目录

使用文件输入/输出 1

使用文件输入/输出：简介 1

了解基本的文件 输入/输出 概念 2

使用文件 输入/输出对话框。 2

打开文件读写 3

读取或写入后关闭打开的文件 5

从文件中读取字符 6

从文件中读取行 8

读取文件中的文本块 12

读取字符至分隔符 15

将字符写入文件 17

将行写入文件 19

将文本块写入文件 20

在文件开头放置文件指针 22

保存文件指针的当前位置 24

回调保存的文件指针位置 25

复制文件 27

移动文件 28

删除文件 30

检查文件是否存在 31

显示文件对话框 33

检查文件或行是否结束 34

使用文件输入/输出

使用文件输入/输出：简介

本章说明如何输入和输出信息至测量例程。使用可用的菜单项，可在读取或写入模式下打开文件。然后可以读取或写入这些文件。使用文件 I / O
命令，可以在测量例程中使用从外部文件读取的数据。同样，使用这些命令，也可以将测量及公差信息写回文件。您也可以使用这些命令执行其他文件操作。

本章详细介绍这些文件 I/O
操作，包括每种操作的有效示例。这些示例中使用的项目在“使用流控制进行分支”一章和“使用表达式和变量”一章。

注：样例代码中在发生循环或分支时使用缩进，以便清楚地显示为某个条件分配的语句。在实际的“编辑”窗口代码中，不会显示任何缩进。

此部分的主题包括：

- 了解基本的文件 输入/输出 概念
- 使用文件 输入/输出对话框
- 打开文件读写
- 读取或写入后关闭打开的文件
- 从文件中读取字符
- 从文件中读取行
- 读取文件中的文本块
- 读取字符至分隔符
- 将字符写入文件
- 将行写入文件
- 将文本块写入文件
- 在文件开头放置文件指针
- 保存文件指针的当前位置
- 回调保存的文件指针位置
- 复制文件
- 移动文件
- 删除文件
- 检查文件是否存在
- 显示文件对话框
- 检查文件结尾或命令行结尾

了解基本的文件 输入/输出 概念

检查文件是否存在
对于所有文件 输入/输出 操作，您可能首先要检查文件是否存在。该检查应加入 IF / THEN 循环，以便在检查失败后可以通知用户。在写入文件时，必须先在 Windows 环境中创建该文件。

参见“检查文件是否存在”。

打开和关闭文件：
对于读取或写入文件的操作，需首先打开文件以执行系统的处理。通过为文件分配一个变量（即文件指针），可完成此操作。打开文件时，可指定是否将打开此文件以执行读取、写入（覆盖）或附加操作。文件打开后，可读取或写入该文件。处理完文件后，应关闭文件指针，这样可关闭文件以便供其它系统进程访问。不能打开已被其他进程打开的文件。

参见'打开文件进行读取或写入"和"读取或写入后关闭打开的文件"。

文件指针和位置：
文件指针是指向文件的变量。文件指针存储打开的文件的名称和位置，然后用于读取或写入该文件。文件打开并设置到文件指针后，指针的作用类似字处理程序中的光标。用于指示您在文件中当前读取或写入的位置。

- 如果要对文件进行附加操作，文件指针通常位于文件的末尾。
- 如果要读取或覆盖文件，文件指针通常应位于文件的开头。

写入或读取时使用定界符

在写入数据时，可考虑用分隔符来分隔数据。这样便于将数据读回到测量例程。分隔符可以是任何字符或字符串。例如，假定您有一个点，其名称为 PNT1，其 X、Y、Z 测量值为 2.5,4.3,6.1。通过以下代码，您可将这些由逗号分隔的值轻松写入数据文件：

```
FILE/WRITELINE,FPTR,PNT1.X + "," + PNT1.Y + "," + PNT1.Z
```

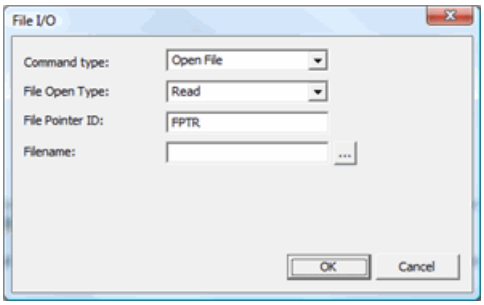
在读取数据时，可以使用指定的定界符分隔接收的数据，并将数据存入变量供以后使用。例如，假定要读入上面所列的 X、Y 和 Z 值。这些值应在一个如下所示的文本行中：2,5,4.3,6.1。您可以使用逗号分隔文本，并使用类似如下的代码行将这些值存入相应的变量：

```
V1=文件/读取行,FPTR,{ValX}+","+{ValY}+","+{ValZ}
```

然后可在测量例程中将 `ValX`、`ValY` 和 `ValZ` 作为标准变量使用。结果为：ValX = 2.5，ValY = 4.3，ValZ = 6.1。

使用文件 输入/输出对话框。

所有的文件 I/O 命令最初通过选择相应的文件 I/O 菜单项（从菜单中选择**插入 | 文件 I/O 命令**）插入测量例程。在“编辑”窗口中有命令后，按 F9 命令访问其相关的文件 I/O 对话框。



文件 输入/输出对话框样例

此对话框只提供一种方法来编辑*当前*文件输入/输出命令。此外，您也可以使用“使用编辑窗口”一章中讨论的技术对“编辑”窗口中的命令进行修改。

用户不能用这个对话框插入一个*新建*文件I/O命令。要新建一个命令，需要在选择适当的菜单选项或在编辑窗口中直接写入命令。

打开文件读写

插入 | 文件 I/O 命令 |
打开文件菜单项允许您在“编辑”窗口中插入一条命令，在测量例程执行的过程中该命令将打开的磁盘文件。

您可以仅为查看信息打开文件，也可以为添加和保存信息打开文件。

此命令在“编辑”窗口中的语法为：

```
<文件指针名> =文件/打开,<文件名>,<打开模式>
```

此命令的一些组成部分说明如下：

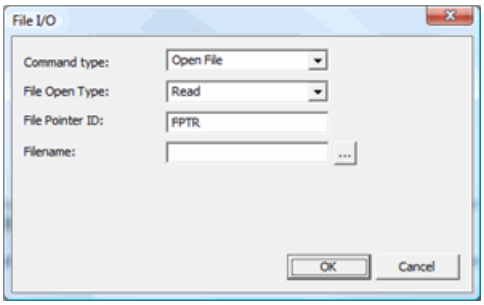
<filepointername>
用户选择的访问已打开的文件所使用的 filepointer 的 ID。此 ID 用于在其他文件 I/O 命令中引用打开的文件。

<filename>
要打开的磁盘文件的文件名。

<openmode>
打开文件时所处的模式。应在以下模式下打开文件：读取、写入或附加。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“文件打开”命令上。
3. 按下F9。



文件打开的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

以下代码打开 TEST.TXT 文件进行读取、写入和附加，然后将文件名存储到文件指针 FPTR。

```
FPTR=文件/打开,C:\PCDMIW\TEST.TXT,读取
FPTR=文件/打开,C:\PCDMIW\TEST.TXT,写入
FPTR=文件/打开,C:\PCDMIW\TEST.TXT,附加
```

