

Tabela de conteúdo

Uso de expressões e variáveis	1
Uso de expressões e variáveis: Introdução	1
Uso de expressões em uma rotina de medição	2
Visualização de valores da expressão	2
Manutenção somente dos valores da expressão	2
Uso de expressões com desvio	3
Uso de expressões com entrada/saída de arquivo	3
Criação de expressões com o construtor de expressão	3
Criação de uma expressão por digitação	4
Criação de uma expressão com o construtor de expressões	4
Verificação da exatidão da expressão	5
Tipo de elemento da expressão	6
ID	6
Extensão	7
Segunda extensão	8
Botão Adicionar	9
Caixa de edição	9
Área de descrição	9
Uso de variáveis com expressões	10
Atribuição de valores a variáveis usando a caixa de diálogo Atribuição	10
Compreensão de componentes da expressão	11
Tipos de operandos	11

Literais	12
Referências	13
Variáveis	20
Estruturas.....	24
Ponteiros.....	26
Matrizes	27
Operadores para expressões.....	40
Precedência	42
Funções	42
Coerção de operandos.....	91
Expressões de ID.....	94
Acesso a propriedades de objeto de um relatório	97
Acessando informações de varredura construída círculos mínimos.....	100

Uso de expressões e variáveis

Uso de expressões e variáveis: Introdução

Uma expressão é uma condição definida pelo usuário, usada com os comandos de controle de fluxo do PC-DMIS. Usando demonstrativos de controle de fluxo, você pode testar essas condições na rotina de medição. Quer a condição seja satisfeita ou não, é possível determinar qual ação o PC-DMIS executará.

As expressões são importantes para fazer com que o PC-DMIS realize tarefas específicas. Quando você usa expressões em conjunto com comandos de controle de fluxo, é possível explorar ainda mais a extensa funcionalidade do PC-DMIS.

Este capítulo explica como criar e usar expressões dentro da janela Edição do PC-DMIS. Para se trabalhar com expressões, a janela Edição precisa ser colocada no modo Comando. Isso permite a visualização direta do código na janela Edição.

Este capítulo aborda os seguintes tópicos principais:

- Uso de expressões em uma rotina de medição
- Criação de expressões com o construtor de expressão
- Uso de variáveis com expressões
- Compreensão de componentes da expressão
- Acesso a propriedades de objeto de um relatório
- Acessando informações de varredura construída círculos mínimos



Se você procurar informações sobre expressões de relatório, consulte Sobre expressões de relatório no capítulo Exibição de resultados de medição no relatório.

Uso de expressões em uma rotina de medição

A Janela Edição do PC-DMIS permite expressões na maioria de seus campos editáveis. Os campos editáveis são geralmente os campos realçados em amarelo quando você pressiona a tecla Tab dentro da janela Edição enquanto está no modo Comando. Os campos que alteram o tipo do elemento não permitem expressões.



A caixa de um elemento automático que especifica o tipo do elemento automático, como Ponto de superfície, Círculo automático, Slot redondo automático etc., não permite expressões.

Os subtópicos desse tópico oferecem uma referência completa às expressões disponíveis.

Visualização de valores da expressão

Para ver o valor de uma expressão, posicione o cursor do mouse sobre a expressão e deixe-o na posição por pelo menos um segundo. A expressão é calculada e uma pequena janela suspensa amarela, que exibe a expressão e seu valor atual, é exibida abaixo do cursor do mouse.

Manutenção somente dos valores da expressão

Para solucionar imediatamente uma expressão da janela Edição, e manter somente o valor, siga estes passos:

1. Selecione o texto da expressão na Janela Edição.
2. Anteceda o texto da expressão com um caractere ` (acento grave).



Suponha que você digite a expressão ``1/7` em um campo numérico. A expressão é resolvida imediatamente e somente o valor (**0,143**) é colocado no campo.

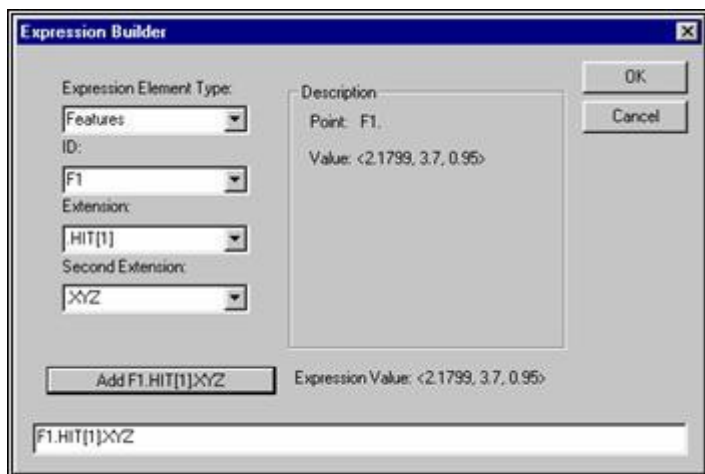
Uso de expressões com desvio

Os comandos de controle de fluxo usam expressões para determinar o fluxo de execução da rotina de medição. Consulte o capítulo Desvio usando controle de fluxo para obter informações sobre quando o desvio pode ou não ocorrer.

Uso de expressões com entrada/saída de arquivo

A gravação de dados em um arquivo de dados externo ou a leitura de dados de um arquivo de dados externo normalmente usa variáveis e outras expressões para gerenciar e armazenar ou exibir efetivamente esses dados. Consulte o capítulo Uso de entrada/saída de arquivo para obter mais informações.

Criação de expressões com o construtor de expressão



Caixa de diálogo Construtor de expressões

O PC-DMIS permite que você crie e inclua expressões na janela Edição digitando-as ou usando a interface da caixa de diálogo **Construtor de expressões**. A opção de menu **Editar | Expressão** exibe a caixa de diálogo **Construtor de expressões**.

Essa caixa de diálogo permite criar uma expressão e inseri-la em um campo editável. Pressionar a tecla F2 com o cursor em um campo que permita expressões também mostra a caixa de diálogo **Construtor de expressões**.

A caixa de diálogo **Construtor de expressões** lista todos os tipos de operadores e de funções disponíveis para expressões.

Criação de uma expressão por digitação

Para criar uma expressão digitando-a diretamente na janela Edição:

1. Abra a janela Edição (**Visualizar | Janela Edição**).
2. Coloque a Janela Edição no modo Comando.
3. Pressione Tab para mover o cursor para um campo editável onde você deseja inserir a expressão. Os campos com um realce em amarelo são considerados "editáveis".
4. Digite a expressão.

Criação de uma expressão com o construtor de expressões



Você precisa estar no modo Comando para a opção Expressão ser ativada.

Para inserir uma expressão usando a caixa de diálogo Construtor de expressões: (**Editar | Expressão**):

1. Abra a janela Edição (**Visualizar | Janela Edição**).
2. Coloque a janela Edição no modo Comando (**Visualizar | Modo Comando**).
3. Mova o cursor para um campo editável onde deseja inserir a expressão.
4. Pressione a tecla F2 com o cursor em um campo que permita expressões. A caixa de diálogo **Construtor de expressões** aparece. A caixa de diálogo **Construtor de expressões** relaciona todos os tipos de operadores, operandos e funções. A seguir estão algumas opções que podem ser referenciadas nessa caixa de diálogo:
 - Tipos de expressões disponíveis
 - Variáveis
 - Elementos
 - Dimensões
 - Alinhamentos
 - Comentários

5. Selecione o tipo de elemento da expressão na primeira lista suspensa. Dependendo de sua opção, outras caixas de combinação são exibidas.
6. Selecione a ID desejada na lista suspensa **ID**.
7. Selecione uma extensão na lista suspensa **Extensão**.
8. Selecione outra extensão na lista suspensa **Segunda extensão**. Se a expressão for utilizável, o botão **Adicionar** fica disponível.
9. Clique no botão **Adicionar**.. A expressão aparece em uma caixa de edição.
10. Clique no botão **OK**. Agora, a expressão aparece onde o cursor estiver na janela Edição.



Você também pode abrir a caixa de diálogo **Construtor de expressões** a partir dessas outras caixas de diálogo:

- A caixa de diálogo **Expressão Se** - Selecione **Inserir | Controle de Fluxo | Se IrPara**. Clique no botão **Expressão**.
- A caixa de diálogo **Atribuição** - Selecione **Inserir | Atribuição**. Clique no botão **Atribuir a** ou **Atribuir de**.

Uma vez criada a expressão, o PC-DMIS insere automaticamente a expressão na próxima posição válida na Janela Edição.

Verificação da exatidão da expressão

Quando o cursor sai do campo onde foi adicionada a expressão, o PC-DMIS tenta verificar a exatidão da expressão. Se há algum problema com a expressão, uma mensagem de erro é exibida indicando um número inválido, ou o texto da expressão pode ficar vermelho. Além disso, as expressões que se referem a objetos inexistentes são exibidas em texto vermelho.

Como o teste de exatidão da expressão ocorre no momento que sai de um campo, um campo que ficou vermelho devido a uma referência a um objeto inexistente (por ex., CÍRCULO1.X), que permanecerá vermelho mesmo que o novo objeto (por ex., CÍRCULO1) , for adicionado. O campo permanece em vermelho até que a expressão seja testada novamente para verificação da exatidão.

Para testar novamente a exatidão da expressão:

1. Mova o cursor para o campo da expressão.

2. Pressione a tecla F2. A caixa de diálogo **Construtor de expressões** será aberta novamente. Todas as alterações feitas na expressão são exibidas na caixa de edição.
3. Pressione a tecla ENTER para fechar a caixa de diálogo.

Tipo de elemento da expressão

A lista suspensa Tipo de elemento da expressão da caixa de diálogo **Construção de expressões (Editar | Expressão)** relaciona os vários tipos de elementos da expressão disponíveis para serem colocados nas expressões. Elas incluem:

- Funções
- Operadores
- Alinhamentos
- Comentários
- Dimensões
- Elementos
- Variáveis

ID

A lista suspensa **ID** da caixa de diálogo **Construtor de expressões (Editar | Expressão)** relaciona o conjunto de itens disponíveis com base no tipo de elemento da expressão selecionado na lista suspensa **Tipo de elemento da expressão**.



A lista de itens disponíveis depende do elemento de expressão selecionado:

- Quando você seleciona **Funções e operadores** na lista suspensa **Tipo de elemento da expressão**, a lista suspensa **ID** contém uma lista das funções e dos operadores disponíveis.
- Quando você seleciona **Elementos** na lista suspensa **Tipo de elemento da expressão**, a lista suspensa **ID** exibe as IDs dos elementos da rotina de medição.

Extensão

A lista suspensa **Extensão** na caixa de diálogo **Construtor de expressões (Editar | Expressão)** é disponibilizada quando o item escolhido na lista suspensa **ID** requer a adição de uma extensão para formar um elemento de expressão usável. A lista suspensa **Extensão** exibe as extensões disponíveis com base no item selecionado na lista suspensa de **ID**.



