振, 0 表示忽略偏振。 REL_LOG: 对数缩放: 用 1 表示对数缩放, 0 表示线性。 REL_REMOVEVIGNETTING: 使用 1 可移除渐晕因子, 否则使用 0。 REL_SCANTYPE: 扫描类型: 用 0 表示 +y, 1 表示 +x, 2 表示 -y, 3 表示 -x 扫描方向。
实体模型 (Shaded Model)	SHA_ROT X: x 旋转 (以度为单位)。 SHA_ROT Y: y 旋转 (以度为单位)。 SHA_ROT Z: z 旋转 (以度为单位)。
点 列 图(Spot Diagram)	SPT_RAYS: 光线密度。
表面矢高 (Surface Sag)	SRS_SAMP: 采样。用 1 表示 33x33, 2 表示 65x65, 以此类推。 SRS_SURF: 表面编号。
一维通用绘图 (Universal Plot 1D)	UN1_CATEGORY: 用 0 表示表面, 用 1 表示系统, 用 2 表示结构。 UN1_PARAMETER: 用 0 表示第一个选项, 用 1 表示第二个选项等。 UN1_SURFACE: 表面或结构编号。 UN1_STARTVAL: 变量的起始值。 UN1_STOPVAL: 变量的终止值。 UN1_STEPS: 步长。 UN1_OPERAND: 优化操作数名称。 UN1_MFLINE: 优化操作数行号。用 0 表示 MF 值。 UN1_PAR1: 操作数参数 1。 UN1_PAR2: 操作数参数 2。 UN1_PAR3: 操作数参数 3。

	<p>UN1_PAR4: 操作数参数 4。</p> <p>UN1_PAR5: 操作数参数 5。</p> <p>UN1_PAR6: 操作数参数 6。</p> <p>UN1_PAR7: 操作数参数 7。</p> <p>UN1_PAR8: 操作数参数 8。</p> <p>UN1_PLOTMIN: 因变量的最小图形值。</p> <p>UN1_PLOTMAX: 因变量的最大图形值。</p> <p>UN1_TITLE: 图形标题。</p>
<p>二维通用绘图 (Universal Plot 2D)</p>	<p>UN2_CATEGORYX: 用 0 表示表面, 用 1 表示系统, 用 2 表示结构。</p> <p>UN2_PARAMETERX: 用 0 表示第一个选项, 用 1 表示第二个选项等。</p> <p>UN2_SURFACEX: 表面或结构编号。</p> <p>UN2_STARTVALX: 变量的起始值。</p> <p>UN2_STOPVALX: 变量的终止值。</p> <p>UN2_STEPSX: 步长。</p> <p>UN2_CATEGORYY: 用 0 表示表面, 用 1 表示系统, 用 2 表示结构。</p> <p>UN2_PARAMETERY: 用 0 表示第一个选项, 用 1 表示第二个选项等。</p> <p>UN2_SURFACEY: 表面或结构编号。</p> <p>UN2_STARTVALY: 变量的起始值。</p> <p>UN2_STOPVALY: 变量的终止值。</p> <p>UN2_STEPSY: 步长。</p> <p>UN2_OPERAND: 优化操作数名称。</p> <p>UN2_MFLINE: 优化操作数行号。用 0 表示 MF 值。</p> <p>UN2_PAR1: 操作数参数 1。</p> <p>UN2_PAR2: 操作数参数 2。</p> <p>UN2_PAR3: 操作数参数 3。</p> <p>UN2_PAR4: 操作数参数 4。</p> <p>UN2_PAR5: 操作数参数 5。</p>

	UN2_PAR6: 操作数参数 6。 UN2_PAR7: 操作数参数 7。 UN2_PAR8: 操作数参数 8。 UN2_SHOWAS: 数据显示。用 0 表示表面，用 1 表示等高线等等。 UN2_CONTOURFORMAT: 等高线格式字符串。 UN2_PLOTMIN: 因变量的最小图形值。 UN2_PLOTMAX: 因变量的最大图形值。 UN2_TITLE: 图形标题。
波前图 (Wavefront Map)	WFM_SAMP: 采样，用 1 表示 32，用 2 表示 64 等。 WFM_FIELD: 视场编号。 WFM_WAVE: 波长编号。 WFM_SUBSR: 子孔径半径。 WFM_SUBSX: 子孔径 X 偏心。 WFM_SUBSY: 子孔径 Y 偏心。

value 参数是指定设置的新数据。修改后的设置文件将替换原设置文件。

另请参阅 GETTEXTFILE。

示例:

```
MODIFYSETTINGS "C:\MySPT.CFG", SPT_RAYS, 24
MODIFYSETTINGS "C:\MyPOP.CFG", POP_SOURCEFILE, "MyStartBeam.ZBF"
```

NEXT

请参阅“FOR, NEXT”。

NSLT

开始非序列闪电追迹。

语法:

```
NSLT surf, source, ray_sampling, edge_sampling
```

详述:

surf 是一个整数值，在混合序列/非序列模式下，表示非序列组件表面的编号。在非序列模式下，surf 设为 1。source 是指所需光源的物体编号。如果 source 设为 0，则会追迹所有光源。ray_sampling 指的是闪电追迹网格的分辨率，有效值在 0（对应于网格分辨率“低 (1X)”）到 5（对应于网格分辨率“1024X”）之间。

edge_sampling 指在物体边缘附近进行闪电追迹时使用的网格分辨率，有效值在 0（对应于网格分辨率“低 (1X)”）到 4（对应于网格分辨率“256X”）之间。有关闪电追迹的更多详细信息，请参阅[闪电追迹控制](The LightningTrace Control)。

NSLT 在执行闪电追迹之前始终调用 UPDATE，以确保所有物体都已经正确加载和更新。

示例:

```
NSLT 1, 0, 3, 2
```

NSTR

开始非序列光线追迹。

语法:

```
NSTR surf, source, split, scatter, usepolar, ignore_errors, random_seed, save,
savefilename, filter, zrd_format
```

详述:

surf 是一个整数值，在混合序列/非序列模式下，表示非序列组件表面的编号。在非序列模式下，surf 设为 1。source 是指所需光源的物体编号。如果 source 设为 0，则会追迹所有光源。如果 split 不为 0，则开启光线分裂，否则关闭光线分裂。如果 scatter 不为 0，则开启散射，否则关闭散射。如果 use_polarization 不为 0，则使用偏振，否则关闭偏振。如果开启光线分裂，将自动选择偏振。如果 ignore_errors 不为 0，则忽略几何错误，若 ignore_errors 设为 0，在光线发生错误时，将终止非序列追迹和宏执行，并且出现错误信息提示。

如果 random_seed 为 0，则随机数生成器将使用随机数做种子，并且每次调用 NSTR 都将生成不同的随机光线。如果 random_seed 是 0 以外的任何整数，则随机数生成器将使用指定数值做种子，并且每次调用 NSTR 都将生成相同的光线。使用 NSTR 进行优化时，建议将非 random_seed 设为非 0 值。

如果 save 省略或者为 0，则不需要提供变量 savefilename、filter 和“zrd 格式”。如果 save 不为 0，则光线将保存到某个文件中。保存的数据文件名称为 savefilename 所指定的名称。文件命名规范和目标文件夹与[将光线数据保存到文件](Saving ray data to a file)中描述的相同。应提供 savefilename 的扩展名，但不包含路径。如果 filter 不为 0，则可选的 filter 名称可以是包含过滤的字符串变量或者双引内的过滤字符。如果不使用过滤，则输入一对空的双引号，比如：“”。有关过滤字符串的更多信息，请参阅[过滤字符串](The filter string)。对于 ZRD 文件，zrd_format 可以是 0、1 或 2，分别表示不压缩全部数据、压缩基本数据或压缩全部数据。有关 ZRD 格式的更多信息，请参阅[光线数据库 (ZRD) 文件](Ray database (ZRD) files)。

NSTR 在进行光线追迹之前始终调用 UPDATE，以确保所有物体都已经正确加载和更新。

相关函数:

NSDD

示例:

```
NSTR 1, 0, 0, 0, 0, 1, 0, 1, "saverays.ZRD", "h2"
```

NUMFIELD

该命令已被废弃。请参阅“SETSYSTEMPROPERTY、SYSP”。

NUMWAVE

该命令已被废弃。请参阅“SETSYSTEMPROPERTY、SYSP”。

OPEN

打开一个现有的文本文件，以供 READ 命令进行读取。

语法:

```
OPEN "filename"  
OPEN A$
```

详述:

提供的文件名可以是双引号内的有效文件，或者包含文件名的字符串变量。请参阅关键字 READ 和 CLOSE。通常在读取所有数据之后会使用 CLOSE 关闭文件。

相关函数:

EOFF

相关关键字:

READ、READNEXT、CLOSE

示例:

```
PRINT "Reading the double-column file TEST.DAT!"  
OPEN "TEST.DAT"  
READ x1, y1  
READ x2, y2  
READ x3, y3  
CLOSE
```

OPENANALYSISWINDOW

打开新的分析窗口。

语法:

```
OPENANALYSISWINDOW type, settingfilename
```

详述:

type 变量是一个 3 字符的字符串代码，指定要执行的分析类型。如需字符串代码的完整列表，请参阅[编程菜单](Programming Tab)的[字符串代码](String Code)栏目。如果未提供或者无法识别字符串代码，则不会打开任何窗口。settingfilename 包含要使用的 CFG 设置文件的名称。如果缺少此变量，则会使用默认设置。

相关函数:

WINL、WINN

相关关键字:

CLOSEWINDOW、MODIFYSETTINGS

示例:

```
FORMAT 0.0
PRINT "There are ", WINN(), " Windows currently open."
OPENANALYSISWINDOW "mtf"
PRINT "Just opened window ", WINL(), "."
PAUSE TIME, 3000
PRINT "There are ", WINN(), " Windows currently open."
CLOSEWINDOW WINL()
PAUSE TIME, 1000
PRINT "There are ", WINN(), " Windows currently open."
```

OPTIMIZE(keywords)

调用优化算法，使用当前评价函数来优化当前镜头。

语法:

```
OPTIMIZE
OPTIMIZE number_of_cycles
OPTIMIZE number_of_cycles, algorithm
```

详述:

number_of_cycles 的表达式必须为 1 到 99 之间的整数值，表示优化算法将运行的周期数。如果 number_of_cycles 计算结果为 0，则优化将以“自动”模式运行，并且在算法检测到过程已收敛时停止。对于 algorithm 变量，使用 0 表示采用阻尼最小二乘法（默认值），使用 1 表示采用正交

下降法。有关更多信息，请参阅[执行优化] (Performing an optimization)。

使用 **MFCN** 函数可在不优化的情况下更新评价函数。

相关函数:

MFCN

示例:

```
PRINT "Starting merit function:", MFCN()
OPTIMIZE
PRINT "Ending merit function :", MFCN()
```

OPTRETURN

通过使用 **ZPLM** 优化操作数将数值返回到优化算法。

语法:

```
OPTRETURN datafield, result
```

详述:

datafield 表达式的计算结果必须为 0 到 50 之间的整数。datafield 指数组中可以存储表达式结果的位置。**OPTRETURN** 的唯一用途是能够优化在 **ZPL** 宏中计算的数值。

在评价函数中必须使用优化操作数 **ZPLM** 来调用 **ZPL** 宏，并通过 **OPTRETURN** 得到返回的数值。有关详细信息，请参阅[通过 **ZPL** 宏优化] (Optimizing with ZPL macros)。

示例:

```
x = sqrt(thic(3) + radi(5))
OPTRETURN j, x+5
```

OUTPUT

指定文本输出的目标。可以输出到屏幕上或文件中。

语法:

```
OUTPUT SCREEN
OUTPUT filename
OUTPUT filename, APPEND
```

详述:

如果仅指定了 **OUTPUT SCREEN**，则所有后续执行的 **PRINT** 命令都将输出到屏幕上。如果提供了一个有效的 filename，则后续 **PRINT** 命令将输出到指定的文件中。如果提供了不含路径的文件名，则文件将保存在默认 的<data>\Macros 文件夹。若要关闭之前创建的文件，可使用关键字 **OUTPUT**

SCREEN，此命令可将后续 **PRINT** 内容直接输出到屏幕上。**SHOWFILE** 将关闭文件，并将文件发送到文本查看器程序，以在屏幕上显示。**PRINTFILE** 将关闭文件，并在当前定义的打印设备上将文件打印出来。需注意，使用 **PRINT** 命令输出文件时，文件为 **Unicode** 格式。若要将文件转换为 **ANSI** 格式，可先关闭文件（例如，使用 **OUTPUT SCREEN**），然后使用关键字 **CONVERTFILEFORMAT**。

如果 filename 后面跟随有关键字 **APPEND**，则后续输出都将附加到该文件中。否则，将覆盖原文件的内容。

示例：

```
OUTPUT "x.txt"
PRINT "This will not appear on the screen, but in the file x.txt."
OUTPUT SCREEN
PRINT "This will appear on the screen."
OUTPUT "x.txt", APPEND
PRINT "This will appear after the first line in the file x.txt."
```

相关关键字：

CLOSE, OPEN, SHOWFILE, PRINTFILE

PARM

该命令已被废弃。请参阅“**SETSURFACEPROPERTY、SURP**”。

PARAXIAL(keywords,programming tab,about the zpl)

用来控制进行光线追迹是近轴光线还是实际光线。

语法：

```
PARAXIAL ON
PARAXIAL OFF
```

详述：

可通过调用函数 **PMOD** 来确定当前近轴模式，如果近轴模式关闭，则 **PMOD** 返回 0，否则返回 1。

PARAXIAL 用于在实际光线追迹与近轴光线追迹之间转换。某些计算需要对近轴光线进行追迹，例如测定畸变和计算系统有效焦距等一阶特性。

示例：

```
mode = PMOD()
IF mode THEN PRINT "Paraxial mode is on!"
IF !mode THEN PRINT "Paraxial mode is not on!"
PARAXIAL ON
PRINT "Now paraxial mode is on!"
PRINT "Restoring original mode..."
```



```
if !mode THEN PARAXIAL OFF
```

PAUSE

暂停宏执行，同时可选择性地显示状态消息。状态消息可以是字符串或数值。在用户点击状态对话框上的“确定”按钮之后，宏继续运行。

语法：

```
PAUSE  
PAUSE "Ready to continue..."  
PAUSE TIME, time  
PAUSE THREADS
```

详述：

此功能可用于调试、显示结果或暂停宏的执行。如果 **PAUSE** 命令之后跟随关键字 **TIME**，则宏执行将在指定的毫秒数内处于休眠状态。例如，要暂停宏 100 毫秒，则语法为

```
PAUSE TIME, 100
```

指定时间结束后，宏将继续执行。此种情况下不会显示状态消息。请注意，系统计时器的精度大约为 10 毫秒，因此 **PAUSE TIME** 所产生的实际延迟介于请求时间与请求时间加上（大约）10 毫秒之间。允许的最大暂停时间是 1,000,000 毫秒（大约 16.7 分钟）。

如果 **PAUSE** 命令之后跟随关键字 **THREADS**，则在当前进行计算的所有窗口都计算完毕之后，宏执行才暂停。当使用 **session** 文件调用 **UPDATE ALL** 或 **LOADLENS** 时，此功能非常有用。

PIXEL

在当前图形窗口上显示单个像素。

语法：

```
PIXEL x 坐标, y 坐标
```

详述：

此功能用于生成点列图。请参阅“图形(GRAPHICS)”。

PLOT

关键字 **PLOT** 支持多个变量，用于简化创建数据图形的任务。

语法:

```
PLOT                                                    NEW
PLOT                                                    TITLE,          string
PLOT                                                    TITLEX,         string
PLOT                                                    TITLEY,         string
PLOT                                                    BANNER,         string
PLOT                                                    WINASPECT,      type
PLOT                                                    COMM1,          string
PLOT                                                    COMM2,          string
PLOT                                                    COMM3,          string
PLOT                                                    COMM4,          string
PLOT                                                    COMM5,          string
PLOT                                                    COMM6,          string
PLOT                                RANGEX,          minx,          maxx
PLOT                                RANGEY,          miny,          maxy
PLOT                                CHECK,          x_increment,  y_increment
PLOT                                TICK,          x_increment,  y_increment
PLOT                                FORMATX,         format_string
PLOT                                FORMATY,         format_string
PLOT DATA,    x_array,    y_array,    number_of_points,    color,    style,    options
PLOT                                LINE,          x1,          y1,          x2,          y2
PLOT                                LABEL,          x,          y,          angle,          size,          string
PLOT GO
```

详述:

PLOT NEW 用于初始化新绘图。所有之前图形相关的数据都会废弃。在创建新的图形时，**PLOT NEW** 总是第一个 **PLOT** 命令。

PLOT TITLE、**PLOT TITLEX** 和 **PLOT TITLEY** 分别用于定义整个图形标题、X 轴标题和 Y 轴标题。string 可以是引号中的文本字符串，也可以是字符串变量。

PLOT BANNER 用于定义横幅标题。string 可以是引号中的文本字符串，也可以是字符串变量。

PLOT WINASPECT 定义所显示窗口的纵横比。type 的有效值为整数 0 到 3，对应生成的纵横比分别为 4x3、5x3、3x4 和 3x5。如果没有 **WINASPECT** 命令，则使用默认纵横比。

PLOT COMMx 用于定义在图形底部显示注释（最多 6 条）。string 参数可以是引号中的文本字符串，也可以是字符串变量。

PLOT RANGEX 和 **PLOT RANGEY** 可分别定义图形中 X 方向和 Y 方向的最小值和最大值。如果没有 **RANGEX** 或 **RANGEY** 命令，则选择默认范围。