注意，您可以使用将完整路径作为输入的输入注释，并在文件/打开命令中使用该输入。使用文件/对话命令也可以完成该任务。参见以下示例：

```
C1=注释/输入,键入文件的完整路径和名称。
V1=文件/对话框, 选择要打开的文件
FPTR=FILE/OPEN,C1.INPUT,READ
FPTR=文件/打开,V1,读取
```

参见显示文件对话框。

读取或写入后关闭打开的文件

插入 | 文件 I/O 命令 |
关闭文件菜单项允许您在“编辑”窗口中插入一条命令，在测量例程执行后该命令将关闭打开的文件。关闭文件可释放打开文件时占用的资源，并将对文件所作的更改存储至磁盘。

此命令在“编辑”窗口中的语法为：

```
文件/关闭, <文件指针名>
```

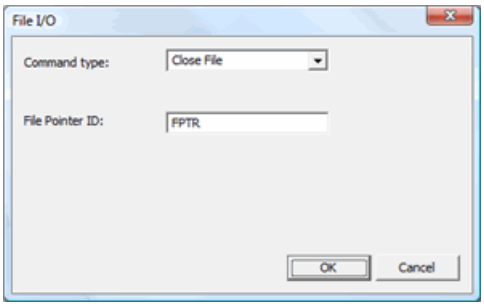
此命令的一些组成部分说明如下：

<filepointername>
识别文件所使用的 ID，此 ID 于打开文件时创建。

<closemode>
此参数有两个选项，即“保持”或“删除”。使用“保持”时，PC-DMIS
只关闭在文件指标中定义的文件。使用“删除”时，PC-DMIS 先关闭文件然后再将其删除。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“文件关闭”命令上。
3. 按下F9。



文件关闭的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

以下代码关闭为文件指针 FPTR 分配的文件：

```
文件/关闭,FPTR,保留
```

此代码使用的是 DELETE 参数，它关闭并删除分配给 FPTR 的文件：

```
文件/关闭,FPTR，删除
```

从文件中读取字符

插入 | 文件 输入/输出 | 读取命令 |
读取字符菜单选项用于在“编辑”窗口中放入一条命令，以便从文件中读取“文件指针名”字段（参见以下语法）中指定的单个字符，并将该字符赋值到在“变量名”字段中指定的变量。

此命令在“编辑”窗口中的语法为：

```
<变量名> = 文件/读取字符,<文件指针名>
```

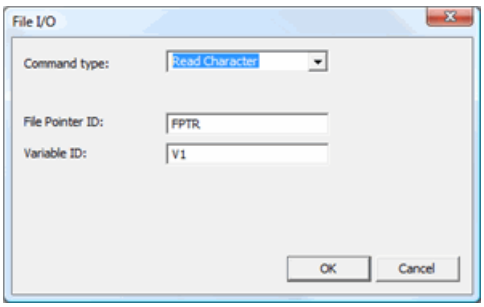
此命令的一些组成部分说明如下：

<filepointername>
用于打开文件的 ID。

<varname>
将占有该字符的变量的名称。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“读取字符”命令上。
3. 按下F9。



读取字符的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

参考以下示例，从数据文件中读入一行，一次读取一个字符，直到遇到空格。

```
V1=文件/存在,test.txt
如果/V1<>0
注释/操作者,可以读取数据文件。单击“确定”继续。
赋值/V3 = ""
FPTR=文件/打开,D:\Program Files\pcdmis35\test.txt,读取
执行/
V2=文件/读取_字符,FPTR
赋值/V3 = V3+V2
直到/V2 == ""
文件/关闭,FPTR
注释/操作者,"文件中文本行的第一个字为：" + V3
END_IF/
ELSE/
COMMENT/OPER, 不能读取数据文件。现在将退出例程。
转到/终止
END_ELSEIF/
终止=标号/
ROUTINE/END
```

代码说明

V1=文件/存在
该行检查指定的文件是否存在。文件必须位于 PC-DMIS 所在的目录，该代码才有效，否则，包含文件的行还必须包含文件的完整路径。V1 接收文件检查结果。如果存在，则为非零值；反之为零。

IF/V1<>0
此行取用 V1 的值，并检查以确定其结果是否为非零值。如为非零值，则会显示注释，表明已准备好开始读取过程。如为 0，则将结束测量例程。

赋值/V3 = ""
该行创建一个空字符串，并将其赋给 V3。代码使用该变量，通过分别读入的字符构造字符串。如果不建立空字符串，则 V3 的默认值为 0。

FPTR=文件/打开
该行打开指定的文件进行读取，并将其分配给默认的文件指针 FPTR。

DO
该行开始 DO / UNTIL 循环。该行绑定 FILE/READ_CHARACTER 代码，以便可以一次一行连续读入行。循环在读入空格字符时退出。

V2=文件/读取字符,FPTR
该行从绑定到文件指针FPTR的打开文件中读入一个字符。字符存储在变量 V2 中。

ASSIGN/V3 = V3+V2
该行使用空的 V3 变量将字符串 V3 与 V2 串联，然后将值重新赋给 V3。所以，在以后运行 DO/UNTIL 循环时，V3 将增加一个字符。

UNTIL/V2 == " "
该行在文件/读取_字符代码从打开的文件中遇到空格字符时终止 DO / UNTIL 循环。

文件/关闭,FPTR
该行关闭打开的数据文件，以供其它系统进程访问。代码的剩余部分完成运行，并在操作者注释中显示数据文件中的第一个字。

从文件中读取行

插入/文件输出/输入命令/读取命令/读取行菜单选项在编辑窗口中放置一命令，在执行时从指定文件中读取行。该命令会将变量标识所指定的变量设置为 1（真）或 0（假），以指示调用是成功（真）还是失败（假）。此命令所需的表达式可用于界定读入的行，并自动使用从文件读入的数据填充变量和引用。输入文件中的信息将读至下一个回车符。

此命令在“编辑”窗口中的语法为：

```
<变量名> = 文件/读取行,<文件指针名>,<表达式>
```

此命令的一些组成部分说明如下：

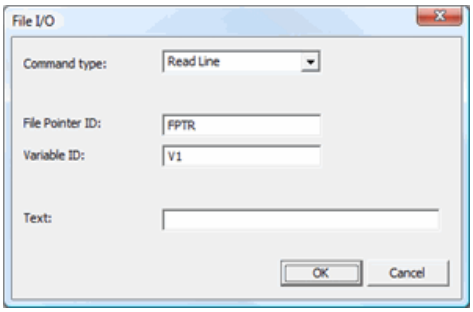
<varname>
变量的名称，该变量将保存指明“读取行”命令是成功还是失败的结果。将返回“确定”或 "EOF"。

<filepointername>
打开文件时为文件指针指定的名称。

<expr>
输入数据的目标变量。输入数据可以用文本进行界定，以便于分析读入的数据行。变量和特征引用应该用大括号括起来。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“读取行”命令上。
3. 按下F9。



读取行的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

请看一下此例，它每次从数据文件中读取一行，直至 FILE/READ_LINE 命令遇到空行。然后测量程序会显示得到的文本块并退出。

```
V1=文件/存在,test.txt
如果/V1<>0
注释/操作者,可以读取数据文件。单击“确定”继续。
赋值/V3 = ""
FPTR=文件/打开,D:\Program Files\pcdmis35\test.txt,读取
执行/
V2=文件/读取行,FPTR,{LINE}
赋值/V3 = V3 + LINE
注释/操作者,"变量 V3 的当前值为:
, V3
UNTIL/V2 == "EOF"
文件/关闭,FPTR
注释/操作者,"文本块显示如下: "
, V3
END_IF/
ELSE/
COMMENT/OPER, 不能读取数据文件。现在将退出例程。
转到/终止
END_ELSEIF/
终止=标号/
ROUTINE/END
```