Suponha que você selecione um elemento **ID** na lista suspensa. O PC-DMIS mostra as extensões que você pode usar para referenciar dados desse elemento (tais como "X", "Y", "Z", "Diam", "Length" etc.) na lista suspensa **Extensão**.

As extensões possíveis incluem estes tipos de dados medidos ou teóricos:

Medido

- All – Todos os valores do elemento estão atribuídos à variável. Consulte o exemplo abaixo.
- X – Valores X medidos dos toques
- Y – Valores Y medidos dos toques
- Z – Valores Z medidos dos toques
- XYZ – Valores XYZ medidos dos toques
- I – Valores I medidos dos toques
- J – Valores J medidos dos toques
- K – Valores K medidos dos toques
- IJK – Valores IJK medidos dos toques

Teórico

- TX – Valores X teóricos dos toques
- TY – Valores Y teóricos dos toques
- TZ – Valores Z teóricos dos toques
- TXYZ – Valores XYZ teóricos dos toques
- TI – Valores I teóricos dos toques
- TJ – Valores J teóricos dos toques
- TK – Valores K teóricos dos toques
- TIJK – Valores IJK teóricos dos toques



Suponha que esse resumo de código seja parte de uma rotina de medição:

```
F1=GENÉRICO/PONTO,DEPENDENTE,CARTESIANO,$
```

```
NOM/XYZ,<8,9,10>,$
```

```
MEAS/XYZ,<7.98,8.98,9.98>,$
```

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

```
MEAS/IJK,<1,0,0>
```

```
ATRIBUIR/MEUELEMENTO=F1.TUDO
```

```
ATRIBUIR/V1=MEUELEMENTO.X
```

```
ATRIBUIR/V2=MEUELEMENTO.TX
```

Quando a rotina de medição completa o resumo de código acima, o PC-DMIS atribui as variáveis com estes valores:

```
V1 = 7,98
```

```
V2 = 8
```

Segunda extensão

A lista suspensa **Segunda extensão** torna-se disponível somente se o item escolhido na lista suspensa **Extensão** requerer a adição de uma segunda extensão, a fim de formar um elemento de expressão utilizável.



Suponha que você esteja referenciando o valor nominal do eixo do local X de uma dimensão chamada "D1". Você iria:

1. Escolher **D1** na lista suspensa **ID**.
2. Selecionar **X** na lista suspensa **Extensão**.
3. Selecionar **Nom** na lista suspensa **Segunda extensão**.

Botão Adicionar

Sempre que você seleciona um elemento de expressão utilizável ou completo usando uma lista, o botão **Adicionar** fica disponível. Esse botão exibe o texto a ser adicionado à expressão.

Por exemplo, se você escolhe os seguintes itens para a expressão:

- Dimensões na lista **Tipo de elemento da expressão**
- D1 na lista **Identidade**
- X na lista **Extensão**
- Nom na lista **Segunda extensão**

Então, o botão **Adicionar** fica ativo e tem o seguinte texto: **Adicionar D1.X.NOM.**

Quando você clica no botão **Adicionar**, o texto é exibido na caixa de edição na parte inferior da caixa de diálogo.



Quando você clica no botão **OK**, o PC-DMIS adiciona o texto da caixa de edição à janela Edição, no campo de expressão onde o cursor está posicionado. Se selecionar um item do campo de expressão da Janela Edição e o texto a ser adicionado contiver parênteses, então o item selecionado seria colocado dentro dos parênteses do texto adicionado.

Caixa de edição

Na parte inferior da caixa de diálogo **Construtor de expressões (Editar | Expressão)** existe uma caixa de edição que mostra a expressão atual. Você pode digitar a expressão diretamente nessa caixa ou usar o botão **Adicionar**.

Área de descrição

A caixa de diálogo **Construtor de expressões (Editar | Expressão)** também contém uma área **Descrição** que fornece informações sobre itens selecionados nas listas suspensas. Um campo próximo ao botão **Adicionar** também mostra o valor atual da expressão.



Expressões inválidas têm um valor 0.

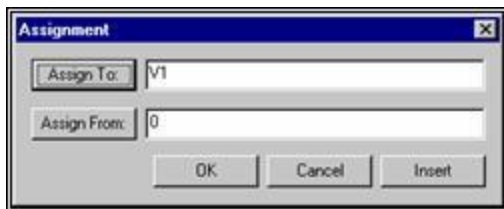
Uso de variáveis com expressões

Variáveis são objetos que contêm valores. Elas se referem a operandos inteiro, real, sequência ou ponto. Variáveis são essenciais para o uso de expressões. Uma variável tem um nome e um valor. O nome é usado para acessar o valor da variável. O nome é constante, o valor pode ser alterado. Você pode atribuir um valor a uma variável usando o comando `ATRIBUIR/`.

Por exemplo, a declaração `ATRIBUIR/V1=2` cria uma variável com o nome de V1 e um valor de 2. `ATRIBUIR/V2=V1+2` acessa o valor de V1. Se V1 ainda tiver um valor de 2 quando esta declaração de atribuição for executada, V2 terá o valor de 4.

Para obter mais informações sobre variáveis, consulte "Variáveis".

Atribuição de valores a variáveis usando a caixa de diálogo Atribuição



Caixa de diálogo Atribuição

A opção de menu **Inserir | Atribuição** exibe a caixa de diálogo **Atribuição**. Use essa caixa para atribuir um valor a uma variável ou a um elemento de dados de um elemento da rotina de medição, uma dimensão ou um alinhamento. O uso do comando atribuição requer uma compreensão básica das expressões do PC-DMIS.

Botão Atribuir a - Esse botão designa a variável que está recebendo o valor calculado na caixa **Atribuir de**. As informações escolhidas usando o botão **Atribuir a** são colocadas na caixa **Atribuir a**. Esse valor pode ser o nome de uma variável ou uma referência a um elemento de dados de um elemento, dimensão ou alinhamento.

O termo "avaliado" representa o resultado da resolução da expressão matemática de um valor.

Botão **Atribuir de** - Esse botão coloca o valor que está sendo atribuído na caixa **Atribuir de**. Se a caixa tiver uma expressão, ela é avaliada no momento da execução e o resultado ou valor do cálculo é atribuído ao objeto especificado na caixa **Atribuir a**.

Botão **Inserir** - Esse botão insere um comando de atribuição na rotina de medição, mantendo a caixa de diálogo **Atribuição** aberta. Uma série de comandos de atribuição podem ser inseridos sem fechar a caixa de diálogo.

Tópicos relacionados:

Compreensão de componentes da expressão

Acesso a propriedades de objeto de um relatório

Compreensão de componentes da expressão

As expressões têm estes tipos de operandos:

- Inteiros
- Números reais
- Seqüências
- Pontos
- Ponteiros de elemento
- Matrizes
- Funções

Eles são discutidos em detalhes a seguir.

Tipos de operandos

Os operandos podem existir na forma de:

- Literais
- Referências

- Variáveis
- Estruturas
- Ponteiros
- Matrizes

Literais

***Inteiros:** 1, -6, 209

Reais: 1, -6, 2,4, -0,1, 345,6789

Sequências: "Olá pessoal", "47", "CÍRCULO 1"

Pontos: Não há uma representação literal disponível para pontos. No entanto, é possível fazer pontos a partir de outras literais que usam a função MPONTO: Por exemplo, `MPONTO(0,0,1)`, `MPONTO(2.2,3.1,4.0)`.

Ponteiro: O nome de um elemento entre chaves: `{CIR1}`, `{LIN2}`, `{F3}`

Matrizes: Não há uma representação literal disponível para matrizes. Entretanto, as matrizes podem ser criadas a partir de outros literais usando-se a função MATRIZ: por exemplo, `MATRIZ(3,5,6)`, `MATRIZ("Olá",2,3,9)`. Estas funções criam 3 matrizes de elementos com os elementos inteiros 3, 5 e 6 no primeiro exemplo e o elemento da sequência "Ola", o elemento duplo 2.3 e o elemento inteiro 9 no segundo exemplo.

Funções: Não há uma representação literal disponível para funções. As funções são definidas usando-se a palavra-chave `FUNÇÃO` e são acessadas por meio das IDs de variáveis. Por exemplo, `ATRIBUIR/Ad2=FUNÇÃO(X,X+2)` define uma função que toma um argumento e adiciona 2 a ela. A função é atribuída à variável Ad2. A função pode ser chamada usando-se a variável Ad2 como a seguir. `ATRIBUIR/Resultado=Ad2(5)`. O valor 7 é atribuído ao resultado.



Os literais numéricos são interpretados como número reais, a menos que o operador ou a função implique no uso de inteiros. Por exemplo, a expressão $10 / 8$ calcula para 1,25 em vez de 1. Observe ainda que a divisão discreta também é possível por meio dos operadores da coerção de operandos. A expressão $\text{INT}(10) / \text{INT}(8)$ é avaliada para 1.

Referências

Referências referem-se aos membros de dados de outros objetos em uma rotina de medição. As referências usam a identidade (ID) de um objeto da rotina de medição seguido por um ponto e uma extensão que se refere ao membro dos dados do objeto.



Se CÍRCULO1 fosse o nome de um círculo medido na rotina de medição, **CÍRCULO1.X** iria se referir ao valor medido do componente X de CÍRCULO1. Todas as referências são calculadas nas coordenadas da peça em relação ao alinhamento atual.

Referências do tipo duplo

As seguintes expressões de referência estão disponíveis:

Extensões válidas para referências de elementos do tipo duplo por exemplo

Formato: <ID do elemento>.<Extensão> -> **CÍRCULO1.X**

CÍRCULO1.X Valor de X medido de CÍRCULO1

CÍRCULO1.Y Valor de Y medido de CÍRCULO1

CÍRCULO1.Z Valor de Z medido de CÍRCULO1

CÍRCULO1.TX Valor teórico (nominal) de X de CÍRCULO1

CÍRCULO1.TY Valor teórico (nominal) de Y de CÍRCULO1

CÍRCULO1.TZ Valor teórico (nominal) de Z de CÍRCULO1

LINHA1.SX Valor de X medido do Pontoinicial da LINHA1

LINHA1.SY Valor de Y medido do Pontoinicial da LINHA1

LINHA1.SZ Valor de Z medido do Pontoinicial da LINHA1

LINHA1.SX Valor teórico de X medido do Pontoinicial da LINHA1
LINHA1.SY Valor teórico de Y medido do Pontoinicial da LINHA1
LINHA1.SZ Valor teórico de Z medido do Pontoinicial da LINHA1

LINHA1.EX Valor de X medido do Pontofinal da LINHA1
LINHA1.ESY Valor de Y medido do Pontofinal da LINHA1
LINHA1.EZ Valor de Z medido do Pontofinal da LINHA1

LINHA1.TEX Valor teórico de X medido do Pontofinal da LINHA1
LINHA1.TEY Valor teórico de Y medido do Pontofinal da LINHA1
LINHA1.TEZ Valor teórico de Z medido do Pontofinal da LINHA1

PONTO.I Componente I medido do vetor para PONTO
PONTO.J Componente J medido do vetor para PONTO
PONTO.K Componente K medido do vetor para PONTO

PONTO.I Componente teórico I do vetor para PONTO
PONTO.J Componente teórico J do vetor para PONTO
PONTO.K Componente teórico K do vetor para PONTO

ELEM1.TIPO O tipo do elemento (isto é círculo, slot, cone). Você pode usar isso para alterar o tipo de um elemento genérico (`ATRIBUIR/Gen1.TIPO=Elem1.TIPO`).

