

目次

フローコントロールを使っでの分岐	1
フロー制御を使用した分岐: はじめに	1
コントロール ペアの使用	2
If / End If	2
Else If / End Else If	4
Else / End Else	6
While / End While	8
Do / Until	10
Select / End Select	12
CASE / END CASE	16
Default Case / End Default Case	17
汎用ループの使用	18
ループの目的	21
ループの作成	21
ループにおける印刷コマンドの使用	22
ラベルの使用	25
GOTOを使用してラベルにジャンプ	26
ループをリセットする	28
エラーにあたっての分岐	29
「On Error」支援のインターフェース	32
サブルーチンでの分岐	33

新しいサブルーチンの作成	34
既存のサブルーチンの編集	39
[引数の編集]ダイアログ ボックスについて	40
サブルーチンの呼び出し	41
サブルーチンの例	49
測定ルーチンの終了	56

フローコントロールを使っ ての分岐

フロー制御を使用した分岐: はじめに

多くフィーチャーのあるパーツをお持ちで、しかし、そのうちの幾つかのフィーチャーを何度も繰り返して測定し、これらのフィーチャーの総合的な一連の統計データを入手したい、とします。ユーザーからの対応次第で、測定ルーチン内の特定の部分にジャンプしたいとします。フローコントロール コマンドを用いて、これらとその他の多くの作業を行うことができます。特定のコマンドの条件のセットアップによって、測定ルーチンのフローをコントロールすることが可能です。

本章はそのようなタスクを実行するために必要な情報を提供します。ここでは構文の条件文、ループ、サブルーチンを説明します。また、多くのコードサンプルも提供します。

この章では、主に以下のトピックについて説明します。

- コントロール ペアの使用
- 汎用ループの使用
- ラベルの使用
- エラー時の分岐
- サブルーチンを用いた分岐
- 測定ルーチンの終了

コメント後の「コマンドモード」コマンド

本章にあるコードサンプルの多くは入力される `COMMENT` コマンドを使用するため、以下のことを考慮してください。



PC-DMISコメントを挿入した後、コマンドモードで追加のPC-DMISコマンドを入力するには、**COMMENT**コマンドの後に最初にEnterキーを2回押す必要があります。これは、ユーザーがコメントにテキストを追加する必要はないが、新しいコマンドを追加する準備ができていることをPC-DMISに通知します。

コントロール ペアの使用

挿入 | フロー制御コマンド | コントロールペアサブメニューには、様々なペアコマンドがあり、編集ウィンドウ内で測定プログラムの適切なフローを制御または「コントロール」します。編集ウィンドウ内にコントロールペアタイプのコマンドを挿入するには、コマンドを入力するか、このサブメニューからコマンドを選択するだけです。



「はい/いいえ」コメントの値をテストするために条件分枝ステートメントを使用する場合、そのテストは大文字の「YES」または「NO」を検索します。小文字の「yes」または「no」では機能しません。コメントについて詳しくは、「レポートコマンドの挿入」章の「プログラマコメントの挿入」トピックを参照して下さい。

If / End If

挿入 | フローコントロール コマンド | コントロール ペア | If / End If メニュー オプションを使用すると、測定プログラムに条件ブロックを追加することができます。IFコマンドとEND IFコマンドの間の項目はIFコマンドの式が真（ゼロ以外）と評価された場合にのみ実行されます。それ以外の場合、実行フローは**END/IF**コマンドの後の最初のコマンドへジャンプします。

IF/ END IF ステートメントの編集ウィンドウのコマンド行は以下です：

IF/式

END_IF/

If / End Ifコマンドを挿入するには、次の手順に従います：

1. 編集ウィンドウ内のご希望の位置に、カーソルを配置して下さい。

コントロール ペアの使用

2. メニュー バーから**If / End If** を選択して下さい。IF / END IF ステートメントが編集ウィンドウに表示されます。

If / End If のコードのサンプル

点要素を測定するか否かをユーザーに尋ねる次の例を考えてみます。

```
C1= COMMENT/YESNO,点フィーチャーPNT1を測定しますか?  
  
IF/C1.INPUT=="YES"  
  
    PNT1=FEAT/POINT,RECT  
  
    ...  
  
    ...  
  
    測定の終了/  
  
END_IF/
```

サンプル コードの説明

C1=COMMENT/YESNO

この行はユーザーからの「はい」または「いいえ」応答を受けて保存します。

IF/C1.INPUT=="YES"

この行は式です。これは、コメント1の入力が、「はい」であるかどうかを検査します。「はい」である場合、IFステートメントは「真」であり、IFステートメント後のステートメントを実行し続け、この場合、**PNT1** 要素が測定されます。「いいえ」の場合、END_IFステートメントに移動します。

END_IF

この行は「IF / END IF」コードブロック内のコマンドの実行を終了します。ユーザーが、コメントについていいえをクリックすると、PC-DMISは、この行に続くコマンドにジャンプします。

Else If / End Else If

挿入 | フローコントロール コマンド | コントロール ペア | Else If / End Else If メニュー オプションを使用すると、測定プログラムに条件ブロックを付け加えることができます。ELSE IFコマンドとEND ELSE IFコマンド間の項目はELSE IFコマンドの式が真（ゼロ以外）と評価された場合にのみ実行されます。「ELSE IF / END ELSE IF」ブロックは、「IF / END IF」または別の「ELSE IF / END ELSE IF」ブロックの直後に位置しなければなりません。その時点でのブロックより上の「IF / END IF」式のすべてが偽と評価された場合、その式が評価されます。式の結果が偽(ゼロ)の場合、END ELSE IF コマンドに続く次のコマンドにジャンプして実行します。現在のブロックより前のIF / ELSE if 式のうち1つでも真と評価された場合、この後に続くすべてのELSE IF / END ELSE IFブロックがスキップされます。

ELSE IF/ END ELSE IF ステートメントの編集ウィンドウのコマンド行は以下です：

ELSE_IF/式

END_ELSE_IF/

ELSE IF / END ELSE IFコマンドを挿入するには:

1. 編集ウィンドウ内で、既存の IF/END IFステートメント、または、ELSE IF/END ELSE IF ステートメントの後の、ご希望の位置にカーソルを配置して下さい。
2. メニュー バーから**Else If / End Else If** を選択して下さい。ELSE IF / END ELSE IF ステートメントが編集ウィンドウに表示されます。




このタイプのブロックは「IF / END IF」または「ELSE IF / END ELSE IF」ブロックの後にあるときにのみ有効です。無効な位置にあるコントロール ペアは、編集ウィンドウ内で赤色のテキストで表示されます。

Else If / End Else If のコードのサンプル

測定された点のX、Y、または、Z値のうちのいずれかが、定義された公差を越える場合、ユーザーにメッセージを表示する、以下の例を参考にして下さい。

コントロール ペアの使用



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

サンプル コードの説明

このコードは、最初のポイントのX値をテストします。条件評価が false だったら、コードはY値にテストします。Y 値の条件評価が false だったら、それはZ 値にテストします。

これらの条件のうち、どれかが真と評価された場合、PC-DMISは、それに関連したコメントを表示し、残りの条件文を省略します。

IF/PNT2.X7.1

この行は式です。これは、測定されたX値が6.9より小さいか7.1より大きいかを検査します。これらの境界の内のどちらかを越える場合には、最初のコメントが実行されます。

END_IF

この行は「IF / END IF」コードブロック内のコマンドの実行を終了します。IF THEN 条件が、偽と評価された場合、PC-DMISは、このラインに続くコマンドにジャンプします。

ELSE_IF/PNT2.Y3.5

この行は最初の ELSE_IF コマンドの式です。これは、測定されたY値が3.3より小さいか、または、3.5より大きいかを検査します。これは、測定されたY値が3.3より小さい

か、または、3.5より大きいかを検査します。これらの境界の内のどちらかを越える場合には、2番目のコメントが実行されます。

END_ELSEIF/

この行は最初の ELSE IF / END ELSE IF コードブロック内のコマンドの実行を終えます。

ELSE_IF/PNT2.Z1.1

この行は二番目の ELSE_IF コマンドの式です。上の ELSE IF / END ELSE IF ブロックが偽と出た場合にのみ、これが実行されます。測定されたZの値が.9より小さいかまたは1.1より大きいかを検査します。これらの境界の内のどちらかを越える場合には、3番目のコメントが実行されます。

END_ELSEIF/

この行は二番目の ELSE IF / END ELSE IF コードブロック内のコマンドの実行を終えます。

Else / End Else

挿入 | フローコントロール コマンド | コントロール ペア | Else / End Elseメニュー オプションを選ぶと、測定ルーチンに、条件ブロックを付け加えることができます。ELSE ブロックの上方にある、その他の「IF / END IF」ブロックと「ELSE IF / END ELSE IF」ブロックすべてが失敗の場合（すべてがゼロと出る）にのみ、ELSEコマンドとEND ELSEコマンドの間の項目は実行されます。「ELSE / END ELSE」ブロックが有効になるためには、一連の「IF / END IF」または「ELSE IF / END ELSE IF」ブロックの末尾に位置しなければなりません。

ELSE / END ELSEステートメントの編集ウィンドウのコマンド行は以下です：

ELSE/

END_ELSE/

「ELSE/ END_ELSE/」 コマンドを挿入するには:

1. 編集ウィンドウ内の希望の位置にカーソルを置きます。「ELSE / END ELSE」ブロックは、「IF / END IF」または「ELSE IF / END ELSE IF」ブロックの後に置かなければならないことに注意して下さい。

コントロール ペアの使用

2. メニュー バーから**Else / End Else** を選択して下さい。ELSE/End ELSE ステートメントが編集ウィンドウに表示されます。

Else / End Else のコードのサンプル

点要素を測定するか否かをユーザーに尋ねる次の例を考えてみます。



```
C1= COMMENT/YESNO,点フィーチャーPNT1を測定しますか? 「いいえ」  
をクリックすると、次のフィーチャーが測定されます。  
IF/C1.INPUT=="YES"  
    PNT1=FEAT/POINT,RECT  
    ...  
    ...  
    ENDMEAS/  
END_IF/  
ELSE  
    PNT2=FEAT/POINT,RECT  
    ...  
    ...  
    ENDMEAS/  
END_ELSE
```

サンプル コードの説明

C1=COMMENT/YESNO

この行はユーザーからの「はい」または「いいえ」応答を受けて保存します。

IF/C1.INPUT=="YES"

