目次

フローコントロールを使っての分枝 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 1
If / End If 2
Else If / End Else If 3
Else / End Else 5
While / End While 7
Do / Until 8
Select / End Select 10
CASE / END CASE 12
Default Case / End Default Case 13
汎用ループの使用 14
ループの目的 16
ループの作成 17
ラベルの使用 17
GOTO を使用してラベルにジャンプ 18
エラーにあたっての分枝 20
「On Error」支援のインターフェース 22
サブルーチンでの分枝 23
新しいサブルーチンの作成 25
既存のサブルーチンの編集 29
[引数の編集]ダイアログ ボックスについて 29
サブルーチンの呼び出し 30
サブルーチンの例 37
測定ルーチンの終了 42

フローコントロールを使っての分枝

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
- エラー時の分枝
- サブルーチンを用いた分枝
- 測定ルーチンの終了

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 ステートメント用の、編集ウィンドウ内のコマンドラインは、以下のようになります:

```
IF/式
END_IF/
```

「IF/ END IF」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 **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

点フィーチャーを測定するか否か、ユーザーに尋ねる、以下の例を参考にして下さい。

```
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」の応答を受けて格納します。
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 ステートメント用の、編集ウィンドウ内のコマンドラインは、以下のようになります:

```
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。

```
PNT2=FEAT/POINT,RECT
…
…
測定の終了/
IF/PNT2.X<6.9 OR PNT2.X>7.1
    COMMENT/OPER,"PNT2 の X の測定値: " + PNT2.X + "が公差範囲外です。"
END_IF/
ELSE_IF/PNT2.Y<3.3 OR PNT2.Y>3.5
    COMMENT/OPER,"PNT2 の Y の測定値: " + PNT2.Y + "が公差範囲外です。"
END_ELSEIF/
ELSE_IF/PNT2.Z<.9 OR PNT2.Z>1.1
    COMMENT/OPER,"PNT2 の Z の測定値: " + PNT2.Z + "が公差範囲外です。"
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/PNT2X7.1
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/
この行ラインは、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了します。
ELSE_IF/PNT2.Z1.1
この行は、2 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、2 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 ステートメント用の、編集ウィンドウ内のコマンドラインは、以下のようになります:

```
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*内のご希望の位置に、カーソルを配置して下さい。「ELSE / END ELSE」ブロックは、「IF / END IF」または「ELSE IF / END ELSE IF」ブロックの後に位置つけられなければならないことに注意して下さい。
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

点フィーチャーを測定するか否か、ユーザーに尋ねる、以下の例を参考にして下さい。

```
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
     …
     …
     測定の終了/
END_IF/
ELSE
     PNT2=FEAT/POINT,RECT
     …
     …
     測定の終了/
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」の応答を受けて格納します。

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、このラインの後にある、「END_ELSE」ラインの前にあるコマンドラインが実行されます。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 ステートメント用の、編集ウィンドウ内のコマンドラインは、以下のようになります:

```
WHILE/式
END_WHILE/
```

「WHILE / END WHILE」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。
2. メニュー バーから **While / End While** を選択して下さい。WHILE / END WHILE ステートメント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。

```
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

このラインは、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。それ以外の場合には、PC-DMIS がこのコマンドに出会うと、ループが 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 ステートメント用の、編集ウィンドウ内のコマンドラインは、以下のようになります:
>
> 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。

```
C1=     COMMENT/INPUT, PC-DMIS が PNT1 フィーチャーを測定すべき回数をタイプ入力して下さい:(整数のみ入力)
ASSIGN/COUNT=0
DO/
     PNT1=FEAT/POINT,RECT
     …
     …
     測定の終了/
     ASSIGN/COUNT=COUNT+1
     COMMENT/OPER,"測定値 " + COUNT + " のうち " + C1.INPUT + " 回"。
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。

SELECT / END SELECT ステートメント用の、編集ウィンドウ内のコマンドラインは、以下のようになります:

```
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
```

フローコントロールを使っての分枝

```
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
        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、このコード一覧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。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 ステートメント用の、編集ウィンドウ内のコマンドラインは、以下のようになります:

```
CASE/式
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。

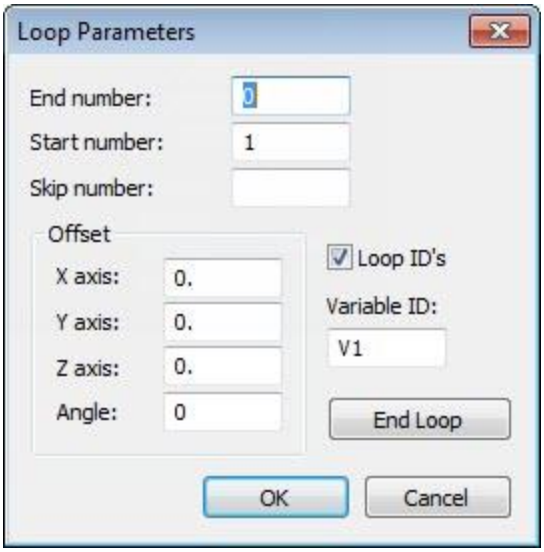
「DEFAULT CASE / END DEFAULT CASE」ステートメント用の、編集ウィンドウ内のコマンドラインは、以下のようになります:

```
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（y または z）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、あるいは、同一パーツ上のパターン間の X、Y、または、Z オフセットをセットアップします。これらは、ループが実行される度に、入力された距離によって、パーツをオフセットします。最初のオフセットは、パーツの原点に基づいています。

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。PC-DMIS は、ループが実行される度に、角度によって、パーツをオフセットします。

ループ内にパーツ配置コマンドがあり、そのループがオフセットを使用している場合、そのパーツ配置用の軸線すべてを定義し*なければなりません*。それに加え、ループ内の配列は、ループ内で測定される要素を使用しなければなりません。

ループの ID - このチェックボックスは、 PC- DMIS 、ループの増分と一致する角括弧内の値として（ループ内）要素 ID を増分するか否かを判定します。

要素 CIR1 がある場合、最初のループで CIR1[1] になり、2 番目のループで CIR1[2] になる…などです。

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:

ループブロックに、STATS/ON コマンド、及び、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。

ループブロックの外に STATS/ON コマンド、及び*ループブロックの中に* 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。

ループブロックの外に STATS/ON コマンド、及び*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、及び*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。編集ウィンドウ内で、「`LOOP/START`」コマンドの後には「`LOOP/END`」コマンドが続く必要があります。

ループの目的

ループ オプションには、主な目的が 3 つあります:

- 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、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 オプションを使用して、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。そのパターンの他の 9 回の繰り返し部分を測定するために、ループ オプションを使用して、この測定ルーチンをオフセットすることができます。

ループ内部でアライメントを使用している場合、PC-DMIS によって、ユーザーは前回保存したアライメントを常に呼び出さなくても、`ALIGNMENT/START` コマンド行におけるアク