PLOT CHECK 用于定义“X”标记的大小，“X”表示在 **PLOT DATA** 命令中标记的数据点。单位是相对于显示宽度的百分数。如果没有 **CHECK** 命令，则选择默认大小。

PLOT TICK 分别用于定义 X 轴和 Y 轴上相邻数据之间的增量。如果没有 **TICK** 命令，则将选择默认增量。

PLOT FORMATX 和 **PLOT FORMATY** 分别用于定义 X 轴和 Y 轴上数值标签的字符串格式。format_string 参数可以是引号中的文本字符串，也可以是字符串变量。format_string 必须是有效的 C 语言格式。常见的 format_string 是“%M.nf”，其中 M 是输出字符串中空格的总数，n 是小数点后面的位数。例如，如果 format_string 为“%6.4f”，则 pi 的值将输出为“3.1415”。若

要定义指数格式，format_string 应为 “%M.nE” 形式，其中 M 是输出的总字符数量，n 是小数点后面的位数。如果 format_string 为 “%9.3E”，则 pi 的值将输出为 “3.14E+000”。如果未指定 FORMATX 和 FORMATY，则将使用默认格式。

PLOT DATA 用于定义一系列要绘制的数据点。变量 x_array 和 y_array 必须是一维数组变量。有关定义数组变量的信息，请参阅[数组变量](Array variables)。number_of_points 定义数组中使用的点数。变量 color 的值为 0 表示黑色，1 到 16 可选择不同的画笔颜色。更多画笔颜色的信息，请参阅[颜色 1-12, 颜色 13-24](Colors 1-12, Colors 13-24)。style 定义线条样式，当前支持的值为 0 或者 1 到 4，0 表示实线，1 到 4 分别表示不同样式的虚线。options 值为 0 表示没有数据点的位置标记，1 表示同时用直线和数据点标记，2 表示没有直线，仅有数据点标记。可以使用多个 PLOT DATA 命令，每个命令都会单独创建一个要绘制的直线或曲线。

PLOT LINE 在点 (x1, y1) 与 (x2, y2) 之间绘制一条直线。此处 x 和 y 的单位是归一化的窗口坐标。最多可以定义 1000 条直线。

PLOT LABEL 输出图形上从坐标 (x, y) 开始的任何文本字符串。此处 x 和 y 的单位是归一化的窗口坐标。angle 表示与 +x 方向的角度（以度为单位）。size 表示大于 0.0 的比例常数，用于定义字体的大小。size 为 1.0 将以正常字体输出，size 为 1.5 表示输出字体比正常字体大 50%。string 可以是引号中的文本字符串，也可以是字符串变量。最多可以定义 100 个标签。

PLOT GO 使用先前 PLOT 命令的所有设置和数据生成所需图形，并在窗口中显示图形。在生成图形后，所有 PLOT 数据都将重置（尽管已经执行了 PLOT NEW 命令）。

相关关键字:

PLOT2D

PLOT2D

关键字 PLOT2D 支持多个变量，这些变量用于简化创建数据图形的任务。通过此关键字，可轻易生成表面、等高线、灰度图和伪彩色图。

语法:

```

PLOT2D                                                    NEW
PLOT2D                                                    TITLE,                string
PLOT2D                                                    COMM1,               string
PLOT2D                                                    COMM2,               string
PLOT2D                                                    COMM3,               string
PLOT2D                                                    COMM4,               string
PLOT2D                                                    COMM5,               string
PLOT2D                                RANGE,                min,                max
PLOT2D                                                    ASPECT,              ratio
PLOT2D                                WINASPECT,            type
PLOT2D                                DATA,                arrayname
PLOT2D                                ACTIVECURSOR,         left,                right,                bottom,                top
PLOT2D                                DISPLAYTYPE,          type
PLOT2D                                CONTOURINTERVAL,      string
PLOT2D                                SURFACESCALE,         scale

```

```

PLOT2D                                LOGPLOT,                                peak,                                decades
PLOT2D                                HIDEADDRESS,                                flag
PLOT2D                                CONFIG,                                configuration
PLOT2D GO

```

详述:

PLOT2D NEW 用于初始化新绘图。所有之前 **plot2d** 图形相关的数据都会废弃。在创建新的图形时，**PLOT2D NEW** 总是第一个 **PLOT 2D** 命令。

PLOT2D TITLE 用于定义主图形标题。 **string** 可以是引号中的文本字符串，也可以是字符串变量。

PLOT2D COMMx 用于定义在图形底部显示注释（最多 5 条）。 **string** 可以是引号中的文本字符串，也可以是字符串变量。

PLOT2D RANGE 可定义图形的最小值和最大值。如果没有 **RANGE** 命令，则将使用默认范围。范围应涵盖所有数据值，否则将截断数据以适应范围。表面和等高线图形将忽略此命令。

PLOT2D ASPECT 定义图形的 X 宽度与 Y 宽度之比。表面图形将忽略此命令。

PLOT2D WINASPECT 定义所显示窗口的纵横比。 **type** 的有效值为整数 0 至 3，对应生成的纵横比分别为 4x3、5x3、3x4 和 3x5。如果没有 **WINASPECT** 命令，则使用默认纵横比。

PLOT2D DATA 用于定义所绘图形的二维数据点。 **arrayname** 必须是二维数组变量。有关定义数组变量的信息，请参阅[数组变量](Array variables)。 **x** 和 **y** 方向的点的数量通过数组维数来定义。允许任何点数，每个方向上最小的点数为 5。假定两个维度的数据构成一个矩形网格，两个方向上的数据间距均匀，虽然图形的总体纵横比已通过 **PLOT2D ASPECT** 定义。仅能使用一个 **PLOT2D DATA** 命令。

PLOT2D ACTIVECURSOR 用于定义等高线、灰度图和伪彩色图上显示数据的上下左右边界。

PLOT2D DISPLAYTYPE 用于选择所生成的图形的类型。 **type** 的值应该是 1 到 6 之间的整数（包括首尾值），分别表示表面、等高线、灰度图、反灰度图、伪彩色图和反伪彩色图。

PLOT2D CONTOURINTERVAL 用于定义等高线格式字符串。仅等高线图使用此命令。有关等高线间隔格式字符串的详细信息，请参阅[等高线格式字符串](The Contour Format String)。

PLOT2D SURFACESCALE 用于定义表面图形的总体垂直缩放。仅表面图使用此命令。默认的缩放值是 0.5 倍

PLOT2D LOGPLOT 用于显示对数比例图。仅灰度图、反灰度图、伪彩色图和反伪彩色图使用此命令。 **peak** 是一个整数，表示要绘制的最大值是 10 的几次方。例如，如果 **peak** 为 3，则所绘制的最大值将为 $1\text{E}+03$ 。 **decades** 值是一个整数，表示从峰值到谷值下降的 10 的幂数。例如，如果 **peak** 为 3， **decades** 为 5，则绘制的范围将是 $1.0\text{E}+03$ 到 $1.0\text{E}-02$ 。请注意，使用 **LOGPLOT** 不会自动提取输入数组中数据的对数。因此，在调用 **PLOT2D GO** 之前，宏本身必须提取数组中 10 的对数值。

PLOT2D HIDEADDRESS 用于隐藏或显示地址数据。如果 **flag** 值为 0，则显示地址，否则不显示地址。另请参阅[地址](Address)。

PLOT2D CONFIG 用于定义显示的多重结构编号，但前提条件是地址框显示当前结构的信息。如果 **configuration** 为 0，则地址框将显示当前结构编号。如果结构是介于 1 到多重结构数量之间的整

数（包含首尾值），则将显示指定结构编号。如果 configuration 是小于 0 的整数，则显示所有结构。

PLOT2D GO 使用先前 PLOT2D 命令的所有设置和数据生成所需图形，并在窗口中显示图形。在生成图形后，所有 PLOT2D 数据都将重置（尽管已经执行了 PLOT2D NEW 命令）。

相关关键字：

PLOT

POLDEFINE

为后续调用 POLTRACE 命令定义输入的偏振状态。

语法：

```
POLDEFINE Jx, Jy, PhaX, PhaY
```

详述：

关键字 POLDEFINE 用于为后续偏振光追迹定义输入的偏振状态。POLDEFINE 需要 Jx 和 Jy 电场强度以及 X 和 Y 相位角（以度为单位）4 个变量。有关 Jx 和 Jy 含义的更多信息，请参阅[定义初始偏振](Defining the initial polarization)。输入的变量值会自动归一化，用单位强度表示。4 个变量默认值分别为 0、1、0 和 0。一旦定义了偏振状态，则会保持不变，除非进行更改。

示例：

```
POLDEFINE 2.0, 2.0, 45.0, -66.0
```

相关关键字：

POLTRACE

POLTRACE

调用 OpticStudio 偏振光线追迹程序，并通过当前系统来追迹特定光线。

语法：

```
POLTRACE Hx, Hy, Px, Py, wavelength, vec, surf
```

详述：

表达式 Hx 和 Hy 的值必须为 -1 到 1 之间，表示归一化的视场坐标。光瞳坐标通过表达式 Px 和 Py 来指定，必须介于 -1 到 1 之间。有关归一化坐标的更多信息，请参阅[约定和定义](Conventions and Definitions)一章中的[归一化视场和光瞳坐标](Normalized field and pupil coordinates)部分。wavelength 表达式的计算结果必须为 1 到已定义波长数量之间的整数。vec 表达式的计算结果必须为 1 到 4 之间的整数（包含首尾值）。surf 表达式的计算结果必须为 1 到表面数之间的整数（包

含首尾值)。

光线的输入偏振状态是由关键字 **POLDEFINE** 定义的。

一旦对光线追迹，光线的偏振数据将会存储于 **vec** 表达式指定的矢量变量中。例如，如果执行了命令 “**POLTRACE Hx, Hy, Px, Py, w, 2, n**”，则数据将存储在 **VEC2** 中。数据用以下格式存储，其中每行的第一个数字表示数组中的位置：

- 0: n, 指定矢量数组中数据的数量
- 1: 通过指定表面后的光线强度
- 2: 电场 X 分量, 实部
- 3: 电场 Y 分量, 实部
- 4: 电场 Z 分量, 实部
- 5: 电场 X 分量, 虚部
- 6: 电场 Y 分量, 虚部
- 7: 电场 Z 分量, 虚部
- 8: 反射光中 S 偏振光的电场振幅, 实部
- 9: 反射光中 S 偏振光的电场振幅, 虚部
- 10: 透射光中 S 偏振光的电场振幅, 实部
- 11: 透射光中 S 偏振光的电场振幅, 虚部
- 12: 反射光中 P 偏振光的电场振幅, 实部
- 13: 反射光中 P 偏振光的电场振幅, 虚部
- 14: 透射光中 P 偏振光的电场振幅, 实部
- 15: 透射光中 P 偏振光的电场振幅, 虚部
- 16: 电场 X 方向相位 P_x
- 17: 电场 Y 方向相位 P_y
- 18: 电场 Z 方向相位 P_z
- 19: 椭圆偏振的长轴长度
- 20: 椭圆偏振的短轴长度
- 21: 椭圆偏振的角度 (以弧度为单位)
- 22: 渐晕光线所在的表面编号, 如果没有渐晕, 则为 0
- 23: 反射光中 S 偏振光的光线振幅, 实部
- 24: 反射光中 S 偏振光的光线振幅, 虚部
- 25: 透射光中 S 偏振光的光线振幅, 实部
- 26: 透射光中 S 偏振光的光线振幅, 虚部
- 27: 反射光中 P 偏振光的光线振幅, 实部
- 28: 反射光中 P 偏振光的光线振幅, 虚部
- 29: 透射光中 P 偏振光的光线振幅, 实部
- 30: 透射光中 P 偏振光的光线振幅, 虚部

如果数组位置 0 中的值为 0, 则表示产生错误, 且偏振数据无效。当指定光线不能被追迹时, 可能出现这种情况。关于提取扩展错误信息的更多详情, 请参阅 **RAYTRACE** 命令。

示例:

```
POLDEFINE 0, 1, 0, 0
POLTRACE 0, 1, 0, 0, pwav(), 1, nsur()
PRINT "Transmission of chief ray at primary wavelength is ", vec1(1)
```

相关关键字:

POLDEFINE, RAYTRACE

POP

计算系统光线的物理光学传播 (POP) ，并将每个表面的结果保存到 ZBF 文件中。有关 POP 功能的描述，请参阅[物理光学传播](Physical Optics Propagation)。关于 ZBF 文件格式的信息，请参阅[Zemax 光束文件 (ZBF) 二进制格式](Zemax Beam File (ZBF) binary format)。

语法:

```
POP outfilename, lastsurface, settingsfilename
```

详述:

此关键字需要三个变量，输出 ZBF 文件的名称 **outfilename**，表示传播到最后面的表达式 **lastsurface**，以及可选变量 **settingsfilename**。如果输出文件名中使用了空格或其它特殊字符，则文件名必须加引号。创建的 ZBF 文件将存储于 <pop> 文件夹中。输出文件名不能包含路径。

若没有提供 **settingsfilename**，则 POP 功能的设置采用之前为当前镜头保存的设置。**settingsfilename** 必须包含完整路径、名称和扩展名。要更改这些设置，请打开“POP”窗口并选择适当的设置，然后点击[保存](Save)。默认情况下，ZPL 中 POP 的所有后续调用都会使用保存的设置。但 **outfilename** 和 **lastsurface** 编号例外，**outfilename** 为 POP 关键字之后的第一个变量，**lastsurface** 为 POP 关键字之后的第二个可选变量。

示例:

```
POP "pop_output.ZBF", 12, "MyPOP.CFG"
```

PRINT

Print 用于将常量文本和变量数据输出到屏幕上或文件中，这取决于关键字 OUTPUT 的当前状态。

语法:

```
PRINT  
PRINT X  
PRINT "The value of x is ", x  
PRINT " x = ", x, " x + y = ", x + y
```

详述:

单独的 PRINT 只能输出一行空白。包含文本变量和表达式列表的 PRINT 将输出双引号中的每个文本字符串，以及每个表达式的数值。PRINT 命令采用由 FORMAT 所指定的数值输出格式。如果

PRINT 列表中的最后一项后面带有逗号，则 PRINT 将不会以回车符结束这一行。

示例:

```
X = 3
PRINT "X equals ",x
```

相关关键字:

REWIND

PRINTFILE

打印文本文件。

语法:

```
PRINTFILE filename
```

详述:

filename 必须是有效的文件名。文件必须为文本文件（如在 ZPL 中通过 OUTPUT 和 PRINT 命令创建的文件），且必须在当前文件夹中。即使没有执行任何 CLOSE 命令，PRINTFILE 也会关闭这个文件。

示例:

```
OUTPUT "test.txt"
PRINT "Print this to the printer."
PRINTFILE "test.txt"
```

相关关键字:

OPEN、OUTPUT、CLOSE、PRINT、PRINTFILE

PRINTWINDOW

打印任何打开的图形窗口或文本窗口。

语法:

```
PRINTWINDOW winnum
```

详述:

winnum 可为整数或运算结果为整数的表达式。winnum 对应于应打印的窗口编号。OpticStudio 以窗口打开的顺序对窗口进行编号，从 1 开始。若用户关闭某些窗口，则 OpticStudio 会从列表中删除已关闭的窗口，但不会对余下的窗口重新编号。重新打开的任何窗口将会使用最小的可用窗口编号。

示例:

```
PRINTWINDOW 5
```


PWAV

设置主波长。

语法:

```
PWAV n
```

详述:

对表达式 *n* 求值，并且将求出的指定序号波长设置为主波长。在新数据生效之前必须执行 UPDATE 命令。

示例:

```
PWAV
```

相关函数:

WAVL、WWGT、PWAV

QUICKFOCUS(keywords)

调整像面之前的表面的厚度，快速聚焦。

语法:

```
QUICKFOCUS mode, centroid
```

详述:

mode 的表达式的计算结果必须为 0、1、2 或 3，分别表示 RMS spot radius、spot x、spot y 或 wavefront OPD。centroid 的表达式的计算结果必须为 0 或 1，分别表示 RMS 以主光线或图像质心作为参考。选择 “best” 焦点作为所有视场的波长加权平均值。

示例:

```
!Focus at best RMS wavefront to centroid  
QUICKFOCUS 3, 1
```

RADI

该命令已被废弃。请参阅 “SETSURFACEPROPERTY、SURP”。

RANDOMIZE

RANDOMIZE 可作为随机数生成器的种子。

语法:

```
RANDOMIZE seed
```

详述:

如果 seed 的计算结果为 0，则 OpticStudio 将使用基于 CPU 时钟的值来作为随机数生成器的种子。否则，提供的值作为随机数生成器的种子。使用相同种子将产生与 RAND 函数相同的随机数系列。

示例:

```
RANDOMIZE  
RANDOMIZE 250
```

相关函数:

RAND

RAYGENERATOR

运行光线生成器，用于通过 Radiant Source Model™ 文件创建一个光源光线文件。

语法:

```
RAYGENERATOR rsmx_file, ray_file, num_rays, min_azimuth, max_azimuth, min_inclination,  
max_inclination, origination_method, origination_radius, spectrum_type, Arg1, Arg2,  
Arg3
```

详述:

RAYGENERATOR 可通过 Radiant Source Model（由输入变量“rsmx 文件”指定）来创建光谱颜色格式(.SDF)的光线文件（请参阅[光源文件](Source File)）。光源模式是由输入变量“rsmx_file”指定的。光线文件的输出名称是通过输入变量“ray_file”来给定的。两个文件名都必须加引号，并包含正确的扩展名，但不包含路径。RSMX 文件必须位于 <data>\Sources\Radiant Source Model Files 文件夹中。输出光线文件将存储于 <data>\Sources\Source Files 文件夹中。