この行は式です。これは、コメント1の入力が、「はい」であるかどうかを検査します。「はい」である場合、IFステートメントは「真」であり、IFステートメント後のステートメントを実行し続け、この場合、**PNT1** 要素が測定されます。「いいえ」の場合、END_IFステートメントに移動します。

END_IF

この行は「IF / END IF」コードブロック内のコマンドの実行を終了します。ユーザーが、コメントについていいえをクリックすると、PC-DMISは、この行に続くコマンドにジャンプします。

ELSE

上記の IF / END IF ブロックが偽と評価すると、この行の後にある END_ELSE 行の前にあるコマンド行が実行されます。この例の場合、PNT2が実行されます。

END_ELSE

この行は 2 番目の ELSE / END_ELSE コードブロック内のコマンドの実行を終えます。

While / End While

挿入 | フローコントロール コマンド | コントロール ペア | While / End While メニュー オプションを使用すると、測定プログラムに条件ループを追加することができます。WHILE コマンドと END WHILE コマンドの間の項目はループをアクティブに保つ条件（または式）が満たされなくなるまで、つまり、WHILE ループの式が偽（言い換えれば、ゼロ）と評価されるまでループ内で実行され続けます。WHILE コマンドは測定プログラム内のどこにでも追加できます。式は各ループの冒頭で検査されます。

WHILE / END WHILE ステートメントの編集ウィンドウのコマンド行は以下です：

WHILE / 式

END_WHILE /

While / End While オプションを挿入するには、次の手順に従います：

1. 編集ウィンドウ内のご希望の位置に、カーソルを配置して下さい。
2. メニュー バーから **While / End While** を選択して下さい。WHILE / END WHILE 文が編集ウィンドウに表示されます。

While / End While のコードのサンプル

測定ルーチンのユーザーによって指定された分量に限定して、要素を測定する、以下の例を参考にして下さい。

コントロール ペアの使用

C1=COMMENT/ INPUT, PNT1を何度測定したいですか? 整数のみをタイプ入力して下さい。

ASSIGN/COUNT=0

WHILE/COUNT < C1.INPUT

PNT2=FEAT/POINT, RECT

...

...

...

測定の終了/

ASSIGN/COUNT=COUNT+1

COMMENT/OPER, "測定値 " + COUNT + " のうち " + C1.INPUT
+ " 回 "。

END_WHILE/

サンプル コードの説明

C1=COMMENT/ INPUT

この行は、ユーザーから整数の入力値を受け取り、変数C1.INPUT内に保持します。

ASSIGN/COUNT=0

この行はユーザー定義変数のCOUNTを初期化し、初期値0を与えます。コードはこの変数を使用してPC-DMISがループ内で要素を測定した回数を数えます。

WHILE/COUNT < C1.INPUT

この行は式です。COUNT値（初期設定0）が、ユーザーによって選択された整数以下であるか、検査します。これが真と出るとWHILE/の後にありEND_WHILE/の前にあるステートメントが実行されます。

ASSIGN/COUNT=COUNT+1

この行はCOUNT変数をひとつずつ増分し、それによって、それが条件テストに失敗した時、ループの外に出ます。

```
COMMENT/OPER,"測定値 " + COUNT + " のうち " + C1.INPUT + " 回"。
```

この行は、合計回数のうち、ループが実行中の回数を示すメッセージを表示します。

```
END_WHILE
```

この行は条件が偽である限り、「WHILE / END WHILE」ブロック内のコマンドの実行を終了します。それ以外の場合、PC-DMISがこのコマンドに出会うと、ループはWHILE ステートメントに戻ります。

Do / Until

挿入 | フローコントロール コマンド | コントロール ペア | Do / Until メニュー オプションを選ぶと、測定ルーチンに、条件ループを追加することができます。DOコマンドとUNTILコマンドの間の項目は、UNTILコマンドの式が真（ゼロ以外）と出るまで、ループ内で実行され続けます。DO / UNTILコマンドは、測定ルーチン内のどこにでも追加できます。式は、各ループの最後で検査されます。

DO / UNTILステートメントの編集ウィンドウのコマンド行は以下です：

```
DO/
```

```
UNTIL/ 式
```

DO / UNTILコマンドを挿入するには:

1. 編集ウィンドウ内のご希望の位置に、カーソルを配置して下さい。
2. メニューバーから**Do / Until** を選択します。DO / UNTIL ステートメントが編集ウィンドウ内に表示されます。

Do / Until のコードのサンプル

測定ルーチンのユーザーによって指定された分量に限定して、要素を測定する、以下の例を参考にして下さい。これは、While / End While の項目での例と類似していますが、PC-DMISが、ループの冒頭のかわりに、その末尾で条件をテストする点が異なります。

コントロール ペアの使用



```
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
```

サンプル コードの説明

C1=COMMENT/INPUT

この行は、ユーザーから整数の入力値を受け取り、変数C1.INPUT内に保持します。

ASSIGN/COUNT=0

この行はユーザー定義変数のCOUNTを初期化し、初期値0を与えます。コードはこの変数を使用してPC-DMISがループ内で要素を測定した回数を数えます。

DO/

DO / UNTIL ループを開始します。すべてのステートメントは、少なくとも一度は実行され、式が偽と評価した場合、測定ルーチン フローはループの外に出ます。

ASSIGN/COUNT=COUNT + 1

この行はCOUNT変数をひとつずつ増分し、それによって、それが条件テストに失敗した時、ループの外に出ます。

COMMENT/OPER,"測定値 " + COUNT + " のうち " + C1.INPUT + " 回"。

この行は、合計回数のうち、ループが実行中の回数を示すメッセージを表示します。

UNTIL/COUNT == C1.INPUT

この行は、条件が偽と出た場合に、DO / UNTIL ループ内でのコマンド実行を終了します。それ以外の場合には、PC-DMISがこのコマンドに出会うと、ループはDO ステートメントに戻ります。

Select / End Select

挿入 | フロー制御コマンド | コントロール ペア | **Select / End Select** メニュー オプションを選ぶと、CASE / END CASEとDefault Case / End Default Case ペアとの組み合わせで使用される条件ブロックの追加ができます。Selectコマンドの式はそのCaseステートメントの式と比較されているデータを提供します。この二つの式が同じものと評価した場合、Case / End Case Blockのステートメントが実行されます。「DEFAULT CASE /END DEFAULT CASE」ブロックはCASE / END CASEとDEFAULT CASE / END DEFAULT CASE ブロックのセットを囲みます。

ELECT / END SELECT ステートメントの編集ウィンドウのコマンド行は以下です：

`SELECT/式`

`END_SELECT/`

「SELECT/ END SELECT」コマンドを挿入するには:

1. 編集ウィンドウ内のご希望の位置に、カーソルを配置して下さい。
2. メニューバーから**Select / End Select** を選択します。SELECT / END SELECT ステートメントが編集ウィンドウ内に表示されます。

Select / End Select のコードのサンプル

以下のペア `SELECT / END_SELECT`、`CASE / END_CASE`、`DEFAULT CASE / END_DEFAULT CASE`はすべて一緒に機能し、代替手段を持つ、複数の条件を評価します。

CIR1からCIR5までのラベルの付いた、5つの円があり、キーボード上のキー入力によって、簡単に円が測定できるオペレーターが望まれる、とします。以下のようなコードを使用することができます:

コントロール ペアの使用

コード全体

DO/

C1=COMMENT/INPUT,測定する円の番号をタイプ入力して下さい:

、CIR1には - 1をタイプ入力

、CIR2には - 2をタイプ入力

、CIR3には - 3をタイプ入力

、CIR4には - 4をタイプ入力

、CIR5には - 5をタイプ入力

それ以外の活字の場合、ループを退く

SELECT/C1.INPUT

CASE/1

CIR1=FEAT/CIRCLE

...

...

測定の終了/

END_CASE

CASE/2

CIR2=FEAT/CIRCLE

...

...

測定の終了/

END_CASE

CASE/3



コントロール ペアの使用

```
CIR3=FEAT/CIRCLE

...

...

測定の終了 /

END_CASE

CASE / 4

    CIR4=FEAT/CIRCLE

    ...

    ...

    測定の終了 /

END_CASE

CASE / 5

    CIR5=FEAT/CIRCLE

    ...

    ...

    測定の終了 /

END_CASE

DEFAULT CASE

    COMMENT/OPER, ここでループを退く。

END_DEFAULT CASE

END_SELECT

UNTIL C1.INPUT < 1 OR C1.INPUT > 5
```

サンプル コードの説明

SELECT/C1.INPUT

このコード ラインは、ユーザーによってタイプ入力された番号、または、ストリング 値（この例の場合、番号）を受け取り、その入力内容から、どのCASE/END_CASE ブロックが実行されるか決定します。SELECT / END_SELECT ペアが、このコード一覽全体を囲んでいることに注意して下さい。すべてのCASE / END_CASEとDEFAULT CASE / END_DEFAULT CASEペアが、これら2つの行間に配置される必要があります。

END_SELECT

これは、「SELECT/END_SELECT」 ペア内のコードの終了をマークします。

CASE/1 から CASE/5まで

C1.INPUTの値により、CASE コードのうちの一つのブロックが実行されます。例えば、C1.INPUTが1と評価すると、CASE 1 コードブロックが実行され、CIR1を測定します。2と評価すると、CASE 2 コードブロックが実行され、CIR2を測定する、といった具合です。

END_CASE

この行は、特定のCASEコード ブロックを終了します。

DEFAULT CASE

C1.INPUTの値が、あらかじめ定義されたCASEステートメントのいずれとも一致しない場合には（値が、1から5までの数字でない場合）、DEFAULT CASEコード ブロックが実行されます。この場合、ループから退出することを知らせるメッセージが表示されます。

Do / UNTIL ループが、コード サンプル全体を囲んでいることに注意して下さい。これを用いると、ユーザーは、CASEステートメントが認識しない文字を選択するまで、COMMENT/INPUTラインから作成されたメニューから選択し続けることができます。

CASE / END CASE

挿入 | フローコントロール コマンド | コントロール ペア | CASE / END CASEメニュー オプションを選ぶと、測定ルーチンに、条件ブロックを付け加えることができます。

コントロール ペアの使用

CASEステートメント用の入力式が、それに対応するSELECTコマンドの入力式と等しい数値を出す場合、CASEコマンドとEND CASEコマンドの間にある項目が実行されます。それ以外の場合には、そのブロックのステートメントは省略されます。CASE / END CASEステートメント ブロックは、SELECTコマンド、あるいは、その前のCASE / END CASEブロックのEND CASEコマンドの直後に置かれる必要があります。また、PC-DMISは、単一の場合ステートメント上で、複数の式を比較することはできません。