ティブなアラインメントを使用することができます。「アライメントの作成と使用」章にある「ループ内でアライメントを使用」トピックを参照して下さい。

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. ボックス内で、必要な変更を行って下さい。
3. 必要に応じてパラメータ（例えば、XY または Z、角の数、オフセットをスキップして、番号を開始、パーツの数など）を選択します。
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
```

ループの手順を完了するには、LOOP/END コマンドで終了しなければなりません。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。これらの方法のうちの 1 つにこのコマンドを挿入することができます: 編集ウインドウで、LOOP/END を入力し、「**挿入 |フロー制御コマンド| End Loop**」メニューアイテムを選択するか、「**ループのパラメーター**」ダイアログボックスから「**End Loop**」ボタンをクリックします。

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:

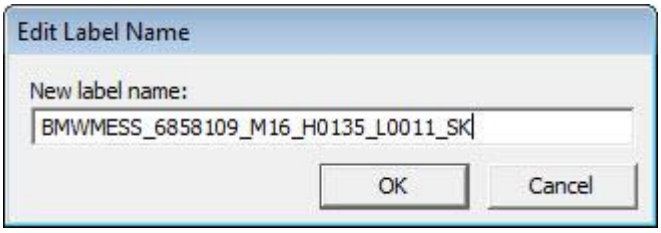
コマンドを入力します。

1. 編集ウイ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. 編集ウイ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。

編集ウィンドウのコマンドモードの LABEL コマンドは次のものを読みます:
`ID = LABEL/`

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. 編集ウインドウのブランク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. 編集ウイ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 | GoTo** を選んで **GoTo** ダイアログボックスにアクセスします。

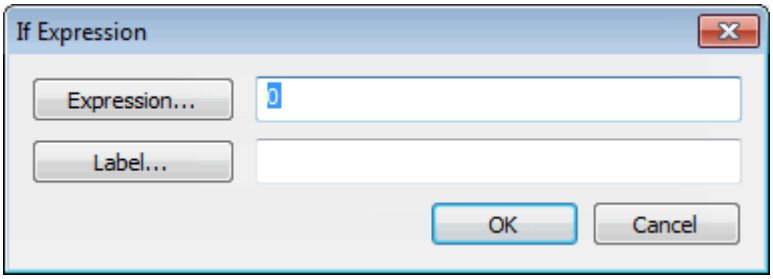
「行き先」ダイアログ ボックス

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**から、目的のラベルを選択するか、**GoTo のラベル**ボックスに、直接的にラベル名を入力します。
5. **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 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 コマンドは次のものを読みます:
`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 コマンドは実行中に無視されます。



「If 式」ダイアログボックス

「**挿入 | フロー制御コマンド | If Goto**」オプションを使用しると、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。

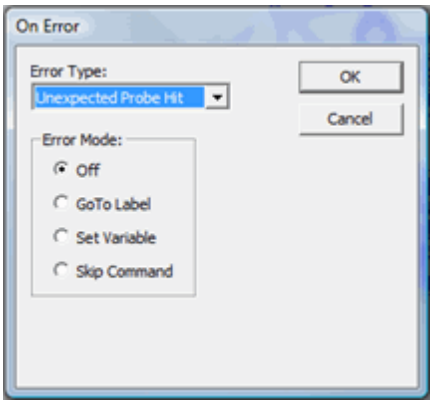
IF GOTO ステートメント用の、編集ウィンドウ内のコマンドラインは、以下のようになります:

```
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 - ラベルがクリックされると、**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。

エラーにあたっての分枝



「エラーにあたって」ダイアログボックス

挿入| フロー制御コマンド| オン・エラー」コマンドは、「**オン・エラー**」ダイアログ・ボックスを開きます。ユーザは機械のエラーが起こったときに実行するアクションを PC- DMIS に伝えるために、このダイアログボックスを使用することができます。

エラータイプ - PC-DMIS は、以下のエラー状況を追跡します:

- 予期しないﾌﾟﾛｰﾌﾞﾋｯﾄ
- 省略されたプローブ ヒット
- 反射装置が発見されず（PC-DMIS ポータブルにおいて、追跡機と共に使用）
- レーザーのエラー（レーザー構成を使用している場合、「ONERROR を使用したレーザープローブエラーの処理」を参照してください）

エラーモード - これらのエラー状況のそれぞれにおいて、以下の行動をとることが可能です:

- **GoTo Label** - 測定ルーチン・フローは定義されたラベルに移ります。「ラベルの使用」を参照してください。
- **変数の設定:** 変数の値を 1 に設定します。
- **オフ:** PC-DMIS は何もしません。
- **スキップコマンド:** 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。

デフォルトでは、すべての測定ルーチンが**オフ**（何もしない）に設定された両種類のエラーのアクションを開始します。各エラーの行動モードは、測定ルーチンを通して変更することができます。

ラベル ID - ユーザはまだ存在していないラベルへの参照を入力できます。

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。

実行中、PC-DMIS が `ON ERROR/ Unexpected Hit / Jump To Label（ON ERROR/ 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ジャンプします。変数値の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ジャンプさせる IF ステートメントを使用してこの変数の値をテストできます。

「On Error」コマンドを使用するには:

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**のリストから、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**」領域で、その条件が満たされた場合に行われるべきアクションを定義するためのオプションのいずれかを選択します。
4. ONERROR コマンドを追加するには、**[OK]**ボタンをクリックしてください。**キャンセル**ボタンは、いかなる変更も行わず、**On Error** ダイアログボックスを閉じます。

