

# 目录

使用流程控制的分支 .....	1
使用流程控制分支：简介 .....	1
使用控制对 .....	2
If / End If .....	2
Else If / End Else If .....	3
Else / End Else .....	6
While / End While .....	7
Do / Until .....	9
Select / End Select .....	11
Case / End Case .....	14
Default Case / End Default Case .....	15
使用一般循环 .....	16
循环的用途 .....	18
创建循环 .....	19
循环使用打印命令 .....	20
使用标签 .....	22
使用 GOTO 跳至标签 .....	23
出错时分支 .....	25
出错支持界面 .....	28

子例程分支.....	29
新建子例程 .....	30
编辑现有子例程 .....	35
了解“自变量编辑”对话框 .....	35
调用子例程 .....	36
子例程示例 .....	45
终止测量例程 .....	52

# 使用流程控制的分支

---

## 使用流程控制分支：简介

假定您的零件包含许多特征，但是您只要反复测量一些特征，获得这些特征的一组全面的统计数据。假定您根据用户的响应跳转的测量例程中的特定部分。使用流程控制命令可以完成此类任务以及许多其它任务。通过设置某些命令的条件，您可以控制测量例程的流程。

本章将介绍完成此类任务所需的信息。包括语法条件语句、循环和子例程。同时还提供了许多样例代码。

此章包括以下主要主题：

- 使用控制对
- 使用一般循环
- 使用标签
- 出错时分支
- 子程序分支
- 终止测量例程

### 注释后的命令模式命令

由于本章中的许多代码示例使用类型化 `COMMENT` 命令，请考虑以下事项：



插入 PC-DMIS 注释后，要在命令模式下键入其他 PC-DMIS 命令，必须先在 `COMMENT` 命令后按 **Enter** 两次。这告诉 PC-DMIS 您不再想在注释中添加文本，而是准备添加新命令。

# 使用控制对

“插入| 流程控制命令 | 控制对”子菜单提供了多个成对的命令，可以在“编辑”窗口中用来控制测量例程的正确流程。要在“编辑”窗口中插入控制对类型的命令，只需键入相应命令，或从该子菜单中选择命令。



使用条件分支语句测试 YES / NO 注释值时，应注意测试查找的是大写字母的“YES”或“NO”值。小写字母“Yes”或“No”不起作用。有关注释的信息，请参见“插入报告命令”一章中的“插入程序员注释”主题。

## If / End If

插入 | 流控制命令 | 控制对 | If / End If 菜单项允许您添加条件块到测量例程。仅在 IF 命令的表达式评估为 true (非 0) 时，方执行 IF 和 END IF 命令之间的项目。否则，执行流将跳至 END/IF 命令之后的第一条命令。

IF / END IF 语句在“编辑”窗口中的命令行为：

```
IF/expression
END_IF/
```

要插入 If / End If 命令，请执行以下步骤：

1. 将光标放在“编辑”窗口中的所需位置。
2. 从菜单栏中选择 **If / End If**。IF / END IF 语句将显示在“编辑”窗口中。

## If / End If 样例代码

参见以下样例，询问用户是否要测量点特征。

```
C1= 注释/是否, 是否要测量点特征 PNT1 ?
IF/C1.INPUT=="YES"
```

使用控制对

```
PNT1=特征/点, 直角  
...  
...  
ENDMEAS/  
END_IF/
```

## 样例代码的说明

**C1=COMMENT/YESNO**

此行取用并存储用户的 YES 或 NO 响应。

**IF/C1.INPUT=="YES"**

此行是表达式。它测试注释 1 的输入是否为 YES。若为 YES，则 IF 语句为 TRUE，并在 IF 语句后继续执行语句，在本例中，测量的是 PNT1 特征。若为 NO，则移至 END\_IF 语句。

**END\_IF**

此行结束执行 IF / END IF 代码块中的命令。若用户在此注释处单击 **No**，PC-DMIS 将转至此行后面的代码。

## Else If / End Else If

**插入 | 流控制命令 | 控制对 | Else If / End Case If** 菜单项允许您添加条件块到测量例程。仅在 ELSE IF 命令的表达式评估为 True (非 0) 时方执行 ELSE IF 和 END ELSE IF 命令之间的项目。ELSE IF / END ELSE IF 块必须紧跟在 IF / END IF 块或其他 ELSE IF / END ELSE IF 块之后。若当前块以上的所有 IF / ELSE IF 表达式评估为 False，则将评估此表达式。若此表达式评估为 false (0)，则执行将移转至 END ELSE IF 命令之后的下一命令。若当前块以上的任何 IF / ELSE IF 表达式均评估为 True，则将跳过此序列中所有后续 ELSE IF / END ELSE IF 块。

ELSE IF / END ELSE IF 语句在“编辑”窗口中的命令行为：

```
ELSE_IF/expression  
END_ELSE_IF/
```

要插入 ELSE IF / END ELSE IF 命令，请执行以下步骤：

1. 将光标放在“编辑”窗口中现有 IF/END IF 语句或 ELSE IF/END ELSE IF 语句后面的所需位置。
2. 从菜单栏中选择 **Else If / End Else If**。ELSE IF / END ELSE IF 语句将显示在“编辑”窗口中。



只有放置于 IF / END IF 或 ELSE IF / END ELSE IF 块之后，此类命令块才有效。无效放置的控制对在“编辑”窗口中将显示为红色文本。

## Else If / End Else If 样例代码

参见以下样例，在测定点的任何一个 X、Y 或 Z 值超过定义的公差时显示一条消息通知用户：

```
PNT2=FEAT/POINT,RECT
```

```
...
```

```
...
```

```
ENDMEAS/
```

```
IF/PNT2.X<6.9 OR PNT2.X>7.1
```

```
    COMMENT/OPER,"The measured X value of PNT2: " + PNT2.X + " is out of  
tolerance."
```

```
END_IF/
```

```
ELSE_IF/PNT2.Y<3.3 OR PNT2.Y>3.5
```

```
    COMMENT/OPER,"The measured Y value for PNT2: " + PNT2.Y + " is out of  
tolerance."
```

```
END_ELSEIF/
```

```
ELSE_IF/PNT2.Z<.9 OR PNT2.Z>1.1
```

## 使用控制对

```
COMMENT/OPER,"The measured Z value for PNT2: " + PNT2.Z + " is out of  
tolerance."  
  
END_ELSEIF/
```

### 样例代码的说明

该代码首先测试点的 **X** 值。如果条件值为假，代码将测试 **Y** 值。如果 **Y** 值的条件值为假，代码将测试 **Z** 值。

如果其中任一条件值为真，**PC-DMIS** 会显示关联的注释，并跳过其它条件语句。

#### IF/PNT2.X7.1

此行是表达式。用于测试 **X** 测量值是否小于 6.9 或大于 7.1。若超出其中任何一个界限，将执行第一条注释。

#### END\_IF

此行结束执行 **IF / END IF** 代码块中的命令。若 **IF THEN** 条件评估为 **false**，**PC-DMIS** 将转至此行后面的命令。

#### ELSE\_IF/PNT2.Y3.5

此行是第一条 **ELSE\_IF** 命令的表达式。它仅在其上面的 **IF / END IF** 块传回 **false** 时执行。此代码行测试 **Y** 测量值是否小于 3.3 或大于 3.5。若超出这两个界限中的任何一个界限，则执行第二条注释。

#### END\_ELSEIF/

此行结束执行第一个 **ELSE IF / END ELSE IF** 代码块中的命令。

#### ELSE\_IF/PNT2.Z1.1

此行是第二条 **ELSE IF** 命令的表达式。仅在其上的 **ELSE IF / END ELSE IF** 块传回 **False** 时，方执行此表达式。用于测试 **Z** 测量值是否小于 0.9 或大于 1.1。若超出其中任何一个界限，将执行第三条注释。

#### END\_ELSEIF/

此行结束执行第二个 **ELSE IF / END ELSE IF** 代码块中的命令。

## Else / End Else

**插入 | 流控制命令 | 控制对 | Else / End Else** 菜单项允许您添加条件块到测量例程。仅在此 **else** 块上方的所有其它 **if / end if** 和 **else if / end else if** 块评估为 **false** 时（均评估为 0），方执行 **ELSE** 和 **END ELSE** 命令之间的项目。ELSE / END ELSE 块必须位于一组 **IF / END IF** 或 **ELSE IF / END ELSE IF** 块的末尾方有效。

ELSE / END ELSE 语句在“编辑”窗口中的命令行为：

```
ELSE/  
END_ELSE/
```

要插入 **Else / End Else** 命令，请执行以下步骤：

1. 将光标放在 *编辑窗口* 中的所需位置。请注意，**Else / END ELSE** 块必须位于 **IF / END IF** 块或 **ELSE IF / END ELSE IF** 块之后。
2. 从菜单栏中选择 **Else / End Else**。ELSE / END ELSE 语句将显示在“编辑”窗口中。

## Else / End Else 样例代码

参见以下样例，询问用户是否要测量点特征。

C1=        注释/是否, 是否要测量点特征 PNT1? 点击否测量下一个特征。

```
IF/C1.INPUT=="YES"
```

```
    PNT1=FEAT/POINT, RECT
```

```
    ...
```

```
    ...
```

```
    ENDMEAS/
```

```
END_IF/
```

```
ELSE
```