代码说明

该代码大部分与“读取字符的样例代码”中所述类似。下面仅介绍该示例独有的代码。

DO
该行开始 DO / UNTIL

循环。该行绑定文件/读取行代码，以便可以一次一行连续读入行。循环在到达文件结尾时退出。

V2=文件/读取行, FPTR,{行}
该行命令读取文件中的一行。该命令不是将文本内容保存在V2中，它与文件/读取字符有区别。

- 该示例中的 V2 可返回两个值：
  将返回“OK”或“EOF”。如果仍存在要读入的行，则返回“OK”。如果已到达文件结尾，则返回“EOF”。
- {LINE} 代码是用户输入的变量，用于存储实际文本。该代码使用大括号括起来，通知 PC-DMIS 它属于变量，不属于任何分隔文本。如果没有大括号，PC-DMIS
  将在文件中搜索字符串“LINE”，并且仅返回“LINE”之后、回车符之前的文本。

赋值/V3 = V3+LINE
该行使用空的 V3 变量将字符串 V3 与 LINE 串联，然后将串联后的值重新赋给 V3。所以，在以后运行 DO/UNTIL 循环时，V3 将增加一行。

UNTIL/V2 == "EOF"
此行测试 DO / UNTIL 循环的条件。在 FILE/READLINE
代码到达文件末尾时退出循环。例程流退出循环后，其余代码结束运行，并在操作员注释里面显示完整的代码块。

其他示例： `Result = File/ReadLine,F1, "Part ID :" + {V1}` -
让所有文本显示在文本“零件 ID：”后的读取行中：被指定为 V1。将 F1
用作文件指针名称在打开的文件中读取该行。读取的结果（成功或失败）会存储在变量结果中。

```
File/ReadLine,F1, "Location:"+{VARX}+ ","+{VARY}+ ","+{VARZ}+ ","+{VARI}+ ","+{VARJ}+ ","+{VARK}
ASSIGN/CIR1.XYZ = MPOINT(VARX, VARY, VARZ)
ASSIGN/CIR1.IJK = MPOINT(VARI, VARJ, VARK)
```

上述三个命令行读取字符串 "Location:" 后用逗号分隔的文本并存储 X, Y, Z 中的值及 CIR1 的 I, J, K 值。

`File/ReadLine,F1, "Value # " + loopvar + " : " + {var2}` -
将使用冒号之后出现的文本填充 var2。此示例中的 loopvar 变量不带大括号，因此属于分隔文本。

前缀是零的数字处理的样例代码

如果您读取的读取的文件包含数行，您讲注意到PC-DMIS忽略了在前面为零的字符。例如，如果行包含了005450，它将严格作为数字来读取值并且返回数值5450，忽略前面的两个零。您可能要或者不要这两个零。

建议您创建一个外部条形码文本文件来读软件和它包含的这两行数据

290291143;582750;0010

291143;5827;0010

您可以使用一些以下样例代码来得到在分号之间的数值：

```
        赋值/第一_值=0
        赋值/第二_值=0
        赋值/第三_值=0
        赋值/LINENUM=1
FPTR=FILE/OPEN,D:\TEMP\CODES.TXT,READ
        执行/
INLINE=FILE/READLINE,FPTR,{FIRST_VALUE}+";"+{SECOND_VALUE}+";"+{THIRD_VALUE}
        注释/OPER,NO,"LINE NUMBER: " + LINENUM
                ,"第一 值: " + 第一_值
                ,"第二 值: " + 第二_值
                ,"第三 值: " + 第三_值
        UNTIL/INLINE == "EOF"
        文件/关闭,FPTR,保留
```

当成功的分析文本的行并且返回到数值，同时也将移除所有数值前缀零后返回。因此，第三_值变量包含了10的值来代替0010.

保留前缀零，您需要把全部行排成一列，在包含数值的文本行利用INDEX,LEFT和MID使用排成一列功能查找分号位置取代。

```
FPTR=FILE/OPEN,D:\TEMP\CODES.TXT,READ
        赋值/LINENUM=1
        执行/
LINESTATUS=FILE/READLINE,FPTR,{LINESTR}
        ASSIGN/LINESTR=STR(LINESTR)
        ASSIGN/FIRST_INDEX=INDEX (LINESTR,";")
        ASSIGN/FIRST_VALUE=STR(LEFT(LINESTR,FIRST_INDEX-1))
        ASSIGN/REMAINSTR=STR(MID(LINESTR,(FIRST_INDEX)))
        ASSIGN/SECOND_INDEX=INDEX (REMAINSTR,";")
        ASSIGN/SECOND_VALUE=STR(LEFT(REMAINSTR,SECOND_INDEX-1))
        ASSIGN/THIRD_VALUE=STR(MID(REMAINSTR,SECOND_INDEX))
        注释/OPER,NO,"LINE NUMBER: " + LINENUM
                ,"第一 值: " + 第一_值
                ,"第二 值: " + 第二_值
                ,"第三 值: " + 第三_值
        赋值/LINENUM=LINENUM+1
        UNTIL/LINESTATUS == "EOF"
        文件/关闭,FPTR,保留
```

代码说明

该代码大部分与文件复制的样例代码中所述类似。下面仅介绍该示例独有的代码。

ASSIGN/FIRST_INDEX=INDEX (LINESTR,";")
此行定位代码行中第一个分号的位置，并将该位置分配给变量 FIRST_INDEX。

ASSIGN/FIRST_VALUE=STR(LEFT(LINESTR,FIRST_INDEX-1))
此行为 FIRST_VALUE 变量分配 LINESTR
变量中第一个分号前（但不包括第一个分号）的字符字符串。LINESTR 包含整行文本。

ASSIGN/REMAINSTR=STR(MID(LINESTR,(FIRST_INDEX)))
此行为变量 REMAINSTR（表示“其余字符串”的意思）分配从 FIRST_INDEX
位置（第一个分号位置）开始至该行结束以外的字符字符串。

ASSIGN/SECOND_INDEX=INDEX(REMAINSTR,";")
此行在 REMAINSTR 变量中搜索另外一个分号（行中第二个分号）并将此位置分配给变量 SECOND_INDEX。

ASSIGN/SECOND_VALUE=STR(LEFT(REMAINSTR,SECOND_INDEX-1))
此行为 SECOND_VALUE 变量分配
REMAINSTR（整行中的第二个分号）变量中第一个分号前（但不包括第一个分号）的字符字符串。

ASSIGN/THIRD_VALUE=STR(MID(REMAINSTR,SECOND_INDEX))
此行为变量 THIRD_VALUE 分配从 SECOND_INDEX 位置开始至该行结束的字符字符串。

读取文件中的文本块

插入 | 文件 输入/输出 | 读取命令 |
读取块菜单选项用于在“编辑”窗口中放入一条命令，以便在执行过程中从打开的文件读取字符块。读入的字符数量用大小参数来指定。