输出光线文件包含的光线数量由“num_rays”定义，分布在由 min_azimuth, max_azimuth, min_inclination 和 max_inclination 四个变量所定义的角度范围内。方位角必须介于 0 到 360 度之间。倾角必须介于 0 到 180 度之间。origination_method 是一个介于 0 到 11 的数字，对应于光线生成器的下拉菜单中可用的 12 个选项。0 表示法向平面，1 表示最接近 X 轴等等。有关更多选项，请参阅光线生成器对话框中的列表（“生成 Radiant Source Model 光线”）。

仅当 origination_method 为 4（在球面上）、5（在 X 柱面上）、6（在 Y 柱面上）、7（在 Z 柱面上）或 11（在长方体上）时，才会使用 origination_radius（始终以毫米为单位进行定义）。当 origination_method 为 4、5、6 或 7 时，origination_radius 用于定义产生光线的球面或柱面的半径。当 origination_method 为 11 时，光线从立方体中心产生，此时 origination_radius 用于定义立方体的半宽。在使用光线生成器对话框创建光线文件时，提供了定义圆柱光源的锥角（端盖），或者矩形光源的偏移量选项。但目前 RAYGENERATOR 关键字没有提供这些选项。如果如果需要向关键字 RAYGENERATOR 中增加光线生成器对话框中提供的任何选项，请联系 OpticStudio 技术支持部门。光谱类型由 spec_type 定义：测量 = 0，光谱文件 = 1，黑体 = 2，窄带 = 3。其余变量取决于光谱类

型。如果光谱类型不使用变量，则只需省略。测量的光谱类型不使用任何其它变量。光谱文件的光谱类型由 **Arg1** 来定义输入光谱文件 (*.SPCD)。文件名应加引号，但不必指定路径。文件必须位于 <data>\Sources\Spectrum Files 中。黑体光谱类型使用 **Arg1** 来定义黑体温度（以开尔文为单位），使用 **Arg2** 定义最小波长，使用 **Arg3** 定义最大波长。窄带光谱类型使用 **Arg1** 来定义中心波长，使用 **Arg2** 来定义光谱宽度。

RAYTRACE

调用 OpticStudio 光线追迹程序，通过当前系统来追迹特定光线。

语法:

```
RAYTRACE hx, hy, px, py, wavelength
```

详述:

表达式 **Hx** 和 **Hy** 的计算结果必须为 -1 到 1 之间，表示归一化的视场坐标。表达式 **Px** 和 **Py** 表示归一化的光瞳坐标，必须介于 -1 到 1 之间。有关归一化坐标的更多信息，请参阅[归一化视场坐标](Normalized field coordinates)。 **wavelength** 变量是可选的，默认为主波长。但如果提供 **wavelength** 表达式，则其结果必须是介于 1 和所定义最大波长数量之间的一个整数。

一旦开始追迹光线，则可使用 ZPL 函数 **RAYX**、**RAYY**、**RAYZ**、**RAYL**、**RAYM** 和 **RAYN** 确定光线交点坐标和方向余弦（在全局坐标中，使用 **RAGX**、**RAGY**、**RAGZ**、**RAGL**、**RAGM** 和 **RAGN** 获取数据）。如果在光线追迹期间产生错误，函数 **RAYE**（表示光线错误）将返回一个非 0 值。如果 **RAYE** 返回负值，则表明在表面上发生了全内反射，此表面编号由返回值的绝对值决定。如果 **RAYE** 返回值为 -9999，则表明无法启动光线追迹。

如果 **RAYE** 返回大于 0 的值，则表明返回光线错过的表面编号。检查光线错误的函数 **RAYE** 是可选的。如果 **RAYE** 不为 0，**RAYX**、**RAYY** 等函数可能会返回无效数据。函数 **RANX**、**RANY** 和 **RANZ** 返回光线相交表面法线的方向余弦，**OPDC** 返回光线的光程差。函数 **RAYV** 返回光线产生渐晕的表面编号，如果光线无渐晕，则 **RAYV** 返回 0。光线通过渐晕表面返回的值可能不准确。

示例:

```
PRINT "Tracing the marginal ray at primary wavelength!"
n = NSUR()
RAYTRACE 0,0,0,1
y = RAYY(n)
PRINT "The ray intercept is ", y
PRINT "Tracing the chief ray at maximum wavelength!"
RAYTRACE 0,1,0,0,NWAV()
y = RAYY(n)
PRINT "The ray intercept is ", y
```

相关关键字:

RAYTRACEX

RAYTRACEX

调用 OpticStudio 光线追迹程序，通过当前系统追迹来自任何起始面的特殊光线。

语法:

```
RAYTRACEX x, y, z, l, m, n, surf, wavelength
```

详述:

表达式 **x**、**y**、**z**、**l**、**m** 和 **n** 定义起始面在局部坐标中的输入光线位置和方向余弦。**surf** 表达式必须为 0 到表面数减一之间的整数（包含首尾值）。**wavelength** 表达式是可选的，默认为主波长。如果已提供 **wavelength** 表达式，则计算结果必须为介于 1 和所定义波长数量之间的一个整数。

如果物面的厚度无限大，并且 **surf** 变量值设为 0，则假定输入坐标与第一个表面相关，而不是与物面相关；虽然光线仍在物方空间介质中定义。否则，OpticStudio 将使用指定坐标，而不进行更改。

一旦进行追迹光线，则可以使用 ZPL 函数 **RAYX**、**RAYY**、**RAYZ**、**RAYL**、**RAYM** 和 **RAYN** 确定光线交点坐标和方向余弦（在全局坐标中，使用 **RAGX**、**RAGY**、**RAGZ**、**RAGL**、**RAGM** 和 **RAGN** 获取数据）。需注意，只有在指定 **surf** 编号之后的表面数据才有效。

如果在光线追迹期间产生错误，函数 **RAYE**（表示光线错误）将返回一个非 0 值。如果 **RAYE** 返回负值，则表明在表面上发生了全内反射，此表面编号由返回值的绝对值决定。如果 **RAYE** 返回大于 0 的值，则表明返回光线错过的表面编号。

检查光线错误的函数 **RAYE** 是可选的。如果 **RAYE** 不为 0，**RAYX**、**RAYY** 等函数可能会返回无效数据。函数 **RANX**、**RANY** 和 **RANZ** 返回光线相交表面法线的方向余弦，**OPDC** 返回光线的光程差。函数 **RAYV** 返回光线产生渐晕的表面编号，如果光线无渐晕，则 **RAYV** 返回 0。光线通过渐晕表面返回的值可能不准确。

示例:

```
n = NSUR()
RAYTRACEX 0,1,0,0,0,1,0,NWAV()
y = RAYY(n)
PRINT "The ray intercept is ", y
```

相关关键字:

RAYTRACE

READ

从 OPEN 命令打开的现有文本文件中读取数据。

语法:

```
READ x
```

```
READ x, y
READ x,y,z,a,b,c,q
```

详述:

文件必须已经打开，请参阅关键字 **OPEN** 了解更多详情。每个 **READ** 命令会从文件中读取一行数据。将该行的第一个有效数据存储于 **READ** 的第一个变量中。如果有第二个字段中的数据，则将其存储于 **READ** 的第二个变量中。因此，在 **READ** 命令中列出的变量数量应与文本文件中的列数相同。文件中的数据应使用空格分隔符。数据可以是任意格式，并且在内部将其提升为双精度。若要使用 Windows 设置中当前选定的小数分隔符，请使用关键字 **READ_LOCALE**。在一行中最多可以读取 2000 个字符。变量的最大数量是 199；若要读取包含更多变量的行，请使用关键字 **READNEXT**。列出的变量必须是有效的 ZPL 变量名。

READNEXT 不支持将数组变量名用作参数。此时，可以先将数据读入标量变量，然后在后续行中将此标量值赋值给数组变量，如下所示：

```
READ x
data(i, j) = x
```

通常在读取所有数据之后，用关键字 **CLOSE** 来关闭文件。请参阅函数 **EOFF**。

示例:

```
PRINT "Reading the double-column file TEST.DAT!"
OPEN "C:\DATA\TEST.DAT"
READ x1, y1
READ x2, y2
READ x3, y3
CLOSE
```

相关函数:

EOFF

相关关键字:

OPEN、**CLOSE**、**READNEXT**、**READSKIP**、**READSTRING**、**READ_LOCALE**

READ_LOCALE

从 **OPEN** 命令打开的现有文本文件中读取数据。

语法:

```
READ_LOCALE x
READ_LOCALE x, y
READ_LOCALE x,y,z,a,b,c,q
```

详述:

文件必须已经打开，请参阅关键字 **OPEN** 了解更多详情。每个 **READ** 命令会从文件中读取一行数据。将该行的第一个有效数据存储于 **READ** 的第一个变量中。如果有第二个字段中的数据，则将其存储

于 READ 的第二个变量中。因此，在 READ 命令中列出的变量数量应与文本文件中的列数相同。文件中的数据应使用空格分隔符。数据可以是任意格式，并且在内部将其提升为双精度。采用 Windows 设置中当前选择的字符作为小数分隔符。若要使用句点作为分隔符，请使用关键字 READ。在一行中最多可以读取 2000 个字符。变量的最大数量是 199；若要读取包含更多变量的行，请使用关键字 READNEXT_LOCALE 或 READNEXT。列出的变量必须是有效的 ZPL 变量名。

通常在读取所有数据之后，用关键字 CLOSE 来关闭文件。请参阅函数 EOFF。

示例：

```
PRINT "Reading the double-column file TEST.DAT!"
OPEN "C:\DATA\TEST.DAT"
READ_LOCALE x1, y1
READ_LOCALE x2, y2
READ_LOCALE x3, y3
CLOSE
```

相关函数：

EOFF

相关关键字：

OPEN、CLOSE、READ、READNEXT、READNEXT_LOCALE、READSKIP、READSTRING

READNEXT_LOCALE

从 OPEN 命令打开的现有文本文件中读取数据。

语法：

```
READNEXT_LOCALE x
READNEXT_LOCALE x, y
READNEXT_LOCALE x, y, z, a, b, c, q
```

详述：

READNEXT_LOCALE 与 READ_LOCALE 几乎相同。主要区别是，READNEXT_LOCALE 将从打开的文件中读取整行数据，直到换行符，而 READ_LOCALE 只能读取足够数量的字符。-

例如，如果数据文件的某一行中含如下数据：

3.0 4.0 5.0

以下两条 READNEXT_LOCALE 命令将读取 x、y 和 z 的值 3.0、4.0 和 5.0：

```
READNEXT_LOCALE x, y
READNEXT_LOCALE z
```

如果行很长或者变量数量很多，则 READNEXT_LOCALE 比 READ_LOCALE 或 READ 更有用。READNEXT_LOCALE 不支持将数组变量名用作变量。此时，可以先将数据读入标量变量，然后在后续行中将此标量值赋值给数组变量，如下所示：

```
READNEXT_LOCALE x
data(i, j) = x
```

示例：

```
OPEN "C:\DATA\TEST.DAT"
READNEXT_LOCALE x1, x2
READNEXT_LOCALE x3
CLOSE
```

相关关键字：

OPEN、CLOSE、READ、READ_LOCALE、READNEXT、READSKIP、READSTRING

READNEXT

从 OPEN 命令打开的现有文本文件中读取数据。

语法：

```
READNEXT x
READNEXT x, y
READNEXT x, y, z, a, b, c, q
```

详述：

READNEXT 与 READ 几乎相同。重要区别是，READNEXT 将从打开的文件中读取整个数据行，直至换行符，而 READ 只能读取足够数量的字符。

例如，如果数据文件的某一行中含如下数据：

```
3.0 4.0 5.0
```

以下两条 READNEXT 命令将读取 x、y 和 z 的值 3.0、4.0 和 5.0：

```
READNEXT x, y
READNEXT z
```

如果行很长或者变量数量很多，则 READNEXT 比 READ 更有用。READNEXT 不支持将数组变量名用作参数。此时，可以先将数据读入标量变量，然后在后续行中将此标量值赋值给数组变量，如下所示：

```
READNEXT x
data(i, j) = x
```

READNEXT 采用句点作为小数分隔符。要使用 Windows 设置中当前选定的小数分隔符，请使用关

键字 READNEXT_LOCALE。

示例:

```
OPEN "C:\DATA\TEST.DAT"  
READNEXT x1, x2  
READNEXT x3  
CLOSE
```

相关关键字:

OPEN、CLOSE、READ、READSKIP、READSTRING、READNEXT_LOCALE

READSKIP

从 OPEN 命令打开的现有文本文件中，读取任何数量的无效字符。

语法:

```
READSKIP n
```

详述:

文件必须已经打开，请参阅关键字 OPEN 了解更多详情。如果 n 是正数，则 READSKIP 从文件中读取正好 n 个字符。如果 n 是负数，则 READSKIP 读取至当前行的结尾。读取的字符被删除。此函数可以有效跳过任何数量的无效字符，从而可以更快地访问大型文件。

通常在读取所有数据之后，用关键字 CLOSE 来关闭文件。请参阅函数 EOFF。

示例:

```
PRINT "Reading the contents of file TEST.DAT!"  
OPEN TEST.DAT  
READSKIP 59  
READSTRING A$  
PRINT A$  
CLOSE
```

相关关键字:

OPEN、CLOSE READ、READNEXT、READSTRING

READSTRING

从 OPEN 命令打开的现有文本文件中读取数据。

语法:

```
READSTRING A$
```

详述:

文件必须已经打开，请参阅关键字 OPEN 了解更多详情。每个 READ STRING 命令将从打开的文件中读取一行字符。读取的整行字符都将存储于指定的变量中。一行中最多可以读取和存储 359 个字符。指定的变量必须是有效的 ZPL 字符串变量名，但不需要提前定义字符串变量。通常在读取所有字符之后，用关键字 CLOSE 来关闭文件。请参阅函数 EOFF。

示例:

```
PRINT "Reading the contents of file TEST.DAT!"  
OPEN TEST.DAT  
READSTRING A$  
PRINT A$  
CLOSE
```

相关关键字:

OPEN、CLOSE READ、READNEXT、READSKIP

RELEASE

请参阅[数组变量] (Array variables)。

RELOADOBJECTS

将 NSC 物体重新加载到 NSC 编辑器中。

语法:

```
RELOADOBJECTS surface, object
```

详述:

surface 的表达式的计算结果必须为整数，表示非序列组件面对应的表面编号。在 NSC 模式下，surface 的值设为 1。object 的表达式的计算结果必须为整数，表示需要加载的物体编号。object 的值为 0 表示重新加载所有物体。此关键字可能需要大量的时间来执行，具体时间取决于定义的物体的数量和类型。

示例:

```
RELOADOBJECTS 1, 0
```

REM、!、#

REM、!和#用来定义备注或注释。

语法:

```
REM (text)
! (text)
(valid zpl line) # (text)
```

详述:

“!”用来表示备注。仅当 REM 命令和“!”符号出现在某一行的开头时，才可表示备注或者注释。
符号可应用在某一行中的任何位置，但仅当“#”不在字符串内部时，才表示注释。“#”符号后面的所有内容都表示注释信息。

示例:

```
REM any text can be placed after the REM command.
! any text can also be placed
! after the exclamation symbol.
x = 5 # this syntax allows comments to be placed on the same line as a command.
```

REMOVEVARIABLES(keywords)

将当前定义的所有变量设置为固定值。

语法:

```
REMOVEVARIABLES
```

RESUMEUPDATES

当禁止显示时，SUSPENDUPDATES 可阻止任何 UI 编辑器窗口被 ZPL 命令更新，而 RESUMEUPDATES 可结束禁止。有关更多信息，请参阅 SUSPENDUPDATES。

RENAMEFILE

RENAMEFILE 用于重命名文件。

语法:

```
RENAMEFILE oldfilename, newfilename
```

详述:

该关键字需要两个文件名，定义为引号中的文字字符串或字符串变量。文件名必须包含完整路径、

名称及扩展名。将文件 oldfilename 重命名为 newfilename。

示例:

```
RENAMEFILE AFILE$, BFILE$
```

相关关键字:

```
COPYFILE  
DELETEFILE
```

RETURN

请参阅 GOSUB。

REWIND

REWIND 将删除 PRINT 命令输出的最后一行，直到上一行的结尾。可以实现在一个文本输出文件的现有行上输出计数器或其它数据。

语法:

```
REWIND
```

示例:

```
PRINT "First line"  
REWIND  
PRINT "New First line"
```

相关关键字:

PRINT

SAVEARCHIVE

将当前镜头保存为 Zemax 存档文件 (*.ZAR)。

语法:

```
SAVEARCHIVE filename
```

详述:

filename 必须包含存档文件的名称，包括扩展名。如果 filename 未指定存档文件的完整路径，则会使用镜头的默认文件夹。请参阅[备份存档文件](Backup To Archive File)。

SAVEDETECTOR(keywords)

将有关探测器的最新数据保存到文件中保存到指定文件中，包括 NSC 矩形探测器、颜色探测器、极坐标探测器或体探测器物体。

语法:

```
SAVEDETECTOR surf, object, filename
```

详述:

该关键字需要指定 surf 和 object 的数值表达式以及 filename。在混合序列/非序列模式下，surf 是非序列组件的表面编号；在非序列模式下，surf 的值设为 1。object 是探测器物体的编号。filename 可包含完整路径，如果没有提供路径，则会使用当前镜头的路径。对于矩形探测器、颜色探测器、极坐标探测器和体探测器物体，扩展名应分别为 DDR、DDC、DDP 或 DDV。

相关关键字:

LOADDETECTOR

SAVELENS

保存当前镜头文件。

语法:

```
SAVELENS filename, session
```

详述:

SAVELENS 用指定的 filename 保存当前镜头文件。原文件夹中的当前镜头名也会更改。若 filename 包含完整路径、名称和扩展名，则会将当前镜头文件保存到指定目录下；若 filename 只含名称和扩展名，则将当前镜头保存在默认文件夹中。如果指定的文件名不存在，则镜头数据将保存在当前文件中。如果变量 session 为非 0 值，则同时保存 session 文件。

示例:

```
SAVELENS  
SAVELENS "NEWCOPY.ZMX"  
SAVELENS NEW$
```

相关关键字:

LOADLENS

SAVEMERIT(keywords)

将当前评价函数保存到文件中。

语法:

```
SAVEMERIT filename
```

详述:

SAVEMERIT 将当前评价函数保存到文件中。如果 filename 包含完整路径（如 C:\MYDIR\MYLENS.MF），则将使用指定路径。如果 filename 不包含完整路径，则将使用默认 <data>\MeritFunction 文件夹（参阅[文件夹](Folders)）。另请参阅 LOADMERIT。

SAVETOLERANCE(keywords)

将当前公差保存到文件中。

语法:

```
SAVETOLERANCE "filename"  
SAVETOLERANCE file$
```

详述:

如果文件名包含完整路径（如 C:\MYDIR\MYLENS.TOL），则将创建指定文件。如果 filename 不包含完整路径，则将使用默认 <data>\Tolerance 文件夹（参阅[文件夹](Folders)）。另请参阅 LOADTOLERANCE。

SAVEWINDOW

将任何文本窗口的文本保存到文件中。

语法:

```
SAVEWINDOW winnum, filename
```

详述:

winnum 值可为整数或运算结果为整数的表达式。winnum 对应于应保存到文件的文本窗口编号。OpticStudio 以窗口打开的顺序对窗口进行编号，从 1 开始。若用户关闭某些窗口，则 OpticStudio 会从列表中删除已关闭的窗口，但不会对余下的窗口重新编号。重新打开的任何窗口将会使用最小的可用窗口编号。

示例:

```
SAVEWINDOW 1, "C:\TEMP\TEXTFILE.TXT"  
SAVEWINDOW 3, A$
```

SCATTER

在进行光线追迹时，用于控制是否追迹序列表面散射。

语法:

```
SCATTER ON  
SCATTER OFF
```

详述:

宏开始位置的默认条件是 **SCATTER OFF**；并且将以默认方式追迹所有光线。如果执行 **SCATTER ON**，则将为所有后续 **RAYTRACE** 命令启用序列表面散射。

SDIA

该命令已被废弃。请参阅“**SETSURFACEPROPERTY、SURP**”。

SETAIM

设置光线瞄准功能的状态。

语法:

```
SETAIM state
```

详述:

此关键字需要一个计算结果为 **0** 或 **1** 的数值表达式。表达式 **state** 是一个代码，**0** 表示关闭光线瞄准，**1** 表示开启光线瞄准。

示例:

```
SETAIM 1
```

相关关键字:

SETAIMDATA

SETAIMDATA

设置光线瞄准功能的各种变量数据。

语法:

```
SETAIMDATA code, value
```

详述:

code 值的使用如下所示:

code	特性

1	如果 value 为 1，则表示勾选[使用光线瞄准缓存] (Use Ray Aiming Cache)，如果 value 为 0，则表示不勾选。
2	如果 value 为 1，则表示勾选[增强型光线瞄准] (Robust Ray Aiming)，如果 value 为 0，则表示不勾选
3	如果 value 为 1，则表示勾选[以视场缩放光瞳漂移] (Scale Pupil Shift Factors by Field)，如果 value 为 0，则表示不勾选。
4、5、6	分别设置 X、Y 和 Z 光瞳漂移的值。
7、8	分别设置 X 和 Y 光瞳压缩的值。

示例:

```
SETAIMDATA 5, 0.34
```

相关关键字:

SETAIM

SETAPODIZATION

该命令已被废弃。请参阅“SETSYSTEMPROPERTY、SYSP”。

SETCONFIG(keywords)

设置多重结构（变焦）系统的当前结构。

语法:

```
SETCONFIG config
```

详述:

config 表达式的计算结果必须为 1 到多重结构的最大数量之间的整数。

示例:

```
SETCONFIG 4
```

相关函数:

CONF, NCON

SETDETECTOR

在矩形探测器物体上，设置任何像素对应的相干或非相干探测器数据。

语法:

```
SETDETECTOR surf, object, pixel, datatype, value
```

详述:

此关键字需要各变量的数值表达式，指定整数 surf、object、pixel 和 datatype，以及新的探测器值。surf 是非序列组件的表面编号；在非序列模式下，surf 的值设为 1。object 是探测器的编号。物体必须是矩形探测器。pixel 是介于 1 到 object 最大像素数之间的数字。datatype 为 0 表示非相干强度，1 表示角度空间中的非相干强度，2 表示相干实部，3 表示相干虚部，4 表示相干振幅。有关这些值的用法和含义的更多信息，请参阅[矩形探测器物体] (Detector Rectangle object)。非相干强度的单位是光源单位（参阅[光源单位] (Source Units)）。相干振幅的单位是光源单位的平方根。

示例:

```
SETDETECTOR 1, 5, 12, 0, 1.0056
```

相关函数:

NSDD

SETMCOPERAND

将多重结构编辑器的任何行或结构设置为任何数值。

语法:

```
SETMCOPERAND row, config, value, datatype
```

详述:

此关键字需要计算结果为整数的数值表达式，用于指定多重结构编辑器的 row 和 config 编号。

如果 config 编号为 0，则 value 的含义如下所示：

datatype = 0，value 是文字字符串或字符串变量，用于指定操作数的名称。

datatype = 1、2 或 3，value 是 1、2 或 3，分别表示多重结构操作数定义的部分参数值。更多关于多重结构操作数的描述，请参阅“多重结构操作数概要”。

如果 config 编号对应于某个已定义的结构，则 value 的含义如下所示：

datatype = 0，value 表示操作数的值。

datatype = 1，value 表示操作数的跟随求解类型中的偏移量。

datatype = 2，value 表示操作数的跟随求解类型中的缩放因子。

datatype = 3，value 表示操作数的求解类型，0 表示固定，1 表示变量，2 表示跟随，3 表示热跟

随。

datatype = 4, value 值是跟随的结构编号。

datatype = 5, value 值是跟随的行编号。

示例:

```
SETMCOPERAND 3, 4, somevalue, 0  
SETMCOPERAND 1, 0, "THIC", 0
```

相关函数:

MCOP

SETNSCPARAMETER(keywords)

在 NSC 编辑器中设置任何物体的参数值。

语法:

```
SETNSCPARAMETER surface, object, parameter, value
```

详述:

此关键字需要 3 个计算结果为整数的数值表达式, 用于指定非序列组件表面编号、物体编号和参数编号。第四个变量 value 是指定参数的新数值。

示例:

```
SETNSCPARAMETER 4, 2, 15, newp15value
```

相关函数:

NPOS, NPAR

相关关键字:

INSERTOBJECT、SETNSCPOSITION、SETNSCPROPERTY

SETNSCPOSITION(keywords)

在 NSC 编辑器中设置任何物体的 x、y、z 或 x 倾斜、y 倾斜、z 倾斜位置。

语法:

```
SETNSCPOSITION surface, object, code, value
```

详述:

此关键字需要 3 个计算结果为整数的数值表达式, 用于指定非序列组件表面编号、物体编号和[代码](code)。code 是 1 到 6, 分别表示 x、y、z 或 x 倾斜、y 倾斜、z 倾斜位置。第四个变量 value 表示指定位置的新数值。

示例:

```
SETNSCPOSITION 4, 2, 2, newyvalue
```

相关函数:
NPOS, NPAR
相关关键字:
INSERTOBJECT, SETNSCPARAMETER, SETNSCPROPERTY

SETNSCPROPERTY(keywords)

设置 NSC 物体的属性。

语法:
SETNSCPROPERTY surface, object, code, face, value

详述:
此关键字需要 4 个计算结果为整数的数值表达式，用于指定非序列组件表面编号（在非序列模式下，surface 值设为 1）、物体编号、指定修改物体对应属性的代码以及物体的对应面编号（如果没有使用设置的属性，则 face 值设为 0）。第五个变量 value 表示指定属性的新数值，可以是引号中的文本、字符串变量或者数值表达式。代码如下：

代码	属性
以下代码表示设置 NSC 编辑器中的值。	
1	设置物体注释。
2	设置参考物体编号。
3	设置[在内部] (inside of) 的物体编号。
4	设置物体材料。
以下代码用于设置[物体属性] (Object Properties) 对话框中的[类型](type) 选项卡上对应的值。	
0	设置物体类型。value 必须是物体的名称，比如表示标准镜头的 “NSC_SLEN”。在分类报表中列出了 NSC 编辑器中每个物体类型的名称。所有 NSC 物体名称均以 “NSC_” 开头。
13	设置[用户自定义孔径](User Defined Aperture)，用 1 表示勾选，用 0 表示未勾选。
14	设置[用户自定义孔径](User Defined Aperture)文件名。
15	设置[使用全局 XYZ 旋转命令] (Use Global XYZ Rotation Order) 复选框，用 1 表示

	勾选，用 0 表示未勾选。
16	设置[光线忽略物体] (Rays Ignore This Object)选项，用 0 表示从不，1 表示始终，2 表示启用。
17	设置[物体为探测器] (Object Is A Detector)复选框，用 1 表示勾选，用 0 表示未勾选。
18	设置[考虑物体] (Consider Objects)列表。此变量是一个字符串，列出需要考虑物体的编号，用空格隔开，比如“2 5 14”。
19	设置[忽略物体] (Ignore Objects)列表。此变量是一个字符串，列出需要忽略物体的编号，用空格隔开，比如“1 3 7”。
20	设置[使用像素插值] (Use Pixel Interpolation)复选框，用 1 表示勾选，用 0 表示未勾选。
30	设置[光线分裂时使用考虑/忽略物体] (Use Consider/Ignore Objects When Splitting)复选框，用 1 表示勾选，用 0 表示未勾选。
以下代码用于设置[物体属性] (Object Properties)对话框中[膜层/散射] (Coat/Scatter)选项卡上对应的值。	
5	设置指定表面的膜层名称。
6	设置指定表面的配置文件名称。
7	设置指定表面的散射模型。0 = 无，1 = 朗伯，2 = 高斯，3 = ABg，4 = 用户自定义，5 = BSDF，6 = ABg 文件，7 = IS 散射库。
8	设置指定表面的散射系数。
9	设置指定表面的散射光线数。
10	设置指定面的高斯 Sigma（高斯散射模型）或采样方位角（BSDF 或 IS 散射库散射模型）。
11	设置指定表面的反射 ABg 数据名称。
12	设置指定表面的透射 ABg 数据名称。

27	设置用户自定义散射 DLL 的名称。
21-26	设置用户定义的散射 DLL 的参数值。
28	设置用户定义的散射数据文件的名称。
29	设置指定表面的[面为] (Face Is)属性。0 表示[物体默认] (Object Default)，1 表示[反射] (Reflective)，2 表示[吸收] (Absorbing)。
31	设置指定面的反射 BSDF 数据文件。value 必须是 BSDF 文件的名称，且不含路径（如 BrownVinyl.bsdf）。
32	设置指定面的透射 BSDF 数据文件。value 必须是 BSDF 文件的名称，且不含路径（如 BrownVinyl.bsdf）。
33	设置指定面的反射 ABg 数据文件。value 必须为 ABGF 文件名，且不含路径（如 SampleABGF.abgf）。
34	设置指定面的透射 ABg 数据文件。value 必须为 ABGF 文件名，且不含路径（如 SampleABGF.abgf）。
35	设置指定面的反射 IS 散射库数据文件。value 必须为 ISX 文件名，不含路径（如 BrownVinyl.ISX）。
36	设置指定面的透射 IS 散射库数据文件。value 必须为 ISX 文件名，不含路径（如 BrownVinyl.ISX）。
37	设置指定表面的薄板散射选项。使用 0 可关闭选项（不勾选复选框中的选项），用 1 可开启选项（勾选复选框中的选项）
38	设置 IS 散射库散射的采样面反射率。用 0 表示前面，1 表示后面。
39	设置 IS 散射库散射的采样面透过率。用 0 表示前面，1 表示后面。
40	设置 IS 散射库散射的采样反射率。用 0 表示 5 度，1 表示 2 度、2 表示 1 度。
41	设置 IS 散射库散射的采样透过率。用 0 表示 5 度，1 表示 2 度、2 表示 1 度。
以下代码用于设置[物体属性] (Object Properties)对话框中 [体散射] (Volume Physics)选项卡	

上对应的值。	
81	在体散射选项卡上设置[模型](Model)的值。用 0 表示[无体散射](No Bulk Scattering)，1 表示[角度散射](Angle Scattering)，2 表示[DLL 文件散射](DLL Defined Scattering)，3 表示[荧光散射]()。
82	设置要用于体散射的平均光程。
83	设置要用于体散射的角度。
84	设置要用于体散射的 DLL 文件名称。
85	设置要传递到 DLL 的参数值，其中 face 值用于指定需要定义的参数。第一个参数为 1，第二个参数为 2，以此类推。
86	设置波长漂移字符串。
以下代码用于设置[物体属性](Object Properties)对话框中[衍射](Diffraction)选项卡上对应的值。	
91	在[衍射](Diffraction)选项卡上设置[分裂](Split)值。0 表示[不按次序分裂](Don't Split By Order)，1 表示[按以下表格分裂](Split By Below)，2 表示[根据 DLL 函数分裂](Split By DLL Function)。
92	设置要用于衍射分裂的 DLL 的名称。
93	设置起始次序值。
94	设置终止次序值。
95、96	在[衍射](Diffraction)选项卡上设置参数值。这此参数传递到衍射分裂 DLL 函数，同样也是[按以下表格分裂](Split By Below)选项使用的次序值。 face 值用于指定需要定义的参数。第一个参数为 1，第二个参数为 2，以此类推。代码 95 用于反射属性，96 用于透射属性。
以下代码用于设置[物体属性](Object Properties)对话框中[光源](Sources)选项卡上对应的值。	
101	设置光源物体的[随机偏振](random polarization)。用 1 表示勾选，用 0 表示未勾选。
102	设置光源物体[逆向光线](reverse rays option)选项。用 1 表示勾选，用 0 表示未勾

	选。
103	设置光源物体琼斯 X 值。
104	设置光源物体琼斯 Y 值。
105	设置光源物体相位 X 值。
106	设置光源物体相位 Y 值。
107	设置光源物体[初始相位](initial phase)值(以度为单位)。
108	设置光源物体[相干长度](coherence length)值。
109	设置光源物体[预传播](pre-propagation)值。
110	设置光源物体采样方法；0 表示随机，1 表示 Sobol 采样。
111	设置光源物体体散射方法；0 表示多次、1 表示一次、2 表示从不。
112	设置[阵列类型](array mode)，0 表示无，1 表示矩形，2 表示圆形，3 表示六边，4 表示六角。
113	设置[光源颜色模型](source color mode)。有关可用模式的完整列表，请参阅[定义光源颜色和光谱的目录](Defining the color and spectral content of sources)。光源颜色模型从 0 开始编号，0 表示系统波长，然后从 1 到 n 分别表示对话框中列出的模型序号。
114-116	分别设置光谱步长数量、起始波长和终止波长。
117	设置光谱文件的名称。
161-162	设置阵列类型整数变量 1 和 2。
165-166	设置阵列类型双精度变量 1 和 2。
181-183	设置光源颜色模型变量，例如，三色刺激值 XYZ。
以下代码用于设置[物体属性](Object Properties)对话框中[梯度折射](Grin)选项卡上对应的值。	
121	设置使用 [DLL 定义梯度折射率介质](Use DLL Defined Grin Media)复选框。用 1 表

	示勾选，用 0 表示未勾选。
122	设置[最大步长] (Maximum Step Size) 值。
123	设置 DLL 名称。
124	设置 Grin DLL 参数。这些是传递到 DLL 的参数。 face 值用于指定需要定义的参数。第一个参数为 1，第二个参数为 2，以此类推。
以下代码用于设置[物体属性] (Object Properties)对话框中[绘图] (Draw)选项卡上对应的值。	
141	设置[不显示此物体] (do not draw)复选框。用 1 表示勾选，0 表示未勾选。
142	设置物体透明度。用 0 表示 100%，1 表示 90%，2 表示 80%，以此类推。
143	设置物体的[绘图精度](drawing resolution)。用 0 表示标准，1 表示中等，2 表示高，3 表示精密，4 表示用户自定义。
144	在[绘图精度](drawing resolution)设置为 Custom 时，设置绘图精度中输入的第一个分辨率的值（例如，环形非球面镜头物体的“角”分辨率）。在使用此代码之前，必须将[绘图精度](drawing resolution)设置为 Custom。
145	在[绘图精度](drawing resolution)设置为 Custom 时，设置绘图精度中输入的第二个分辨率的值（例如，环形非球面镜头物体的“径向”分辨率）。在使用此代码之前，必须将[绘图精度](drawing resolution)设置为 Custom。
以下代码用于设置[物体属性] (Object Properties)对话框中[散射路径] (Scatter To)选项卡上对应的值。	
151	设置散射路径模型。用 0 表示散射路径(Scatter To List)，1 表示重点采样。
152	设置重点采样目标数据。变量必须是一个字符串，列出光线编号、物体编号、尺寸和极值，并且用空格分隔，以下是用于设置光线 3、物体 6、尺寸 3.5 和极值 0.6 重点采样的语法：“3 6 3.5 0.6”。
153	设置[散射路径](Scatter To List)值。变量是一个字符串，列出了散射路径上的物体编号，用空格分隔，比如“4 6 19”。
以下代码用于设置[物体属性] (Object Properties)对话框中[双折射] (Birefringence)选项卡对应的值。	