On Error コマンド用の編集ウィンドウ上のコマンドラインは、以下のようになります:

```
ONERROR/<type>,<mode>
```

ここで、<タイプ>はエラーの種類で、<モード>は取られたエラーモードの類別です。

「On Error」支援のインターフェース

すべてのインターフェースが、「On Error」コマンドに対応しているわけではありません。お客様のインターフェースが対応しているか、確認するには、以下の表をご覧下さい。

- インターフェースが一覧に表示されている場合、小さな黒いボックスはそのエラーのタイプがそのインターフェースに対応していることを示します。
- お客様のインターフェースが一覧に表示されていない場合、ON ERROR コマンドを使用することはできません。

対応のフィーチャー	予期しないﾌﾟﾛｰﾌﾞﾋｯﾄ	省略されたプローブ ヒット	反射装置が発見されず
B&S 標準	■	■	-
DEA		■	-
ELM	■	■	-
フェデラル/ルノー	■	■	-
ヨハンセン	■	■	-
ライカ追跡機			■
ライツ	■	■	-
LK Direct (別名 LKRS232)	■	■	-
LKﾄﾞﾗｲﾊﾞ	■	■	-

アラインメント	■	■	-
ミツトヨ ブライト	■	■	-
ミツトヨ	■	■	-
Mora	■	■	-
Omnitech	■	■	-
レニショー	■	■	-
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

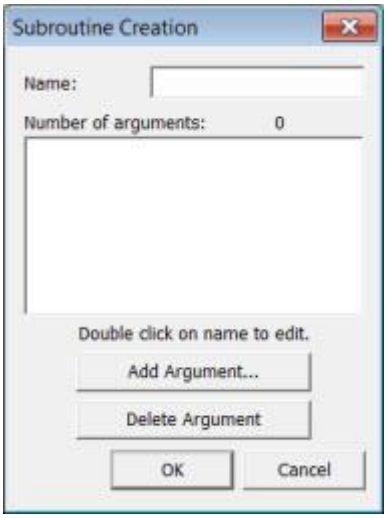
SUBROUTINE 入力によるサブルーチンの作成

編集ウィンドウ内に SUBROUTINE と入力し、その後、タブキーを押すことにより、このコマンドを挿入することが可能です。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と引数を特定する必要があります。詳細は以下のサブルーチン構文と例を見てください。

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。



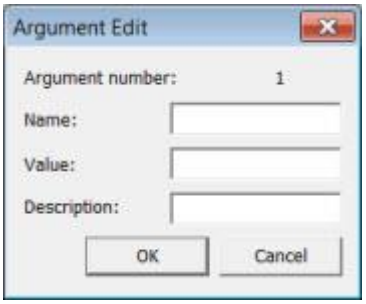
「サブルーチン作成」ダイアログ ボックス

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/<名前>,
<A1> = <引数 1> : <説明>,
<A2> = <引数 2> : <説明>,
=
<コマンド>
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。