此命令在“编辑”窗口中的语法为：

```
<变量名>=文件/读取块,<fptrname>,<大小>
```

此命令的一些组成部分说明如下：

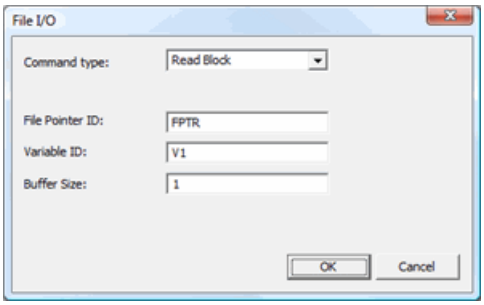
<varname>
接收指出读取块操作是成功还是失败的值的变量的 ID。

<fptrname>
打开文件时为文件指针指定的名称。

<size>
要读取的字符数。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“读取块”命令上。
3. 按下F9。



读取块的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

假定您有多个外部数据文件，包含各种零件数据，每个文件的前几个字符指示文件的内容。您可以使用文件/读取块 命令仅读取前几个字符，然后再决定读入和处理每一行。参考以下代码：

```
C3=注释/输入, 请键入名称
所寻找的文件代码
ASSIGN/BLOCKSIZE = LEN(C3.INPUT)
ASSIGN/FILECODE = C3.INPUT
执行/
C1=注释/输入,请键入完整的路径、
, 文件名和扩展名用于
要处理的文件。
,键入 [Q] 退出。
IF/C1.INPUT== "Q" OR C1.INPUT=="q"
COMMENT/OPER, 您选择了退出。例程现在正在退出。
转到/终止
END_IF/
V1=FILE/EXISTS,C1.INPUT
如果/V1<>0
COMMENT/OPER,“数据文件 [" + C1.INPUT + "] 存在。单击“确定”继续。”
FPTR=FILE/OPEN,C1.INPUT,READ
V2=文件/读取块,FPTR,BLOCKSIZE
文件/关闭,FPTR
IF/V2<>FILECODE
注释/操作者,"文件的代码 [" + V2 + "] 与"
, “文件代码 [" + FILECODE + "] 匹配。”
END_IF/
直到/V2==FILECODE
COMMENT/OPER,“文件 [" + C1.INPUT + "] 匹配。”
,"文件的代码 [" + V2 + "] 与"
, “文件代码 [" + FILECODE + "] 匹配。”
```

```
COMMENT/OPER, 例程处理了此文件。
END_IF/
ELSE/
COMMENT/OPER,“数据文件 [" + C1.INPUT + "] 不存在。请使用现有数据文件重试。”
转到/终止
END_ELSEIF/
终止=标号/
ROUTINE/END
```

代码说明

该代码大部分与“读取字符的样例代码”或“读取行的样例代码”中所述类似。

下面仅介绍该示例独有的代码。

ASSIGN/BLOCKSIZE = LEN(C3.INPUT)
此行创建一个用户定义的名称为 BLOCKSIZE 的变量，该变量包含有一个与 C3.INPUT 中的字符数相等的整数。这个整数将作为要读取的字符块的大小。

ASSIGN/FILECODE = C3.INPUT
此行创建 FILECODE 变量，并为其分配变量 C3.INPUT 的值。

C1=COMMENT/INPUT
这条注释将用户输入的完整路径存储到 C1.INPUT 变量。

V1=FILE/EXISTS,C1.INPUT
此行检查在 C1 注释中定义的文件名的存在与否。

DO/
该行开始 DO / UNTIL
循环。该行绑定代码块，使用户可以指定要读取的文件。该行将继续循环，直到为 FILECODE 变量指定的文本与从文件读取的文本匹配。

V2=文件/读取块,FPTR,BLOCKSIZE
该行读取 BLOCKSIZE 变量中包含的整数个字符。该文保存在V2变量中。

IF/V2<>FILECODE
该行开始 IF / END IF 代码块，测试 V2 变量中的文本与 FILECODE 变量中存储的文本是否匹配。如果匹配，例程继续运行。否则，将显示一条消息，表明两个代码不匹配。

UNTIL/V2==FILECODE
此行检查 DO / UNTIL 循环的条件，确定 V2 变量中的文字是否符合 FILECODE 变量中的文字。如语句的结果为 False，则 DO 循环会再次运行，可让用户选择一不同的文件名。如语句的结果为 True，则表明此循环已存在，例程会显示一条表明符合的消息。然后 PC-DMIS 可继续从指定的数据文件读取每个数据行。

读取字符至分隔符

插入 | 文件输入/输出命令 | 读取命令 |
读取行菜单项用于在“编辑”窗口中插入一条命令，该命令在执行过程中会从指定的文件读取“最多”一个分隔符前的所有文本。此命令所读取的所有文本都将放入指定的目标变量。当 PC-DMIS 遇到以下对象，该命令将停止读取文本：

- 定义的定界符
- 回车
- 换行字符

如果达到文件的末尾，目标变量将设置为“EOF”（文件末尾）。

此命令在“编辑”窗口中的语法为：

```
<变量名> = 文件/读取至,<fptrname>,<定界符>
```

此命令的一些组成部分说明如下：

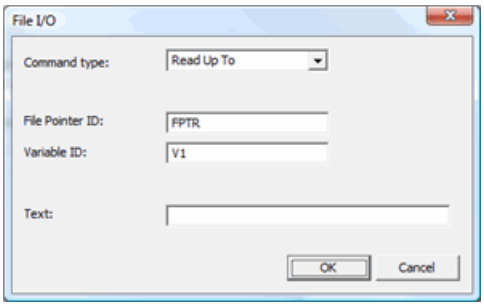
<varname>
目标变量的名称。

<fptrname>
打开文件时为文件指针指定的名称。

<delimiters>
包含零个或多个分隔符的字符串。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 选择 **编辑窗口**打开“编辑”窗口。
2. 将光标放在“文件/读取”命令上。
3. 按下F9。此时将打开**文件 I/O**对话框。



当显示该对话框时，请执行以下操作：

1. 键入变量名，该变量将接收读入**变量标识**框的信息。
2. 在**文件指针标识**框中键入文件指针名。

3. 在**文本**框中键入定界符（务必要将选定定界符用引号引起来）。
4. 单击**确定**。

读取至的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

例如：假定在 c:\temp 目录中有一个名为“sample.txt”的文本文件，该文件的第一行包含以下信息：

root:x:0:0:root:/root:/bin/bash

要将“读取至”命令用于此文件，请执行以下步骤：

1. 在“编辑”窗口中插入“文件/打开”命令。
2. 使用所选的文件指针名来给“文件打开”命令命名。此示例将“样例”用作文件指针名。

“文件打开”命令应如下所示：

```
样例 =文件/打开,C:\TEMP\SAMPLE.TXT,读取
```

现在，使用 PC-DMIS
的“读取至”命令定义一些变量，用以调用不同的数据段。此示例使用以下变量来查找用作定界符的“:”（不带引号）。

```
用户名 =文件/读取至,样例,:
密码 =文件/读取至,样例,:
用户 =文件/读取至,样例,:
```

这样，当 PC-DMIS 执行这些行时，它将设置

用户名 = root

密码 = x

用户 = 0

要在执行过程中将以下设置显示于屏幕上，可以使用如下所示的操作者注释：

```
注释/操作者,以下文本从 sample.txt 中读入
,用户名:
,用户名:
,密码:
,密码:
,用户:
,用户
```