171	设置[双折射介质] (Birefringence Media)复选框。用 0 表示未勾选，用 1 表示勾选。
172	设置双折射介质模式。用 0 表示追迹寻常光线和非寻常光线，1 表示只追迹寻常光线，2 表示只追迹非寻常光线，3 表示波片模式。
173	设置[双折射介质反射] (Birefringent Media Reflections)状态。用 0 表示追迹反射和折射光线，1 表示只追迹反射光线，2 表示只追迹折射光线。
174-176	设置 Ax、Ay 和 Az 值。
177	设置轴长。
以下代码不设置任何值，此处只是为了返回函数 NPRO 的值。	
200	通过函数 NPRO 确定某个物体的折射率。语法为： <code>NPRO(表面, 物体, 200, 波长编号)</code>
201-203	通过函数 NPRO 确定某个物体的 nd (201)、vd (202) 和 dpgf (203) 参数（使用模型玻璃求解方式）。语法为： <code>NPRO(表面, 物体, 201, 0)</code>

以后会增加其它代码值，根据用户的需求来设置 NSC 物体的其它属性。

示例：

```
SETNSCPROPERTY 1, 2, 0, 0, "NSC_SLEN"
```

相关函数：

NPOS、NPAR、NPRO

相关关键字：

INSERTOBJECT、SETNSCPARAMETER、SETNSCPOSITION

SETOPERAND(keywords)

将评价函数编辑器中的任何行和列设置为某个数值。

语法：

```
SETOPERAND row, col, value
```

详述：

此关键字需要计算结果为整数的数值表达式，用于指定评价函数编辑器的行和列。整数 col 为 1 表示操作数类型，2 表示 int1，3 表示 int2，4 到 7 表示 data1 到 data4，8 表示目标，9 表示权重。若要设置与操作数相关联的注释字符串，请将 col 设为 10。若要设置操作数类型，可将 col 设为

1，并使用函数 **ONUM** 返回的整数操作数。设置操作数类型的另一种方法是将 **col** 设为 **11**，并与操作数类型的字符串名称配合使用，如 **EFFL** 或 **DIST**。如果 **col** 为 **10** 或 **11**，则 **value** 应该是字符串常量或变量。**col** 为 **12** 表示 **data5**，**13** 表示 **data6**。请注意，无法设置当前值和贡献百分比，但会计算其数值。

示例:

```
SETOPERAND 1, 8, tarvalue
SETOPERAND 3, 11, "EFFL"
SETOPERAND 5, 10, "Operand Number 5"
```

相关函数:

MFCN, OPER, ONUM

SETSTDD

该命令已被废弃。请参阅 SETSURFACEPROPERTY、SURP。

SETSURFACEPROPERTY, SURP

设置表面的属性。

语法:

```
SETSURFACEPROPERTY surface, code, value1, value2
SURP surface, code, value1, value2
```

详述:

surface 是一个计算结果为整数的表达式，指定表面编号。**code** 是一个计算结果为整数的表达式，或者是助记符，用于指定要修改表面的哪些属性。第三个和第四个变量表示指定表面属性的新值，可能是引号中的文本、字符串变量或数值表达式，具体取决于代码。对于大多数代码，需要修改的属性由 **value1** 参数定义。少数操作数同时需要 **value1** 和 **value2** 定义，如下表所述。

如果要修改的属性由多重结构编辑器控制，那么当前结构的多重结构数据也会发生变化，以显示更改的属性。

若要将某个表面设置为光阑面，请参阅 **STOPSURF**。

SURP 是 **SETSURFACEPROPERTY** 的缩写，功能完全相同。

代码	特性
表面基础数据。请参阅 [镜头数据](Lens Data)。	
0 或 TYPE	表面类型。value 是物体的名称，比如表示标准表面的 STANDARD 。在表面数

	<p>据概要的分类报表中列出了当前镜头数据编辑器中的每个表面类型名称。</p> <p>若要将表面类型更改为用户自定义表面，请首先使用代码 9 (SDLL) 来设置 DLL 名称，然后将新的表面类型设置为 USERSURF。</p> <p>另请参阅[代码 17] (Code 17)。</p>
1 或 COMM	注释。
2 或 CURV	以镜头单位表示的曲率（不是半径）。使用 0 表示无穷大半径。
3 或 THIC	厚度，以镜头单位表示。
4 或 GLAS	玻璃名称。另请参阅[代码 18] (Code 18)。
5 或 CONI	圆锥系数。
6 或 SDIA	净半口径或半口径。如果 value 为 0 或正数，则净半口径或半口径的求解类型设置为固定。如果 value 为负数，则净半口径或半口径的求解类型设置为自动，并且净半口径或半口径的值由下一个 UPDATE 关键字计算出来。
7 或 TCE	热膨胀系数。
8 或 COAT	膜层名称。将 value1 设置为空白字符串可删除膜层。
9 或 SDLL	用户自定义表面 DLL 名称。
10 或 PARM	参数值。value1 为新的数值。value2 为参数编号。
11 或 EDVA	附加数据值。value1 为新的数值。value2 为附加数据编号。
12	表面颜色，用 0 表示默认值。
13	表面透明度。
14	行颜色。
15	表面不能是超半球面。使用 1 可避免表面成为超半球面。
16	忽略表面。用 1 表示忽略表面， 0 表示不忽略表面。
17 或 CODE	表面类型的整数代码。整数代码是使用代码 0 设置表面名称的备用选项。有关更多详情，请参阅上文中对代码 0 的说明。

18 或 GLAN	玻璃编号。另请参阅[代码 4] (Code 4)。
表面孔径数据。请参阅[表面属性对话框中孔径选项卡](Surface properties aperture tab)。	
20 或 ATYP	表面孔径类型代码。
21 或 APP1	表面孔径参数 1。
22 或 APP2	表面孔径参数 2。
23 或 APDX	表面孔径 x 偏心。
24 或 APDY	表面孔径 y 偏心。
25 或 UDA	用户自定义孔径 (UDA) 文件名。
26 或 APPU	表面孔径跟随求解的表面编号。用 0 表示无。
27 或 CHZN 表面的延伸区。28 或 MCSD 机械半口径。如果 value 为 0 或正数，则机械半口径的求解类型设置为固定。如果 value 为负数，则机械半口径的求解类型设置为自动，并且机械半口径的值由下一个 UPDATE 关键字计算出来。物理光学传播设置。请参阅[表面属性对话框中物理光学选项卡](Surface specific settings)。	
30	物理光学设置[用几何光线传播到下一表面] (Use Rays To Propagate To Next Surface)。用 1 表示勾选，0 表示未勾选。
31	物理光学设置[不用几何光线数据缩放光束尺寸] (Do Not Rescale Beam Size Using Ray Data)。用 1 表示勾选，0 表示未勾选。
32	物理光学设置[采用角谱算法] (Use Angular Spectrum Propagator)。用 1 表示勾选，0 表示未勾选。
33	物理光学设置[在实体模型上显示 ZBF] (Draw ZBF On Shaded Model)。用 1 表示勾选，0 表示未勾选。
34	物理光学设置 Recompute Pilot Beam Parameters。1 表示勾选，0 表示未勾选。
35	物理光学设置[折射后重新采样] (Resample After Refraction)。1 表示勾选，0 表示未勾选。

36	物理光学设置[自动重新采样](Auto Resample)。1 表示勾选，0 表示未勾选。
37	物理光学设置[新建 X 采样](New X Resample)。1 表示 32，用 2 表示 64 等。
38	物理光学设置[新建 Y 采样](New Y Resample)。1 表示 32，2 表示 64 等。
39	物理光学设置[新建 X 宽度](New X-Width)。数组的 X 方向全宽。
40	物理光学设置[新建 Y 宽度](New Y-Width)。数组的 Y 方向全宽。
41	物理光学设置[输出 Pilot 半径](Output Pilot Radius)。使用 0 表示最佳拟合，1 表示更短，2 表示更长，3 表示 x，4 表示 y，5 表示平面，6 表示用户。
42、43	分别为物理光学设置 X-Radius 和 Y-Radius。
44	物理光学设置[采用 X 轴参考](Use X-axis Reference)。1 表示勾选，0 表示未勾选。
膜层设置。请参阅[表面属性对话框中膜层选项卡](Surface coating tab)。另请参阅上文的 code 8。	
50	使用多膜层和折射率偏移。用 1 表示勾选，0 表示未勾选。
51	多膜层值。value1 为新的数值。value2 为膜层编号。
52	多膜层状态。value1 为状态，用 0 表示固定，1 表示变量，或者用 n+1 表示跟随膜层 n。value2 为膜层编号。
53	膜层折射率偏移值。value1 为新的数值。value2 为膜层编号。
54	膜层折射率偏移状态。value1 为状态，用 0 表示固定，用 1 表示变量，或者用 n+1 表示跟随膜层 n。value2 为膜层编号。
55	膜层消光偏移值。value1 为新的数值。value2 为膜层编号。
56	膜层消光偏移状态。value1 为状态，用 0 表示固定，用 1 表示变量，或者用 n+1 表示跟随膜层 n。value2 为膜层编号。
表面倾斜和偏心数据。请参阅[表面属性对话框中倾斜/偏心选项卡](Surface tilt/decenter tab)。	

60 或 BOR	面之前的倾斜和偏心顺序。用 0 表示偏心/倾斜，用 1 表示倾斜/偏心。
61 或 BDX	面之前 x 偏心。
62 或 BDY	面之前 y 偏心。
63 或 BTX	面之前 x 倾斜
64 或 BTY	面之前 y 倾斜
65 或 BTZ	面之前 z 倾斜
66 或 APU	面之后的状态：0 表示直接，1 或 2 表示跟随或翻转当前表面，3 或 4 表示跟随或翻转当前表面减 1，5 或 6 表示跟随或翻转当前表面减 2，以此类推。
70 或 AOR	面之后倾斜和偏心顺序。用 0 表示偏心/倾斜，用 1 表示倾斜/偏心。
71 或 ADX	面之后 x 偏心。
72 或 ADY	面之后 y 偏心。
73 或 ATX	面之后 x 倾斜。
74 或 ATY	面之后 y 倾斜。
75 或 ATZ	面之后 z 倾斜。
76	坐标返回状态。仅在坐标变换面上有效。用 0 表示无，1 表示仅方向，2 表示 XY 方向，3 表示 XYZ 方向。
77	坐标返回至表面。仅在坐标变换面上有效。
表面散射数据。请参阅[表面属性对话框中散射选项卡](Surface properties scattering tab)。	
80	设置散射代码：0 表示无，1 表示朗伯，2 表示高斯，3 表示 ABg，4 表示 DLL，5 表示 BSDF，6 表示 ABg 文件，7 表示 IS 散射库。
81	设置散射因子，在 0.0 到 1.0 之间。
82	设置高斯散射 Sigma。

83	设置 ABg 文件名称。
84	设置用户自定义散射 DLL 的名称。若要设置参数，请参阅 Code181。
85	设置用户自定义散射 DLL 使用的数据文件名称。
86	设置 BSDF 文件名称。value 必须是 BSDF 文件的名称，且不含路径（如 BrownVinyl.bsdf）。
87	设置 ABg 数据文件名称。value 必须是 ABGF 文件的名称，且不含路径（如 SampleABGF.abgf）。
88	设置 IS 散射库数据文件名称。value 必须为 ISX 文件名，且不含路径（如 BrownVinyl.ISX）。
110	设置 IS 散射库散射的边。用 0 表示前面，1 表示后面。
111	设置 IS 散射库散射的采样。用 0 表示 5 度，1 表示 2 度、2 表示 1 度。
表面绘制数据。请参阅[表面属性对话框中绘制选项卡](Surface properties draw tab)。	
90	设置[隐藏这个表面的光线] (Hide Rays To This Surface)复选框状态：0 表示关闭，1 表示开启。
91	设置[光线忽略这个面] (Skip Rays To This Surface)复选框状态：0 表示关闭，1 表示开启。
92	设置[不显示此表面] (Do Not Draw This Surface)复选框状态：0 表示关闭，1 表示开启。
93	设置[不显示此表面的边缘] (Do Not Draw Edges From This Surface)复选框状态：0 表示关闭，1 表示开启。
96	设置[边缘显示] (Draw Edges As)状态：0 表示平方，1 表示锥形，2 表示平面。
97	设置[镜面基底] (Mirror Substrate)状态：0 表示无，1 表示平面，2 表示弯曲。
98	设置镜面基底厚度值。

用户自定义面散射 DLL 参数。请参阅[表面属性对话框中散射选项卡](Surface properties scattering tab)。	
181-186	设置用户自定义的散射 DLL 参数 1-6。

更改表面属性之后，通常需要执行 UPDATE 关键字（参阅 UPDATE），以更新光瞳位置、求解以及正确光线追迹所必需的其它数据。OpticStudio 不会自动执行 UPDATE，因为执行所有关键字 SETSURFACEPROPERTY 之后再执行一次 UPDATE，这样运行速度会更快。函数 SPRO 使用类似的语法和相同代码值来“获取”表面属性值，而不是“设置”这些值。

示例：

```
!Set the glass type on surface 7 to BK7
SETSURFACEPROPERTY 7, GLAS, "BK7"
!Set the thickness of surface 2 to the thickness of surface 1
SETSURFACEPROPERTY 2, THIC, THIC(1)
!Set the value of parameter 4 on surface 11 to 7.3
SURP 11, PARM, 7.3, 4
```

相关函数：

SPRO

相关关键字：

SETSYSTEMPROPERTY、UPDATE

SETSYSTEMPROPERTY、SYSP

设置系统的属性，如系统孔径、视场、波长和其它数据。

语法：

```
SETSYSTEMPROPERTY code, value1, value2
SYSP code, value1, value2
```

详述：

此关键字需要一个计算结果为整数的表达式，指定要修改系统的哪些属性。第二个和第三个变量是指定属性的新数值，可能是引号中的文本、字符串变量或数值表达式，具体取决于 code。对于大多数 code，需要修改的属性由 value1 参数定义。少数操作数同时需要 value1 和 value2 定义，如下表所述。

如果要修改的属性由多重结构编辑器控制，那么当前结构的多重结构数据也会发生变化，以显示更改的属性。。

SYSP 是 SETSYSTEMPROPERTY 的缩写，功能完全相同。

代码	特性
4	设置[折射率数据与环境匹配] (Adjust Index Data To Environment)，用 0 表示关闭，1 表示开启。请参阅[折射率数据与环境匹配] (Adjust Index Data To Environment)。
10	孔径类型代码。有关代码值的详情，请参阅[孔径类型] (Aperture Type)。
11	孔径值。请参阅[孔径值] (Aperture Value)。
12	切趾类型代码。用 0 表示均匀，1 表示高斯，2 表示余弦立方。请参阅[切趾类型] (Apodization Type)。
13	切趾因子。请参阅[切趾因子] (Apodization Factor)。
14	远心物空间。用 0 表示关闭，1 表示开启。请参阅[远心物空间] (Telecentric Object)。
15	在更新时迭代求解。用 0 表示关闭，1 表示开启。请参阅[在更新时迭代求解] (Iterate Solves When Updating)。
16	镜头标题。请参阅[镜头标题] (Lens Title)。
17	镜头备注。请参阅[备注] (Notes)。
18	无焦像空间。用 0 表示关闭，1 表示开启。请参阅[无焦像空间] (Afocal Image Space)。
21	全局坐标参考面。请参阅[全局坐标参考面] (Global Coordinate Reference Surface)。
23	玻璃库列表。用一个字符串或字符串变量表示，指定玻璃库，如 SCHOTT。若要指定多个库，可列出以空格分隔的多个玻璃库名称，比如 SCHOTT HOYA OHARA。
24	系统温度（以摄氏度为单位）。请参阅[以摄氏度表示的温度] (Temperature in degrees C)。

25	以大气压表示的系统压强。请参阅[ATM 压强] (Pressure in ATM)。
26	OPD 参考方法。用 0 表示绝对，用 1 表示无限，用 2 表示出瞳，用 3 表示绝对 2。请参阅[OPD 参考] (Reference OPD)。
30	镜头单位代码。用 0 表示毫米，1 表示厘米，2 表示英寸，3 表示米。更改镜头单位不会缩放或转换镜头数据，只是更改显示镜头规格数据的方式。请参阅[镜头单位] (Lens Units)。
31	光源单位前缀。用 0 表示法，1 表示皮，2 表示纳，3 表示微，4 表示毫，5 表示无、6 表示千，7 表示百万，8 表示千兆，9 表示百亿。请参阅[光源单位] (Source Units)。
32	光源单位。用 0 表示瓦，1 表示流明，2 表示焦耳。请参阅[光源单位] (Source Units)。
33	分析单位前缀。用 0 表示法，1 表示皮，2 表示纳，3 表示微，4 表示毫，5 表示无，6 表示千，7 表示百万，8 表示千兆，9 表示百亿。请参阅[分析单位] (Analysis Units)。
34	分析单位的[“每” 面积] (“per” Area)。用 0 表示每平方毫米，1 表示每平方厘米，2 表示每平方英寸，3 表示每平方米，4 表示每平方英尺。
35	MTF 单位代码。0 表示周期/毫米，1 表示周期/毫弧度。请参阅 [MTF 单位] (MTF Units)。
40	膜层文件名称。请参阅[膜层文件] (Coating File)。
41	散射配置文件名称。请参阅[散射配置文件] (Scatter Profile)。
42	ABg 数据文件名称。请参阅[ABg 数据文件] (ABg Data File)。
43	GRADIUM Profile 名称。请参阅 GRADIUM Profile。
50	NSC 每条光线最大交点数目。请参阅[每条光线最大交点数目] (Maximum Intersections Per Ray)。
51	NSC 每条光线最大段数。请参阅[每条光线最大段数] (Maximum Segments Per