ELEM1.TUDO Refere-se a todos os elementos do elemento. Isso é útil para copiar informações para um elemento genérico. (`ATRIBUIR/ Gen1.TUDO = Elem1.TUDO`)

Vetor de superfície

BORDA.SURFI
BORDA.SURFJ
BORDA.SURFK
BORDA.TSURFI
BORDA.TSURFJ
BORDA.TSURFK

Vetor de ângulo

CIR.ANGI
CIR.ANGJ
CIR.ANGK
CIR.TANGI
CIR.TANGJ
CIR.TANGK

Raio

CÍRCULO1.R
CÍRCULO1.TR
CÍRCULO1.RAD
CÍRCULO1.TRAD
CÍRCULO1.RADIUS
CÍRCULO1.PR – Raio polar
CÍRCULO1.TPR – Raio polar teórico
CÍRCULO1.TRADIUS (somente os primeiros caracteres são significativos)

Diâmetro

CÍRCULO1.D
CÍRCULO1.TD
CÍRCULO1.DIAM
CÍRCULO1.TDIAM
CÍRCULO1.DIAMETER
CÍRCULO1.TDIAMETER (somente os primeiros caracteres são significativos)

Ângulo

CONE.A
CONE.TA
CONE.ANG
CONE.TANG
CONE.ANGLE
CONE.TANGLE
CONE.PA – Ângulo polar
CONE.TPA – Ângulo polar teórico (Somente os primeiros caracteres são significativos)

Comprim.

LINHA.L
LINHA.TL
LINHA.LEN
LINHA.TLEN
LINHA.LENGTH
LINHA.TLENGTH (somente os primeiros caracteres são significativos)

Altura

CILINDRO.PH – Altura polar
CILINDRO.TPH – Altura polar teórica

Raio, Ângulo, Altura

PONTO.RAH – Ponto com raio, ângulo e altura medidos

PONTO.TRAH – Ponto com raio, ângulo e altura teóricos

Área

BLOB1.AREA - Retorna o valor de área medido para um elemento blob.

BLOB1.TAREA - Retorna o valor de área teórico para um elemento blob.

Por exemplo, `ATRIBUIR/V1=BLOB1.AREA` retorna o valor de área medido do elemento BLOB1 e o atribui para a variável V1..

Atualmente, somente o elemento Blob funciona com essas extensões de área. Para mais informações sobre o elemento Blob, consulte o tópico "Blob do Vision" na documentação "PC-DMIS Vision".

Extensões válidas para referência de dimensão do tipo duplo por exemplo

Formato: <ID da dimensão>.<EIXO>.<Elemento da dimensão> -> `DIM1.X.NOM`

DIM1.X.NOM	O valor nominal do local do eixo X de DIM1
DIM1.X.MEAS	O valor medido do local do eixo X de DIM1
DIM1.X.MAX	O desvio máximo da localização do eixo X de DIM1
DIM1.X.MIN	O desvio mínimo da localização do eixo X de DIM1
DIM1.X.PTOL	O valor de tolerância positiva para a localização do eixo X de DIM1
DIM1.X.MTOL	O valor de tolerância negativa para a localização do eixo X de DIM1
DIM1.X.DEV	O desvio na localização do eixo X de DIM1
DIM1.X.OUTTOL	O valor fora da tolerância na localização do eixo X de DIM1
DIM1.Y.NOM	O valor nominal para a localização do eixo Y de DIM1
DIM1.Z.DEV	O desvio na localização do eixo Z de DIM1
DIM3.PA.MEAS	O valor medido para a localização do ângulo polar de DIM3
DIM4.M.PTOL	O valor de tolerância positiva para o eixo M de DIM4
DIM4.PTOL	O valor de tolerância positiva para o eixo M de DIM4 (consulte a *observação em "Eixos válidos" a seguir).
DIM5.BTOL	A Tolerância Bônus onde DIM5 é uma Posição.

Eixos válidos:

X, Y, Z, D, R, A, T, V, L, RP, AP, M, PD, RS, RT, S, H, DD, DE, PR



* As dimensões que têm somente um eixo por definição (como, circularidade, concentricidade etc.) podem omitir o qualificador do eixo. Se você usar o qualificador de eixos, observe que todos esses tipos de dimensões (que têm somente um eixo) usam o qualificador do eixo M, com exceção das dimensões dos ângulos de 2D e de 3D, que usam o qualificador do eixo A.

Extensões válidas para referências de alinhamento do tipo duplo por exemplo:

Formato: <ID do alinhamento>.<Eixo ou origem do alinhamento>.<Componente do eixo ou da origem do alinhamento> -> A1.ORIGEM.X

A1.ORIGIN.X	Componente de alinhamento X da origem medida de A1
A2.ORIGIN.Y	Componente de alinhamento Y da origem medida de A2
A1.ORIGIN.Z	Componente de alinhamento Z da origem medida de A1
A1.XAXIS.I	Componente de alinhamento I do eixo medido X de A1
A1.YAXIS.J	Componente de alinhamento J do eixo medido Y de A1
A1.ZAXIS.K	Componente de alinhamento K do eixo medido Z de A1
A1.CORIGIN.X	Componente de alinhamento X da origem baseada em dados teóricos (C de CAD) de A1
A1.CXAXIS.J	Componente de alinhamento J do eixo X baseado em dados teóricos (C de CAD) de A1

Referências do tipo ponto

As seguintes expressões de referência estão disponíveis:

Extensões válidas para referências de elementos do tipo ponto por exemplo

Formato: <ID do elemento>.<Extensão> -> CÍRCULO1.XYZ

CÍRCULO1.XYZ	Centróide medido de CÍRCULO1
CÍRCULO1.TXYZ	Centróide teórico de CÍRCULO1
LINHA1.SXYZ	Ponto inicial medido de LINHA1
LINHA1.TSXYZ	Ponto inicial teórico de LINHA1
LINHA1.EXYZ	Ponto final medido de LINHA1
LINHA1.TEXYZ	Ponto final teórico de LINHA1
CÍRCULO1.IJK	Vetor medido de CÍRCULO1
CÍRCULO1.TIJK	Vetor teórico de CÍRCULO1
BORDA.SURFIJK	Vetor de superfície medido de BORDA
BORDA.TSURFIJK	Vetor de superfície teórico de BORDA
AUTOCIR1.ANGIJK	Vetor de ângulo medido de AUTOCIR1
AUTOCIR1.TANGIJK	Vetor de ângulo teórico de AUTOCIR1

Extensões válidas para referência de alinhamento do tipo ponto por exemplo

Formato: <ID do alinhamento>.<Eixo ou origem do alinhamento> -> A1.XAXIS

A1.ORIGIN	Origem medida de alinhamento A1
A1.XAXIS	Eixo X medido de alinhamento A1
A1.YAXIS	Eixo Y medido de alinhamento A1
A1.ZAXIS	Eixo Z medido de alinhamento A1
A1.CORIGIN	Origem teórica de alinhamento A1
A1.CXAXIS	Eixo X teórico de alinhamento A1
A1.CYAXIS	Eixo Y teórico de alinhamento A1
A1.CZAXIS	Eixo Z teórico de alinhamento A1

Referências do tipo seqüência

As referências aos comentários são os únicos tipos de objetos que são do tipo seqüência. Somente comentários ENTRADA ou comentários SIM/NÃO podem ser referidos por meio de referências. Esses tipos de comentários têm uma ID que VOCÊ pode utilizaR para identificar o comentário.

Formato: <ID do comentário>.INPUT -> C1.INPUT

C1.INPUT - valor da entrada (a partir do operador) para o comentário C1

Os tipos de comentário SIM/NÃO definem a entrada para a sequência sim ou não apropriada com base no idioma atual do PC-DMIS. Na versão em inglês do PC-DMIS, se o operador pressiona o botão Sim, a sequência é definida para "SIM". Se o operador pressiona o botão Não, a sequência é definida para "NÃO". Ao comparar as sequências para testar as opções "SIM" ou "NÃO", a comparação é sensível a maiúsculas e minúsculas. Além disso, a comparação com "sim" ou "não" sempre falha, mesmo se a entrada do comentário SIM/NÃO estiver definida para "SIM" ou "NÃO".

Variáveis

As variáveis podem ser de qualquer um dos sete tipos de operandos: inteiro, real, sequência, ponto, apontador de elemento, matriz e função. As variáveis passam a existir e recebem seus valores e tipos por meio da declaração [ATRIBUIR](#).

A ID da variável pode ser qualquer sequência alfanumérica que não comece por um caractere numérico. Underlines também podem ser usados na ID da variável fornecida, contanto que não sejam o primeiro caractere.



Desde que a rotina de medição permaneça aberta, o PC-DMIS salva os valores das variáveis entre execuções. Isso significa que quando a execução termina, o PC-DMIS usa os valores como sendo o final da rotina quando você a reinicia. Você pode ou não querer este comportamento. Se deseja que os valores das variáveis sejam novos, o indicado é apagar todos os valores com declarações de atribuição no início da rotina. Por exemplo, se você usa um valor de variável V1 em alguns cálculos, pode usar [ATRIBUIR/V1=0](#) para apagar tal variável.



Se a janela Edição está ativa, o PC-DMIS indica o valor atual da variável, independente de onde o cursor esteja posicionado no campo. Durante a execução, os valores da variável se alteram com base no fluxo da execução. Posicione o ponteiro do mouse sobre a variável desejada para encontrar o valor atual.

[ATRIBUIR/V1=2, 2+2](#)

A variável V1 é um número real com o valor 4,2.

Uso de expressões e variáveis

```
ATRIBUIR/VAR1=CÍRCULO1.X
```

A variável VAR1 é um número real com um valor igual ao medido em CÍRCULO1.X no momento da atribuição.

```
ATRIBUIR/MYVAR=LINHA1.XYZ
```

A variável MYVAR é um ponto com o mesmo valor do centroide medido de LINHA1 no momento da atribuição.

```
ATRIBUIR/SVAR="Olá pessoal"
```

A variável SVAR é uma sequência com o valor "Olá pessoal".

Nesses exemplos, valores são atribuídos às variáveis. Uma vez atribuído um valor a uma variável, você pode usar essa variável como um operando em qualquer campo de expressão.

Aqui, V1 é usado em um campo numérico. Ela é usado como o valor pré-toque do comando de pré-toque:

```
ATRIBUIR/V1=1/3PRÉ-TOQUE/V1
```



Como as expressões podem ser utilizadas na maioria dos campos modificáveis, a seguinte expressão também é válida e tem o mesmo efeito: PRETOQUE / 1/3.