<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。

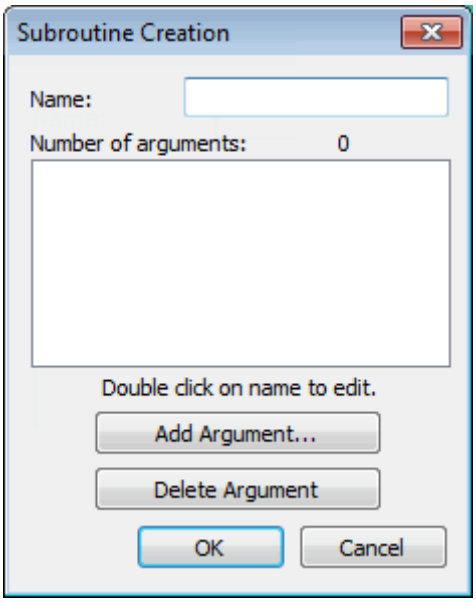
サブルーチン コマンド ブロックの例

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 = <オペレーター> : OPERATOR NAME,
   SHIFT = <シフト> : SHIFT TIME,
=
COMMENT/REPT,OPNAME
COMMENT/REPT,SHIFT
ｻﾌﾞﾙｰﾁﾝの終了/
```

[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したときに使用する名称であるため、一つの測定ルーチンに複数のサブルーチンがある場合はそれぞれ一意の名称でなくてはなりません。

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。

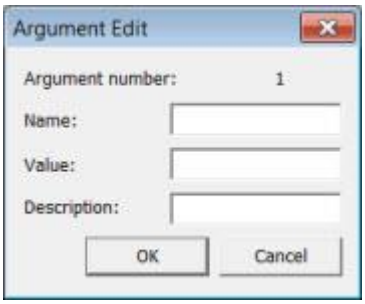
引数を追加 - このボタンは、サブルーチンに新規の引数を追加します。このボタンをクリックすると**引数を編集**ダイアログ ボックスが現れます。このダイアログ ボックスに関する情報については、「引数の編集ダイアログ ボックスについて」の項目を参照して下さい。

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することが可能です。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配置し、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可能です。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 ボックスの理解」の項目を参照して下さい。

[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
- 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、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。

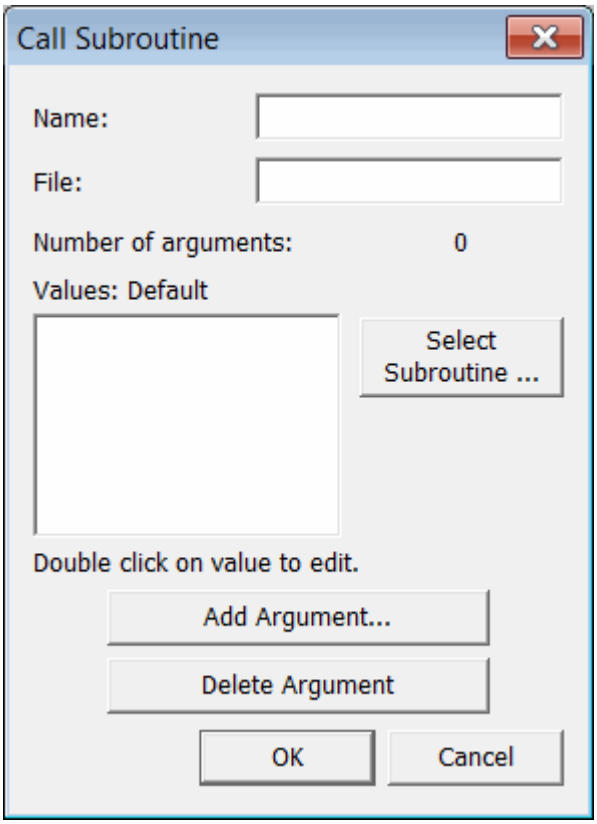
CALLSUB 入力によるサブルーチンの呼び出し

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で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

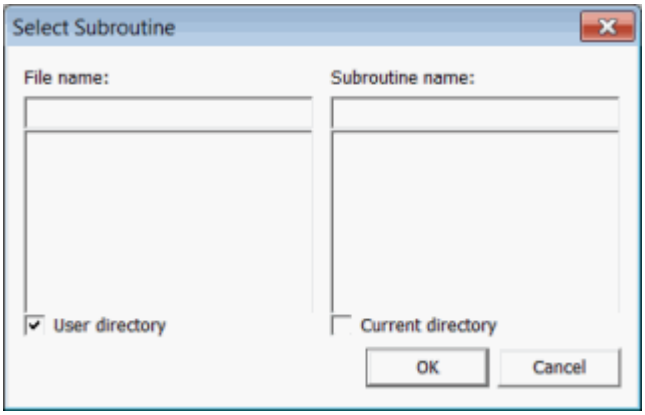
1. サブメニューから[**挿入 | フローコントロール コマンド | サブルーチン呼び出し**]オプションを選択して下さい。*サブルーチン呼び出し*ダイアログ ボックスが開きます。このダ

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。CALLSUB コマンドが編集ウィンドウ内の選択された場所に表示されます。

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/<名称>, <ファイル>:<引数 1>,<引数 2>,
```

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、代わりに、サブルーチン内の最初の引数用に定義されたデフォルト設定値が使用されます。

<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、代わりに、サブルーチン内の２番目の引数用に定義されたデフォルト設定値が使用されます。この構文サンプルは、2 つの引数だけを示します。サブルーチンで必要な場合、他の引数を引き渡せます。