## 使用控制对

```
PNT2=FEAT/POINT,RECT  
  
...  
  
...  
  
ENDMEAS/  
  
END_ELSE
```

### 样例代码的说明

#### C1=COMMENT/YESNO

此行取用并存储用户的 YES 或 NO 响应。

#### IF/C1.INPUT=="YES"

此行是表达式。它测试注释 1 的输入是否为 YES。若为 YES，则 IF 语句为 TRUE，并在 IF 语句后继续执行语句，在本例中，测量的是 PNT1 特征。若为 NO，则移至 END\_IF 语句。

#### END\_IF

此行结束执行 IF / END IF 代码块中的命令。若用户在此注释处单击 **No**，PC-DMIS 将转至此行后面的代码。

#### ELSE

若上述 IF / END IF 块评估为 False，则执行此行之后的 END\_ELSE 行之前的命令行。在本例中，执行了 PNT2。

#### END\_ELSE

此行结束执行 ELSE / END\_ELSE 代码块中的命令。

## While / End While

**插入 | 流控制命令 | 控制对 | While / End While** 菜单项允许您添加条件循环到测量例程。WHILE 和 END WHILE 命令之间的各项将一直循环执行，直至不再符合使循环处于活动状态的条件（或表达式），即 while 循环的表达式求值为假（零）。WHILE 命令可添加至测量例程中的任意位置。表达式将在每个循环的开始进行检验。

WHILE / END WHILE 语句在“编辑”窗口中的命令行为：

```
WHILE/expression
END_WHILE/
```

要插入 While / End While 选项，请执行以下步骤：

1. 将光标放在“编辑”窗口中的所需位置。
2. 从菜单栏中选择 **While / End While**。WHILE / END WHILE 语句将显示在“编辑”窗口中。

## While / End While 样例代码

参见以下示例，对特征测量测量例程用户指定的次数。

C1=注释/输入，您希望 PNT1 测量几次？请只输入一个整数。

```
ASSIGN/COUNT=0
WHILE/COUNT<C1.INPUT
    PNT2=FEAT/POINT,RECT
    ...
    ...
    ...
ENDMEAS/
ASSIGN/COUNT=COUNT+1
COMMENT/OPER,"Measured "+COUNT+" out of "+C1.INPUT+" times."
END_WHILE/
```

### 样例代码的说明

**C1=COMMENT/INPUT**

此行取用用户输入的整数并将其存储至变量 C1.INPUT。

**ASSIGN/COUNT=0**

使用控制对

此行初始化用户定义的变量 `COUNT`，并为其分配初始值 0。代码使用此变量计算 PC-DMIS 测量循环内特征的次数。

```
WHILE/COUNT<C1.INPUT
```

此行是表达式。测试 `COUNT` 的值（初始值设置为 0）是否小于用户所选的整数。如果测试值为 `True`，将执行 `WHILE/` 之后、`END_WHILE/` 之前的语句。

```
ASSIGN/COUNT=COUNT+1
```

此行以 1 为增量递增 `COUNT` 变量，从而在不满足条件测试时最终退出循环。

```
COMMENT/OPER,"Measured "+COUNT+" out of "+C1.INPUT+" times."
```

此行显示一条表明循环运行次数超出总数的消息。

```
END_WHILE
```

此行在条件为 `False` 时结束执行 `WHILE / END WHILE` 块中的命令。否则，在 PC-DMIS 遇到此命令时循环将返回 `WHILE` 语句。

## Do / Until

**插入 | 流控制命令 | 控制对 | Do / Until** 菜单项允许您添加条件循环到测量例程。DO 和 UNTIL 命令之间的项目将在循环中继续执行，直至 UNTIL 命令表达式评估为 `TRUE`（非 0）。DO/ UNTIL 命令可添加至测量例程中的任意位置。此表达式在每个循环结束时测试。

DO / UNTIL 语句在“编辑”窗口中的命令行为：

```
DO/  
UNTIL/ expression
```

要插入 DO / UNTIL 命令，请执行以下步骤：

1. 将光标放在“编辑”窗口中的所需位置。
2. 从菜单栏中选择 **Do / Until**。DO / UNTIL 语句将显示在“编辑”窗口中。

## Do / Until 样例代码

参见以下示例，对特征测量测量例程用户指定的次数。该代码与 **While / End While** 主题中提供的示例类似，只是 **PC-DMIS** 在循环的结尾（而不是开头）测试条件。

```
C1=      COMMENT/INPUT,Type the number of times PC-DMIS should measure the PNT1
feature:(type an integer only)

ASSIGN/COUNT=0

DO/

    PNT1=FEAT/POINT,RECT

    ...

    ...

    ENDMEAS/

    ASSIGN/COUNT=COUNT+1

    COMMENT/OPER,"Measured "+COUNT+" out of "+C1.INPUT+" times."

UNTIL/COUNT==C1.INPUT
```

### 样例代码的说明

#### **C1=COMMENT/INPUT**

此行取用用户输入的整数并将其存储至变量 **C1.INPUT**。

#### **ASSIGN/COUNT=0**

此行初始化用户定义的变量 **COUNT**，并为其分配初始值 **0**。代码使用此变量计算 **PC-DMIS** 测量循环内特征的次数。

#### **执行/**

开始 **DO / UNTIL** 循环。所有语句至少执行一次，表达式评估为 **False** 后测量程序流退出循环。

使用控制对

```
ASSIGN/COUNT=COUNT + 1
```

此行以 1 为增量递增 COUNT 变量，从而在不满足条件测试时最终退出循环。

```
COMMENT/OPER, "测量的"+ COUNT +"超出"+ C1.INPUT +"次数"。
```

此行显示一条表明循环运行次数超出总数的消息。

```
UNTIL/COUNT == C1.INPUT
```

此行在条件评估为 **False** 后结束执行 DO / UNTIL 循环内的命令。否则，在 PC-DMIS 遇到此命令时循环将返回 DO 语句。

## Select / End Select

插入 | 流程控制命令 | 控制对 | **Select / End Select** 菜单选项用于添加与 CASE / END CASE 对和 Default Case / End Default Case 对一起使用的条件块。Select 命令的表达式将提供与 Case 语句中的表达式进行比较的数据。如果两个表达式的值相同，将执行 Case / End Case 块中的语句。SELECT / END SELECT 块位于 CASE / END CASE 和 DEFAULT CASE / END DEFAULT CASE 块组之外。

SELECT / END SELECT 语句在“编辑”窗口中的命令行为：

```
SELECT/expression  
END_SELECT/
```

要插入 Select / End Select 命令，请执行以下步骤：

1. 将光标放在“编辑”窗口中的所需位置。
2. 从菜单栏中选择 **Select / End Select**。SELECT / END SELECT 语句将显示在“编辑”窗口中。

## Select / End Select 样例代码

SELECT / END\_SELECT、CASE / END\_CASE 和 DEFAULT CASE / END\_DEFAULT CASE 对配合使用，评估多种条件，提供广泛的替代方案。

假定您有五个圆，标号为 CIR1 到 CIR5，如果您希望操作者只需按键盘上的键即可测量某个圆。可以使用类似如下代码：

### 整个代码

```
DO/

C1=注释/输入, 键入一个数字测量该圆：

,对于 CIR1 - 键入 1
,对于 CIR2 - 键入 2
,对于 CIR3 - 键入 3
,对于 CIR4 - 键入 4
,对于 CIR5 - 键入 5
,任何其它字符将退出循环
SELECT/C1.INPUT
    CASE/1
        CIR1=FEAT/CIRCLE
        ...
        ...
        ENDMEAS/
    END_CASE
CASE/2
    CIR2=FEAT/CIRCLE
    ...
    ...
    ENDMEAS/
END_CASE
CASE/3
```

## 使用控制对

```
        CIR3=FEAT/CIRCLE
        ...
        ...
        ENDMEAS/
    END_CASE
    CASE/4
        CIR4=FEAT/CIRCLE
        ...
        ...
        ENDMEAS/
    END_CASE
    CASE/5
        CIR5=FEAT/CIRCLE
        ...
        ...
        ENDMEAS/
    END_CASE
    DEFAULT CASE
        注释/操作者,现在退出循环。
    END_DEFAULT CASE
END_SELECT
UNTIL C1.INPUT < 1 OR C1.INPUT > 5
```

### 样例代码的说明

#### **SELECT/C1.INPUT**

此代码行取用用户键入的数字或字符串值（本例中为数字），并确定将从输入执行哪一个 CASE/END\_CASE 块。请注意 SELECT / END\_SELECT 对包围了整个代码列表。所有 CASE / END\_CASE 和 DEFAULT CASE / END\_DEFAULT CASE 对必须在这两行之间。

#### **END\_SELECT**

标记 SELECT / END\_SELECT 对内的代码结束。

**CASE/1 至 CASE/5**

根据 `C1.INPUT` 值，将执行其中一个 **CASE** 代码块。例如，如果 `C1.INPUT` 值为 1，则代码的 **CASE 1** 块执行并测量 `CIR1`。如果其值为 2，则代码的 **CASE 2** 块执行并测量 `CIR2`，以此类推。

**END\_CASE**

这些行结束特定的 **Case** 代码块。

**DEFAULT CASE**

若 `C1.INPUT` 值不符合任何已定义 **CASE** 语句（如果该值不是 1 至 5 之间的数字），则执行 **DEFAULT CASE** 代码块。本例显示了一条消息，告知您退出循环。