Pode-se fazer referência aos componentes de variáveis do tipo ponto individualmente, usando a notação de extensão de ponto usada para referências.

```
ATRIBUIR/ V1=MPOINT(3, 4, 5)
```

V1 é do tipo ponto com o valor 3, 4, 5.

```
ATRIBUIR/XVAR=V1.X
```

XVAR é do tipo duplo com o valor 3.

```
ATRIBUIR/YVAR=V1.Y
```

YVAR é do tipo duplo com o valor 4.

```
ATRIBUIR/IVAR=V1.I
```

IVAR é do tipo duplo com o valor de 3.

```
ATRIBUIR/REDUNVAR=V1.XYZ
```

REDUNVAR é do tipo ponto com o valor 3, 4, 5.

As extensões a seguir são equivalentes umas às outras. Ambas são fornecidas para esclarecer o significado de uma expressão em uma rotina de medição.

Dado que V1 é do tipo ponto.

V1.X é o mesmo que V1.I

V1.Y é o mesmo que V1.J

V1.Z é o mesmo que V1.K

V1.XYZ é o mesmo que V1.IJK e V1 sem nenhuma extensão.

Se uma variável do tipo sequência tem um valor de sequência igual ao nome da ID de um elemento, dimensão ou alinhamento, a variável pode ser usada como um objeto de referência:

```
ATRIBUIR/V1="CÍRCULO1"
```

Os operandos a seguir são possíveis e válidos e forneceram um elemento com o nome CÍRCULO1 existente.

V1.X - valor X medido de CÍRCULO1

V1.TX - O valor X teórico de CÍRCULO1

V1.Diameter - diâmetro medido de CÍRCULO1

V1.Radius - O raio medido de CÍRCULO1

Esse tipo de endereçamento indireto disponível em variáveis de sequência só está disponível em um nível de endereçamento indireto. O seguinte não funcionará.



```
ATRIBUIR/V1="CÍRCULO1"  
ATRIBUIR/V2="V1"
```

V2.X - Isso é avaliado como 0 em vez do valor medido atual de CÍRCULO1.X.



A referência V2.X *não* é sinalizada como um erro com texto em vermelho, mesmo que uma expressão acima dela defina seu tipo como uma sequência. O motivo para ela não pode ser sinalizada como um erro é que o fluxo de execução da rotina de medição não é conhecido até o momento da execução.

Entretanto, se usar chaves, a seguinte expressão funcionará:



```
ATRIBUIR/V1={CÍRCULO1} ATRIBUIR/V2={V1}
```

V2.X - Isso fornece o valor de CÍRCULO1.X.

Considere o seguinte exemplo:



```
ATRIBUIR/V1="CÍRCULO1"  
ATRIBUIR/V2="V1" SE/CÍRCULO1.X>CIRCLE1.TX,GOTO,L2  
L1=RÓTULO/ ATRIBUIR/V3=V2.X  
GOTO/RÓTULO,L3 L2=RÓTULO /  
ATRIBUIR/V2=MPOINT(2,5,7)  
GOTO/RÓTULO,L1 L3=RÓTULO/
```

Durante a execução da rotina de medição, se o valor de CÍRCULO1.X é maior do que o valor de CÍRCULO1.TX, então a expressão V2.X é válida e avaliada como 2. Caso contrário, a expressão V2.X é avaliada como 0, uma vez que o valor de V2 no momento de ATRIBUIR para V3 é a sequência "V1". É responsabilidade do programador da peça assegurar que as expressões façam o que é esperado nesses casos.



Quase todas as referências do elemento podem ser usadas do lado esquerdo da declaração de `ATRIBUIR` para colocar um valor em um membro de dados medido ou teórico de um elemento. A única exceção é para os componentes de vetores I, J, K únicos. Para atribuir a vetores, você tem que atribuir de uma só vez o vetor completo usando uma expressão que avalie para um ponto. Os dados do vetor são normalizados à medida que são inseridos nos membros de dados do vetor do elemento.



```
ATRIBUIR/CÍRCULO1.I=2-illegal  
ATRIBUIR/ATRIBUIR1.IJK=MPOINT(2,0,0)-legal(o vetor  
é normalizado para 1,0,0)
```

Para obter informações sobre o uso de variáveis com dimensões, consulte o tópico Dimensionamento de variáveis no capítulo Uso de dimensões legadas.

Estruturas

Você pode usar um tipo de variável chamada *estruturas* para colocar extensões em uma variável para identificar o subelemento dessa variável. Você pode ver isso neste resumo de código:



```
ATRIBUIR/V1.HEIGHT=6
ATRIBUIR/V1.WIDTH=4.3
ATRIBUIR/V1.MODE="CIRCULAR"
ATRIBUIR/V1.POINT
= MPOINT(100.3,37.5,63.1)
```

Em que:

- `V1` é a estrutura
- `HEIGHT`, `WIDTH`, `MODE` e `POINT` são sub-elementos da estrutura

Regras para estruturas

- Da mesma forma que Variáveis, as estruturas não precisam ser declaradas.
- Os subelementos de uma estrutura podem ser qualquer um destes tipos de variável:
 - Inteiro
 - Dupla
 - Ponto
 - Ponteiro do elemento
 - Função
 - Matriz
 - Estrutura

Por exemplo, é possível ter elementos de estrutura que sejam matrizes e elementos de matriz que sejam estruturas. Você pode ver as expressões válidas nos seguintes resumo de código:

```
ASSIGN/CAR.LEFTSIDE.DOOR[2].QUADRANT[3].JOINT[5].HIT[4]=MPOINT(558.89,910.12,42.45)
```

```
COMMENT/OPER,"Posição atual de Z:
"+CAR.LEFTSIDE.DOOR[2].QUADRANT[3].JOINT[5].HIT[4].Z
```

```
ASSIGN/CURRENTJOINT=LEFTSIDE.DOOR[2].QUADRANT[3].JOINT[5]
```

```
COMMENT/OPER,"Próximo toque: "+CURRENTJOINT.HIT[4]
```

Estruturas com variáveis do tipo ponto

Se uma variável é do tipo ponto, você ainda pode usar as extensões .X, .Y, .Z, .I, .J e .K para obter itens individuais do ponto. Você também pode usar qualquer uma das extensões desse exemplo nas estruturas sem forçar o seu uso como elementos de ponto, como mostrado aqui.



```
ATRIBUIR/V1.X="Alguma
string" ATRIBUIR/V1.Y=ARRAY(1,3,5,9,7)
ATRIBUIR/V1.Z=MPOINT(3,5,7)
```

COMENT/RELAT,V1.X	A saída é "Alguma seqüência"
COMENT/RELAT,V1.Y[2]	A saída é 3, o segundo elemento da matriz.
COMENT/RELAT,V1.Z.Y	A saída é 5, o valor Y do MPOINT.

Combinando estruturas com a capacidade de função da linguagem de expressão do PC-DMIS, é possível ter referências à estrutura dinâmica, conforme mostrado aqui:



```
ATRIBUIR/ESTRUTDINÂM=FUNCTION((X,Y),X.Y)
C1=COMENT/ENTRADA,Insira no item
ATRIBUIR/STRTESTE=C1.INPUT
ATRIBUIR/FRONT=LEFT(STRTESTE,INDEX(STRTESTE,".")-1)
ATRIBUIR/TRAS=MID(STRTESTE,INDEX(STRTESTE,".") )
ATRIBUIR/RESULT=DYNAMICSTRUCT(FRONT,TRAS)
```

Essa parte do exemplo solicita a inserção de uma referência de variável, divide a referência no primeiro '.' e, em seguida, atribui o **RESULTADO** para ser igual à referência usando a função **DYNAMICSTRUCT**.

Portanto, se você digitou `V1.Y[4]` para a variável `C1.INPUT`, o `RESULTADO` seria o valor 9 (o quarto elemento da matriz atribuída a `V1.Y`).

A avaliação do tempo de aprendizagem de expressões foi aprimorado para mostrar com precisão todos os elementos de uma estrutura ou matriz.

Ponteiros

Os ponteiros também são conhecidos como "Ponteiros de elemento". Consulte no Glossário o termo "Ponteiros de elemento" para obter mais informações.

Os ponteiros oferecem um modo de fazer referência a um elemento por meio de uma variável ou passar objetos usando o subcomando de chamada. Os ponteiros são similares ao endereçamento indireto por meio de nomes de sequência. Entretanto, a vantagem de usar ponteiros é com as subrotinas. Os ponteiros, diferentemente das sequências, quando passados como argumentos de uma subrotina, permitem uma modificação direta do objeto apontado pela subrotina. Os ponteiros não são usados em expressões complexas. Se eles forem usados em uma expressão complexa, o ponteiro é avaliado como zero.

Considere os seguintes exemplos.

Exemplo de uso do ponteiro:

Nesse exemplo, `V1` é definido como um ponteiro que aponta para `CIR1`.

```
ATRIBUIR/V1={CIR1}
```

Nesse exemplo, a `DIST` é atribuído o valor da distância de `CIR1` a partir da origem:

```
ATRIBUIR/DIST=DOUBLE (V1.XYZ)
```

Também é possível colocar uma expressão entre chaves para obter um ponteiro de elemento. Agora, os exemplos a seguir são todas formas válidas de se obter o ponteiro para o elemento `CIR1`:

```
ATRIBUIR/CONTELEM = 1
```

```
ATRIBUIR/V1={"CIR" + CONTELEM}
```

Atribui a expressão `"CIR1"` para `V1`.

```
ATRIBUIR/V2="CIR1"
```

Uso de expressões e variáveis

```
ATRIBUIR/V3={V2}
```

Atribui a expressão "CIR1" da variável V2 para a variável V3.

```
C1=COMENT/ENTRADA,Digite um nome de elemento.
```

```
ATRIBUIR/V4={C1.INPUT}
```

Obtém o nome do elemento de C1.INPUT e o coloca na variável, V4.

Exemplo de subrotina:

Na rotina de medição chamada:

```
CS1=CHAMARSUB/SUB.PRg,CHANGEX,{CIR1}
```

Na subrotina:

```
GEN1=GENÉRICO/ELEMENTO
```

```
SUBROTINA/CHANGEX,ARG1={GEN1}
```

(Durante a execução, quando o código passa CIR1 para a rotina, CIR1 toma o lugar de GEN1)

```
ARG1.X=5
```

(Define o valor de X medido de CIR1 para 5)

```
END/SUB-ROTINA
```

Exemplo de expressão complexa:

```
ATRIBUIR/V1={CIR1}+2
```

{CIR1} é avaliado como zero e então a expressão inteira é avaliada como 2.

Matrizes

Há três tipos de matrizes disponíveis: elementos, toques e variáveis



Embora as matrizes multidimensionais sejam exibidas como multidimensionais no software, na realidade elas podem ser utilizadas apenas como matrizes de dimensão única até que você as preceda com um comando ÍNDICES DE MATRIZ (consulte o tópico "Objeto de índices de matriz:").