CASE / END CASEステートメントの編集ウィンドウでのコマンド行は下記です:

CASE/expression

END_CASE/

「CASE / END CASE」 オプションを挿入するには:

1. 編集ウィンドウ内のご希望の位置に、カーソルを配置して下さい。上記に述べられた、位置付けに関する必要条件に注意して下さい。
2. メニューバーから**Case / End Case**を選択します。CASE / End Case ステートメントが編集ウィンドウ内に表示されます。

Default Case / End Default Case

挿入 | フローコントロール コマンド | コントロール ペア | Default Case / End Default Caseメニュー オプションを選ぶと、測定ルーチンに、条件ブロックを付け加えることができます。「SELECT / END SELECT」ブロック内の、先行する「CASE / END CASE」ブロックの、その他すべての式が、偽と出た場合に、「DEFAULT CASE」と「END DEFAULT CASE」の間の項目は、実行されます。「SELECT / END SELECT」ブロック内では、ひとつの「DEFAULT CASE / END DEFAULT CASE」ブロックのみが存在できます。「DEFAULT CASE / END DEFAULT CASE」ブロックは、「SELECT / END SELECT」ブロック内のすべての「CASE / END CASE」ブロックの後に置かれる必要があります。

DEFAULT CASE / END DEFAULT ステートメントの編集ウィンドウでのコマンド行は下記です:

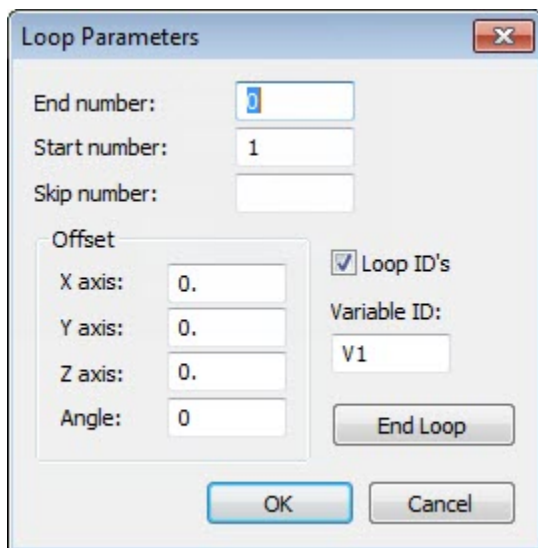
DEFAULT CASE/

END_DEFAULT_CASE/

「DEFAULT CASE / END DEFAULT CASE」コマンドを挿入するには:

1. 上記に記述された位置付けの制限に注意して、編集ウィンドウ内のご希望の位置に、カーソルを配置して下さい。
2. メニューバーから**Default Case / End Default Case**を選択します。DEFAULT CASE / End DEFAULT CASE ステートメントが編集ウィンドウ内に表示されます。

汎用ループの使用



[ループパラメータ] ダイアログボックス

挿入|フロー制御コマンド|ループメニューオプションは、ループ・パラメータダイアログボックスが表示されます。オフセットのどれで、あるいはどれもなしでも測定ルーチン(あるいは測定ルーチンの一部)を繰り返すLOOPコマンドを作成するには、このダイアログ・ボックスを使用することができます。LOOPコマンドは測定ルーチンのどこにでも追加できますが、この機能は測定ルーチンの始めと終わりで最も役に立ちます。

開始番号 - このボックスは、一連のパーツのなかの開始位置番号をPC-DMISに指示します。



10個のパートがあり、位置番号 5 から開始したいとします。この場合、パート総数として10、開始位置として5を入力します。

終了番号 - このボックスは、測定ルーチンによって何回ループするべきであるかをPC-DMISに伝えます。この数は、通常、取付具が X (Y、または、Z) 方向に持つパーツの総数 (または、パーツのパターン数) と同じ数です。PC-DMISはまた、開始のパーツ (パターン) 番号についても尋ねます。



X方向に10個のパートがあり、位置番号5から開始したいとします。**終了番号**ボックスには、10を入力します。**開始番号**ボックスには5を入力します。

スキップ数 - ループでは、PC-DMIS測定ルーチンを何回表示された数を繰り返します。**スキップ番号**ボックスを使用すると指定した増分をスキップすることができます。このボックスを使用して**最初の増分**をスキップすることはできませんが、**開始番号**を2に変更することで同じ効果でスキップすることができます。



ループが3増えるごとにスキップするパラメータを設定したいとします。数字3を指示すると、PC-DMISは最初と2番目のパートを測定してから、スキップして4番目のパートに進みます。

オフセット - オフセット エリアには、以下のボックスがあります:

X軸、Y軸、Z軸 - これらのボックスは、パーツ間のX、Y、およびZオフセット距離、または同じパーツ上のパターンを定義します。ループが実行されると、これらの値は、定義されたオフセット距離だけ部品をオフセットします。

角度 - このボックスは、パーツ間、あるいは、同一パーツ上のパターン間の角オフセットをセットアップします。最初のオフセットは、パーツの原点に基づいています。実行がループすると、PC-DMISはパーツを角度値でオフセットします。

ソフトウェアは、オフセット距離をパーツの原点に基づいています。最初のループでは、PC-DMISはパーツの原点を使用します。ループを2回目に通過すると、オフセット距離だけ移動します。3回目のループでは、オフセット距離に2を掛けた数だけ移動し、以下同様に続きます。



ループ内部にアライメントコマンドがあり、ループがオフセットを使用する場合、そのアライメントのためのすべての軸を定義しなければなりません。さらに、ループ内部のアライメントは、ループ内部で測定される要素を使用しなければなりません。

ループID - このチェックボックスは、PC-DMISが要素IDの角括弧内に現在のループ値を表示するかどうかを決定します。



CIR1の円要素があるとして。ループを最初に通過するとCIR1[1]になり、ループを2回目に通過するとCIR1[2]になります。

統計データベース内の要素ID

ユーザが、**ループID**のチェックボックスを選択すれば、いくつかのケースでは、データベースに統計データを送信する時に、PC-DMISは、データベースにこれらのループIDを表示しない場合があります。

手順は以下のとおり:

ループブロック内部に、**STATS/ON** コマンドと**STATS/UPDATE** コマンドがある場合、ループIDはデータベース内部に表示されません。

ループブロック外に**STATS/ON** コマンド、ループブロック内に**STATS/UPDATE** コマンドがある場合、ループIDはデータベース内に表示されません。

ループブロックの外に**STATS/ON** コマンド、ループブロックの外に**STATS/UPDATE** コマンドがある場合、ループIDはそのデータベース内に表示されません。

ループブロックの中に**STATS/ON** コマンド、ループブロックの外に**STATS/UPDATE** コマンドがある場合、ループIDはそのデータベース内に表示されません。

変数ID - このボックスを用いると、ループのその時点での反復（または、特定されたループ数のうちのその時点でのループ）を追跡するために使用する変数名を定義することができます。測定ルーチン実行中、この変数は、ループのその時点での反復数を同じ数を表示します。

End Loop - このボタンを押すと、ループのプロセスが終了します。編集ウィンドウで、[LOOP/START](#)コマンドに続いて[LOOP/END](#)コマンドを実行する必要があります。

ループの目的

ループ機能には主に3つの用途があります：

- パーツの格子を持つ、複数パーツ用の取付具があるとします。その取付具は、横列間の間隔として一貫した空間を必要とします。位置変換 / 回転オフセットを用いると、パーツの格子内で、ひとつのパーツから次のパーツへと指示することができます。
- パーツをひとつ持つ取付具があり、測定ルーチンの各ループの前に、取付具の持つパーツを新規のものに交換したい、とします。新規のパーツに置き換える時には、[COMMENT](#)コマンドを用いてCMMを停止することができます。そのコマンドは、ループの冒頭、または、末尾のいずれにも置くことが可能です。
- ループ機能を使用して、同一のパーツの別の部分を測定するために、測定ルーチンを回転することができます。例えば、パーツ上で10回繰り返されている、複雑な穴のパターンを測定するための測定ルーチンを作成するとします。ユーザの測定ルーチンは、ホールパターンの一つを測定する必要があります。その後、ループ機能を使用して測定ルーチンをオフセットし、パターンの他の9つの出現を測定できます。



ループ内部でパーツ配置を使用する場合、前回保存したアラインメントを常に呼び出すかわりに、PC-DMIS3.6以降では、[ALIGNMENT/START](#)コマンドラインのアクティブなアラインメントを使用することができます。「アラインメントの作成と使用」の章にある「ループ内でアラインメントを使用」の項目を参照して下さい。

ループの作成

1. メニューバーから**挿入 | フローコントロール コマンド | ループ**を選択すると、[ループ用パラメータ](#) ダイアログ ボックス が表示されます。
2. ダイアログボックスで値を定義します。
3. 必要に応じてパラメーターを選択します（**パーツ数、開始番号、スキップ番号、オフセット角度**など）。

4. 編集ウィンドウ内の、ループ開始予定の位置に、カーソルを配置して下さい。
5. [OK]ボタンをクリックして、**LOOP/START**コマンドブロックを挿入します。

ループ用の編集ウィンドウのコマンドラインは、以下のようです:



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



ループ手順を完了するには、**LOOP/END**コマンドでループを終了する必要があります。PC-DMISは**LOOP / START**および**LOOP / END**コマンドの間にあるすべての編集ウィンドウコマンドをループします。

LOOP/END コマンドは次のいずれかの方法で挿入できます：

- [編集]ウィンドウで、**LOOP/END**と入力します。
- 挿入|フロー制御コマンド|ループの終了メニュー項目を選択します。
- [ループパラメータ]ダイアログボックスで、[ループの終了]ボタンをクリックします。

ループにおける印刷コマンドの使用

PC-DMISでのループにおいて**PRINT**コマンドを使用するには以下を行います

1. **LOOP/START**および**LOOP/END**コマンドを測定ルーチンに挿入します。
2. **PRINT**コマンドをループコマンドの間の挿入したい位置に挿入します。
3. 編集ウィンドウで**PRINT**コマンドをクリックします。次に、F9キーを押して、その**PRINT**コマンドに対する出力設定ダイアログボックスを表示します。レポートウィンドウの出力およびプリンターオプションの設定方法について詳しくは、「基本的なファイルオプションの使用」章にある「レポートウィンドウの出力および印刷オプションの設定」トピックを参照してください。
4. 出力フォーマットタブの1つを選択して (レポート、**DMIS**または**エクセル**)、**PRINT**コマンドのフォーマットを定義します。