请注意 **DO / UNTIL** 循环是如何包围住整个代码示例的。如此可让用户继续从 **COMMENT/INPUT** 行创建的菜单选择，直至该用户选择一个 **CASE** 语句未识别的字符。

## Case / End Case

**插入 | 流控制命令 | 控制对 | Case / End Case** 菜单项允许您添加条件块到测量例程。若 **Case** 语句的表达式评估的值等于相应 **SELECT** 命令的表达式，则执行 **CASE** 和 **END CASE** 命令之间的项目。否则将跳过语句块。**CASE / END CASE** 语句块必须直接位于 **SELECT** 命令或前一个 **CASE / END CASE** 块的 **END CASE** 命令之后。此外，**PC-DMIS** 不能在单个 **Case** 语句上比较多个表达式。

**CASE / END CASE** 语句在“编辑”窗口中的命令行为：

```
CASE/expression
END_CASE/
```

要插入 **Case / End Case** 选项，请执行以下步骤：

1. 将光标放在“编辑”窗口中的所需位置。请注意以上所述的位置要求。
2. 从菜单栏中选择 **Case / End Case**。**CASE / End CASE** 语句将显示在“编辑”窗口中。



## Default Case / End Default Case

插入 | 流控制命令 | 控制对 | **Default Case / End Default Case** 菜单项允许您添加条件块到测量例程。若相应 **SELECT / END SELECT** 块中前一 **CASE / END CASE** 块中的其他所有表达式均评估为 **False**，则执行 **DEFAULT CASE** 和 **END DEFAULT CASE** 命令之间的项目。**SELECT/ END SELECT** 块中只允许有一个 **DEFAULT CASE / END DEFAULT CASE** 块。**DEFAULT CASE / END DEFAULT CASE** 块必须位于 **SELECT / END SELECT** 块中所有 **CASE / END CASE** 块后面。

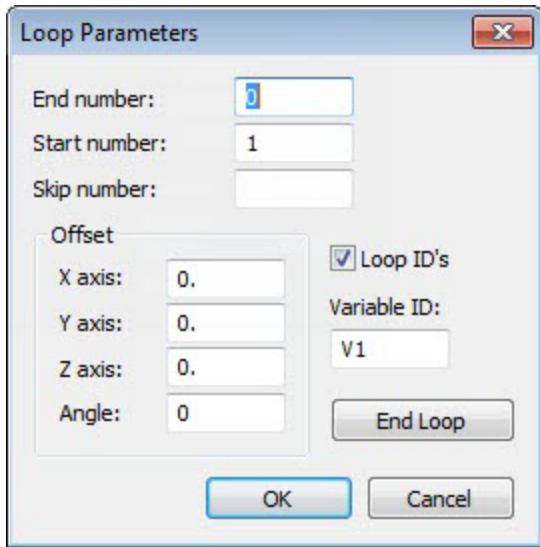
**DEFAULT CASE / END DEFAULT CASE** 语句在“编辑”窗口中的命令行为：

```
DEFAULT CASE/  
END_DEFAULT_CASE/
```

要插入 **DEFAULT CASE/ END DEFAULT CASE** 命令，请执行以下步骤：

1. 将光标放在“编辑”窗口中的所需位置（请注意以上所述的位置限制）。
2. 从菜单栏中选择 **Default Case / End Default Case**。**DEFAULT CASE / End DEFAULT CASE** 语句将显示在“编辑”窗口中。

## 使用一般循环



循环参数对话框

**插入 | 流控制命令 | 循环**菜单显示**循环参数**对话框。可使用该对话框在有偏置或无偏置的情况下创建重复测量例程的 **LOOP** 命令。**LOOP** 命令可添加到测量例程中的任意位置，尽管此功能在测量例程开头或结束处最有用。

**起始编号** - 该框告知 PC-DMIS 零件序列中的开始位置编号。



假设有 10 个零件，您想从位置编号 5 开始。您输入 10 作为零件总数，5 表示起始位置。

**终止编号** - 该框告知 PC-DMIS 测量例程中的循环次数。此数字通常与 X、Y 或 Z 方向夹具夹持的零件（或零件上的样式）数相同。PC-DMIS 也要求输入起始零件（样式）号。

## 使用一般循环



假设 X 方向有 10 个零件，您想从位置编号 5 开始。对于**终止编号**框，输入 10。对于**起始编号**框，输入 5。

**跳过编号** - 在循环中 PC-DMIS 重复指定次数的测量例程。**跳过编号**框使您可以跳过指定的增量。注意，使用此框，您无法**跳过**第一个增量，但您可以将**起始编号**更改为 2，从而通过这种方法有效地跳过。



假设您可以将此参数设置为跳过循环的所有第三个增量。若指示跳过数字 3，则 PC-DMIS 将测量第一个和第二个零件，然后跳至第四个零件。

**偏置** - 偏置区域包含下列框：

**X 轴、Y 轴和 Z 轴** - 这些框定义零件或同一零件上的图案之间的 X、Y 和 Z 偏移距离。循环执行后，这些值将零件偏移定义的偏移距离。

**角度** - 该框设置零件之间、或相同零件上样式之间的角度偏置。第一次偏置从零件原点开始。执行循环之后，PC-DMIS 就将零件偏移角度值。

软件将偏移距离基于零件的原点。第一次完成循环，PC-DMIS 使用零件的原点。第二次完成循环，它移动偏移距离。第三次完成循环，它移动偏移距离乘以 2，依此类推。



如果循环中有坐标系命令，并且循环使用偏移，则**必须**定义该坐标系的所有轴。此外，循环内部的坐标系必须使用在循环内部测量的特征。

**循环 ID** - 此复选框确定 PC-DMIS 是否在特征 ID 中的方括号中显示当前循环值。



假设您具有 CIR1 的圆形特征。它在第一次完成循环时变为 CIR1[1]，在第二次完成循环时变为 CIR1[2]，依此类推。

### 统计数据库中的特征 ID

若选择**循环 ID** 复选框，并发送统计数据至数据库，在某些情况下，PC-DMIS 将不在数据库中显示这些循环 ID。

参见以下步骤：

如果在**循环块内部**有 STATS/ON 命令和 STATS/UPDATE 命令，则循环 ID 不显示在数据库内部。

如果在**循环块外部**有 STATS/ON 命令并在**循环块内部**有 STATS/UPDATE 命令，则循环 ID 不显示在数据库内部。

如果在**循环块内部**有 STATS/ON 命令并在**循环块外部**有 STATS/UPDATE 命令，则循环 ID 显示在数据库内部。

如果在**循环块内部**有 STATS/ON 命令并在**循环块外部**有 STATS/UPDATE 命令，则循环 ID 显示在数据库内部。

**变量 ID** - 该框允许您定义跟踪循环当前迭代（或指定循环次数中的当前循环）使用的变量名称。在测量例程执行过程中，此变量将等于循环的当前迭代次数。

**结束循环** - 该按钮完成循环过程。在“编辑”窗口中，必须在命令 LOOP/START 之后加上命令 LOOP/END。

## 循环的用途

循环功能主要有三项用途：

## 使用一般循环

- 您具有一个多零件的夹具，用以夹持一个零件网格。夹具应在各行之间使用一致的间距。利用平移/旋转偏置，您可以从零件网格中的一个零件转换到下一个零件。
- 您具有夹持一个零件的夹具，并且想在每个测量例程循环之前换入新的零件。将此零件取代为新零件时，可使用 `COMMENT` 命令停止 `CMM`。此命令可位于循环开始或结束处。
- 您想使用循环功能来旋转测量例程，以测量同一零件的不同部分。例如，您可能会创建一个测量例程来测量一个复杂的孔模式，该模式在零件上重复了十次。您的测量例程只需要测量其中一个孔模式。然后，您可以使用循环功能来偏移测量例程，以测量模式的其他九次出现。



若在循环中使用坐标系，`PC-DMIS` 允许您在 `ALIGNMENT/START` 命令行中使用活动坐标系，而不是始终调用先前已储存的坐标系。请参见“创建和使用坐标系”一章中的“在循环中使用坐标系”主题。

## 创建循环

1. 从菜单上选择 **插入 | 流程控制命令 | 循环** 以显示 **循环参数** 对话框。
2. 在对话框中定义值。
3. 根据需要选择参数（例如“**零件数**”、“**起始号**”、“**跳过号**”和“**偏置角度**”）。
4. 将光标放在编辑窗口中要开始循环的位置，
5. 单击 **确定** 按钮以插入 `LOOP/START` 命令块。

这个选项在编辑窗口中的命令为：

```
VARNAME = LOOP/START, ID = Y/N, NUMBER = 0, START = 1, SKIP  
= ,  
OFFSET: XAXIS = 0, YAXIS = 0, ZAXIS = 0, ANGLE = 0
```



要完成循环程序，必须以 `LOOP/END` 命令结束循环。PC-DMIS 循环 `LOOP/START` 和 `LOOP/END` 命令包含的任何编辑窗口命令。

您可以通过以下方式之一插入此命令：

- 在“编辑”窗口中，键入 `LOOP/END`。
- 选择 **插入 | 流控制命令 | 结束循环** 菜单项。
- 在 **循环参数** 对话框中，单击 **结束循环** 按钮。