Matrizes de elementos

Quando um elemento é medido mais de uma vez durante a execução de uma rotina de medição em função de algum tipo de loop, o software cria uma matriz de elementos automaticamente. O número de elementos na matriz de elementos é igual ao número de vezes que o elemento foi executado.



Se um elemento círculo medido está dentro de um loop enquanto ele é executado cinco vezes, então existe uma matriz de cinco círculos medidos. Se a ID do círculo medido é CIR1, então você pode usar uma expressão de matriz para acessar ocorrências individuais do objeto círculo medido. Você deve usar colchetes para indicar a ocorrência que deseja, desta maneira:

```
ATRIBUIR/V1=CIR1 [3] .X
```

V1 é atribuído o valor X medido da terceira ocorrência do círculo CIR1.



Quando há uma matriz de elementos para um determinado elemento, mas a notação de matriz não é utilizada em uma referência para aquele elemento, é usada a ocorrência mais recente. No exemplo acima, a referência CIR1.X deveria ser igual a CIR1 [5] .X, considerando que a quinta ocorrência seria a mais recente do objeto.

Você pode usar expressões entre colchetes de uma expressão de matriz:

CIR1 [3] .X e CIR1 [2+1] .X seriam, portanto, equivalentes.

Este próximo exemplo usar dois blocos de comando de loop Enquanto/Fim enquanto. O primeiro bloco executa o círculo CIR1 cinco vezes. O segundo bloco usa a variável

Uso de expressões e variáveis


V1 dentro de colchetes, desta maneira, CIR1[V1].XYZ, paa enviar o centroide medido de cada uma das cinco execuções para a janela Relatório:



```
ATRIBUIR/V1=1
ENQUANTO/V1<6
CIR1      =ELEM/CÍRCULO,CARTESIANO,INT,MÍNIMO_QDR
          TEÓR/<40,30,-4.824>,<0,0,1>,30
          REAL/<40.002,29.991,-4.836>,<0,0,1>,29.982
          MED/CÍRCULO,4,ZMAIS
          TOQUE/BÁSICO,NORMAL <41.984,44.868,-
2.885>,<-0.132272,-0.9912135,0>,<41.972,44.85,-
2.891>,USAR TEÓR=SIM
          TOQUE/BÁSICO,NORMAL,<51.721,39.36,-
5.094>,<-0.781412,-0.6240155,0>,<51.706,39.375,-
5.107>,USAR TEÓR=SIM
          TOQUE/BÁSICO,NORMAL,<54.792,32.491,-
5.44>,<-0.9861119,-0.1660821,0>,<54.775,32.474,-
5.453>,USAR TEÓR=SIM
          TOQUE/BÁSICO,NORMAL,<52.526,21.748,-
5.879>,<-0.8350841,0.5501223,0>,<52.537,21.764,-
5.893>,USAR TEÓR=SIM
          FIMMED/
          ATRIBUIR/V1=V1+1
        FIM_ENQUANTO/
        ATRIBUIR/V1=1
        ENQUANTO/V1<6
        COMENT/RELAT,
        "Centroide de CIR1, ocorrência #" + V1
        CÍR1[V1].XYZ
        COMENT/RELAT,
        -----
        ATRIBUIR/V1=V1+1
      FIM_ENQUANTO/
```



Esta é a saída gerada para a janela Relatório:

	PART NAME : Top Holes - Concentric		May 23, 2022	15:25
	REV NUMBER : Rev1	SER NUMBER : 12345	STATS COUNT : 1	


```

Centroid of CIR1, instance #1
<39.994, 30.016, -4.833>
-----
Centroid of CIR1, instance #2
<40.039, 30.011, -4.821>
-----
Centroid of CIR1, instance #3
<40.032, 30.013, -4.819>
-----
Centroid of CIR1, instance #4
<39.991, 30.013, -4.819>
-----
Centroid of CIR1, instance #5
<40.016, 30.003, -4.83>
-----

```

As matrizes também existem em dimensões e alinhamentos que foram executados várias vezes em um determinado processo de execução. Além disso, `Dim1[2]`. Nom e `Alin1[4]`. Origin estariam disponíveis considerando que a Dimensão "Dim1" foi executada no mínimo duas vezes e o alinhamento "Alin1" foi executado no mínimo quatro vezes.

Se uma referência de matriz de elementos está fora das fronteiras (por exemplo, o usuário solicita `CIR1[2,5]` ou `CIR1["Olá pessoal"]`), o item de fronteira superior ou inferior é retornado. Se `CIR1` tem 3 ocorrências, então `CIR1[4]` e acima retorna o valor para `CIR1[3]` e `CIR1[0]`, e abaixo, retorna o valor para `CIR1[1]`. Todas as expressões entre colchetes são forçadas para inteiros, assim 2,5 ficaria 2 e "Olá pessoal" ficaria 0.

Objeto de índices de matriz

Por padrão, as matrizes de elementos são sempre matrizes de uma dimensão. Se for mais conveniente tratar uma matriz de elementos como uma matriz multidimensional, você pode fazer isso com um objeto de índices de matriz.

O objeto de índices de matriz permite especificar fronteiras superior e inferior para várias dimensões de matrizes.

- Quando você define as fronteiras superior e inferior da primeira dimensão, o PC-DMIS cria uma matriz bidimensional onde a primeira dimensão é delimitada e a segunda dimensão não é delimitada.
- Quando você define as fronteiras superior e inferior das duas primeiras dimensões de uma matriz, o PC-DMIS cria uma matriz tridimensional. A última dimensão é sempre não delimitada.



Suponha que o elemento F1 está localizado dentro de um loop WHILE (Enquanto) aninhado. O loop WHILE interno executa cinco vezes e o loop WHILE externo executa três vezes. No término da execução, F1 foi executado 15 vezes e existem, portanto, 15 ocorrências de F1.

Considere o seguinte segmento de exemplo de rotina de medição:

```
MATRIZ_ÍNDICES/1..5,..
```

```
ATRIBUIR/V1 = 1
```

```
WHILE/V1<=3
```

```
    ATRIBUIR/V1=2
```

```
    WHILE/V2<=5
```

```
        PNT1=ELEM/PONTO,RETANG
```

```
        TEÓR/V2,V1,0,0,0,1
```

```
        REAL/1,1,0,0,0,1
```

```
        MED/PONTO,1
```

```
        TOQUE/BÁSICO,V2,V1,0,0,0,1,1,1,0
```

```
        FIMMED//
```

```
        ATRIBUIR/V2=V2+1
```

```
        COMENT/RELAT,"Localização de F1["+V2+", "+V1+"]  
        : "+F1[V2,V1].XYZ
```

```
    END_WHILE/
```

```
    ATRIBUIR/V1=V1+1
```

```
END_WHILE/
```

Esse segmento do código cria uma grade 3 X 5 de 15 pontos medidos.

O comando de índices de matriz limitou a primeira dimensão da matriz de elementos para ficar entre 1 e 5 inclusive. Assim, no relatório de inspeção, em vez de aparecer como F1[1] – F1[15], os objetos aparecem como F1[1, 1] – F1[5, 3], mais consistentes com o layout dos elementos. Observe que o comentário também se refere à matriz de elementos usando sintaxe de matriz bidimensional.

Para inserir um objeto índices de matriz em uma rotina de medição:

1. Usando o teclado, digite **"Matriz"** em uma linha vazia janela Edição.
2. Pressione a tecla Tab no teclado.



Se você desmarca a caixa de seleção **Exibir colchetes para matrizes de elementos**, o elemento não é exibido com o nome entre colchetes. Consulte a descrição de "Exibir colchetes para matrizes de elementos" no tópico "Opções de configuração: guia Configurar ID" do capítulo "Configuração de preferências".

Matrizes de toques

Os toques de um determinado elemento estão disponíveis em uma matriz e podem ser acessados por meio de expressões, usando sintaxe de matriz no formato <IDElem>.Hit[<Expressão da matriz>].<Extensão> ou no formato <IDElem>.RawHit[<Expressão da matriz>].<Extensão>. O Hit retorna dados compensados da sonda quando a compensação de sonda estiver ativada. RawHit sempre retorna dados não compensados. As extensões válidas são X, Y, Z, I, J, K, TX, TY, TZ, TI, TJ, TK, XYZ, TXYZ, IJK e TIJK

```
Círculo1.Hit[1].XYZ
```

O centróide medido (sonda compensada) de toque 1 de "Círculo1".

```
Círculo1.Hit[2].IJK
```

O vetor medido de toque 2 de "Círculo1"

Os dados de toque estão disponíveis para todos os objetos que possuem toques, independentemente de os toques reais estarem ou não exibidos na janela Edição. Portanto, os toques podem ser obtidos para varreduras e elementos automáticos.

Uso da matriz de toques para definir as entradas de elementos construídos

Você pode usar a matriz de toques para definir as entradas de elementos construídos.

Uso de expressões e variáveis

Para elementos construídos, quando você usa os métodos `feature.HIT[start..end]` ou `feature.HITS[start..end]`, as propriedades de início e fim são opcionais:

- Se o valor inicial não está definido, o PC-DMIS assume que o valor seja "1".
- Se o valor final não está definido, o PC-DMIS assume que o valor seja o número total de toques para o elemento.

Isso é o mesmo que usar "`feature.NUMHITS`".



Estes exemplos mostram três maneiras diferentes de usar os mesmos toques de CIR1:

Exemplo 1



Nesse exemplo, os valores inicial e final estão explicitamente definidos:



```
CIR3      =ELEM/CÍRCULO,CARTESIANO,INT,NÃO
           TEÓR/<20.97629,22.90352,0>,<0,0,-1>,20.97629
           ATUAL/<20.96578,22.89023,-0.01243>,<0,0,-
1>,20.96578
           CONSTR/CÍRCULO,REV,CIR1.TOQUE[1..CIR1.NUMTOQ
UES]
```

Exemplo 2



Nesse exemplo, somente o valor inicial de "1" está definido. Como o valor final não está definido, o PC-DMIS define o valor final como "`CIR1.NUMTOQUES`":



```
CIR3      =ELEM/CÍRCULO,CARTESIANO,INT,NÃO
           TEÓR/<20.97629,22.90352,0>,<0,0,-
1>,20.97629
           ATUAL/<20.96578,22.89023,-0.01243>,<0,0,-
1>,20.96578
           CONSTR/CÍRCULO,REV,CIR1.TOQUE[1..]
```

Exemplo 3

Nesse exemplo, os valores inicial e final não estão definidos. Devido a isso, o PC-DMIS define o valor inicial como "1" e o valor final como "CIR1.NUMTOQUES":



```
CIR3      =ELEM/CÍRCULO, CARTESIANO, INT, NÃO
           TEÓR/<20.97629, 22.90352, 0>, <0, 0, -
1>, 20.97629
           ATUAL/<20.96578, 22.89023, -0.01243>, <0, 0, -
1>, 20.96578
           CONSTR/CÍRCULO, REV, CIR1.TOQUE[. .]
```

As seções a seguir descrevem algumas funções de matriz adicionais úteis para localizar os pontos mínimo ou máximo em uma varredura:

Atribuição de um intervalo de toques a uma matriz

Um intervalo de toques pode ser atribuído a uma matriz usando esta sintaxe:

<ID Elemento>.<TipoToque>[<NúmInicial>..<NúmFinal>].<Extensão>

onde

<ID Elemento> é o nome do elemento.