将字符写入文件

插入 | 文件 输入/输出 | 写入命令 |
写入字符菜单选项用于在“编辑”窗口中插入一条命令，此命令将在执行时将单个字符输出到磁盘文件。

此命令在“编辑”窗口中的语法为：

```
文件/写入字符,<fptrname>,<表达式>
```

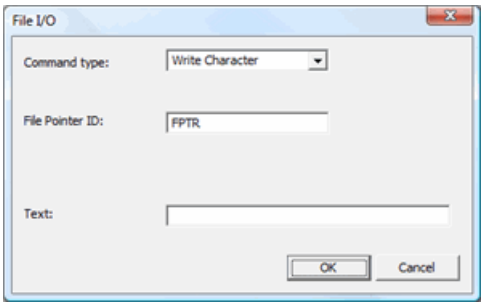
此命令的一些组成部分说明如下：

<fptrname>
打开文件时为文件指针指定的名称。

<expr>
要写入到文件的字符。如果表达式的结果为有多个字符，仅写入第一个字符。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“写入字符”命令上。
3. 按下F9。



写入字符的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在**文件输入/输出**对话框中输入。

参考以下代码，将用户提供的字符串写入数据文件，一次写入一个字符。

```
C1=注释/输入,键入要写入文件的文件名
,到（包括完整路径）。
FPTR=FILE/OPEN,C1.INPUT,WRITE
C2=注释/输入,键入要发送到文件的内容。
,这将发送字符串，一次发送
,一个字符。
```

```
赋值/COUNT = 0
ASSIGN/LENGTH = LEN(C2.INPUT)
执行/
ASSIGN/WRITETHIS = MID(C2.INPUT,COUNT,1)
文件/写入字符,FPTR,WRITETHIS
赋值/COUNT = COUNT + 1
UNTIL/COUNT == 长度
```

代码说明

该代码大部分与“读取字符的样例代码”或“读取行的样例代码”中所述类似。

下面仅介绍该示例独有的代码。

FPTR=FILE/OPEN,C1.INPUT,WRITE
此行打开 C1 注释中指定的文件以执行写入，并将其分配给文件指针 FPTR。只要文件指针是从数据文件的起始处开始，则文件中的所有数据均将被覆盖。

赋值/COUNT = 0
该行为用户定义的变量 COUNT 赋予值零。赋值/COUNT = 0 该行为用户定义的变量 COUNT 赋予值零。用于循环，输出字符串，一次输出一个字符。

ASSIGN/LENGTH = LEN(C2.INPUT)
此行使用 LEN( ) 函数传回字符串长度。该函数取一个参数，即字符串。它会计算字符串中的字符数（包括空格），并传回字符数的整数值。在本例中，由用户定义的变量 LENGTH 保存该值。

DO/
该行开始 DO / UNTIL 循环。DO 和 UNTIL 语句之间的代码将执行到循环条件值为 True。

ASSIGN/WRITETHIS = MID(C2.INPUT,COUNT,1)
此行创建一个用户定义的变量 WRITETHIS，使用 MID( ) 函数传回 C2.INPUT 字符串的子字符串字符，并将其分配给 WRITETHIS。

MID( ) 取三个参数。

- 参数 1：要获取其值的字符串。在本例中，使用的是 C2.INPUT。
- 参数 2：是在字符串中获取字符的位置。字符串的第一个字符的位置为 0，第二个位置为 1，第三个位置为 2，依此类推。在该示例中使用变量 COUNT。
- 参数 3：是从第二个参数的位置开始获取的字符数。在该示例中使用值 1（该示例一次仅写入一个字符，所以不必获取更多字符）。

文件/写入_字符,FPTR,WRITETHIS
该行将 WRITETHIS 变量中存储的字符写入文件指针 FPTR 指定的文件。

赋值/COUNT = COUNT + 1
该行接受当前的 COUNT 值，以 1 为增量递增，然后将新值放回 COUNT。

UNTIL/COUNT == LENGTH
此行测试 DO / UNTIL 循环的条件。在该示例中，循环将不断递增 COUNT 变量，直到其值与 LENGTH 变量相同。然后循环将退出，终止例程。

将行写入文件

插入 | 文件 I/O | 写入命令 | 写入行
菜单选项用于在“编辑”窗口中插入一条命令，此命令将在执行时将一行文本输出到磁盘文件中。使用表达式语法，将变量与测量例程信息输出至文件。写出的文字后面会自动附加一回车符。

此命令在“编辑”窗口中的语法为：

```
文件/写入行,<fptrname>,<表达式>
```

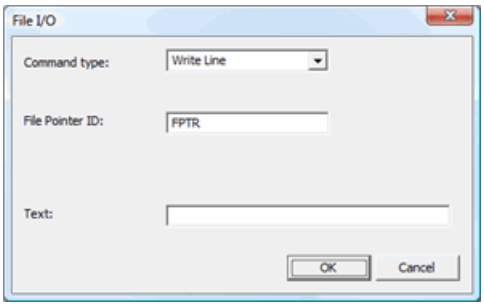
此命令的一些组成部分说明如下：

<fptrname>
打开文件时指定的文件参考的名称。

<expr>
要写入文件的文本。在此字段中可使用表达式。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“写入行”命令上。
3. 按下F9。



写入行的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

假定您要将某些测定 XYZ
值导出到数据文件。以下代码使您可以输入特征标号和数据文件，并将该特征的 X、Y 和 Z
数据发送到数据文件。

```
C1=注释/输入,键入要使用的特征的标号
,使用。
```

```
C2=注释/输入,键入要写入文件的文件名
,到（包括完整路径）。
FPTR=FILE/OPEN,C2.INPUT,APPEND
ASSIGN/FEATNAME = C1.INPUT
ASSIGN/ALLVALS = FEATNAME.X + "," + FEATNAME.Y + "," + FEATNAME.Z
注释/操作者,"要写入的文本为: " + ALLVALS
文件/写入行,FPTR,ALLVALS
文件/关闭,FPTR
```

代码说明

该代码大部分与“读取字符的样例代码”或“读取行的样例代码”中所述类似。

下面仅介绍该示例独有的代码。

FPTR=FILE/OPEN,C2.INPUT,APPEND
此行打开在 C2 注释中指定的文件以执行附加操作，并将其分配给文件指针 `FPTR`。如将 `APPEND` 更改为 `WRITE`，会覆盖数据文件中现有的内容。

ASSIGN/FEATNAME = C1.INPUT
此行将 `C1.INPUT` 的特征标签字符串分配给用户定义的变量 `FEATNAME`。

ASSIGN/ALLVALS=FEATNAME.X+","+FEATNAME.Y+","+ FEATNAME.Z
此行将 `FEATNAME.X,FEATNAME.Y,FEATNAME.Z` 的值分配给用户定义的变量 `ALLVALS`，换言之，即变量 ALLVALS 拥有键入到 C1 输入注释的特征标签的 X、Y 及 Z 的值。

FILE/WRITELINE,FPTR,ALLVALS
此行将 `ALLVALS` 中包含的值写入文件指针 `FTPR` 指定的文件。

将文本块写入文件

插入 | 文件 输入/输出 | 写入命令 |
写入块菜单选项用于在“编辑”窗口中插入一条命令，此命令将在执行时将一个文本块输出到磁盘文件中。使用表达式语法，将变量与测量例程信息输出至文件。与写入行命令不同的是，写入块*不*在末尾附加回车符。