## 循环使用打印命令

要在 PC-DMIS 中循环使用 `PRINT` 命令，请执行以下操作：

1. 在测量例程中插入 `LOOP/START` 和 `LOOP/END` 命令。
2. 在所需的循环命令之间插入 `PRINT` 命令。
3. 在“编辑”窗口中，单击 `PRINT` 命令。然后按 **F9** 键以显示该 `PRINT` 命令的**输出配置**对话框。有关如何为“报告”窗口设置输出和打印机选项的详细信息，请参见“使用基本文件选项”一章中的“为报告窗口设置输出和打印机选项”主题。
4. 选择输出格式选项卡之一（**报告**、**DMIS** 或 **Excel**）以定义 `PRINT` 命令的格式。
5. 选中选项卡页面左上角的复选框，以启用输出选项。
6. 定义将生成的报告发送到的位置：
  - a. 单击路径和文件名框右侧的浏览 ( ... ) 按钮以显示**另存为**对话框。
  - b. 在**另存为**对话框中，定义文件名，然后单击**保存**。
7. 进行所需的任何其他更改。例如，如果要在 PC-DMIS 生成每个报告后查看它们，请选中**显示报告**复选框。
8. 如果要生成多种格式的输出，请按下一个选项卡并重复步骤 4 至 7。
9. 单击**确定**按钮以保存对 `PRINT` 命令的更改。

## 使用一般循环

`PRINT` 命令在循环内部的工作与在循环外部的工作几乎相同。区别在于，**PC-DMIS** 在内部跟踪输出的内容，仅输出在执行最后一个 `PRINT` 命令之后执行的数据。



例如，考虑下面的代码片段。当 **PC-DMIS** 第一次执行 `PRINT` 命令时，软件将创建第一个输出文件，包含*所有*收集到的数据。这意味着第一个输出文件还包括在 `LOOP/START` 命令之前收集的数据。下次执行 `PRINT` 命令时，**PC-DMIS** 仅执行循环中的内容。这意味着第二个数据文件仅包含循环中的数据。重复进行直到测量例程结束。

·  
·  
·

### 循环开始

```
CIR1 = CIRCLE (CONTACT)
```

```
CIR2 = CIRCLE (CONTACT)
```

移动点

移动点

移动点

移动点

```
CIR3 = CIRCLE (CONTACT)
```

移动点

移动点

移动点

```
CIR4 = CIRCLE (CONTACT)
```

```
CIR5 = CIRCLE (CONTACT)
```

```
CIR6 = CIRCLE (CONTACT)
```

```
CIR7 = CIRCLE (CONTACT)
```

```
CIR8 = CIRCLE (CONTACT)
```

```
通过的 LOC1 : CIR1
```

```
打印报告
```

循环终止

·  
·  
·

## 使用标签

您可以使用带 `GOTO` 的 [标签](#) 命令或 `IF_GOTO` 命令根据特定条件控制执行流发送的位置。标签名称不能有空格且不得超过 230 个字符。PC-DMIS 以全大写字母显示标签名称。

要创建 [标签](#) 命令，请执行以下操作之一：

键入命令。

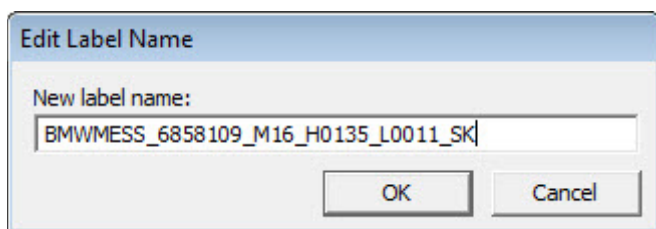
1. 将光标放在“编辑”窗口的空白行中。
2. 键入 `LABEL`，然后按 `Tab` 键。
3. 在突出显示的字段中，键入新的标签名称。

或者



### 选择命令

1. 将光标放在“编辑”窗口中。
2. 从菜单栏中选择 **插入 | 流控制命令 | 标签**，访问 **编辑标签名称** 对话框。
3. 在 **新建标签名称** 框中键入标签名称。



编辑标号名对话框

4. 单击 **确定**，将标签名称插入“编辑”窗口的下一个位置。

**LABEL** 命令在“编辑”窗口中的命令模式为：

`ID = LABEL/`

其中，**ID** 表示指定的标签名称。

有关使用带标签的 **GOTO** 命令信息，请参见“使用 **GOTO** 跳至标签”主题。

有关使用带标签的 **IF\_GOTO** 命令的信息，请参见“根据条件跳至标签”主题。

## 使用 **GOTO** 跳至标签

通过创建 **GOTO** 命令，您可以让测量例程跳至特定标签。

要创建 **GOTO** 命令，请执行以下操作之一：

### 键入命令。

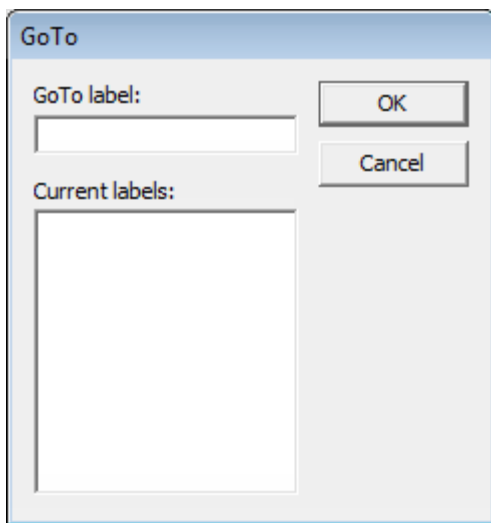
1. 将光标放在“编辑”窗口的空白行中。

2. 键入 GOTO，然后按 Tab 键。
3. 键入要跳转到的标签名称。

或者

### 选择命令

1. 将光标放在“编辑”窗口中。
2. 选择菜单选项**插入 | 流量控制命令 | 跳至**打开**跳至**对话框。



跳至对话框

3. 如果测量例程中已存在这些标签，它们将显示在**当前标签**框中。
4. 从**当前标签**框中选择所需的标签，或者在**跳至标签**框中键入标签名称。
5. 单击**确定**，将 GOTO 命令插入“编辑”窗口的下一个位置。

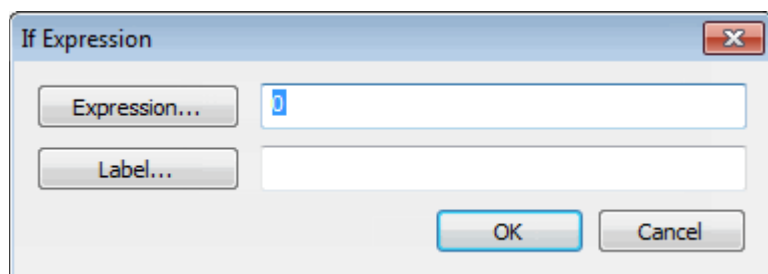
GOTO 命令在“编辑”窗口中的命令模式为：

GOTO/label\_ID

其中，label\_ID 表示指定标签的名称。标签名称不能有空格且不得超过 230 个字符。



若尚未创建选项卡，则标签名称会在“编辑”窗口命令模式中以红色突出显示，该 GOTO 命令将在执行过程中被忽略。



*If 表达式对话框*

通过插入 | 流控制命令 | If Goto 选项，可使用 **If Expression** 对话框在测量例程中创建 IF GOTO 语句。执行测量例程且 PC-DMIS 遇到 IF GOTO 语句时，若指定的表达式计算为非 0 值，则例程流将移至标签标识。

IF\_GOTO 语句在“编辑”窗口中的命令行为：

```
IF_GOTO/expression, GOTO=Label
```

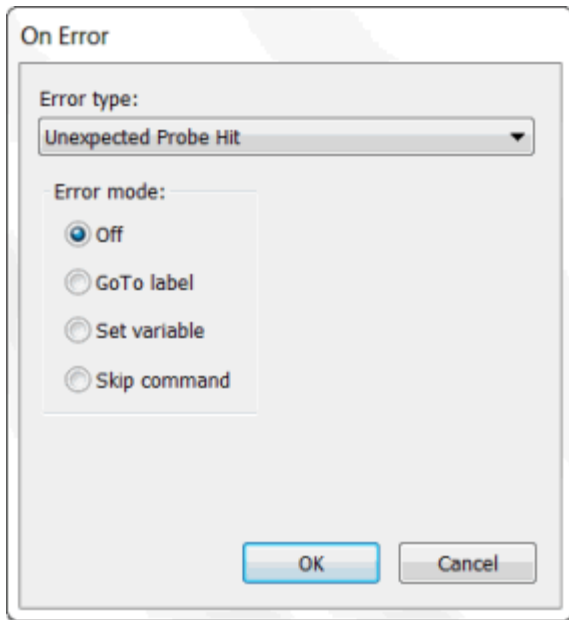
**Expression** - 如果单击了 **Expression**，表达式建立器将打开。利用表达式构造器，您可以创建多种需要在测量例程中使用的不同表达式。一旦创建了表达式，其将显示在 **Expression** 对话框中。有关表达式创建的信息，请参见“使用表达式和变量”一章。