	Ray)。
52	NSC 最大嵌套/接触物体数目。 请参阅[最大嵌套/接触物体数目] (Maximum Nested/Touching Objects)。
53	NSC 光线追迹相对阈值强度。 请参阅[最小相对阈值强度] (Minimum Relative Ray Intensity)。
54	NSC 光线追迹绝对阈值强度： 请参阅[最小绝对阈值强度] (Minimum Absolute Ray Intensity)。
55	NSC 系统单位下的胶合距离。 请参阅[系统单位下的胶合距离] (Glue Distance In Lens Units)。
56	NSC 系统单位下的截止光线长度。 请参阅[系统单位下的截止光线长度] (Missed Ray Draw Distance)。
57	NSC 文件打开时重新追迹光线。 用 0 表示否，1 表示是。请参阅[文件打开时重新追迹光线] (Retrace Source Rays Upon File Open)。
58	NSC 光线文件在内存中的最大光线数目， 请参阅[光线文件在内存中的最大光线数目] (Maximum Source File Rays In Memory)。
59	简单光线分裂。 用 0 表示否，1 表示是。请参阅[简单光线分裂] (Simple Ray Splitting)。
60	偏振 Jx。 请参阅 Jx, Jy, X-Phase, Y-Phase。
61	偏振 Jy。 请参阅 Jx, Jy, X-Phase, Y-Phase。
62	偏振 X 相位。 请参阅 Jx, Jy, X-Phase, Y-Phase。
63	偏振 Y 相位。 请参阅 Jx, Jy, X-Phase, Y-Phase。
64	将膜层相位转换为等效几何光线。 用 0 表示否，1 表示是。请参阅[将膜层相位转换为等效几何光线] (Convert thin film phase to ray equivalent)。
65	无偏振。 用 0 表示否，1 表示是。请参阅[无偏振] (Unpolarized)。

66	偏振参考方式。用 0 表示 X 参考轴，1 表示 Y 参考轴，2 表示 Z 参考轴。请参阅[偏振参考方式] (Method)。
70	光线瞄准。用 0 表示关闭，1 表示近轴，2 表示实际。请参阅[光线瞄准] (Ray Aiming)。
71、72、73	光线瞄准光瞳漂移 x、y 和 z。请参阅[光瞳漂移，光瞳压缩] (Pupil Shift, Pupil Compress)。
74	使用光线瞄准缓存。用 0 表示否，1 表示是。请参阅[使用光线瞄准缓存] (Use Ray Aiming Cache)。
75	增强型光线瞄准。用 0 表示否，1 表示是。请参阅[增强型光线瞄准(慢)] (Robust Ray Aiming (slow))。
76	以视场缩放光瞳漂移，用 0 表示否，1 表示是。请参阅[光瞳漂移，光瞳压缩] (Pupil Shift, Pupil Compress)。
77、78	光线瞄准光瞳压缩 x, y。请参阅[光瞳漂移，光瞳压缩] (Pupil Shift, Pupil Compress)。
100	视场类型代码。请参阅[视场](Fields)。
101	视场数目。
102、103	视场编号为 value1，value2 是视场 x,y 坐标
104	视场编号为 value1，value2 是视场权重
105、106	视场编号为 value1，value2 是视场渐晕 x 偏心、y 偏心
107、108	视场编号为 value1，value2 是视场渐晕压缩 x，压缩 y
109	视场编号为 value1，value2 是视场渐晕角
110	视场归一化方式，value1 为 0 表示径向，为 1 表示矩形

200	主波长编号。请参阅 wavelength。
201	波长数目
202	波长编号为 value1, value2 是以微米为单位的波长。
203	波长编号为 value1, value2 是波长权重
901	在多线程计算（比如优化）中使用的 CPU 数量。如果传递的值为 0，则 CPU 数量将重置为默认值。使用函数 SYPR 测试此值时，将返回操作系统中可用 CPU 的总数。

通常，只有在执行了关键字 UPDATE 之后，系统属性的更改才会生效。

示例：

```
!Set the number of wavelengths to 3
SETSYSTEMPROPERTY 201, 3
!Set the number of fields to 4
SYSP 101, 4
```

相关函数：

SYPR

相关关键字：

SETSURFACEPROPERTY, UPDATE

SETTEXTSIZE

更改 GTEXT 命令绘制的字符大小。

语法：

```
SETTEXTSIZE xsize, ysize
```

详述：

变量值表示每个字符占图形窗口的比例。例如，默认文本大小为 70 40。这意味着，每个字符是图形窗口宽度的 1/70，是图形窗口高度的 1/40。变量 0 可将文本大小恢复为默认值。

示例：

```
!Make text twice default size
SETTEXTSIZE 35, 20
!Restore text size to default
SETTEXTSIZE
```

SETTITLE

该命令已被废弃。请参阅“SETSYSTEMPROPERTY、SYSP”。

SETTOL(keywords)

设置公差数据编辑器中任何公差操作数对应不同参数列的数据。

语法:

```
SETTOL row, column, data
```

详述:

row 的值必须是大于 0 且小于或等于当前公差操作数数量的整数表达式。若要设置操作数类型，请将 column 设为 0。data 是包含操作数名称的文字字符串或字符串变量，如“TTHI”。若要设置某操作数类型使用的整数值 1、2 或 3，请使用 column 编号 1、2 或 3。若要设置最小或最大公差值，请分别使用 column 编号 4 或 5。若要设置[在反向公差分析时不调整](Ignore this Operand During Tolerancing)复选框，请使用 column 编号 6，data1 表示勾选，0 表示未勾选。若要设置注释，请使用 column 编号 99，data 是包含注释的文字字符串或字符串变量。公差操作数的当前值不能通过 SETTOL 来设置。

另请参阅 INSERTTOL 和 DELETETOL。

示例:

```
SETTOL 16, 0, “TRAD”
```

相关函数:

TOLV、\$TOLOPERAND、\$TOLCOMMENT

SETUNITS

该命令已被废弃。请参阅“SETSYSTEMPROPERTY、SYSP”。

SETVAR

修改优化变量的状态。

语法:

```
SETVAR surf, code, status, object
```

或

```
SETVAR config, M, status, operand
```

详述:

surf 必须是计算结果为 0 到最大表面数量之间的整数。config 必须是计算结果为 1 到多重结构数量之间的整数。code 必须是下列字符串之一:

R 表示曲率半径
T 表示厚度
C 表示圆锥系数
G 表示玻璃
I 表示玻璃折射率
J 表示玻璃阿贝数
K 表示玻璃 dpgf
Pn 表示参数 n
D 表示热膨胀系数
En 表示附加数据值 n
M 表示多重结构数据, 请参阅以下详述
Nn 表示非序列元件位置数据, 1 到 6 分别表示 x、y、z、tx、ty、tz
On 表示非序列元件参数数据, 其中 n 是参数编号

如果 status 计算结果为 0, 则表示删除变量状态。否则表示将对应值设为变量。如果 code 是 Nn 或 On, 则必须提供物体编号; 否则将忽略此参数。如果 code 为 M, 则表示设置多重结构编辑器中的值为变量, operand 表示操作数编号。

示例:

```
SETVAR j+3, R, 1
SETVAR 5, P6, 0
SETVAR surfk+2, E06, status
SETVAR config, M, status, operand
SETVAR 1, O32, 1, 5
```

SETVECSIZE

更改 VEC1、VEC2、VEC3 和 VEC4 数组最大尺寸。

语法:

```
SETVECSIZE n
```

详述:

表达式必须是计算结果为 1 到 18000000 之间的整数变量。所有四个矢量变量的尺寸大小始终相同。在调整尺寸大小期间, 所有矢量数据都会丢失。矢量的初始大小为 1000。

SETVIG(keywords)

设置镜头的渐晕因子。

语法:

```
SETVIG
```

详述:

有关渐晕因子的描述, 请参阅[渐晕因子] (Vignetting factors)。

SHOWBITMAP

在查看器窗口中显示 BMP、JPG 或 PNG 文件。

语法:

```
SHOWBITMAP filename
```

详述:

此关键字需要指定 BMP、JPG 或 PNG 文件的名称。必须包括扩展名。如果文件名中使用了空格或其它特殊字符, 则文件名必须加引号。文件必须位于 <data>\<images> 文件夹中。该命令将打开新的窗口来显示文件。

示例:

```
SHOWBITMAP "BARChart.BMP"
```

SHOWFILE

使用 OpticStudio 文件查看器在屏幕上显示一个文本文件。

语法:

```
SHOWFILE filename, saveflag
```

详述:

filename 必须为有效的文件名。文件必须为文本文件 (如在 ZPL 中通过 OUTPUT 和 PRINT 命令创建的文本), 且必须在当前文件夹中。一旦显示了文本文件, 便可以像其他文件文件一样, 上下滚动数据或者打印。使用 OUTPUT 和 SHOWFILE (而非 PRINT 命令) 的主要优势是能够滚动和打印数据。即使没有执行任何 CLOSE 命令, SHOWFILE 也会关闭文件。如果 saveflag 为 0 或者省略, 则在关闭窗口时同时删除文件。如果 saveflag 是非 0 值, 那么在窗口关闭后文件依然存在。

示例:

```
OUTPUT "test.txt"  
PRINT "Print this to a file."  
SHOWFILE "test.txt"
```

相关关键字:

OPEN、OUTPUT、CLOSE、PRINT、PRINTFILE

SOLVEBEFORESTOP

使 ZPL 求解能够设置在光阑面之前的表面上。

语法:

```
SOLVEBEFORESTOP
```

详述:

此关键字仅用于 ZPL 宏求解。必须将其放置在宏的第一行，以防止 OpticStudio 出现错误信息。使用此关键字意味着用户了解此类求解在什么情况下有效，在什么情况下无效。

在光阑面之前放置的任何求解都不能基于光纤数据。例如，不能使用关键字 RAYTRACE 及其返回的数据。OpticStudio 不会执行任何检查来确保有无基于光纤数据。如果宏确实包含基于光线的数据，则宏求解可能计算出错误数值。

请参阅[使用 ZPL 宏求解] (Using ZPL Macro solves)。

SOLVERETURN

在调用求解的编辑器中返回计算的求解值。

语法:

```
SOLVERETURN x
```

详述:

此关键字仅用于 ZPL 宏求解。宏计算单个值，并通过使用此关键字调用求解，且在编辑器中返回计算出的值。如果存在多个 SOLVERETURN 关键字，则仅考虑最后调一个 SOLVERETURN，之前的调用都会忽略。

如果在宏中发生错误或出现无效条件，且无法计算求解值，则宏直接返回而不调用 SOLVERETURN。若没有调用 SOLVERETURN 则表明无法计算求解，且在光学系统中有错误条件。这在优化过程中尤为重要。

请参阅“使用 ZPL 宏求解”。

SOLVETYPE

更改指定表面上的求解状态和数值。仅支持部分求解类型；有关设置其它类型求解的信息，请联系 OpticStudio 技术支持。

语法:

```
SOLVETYPE surf, CODE, arg1, arg2, arg3, arg4
```


详述:

surf 必须是计算结果为 0 到表面最大数量之间的整数变量。code 必须是下表中列出的助记符。表达式 **arg1** 到 **arg4** 是计算结果为“SOLVES”中指定的第 1 到第 4 个求解参数。注意，对于跨列跟随求解，列编号在[列编号的整数代码](Integer codes for column numbers)中定义。对于非序列跟随求解，变量是 NSC 求解对话框中[求解类型](Solve Type)后面的第一到第四行。某些代码不需要指定所有变量，可以省略部分变量。

关键字 SOLVETYPE 的代码

求解类型	代码
曲率：固定（关闭求解）	CF
曲率：变量	CV
曲率：边缘光线	CM
曲率：主光线	CC
曲率：跟随	CP
曲率：边缘光线法线	CN
曲率：主光线法线	CO
曲率：等光程	CA
曲率：元件光焦度	CE
曲率：同轴面	CQ
曲率：同曲率中心	CR
曲率：F/#	CG
曲率：ZPL 宏	CZ
厚度：固定（关闭求解）	TF
厚度：变量	TV
厚度：边缘光线高度	TM

厚度：主光线高度	TC
厚度：边缘厚度	TE
厚度：跟随	TP
厚度：光程差	TO
厚度：位置	TL
厚度：补偿器	TX
厚度：曲率中心	TY
厚度：光瞳位置	TU
厚度：ZPL 宏	TZ
玻璃：固定（关闭求解）	GF
玻璃：模型	GM
玻璃：跟随	GP
玻璃：替代	GS
玻璃：偏移	GO
半口径：自动	SA
半口径：用户自定义	SU
半口径：跟随	SP
半口径：最大	SM
半口径：ZPL 宏	SZ
圆锥系数：固定（关闭求解）	KF
圆锥系数：跟随	KP

圆锥系数: ZPL 宏	KZ
参数: 固定 (关闭求解)。将 “p” 替换为代码中的参数编号, 例如 PF_3 将关闭对参数 3 的求解。	PF_p
参数: 跟随。将 “p” 替换为代码中的参数编号, 例如 PF_4 将设置对参数 4 的求解。	PP_p
参数: 主光线。将 “p” 替换为代码中的参数编号, 例如 PC_1 将设置对参数 1 的求解。	PC_p
参数: ZPL 宏。将 “p” 替换为代码中的参数编号, 例如 PZ_1 将设置对参数 1 的求解。	PZ_p
热膨胀系数: 固定 (关闭求解)	HF
热膨胀系数: 跟随	HP
附加数据值: 固定 (关闭求解)。将 “e” 替换为代码中的附加数据编号, 例如 EF_3 将关闭对附加数据值 3 的求解。	EF_e
附加数据值: 跟随。将 “e” 替换为代码中的附加数据编号, 例如 EP_4 将设置对附加数据值 4 的求解。	EP_e
附加数据值: ZPL 宏。将 “e” 替换为代码中的附加数据编号, 例如 EZ_4 将设置对附加数据值 4 的求解。	EZ_e
非序列元件跟随 X、Y、Z、X 倾斜、Y 倾斜、Z 倾斜、材料。将 “o” 替换为代码中的物体编号, 例如 NSC_PX_14 将设置对物体 14 的跟随求解。NSC_PMAT_o 与 NSC_MATP_o 相同。	NSC_PX_o、NSC_PY_o、NSC_PZ_o、 NSC_PTX_o、NSC_PTY_o、 NSC_PTZ_o、NSC_PMAT_o (参阅左侧)
非序列元件材料为固定、模型玻璃、跟随或	NSC_MATF_o、NSC_MATM_o、

偏移。将“o”替换为代码中的物体编号，例如 NSC_MATM_11 将物体 11 上的材料设置为模型玻璃。NSC_MATP_o 与 NSC_PMAT_o 相同。	NSC_MATP_o、NSC_MOFF_o (参阅左侧)
非序列元件 ZPL 宏对 X、Y、Z、X 倾斜、Y 倾斜和 Z 倾斜求解。将“o”替换为代码中的物体编号，例如 NSC_ZX_14 将设置对物体 14 的宏求解。	NSC_ZX_o、NSC_ZY_o、NSC_ZZ_o、 NSC_ZTX_o、NSC_ZTY_o、NSC_ZTZ_o (参阅左侧)
非序列元件参数跟随。将“o”替换为物体编号，将“p”替换为代码中的参数编号，例如 NSC_PP_11_7 将设置对物体 14、参数 7 的跟随求解。	NSC_PP_o_p (参阅左侧)
非序列元件 ZPL 宏求解。将“o”替换为物体编号，将“p”替换为代码中的参数编号，例如 NSC_ZP_11_7 将设置对物体 14、参数 7 的宏求解。	NSC_ZP_o_p (参阅左侧)

示例:

```
!The following line will add a glass pickup solve
! on surface 7, picking up from surface 5:
SOLVETYPE 7, GP, 5
!Add a thickness pickup with a scale factor of -1:
SOLVETYPE 7, TP, 5, -1

!Set a pickup solve on surface 1, NSC object 12 Z position,
! pick up from object 11, with a scale factor of 2, offset 3,
! from the parameter 7 column.Note the column number is argument 4.
!The column number is 0 for the same column, 1-6 for x, y, z, tilt x, tilt y, tilt z,
! respectively.The column number for the parameter columns
! is 6 + the desired parameter number.
!In summary, the syntax is:
!SOLVETYPE, surf, code, object, scale, offset, column
! where code has the object/parameter number embedded as shown in the table above.
!The syntax for this example is:
SOLVETYPE 1, NSC_PZ_12, 11, 2, 3, 13
```

相关函数:

SOLV

[求解综述] (Summary of Solves)

求解类型	参数 1	参数 2	参数 3	参数 4	代码	整数
曲率：固定						0
曲率：变量					V	1
曲率：边缘光线角度	角度				M	2
曲率：主光线角度	角度				C	3
曲率：拾取	表面	比例因子		列	P	4
曲率：边缘光线法线					N	5
曲率：主光线法线					N	6
曲率：等光程					A	7
曲率：元件光焦度	光焦度				X	8
曲率：同轴面	同中心的表面				S	9
曲率：同曲率中心	同心的表面				R	10
曲率：F 数	近轴 F/#				F	11
曲率：ZPL 宏	宏名称				Z	12
厚度：固定						0