汎用ループの使用

5. タブページの左上にあるチェックボックスを選択して、出力のオプションを有効にします。
6. 生成されたレポートの送信先を定義する
 - a. パスおよびファイル名ボックスの右側にある参照 (...) ボタンをクリックして、名前を付けて保存ダイアログボックスを表示します。
 - b. 名前を付けて保存ダイアログボックスでファイル名を定義して、保存をクリックします。
7. 必要なその他の変更を行います。例えば、PC-DMISが各レポートを生成した後に、それらを表示したい場合、**レポートの表示**チェックボックスを選択します。
8. 複数のフォーマットで出力を生成したい場合は、[次へ] タブを押してステップ4～7を繰り返します。
9. **OK**ボタンをクリックしてPRINTコマンドに対する変更を保存します。

PRINTコマンドはループ外側で機能するのと殆ど同様にループ内側で機能します。その違いは、内部的にPC-DMISが出力された内容を追跡し、最後のPRINTコマンドを実行した後に実行されたデータのみを出力することです。



例えば、下記の短いコードについて考えてみましょう。PC-DMISが初めて `PRINT` コマンドを実行すると、ソフトウェアはそのポイントまでのすべての収集されたデータから成る最初の出力ファイルを作成します。これは、最初の出力ファイルには `LOOP/START` コマンドの前に収集されたデータも含まれることを意味します。次回 `PRINT` コマンドが実行されると、PC-DMISはループ内部の内容だけを実行します。これは、2番目のデータファイルはループ内部ならのデータのみから成ることを意味します。これは測定ルーチンが終了するまで繰り返されます。

・
・
・

ループの開始点

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

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

移動点

移動点

移動点

移動点

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

移動点

移動点

移動点

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

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

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

ラベルの使用

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

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

```
LOC1合格: CIR1
```

レポートの印刷

ループの終了点

・
・
・

ラベルの使用

実行フローは、特定の条件に基づいて送信される場所を制御するために[GOTO](#)または[IF_GOTO](#)コマンドで[LABEL](#)コマンドを使用することができます。ラベル名はスペースを持つことができず、230文字を超過してはなりません。PC-DMISはすべて頭文字を使用して、ラベル名を表示します。

[LABEL](#)コマンドを作成するためには、下記のうちの1つを行ってください:

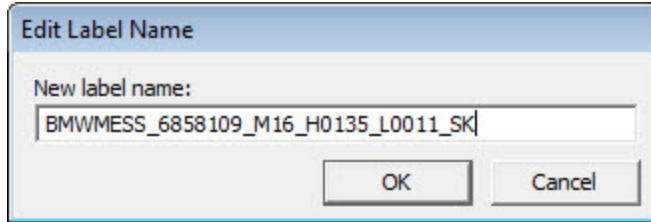
コマンドを入力します。

1. 編集ウィンドウのブランク行にカーソルを置いてください。
2. LABELをタイプし、次にタブ・キーを押します。
3. 強調されたエリアで、新規ラベル名をタイプしてください。

または

コマンドを選択する

1. 編集ウィンドウにカーソルを置いてください。
2. メニューバーから、**挿入| フロー制御コマンド| ラベル**を選び、**ラベル名の編集**ダイアログボックスにアクセスします。
3. **新規ラベル名**ボックスに、ラベルのIDを入力して下さい。



[ラベル名編集] ダイアログボックス

4. **OK**をクリックすると編集ウインドウの次の可能な位置にラベル名が挿入されます。

編集ウインドウのコマンドモードにおけるLABELコマンドは
`ID = LABEL/` と記述されます

ここでIDは、指定されたラベルの名前を表わします。

ラベルを備えたGOTOコマンドの使用に関する情報については、「GOTOを使用するラベルにジャンプする」トピックを参照してください。

ラベルを備えた IF_GOTOコマンドの使用に関する情報については、「GOTOを使用するラベルにジャンプする」トピックを参照してください。

GOTOを使用してラベルにジャンプ

GOTOコマンドの作成により特定のラベルにジャンプするように測定ルーチンに伝えることができます。

GOTOコマンドを作成するためには、下記のうちの1つを行ってください:

コマンドを入力します。

1. 編集ウインドウの空白行にカーソルを置いてください。
2. GOTOをタイプし、次にタブ・キーを押します。
3. ジャンプ先のラベル名を入力します。

または

コマンドを選択する

1. 編集ウィンドウにカーソルを置いてください。
2. メニューオプション [挿入 | フロー制御コマンド | Goto] を選択して、[GoTo] ダイアログボックスを開きます。



[GoTo] ダイアログボックス

3. ラベルが測定ルーチンの中にすでに存在する場合、それらが**現在のラベル**ボックスに現れます。
4. [**現在のラベル**] ボックスから、希望のラベルを選択するか、[**GoTo ラベル**] ボックスにラベル名を入力します。
5. **OK**をクリックすると編集ウィンドウの次の可能な位置にGOTOコマンドが挿入されます。

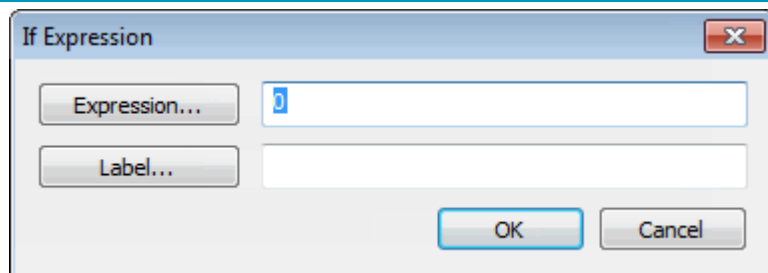
編集ウィンドウのコマンドモードのGOTOコマンドは次のように記述されます:

`GOTO/label_ID`

ここでlabel_IDは、指定されたラベルの名前を表わします。ラベル名はスペースを持つことができず、230文字を超過してはなりません。



ラベルが未作成の場合、ラベル名は編集ウィンドウのコマンドモードの**赤**で強調表示されます。また、そのGOTOコマンドは実行中、無視されます。



[If 式] ダイアログボックス

「挿入 | フロー制御コマンド | If Goto」オプションを使用すると、測定ルーチンの内に「IF GOTO」文をつくるために「If Expression」ダイアログボックスを使うことができます。測定ルーチンの実行中に、PC-DMISがIF GOTOステートメントに出会うと、特定の式がゼロ以外の値を出した場合に、ルーチンのフローはラベルIDの所在位置に指します。

IF_GOTO ステートメントの編集ウィンドウのコマンド行は以下です：

IF_GOTO/式, GOTO=Label

式 - 式をクリックすると、式ビルダが開きます。式ビルダを用いると、測定ルーチンに必要な様々な式を作成できます。式が作成されたら、それは[式]ボックスに表示されます。式の作成に関する情報については、「式及び変数の使用」の章を参照して下さい。

ラベル - ラベルをクリックされると、**Goto**のダイアログボックスが表示されます。**Goto**ダイアログボックスで、定義された式が満たされたときにルーチンの流れはどのラベルへ移動するかを選択することができます。選択されたラベルは、[ラベル]ボックスに表示されます。あるいは、ラベル名を知っていれば、ボックスにそれをタイプすることができます。ラベルについての詳細は、「ラベルにジャンプする」を参照してください。

ループをリセットする

Reset Loop コマンドはループ内の実行ツリーをリセットします。Reset Loop コマンドを追加し、以下のコマンド対を使って複数のパート測定をサポートすることができます：

エラーにあたっての分枝

- While/End While
- Do/Until
- Loop/End Loop

以上のループの一つの内部で Reset Loop コマンドを使用する必要があります。Reset Loop コマンドはループの外側に置くと効果がありません。Reset Loop コマンドを挿入するには、[挿入 | フロー制御コマンド | Reset Loop] メニューオプションを選択します。Reset Loop コマンドに対する編集ウィンドウのコマンド行は **RESET LOOP/** です。

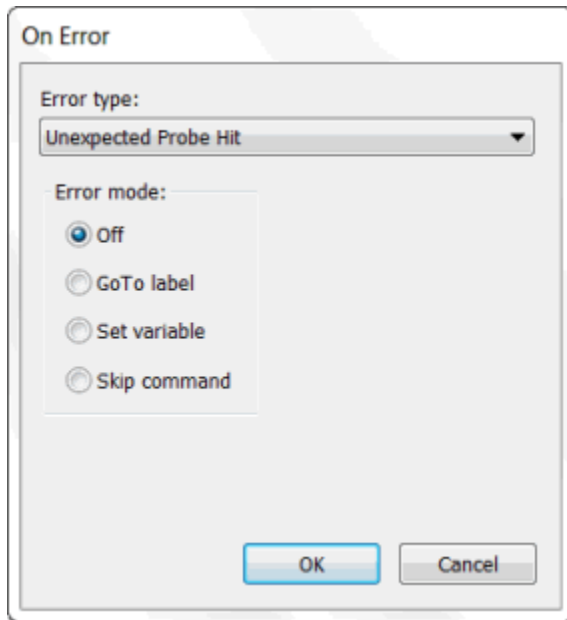
複数パート測定では、ループで測定された最後のパートのレポートを取得できます。これを行うには、下記の画像に示すとおりループの開始点に Reset Loop コマンドを挿入します。

```
WHILE/COUNTER<=3
  RESET LOOP/
  COMMENT/YES, NO, FULL SCREEN=YES, AUTO-CONTINUE=YES, TIME DELAY=1, QVO=NO,
  Wait!
  F1
  --CIRCLE/CIRCLE, DEPENDENT, CARTESIAN, OUT, S
  HORIZ/XY1, <0, 0, 0>, S
  HORIZ/XY2, <0, 0, 0>, S
  HORIZ/XY3, <0, 0, 1>, S
  HORIZ/XY4, <0, 0, 1>, S
  DIAMETER/2, COUNTER
  DIM LOC1= LOCATION OF CIRCLE F1 UNITS=MM, S
  GRAPH=OFF TEXT=OFF MOLT=10.00 OUTPUT=BOTH HALF ANGLE=NO
  AX HORIZAL, <TOL, <TOL, HEAD, DEV, OUTTOL
  D 3.000 0.000 0.000 3.000 0.000 ----#----
  END OF DIMENSION LOC1
  PRINT/REPORT, EXEC MODE=END, S
  TO FILE=ON, AUTO=4, OUTPUT FORMAT/TXT, S
  REPORT REPORT=YES, AUTO OPEN=OFF, S
  TO PRINTER=OFF, COPIES=1, S
  TO DIM REPORT=ON, FILE OPTION=TRUEX, FILENAME=C:\Users\shuchanbu.Trive
  REPORT THEORETICAL=NO, REPORT FEATURE WITH DIMENSIONS=NO, S
  TO EXCEL=ON, AUTO=9, FILENAME=C:\Users\shuchanbu.Trive\downloads\Whil
  PREVIOUS SUM=DELETE, INSTANCES
  EXCEL FORM1 =EXCELFORMREPORT/TEMPLATE=Form Row View, TYPE=EXCEL
  RESULT FOLDER/C:\Users\Public\Documents\Usageage\PC\DMIS\3024.2\ExcelFor
  ASUM/COUNTER=COUNTER+1
  END WHILE/
  END OF MEASUREMENT FOR
```