**Label** - 如果单击了 **Label**，**Goto** 对话框将打开。可从 **Goto** 对话框中选择当遇到定义表达式时，您希望流移动至哪个标签位置。选定的标签将出现在 **Label** 对话框中。或者如果您知道标签名称，也可在对话框中输入名称。有关标签的更多信息，请参见“跳至该标签”。

---

## 出错时分支

使用插入 | 流控制命令 | 出错选项打开 **On Error** 对话框：



出错对话框

当测量机出现错误时，可以通过该对话框命令 **PC-DMIS** 采取一定操作。

**错误类型 - PC-DMIS** 追踪这些出错情况：

- 意外测头测点
- 未命中的测头测点
- 找不到目标 - 其用于 **PC-DMIS** 关节臂追踪系统。
- 激光错误 - 如果您使用激光设定，请参阅 **PC-DMIS** 激光文档中的“使用出错命令处理激光传感器错误”主题。
- 温度超出阈值 - 如果 X 轴、Y 轴或 Z 轴范围或零件的一个或多个温度高于或低于温度补偿命令定义的最高或最低限制，则测量例程中的温度补偿命令会产生此错误。



出错命令必须放置在测量例程的温度补偿命令之上。

- **校准超出限制** - 如果一个或多个校准错误检查超出限制，则测量例程中的检查校准限制命令会引发此错误。有关检查校准限制命令的更多信息，请参阅 **PC-DMIS** 工具包模块文档中的「关于检查校准限制」。



出错命令必须放置在测量例程的检查校准限制命令上。

**错误模式** - PC-DMIS 可以根据错误类型采取这些可能的操作：

- **关**: PC-DMIS 什么也不做。
- **GoTo 标签** - 测量例程流移动至定义的标签（参见“使用标签”）。可用的选项：
  - **标签 ID** - 向标签中键入尚不存在的参考。
  - **当前标签** - 列车测量例程中的所有标签。
- **设置变量** - 将一个变量的值设为 1。
- **跳过命令** - 测量例程流跳过当前命令，移动至测量例程中的下一条标记命令。



有关 **PC-DMIS** 如何报告在执行期间使用跳过特征的尺寸的详细信息，请参见文档中的“报告使用执行期间跳过特征的尺寸”章节。

默认所有测量例程以将两种错误类型设置为**关**（**PC-DMIS** 不执行任何操作）作为开始操作。在整个测量例程中，可以更改每种错误类型的操作模式。



若在执行过程中，PC-DMIS 遇到 `ONERROR/UNEXPECTED HIT/JUMPTOLABEL` 命令，则测量例程中该点之后的任何未预期测点将使得执行移转至指定标签。操作“变量值设为 1”将促使在指定错误类型发生之时设置变量。然后可使用 IF 语句测试此变量值，以使执行移转至测量例程中的新点处。

### 若要使用“出错”命令

1. 选择**插入 | 流控制命令 | 出错命令**打开**出错**对话框。
2. 从**错误类型**列表中选择符合分支条件的错误类型。
3. 在**错误模式**区域中，选择定义满足条件时要执行的操作的选项。
4. 若要使用出错命令，单击 **OK**。关闭对话框，而不应用任何更改，单击**取消**。

## 出错支持界面

有些接口支持 On Error 命令。要查看您的接口是否支持该命令，请参阅下表。

- 如果您的接口位于表格中，则小黑框会指示接口支持的错误类型。
- 如果你的接口不在表格中，那么它不能使用 On Error 命令。

支持界面	意外测头触测	未命中测头触测	扫描未发现
DEA	-	■	-
FDC	■	■	-
I++ DME 客户端	■	■	-
Johansson	■	■	-
Leica Tracker	-	-	■

Leitz	■	■	-
LK Direct ( 又称 LKRS232 )	■	■	-
LK 驱动程序	■	■	-
Mora	■	■	-
Sharpe	■	■	-
Sheffield	■	■	-
Wenzel	■	■	-
Zeiss	■	■	-

## 子例程分支

子例程选项用于从当前文件（或外部文件）中访问测量例程命令，并重复使用这些命令。PC-DMIS 允许您将信息传递给子例程中的“自变量”（或局部变量）。可传入子例程的自变量类型有数值、变量、文本字符串和特征名称。



子例程命令块包含在 **SUBROUTINE** 和 **ENDSUB** 命令中。

在测量例程中创建子例程后，即可从当前测量例程或其他测量例程“调用”此子例程，从而使得测量例程执行流转入指定的子例程，执行子例程命令区块中包含的命令。然后测量例程流将在调用语句后立即返回此语句。



使用 **CALLSUB** 命令调用子例程。

## 外部子程序

外部子例程或正被调用的测量例程以外测量例程中的子例程，不访问所调用的程序的特征、变量或坐标系。子例程仍将访问其测量例程之中的项目。外部测量例程和调用的测量例程必须使用相同的测量单位。

## 嵌套子程序

你可以在一个子程序中嵌套另一个子程序。唯一的限制条件就是嵌套子程序数量太多会导致内存不够。

## 新建子例程

### 通过键入 **SUBROUTINE** 创建子例程

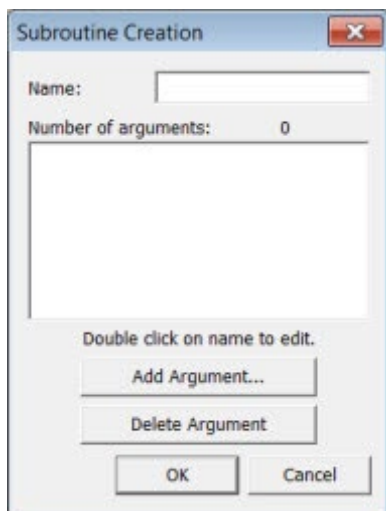
你可以在“编辑”窗口里命令模式下输入命令 **SUBROUTINE** 然后按 **TAB**。插入命令后，需指定子例程的名称以及所含的全部自变量。相关信息，请参阅子例程语法和下例。

输入 **ENDSUB** 命令然后点击 **TAB** 键来终止命令块。在编辑窗口里输入该命令块内部输入的任何命令将会被做为该子程序的一部分而且当子程序被调用时将会被执行。

### 通过使用子例程菜单项创建子例程

1. 从子菜单中选择**插入 | 流控制命令 | 子例程**。屏幕上出现**创建子例程**对话框。有关此对话框的信息，请参见“了解创建子例程对话框”主题。





子例程创建对话框

2. 在**名称**框中输入子例程名称。该名称最多为 180 个字符。如果输入的字符为 181 个或超过 181 个，则会出现“命令行太长”错误消息。

单击**确定**按钮后，该名称会自动缩短为前 180 个字符，同时会关闭**创建子例程**对话框。

3. 如果你的子例程使用了自变量（将信息所有者传递到子例程），单击**添加自变量**按钮来逐个添加自变量。屏幕上显示**自变量编辑**对话框。有关该对话框的信息，请参阅“了解自变量编辑对话框”主题。



自变量编辑对话框

4. 在**名称**框中键入自变量的名称。

5. 在**值**框中键入自变量的默认值。若没有值传递到 `CALLSUB` 语句中的子例程，则子例程将使用默认值。有效的自变量值可以是数值、变量、文本字符串和特征名称。
6. 若要描述自变量，请在**描述**框中键入信息。
7. 在**自变量编辑**对话框中单击 **确定**来创建自变量。
8. 对要在子例程中出现的每个自变量重复步骤 3 至 7。
9. 单击**创建子例程**对话框中的**确定**按钮来结束子例程的创建。子例程将会出现在“编辑”窗口里包含全部定义过的自变量。
10. 选择**插入 | 流控制命令 | 结束子例程**菜单项可结束子例程。此操作可在“编辑”窗口中放入 `ENDSUB/` 命令，完成子例程的命令块。任何要在子例程中添加的其他测量例程命令必须添加到子例程命令块内，`ENDSUB` 命令之前。

### 子例程命令块的语法

在编辑窗口命令行的一个样例子程序块语句如下：

```
SUBROUTINE/<名称>,
<A1> = <Arg1> : <描述>,
<A2> = <Arg2> : <描述>,
=
<命令>
ENDSUB/
```

**SUBROUTINE/** 是用于开始子例程命令块的命令。

**<Name>** = 子例程的名称。该名称最多为 256 个字符。如果输入的字符超过 257，则该名称会自动缩短为 256 个字符。

**<A1>** = 在子例程中使用的第一个自变量（或本地变量）。该变量通常不能在子例程外访问。

**<A2>** = 在子例程中使用的第二个自变量。该变量通常不能在子例程外访问。其他的自变量可以根据需要进行添加。

子例程分支

**<Arg1>** = 第一个自变量的默认值。

**<Arg2>** = 第二个自变量的默认值。

**<Description>** = 自变量的描述。

**<Commands>** = 其他需要在“编辑”窗口命令插入的自变量命令应该在自变量之后而且在 ENDSUB 之前。

**ENDSUB/** 是用来结束子例程命令块的命令。

### 子例程命令块示例

例如，一个结束的子程序包含执行数据而且显示在报告窗口里将会如下显示：

子程序/得到\_操作者\_信息

```
OPNAME = <Operator> : OPERATOR NAME,
```

```
SHIFT = <Shift> : SHIFT TIME,
```