此命令在“编辑”窗口中的语法为：

```
文件/写入块,<fptrname>,<表达式>
```

此命令的一些组成部分说明如下：

<fptrname>
打开文件时指定的文件参考的名称。

<expr>
要写入文件的文本。在此字段中可使用表达式。

回车与换行：不同于写入行命令，写入块命令*不会*在末尾追加换行符。不过，如需将文字放在文本块中新的一行，可在引用的字符串外面使用 CHR(10)
代码，手动插入回车字符和换行字符，如本例所示：

```
FILE/WRITEBLOCK,FPTR,“CHR(10) 在新行上插入文本... " + CHR(10) + " ...。”
```

这将在输出文件内生成此结果：

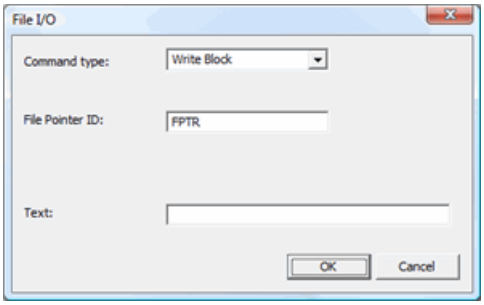
```
CHR(10) 在新行上插入文本...

...。
```

注意，如果 CHR(10) 在引号里面，则 CHR(10) 实际的文字将被发送给文件。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“写入块”命令上。
3. 按下F9。



写入块的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

以下代码将用户输入的内容写入输入注释，附加一个冒号作为定界符。

```
C1=注释/输入,键入任意字符串。PC-DMIS 将追加一个冒号（用于分隔），并将字符串写入所选的文件。
C2=注释/输入,键入要写入文件的文件名
,到（包括完整路径）。
FPTR=FILE/OPEN,C2.INPUT,APPEND
ASSIGN/WRITETHIS = C1.INPUT + ":"
注释/操作者,"要写入的文本为: " + WRITETHIS
文件/写入行,FPTR,WRITETHIS
文件/关闭,FPTR
```

代码说明

该代码大部分与“读取字符的样例代码”或“读取行的样例代码”中所述类似。

下面仅介绍该示例独有的代码。

FPTR=FILE/OPEN,C2.INPUT,APPEND
此行打开在 C2 注释中指定的文件以执行附加操作，并将其分配给文件指针 `FPTR`。

ASSIGN/WRITETHIS = C1.INPUT + ":"
此行会将一个冒号附加到 `C1.INPUT` 中所包含的文本，并将新的字符串分配给用户定义的变量 `WRITETHIS`。

FILE/WRITELINE,FPTR,WRITETHIS
此行将 `WRITETHIS` 中包含的值写入文件指针 `FTPR`
指定的文件。然后，您可以使用冒号作为定界符从文件中读入文本。

在文件开头放置文件指针

插入 | 文件 输入/输出 | 定位命令 |
倒回开头菜单选项用于在“编辑”窗口中插入一条命令，此命令会将文件指针定位到文件流的开头。

此命令在“编辑”窗口中的语法为：

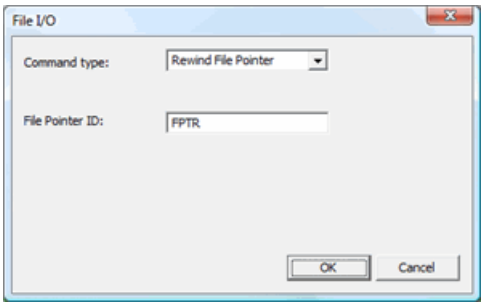
```
文件/倒回,<fptrname>
```

此命令的一些组成部分说明如下：

<fptrname>
要重新定位在文件起始处的文件指针的名称。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“倒回开头”命令上。
3. 按下F9。



倒回开头的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

参考以下示例，从外部文件中读入数据，一次读入一行。在每一行之后，您可以选择从头开始，从文件的开头读取。以下说明如何使用“文件/倒回”命令。

```
C1=注释/输入,请键入要读取的文件
,（包括完整路径）
V1=FILE/EXISTS,C1.INPUT
如果/V1<>0
执行/
FPTR=FILE/OPEN,C1.INPUT,READ
C2=注释/是否,您是否要从开头读取？
IF/C2.INPUT == "YES"
文件/倒回,FPTR
END_IF/
V2=文件/读取行,FPTR,{LINE}
注释/操作者,"当前行为：" + LINE
UNTIL/V2 == "EOF"
END_IF/
文件/关闭,FPTR
COMMENT/OPER, 例程正在退出。
```

代码说明

该代码大部分与“读取字符的样例代码”或“读取行的样例代码”中所述类似。

下面仅介绍该示例独有的代码。

C2=COMMENT/YESNO
此行询问您是否要从起始处开始读取文件。它把“是/否”响应保存在变量 `C2.INPUT` 中。

IF/C2.INPUT == "YES"
此行开始 IF / END IF 块。它测试 `C2.INPUT` 的条件是否具有值“是”。如条件为 True，则 PC-DMIS 会在以下 `IF` 语句中执行代码行。如条件为 False，则 PC-DMIS 会在以下 `END_IF` 语句中执行代码。

文件/倒回,FPTR
该行将文件指针倒回数据文件的开头。

END_IF/
该行退出 IF / END IF 代码块。

保存文件指针的当前位置

插入 | 文件 I/O | 定位命令 |
保存文件位置菜单选项用于在“编辑”窗口中插入一条命令，此命令会将文件指针定位到文件流的开头。所保存的位置可以在随后使用“回调文件位置”命令来回调。

此命令在“编辑”窗口中的语法为：

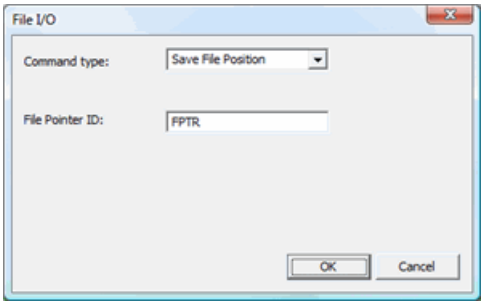
```
文件/保存位置,<fptrname>
```

此命令的一些组成部分说明如下：

<fptrname>
保存其文件位置的文件指针的名称。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“保存文件位置”命令上。
3. 按下F9。



保存文件位置的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

参考以下示例，从外部文件中读入数据，一次读入一行。在每一行之后，您可以选择保存文件位置，以便以后回调。以下说明如何使用“文件/保存位置”命令。

```
C1=注释/输入,请键入要读取的文件
,（包括完整路径）
V1=FILE/EXISTS,C1.INPUT
如果/V1<>0
执行/
FPTR=FILE/OPEN,C1.INPUT,READ
C2=注释/是否,是否要保存文件位置，以便以后回调？循环将退出。
IF/C2.INPUT == "YES"
```

```
文件/保存位置,FPTR
转到/退出循环
END_IF/
V2=文件/读取行,FPTR,{LINE}
注释/操作者,"当前行为: " + LINE
UNTIL/V2 == "EOF"
END_IF/
文件/关闭,FPTR
退出循环=标号/
注释/操作者,您已停止读取。
ROUTINE/END
```