Reset Loop は実行ツリーからすべてのオブジェクトを削除します。

エラーにあたっての分枝

挿入| フロー制御コマンド| エラー時」オプションは「エラー時」ダイアログボックスを開きます。



「エラー時」ダイアログ ボックス

このダイアログボックスを使用して、機械のエラーが発生したときに実行するアクションをPC-DMISに通知することができます。

エラータイプ - PC-DMISは以下のエラー条件を追跡します。

- 予期しないプローブ取込み点
- 行われなかったプローブ取込み点
- リフレクタが見つからない - これはPC-DMISポータブルにおいてトラッカと一緒に使用されます。
- レーザーエラー - レーザー構成を使用している場合、PC-DMIS Laserドキュメントの「オンエラーコマンドでのレーザーセンサーエラーの処理」トピックを参照してください。
- 制限範囲外の温度 - X軸、Y軸またはZ軸スケールまたはパートに対する温度の1つまたは複数温度補償コマンドで定義される上限を超えるか、下限を下回る場合、測定プログラム内の温度補償コマンドがこのエラーを発生させます。



測定プログラムにおいてON ERRコマンドは温度補償コマンドの上に置く必要があります。

- 制限範囲外 (限界外) の校正 - 1つまたは複数の校正エラー確認が制限範囲外にある場合、測定ルーチンにおける構成限界の確認コマンドがこのエラーを発生させます。校正限界の確認コマンドについて詳しくは、PC-DMISツールキットモジュールドキュメントの「校正限界の確認について」を参照してください。



測定ルーチンにおいてオンエラーコマンドは校正限界の確認コマンドの上に置く必要があります。

エラーモード - PC-DMISはエラータイプに応じて、下記の操作を実行することができます。

- オフ: PC-DMISは何もしません。
- **GoTo ラベル** - 測定ルーチンのフローは定義されたラベルに移動します (「ラベルの使用」を参照)。下記のオプションが使用可能になります。
 - **ラベルID** - まだ存在しないラベルへの参照を入力します。
 - **現在のラベル** - 測定ルーチン内のすべてのラベルを一覧表示します。
- **変数の設定** - 変数の値を1に設定します。
- **コマンドをスキップする** - 測定ルーチンはその時点でのコマンドを省略し、測定ルーチン内にマークされた次のコマンドに移動します。



PC-DMISが実行中にスキップされた要素を使用する寸法をレポートする方法の詳細については、このドキュメントの「実行中にスキップされた要素を使用した寸法のレポート」トピックを参照してください。

デフォルトでは、すべての測定ルーチンが**オフ** (PC-DMISは何もしません) に設定された両タイプのエラーに対するアクションを開始します。各エラーの動作モードは測定ルーチンを通して変更することができます。



実行中、PC-DMISがON ERROR/UNEXPECTED HIT/JUMPTOLABEL（オンエラー / 予期せぬ取込み点 / ラベルにジャンプ）コマンドに遭遇すると、測定プログラムにおけるその点の後で発生する予期せぬ任意の取込み点によって、実行が指定されたラベルに飛びます。「変数の値を1に設定」に対する動作によって、指定されたエラータイプが発生するとすぐに変数が設定されます。測定プログラムの新しい点に実行をジャンプさせるIFステートメントを使用して、変数のこの値をテストできます。

On Errorコマンドの使用

1. 挿入|フロー制御コマンド|オン・エラー」を選択して、「オン・エラー」ダイアログ・ボックスを開きます。
2. エラータイプリストから、分岐の条件としてエラーの種類を選択します。
3. エラーモードエリアで、オプションを選択して、条件が満たされるときに発生する動作を定義します。
4. On Errorコマンドを追加するにはOKをクリックします。変更を適用せずにダイアログボックスを閉じるには取り消しをクリックします。

「On Error」支援のインターフェース

On Errorコマンドを支援するインターフェースもあります。インターフェースがコマンドを支援しているかどうかを確認するには、次の表を参照してください。

- インターフェースがテーブル内にある場合、小さい黒いボックスは、インターフェースが支援するエラーの種類を示します。
- インターフェースがテーブルにない場合は、On Errorコマンドを使用できません。

サポートされるインターフェース	予期しないプローブ取込点	省略されたプローブヒット	反射装置が発見されず
DEA	-	■	-
FDC	■	■	-
I++ DME Client	■	■	-

ヨハンセン	■	■	-
ライカ追跡機	-	-	■
ライツ	■	■	-
LK Direct (別名 LKRS232)	■	■	-
LKトライバ	■	■	-
Mora	■	■	-
シャープ	■	■	-
Sheffield	■	■	-
Wenzel	■	■	-
Zeiss	■	■	-

サブルーチンでの分枝

サブルーチンは、お客様の測定ルーチン内や、外部測定ルーチン内の、コードのブロックであり、通常繰り返して参照され、より簡潔なプログラム作成が可能になります。PC-DMISでは、サブルーチン内の「引数」（または、ローカル変数）に情報を渡すことができます。サブルーチンに引き渡すことが可能な引数のタイプは、数値、変数、テキスト文字列と要素名です。



サブルーチンコマンドブロックはSUBROUTINEコマンドとENDSUBコマンドの間にあります。

測定ルーチン内にサブルーチンを作成した後、それを、その時点での測定ルーチンや別の測定ルーチンから「呼び出す」ことができ、それを行うと、測定ルーチンの実行フローが特定のサブルーチンに入り込み、サブルーチン コマンド ブロック内のコマンドを実行します。測定ルーチンのフローは、その後、呼び出しステートメントの直後に続くステートメントへと戻ります。



サブルーチンはCALLSUBコマンドを用いて呼び出されます。

外部サブルーチン

外部サブルーチン、または、呼び出し測定ルーチンの外部にある測定ルーチンのサブルーチンは、呼び出し測定ルーチンにある要素、変数、または、アラインメントにアクセスすることはできません。サブルーチンは、それ自身の測定ルーチン内の項目にはアクセスできます。外部測定ルーチンと呼び出し測定ルーチンは、同じ測定単位を使用する必要があります。

サブルーチンのネスト化

他のサブルーチン内でサブルーチンをネストできます。引数とネストされるサブルーチンの数はメモリの空き容量のみによって制限されます。

新しいサブルーチンの作成

SUBROUTINE入力によるサブルーチンの作成

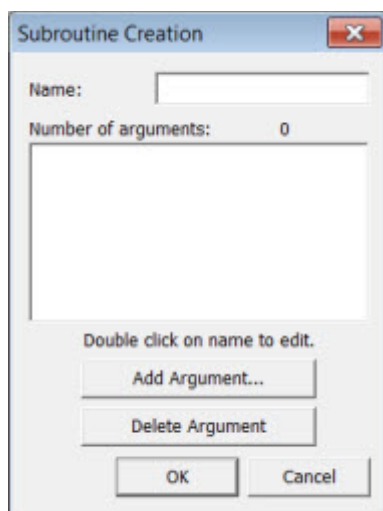
編集ウィンドウのコマンドモードでSUBROUTINEと入力して、このコマンドを挿入することができます。コマンドが挿入されると、そのサブルーチン名と引数を特定する必要があります。これについては以下のサブルーチンの構文および例を参照してください。

ENDSUBコマンドを入力することとTABを押すことで コマンド ブロックを終了します。このコマンドブロックの中に置任意の編集ウィンドウのコマンドも、サブルーチンの一部であると考えられて、サブルーチンが呼ばれるとき、実行されます。

サブルーチン メニュー項目を使用したサブルーチンの作成

1. サブメニューから**挿入 | フローコントロール コマンド | サブルーチンオプション**を選択して下さい。これによって、**[サブルーチン作成]**ダイアログ ボックスが現れます。このダイアログ ボックスに関する情報については「サブルーチン作成ダイアログ ボックスについて」の項目を参照して下さい。

サブルーチンでの分枝

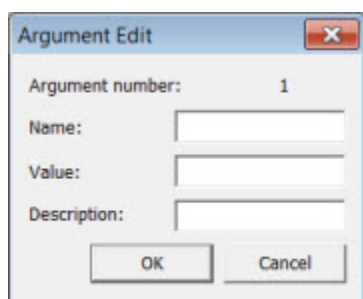


[サブルーチン作成] ダイアログボックス

2. **名称**ボックス内に、サブルーチンの名称を入力して下さい。名前の長さは180文字以内に制限されています。181文字以上入力すると、「行が長すぎます」のエラーメッセージが現れます。

OKボタンを選択すると、その名前が最初の180文字に自動的に短縮され、**サブルーチン作成**ダイアログボックスが閉じます。

3. サブルーチンが引数(サブルーチン内に引き渡された情報のプレースホルダ)を使用する場合、**[引数を追加]**ボタンをクリックして、それらをひとつずつ追加します。**引数の編集**ダイアログボックスが現れます。このダイアログボックスに関する情報については、「**引数の編集ダイアログボックスについて**」の項目を参照して下さい。