<TipoToque> pode ser a palavra "TOQUE" para dados compensados ou "TOQUECRU" para dados não compensados. Se a compensação da sonda está desligada, os valores de retorno são sempre não compensados.

<NúmInicial> é uma expressão que identifica o primeiro valor de índice do intervalo de toques.

<NúmFinal> é uma expressão que identifica o segundo valor de índice do intervalo de toques.

<Extensão> identifica o tipo de dados. As extensões possíveis incluem os tipos de dados medidos ou teóricos listados na próxima seção.



Exemplo de como usar em uma expressão valores Inicial e Final implícitos.

Suponha que você deseja encontrar todos os pontos de toque que definem um plano (PLN1). A expressão pode ser escrita desta maneira:

```
PLN1.TOQUE[1..PLN1.NUMTOQUES]
```

Um modo mais curto de fazer isso é usar valores de toque Inicial e Final inferidos. O mesmo código pode ser escrito desta maneira:

```
PLN1.TOQUE[. .]
```

Nesse caso, como você não definiu os valores de toque Inicial e Final, o PC-DMIS assume que o valor de toque Inicial é 1 e que o valor de toque Final é NUMTOQUES.

Medido

- X – Valores X medidos dos toques
- Y – Valores Y medidos dos toques
- Z – Valores Z medidos dos toques
- XYZ – Valores XYZ medidos dos toques
- I – Valores I medidos dos toques
- J – Valores J medidos dos toques
- K – Valores K medidos dos toques
- IJK – Valores IJK medidos dos toques

Teórico

- TX – Valores X teóricos dos toques
- TY – Valores Y teóricos dos toques
- TZ – Valores Z teóricos dos toques
- TXYZ – Valores XYZ teóricos dos toques
- TI – Valores I teóricos dos toques
- TJ – Valores J teóricos dos toques
- TK – Valores K teóricos dos toques
- TIJK – Valores IJK teóricos dos toques

Por exemplo:

```
ATRIBUIR/V1=VARREDURA1.TOQUE[1..10].X
```

V1 é atribuído a uma matriz de 10 valores que são os valores X medidos dos primeiros 10 toques de VARREDURA1.

```
ATRIBUIR/V2=VARREDURA.TOQUE[1..VARREDURA1.NÚMTOQUES].XYZ
```

V2 é atribuído a uma matriz de pontos de cada um dos centróides dos toques na varredura.

Ordenação de matrizes

O PC-DMIS permite ordenar matrizes em ordem crescente ou decrescente. As duas expressões a seguir obtêm uma matriz e retornam uma matriz ordenada:

Para ordenar em ordem *crescente*, use:

```
SORTUP(<matriz>)
```

Para ordenar em ordem *descendente*, use:

```
SORTDOWN(<matriz>)
```

Por exemplo:

```
ATRIBUIR/V1=MATRIZ(5,8,3,9,2,6,1,7)
```

V1 é atribuído à matriz de "5,8,3,9,2,6,1,7".

```
ATRIBUIR/V2=ORDENARPCIMA(V1)
```

V2 conterá os valores da matriz ordenados em ordem crescente: "1,2,3,5,6,7,8,9"

```
ATRIBUIR/V3=ORDENARPBAIXO(V1)
```

V3 conterá os valores da matriz ordenados em ordem decrescente:

"9,8,7,6,5,3,2,1"

Retorno dos maiores ou menores valores de índice de uma matriz:

Uma matriz pode ser inserida em uma função e retornar o número do índice do elemento que tem o maior ou o menor valor utilizando estas funções:

Para retornar o valor do índice do elemento com o *maior* valor, use:

```
MAXINDEX(<matriz>)
```

Para retornar o valor do índice do elemento com o *menor* valor, use:

```
MININDEX(<matriz>)
```

Por exemplo,

```
ATRIBUIR/V1=MATRIZ(5,8,3,9,2,6,1,7)
```

V1 é atribuído à matriz de "5,8,3,9,2,6,1,7".

Uso de expressões e variáveis

`ATRIBUIR/V2=MAXINDICE (V1)`

V2 conterá o valor de índice da matriz de 4. O valor real desse elemento da matriz é 9.

`ATRIBUIR/V3=MININDICE (V1)`

V3 conterá o valor de índice da matriz de 7. O valor real desse elemento da matriz é 1.

Os valores de índice retornados podem então ser utilizados para obter o valor de elemento real da matriz.

Retorno de valores de índice ordenados de uma matriz

Uma matriz pode ser inserida em uma função, ordenar os valores da matriz em ordem crescente ou decrescente e, em seguida, retornar os valores de índice usando estas funções:

Retornar as posições de índice da matriz na ordem dos valores classificados do maior ao menor uso:

`MAXINDICES (<matriz>)`

Retornar as posições de índice da matriz na ordem dos valores classificados do menor ao maior uso:

`MININDICES (<matriz>)`

Por exemplo:

`ATRIBUIR/V1=MATRIZ (4, 8, 2, 9, 5, 7)`

V1 é atribuído à matriz de "4,8,2,9,5,7".

`ATRIBUIR/V2=MAXINDICES (V1)`

V2 conterá uma matriz com esses valores: "4,2,6,5,1,3".

`ATRIBUIR/V3=MININDICES (V1)`

V3 conterá uma matriz com esses valores: "3,1,5,6,2,4".

Exemplo da utilização de funções de matriz para encontrar os pontos mínimo e máximo em uma varredura

O principal objetivo das funções de matriz de toque discutidas acima é fornecer uma forma fácil de encontrar os pontos mínimo e máximo em uma varredura.

Para dimensionar o ponto de VARREDURA1 que tem o maior valor X medido, poderia usar esta expressão:



```
ATRIBUIR/MAXINDICEPT=MAXINDICE (VARREDURA1.TOQUE [1.
.VARREDURA1.NÚMTOQUES] .X)
D1=LOCALIZAÇÃO DO ELEMENTO VARREDURA1.TOQUE [MAXINDICEPT]
```

Para encontrar os três pontos mais altos no eixo Z de VARREDURA2, esta expressão poderia ser usada:



```
ATRIBUIR/MI=MAXINDICES (VARREDURA2.TOQYE [1.
.VARREDURA2.NÚMTOQUES] .Z)
ATRIBUIR/TRÊSPONTOS=MATRIZA (VARREDURA2.TOQUE [MI [1]
].XYZ,VARREDURA2.TOQUE [MI [2] ].XYZ,VARREDURA2.TOQUE [MI [3]
].XYZ)
```

Matrizes de variáveis

As matrizes de variáveis não precisam ser declaradas. As matrizes de variáveis passam a existir por meio da declaração de atribuição quando a expressão à direita da declaração de atribuição calcula uma matriz ou quando, à esquerda da instrução de atribuição, refere-se a um elemento em uma matriz de variáveis.

```
ATRIBUIR/V1=Matriz (3,4,5,6,7)
Cria uma matriz de 5 elementos e a atribui a V1.
```

```
ATRIBUIR/V2=V1 [3]
Atribui a V2 o valor do terceiro elemento da matriz V1: 5.
```

```
ATRIBUIR/V1 [4]=23
Atribui ao 4º elemento da matriz V1 o valor 23.
```

As matrizes são criadas e alocadas dinamicamente. Portanto, uma matriz pode ser criada usando-se uma referência de matriz à esquerda de uma declaração de atribuição.

```
ATRIBUIR/V3 [5]=8
Cria dinamicamente uma matriz com o 5º elemento definido como igual a 8.
```

Ao referir um elemento da matriz que nunca recebeu um valor, a expressão da matriz será calculada para 0.

```
ATRIBUIR/V3 [5]=8
```


Uso de expressões e variáveis

```
ATRIBUIR/V4=V3[5]
```

V4 é definido como o valor 8.

```
ATRIBUIR/V5=V3[6]
```

Se o sexto elemento de V3 nunca tiver sido definido, V5 É definido como 0.

Como outros tipos de matrizes, as expressões podem ser usadas entre colchetes.

```
ATRIBUIR/V3[5]=8
```

```
ATRIBUIR/V4=V3[2+3]
```

V4 é definido como o valor 8.

As matrizes de variáveis podem ter várias dimensões.

```
ATRIBUIR/V6=Matriz(Matriz(4,7,2),Matriz(9,2,6))
```

V6 é definida como uma matriz bi ou tridimensional, em que V6[1, 1] é igual a 4, V6[1, 2] é igual a 7, V6[1, 3] é igual a 2, V6[2, 1] é igual a 9, V6[2,2] é igual a 2 E V6[2,3] é igual a 6.

```
ATRIBUIR/V7=V6[2,1]
```

V7 é definido como o valor 9.

As matrizes de variáveis podem ter índices negativos.

```
ATRIBUIR/V8[-3]=5
```

O índice -3º da matriz V8 é definido para 5.

A atribuição da matriz substituirá os valores anteriores:

```
ATRIBUIR/V8="Olá"
```

A variável V8 é igual à sequência "Olá".

```
ATRIBUIR/V8[2]=5
```

V8 não é mais de tipo sequência, mas de tipo matriz, o segundo elemento que possui um valor 5.

```
ATRIBUIR/V8=9
```

V8 não é mais uma matriz, mas um inteiro com valor 9.

As matrizes podem ser constituídas de vários tipos:

```
ATRIBUIR/V9=Matriz("Olá",3,2.9,{ELEM1})
```

Cria uma matriz V9 com 4 elementos. O primeiro elemento é uma sequência, o segundo elemento é um inteiro, o terceiro elemento é um número real e o quarto elemento é um ponteiro para ELEM1.

As matrizes podem ser aumentadas em tamanho para incluir mais elementos:



```
ATRIBUIR/V10=ARRAY (3,1,5)
```

```
ATRIBUIR/V10[LEN(V10)+1]=7
```

A primeira instrução cria uma matriz inicial V10 com 3 elementos (3, 1 e 5). A segunda instrução aumenta a matriz em V10 por um elemento e dá ao elemento final um valor de 7.

Operadores para expressões

Os seguintes operadores básicos estão disponíveis no PC-DMIS:

+ Adição: *<Expressão> + <Expressão>*

Adiciona duas expressões juntas. No caso de sequências, elas são concatenadas.

- Subtração: *<Expressão> - <Expressão>*

Subtrai a segunda expressão da primeira.

* Multiplicação: *<Expressão> * <Expressão>*

Multiplica duas expressões.

/ Divisão: *<Expressão> / <Expressão>*

Divide a primeira expressão pela segunda.

^ Exponenciação: *<Expressão> ^ <Expressão>*

Eleva a primeira expressão à potência da segunda.