代码说明

该代码与"倒回开头的样例代码"中所述类似。

下面仅介绍该示例独有的代码。

C2=COMMENT/YESNO
此行询问您是否要保存当前文件位置并退出循环。它把“是/否”响应保存在变量 `C2.INPUT` 中。

文件/保存_位置,FPTR
该行将文件指针的位置存储在文件流中。

只要在同一测量例程中打开文件指针名相同的同一文件，您可以回调存储的文件位置，并在您离开的位置继续读取。要继续该示例，请参见“回调文件位置的样例代码”主题。

回调保存的文件指针位置

插入 | 文件 输入/输出 | 定位命令 |
回调文件位置用于在“编辑”窗口中插入一条命令，此命令将回调先前保存的文件位置。使用“保存文件位置”命令来保存已打开文件中的位置。

此命令在“编辑”窗口中的语法为：

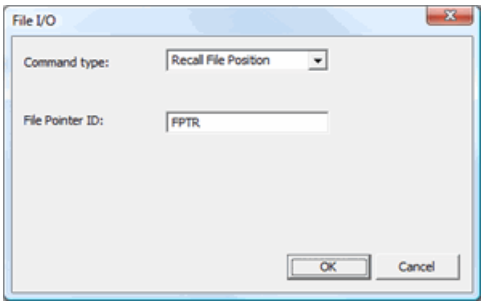
```
文件/回调位置, <fptrname>
```

此命令的一些组成部分说明如下：

<fptrname>
回调其位置的文件指针的名称。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“回调文件位置”命令上。
3. 按下F9。



回调文件位置的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

以下示例打开以前关闭的文件，使用以前的文件指针，并回调存储的文件指针保存位置。然后从该位置读入数据。以下代码说明如何使用文件/回调位置命令，并继续“保存文件位置的样例代码”章节中提供的样例代码。

```
COMMENT/OPER, 已回调先前存储的文件位置。
FPTR=FILE/OPEN,C1.INPUT,READ
文件/倒回,FPTR
注释/操作者,要测试，文件已倒回。
,将读入第一行以测试倒回。
V3=文件/读取行,FPTR,{LINE}
注释/操作者,第一行为：
，行
文件/倒回,FPTR
文件/回调位置,FPTR
注释/操作者,以前存储的文件位置已回调。
,现在将输出该行存储位置的数据。
V4=文件/读取行,FPTR,{STORED}
注释/操作者,存储位置的文本为：
，已存储
```

代码说明

该代码与"倒回开头的样例代码"中所述类似。

下面仅介绍该示例独有的代码。

文件/回调位置,FPTR
该行回调文件流中为文件指针 FPTR 存储的文件指针位置。

V4=文件/读取行,FPTR,{STORED}
该行读入存储的文件指针位置之后的下一行，并将其赋给用户定义的变量 STORED。该变量在下一个操作者注释中显示。

复制文件

插入 | 文件 输入/输出 |
文件复制菜单选项用于在“编辑”窗口中插入一条命令，此命令将在执行时产生文件复制操作。

此命令在“编辑”窗口中的语法为：

```
文件/复制,<源文件名>,<目标文件名>,<替换模式>
```

此命令的一些组成部分说明如下：

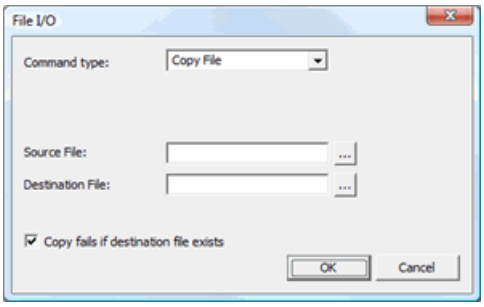
<srcfilename>
源文件（位于复制的文件中）的名称

<destfilename>
目标文件（要复制到的文件）的名称

<replacemode>
目标文件已存在时要采取的操作。如果目标文件已存在，有两种模式：覆盖与失败。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“文件复制”命令上。
3. 按下F9。



文件复制的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

以下代码要求输入要复制的文件名以及要复制到的目标目录和文件。

```
C1=注释/输入,请键入要复制的文件。
,（包括完整文件路径）
C2=注释/输入,请键入目标文件名。
,（包括完整文件路径）
```

```
V1=FILE/EXISTS,C1.INPUT
如果/V1<>0
注释/操作者,存在要复制的文件。正在进行复制。
FILE/COPY,C1.INPUT,C2.INPUT,FAIL_IF_DEST_EXISTS
V2=FILE/EXISTS,C2.INPUT
如果/V2==0
COMMENT/OPER,“文件不存在于：”+ C2.INPUT
,正在终止复制。
ROUTINE/END
END_IF/
ELSE/
注释/操作者,文件复制成功。
ROUTINE/END
END_ELSEIF/
END_IF/
注释/操作者,要复制的文件不存在。
```

代码说明

该代码大部分与读取字符的样例代码或读取行的样例代码中所述类似。

下面仅介绍该示例独有的代码。

C1=COMMENT/INPUT
此行取用要复制的文件的完整路径，并将此路径填入 `C1.INPUT` 变量。

C2=COMMENT/INPUT
此行取用目标文件的完整路径，并将此路径填入 `C2.INPUT` 变量

FILE/COPY,C1.INPUT,C2.INPUT,FAIL_IF_DEST_EXISTS
此行将初始文件复制到目标文件。该命令有三个参数。

- 参数 1 是 `C1.INPUT`。这是要复制的文件的完整路径。
- 参数 2 是 `C2.INPUT`，或者说是目标文件的完整路径。
- 参数

3，在该示例中，如果遇到与目标文件名称相同的现有文件，则放弃“文件/复制”过程。您可以设置该参数，以便覆盖同名的现有文件。

移动文件

插入 | 文件 输入/输出 |
文件移动菜单选项用于在“编辑”窗口中插入一条命令，此命令将在执行时产生文件移动操作。

此命令在“编辑”窗口中的语法为：

文件/移动,<旧文件名>,<新文件名>

此命令的一些组成部分说明如下：

<oldfilename>
文件的位置与名称。

<newfilename>
文件的新位置与新名称。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“文件移动”命令上。
3. 按下F9。

文件移动的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

以下代码要求输入要移动的文件名以及要移动到的位置目录和文件名，然后执行文件移动操作。

```
C1=注释/输入,请键入要移动的文件。
,（包括完整文件路径）
C2=注释/输入,请键入目标文件名。
,（包括完整文件路径）
V1=FILE/EXISTS,C1.INPUT
如果/V1<>0
注释/操作者,存在要移动的文件。正在移动文件。
FILE/MOVE,C1.INPUT,C2.INPUT
V2=FILE/EXISTS,C2.INPUT
如果/V2==0
COMMENT/OPER,“文件不存在于：”+ C2.INPUT
,无法正确移动。
ROUTINE/END
END_IF/
```

```
ELSE/
注释/操作者,文件移动成功。
ROUTINE/END
END_ELSEIF/
END_IF/
注释/操作者,原文件不存在。请重试。
```