=

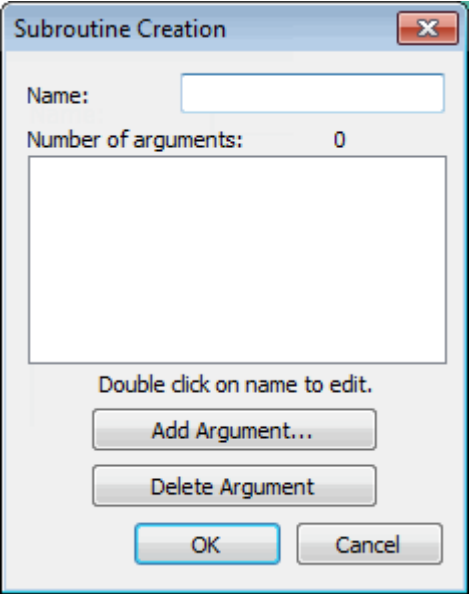
命令/回调，操作者名称

命令/回调，转换

终止子例程/

## 了解子程序创建对话框

若要访问子例程创建对话框，选择插入 | 流控制命令 | 子例程。



子例程创建对话框

下列选项在子例程创建对话框中可用。

**名称** - 该方框定义子例程。这是稍后调用子例程时要使用的名称。如果一个测量例程中有多个子例程，则每个名称必须是唯一的。

该名称最多为 180 个字符。如果在**名称**框中输入的字符为 181 个或超过 181 个，则会出现“命令行太长”错误消息。

单击**确定**按钮后，该名称会自动缩短为前 180 个字符，同时会关闭子例程创建对话框。

**自变量数** - 显示所创建的子例程自变量的列表。此区域中的自变量将以此种形式出现：

<NAME> = <VALUE> : <DESCRIPTION>

例如，如果您的自变量之一被命名为 "Diameter"，默认值为 3，则您在此列表中的自变量可能显示为：

## 子例程分支

直径 = 3 : 孔的直径

无论其它值是否源自 `CALLSUB` 命令，PC-DMIS 都将使用这个默认值。

要编辑自变量，双击要更改的自变量。屏幕上将打开**自变量编辑**对话框，允许您进行更改。有关此对话框的信息，请参见“了解自变量编辑对话框”。

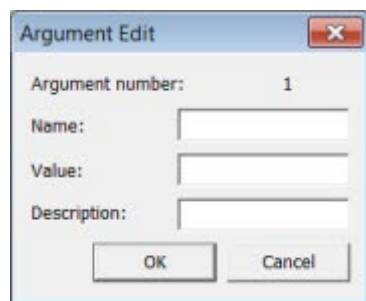
**添加自变量** - 该按钮可向子例程中添加新自变量。单击该按钮会显示**自变量编辑**对话框。有关该对话框的信息，请参阅“了解自变量编辑对话框”。

**删除自变量** - 该按钮可用于删除子例程中的自变量。从列表中选择自变量，然后单击**删除自变量**按钮进行删除。

## 编辑现有子例程

要编辑现有子例程，始终可使用“编辑”窗口的命令模式来直接编辑子例程。或者，也可将光标置于“编辑”窗口中的子例程之上以访问**创建子例程**，并按 **F9**。屏幕上显示**创建子例程**对话框。如需此对话框的信息，请参见“了解创建子例程对话框”主题。

## 了解“自变量编辑”对话框



自变量编辑对话框

只要在**创建子例程**对话框（**插入 | 流控制命令 | 子例程**）或**调用子例程**（**插入 | 流控制命令 | 调用子例程**）对话框中创建或编辑自变量，则将显示**自变量编辑**对话框。

**自变量编辑**对话框可用于以下两种情况：

- 在子例程命令块定义一个子例程自变量和它们的默认值。
- 从 `CALLSUB` 命令中定义这个值放入子程序中。

**自变量编辑**对话框中的可用选项如下：

**名称** - 该框用于输入所创建或编辑的自变量的名称。

**值** - 该框用于定义自变量值。

如果创建或编辑子例程命令，假设在 `CALLSUB` 语句中没有声明具体值载入到子例程中，就使用默认值。

如果创建或编辑 `CALLSUB` 命令，载入一个值到子程序中。

有效值

- 数字
- 变量
- 字符串 - 字符串必须使用双引号围绕。
- 特征名 - 特征名必须使用圆弧括，例如{F1}。

**说明** = 子例程中自变量的说明。此说明将显示在“编辑”窗口子例程命令块中的自变量旁边。

## 调用子例程

要调用子例程，您需要在测量例程中插入 `CALLSUB` 命令，以从当前测量例程调用现有子例程或从外部测量例程调用子例程。



当执行 `CALLSUB` 命令时，PC-DMIS 将执行子例程中的所有命令。这甚至包括未标记的命令。

被调用的子程序必须是在有效的 PC-DMIS 版本（版本 2018 R2 到 2024.1）中创建的。

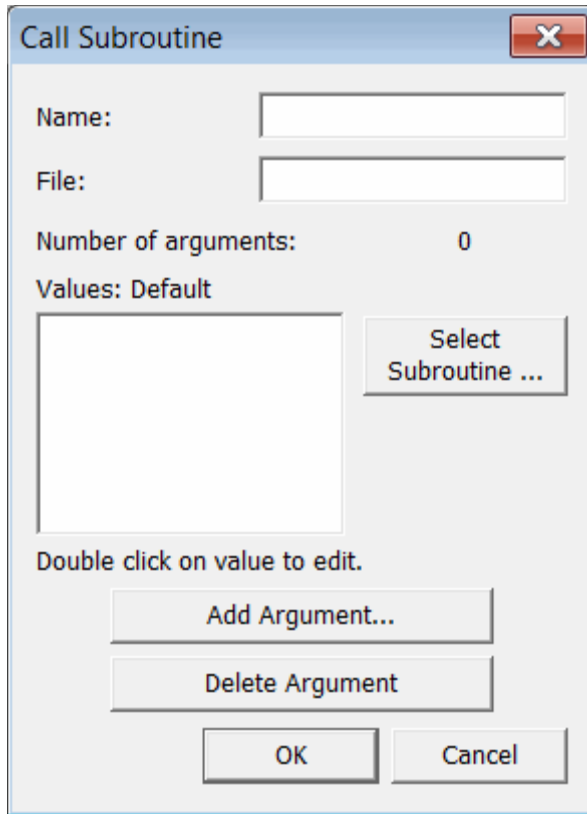
### 键入 `CALLSUB` 调用子例程

您可以在“编辑”窗口中输入 `CALLSUB`，然后按下键盘上的 **Tab** 键，以在您希望命令在“编辑”窗口中显示的位置插入该命令。

插入命令后，需要指定子例程的名称（如果它是外部测量例程）以及其他任何可用的自变量值。有关传递自变量的示例，请参见“向子例程中传递自变量”。

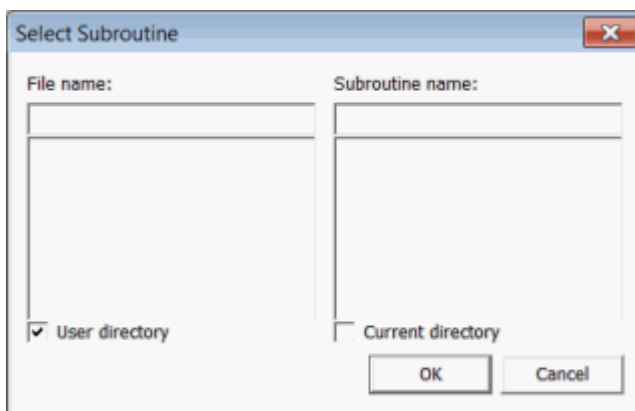
### 使用调用子例程菜单项调用子例程

1. 从子菜单中选择 **插入 | 流控制命令 | 调用子例程** 选项。`调用子例程` 对话框打开。参考“理解调用子例程对话框”得到此对话框更多的信息。



调用子程序对话框

2. 点击**选择子程序**按钮。选择子例程 对话框打开。



选择子例程序对话框



3. 选择**用户目录**复选框或者**当前目录**复选框或者两者。如果子例程来自的测量例程位于的目录是指定用于搜索子例程的目录，选择**用户目录**复选框。如果来自当前目录，则选择**当前目录**复选框。PC-DMIS 将列出所有测量例程供您选择。
4. 选择所需子例程所在的测量例程。您将看到与所选测量例程相关联的所有子例程都显示在**子例程名称**框中。
5. 选择你想要调用的子程序
6. 单击**确定**按钮。你想要调用的子例程信息将会在**调用子例程**对话框的**名称**和**文件**框中出现。
7. 若要向子例程中传递信息，请单击**添加自变量...**按钮，然后使用此**自变量编辑**对话框定义自变量和传递值。有关此对话框的信息，请参见“了解自变量编辑对话框”。有关传递自变量的示例，请参见“向子例程中传递自变量”。
8. 再单击**确定**按钮。PC-DMIS 在编辑窗口的选定位置添加 **CALLSUB** 命令。

### **CALLSUB 命令的语法**

编辑窗口命令行里调用子程序的程序语法实例如下：

```
CS1          =CALLSUB/<Name>, <File>:<Arg1>,<Arg2>,
```