厚度：变量					V	1
厚度：边缘光线高度	高度	光瞳区域			M	2
厚度：主光线高度	高度				C	3
厚度：边缘厚度	厚度	径 向 高 度 (用零表示 净口径或半 直径)			E	4
厚度：拾取	表面	比例因子	偏移	列	P	5
厚度：光程差	光程差 OPD	光瞳区域			O	6
厚度：位置	表面	到表面的长 度			T	7
厚度：补偿器	表面	表面厚度总 和			S	8
厚度：曲率中心	设置表面在 曲率中心				X	9
厚度：光瞳位置	-	-	-	-	U	10
厚度：ZPL 宏	宏名称				Z	11
玻璃：固定						0
玻璃：模型	折射率 Nd	阿贝数 Vd	Dpgf			1
玻璃：拾取	表面				P	2
玻璃：替换	玻璃库名称				S	3
玻璃：偏移	折 射 率 Nd 偏移	阿 贝 数 Vd 偏移			O	4

半直径：自动						0
半直径：固定					U	1
半直径：拾取	表面	比例因子		列	P	2
半直径：最大值					M	3
半直径：ZPL 宏	宏名称				Z	4
圆锥系数：固定						0
圆锥系数：变量					V	1
圆锥系数：拾取	表面	比例因子		列	P	2
圆锥系数：ZPL 宏	宏名称				Z	3
参数：固定						0
参数：变量					V	1
参数：拾取	表面	比例因子	偏移	列	P	2
参数：主光线	视场	波长			C	3
参数：ZPL 宏	宏名称				Z	4
TCE：固定						0
TCE：变量					V	1
TCE：拾取	表面				P	2

STOPSURF

STOPSURF 通过设置编号重新定义光阑面的位置。

语法:

```
STOPSURF surf
```

详述:

在新数据生效之前必须执行 UPDATE 命令。

示例:

```
STOPSURF n+2
```

相关关键字:

UPDATE

SUB

请参阅 GOSUB。

SURFTYPE

该命令已被废弃。请参阅 SETSURFACEPROPERTY、SURP。

SUSPENDUPDATES

SUSPENDUPDATES 与 RESUMEUPDATES 配合使用。在暂停时，阻止任何 UI 编辑器窗口被 ZPL 命令更新。在执行系统非常多的更新时（比如插入成百上千个评价函数行），此命令可极大提高宏执行速度。暂停和恢复调用的次数都会被统计，只要 RESUMEUPDATES 的次数不同于 SUSPENDUPDATES 的次数，便不会执行任何系统更新，否则宏完成执行。

注意，内部镜头状态仍然会进行更改，只是用户界面不会反映更改。

此外，当调用 RESUMEUPDATES 时，UI 不会自动更新；若要更新 UI，请使用 UPDATE EDITORS 命令。

示例:

```
SUSPENDUPDATES  
n = 5
```



```

FOR i = 1, n, 1
    INSERTMFO i
    SETOPERAND i, 1, ONUM("OPDX")
    SETOPERAND i, 6, (i/n)
    SETOPERAND i, 8, 0
    SETOPERAND i, 9, 1
NEXT
RESUMEUPDATES
UPDATE EDITORS

```

TELECENTRIC

该命令已被废弃。请参阅“SETSYSTEMPROPERTY、SYSP”。

TESTPLATEFIT

TESTPLATEFIT 可调用样板匹配程序。参阅[样板匹配] (Test Plate Fitting)。

语法:

```
TESTPLATEFIT tpd_file, log_file, method, number_cycles
```

详述:

此关键字需要提供样板数据文件的字符串表达式以及输出日志文件的文件名。method 是介于 0 到 4（包含首尾值）之间的整数，分别表示尝试所有方法、最佳到最劣、最劣到最佳、长到短以及短到长。整数 number_of_cycles 为 0 表示自动，否则表示要执行的优化迭代次数。请注意，tpd_file 名称不能包含路径，因为所有样板文件都在固定文件夹中，但对于日志文件，则应该包含路径。

此关键字可能需要很长时间来执行。建议在完成拟合后显示日志文件，或者使用其它方法来表明拟合已完成。

示例:

```

TESTPLATEFIT "optico.tpd", "c:\temp\logfile.dat", 0, 0
SHOWFILE "c:\temp\logfile.dat"

```

THIC

该命令已被废弃。请参阅“SETSURFACEPROPERTY、SURP”。

TIMER

重置内部时钟。此功能与 ZPL 函数 ETIM() 配合使用，来计算从上一次 TIMER 命令以来所经历的时间。

语法:

```
TIMER
```

详述:

TIMER 和 ETIM() 主要用于测试 ZPL 编译器和各种程序结构的执行效率。

示例:

```
i = 0
TIMER
LABEL 1
x = RAND(1000)
i = i+1
if i < 10000 THEN GOTO 1
FORMAT .1
PRINT "Elapsed time:", ETIM(), "Seconds"
```

TOLERANCE

运行公差分析，并将公差报告保存到一个文本文件中。

语法:

```
TOLERANCE top_file_name, out_file_name
```

详述:

TOLERANCE 根据 top_file_name 定义的公差选项文件中的设置，对当前镜头执行公差分析。此文件名必须以扩展名 TOP 结尾，并且必须存储于 <data>\Configs 文件夹中（参阅[文件夹] (Folders)）。通过在公差对话框中保存公差选项来创建 TOP 文件，请参阅[其它按钮] (Other buttons)。outfilename 将是一个文本文件（可以使用任何扩展名），并且文件将与当前镜头文件存储于同一个文件夹中。top_file_name 和 out_file_name 都不能指定路径。

示例:

TOLERANCE example.TOP, output.txt

UNLOCKWINDOW

解锁任何一个或者所有锁定的窗口。

语法:

```
UNLOCKWINDOW winnum
```

详述:

请参阅[图形窗口操作](Graphic windows operations)。如果 winnum 为 0，则将解锁所有已打开的窗口。如果变量 winnum 为 -1，则在宏执行结束后将解锁当前执行的窗口。

示例:

```
UNLOCKWINDOW 2
```

相关关键字:

LOCKWINDOW

UPDATE

更新光瞳位置、折射率数据、近轴常量、净半口径或半口径、最大视场归一化值、求解、非序列物体以及序列和非序列光线追迹所需要的其它数据。如果在上次更新之后又更改了系统数据（比如表面类型、曲率半径、厚度、系统孔径、波长、非序列物体类型或参数），则在追迹或评估系统之前，必须使用关键字 UPDATE。

如果 UPDATE 命令后跟 “ALL”，则还会更新所有打开的窗口（ZPL 宏窗口除外）。如果 UPDATE 命令后跟一个计算结果为整数的表达式，则更新整数指定的窗口，但前提是此窗口不是当前正在执行的宏窗口。如果 UPDATE 后跟 EDITORS，则会更新任何已打开的电子表格编辑器，以显示镜头数据的最新更改。如果 UPDATE 后跟 MACROS，则会更新除当前调用宏窗口之外的其他已打开 ZPL 宏窗口。

语法:

```
UPDATE  
UPDATE ALL  
UPDATE n  
UPDATE EDITORS  
UPDATE MACROS
```

VEC1、VEC2、VEC3、VEC4

这些关键字用于设置变量 VEC1、VEC2、VEC3 和 VEC4。每个矢量可以存储一个双精度浮点数的数组。

语法:

```
VEC1 subscript, value
VEC2 subscript, value
VEC3 subscript, value
VEC4 subscript, value
```

详述:

VEC1-4 用于存储数组中的数据。subscript 的值可以是任何表达式，计算时对其进行向下取整。生成的整数必须介于 0 到当前最大矢量大小之间，初始矢量大小为 1000。ZPL 函数 **VEC1-4** 可用于提取数据。若要更改数组大小，请使用关键字 **SETVECSIZE**。

示例:

```
i = 0
LABEL 1
i = i + 1
VEC1 i, i
IF i < 10 THEN GOTO 1
j = 0
LABEL 2
j = j + 1
VEC2 j, VEC1(j) * VEC1(j)
IF j < 10 THEN GOTO 2
i = 0
LABEL 3
i = i + 1
PRINT "x = ", VEC1(i), " x*x = ", VEC2(i)
IF i < 10 THEN GOTO 3
PRINT
PRINT "All done!"
```

WAVL、WWGT

这些命令已被废弃。请参阅“SETSYSTEMPROPERTY、SYSP”。

XDIFFIA

计算[扩展衍射图像分析](Extended Diffraction Image Analysis)功能，并将结果保存到一个 ZBF 文件中。有关 ZBF（[Zemax 光束文件](Zemax Beam File)）格式的说明，请参阅[Zemax 光束文件 (ZBF) 二进制格式](Zemax Beam File (ZBF) binary format)。有关扩展衍射图像分析功能的描述，请参阅 [扩展衍射图像分析](Extended Diffraction Image Analysis)。

语法:

```
XDIFFIA outfilename, infilename
```

详述:

此关键字需指定输出 ZBF 文件的名称以及可选择性输入 IMA 或 BIM 文件的名称。如果未提供 outfile 的扩展名，则将默认扩展名为 ZBF。必须在 infile 中提供扩展名。如果文件名中使用了空格或其它特殊字符，则文件名必须加引号。outfile 会存储于 <pop> 文件夹中，infile 必须存放在 <data>\<images> 文件夹中。文件名中不能包含路径。

扩展衍射图像分析功能的设置是之前为当前镜头保存的设置。要修改这些设置，请打开[扩展衍射图像分析](Extended Diffraction Image Analysis)窗口并选择适当的设置，然后点击 [保存](Save)。XDIFFIA 的所有后续调用都将使用已保存的设置。输出文件名和输入源文件例外，输出文件名为 XDIFFIA 关键字之后的第一个变量，输入源文件为 IMA 关键字之后的第二个可选变量。

示例：

```
XDIFFIA "output will be in this file.ZBF", "SOMEIMAFILE.IMA"
```

相关函数：

ZBFCLR、ZBFSHOW、ZBFSUM、ZBFMULT

ZBF2MAT

将 ZBF 文件转换为 Matlab MAT 文件。

语法：

```
ZBF2MAT zbffilename, matfilename, surfacerange, integer1, integer2
```

详述：

ZBF2MAT 的前两个变量是字符串，分别提供要转换的 ZBF 文件全名和要创建的 MAT 文件全名，包括路径和扩展名。surfacerange 是一个整数标记，指定要转换个别表面上的 ZBF 文件，或者指定表面范围内的一系列 ZBF 文件。如果 surfacerange 设置为 0，则可以忽略变量 integer1 和 integer2。如果 surfacerange 不为 0，则 integer1 是要转换的起始面，integer2 是要转换的终止面。需注意，ZBF 文件名有一些限制，参阅[MAT 文件生成器](MAT File Generator)。

ZBFCLR

清除 ZBF 文件中的复振幅数据。

语法：

```
ZBFCLR filename
```

详述：

此关键字只需要提供 ZBF 文件名。

所有 ZBF 相关关键字的通用说明：

有关 ZBF（Zemax 光束文件）格式的说明，请参阅 [Zemax 光束文件 (ZBF) 二进制格式](Zemax Beam

File (ZBF) binary format)。OpticStudio 要求文件的扩展名为 ZBF，并且会根据需要来添加或替换扩展名。如果文件名中使用了空格或其它特殊字符，则文件名必须加引号。文件必须存储于<pop>文件夹中，所有 ZBF 输出文件都会存储于此文件夹中。

示例:

```
ZBFCLR "some beam file name.ZBF"  
ZBFCLR N$
```

ZBFMULT

将 ZBF 文件中的复振幅数据乘以一个复数因子。

语法:

```
ZBFMULT filename, Ax, Bx, Ay, By
```

详述:

此关键字需要提供 ZBF 文件的名称，以及复数的实部 (A) 和虚部 (B)乘以 ZBF 文件中的每个点。对于 x 和 y 偏振光方向，有单独的 A 和 B。所得的数据将重新写入到相同的文件名中。

关于所有 ZBF 关键字的注解说明，请参阅[用途](Purpose)。

示例:

```
ZBFMULT "some beam file name.ZBF", 0.0, 1.0, 0.0, -1.0
```

ZBFPROPERTIES

打开指定的 ZBF 文件，并将各种光束数据存储 在某个矢量变量中。

语法:

```
ZBFPROPERTIES filename, vector
```

详述:

此关键字需要提供 ZBF 文件的名称，以及用于放置数据的矢量编号。vector 的值必须是 1 到 4 之间。在执行 ZBFPROPERTIES 函数后，以下光束数据会放置在指定的矢量中：nx、ny、dx、dy、waist_x、waist_y、position_x、position_y、rayleigh_x、rayleigh_y、波长（以镜头单位表示）、总功率、峰值照度（单位面积的能量）、is_polarized 标记（0 表示否，1 表示是）、介质折射率；光束数据分别存储于矢量位置 1 到 15。

关于所有 ZBF 关键字的注解说明，请参阅[用途](Purpose)。

示例:

```
ZBFPROPERTIES "TEST.ZBF", 1  
rayleighx = VEC1(9)
```

ZBFREAD

打开指定的 ZBF 文件，并将电场和光束属性数据放置在两个用户自定义的数组变量中。

语法:

```
ZBFREAD filename, beamname, propertyname
```

详述:

此关键字需要提供 ZBF 文件名，和之前调用 DECLARE 所定义的两个数组名称。beamname 必须是一个三维数组，非偏振光束的最小尺寸为(nx, ny, 2)，偏振光束的最小尺寸为 (nx, ny, 4)。propertyname 数组必须是最小尺寸为 14 的一维数组。在执行 ZBFREAD 函数后，以下光束数据存储在指定 propertyname 数组中：nx、ny、dx、dy、waist_x、waist_y、position_x、position_y、rayleigh_x、rayleigh_y、波长（以镜头单位表示）、总功率、峰值照度（单位面积的能量）、is_polarized 标记（0 表示否，1 表示是）、介质折射率；光束数据分别存储于矢量位置 1 到 15。电场数据将存储于 beamname 数组中。beamname 数组的第三维为 1 表示 Ex 实部，2 表示 Ex 虚部，如果光束偏振，3 表示 Ey 实部，4 表示 Ey 虚部。

关于所有 ZBF 关键字的注解说明，请参阅[用途](Purpose)。另请参阅 ZBFWRITE。

示例:

```
!First get the beam size
ZBFPROPERTIES "TEST1.ZBF", 1
nx = vec1(1)
ny = vec1(2)
ip = vec1(14) !The "is polarized" flag

!Allocate enough memory to hold the beam
IF (ip == 0) THEN DECLARE B, DOUBLE, 3, nx, ny, 2
IF (ip == 1) THEN DECLARE B, DOUBLE, 3, nx, ny, 4
DECLARE P, DOUBLE, 1, 20
ZBFREAD "test1.zbf", B, P

FOR j, 1, ny, 1
    FOR i, 1, nx, 1
        FORMAT 4.0
        PRINT i, j,
        FORMAT 12.6
        IF (ip == 1)
            PRINT B(i, j, 1),
            PRINT B(i, j, 2),
            PRINT B(i, j, 3),
            PRINT B(i, j, 4)
        ELSE
            PRINT B(i, j, 1),
            PRINT B(i, j, 2)
        ENDIF
    NEXT
NEXT
```

```
! save the beam
ZBFWRITE "TEST2.ZBF", B, P

! release the allocated memory
RELEASE B
RELEASE P
```

ZBFRESAMPLE

按照新的宽度和点间距，重新采样 ZBF 文件。

语法:

```
ZBFRESAMPLE filename, nx, ny, wx, wy, decenterx, decentery
```

详述:

此关键需要提供 ZBF 文件的名称和六个数字。光束将按要求重新采样并插值，以创建一个在 x 和 y 方向分别有 nx 和 ny 个点的光束文件，总宽度为 wx 和 wy。nx 和 ny 值必须是 2 的幂，比如 32、64、128 等等。可选择性提供 decenterx 和 decentery 值，用于将光束偏心。如果 nx 或 ny 为 0，则不会对现有光束采样进行任何更改。如果 wx 或 wy 为 0，则不会对现有光束宽度进行任何更改。ZBF 文件中的长度单位自动转换为当前镜头单位。所得的数据将重新写入到相同的文件名中。

示例:

```
ZBFRESAMPLE "TEST.ZBF", 128, 128, 25.4, 25.4, 0, 0.4
```

ZBFSHOW

在查看器窗口中显示 ZBF 文件。

语法:

```
ZBFSHOW filename
```

详述:

此关键字仅需要提供 ZBF 文件名。此命令将打开一个新窗口来显示光束文件。

示例:

```
ZBFSHOW "new beam data.ZBF"
ZBFSHOW N$
```


ZBFSUM

对两个 ZBF 文件中的数据进行相干或非相干求和，并将生成的数据存储于第三个 ZBF 文件中。

语法:

```
ZBFSUM coherent, filename1, filename2, outfilename
```

详述:

此关键字需要提供一个整数，来表示求和为相干或者非相干，0 以外的任何值表示相干，0 表示非相干。若执行了非相干求和，则输出数据只有实数。如果两个源文件在 x 和 y 方向的点数量、点间距和参考半径不同，那么列出的第二个源文件会进行缩放、插值或者调整参考半径，以匹配第一个源文件。ZBF 文件中的长度单位自动转换为当前镜头单位。outfilename 可以与其中一个源文件名称相同，在此情况下，将覆盖原始文件。

示例:

```
ZBFSUM 1, "a.ZBF", "b.zbf", "coherent a plus b.zbf"  
ZBFSUM 0, "a.ZBF", "b.zbf", "incoherent a plus b.zbf"  
ZBFSUM 0, A$, B$, C$
```

ZBFTILT

将 ZBF 文件中的数据乘以一个复数相位因子，以在光束中引入相位倾斜。

语法:

```
ZBFTILT filename, cx, cy, tx, ty
```

详述:

此关键需要提供 ZBF 文件的名称和四个数字。光束的相位将按照 $\theta = (x - cx)tx + (y - cy)ty$ 提供的相位角度进行修改。cx 和 cy 值是相位倾斜的中心，tx 和 ty 是倾斜的斜率(弧度每镜头单位长度)。x 和 y 指的是光束文件中的位置，中心坐标 (x = 0, y = 0) 位于点 (nx/2 + 1, ny/2 + 1)，其中 nx 和 ny 是 x 和 y 方向的点数。ZBF 文件中的长度单位自动转换为当前镜头单位。所得的数据将重新写入到相同的文件名中。

示例:

```
ZBFTILT "TEST.ZBF", 0.0, 0.0, 0.0, 0.01237
```

ZBFWRITE

将电场和光束属性数据数组写入到一个 ZBF 文件中。

语法:

```
ZBFWRITE filename, beamname, propertyname
```

详述:

此关键字需要提供 ZBF 文件名, 和之前调用 DECLARE 所定义的两个数组名称。beamname 必须是一个三维数组, 非偏振光束的最小尺寸为(nx, ny, 2), 偏振光束的最小尺寸为 (nx, ny, 4)。propertyname 数组必须是最小尺寸为 14 的一维数组。在执行 ZBFREAD 函数后, 以下光束数据存储在指定 propertyname 数组中: nx、ny、dx、dy、waist_x、waist_y、position_x、position_y、rayleigh_x、rayleigh_y、波长 (以镜头单位表示)、总功率、峰值照度 (单位面积的能量)、is_polarized 标记 (0 表示否, 1 表示是)、介质折射率; 光束数据分别存储于矢量位置 1 到 15。电场数据将存储于 beamname 数组中。beamname 数组的第三维为 1 表示 Ex 实部, 2 表示 Ex 虚部, 如果光束偏振, 3 表示 Ey 实部, 4 表示 Ey 虚部。

另请参阅 ZBFREAD。

ZRD2MAT

将 ZRD 文件转换为 Matlab MAT 文件。

语法:

```
ZRD2MAT zrdfilename, matfilename, savesegment, integer1, integer2
```

详述:

ZRD2MAT 的前两个变量是字符串, 分别提供要转换的 ZBF 文件全名和要创建的 MAT 文件全名, 包括路径和扩展名。savesegment 是一个整数标记, 指定要转换 ZRD 文件中的单个光线, 或者一系列光线。如果 savesegment 设置为 0, integer1 表示要进行的转换的第一束光线编号, integer2 表示最后一束光线的编号。如果 savesegment 不为 0, 则仅转换一段光线; 此种情况下, integer1 是光线编号, integer2 是光线段编号。

ZRDAPPEND

将一个 ZRD 文件中的数据附加到另一个文件的末尾, 第一个文件保持不变。只有在同一系统中追踪两个 ZRD 文件时此命令才有效, 这样不会修改物体编号或者坐标。

语法:

```
ZRDAPPEND infilename, outfilename
```

详述:

此关键字需要提供两个 ZRD 文件的名称。有关 ZRD 文件格式的信息, 请参阅 “[光线数据库 (ZRD) 文件] (Ray database (ZRD) files)”。可以通过文本字符串或字符串变量定义每个文件名。文件名必须包含完整

路径名。建议文件名使用 **ZRD** 扩展名。输入和输出文件名不能相同，且必须使用相同的 **ZRD** 格式。另请参阅 **ZRDSUM**。

示例：

```
ZRDAPPEND "C:\TEMP\CART.ZRD", "C:\TEMP\HORSE.ZRD"
```

ZRDFILTER

打开 **ZRD** 光线数据库文件，应用过滤，并将过滤后的光线子集保存到一个新的 **ZRD** 文件中。

语法：

```
ZRDFILTER infilename, outfilename, filterstring
```

详述：

此关键字需要提供输入 **ZRD** 文件名称、过滤后的输出 **ZRD** 文件名称以及过滤字符串。有关 **ZRD** 文件格式的信息，请参阅 “[光线数据库 (ZRD) 文件](Raydatabase (ZRD) files)”。有关过滤字符串格式的信息，请参阅 “[过滤字符串](The filter string)”，可以通过文本字符串或者字符串变量定义每个文件名和过滤字符串。文件名必须包含完整路径名。建议文件名使用 **ZRD** 扩展名。输入和输出文件名不能相同。

示例：

```
ZRDFILTER "C:\TEMP\TEST1.ZRD", "C:\TEMP\TEST1_H4.ZRD", "H4"
```

ZRDPLAYBACK

读取一个 **ZRD** 文件，并将光线振幅数据添加到一个或所有探测器中。只有在通过当前加载的系统生成 **ZRD** 文件时才可使用此命令，这样不会修改物体编号或者坐标。

语法：

```
ZRDPLAYBACK zrdfilename, surface, detector, clear, filterstring
```

详述：

此关键字需要提供 **ZRD** 文件名称。有关 **ZRD** 文件格式的信息，请参阅 “[光线数据库 (ZRD) 文件](Raydatabase (ZRD) files)”。可以通过文本字符串或字符串变量定义每个文件名。文件名不能包含路径，并且文件必须与当前镜头文件在同一个文件夹中。在混合序列/非序列模式下，变量 “**surface**” 表示非序列组件的表面编号，在非序列模式下，变量 “**surface**” 设为 1。整数 “**detector**” 表示指定探测器的物体编号，为 0 表示所有探测器。如果 “**clear**” 设置为非 0 值，则在添加光线振幅数据之前清除指定探测器。可选的 “**filterstring**” 在添加光线振幅数据之前应用于光线。

示例：

```
ZRDPLAYBACK "MYRAYS.ZRD", 1, 0, 1, "H3 & H5"
```

ZRDSAVERAYS

打开 ZRD 光线数据库文件，应用过滤，并保存过滤子集中与指定物体编号相交的光线。

语法:

```
ZRDSAVERAYS infilename, outfilename, filterstring, object
```

详述:

此关键字需要提供输入 ZRD 文件名称、输出源光线 DAT 文件的名称、过滤字符串以及物体编号。有关 ZRD 文件格式的信息，请参阅“[光线数据库 (ZRD) 文件](Raydatabase (ZRD) files)”。源光线文件在“[光源文件](Source File)”中定义。有关过滤字符串格式的信息，请参阅“[过滤字符串](The filter string)”。可以通过文本字符串或字符串变量定义每个文件名和过滤字符串。文件名必须包含完整路径名。建议输入文件名使用 ZRD 扩展名，输出文件使用 DAT 扩展名。输入和输出文件名不能相同。如果不需要过滤，则变量“filterstring”应替换为两个双引号，如下所示：""。

示例:

```
ZRDSAVERAYS "C:\TEMP\TEST1.ZRD", "C:\TEMP\TEST1_H6.DAT", "", 6  
ZRDSAVERAYS "C:\TEMP\TEST2.ZRD", "C:\TEMP\TEST2_H5.SDF", "", 5
```

ZRDSUM

合并两个 ZRD 文件。只有在同一系统中追迹两个 ZRD 文件时此命令才有效，这样不会修改物体编号或者坐标。*语法:*

```
ZRDSUM infilename1, infilename2, outfilename
```

详述:

此关键字需要提供两个输入 ZRD 文件的名称和输出 ZRD 文件的名称。请参阅“[光线数据库 (ZRD) 文件](Ray database (ZRD) files)”。可以通过文本字符串或字符串变量定义每个文件名。文件名必须包含完整路径。建议文件名使用 ZRD 扩展名。输入和输出文件名不能相同，两个输入文件必须使用相同的 ZRD 格式，输出文件也会使用相同的 ZRD 格式。另请参阅 ZRDAPPEND。

示例:

```
ZRDSUM "C:\TEMP\TEST1.ZRD", "C:\TEMP\TEST2.ZRD", "C:\TEMP\TEST_SUM.ZRD"
```

示例宏 1

以下示例宏将输出每个已定义视场角主光线与像面的交点坐标。

```
nfield = NFLD()
maxfield = MAXF()
n = NSUR()
FOR i, 1, nfield, 1
    hx = FLDX(i)/maxfield
    hy = FLDY(i)/maxfield
    PRINT "Field number ", i
    RAYTRACE hx,hy,0,0,PWAV()
    PRINT "X-field angle : ",FLDX(i)," Y-field angle : ", FLDY(i)
    PRINT "X-chief ray: ",RAYX(n), " Y-chief ray: ", RAYY(n)
    PRINT
NEXT
PRINT "All Done!"
```

宏的第一行调用 **NFLD()** 函数，此函数返回已定义的视场数量，并将其赋值给变量“numfield”。第二行调用 **MAXF()**，返回最大径向视场，并将其存储在变量“maxfield”中。通过调用函数 **NSUR()**，将表面数量存储在变量“n”中。最后定义 **FOR** 循环，以 **i** 作为视场位置的计数器，从 1 开始，循环的最大值为 **nfield**，增量为 1。定义“hx”和“hy”的两行语法中使用了函数 **FLDX()** 和 **FLDY()**，分别返回当前视场位置编号“i”的 x- 和 y- 视场值。主光线通过关键字 **RAYTRACE** 进行追迹。注意，主光线与光瞳的中心相交，所以两个光瞳坐标均为 0。**PWAV()** 函数返回主波长编号，通常用于主光线。各种 **PRINT** 命令将像面上的主光线坐标输出到屏幕中。

示例宏 2

以下示例宏将估算当前光学系统的轴向 **RMS** 光斑半径。通过系统追迹大量随机光线，并记录这些光线与主光线之间的径向偏移。然后应用当前波长权重估算 **RMS** 光斑半径。

```
PRINT "Primary wavelength is number ",
FORMAT .0
PRINT PWAV(),
FORMAT .4
PRINT " which is ", WAVL(PWAV()), " micrometers."
PRINT "Estimating RMS spot radius for each wavelength."

!How many random rays to trace to make estimate?
n = 100

!Initialize the timer
TIMER
```

```

!Store the number of surfaces for later use
ns = NSUR()

!Start at wavelength 1
weightsum = 0
wwrms = 0
FOR w, 1, NWAV(), 1
    rms = 0
    FOR i, 1, n, 1
        hx = 0
        hy = 0
        angle = 6.283185 * RAND(1)
        !SQRT yields uniform distribution in pupil
        radius = SQRT(RAND(1))
        px = radius * COSI(angle)
        py = radius * SINE(angle)
        RAYTRACE hx, hy, px, py, w
        x = RAYX(ns)
        y = RAYY(ns)
        rms = rms + (x*x) + (y*y)
    NEXT
    rms = SQRT(rms/n)
    wwrms = wwrms + (WWGT (w) * rms)
    weightsum = weightsum + WWGT(w)
    FORMAT .4
    PRINT "RMS spot radius for ", WAVL(w),
    FORMAT .6
    PRINT " is ", rms
NEXT
wwrms = wwrms / weightsum
PRINT "Wavelength weighted rms is ", wwrms
FORMAT .2
t = ETIM()
PRINT "Elapsed time was ",t," seconds."

```

注意在前两条 PRINT 命令中使用了尾部逗号。表示前三个 PRINT 命令输出的数据均在同一行中显示。同一行中的 FORMAT 命令只是更改输出数值的显示方式。“!”字符表示注释，在运行宏时将忽略注释。

[从宏中调用宏](Calling a Macro from within a Macro)

若要从 ZPL 宏中调用另一个 ZPL 宏，请使用关键字 CALLMACRO。使用关键字 CALLSETDBL 和 CALLSETSTR 以及函数 CALD 和 \$CALLSTR 在宏之间传递数据。

通常情况下，从“ZPL 宏”对话框中执行的第一个宏是父宏。父宏调用其它宏，并且这些宏称为“子”宏，父宏将创建 51 个数值和 51 个字符串的缓存区。缓存区可用于设置或检索数字和字符串值，这样可以使宏共享数据，或者将某个变量传递到另一个变量。缓存区对于每个父宏及其调

用的任何子宏都是唯一的。如果有多个父宏同时执行（例如，在两个窗口中分别更新不同的宏），则每个父宏都有自己的缓存区，并且不会在父项之间共享数据。

要了解父宏和子宏的工作方式，最简单的方法是通过示例。以下示例将调用两个很简单的宏。对可以调用或嵌套的宏数量没有硬性限制。

父宏和子宏分别命名为 **PARENT.ZPL** 和 **CHILD.ZPL**。**PARENT.ZPL** 宏为：

```
CALLSETDBL 1, 3.5
CALLSETSTR 1, "Hello World"
CALLMACRO CHILD.ZPL
PRINT CALD(1)
```

CHILD.ZPL 宏为：

```
PRINT "Executing child macro"
PRINT CALD(1)
A$ = $CALLSTR(1)
PRINT A$
CALLSETDBL 1, 7.11
```

在执行 **PARENT.ZPL** 宏时，使用关键字 **CALLSETDBL** 将 3.5 放在父宏数字缓存区位置 1 处。使用关键字 **CALLSETSTR** 将字符串 “Hello World” 放置在父宏字符串缓存区位置 1 处。可使用 0 到 50 来定义任何数字或字符串在缓存区的位置。

PARENT.ZPL 宏随后调用子宏 **CHILD.ZPL**。子宏将消息 “Executing child macro” 输出到子宏的显示窗口，但此输出窗口在执行期间不可见。子宏随后使用 **CALD(1)** 来提取数值缓存区中位置 1 处的数值，并使用 **\$CALLSTR(1)** 来提取字符串缓存区位置 1 处的字符串值，并输出这些值。最后，子宏使用 7.11 覆盖原索引位置 1 处的数值，到此子宏执行结束。

控制会返回到父宏中，将子宏输出窗口的内容复制到父项的输出窗口，并清除子宏输出窗口的内容。最后输出数值缓存区位置 1 处修改后的数值。执行此宏的最终输出结果如下所示：

```
Executing child macro
3.5000
Hello World
7.1100
```

OpticStudio 中的 “宏” 文件夹中附带有样本宏 **PARENT.ZPL** 和 **CHILD.ZPL**。

[使用 ZPL 宏求解](Using ZPL Macro Solves)

ZPL 宏求解可调用用户自定义的 ZPL 来定义求解值。有关示例，请参阅关键字 “**SOLVEBEFORESTOP**”。有关求解的更多信息，请参阅镜头数据编辑器的 “[设置菜单](Setup Tab)” 的 “[求解](Solves)” 栏目。

宏求解调用用户自定义的 ZPL 宏来计算求解。任何可在宏中计算的数值都可以返回到调用求解宏的编辑器中。宏可利用编辑器中其他位置的数据，比如之前的表面。一旦计算出数据，可使用关

键字 **SOLVERETURN** 将数据传递回编辑器。

举一个简单的例子，下面是一个宏，用于计算表面 1 和 2 之间的一阶光焦度，并且返回光焦度作为求解值：

```
n1 = INDX(1)
n2 = INDX(2)
c2 = CURV(2)
SOLVERETURN (n2-n1)*c2
```

上例宏中直接通过表面编号来引用特定表面。这样有一个缺点，如果插入或删除了新表面，则需要修改宏中对应的表面编号。此外，如果宏求解引用固定的表面编号，则某些功能不会正常运行，比如镜面基底绘制。在这种情况下，可使用 **ZPL** 函数 **SURC** 在镜头数据编辑器的注释列中查找包含特定文本的表面。假定在镜头数据编辑器中的第一个表面上输入了注释“My Surface”。可以通过以下修订后的宏来计算上述光焦度：

```
A$ = "My Surface"
SURF = SURC(A$)
n1 = INDX(SURF)
n2 = INDX(SURF+1)
c2 = CURV(SURF+1)
SOLVERETURN (n2-n1)*c2
```

也可以使用 **ZPL** **SOSO** 数值函数来提取需要的表面或物体编号。

在求解宏中尽可能使用表面注释和 **SURC** 函数。

[有关 ZPL 宏求解的重要注意事项](Important Considerations for ZPL Macro Solves)

宏求解非常通用，基本上可以使用任何计算来确定求解值。可以使用 **ZPL** 支持的所有函数和关键字。通过宏菜单执行 **ZPL** 宏与作为求解执行 **ZPL** 宏并没有任何区别。但是，某些 **ZPL** 关键字和函数不能在宏求解中使用。例如，如果从求解中调用“**UPDATE**”将更新所有求解，这将再次调用宏，从而导致无限循环。如果宏求解中使用了关键字 **INPUT**，那么每次调用求解都需要用户输入数据，这可能需要操作多次。宏求解中也不能使用在编辑器中设置数值的关键字和创建评价函数的关键字。通常，宏求解应简短、简单，避免冗长计算，并且不能修改任何镜头数据。宏求解不能取决于编辑器中后续求解的数据，这样会产生错误数据。因为如果先调用求解，然后后续求解又修改了源数据，那么前面的宏求解也会出错。**OpticStudio** 不会自动进行验证求解宏。求解宏功能虽然强大并且灵活，但必须谨慎使用。

只有当宏中的计算不基于任何光线数据时，宏求解才能放置在光阑面之前。有关更多详情，请参阅 **ZPL** 关键字“**SOLVEBEFORESTOP**”的描述。

如果在宏中出现错误或无效条件，且无法计算求解值，则宏不会调用 **SOLVERETURN**。缺少 **SOLVERETURN** 的调用表明无法计算求解，且在光学系统中出现错误条件。这在优化过程中尤为重要。

[列编号的整数代码](Integer Codes for Column Numbers)

如果要通过 ZPL 宏语言设置跟随求解，则列编号是必需提供的。列编号的定义方式如下：**0**：当前列，即求解所放置的列。这是所有跟随求解的默认设置。

1-4：分别表示半径、厚度、圆锥系数和半口径。

5-17：分别表示参数 **0** 到 **12**