[引数編集] ダイアログボックス

4. **名前**ボックス内に、引数の名前を入力して下さい。

5. 数値ボックス内に、引数のデフォルト設定値を入力して下さい。CALLSUBステートメントからサブルーチンに何の値も渡されない場合、サブルーチンはそのデフォルト設定値を使用します。有効な引数の値は、数値、変数、文字列、および要素名です。
6. 引数に説明を付けたい場合は、説明ボックス内に入力して下さい。
7. 引数を作成するために、引数の編集ダイアログ ボックスのOKをクリックして下さい。
8. サブルーチンに含めたい引数それぞれについて、ステップ3から7までを繰り返して下さい。
9. サブルーチン作成を完了するには、サブルーチン作成ダイアログ ボックスのOK ボタンをクリックして下さい。このサブルーチンが、あらかじめ定義された引数すべてと共に、編集ウィンドウ内に表示されます。
10. 挿入 | フローコントロール コマンド | サブルーチン終了メニューオプションを選択して、サブルーチンを終了します。編集ウィンドウに「ENDSUB/」コマンドが配置され、サブルーチンコマンド ブロックを終了します。その他の測定ルーチンのコマンドをサブルーチンに含めたい場合、このサブルーチンのコマンド ブロック内でENDSUBコマンドの前に加える必要があります。

サブルーチン コマンド ブロック用のシンタックス

サブルーチンコマンドブロックのサンプル用の編集ウィンドウのコマンドライン シンタックスは、以下のようになります:



```
SUBROUTINE/ <Name> ,
<A1> = <Arg1> : <Description> ,
<A2> = <Arg2> : <Description> ,
=
<Commands>
ENDSUB/
```

SUBROUTINE/ は、サブルーチンのコマンド ブロックを開始するために使用されるコマンドです。

<名前> = サブルーチン名。名前の長さは256文字以内に制限されています。257文字以上を入力した場合、名前は自動的に256文字に短縮されます。

<A1> = サブルーチン内で使用される最初の引数（あるいは、ローカル変数）。この変数は、一般的に、サブルーチンの外側からはアクセスできません。

<A2> = サブルーチン内で使用される2番目の引数。この変数は、一般的に、サブルーチンの外側からはアクセスできません。必要に応じて、追加の引数を追加できます。

<引数1> = 最初の引数のデフォルト値。

<引数2> = 2番目の引数のデフォルト値。

サブルーチンでの分枝

<説明> = 引数の内容説明。

<コマンド> = その他の編集ウィンドウ用のコマンドを、必要に応じて、引数の後、ENDSUBコマンドの前に挿入することが可能です。

ENDSUB/は、サブルーチンのコマンドブロックを終了するために使用されるコマンドです。

サブルーチン コマンド ブロックの例

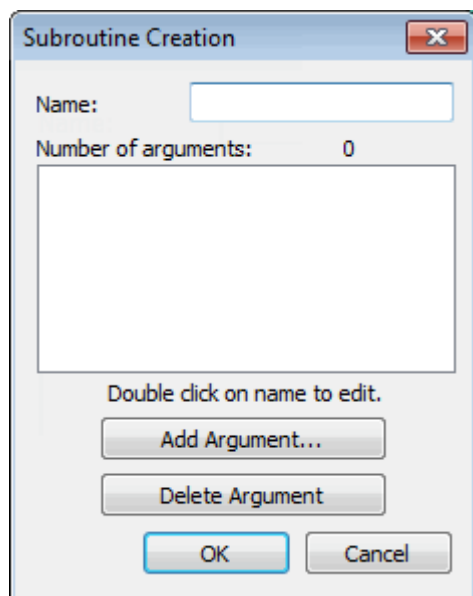
例えば、オペレーターのデータを受け取り、それをレポートに表示する、サブルーチンの完了したものは、以下のようになります：



```
SUBROUTINE/GET_OPERATOR_INFO,  
    OPNAME = <Operator> : OPERATOR NAME,  
    SHIFT = <Shift> : SHIFT TIME,  
    =  
    COMMENT/REPT,OPNAME  
    COMMENT/REPT,SHIFT  
ENDSUB/
```

[サブルーチンの作成]ダイアログ ボックスについて

サブルーチンの作成ダイアログボックスにアクセスするには、挿入 |フロー制御コマンド|サブルーチンを選択します。



[サブルーチン作成] ダイアログボックス

下記のオプションがサブルーチンの作成ダイアログボックスで使用できます。

名称 ボックスはサブルーチンを定義します。これは、後でサブルーチンを呼び出すときに使用する名前です。1つの測定ルーチンに複数のサブルーチンがある場合、それぞれの名前は一意でなければなりません。

名前の長さは180文字以内に制限されています。名称ボックスに181文字以上入力すると、「行が長すぎます」のエラーメッセージが現れます。

OKボタンを選択すると、その名前が最初の180文字に自動的に短縮され、サブルーチン作成ダイアログボックスが閉じます。

引数の番号 この一覧には、作成中のサブルーチンの引数が表示されます。このエリアでは、引数が以下の形式で表示されます:

<名称> =<数値> :<内容説明>

例えば、引数うちのひとつが「直径」という名称で、そのデフォルト設定値が3である場合、この一覧内で、引数は以下のように表示されます:



直径 = 3 : 穴の直径

CALLSUBコマンドから渡される、別の値がない時には、PC-DMISは常にデフォルト設定値を使用します。

引数を編集するには、変更したい引数をダブルクリックします。**引数の編集**ダイアログ ボックスが開き、それを用いて、必要な変更を行うことができます。このダイアログ ボックスに関する情報については、「**引数の編集ダイアログ ボックスについて**」の項目を参照して下さい。

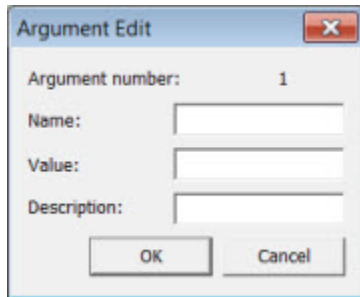
引数を追加 - このボタンは、サブルーチンに新規の引数を追加します。このボタンをクリックすると**引数を編集**ダイアログ ボックスが現れます。このダイアログ ボックスに関する情報については、「**引数の編集ダイアログ ボックスについて**」の項目を参照して下さい。

引数を削除 - このボタンを用いると、サブルーチンから引数を削除することができます。一覧からその引数を選択し、**引数を削除**ボタンをクリックして、それを削除します。

既存のサブルーチンの編集

既存のサブルーチンを編集するには、常に編集ウィンドウのコマンド モードを用いて、直接そのサブルーチンを編集します。その代わりに、編集ウィンドウ内のサブルーチン上にカーソルを置き、F9を押して [**サブルーチン作成**] ダイアログ ボックスにアクセスすることもできます。これによって、[**サブルーチン作成**]ダイアログボックスが表示されます。このダイアログボックスについては「**サブルーチン作成ダイアログボックスの理解**」トピックを参照して下さい。

[引数の編集]ダイアログ ボックスについて



[引数編集] ダイアログボックス

サブルーチン作成ダイアログ ボックス(挿入 | フロー制御コマンド | サブルーチン)、または、サブルーチンの呼び出しダイアログ ボックス(挿入 | フロー制御コマンド | サブルーチンの呼び出し)内で、引数の作成、または、編集を選択するたびに、引数編集ダイアログ ボックスが表示されます。

引数編集ダイアログ ボックスは、以下の2つの状況下で使用が可能です:

- **SUBROUTINE** コマンド ブロック内で、サブルーチンの引数、及び、それらのデフォルト設定値を定義するため。
- **CALLSUB** コマンドからサブルーチン内に渡される数値を定義するには、

次のオプションは引数編集ダイアログ・ボックスにおいて利用可能です:

名前 - このボックスには、作成か編集している引数の名前を定義します。

値 - このボックスには、引数の値を定義します。

SUBROUTINE コマンドを作成、または、編集する場合、これが、**CALLSUB** ステートメントからサブルーチンに何の値も渡されない場合に使用されるデフォルト設定値となります。

CALLSUB コマンドを作成、または、編集する場合、これが、サブルーチンに渡される値となります。

その場合の有効な値は:

- 数値
- 変数

サブルーチンでの分枝

- テキスト文字列- テキスト文字列は、二重引用符で囲まれていなければなりません。
- フィーチャー名 - 例えば、{F1} のように、フィーチャー名は、波状カッコで囲まれていなければなりません。

説明 - サブルーチンの引数の説明を定義します。この内容説明は、編集ウィンドウ内の **SUBROUTINE** コマンド ブロックにおいて、引数の横に表示されます。

サブルーチンの呼び出し

サブルーチンを呼び出すには、**CALLSUB** コマンドをお客様の測定ルーチン内に挿入する必要があります、それによって、その時点での測定ルーチンから既存のサブルーチンを呼び出すか、または、外部測定ルーチンからサブルーチンを呼び出します。



CALLSUB コマンドが実行されると、PC-DMISはサブルーチンでのすべてのコマンドを実行します。これにはマークされないコマンドも含まれます。

呼び出されたサブルーチンは、有効なPC-DMISバージョン（バージョン 2020 R2 から 2026.1 まで）で作成されている必要があります。

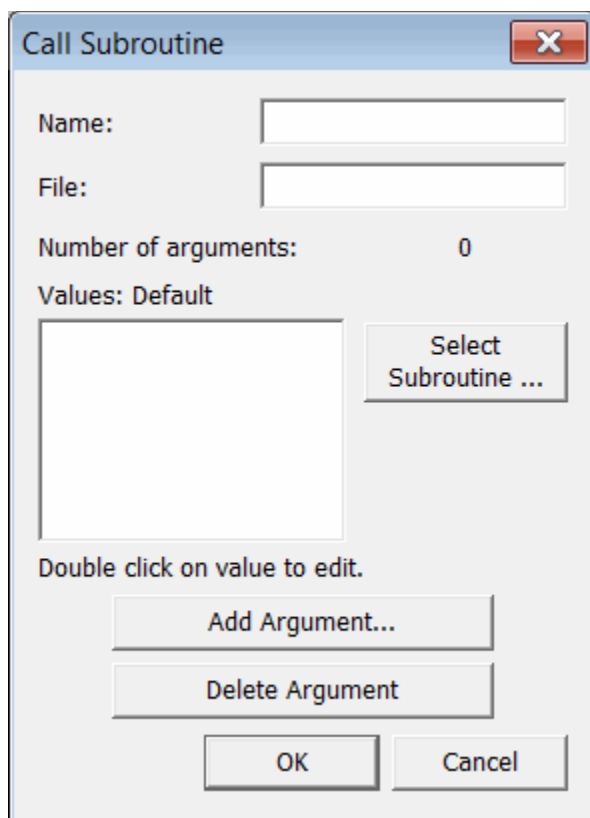
CALLSUB入力によるサブルーチンの呼び出し

編集ウィンドウ内で**CALLSUB**と入力して、このコマンドを挿入することができるうえ、その後、編集ウィンドウにコマンドを予定の場所に表示するようにキーボードでTabキーを押します。