**CS1** = 赋予 **CALLSUB** 命令的标签 ID。

**<Name>** = 调用的子例程的名称。

**<文件>** = 包括了调用子程序路径里的全部测量例程。如果此字段为空白，PC-DMIS 将在当前测量例程中查找子例程。

**<Arg1>** = 子例程里传递的第一个自变量值。如果此字段为空白，PC-DMIS 将改用为子例程中的第一个自变量定义的默认值。

**<Arg2>** = 子例程中传递的第二个自变量值。如果此字段为空白，PC-DMIS 将改用为子例程中的第二个自变量定义的默认值。这个语法例子只显示两个自变量。如果需要，您可以将其他自变量传递至您的子例程。



在“**CALLSUB**”命令中，应保留一组指针，指向为子例程创建的所有对象，以便于以后使用子例程的标识引用。有关指针的更多信息，请参见“使用表达式和变量”一章中的“指针”。

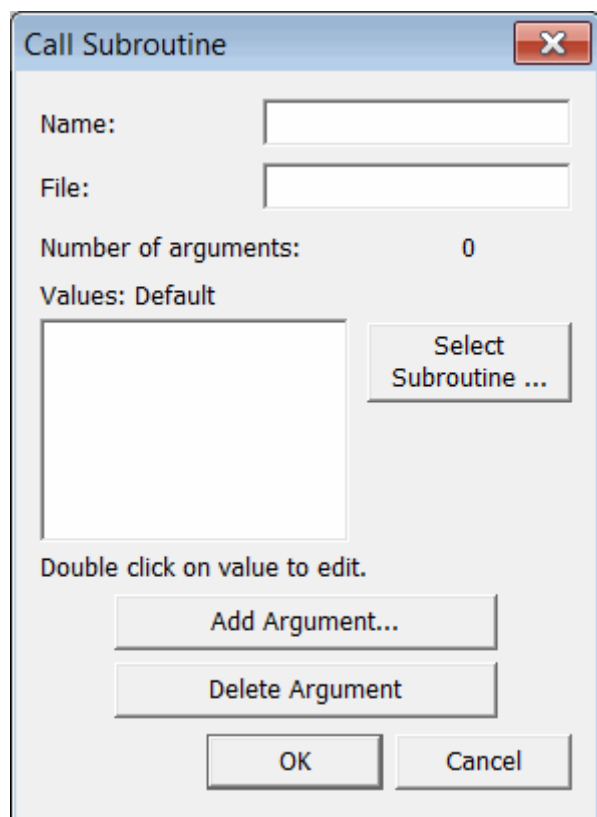
### **CALLSUB 命令示例**

```
CS1  
=CALLSUB/GET_OPERATOR_INFO,D:\MEASROUTINES\V42SUBROUTINETEST.PRG  
:V1,V2,,
```

此示例 **CALLSUB** 命令 **CS1** 调用名为 **GET\_OPERATOR\_INFO** 的子例程，该子例程位于 **D:\MEASROUTINES\** 目录中的测量例程 **V42SUBROUTINETEXT.PRG** 内。

它向子程序传递两个变量参数 **V1** 和 **V2**。

## 了解“调用子程序”对话框



调用子程序对话框

下文说明了调用子例程对话框中可用的各个选项。

### 名称框



名称框包含使用选择子例程...按钮选择后的子例程名称。

该名称最多为 180 个字符。如果在名称框中输入的字符为 181 个或超过 181 个，则会出现“命令行太长”错误消息。

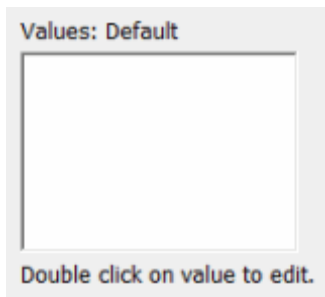
单击确定按钮后，该名称会自动缩短为前 180 个字符，同时会关闭调用子例程对话框。

### 文件框



文件框包含已调用的子例程文件的目录路径。

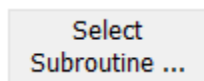
### 值框



值框包含与该子例程相关的每个自变量的值的列表。执行此子例程时，这些值将传入子例程中。

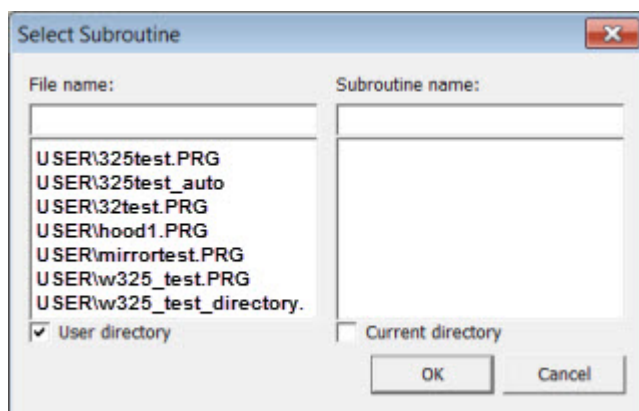
要更改这些值，双击要更改的值。屏幕上显示**自变量编辑**对话框。有关此对话框的信息，请参见“了解自变量编辑对话框”。

### 选择子例程按钮



按**选择子例程**按钮可显示**选择子例程**对话框。

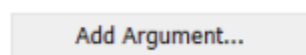
## 子例程分支



选择子例程序对话框

此对话框允许您通过搜索用户目录或当前目录，调用之前创建的子例程。有关**选择子例程**对话框信息，请参见“调用子例程”。

### 添加自变量按钮



这个按钮在子程序中为自变量定义一个值。

### 删除自变量按钮



此按钮可用于删除**值**框中的自变量。选择所显示的值，然后单击**删除自变量**按钮。与该值相关的自变量将被删除。

## 使用“调用子程序”对话框添加新的自变量：

使用 `CALLSUB` 命令操作子程序添加新的自变量：

1. 在**调用子例程**对话框中，单击**添加自变量**按钮以打开**自变量编辑**对话框。
2. 单击**值**框。
3. 在**值**框中输入自变量值。
4. 单击**确定**。

## 使用调用子例程框编辑现有自变量

在 `CALLSUB` 命令中编辑现有的自变量：

1. 在**调用子例程**对话框中，双击要更改的自变量的值。此时将显示一个值框，其中包含被调用子例程的默认值。
2. 键入新值。
3. 单击**确定**按钮。

有关如何编辑或创建子例程新自变量的详细信息，请参见“新建子程序”和“编辑现有子例程”。

## 从调用子例程命令中删除自变量：

1. 将光标放在 `CALLSUB` 命令上。
2. 按 **F9** 访问**调用子例程**对话框。
3. 从自变量列表中选择一个或多个自变量。
4. 单击**删除自变量**按钮。
5. 单击**确定**。

您也可以从“编辑”窗口文本中直接删除自变量。要执行此操作，

1. 使 **PC-DMIS** 处于命令模式。
2. 将光标置于 `CALLSUB` 命令上，然后按 **TAB**，直到突出显示所需的自变量。
3. 输入字母 **"del"** 以删除自变量。应注意按 **Delete** 或 **Backspace** 键并不能真正删除自变量，只是将自变量更改为空自变量。

## 在多臂模式使用 CALLSUB 语句

若将 CALLSUB 语句分配给臂 1，调用子例程时该子例程中的所有命令均被分配给臂 1。

若将 CALLSUB 语句分配给臂 2，调用子例程时该子例程中的所有命令均被分配给臂 2。

若将 CALLSUB 语句标记为双臂，在调用此子例程时，子例程中标记将保留其初始设置。

若子例程中包含 MOVE/SYNC 命令，且将 CALLSUB 语句分配给臂 1 或臂 2，执行时 PC-DMIS 将显示错误，指出无效且没有调用子例程。

有关为特定臂指派执行命令的信息，请参阅“使用多臂模式”一章中的“为臂指派命令”主题。

## 子例程示例

参考以下话题中的信息的传递自变量的实例以及总体的子程序。

- 将自变量传递给子例程
- 子例程样例代码

## 将自变量传递给子程序

您可传入子例程的自变量类型有数值、变量、文本字符串和特征名称。若要将值传递至自变量中，在自变量编辑对话框的**值**框中键入一个值，或在命令模式下直接键入插入的 CALLSUB 命令。

有关该操作方法的详细信息，请参见“创建新子例程”以及“了解自变量编辑对话框”。

### 将变量传递给子例程

能够传递回数据的自变量是变量。如果将变量用作子例程的自变量，对相应变量进行的任何更改都将传回，成为已传入的变量的值。



### 传递变量的示例：

该实例叙述了一个变量值是如何作为一个自变量在子程序中被使用，修改以及从子程序中传递回来的：

```

CS4      ASSIGN/V1=6
        =CALLSUB/MYSUB, :V1,,
        SUBROUTINE/MYSUB,
            A1 = 0 : FIRST ARGUMENT,
            =
        ASSIGN/A1=A1+1
        COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-CONTINUE=NO,OVC=NO,
        The value of A1 is:
        A1
        ENDSUB/