% Módulo: *<Expressão> % <Expressão>*

Retorna o resto da divisão de uma expressão por outra.

- Inverso aditivo-*<Expressão>*

Retorna o inverso aditivo da expressão.

Uso de expressões e variáveis

! Não lógico: *!<Expressão>*

Retorna o não lógico da expressão.

== Igual a: *<Expressão> == <Expressão>*

Retornar o valor 1 se as expressões forem iguais. Caso contrário, retorna 0. (Dois sinais de igual são usados para diferenciar do operador de atribuição = na declaração da atribuição).

<> Não igual a: *<Expressão> <> <Expressão>*

Retorna o valor 1 se as expressões não forem iguais. Caso contrário, retorna 0.

> Maior que: *<Expressão> > <Expressão>*

Retorna o valor 1 se a primeira expressão for maior que a segunda. Caso contrário, retorna 0.

>= Maior ou igual a: *<Expressão> >= <Expressão>*

Retorna o valor 1 se a primeira expressão for maior ou igual à segunda. Caso contrário, retorna 0.

< Menor que: *<Expressão> < <Expressão>*

Retorna o valor 1 se a primeira expressão for menor que a segunda. Caso contrário, retorna 0.

>= Menor ou igual a: *<Expressão> >= <Expressão>*

Retorna o valor 1 se a primeira expressão for menor ou igual à segunda. Caso contrário, retorna 0.

AND E lógico: *<Expressão> AND <Expressão>*

Retorna o valor 1 se ambas as expressões não forem avaliadas como 0. Caso contrário, retorna 0.

OR Ou Lógico: *<Expressão> OR <Expressão>*

Retorna o valor 1 se nenhuma das expressões for avaliada como 0. Caso contrário, retorna 0.

() Parênteses: *(<Expressão>)*

Atribui precedência de cálculo para a expressão dentro dos parênteses.

Precedência

As expressões são avaliadas com a precedência mostrada abaixo (relacionada da maior precedência para a menor).

Maior precedência

- Operandos
- (menos unário), !, (), funções (como ABS, COS, STR, LEN, CROSS etc.)
- ^
- *, /, %
- +, -
- ==, <>, <, <=, >, >=
- E
- OR

Menor precedência

Funções

Funções são expressões específicas do PC-DMIS ou expressões definidas pelo usuário que geralmente usam parâmetros e depois retornam resultados. Os parâmetros são substituídos pela expressão antes de a expressão ser avaliada.

Lista de funções

A lista alfabética a seguir contém todas as funções disponíveis para a linguagem de expressão do PC-DMIS.

- ABS (matemático)
- ACOS (matemático)
- ANGLEBETWEEN (ponto)
- ARCSEGMENTENDINDEX (diversos)
- ARCSEGMENTSTARTINDEX (diversos)
- ARRAY (matriz)
- ASIN (matemático)
- ATAN (matemático)
- CHR (sequência)
- CONCAT (sequência)

Uso de expressões e variáveis

- COS (matemático)
- CROSS (ponto)
- DEG2RAD (matemático)
- DELTA (ponto)
- DIST2D (ponteiro)
- DIST3D (ponteiro)
- DOT (ponto)
- ELEMENT (sequência)
- EOF (diversos)
- EOL (diversos)
- EQUAL (matriz)
- EQUAL (sequência)
- EXP (matemático)
- FORMAT (sequência)
- FUNCTION (função)
- GETCOMMAND (ponteiro)
- GETPROGRAMINFO (sequência)
- GETROTABDATA (diversos)
- GETSETTING (sequência)
- GETTEXT (sequência)
- GETTRACEVALUE (sequência)
- IF (diversos)
- INDEX (sequência)
- ISIOCHANNELSET (diversos)
- LEFT (sequência)
- LEN (matriz)
- LEN (ponteiro)
- LEN (sequência)
- LINESEGMENTENDINDEX (diversos)
- LINESEGMENTSTARTINDEX (diversos)
- LN (matemático)
- LOG (matemático)
- LOWERCASE (sequência)
- MAX (matriz)
- MID (sequência)
- MIN (matriz)
- MPOINT (ponto)
- ORD (sequência)
- PCDMISAPPLICATIONPATH (sequência)
- PCDMISUSERHIDDEN DATAPATH (sequência)

- PCDMISUSERVISIBLEDATAPATH (sequência)
- PCDMISSYSTEMHIDDENDATAPATH (sequência)
- PCDMISSYSTEMVISIBLEDATAPATH (sequência)
- PCDMISSYSTEMREPORTINGPATH (sequência)
- PROBADATA (diversos)
- QUALTOOLDATA (diversos)
- RAD2DEG (matemático)
- RIGHT (sequência)
- ROUND (matemático)
- SETROTABDATA (diversos)
- SIN (matemático)
- SQRT (matemático)
- SYSTEMDATE (sequência)
- SYSTEMTIME (sequência)
- SYSTIME (sequência)
- TAN (matemático)
- TUTORELEMENT (diversos)
- UNIT (ponto)
- UPPERCASE (sequência)

Funções de sequências

As funções a seguir são usadas com cadeias de texto.

CHR

Conversão de caracteres: *CHR(<inteiro>)*

Essa função retorna uma sequência, que consiste no caractere correspondente ao valor decimal ASCII.

CONCAT

Esta função dispõe em cadeia todas as sequências especificadas nas expressões 1 a N em uma só sequência: *CONCAT(<expressão1>, <expressão2>, & <expressãoN>)*

ELAPSEDEXECUTIONTIME

Tempo de execução decorrido formatado: *ELAPSEDEXECUTIONTIME()*

Essa função retorna o tempo transcorrido desde que a rotina de medição ou mini-rotina começou a ser executada. O tempo de execução transcorrido é o tempo gasto durante a parte DCC da execução, não sendo controlado o tempo para pausas devido à atenção requerida pelo usuário. Tais pausas incluem pausas de execução durante execução de comentários ou mensagens do PC-DMIS, e mensagens de erro que podem parar totalmente a execução.

Você gravar o tempo de execução decorrido em qualquer ponto da rotina de medição ou mini-rotina atribuindo a função a uma variável, tal como:



```
ATRIBUIR/V1=ELAPSEDEXECUTIONTIME ( )
```

O formato padrão do tempo retornado é "hh:mm:ss". Você também pode medir o tempo de execução transcorrido em outros formatos:

- Use ATRIBUIR/V1=FORMAT(ELAPSEDEXECUTIONTIME(),"hh:mm:ss") ou ASSIGN/V1=ELAPSEDEXECUTIONTIME() para obter o tempo em horas, minutos e segundos.
- Use ATRIBUIR/V1=FORMAT(ELAPSEDEXECUTIONTIME(),"mm:ss") ou ASSIGN/V1=ELAPSEDEXECUTIONTIME() para obter o tempo em minutos e segundos.
- Use ATRIBUIR/V1=FORMAT(ELAPSEDEXECUTIONTIME(),"ss") para obter o tempo em segundos.

ELEMENT

Local da subsequência delimitada: *ELEMENT(<Inteiro>, <Sequência1>, <Sequência2>)*

Essa função retorna a enésima subsequência (elemento) para a sequência2 usando a sequência1 como o texto delimitador que divide os elementos na sequência2.



Suponha que a sequência2 é "6, 12, 8, 4, 5" e a sequência1 é um caractere de vírgula ",". Os cinco elementos que podem ser recuperados individualmente com o comando do elemento são "6", "12", "8", "4" e "5".

EQUAL

Comparação de sequência não sensível a maiúsculas e minúsculas:

EQUAL(<Sequência1>, <Sequência2>)

Essa função compara duas sequências (ignorando maiúsculas e minúsculas) para determinar se elas são idênticas. Ela retorna um inteiro definido como 1 se as sequências são iguais, e 0 se não são iguais.

FORMAT

Formato: *FORMATO(<Sequência>,<Inteiro, duplo ou ponto>)*

Essa função usa duas expressões e retorna uma sequência formatada, semelhante à utilização da função *sprintf* do C++.

- A expressão 1 deve ser do tipo *sequência* e conter um ou três especificadores de formato. Se for de um tipo diferente, o avaliador da expressão tentará forçá-la para uma sequência. A sequência deve conter *um* especificador de formato se a Expressão 2 for dos tipos inteiro ou duplo e *três* especificadores de formato (consulte os parágrafos a seguir) se a Expressão 2 for do tipo ponto.
- Espera-se que a expressão 2 seja do tipo *inteiro*, *duplo* ou *ponto*. Se um tipo diferente for utilizado, o valor da expressão será 0.

Especificador de formato para a função FORMAT:

O especificador de formato deve ter a mesma sintaxe que o especificador de formato utilizado na função *sprintf* utilizada na linguagem de programação C++.

Um especificador de formato consiste em campos opcionais e obrigatórios, com a seguinte sintaxe:

`%[sinalizadores] [comprimento] [.precisão] tipo`

Cada campo do especificador de formato é um único caractere ou um número que significa uma opção de formato específica. O especificador de formato mais simples usa somente o sinal de porcentagem e um caractere de tipo (por

exemplo, %d). Se um sinal de porcentagem for seguido de um caractere que não tenha significado como um campo de formato, o caractere é copiado para STDOUT. Por exemplo, para imprimir um caractere de sinal de porcentagem, use %%.

O campos opcionais sinalizador, comprimento e precisão, que aparecem antes do caractere de tipo, controlam outros aspectos da formação. Eles são descritos a seguir:

sinalizadores

Estes *caracteres opcionais* controlam a justificação da saída e a impressão de sinais, espaços em branco, separadores decimais e prefixos octais ou hexadecimais. Mais de um sinalizador pode aparecer em um especificador de formato.

Aqui estão os sinalizadores possíveis:

–

Significado: Alinha o resultado à esquerda dentro da largura de campo especificada.

Padrão: Alinha à direita.

+

Significado: Prefixa o valor de saída com um sinal (+ ou –) se o valor de saída for do tipo com sinal.

Padrão: O sinal aparece somente para valores com sinal negativo (–).

0

Significado: Se a largura estiver prefixada com um 0, são adicionados zeros até que a largura mínima seja alcançada. Se 0 e – aparecerem, o 0 é ignorado. Se 0 for especificado com um formato inteiro (i, u, x, X, o, d) o 0 é ignorado.

Padrão: Sem preenchimento.

espaço em branco (' ')

Significado: Prefixa o valor da saída com um espaço em branco se o valor da saída tiver sinal e for positivo; o espaço em branco é ignorado se os sinalizadores espaço em branco e + aparecerem.

Padrão: Nenhum espaço em branco aparece.

#

Significado 1: Quando utilizado com o tipo o, x ou X, o sinalizador # prefixa qualquer valor de saída diferente de zero com um 0, 0x ou 0X, respectivamente.

Padrão 1: Nenhum prefixo aparece.

Significado 2: Quando utilizado com o tipo e, E ou f, o sinalizador # força que o valor da saída contenha uma vírgula decimal em todos os casos.

Padrão 2: A vírgula decimal aparece somente se houverem dígitos após ela.