CALLSUB コマンド内では、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。

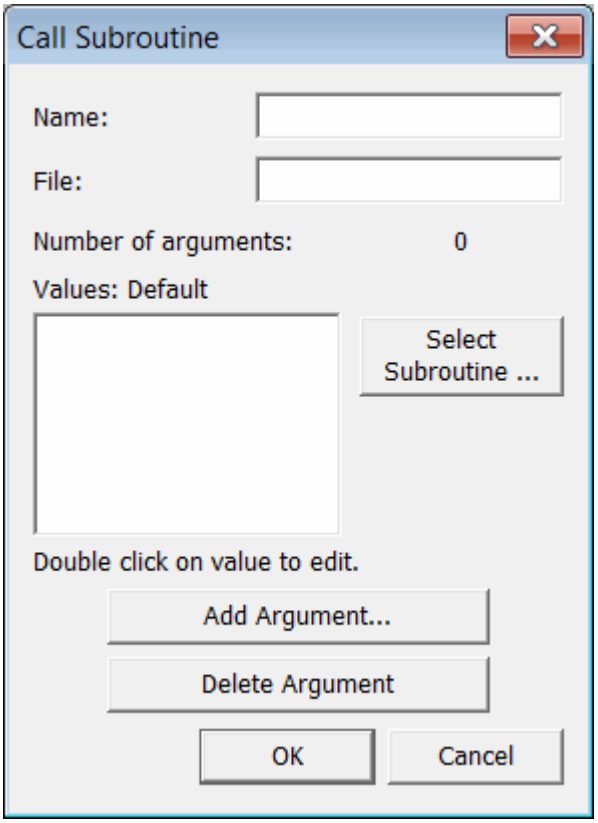
CALLSUB コマンドの例

```
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 つの数値ーこの場合、変数 V1 と V2ーをサブルーチン内に渡します。

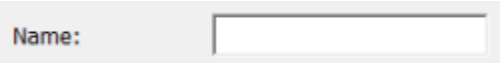
[サブルーチンの呼び出し]ダイアログ ボックスについて



「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。

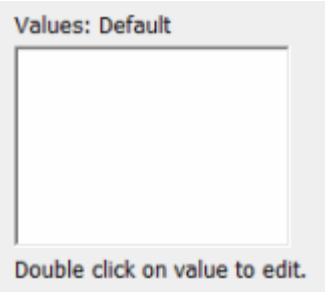
名前の長さは 180 文字以内に制限されています。**名称**ボックスに 181 文字以上入力すると、「行が長すぎます」のエラーメッセージが現れます。

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。

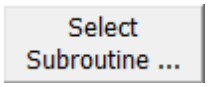
値 ボックス



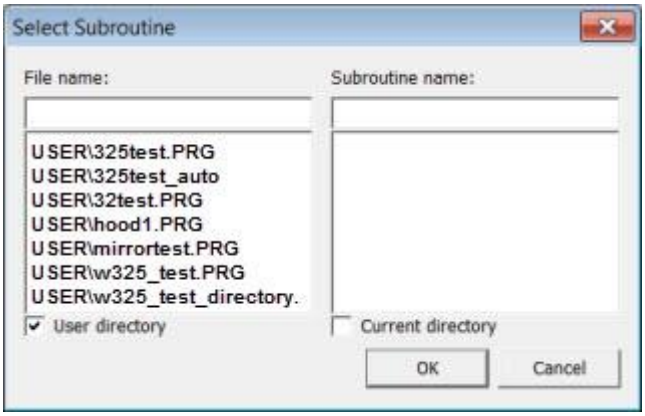
数値ボックスには、そのサブルーチンに関連した各引数の値の一覧があります。サブルーチンが実行される時、これらの値はサブルーチン内に渡されます。

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。
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。値を引数内に渡すには、**引数編集**ダイアログ ボックスの**数値**ボックス内に値をタイプ入力するか、または、コマンド モードにおいて、挿入された CALLSUB コマンドに直接、値を入力して下さい。

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

データを渡し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:

変数 V1 には、数値 6 が割り当てられています:

```
ASSIGN/V1=6.
```

サブルーチン呼び出しが、V1 を最初の引数として渡します:

```
CS1 =CALLSUB/MYSUB,:V1,,,
```

そのサブルーチンは、以下のように定義されます:

```
SUBROUTINE/MYSUB,
A1=0:FIRST ARGUMENT,
=
ASSIGN/A1=A1+1
ENDSUB/
```

A1 は最初の引数の名称であり、呼び出しが行われると、A1 は、V1 が呼び出し時に割り当てられた値と同じ値、6 を持ちます。

サブルーチンには、このステートメントひとつがあります:

```
ASSIGN/A1=A1+1.
```

これは、A1 の値を 7 に増加します。

その後、サブルーチンは ENDSUB/ コマンドによって終了します。

実行フローは、CALLSUB コマンドの直後に続く、ステートメントに戻ります。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、ここで V1 は、7 の値を持ちます。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,
=
COMMENT/OPER,NO,"最初の数字の和, "+NUM1+", ＋2 番目の数字, "+NUM2+",
は: "
,NUM1+NUM2
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。デフォルト設定値、NUM1 として 1、及び、NUM2 として 1 が、かわりに使用され、その合計は 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、そして、NUM2 は 10 で、合計 15 となります。

テキスト ストリングをサブルーチン内に渡す

引数はまた、テキスト ストリングを受け取ることも可能です。テキスト ストリングを渡すには、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、ストリング値がどのように渡されるかを表示します。それは、2 つのストリング値を 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=<オペレータ>:OPERATOR NAME,
        SHIFT=<シフト>: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、COMMENT コマンドが、渡されたストリングをインスペクション 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

要素名はカーリーブラケット内で渡されます。例えば、{F1} を使用して編集ウィンドウ内部で F1 として指定される要素を呼び出します。また、要素名を渡すときサブルーチンはその要素にフルアクセスできます。

渡される要素名の例:

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},,

SUBROUTINE/CHANGE_THEO_X,
FEAT1={F1}:PASSED FEATURE NAME,
=
C1 =COMMENT/INPUT,NO,"PASSED FEATURE:"
,FEAT1
,"現在の理論値 X は: "+FEAT1.TX
,「現在のテオXををタイプ入力して下さい：」
ASSIGN/FEAT1.TX=C1.INPUT
COMMENT/OPER,NO,"要素が次の値に更新されました"+FEAT1.TX
ENDSUB/
```

サブルーチンは引き渡された要素にアクセスできるため、`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。この段階で、PNT1 はその理論的な X 値を永久に変更します。

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>,$
```

フローコントロールを使っての分枝

```
MEAS/IJK,<0,0,1>
C1 =COMMENT/YESNO,NO,FULL SCREEN=NO,AUTO-CONTINUE=NO,
PNT1 用の理論値を変更しますか?
IF/C1.INPUT=="YES"
CS1 =CALLSUB/CHANGETHEO,:,
END_IF/
COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-CONTINUE=NO,
PNT1 用の XYZ 理論値、及び、実績値は:
"Theo X= "+PNT1.TX
"Theo Y= "+PNT1.TY
"Theo Z= "+PNT1.TZ
--------------------
"Actl X= "+PNT1.X
"Actl Y= "+PNT1.Y
"Actl 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,
PNT1 用の新しい X 理論値を入力して下さい。
"現在の値は"+PNT1.TX
ASSIGN/PNT1.TX=C2.INPUT
C3 =COMMENT/INPUT,NO,FULL SCREEN=NO,
PNT1 用の新しい Y 理論値を入力して下さい。
"現在の値は "+PNT1.TY
ASSIGN/PNT1.TY=C3.INPUT
C4 =COMMENT/INPUT,NO,FULL SCREEN=NO,
PNT1 用の新しい Z 理論値を入力して下さい。
"現在の値は "+PNT1.TZ
ASSIGN/PNT1.TZ=C4.INPUT
ENDSUB/
```

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」の応答を受けて格納します。
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、及び、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、及び、C4.INPUT に保存します。
ASSIGN/PNT1.TX = C2.INPUT
この行は C2.INPUT から X の理論値を受け取り、それを 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）を保持します。
ASSIGN/PNT1.TY = C3.INPUT
この行は、C3.INPUT から Y の理論値を取り、それを PNT1.TY 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）を保持します。
ASSIGN/PNT1.TZ = C4.INPUT
この行は、C4.INPUT から Z の理論値を取り、それを 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ラインに戻ります。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。

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。

フローコントロールを使っての分枝

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。