```

### 带执行流程的代码说明

首先，V1 变量被赋值为 6。

然后，CALLSUB 语句调用子例程并将 V1 变量作为第一个自变量传递。

SUBROUTINE/MYSUB 语句将 MYSUB 定义为子例程。

A1 是第一个自变量的名称；因此，当进行调用时，A1 最初的值与调用时 V1 的值相同，即 6。

然后，子例程执行它的两条语句：

- 第一条语句是 ASSIGN 命令，它将 A1 的值增加到 7。
- 第二条语句是显示更新的 A1 值的 COMMENT/OPER 命令。

然后，子例程以 ENDSUB/ 命令结束。

然后，执行流程返回到直接跟在 CALLSUB 命令之后的语句。这是另一个 COMMENT/OPER 命令，显示 V1 的最终值。





做为执行跳转回来，在这个程序中任何变量做为自变量 **V1**，会更新成符合当前子程序中的变量值，在此程序中的 **A1**。因此，从子例程返回后，**V1** 现在的值为 **7**。此值从子例程传回。

### 向子程序中传递数字值

自变量同样可是是数字特征。



#### 传递数值的示例：

此样例显示了如何在子程序中传递数字值。它不能同时传递两个数值然后并且添加它们。

```
CS1      =CALLSUB/SUM_NUMBERS, :,  
CS2      =CALLSUB/SUM_NUMBERS, :5,10,,  
          SUBROUTINE/SUM_NUMBERS,  
            NUM1 = 1 : FIRST NUMBER,  
            NUM2 = 1 : SECOND NUMBER,  
            =  
          ASSIGN/TOTAL=NUM1+NUM2  
          COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-CONTINUE=NO,OVC=NO,  
            NUM1 + "+" + NUM2 + "=" + TOTAL  
          ENDSUB/
```

在第一个 **CALLSUB** 命令 (**CS1**) 中，没有数字值传递到子例程中。改用默认数字，即 **NUM1** 为 1，**NUM2** 为 1，总和为 2。

在第二个 **CALLSUB** 命令 (**CS2**) 中，传递了两个数字值：5 和 10。所以，**NUM1** 是 5，**NUM2** 是 10，总和是 15。

### 将文本字符串传递给子例程

自变量同样可以是文本字符。传递一个文本字符，确保文本字符特征处于双引号标记。



### 传递文本字符的示例：

该样例显示了如何传递字符值到子程序中。它传递了两个字符值到两个参数中然后在报告中显示它们：

```
CS1      =CALLSUB/GET_OPERATOR_INFO,:"BOB JONES","MORNING",,,
          SUBROUTINE/GET_OPERATOR_INFO,
          OPNAME = "<NO PASSED OPERATOR>" : OPERATOR NAME,
          SHIFT = "<NO PASSED SHIFT>" : SHIFT TIME,
          =
          COMMENT/REPT,
          OPNAME
          COMMENT/REPT,
          SHIFT
          ENDSUB/
```

第一个自变量，`OPNAME` 收到传递过来的值 "BOB JONES" 然后第二个自变量，`转换`收到 "MORNING"。然后，两个 `COMMENT/REPT` 命令将传递的字符串发送到检验报告。

### 将特征名传递给子例程

您可以传递大括号内的特征名称。例如，在 `CALLSUB` 命令中，如果使用 `{PNT1}` 作为参数，则软件将 `PNT1` 传递给子例程。当您以这种方式传递特征名称时，子例程可以完全访问该特征。



### 传递特征名的示例：

该样例将在特征名 `PNT1` 中的变量传递到子程序中，使你的子程序完成特征的传递。如果没有特征传递过去则使用特征的默认名称 `F1`。然后该子程序会询问你一个 新的值用来改变特征的 `X` 的理论值。

```
CS1      =CALLSUB/CHANGE_THEO_X,:{PNT1},,
```

## 子例程分支

```
C1          =COMMENT/INPUT,NO,FULL SCREEN=NO,

            Passed feature:

            FEAT1

            The current theo X is:

            FEAT1.TX

            Type a new theo X value:

            ASSIGN/FEAT1.TX=C1.INPUT

            COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-CONTINUE=NO,OVC=NO,

            "Feature updated to " + FEAT1.TX
```

由于使用子例程可访问传递的特征，因此语句 `ASSIGN/FEAT1.TX=C1.INPUT ?>` 可修改 PNT1 实际特征的 X 理论值。PNT1 将永久更改其 X 理论值。

## 子程序样例代码

以下样例代码使操作者可以选择在测量特征后更改特征的 X、Y 和 Z 理论值。以后的运行将使用新的理论值。

```
PNT1        =GENERIC/POINT,DEPENDENT,CARTESIAN,$

            NOM/XYZ,<5,10,15>,$

            MEAS/XYZ,<7,12,17>,$

            NOM/IJK,<0,0,1>,$

            MEAS/IJK,<0,0,1>

C1          =COMMENT/YESNO,NO,FULL SCREEN=NO,AUTO-CONTINUE=NO,

            Do you want to change the theoretical values for PNT1?

            IF/C1.INPUT=="YES"
```

```

CS1      =CALLSUB/CHANGETHEO, :,

          END_IF/

          COMMENT/OPER, NO, FULL SCREEN=NO, AUTO-CONTINUE=NO,

          The XYZ theoretical and actual values for PNT1 are:

          "Theo X= "+PNT1.TX

          "Theo Y= "+PNT1.TY

          "Theo Z= "+PNT1.TZ

          -----

          "Act1 X= "+PNT1.X

          "Act1 Y= "+PNT1.Y

          "Act1 Z= "+PNT1.Z

          ROUTINE/END

          SUBROUTINE/CHANGETHEO,

              POINT1={PNT1} :,

              =

          DIMINFO/PNT1; ICON, DIMID, FEATID, VERT, HORIZ, , $

              HEADINGS, ; MEAS, , , , , , ,

C2      =COMMENT/INPUT, NO, FULL SCREEN=NO,

          Type the new X theo value for PNT1.

          "Its current value is "+PNT1.TX

          ASSIGN/PNT1.TX=C2.INPUT

C3      =COMMENT/INPUT, NO, FULL SCREEN=NO,

          Type the new Y theo value for PNT1.

          "Its current value is "+PNT1.TY

          ASSIGN/PNT1.TY=C3.INPUT

C4      =COMMENT/INPUT, NO, FULL SCREEN=NO,

```

## 子例程分支

```
Type the new Z theo value for PNT1.  
  
"Its current value is "+PNT1.TZ  
  
ASSIGN/PNT1.TZ=C4.INPUT  
  
ENDSUB/
```

### 样例代码的说明

**C1=COMMENT/YESNO**

此行取用并存储用户的 YES 或 NO 响应。

**IF/C1.INPUT=="YES"**

此行是表达式。它测试注释 1 的输入是否为 YES。若为 YES，则 IF 语句为 TRUE，并在 IF 语句后继续执行语句，在本例中，测量的是 **PNT1** 特征。若为 NO，则移至 **END\_IF** 语句。

**CS1=CALLSUB/CHANGETHEO,;**

此行调用名称为 **CHANGETHEO** 的子例程。测量例程流现在跳至 **SUBROUTINE/CHANGETHEO** 行。

**SUBROUTINE/CHANGETHEO**

此行初始化 **CHANGETHEO** 子例程。测量例程流继续执行此行和 **ENDSUB/** 行之间的代码。

**POINT1 = {PNT1} :**

此是子例程的唯一自变量。允许子例程访问 **PNT1** 特征信息。

**C2=COMMENT/INPUT, C3=COMMENT/INPUT, C4=COMMENT/INPUT**

这些输入注释均取用用户新的 X、Y、Z 理论值，并将它们分别存储在 **C2.INPUT**、**C3.INPUT** 和 **C4.INPUT** 中。

**ASSIGN/PNT1.TX = C2.INPUT**

此行从 **C2.INPUT** 取用 X 理论值，并将其分配给 **PNT1.TX** 变量。**PNT1.TX** 是 **PC-DMIS** 变量，其值为具有 **PNT1** ID 标签的点的 X 理论值（以 TX 表示）。

**ASSIGN/PNT1.TY = C3.INPUT**

此行从 C3.INPUT 取用 Y 理论值，并将其分配给 PNT1.TY 变量。PNT1.TY 是 PC-DMIS 变量，其值为具有 PNT1 ID 标签的点的 Y 理论值（以 TY 表示）。

**ASSIGN/PNT1.TZ = C4.INPUT**

此行从 C4.INPUT 取用 Z 理论值，并将其分配给 PNT1.TZ 变量。PNT1.TZ 是 PC-DMIS 变量，其值为具有 PNT1 ID 标签的点的 Z 理论值（以 TZ 表示）。

**ENDSUB/**

此行结束子例程，调用子例程后测量例程流立即返回该行。在本例中，为 END\_IF/ 语句。测量例程流继续执行显示 X、Y 和 Z 理论值和实际值的下一操作员注释，然后以 ROUTINE/END 命令结束测量例程。

### 注释后的命令模式命令



插入 PC-DMIS 注释后，要在命令模式下键入其他 PC-DMIS 命令，必须先在 COMMENT 命令后按 Enter 两次。这告诉 PC-DMIS 您不再想在注释中添加文本，而是准备添加新命令。

## 终止测量例程

插入 | 流控制命令 | 结束例程菜单项插入 ROUTINE/END 命令到“编辑”窗口。只要 PC-DMIS 在执行例程期间遇到此命令，其将立即停止测量例程执行。

当你终止较通常提早终止一个测量例程时，这条指令是非常有利的，但必须定义条件。