代码说明

该代码大部分与文件的样例代码中所述类似。

下面仅介绍该示例独有的代码。

FILE/MOVE,C1.INPUT,C2.INPUT
此行将初始文件复制到目标文件。此条命令取用两个参数。

- 参数 1 是 `C1.INPUT`。这是要移动的文件的完整路径。
- 参数 2 是 `C2.INPUT`，或者说是目标文件的完整路径。

删除文件

插入 | 文件 输入/输出 |
文件删除菜单选项用于在“编辑”窗口中插入一条命令，此命令将在执行时产生文件删除操作。

此命令在“编辑”窗口中的语法为：

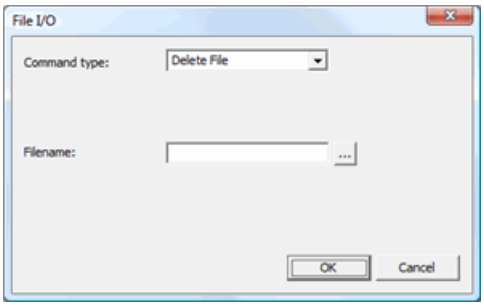
```
文件/删除,<文件名>
```

此命令的一些组成部分说明如下：

<filename>
要删除的文件的名称。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“文件删除”命令上。
3. 按下F9。



文件删除的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

以下代码要求输入文件名，然后删除该文件。

```
C1=注释/输入,请键入要删除的文件。
,（包括完整文件路径）
V1=FILE/EXISTS,C1.INPUT
如果/V1<>0
注释/操作者,文件存在。可以删除。
FILE/DELETE,C1.INPUT
V2=文件/存在,
如果/V2==0
注释/操作者,文件删除成功
ROUTINE/END
END_IF/
ELSE/
注释/操作者,文件仍存在
ROUTINE/END
END_ELSEIF/
END_IF/
注释/操作者,要删除的文件不存在。选择已存在的文件。
```

代码说明

该代码大部分与文件移动的样例代码中所述类似。

下面仅介绍该示例独有的代码。

FILE/DELETE,C1.INPUT
此行删除指定的文件。这条命令取一个参数，即要删除的文件的名称。本例选择的参数是 `C1.INPUT`。

检查文件是否存在

插入 | 文件输入输出命令 |
文件存在用于在“编辑”窗口中插入一条命令，此命令将在执行时检查文件是否存在并用所得结果设置所提供的变量。

此命令在“编辑”窗口中的语法为：

```
<变量名> = 文件/存在,<文件名>
```

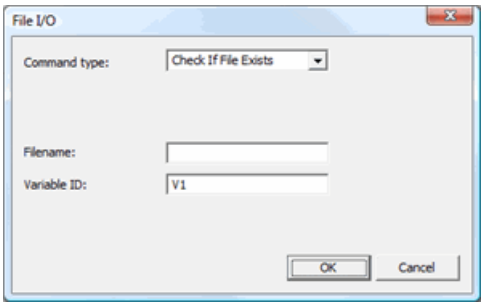
此命令的一些组成部分说明如下：

<filename>
正在检查的文件的名称，以确定在磁盘上是否存在此文件。

<varname>
设置为所执行的检查的结果的变量的名称。如果文件存在，则设置为 1，否则设置为 0。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“文件存在”命令上。
3. 按下F9。



文件存在的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

以下代码要求输入文件名，然后检查该文件是否存在。

```
C1=注释/输入,请键入要检查的文件。
V1=FILE/EXISTS,C1.INPUT
如果/V1<>0
        注释/操作者,文件存在。
END_IF/
ELSE/
        注释/操作者,文件不存在。
END_ELSEIF/
```

代码说明

该代码大部分与读取字符的样例代码或读取行的样例代码中所述类似。

下面仅介绍该示例独有的代码。

V1=FILE/EXISTS,C1.INPUT
此行检查以确定指定的文件是否存在。此文件必须放于 PC-DMIS 所在的目录中，此代码方能工作，否则包含有文件的这一行还必须包含该文件完整的路径。V1 接收文件检查结果。如果存在，则为非零值；反之为零。

显示文件对话框

插入 | 文件 I/O 命令 |
文件对话框菜单项可插入一条命令到“编辑”窗口，在执行过程中会显示**打开**对话框。这可让操作员在执行时选择文件名。它将存储在指定的变量中选择的文件的名称。

此命令在“编辑”窗口中的语法为：

```
<变量名> = 文件/对话框,<表达式>
```

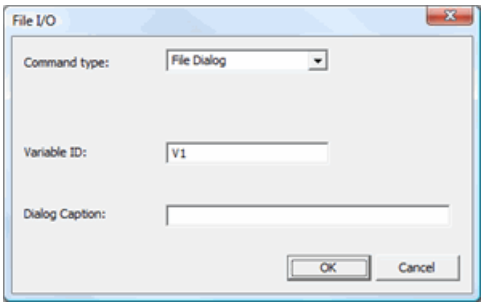
此命令的一些组成部分说明如下：

<varname>
变量名称，此变量名称将设置为用户在文件对话框中选择的文件的名称。

<expr>
将显示在文件对话框的标题栏中的文本。

要访问此“文件 输入/输出”命令的关联对话框，请执行以下步骤：

1. 打开编辑窗口。
2. 将光标放在“文件对话框”命令上。
3. 按下F9。



文件对话框的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在文件输入/输出对话框中输入。

以下代码提供一个对话框，可以选择要删除的文件。

```
V1=文件/对话框,选择要删除的文件。
V2=文件/存在,V1
如果/V2<>0
注释/操作者,文件存在。可以删除。
文件/删除,V1
```

```
V3=文件/存在,
如果/V3==0
注释/操作者,文件删除成功
ROUTINE/END
END_IF/
ELSE/
注释/操作者,文件仍存在
ROUTINE/END
END_ELSEIF/
END_IF/
注释/操作者,要删除的文件不存在。选择已存在的文件。
```

该代码大部分与读取字符的样例代码或读取行的样例代码中所述类似。

下面仅介绍该示例独有的代码。

V1=FILE/Dialog, 选择要删除的文件
此行显示一个名称为“选择要删除的文件”的对话框。您可浏览文件，单击**打开**时，PC-DMIS 将把选择的文件的完整路径指派给 V1。例程的其余部分会删除所选的文件。

检查文件或行是否结束

PC-DMIS可以让您在状态测试时使用EOF或EOL来检查文件是否结束。

EOF代表文件的末尾。此功能采用文件字符串指示器。当适当放置在条件声明中，如果文件指示器已经到达指定文件末尾，将能看到它。如果有，那么此功能返回真值。

EOL代表行的末尾。此功能采用文件字符串指示器。当适当放置在条件声明中时，如果文件指示器已经达到指定文件的行末尾，将能看到它。如果有，那么此功能将返回真值。此任务最好在循环中。

此命令在“编辑”窗口中的语法为：

EOF(<文件指针>) 或 EOL(<文件指针>)

此命令的一些组成部分说明如下：

<filepointer>
正在检查的文件指针的名称。

EOF 和 EOL 的样例代码

注：下面的样本代码应该在“编辑”窗口的命令模式下输入，而不是在**文件输入/输出**对话框中输入。

以下代码可打开 test.txt 并读取文件。只要尚未到达文件的末尾（指定代码 `WHILE/!EOF`），PC-DMIS 即会逐字符读取此文件，并为 V1 分配一个字符。

如果PC-DMIS运行到文件的一个命令行的结尾，PC-DMIS显示该命令行的最后一个字符。

这会重复，直到到达文件结尾。PC-DMIS显示文本“已达到文件末尾……”。

```
FPTR=文件/打开,D:\temp\test.text,读取
WHILE/!EOF("FPTR")
V1=文件/读取_字符,FPTR
IF/EOL("FPTR")
注释/操作者，否，“到达命令行结尾 。最后一个字符是：”
，V1
END_IF/
END_WHILE/
注释/操作者，否，“到达命令行结尾，最后一个字符是...”
```