いったんそのコマンドが挿入されると、そのサブルーチン名、外部測定ルーチン内にある場合その所在位置、そして、利用可能な引数に渡す数値を特定する必要があります。渡す引数の例については、「引数をサブルーチンへ引渡し」を参照して下さい。

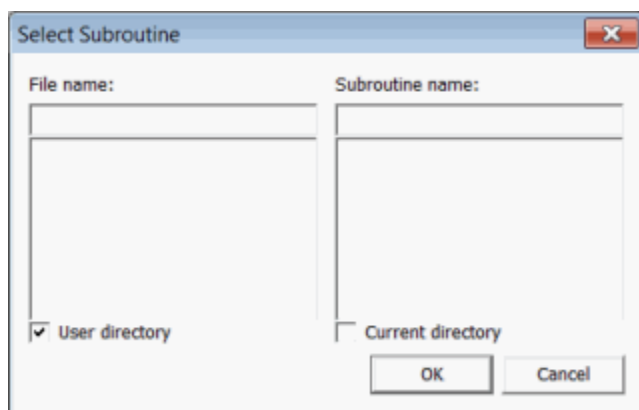
Call Subメニュー項目を用いたサブルーチンの呼び出し

1. サブメニューから[挿入 | フローコントロール コマンド | サブルーチン呼び出し] オプションを選択して下さい。サブルーチン呼び出しダイアログ ボックスが開きます。このダイアログ ボックスに関する情報については「サブルーチン呼び出しダイアログ ボックスについて」を参照して下さい。



サブルーチン呼び出しダイアログ ボックス

2. サブルーチン選択ボタンをクリックします。サブルーチン選択ダイアログ ボックスが開きます。



[サブルーチン選択] ダイアログボックス

3. ユーザー ディレクトリチェック ボックス、現在のディレクトリチェック ボックス、またはその両方を選択して下さい。サブルーチンが含まれる測定ルーチンが、サブルーチン検索で指定されたディレクトリ内にある場合、ユーザー ディレ

サブルーチンでの分枝

クトリチェック ボックスを選択して下さい。その時点でのディレクトリ内にある場合には、**現在のディレクトリチェック** ボックスを選択して下さい。PC-DMISは、選択可能な測定ルーチンすべてを一覧表示します。

4. ご希望のサブルーチンを含む測定ルーチンを選択して下さい。選択された測定ルーチンに関連したサブルーチンすべてが**サブルーチン名**ボックス内に表示されます。
5. 呼び出したいサブルーチンを選択して下さい。
6. **OK**ボタンをクリックします。呼び出し予定のサブルーチンについての情報が、[サブルーチン呼び出し]ダイアログ ボックスの[名称]ボックスと[ファイル]ボックス内に表示されます。
7. そのサブルーチン内に情報を渡す場合、**引数追加**ボタンをクリックし、**引数編集**ダイアログ ボックスを使用して、渡す引数と数値を定義して下さい。このダイアログ ボックスに関する情報については、「引数編集ダイアログ ボックスについて」を参照して下さい。渡す引数の例については、「引数をサブルーチンへ引渡し」を参照して下さい。
8. 再び**OK**ボタンをクリックして下さい。PC-DMISは、編集ウィンドウの選択した場所に**CALLSUB**コマンドを追加します。

CALLSUBコマンド用のシンタックス

サブルーチン呼び出し用の編集ウィンドウのコマンドライン シンタックスは、以下のようです:



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

CS1 = **CALLSUB**コマンドに与えられたラベルID。

<名称> = 呼び出されるサブルーチン名。

<ファイル> = 呼び出すサブルーチンを含む、測定ルーチンへの完全な経路。このフィールドが空白である場合、PC-DMISは、その時点での測定ルーチン内で、そのサブルーチンを探します。

<引数1> = サブルーチン内の最初の引数に渡される値。このフィールドが空白である場合、PC-DMISは、代わりにサブルーチンの最初の引数に定義されたデフォルト値を使用します。

<引数1> = サブルーチン内の2番目の引数に渡される値。このフィールドが空白である場合、PC-DMISは、代わりにサブルーチンの二番目の引数に定義されたデフォルト値を使用します。この構文サンプルは、2つの引数だけを示します。必要に応じて、サブルーチンに他の引数を渡すことができます。



CALLSUBコマンドでは、サブルーチン用に作成されたすべてのオブジェクトへの1セットのポインタを保持して、後でサブルーチンIDを使って簡単にそれらを参照することができるようにする必要があります。ポインタについて詳しくは、「式および変数の使用」章にある「ポインタ」を参照して下さい。

CALLSUBコマンドの例

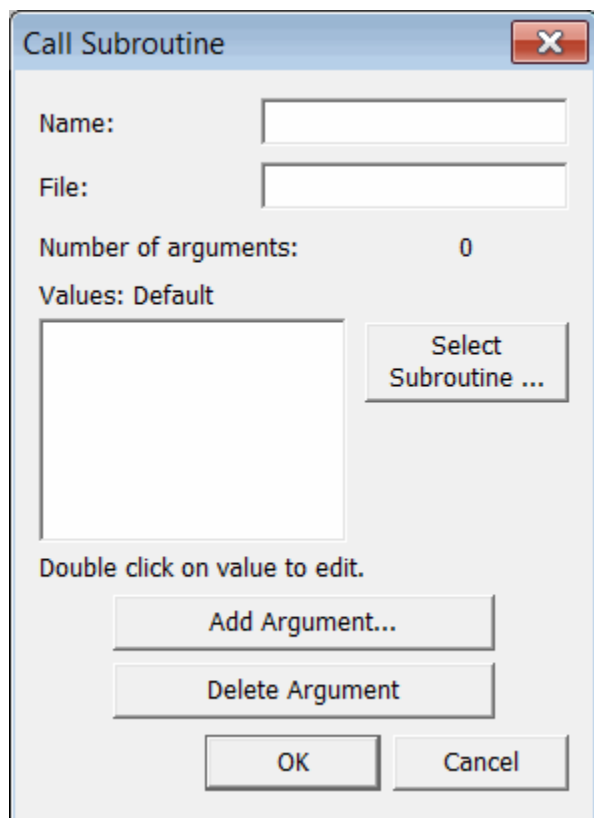


```
CS1          =CALLSUB/GET_OPERATOR_INFO,D:\MEASROUTINES\V42SUBROUTINETEST.PRG:V1,V2,,
```

この、**CALLSUB**コマンドの例、CS1は、D:\MEASROUTINES\ ディレクトリ内にある、測定ルーチンV42SUBROUTINETEST.PRG内に位置する、GET_OPERATOR_INFOという名称のサブルーチンを呼び出します。

これは、2つの数値—この場合、変数V1とV2—をサブルーチン内に渡します。

【サブルーチンの呼び出し】ダイアログ ボックスについて



サブルーチン呼び出しダイアログ ボックス

次にサブルーチンの呼び出しダイアログ ボックス内で利用可能なオプションを説明します。

名前 ボックス



名称ボックスには、サブルーチン選択...ボタン使用後に選択されたサブルーチンの名称が含まれます。

名前の長さは180文字以内に制限されています。名称ボックスに181文字以上入力すると、「行が長すぎます」のエラーメッセージが現れます。

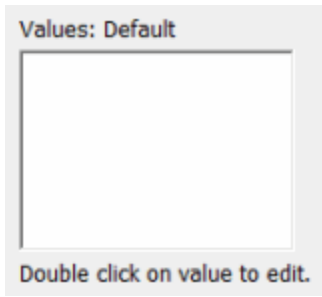
OKボタンを選択すると、その名前が最初の180文字に自動的に短縮され、サブルーチン呼び出しダイアログボックスが閉じます。

ファイル ボックス



ファイルボックスには、呼び出されたサブルーチン ファイルのディレクトリパスが表示されています。

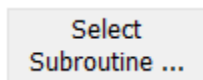
値 ボックス



数値ボックスには、そのサブルーチンに関連した各引数の値の一覧があります。サブルーチンが実行される時、これらの値はサブルーチン内に渡されます。

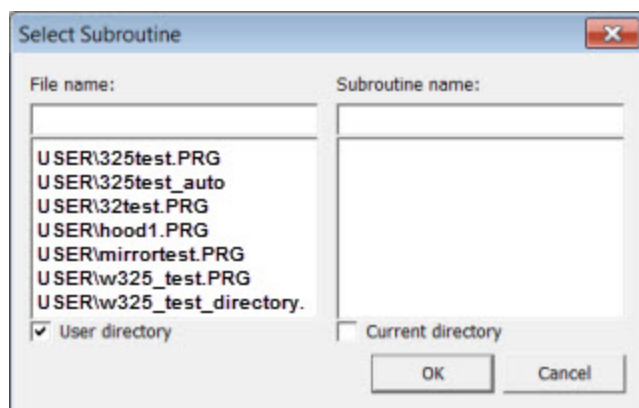
これらの値を編集するには、変更したい値をダブルクリックします。引数の編集ダイアログ ボックスが現れます。このダイアログ ボックスに関する情報については、「引数の編集ダイアログ ボックスについて」の項目を参照して下さい。

サブルーチンの選択ボタン



サブルーチン選択ボタンを押すとサブルーチン選択ダイアログ ボックスが現れます。

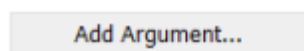
サブルーチンでの分枝



[サブルーチン選択] ダイアログボックス

このダイアログ ボックスでは、ユーザー ディレクトリまたはその時点でのディレクトリ内を検索し、以前に作成されたサブルーチンを呼び出すことができます。**サブルーチン選択**ダイアログ ボックスに関する情報については、「サブルーチンの呼び出し」を参照して下さい。

引数の追加 ボタン



このボタンは、サブルーチン用の引数に渡す値を定義します。

引数の削除 ボタン



このボタンを押すと、値ボックス内の引数を削除することができます。表示された値を選択し、**引数を削除**ボタンをクリックして下さい。その値に関連した引数が削除されます。

「サブルーチン呼び出し」ダイアログ ボックスを用いて、新規の引数を追加するには:

サブルーチン内に渡すための、新規の引数をCALLSUBコマンドに付け加えるには:

1. サブルーチン呼び出しダイアログ ボックスで引数追加ボタンをクリックして引数編集ダイアログ ボックスを開きます。

2. 値ボックスをクリックして下さい。
3. 値ボックス内では、引数の値を入力できます。
4. **[OK]** をクリックします。

「サブルーチン呼び出し」ダイアログ ボックスを用いて、既存の引数を編集するには:

CALLSUBコマンド内の、既存の引数を編集するには:

1. サブルーチン呼び出しダイアログ ボックスで、変更予定の引数の値をダブルクリックして下さい。数値ボックスが現れ、呼び出されたサブルーチンのデフォルト値を表示します。
2. 新しい値を入力します。
3. **OK** ボタンをクリックします。

サブルーチン用の新規の引数を編集、または、作成する方法に関する、より詳しい説明については、「新しいサブルーチンの作成」、及び、「既存のサブルーチンの編集」を参照して下さい。

Call Subroutine コマンドから引数を削除するには :

1. **CALLSUB** コマンド上に、カーソルを配置して下さい。
2. サブルーチン呼び出しダイアログ ボックスにアクセスするために、F9を押して下さい。
3. 引数のリストから、1つ以上の引数を選択します。
4. **引数削除** ボタンをクリックして下さい。
5. **[OK]** をクリックします。

また、編集ウィンドウ内のテキストにある引数を直接削除することも可能です。このためには次の操作を行います:

1. PC-DMISをコマンド モードにして下さい。
2. **CALLSUB** コマンド上にカーソルを配置し、ご希望の引数がハイライト表示されるまで、**タブ** キーを押して下さい。

サブルーチンでの分枝

3. 引数を削除するには、文字「del」と入力します。Delete または Backspace キーを押すことは本当に引数を削除しなくて、それらは、単に空の引数に引数を変更します。

複数アームモードでの**CALLSUB**ステートメントの使用

CALLSUB ステートメントをアーム 1 に割り当てた場合、サブルーチンが呼び出されたときにそのサブルーチン内のコマンドすべてが、アーム1に割り当てられます。

CALLSUB ステートメントをアーム2に割り当てた場合、サブルーチンが呼び出されたときにそのサブルーチン内のコマンドすべてが、アーム2に割り当てられます。

CALLSUB ステートメントを両アーム用にマークした場合、PC-DMISは、そのサブルーチンのマークをオリジナル設定のままにしておきます。

サブルーチンが**MOVE/SYNC** コマンドを含み、**CALLSUB** ステートメントがアーム1またはアーム2に割り当てた場合、PC-DMISは実行時に、これが無効であり、サブルーチンが呼び出されないことを示すエラー メッセージを表示します。

特定アームの実行のためにコマンドを割り当てる情報については、「複数アーム モードの使用」の章にある「アームにコマンドを設定」の項目を参照して下さい。

サブルーチンの例

一般に、渡される引数とサブルーチンの例については、以下の項目内の情報を参考にして下さい。

- サブルーチン内に引数を引き渡し
- サブルーチンのコードのサンプル

サブルーチンへの引数の引渡し

サブルーチンに渡すことができる引数のタイプは数値、変数、テキスト文字列および要素名です。値を引数に渡すには [引数編集] ダイアログ ボックスの [数値] ボックスに値を入力するか、またはコマンドモードで挿入された**CALLSUB** コマンドに直接値を入力します。

これを行う方法の詳細については、「引数の編集ダイアログボックスについて」 および「新規サブルーチンの作成」を参照してください。

変数をサブルーチン内に渡す

データを渡り返すことができる引数は、変数です。変数をサブルーチンへの引数として用いる時、サブルーチン内で、それに対応する変数に起こる変化は渡し戻され、渡された側の変数の値になります。



渡される変数の例:

この例は、サブルーチンへの引数として使用される変数の値が、どのように変更され、どのようにサブルーチンに渡し返されるかを表示します:



```

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/

```

実行フローによるコード説明

まず、V1変数に値6が割り当てられます。

次に、CALLSUBステートメントはサブルーチンを呼び出し、最初の引数としてV1変数を渡します。

SUBROUTINE/MYSUBステートメントは、MYSUBをサブルーチンとして定義します。

A1は最初の引数の名前です。したがって、呼び出しが行われると、A1は最初、呼び出し時のV1と同じ値6になります。

次に、サブルーチンは2つのステートメントを実行します:

- 最初のステートメントは、A1の値を7に増分するASSIGNコマンドです。
- 2番目のステートメントは、更新されたA1値を表示するCOMMENT/OPERコマンドです。

サブルーチンでの分枝

その後、サブルーチンはENDSUB/コマンドによって終了します。

その後、実行のフローは、CALLSUBコマンドの直後のステートメントに戻ります。これは、V1の最終値を表示する別のCOMMENT/OPERコマンドです。



実行がジャンプして戻ると、引数として用いられた変数、この場合はV1、は、サブルーチン内の、それに対応する変数、この場合はA1、に更新されます。したがって、サブルーチンから戻った後、V1の値は7になります。その値が、サブルーチンに渡し返されます。

数値をサブルーチン内に渡す

引数はまた、数値を受け取ることも可能です。



渡される数値の例:

この例は、サブルーチン内で、番号の値がどのように渡されるかを表示します。最大2つの番号を通過して、それらを合計します。



```
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/
```

最初のCALLSUBコマンド（CS1）において、サブルーチン内に渡された数値はありません。代わりに、デフォルトの数値であるNUM1に1、NUM2に1が使用され、合計は2になります。

2番目のCALLSUBコマンド（CS2）において、2つの数値、5と10がサブルーチン内に渡されます。したがって、NUM1は5、NUM2は10、合計は15です。

テキスト スtringをサブルーチン内に渡す

引数はまた、テキスト Stringを受け取ることも可能です。テキスト Stringを渡すには、英数字が、ダブル引用符内に入っていることを確認して下さい。



渡される文字列の例:

この例は、サブルーチン内で、String値がどのように渡されるかを表示します。それは、2つのString値を2つのパラメータ内に渡し、その後、それらをレポートに表示します:



```
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/
```

最初の引数OPNAMEは、値「BOB JONES」を渡され、2番目の引数SHIFTは、「MORNING」を渡されます。その後、この二つのCOMMENT/REPTTコマンドが、渡された文字列を検査レポートに送ります。

フィーチャー名をサブルーチン内に渡す

中括弧内に要素名を渡すことができます。例えば、CALLSUBコマンドで、パラメータとして{PNT1}を使用すると、ソフトウェアはPNT1をサブルーチンに渡します。この方法で要素名を渡すと、サブルーチンはその要素に完全にアクセスできます。



渡される要素名の例:

サブルーチンでの分枝

この例は、サブルーチン内にPNT1というフィーチャー名を渡し、これによって、このサブルーチンはそのフィーチャーに完全なアクセスを得ます。渡される値がない場合、F1というデフォルト フィーチャー名が使用されます。このサブルーチンは、それから、新規の値について尋ね、そのフィーチャーのXの理論値を変更します。




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

当サブルーチンは引き渡された要素にアクセスできるので、`ASSIGN/FEAT1.TX = C1.INPUT` ?>ステートメントが、実際のPNT1要素のXの理論値を変更します。PNT1は、ここで、Xの理論値を恒久的に変更します。

サブルーチンのコードのサンプル

以下のコードサンプルを用いると、オペレーターは、フィーチャー測定後に、その理論的X、Y、及び、Z値の変更を選択することができます。それ以降の実行時には、更新された理論値が使用されます。



```

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

```

サンプル コードの説明

C1=COMMENT/YESNO

サブルーチンでの分枝

この行はユーザーからの「はい」または「いいえ」応答を受けて保存します。

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

この行は式です。これは、コメント1の入力が、「はい」であるかどうかを検査します。「はい」である場合、IFステートメントは「真」であり、IFステートメント後のステートメントを実行し続け、この場合、PNT1 要素が測定されます。「いいえ」の場合、END_IFステートメントに移動します。

```
CS1=CALLSUB/CHANGETHEO, :,
```

この行では CHANGETHEO という名のサブルーチンを呼び出します。ここで測定ルーチンのフローは、SUBROUTINE/CHANGETHEO行にジャンプします。

```
SUBROUTINE/CHANGETHEO
```

この行は CHANGETHEO サブルーチンを初期化します。測定ルーチンのフローは、この行とENDSUB/ 行の間のコードを実行し続けます。

```
POINT1={PNT1}:,
```

これはサブルーチンの唯一の引数です。これによって、サブルーチンは、PNT1要素の情報にアクセスできます。

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

これらの入力コメントはすべて、ユーザーからのX、Y および Z の新しい理論値を受け取り、それらをそれぞれ C2.INPUT、C3.INPUT および C4.INPUT に保存します。

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

この行は C2.INPUTからXの理論値を取得し、PNT1.TX 変数に割り当てます。PNT1.TXはPC-DMISの変数で、PNT1のIDラベルと共に、Xの理論値（TXで示される）を持ちます。

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

この行は C2.INPUTからXの理論値を取得し、PNT1.TX 変数に割り当てます。PNT1.TYはPC-DMISの変数で、PNT1のIDラベルと共に、Yの理論値（TYで示される）を持ちます。

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

この行は C2.INPUT から Z 理論値を取得し、`PNT1.TZ` 変数に割り当てます。`PNT1.TZ` は PC-DMIS の変数で、PNT1 の ID ラベルと共に、Z の理論値（TZ で示される）を持ちます。

ENDSUB/

この行は サブルーチンを終了し、測定ルーチン フローは サブルーチンの呼び出し直後の行に戻ります。この場合、それは `END_IF` / ステートメントです。その後、測定ルーチン フローは、X、Y、及び、Z の理論値と実績値を表示する次のオペレーター コメントを実行し続け、それから、このパーツプログラムは、`ROUTINE/END` コマンドによって終了します。

コメント後の「コマンドモード」コマンド



PC-DMIS コメントを挿入した後、コマンドモードで追加の PC-DMIS コマンドを入力するには、`COMMENT` コマンドの後に最初に Enter キーを 2 回押す必要があります。これは、ユーザーがコメントにテキストを追加する必要はないが、新しいコマンドを追加する準備ができていることを PC-DMIS に通知します。

測定ルーチンの終了

挿入 | フロー制御コマンド | ルーチンの終了メニュー オプションを選ぶと、`ROUTINE/END` コマンドが、編集ウィンドウ内に挿入されます。PC-DMIS が測定ルーチンの実行中にこのコマンドに遭遇するとき、いつでも直ちに測定ルーチンの実行を中止します。

あらかじめ定義された条件に基づいて、通常より早く測定ルーチンを終えたい時、このコマンドは役に立ちます。