Significado 3: Quando utilizado com o formato g ou G, o sinalizador # força que o valor da saída contenha uma vírgula decimal em todos os casos e impede o truncamento de zeros finais.

Padrão 3: A vírgula decimal aparece somente se houverem dígitos após ela. Os zeros finais são truncados. Isso é ignorado quando utilizado com d, i ou u.

largura

Esse segundo campo, ou argumento, opcional controla o número mínimo de caracteres impressos. Esse é um número inteiro decimal não negativo.

- Se o número de caracteres no valor de saída for menor que a largura especificada, espaços em branco serão adicionados à esquerda ou à direita dos valores — dependendo de se o indicador — (para alinhamento à esquerda) tiver sido especificado — até a largura mínima ser atingida.
- Se o comprimento for prefixado com 0, serão adicionados zeros até que o comprimento mínimo seja alcançado (não é útil para números alinhados à esquerda).
- A especificação de comprimento nunca faz com que um valor seja truncado. Se o número de caracteres no valor da saída for maior do que o comprimento especificado ou se o comprimento não for fornecido, todos os caracteres do valor serão impressos (sujeito à especificação de precisão listada a seguir).

precisão

Esse terceiro campo, ou argumento, opcional especifica o número de caracteres a serem impressos, o número de casas decimais ou o número de dígitos significativos. À diferença da especificação de largura, a especificação de precisão pode causar o truncamento do valor da saída ou o arredondamento de um valor de ponto flutuante. Esse é um número inteiro decimal não negativo, precedido por um ponto (.).

tipo

Esse caractere obrigatório determina se o argumento associado é um número inteiro, um duplo ou um ponto. A lista de tipos disponíveis inclui os seguintes:

d - inteiro decimal com sinal

i - inteiro decimal com sinal

o - inteiro octal sem sinal

u - inteiro decimal sem sinal

x - inteiro hexadecimal sem sinal, utilizando "abcdef"

X - inteiro hexadecimal sem sinal, utilizando "ABCDEF"

e - duplo no formato exponencial [-]d.dddd e [sinal]ddd

E - o mesmo que e, exceto que utiliza E para introduzir o expoente

f - duplo com o formato [-]dddd.dddd

g - formata para o formato e ou f dependendo do que for mais compacto

G - o mesmo que g, exceto que utiliza E para introduzir o expoente

Exemplo de FORMAT

Este exemplo mostra diversas declarações que utilizam a função FORMATO dentro de uma rotina de medição:

ATRIBUIR/V1=PROBEDATA("DESLOCAMENTO")	V1 se torna ponto de tipo representando os deslocamentos da sonda atual. Utilizando os valores da rotina de medição utilizados nesse exemplo, V1 se torna: <-1,8898, 1,8898, 5,704>
ATRIBUIR/V3=FORMATO("%.5f,%.5f,%.5f",V1)	V3 se torna um tipo sequência. A sequência é formatada utilizando o objeto ponto da variável V1. V3 agora tem - 1,88976; 1,88976; 5,70403
ATRIBUIR/V4=1.123456789	V4 se torna um tipo duplo.
ATRIBUIR/V5=FORMATO("%.5f",V4)+FORMATO("%.6f",V4)+FORMATO("%.7f",V4)+FORMATO("%.8f",V4)	V5 se torna um tipo sequência com este valor: 1,12346 1,123457 1,1234568 1,12345679
ATRIBUIR/V6A="O valor de V4 é: "+FORMATO("%.8f",V4)	V6A se torna um tipo sequência com o valor: O valor de V4 é: 1,12345679
ATRIBUIR/V6B=FORMATO("O valor de V4 é: %.8f",V4)	O resultado da expressão permanece o mesmo que o V6A acima.
ATRIBUIR/V7=4444	V7 se torna um tipo duplo pois todos os números são assumidos como duplos a menos que

	sejam forçados para um inteiro.
ATRIBUIR/V8=FORMATO("%o",INT(V7))	V8 se torna um tipo sequência com este valor: 10534
ATRIBUIR/V9=FORMATO("%u",INT(-1))	V9 se torna um tipo sequência com este valor: 4294967295
ATRIBUIR/V10=FORMATO("%x",INT(2143))	V10 se torna um tipo sequência com este valor: 85f
ATRIBUIR/V11=FORMATO("%X",INT(9567))	V11 se torna um tipo sequência com este valor: 255F
ATRIBUIR/V12=FORMATO("%e",0.0005432)	V12 se torna um tipo sequência com este valor: 5,432000e-004
ATRIBUIR/V13=FORMATO("%E",145.3421)	V13 se torna um tipo sequência com este valor: 1,453421E+002
ATRIBUIR/V14=FORMATO(",%6d,",INT(1))	V14 se torna um tipo sequência com este valor: , 1,
ATRIBUIR/V15=FORMATO(",%-6d,",INT(1))	V15 se torna um tipo sequência com este valor: , 1,

GETSETTING

Essa função permite que você retorne diversas configurações do PC-DMIS, dependendo do parâmetro de sequência inserido.

GETSETTING(<Sequência>)

Estes parâmetros de sequência podem ser utilizados:

- "DCC Mode" – Retorna 1 se o PC-DMIS estiver no Modo DCC, e 0 caso contrário.

- "Manual Mode" – Retorna 1 se o PC-DMIS estiver no Modo Manual, e 0 caso contrário.
- "Current Alignment" – Retorna uma sequência do alinhamento atual.
- "Current Workplane" – Retorna uma sequência do plano de trabalho atual.
- "Workplane Value" – Retorna um valor numérico do plano de trabalho atual.
- "PreHit" – Retorna o valor de pré-toque atual como um número de precisão dupla.
- "Retract" – Retorna o valor de retração atual como um número de precisão dupla.
- "Check" – Retorna o valor de verificação atual como um número de precisão dupla.
- "Touch Speed" – Retorna o valor da velocidade de toque atual como um número de precisão dupla.
- "Move Speed" – Retorna o valor de velocidade de movimento como um número de precisão dupla.
- "Fly Mode" – Retorna 1 se o PC-DMIS utilizar o Modo Fly, e 0 caso contrário.
- "Ph9 present" – Retorna 1 se o Ph9/Ph10 estiver presente, e 0 caso contrário.
- "Manual CMM" – Retorna 1 se a CMM for uma CMM manual, e 0 caso contrário.
- "LangStr(<Número ou ID>)" – Retorna uma sequência a partir dos recursos do PC-DMIS no idioma em uso, usando um número de ID de recurso ou de uma destas IDs:

"Yes", "No", "Oper", "Rept", "Input", "Doc", "YesNo", "Readout", "Internal", "External", "Rect ", "Polr ", "Out", "In", "Least_Sqr", "Min_Sep", "Max_Insc", "Min_CircSc", "Fixed_Rad", "Workplane", "Xaxis", "YAxis", "ZAxis", "Xplus", "Xminus", "YPlus", "YMinus", "ZPlus", "ZMinus", "Point", "Plane", "Line", "Circle", "Sphere", "Cylinder", "Round_Slot", "Square_slot", "Cone", ou "None".

Se o valor utilizado for um número positivo, o PC-DMIS retira a sequência de seu arquivo resource.dll. Se utilizar um valor negativo, o PC-DMIS retira a sequência de seu arquivo strings.dll (a tabela de sequências).

- "Extended Sheet Metal" - Retorna 1 se a caixa de seleção **Mostrar opções de chapa metálica estendida** estiver marcada dentro da caixa de diálogo **Opções de configuração**, e 0 caso contrário.

- "LastHitMove(X)" – Retorna o valor X do comando TOQUE /BÁSICO ou MOVER/PONTO mais recente. O PC-DMIS precisa estar no modo DCC para que isso funcione.
- "LastHitMove(Y)" – Retorna o valor Y do comando TOQUE /BÁSICO ou MOVER/PONTO mais recente. O PC-DMIS precisa estar no modo DCC para que esse parâmetro funcione.
- "LastHitMove(Z)" – Retorna o valor Z do comando TOQUE /BÁSICO ou MOVER/PONTO mais recente. O PC-DMIS precisa estar no modo DCC para que esse parâmetro funcione.

Você pode usar a função GETSETTING para determinar se o PC-DMIS está no modo Manual ou DCC, como mostrado abaixo:

`ATRIBUIR/MODOVARDCC=GETSETTING("Modo DCC")` - Isso fornece à variável MODOVARDCC o valor 1 se o PC-DMIS estiver no modo DCC, caso contrário, fornece 0.

`ATRIBUIR/MODOVARMAN=GETSETTING("Modo Manual")` - Isso fornece à variável MODOVARMAN o valor 1 se o PC-DMIS estiver no modo Manual, caso contrário, fornece 0.

Você pode usar a função GETSETTING para determinar o plano de trabalho atual, como mostrado abaixo:

`ATRIBUIR/PLANOTRABALHO_ID=GETSETTING("Plano de trabalho atual")` - Isso oferece à variável PLANOTRABALHO_ID o valor de sequência do plano de trabalho atual (ZMAIS, ZMENOS etc.).

`ATRIBUIR/PLANOTRABALHO_VALOR=GETSETTING("Valor do plano de trabalho")` - Isso oferece à variável PLANOTRABALHO_VALOR um valor numérico representando o plano de trabalho. Os planos de trabalho possuem estes valores associados a eles: ZMAIS = 0, ZMENOS = 3, XMAIS = 1, XMENOS = 4, YMAIS = 2 ou YMENOS = 5.

GETTEXT

Essa função retorna o texto atual de um campo de dados específicos:

`GETTEXT(<Sequência ou Inteiro>, <Inteiro>, <Ponteiro>)`

Essa função tem três campos.

Primeiro campo — Número ou descrição do campo de dados

O primeiro campo poderá ser tanto uma descrição de sequência do campo de dados, indicado pelo item (A) na imagem abaixo ou o número do campo de dados, indicado no item (C) na imagem abaixo.



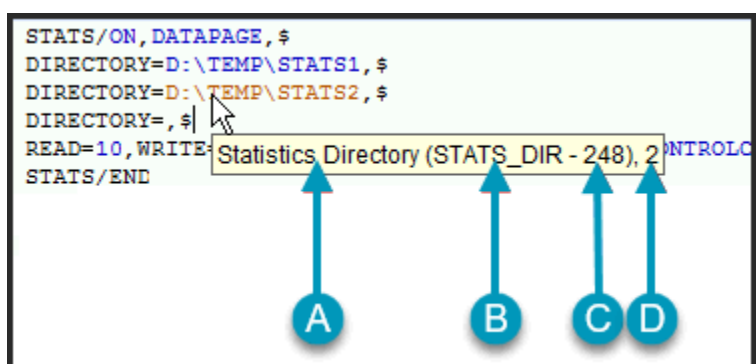
O item (B) na imagem abaixo não é usado nessa função, mas é usado às vezes em automação ou expressões de relatório.

Para obter esses valores:

1. Coloque o PC-DMIS no Modo Comando. Clique com o botão direito do mouse em qualquer lugar da Janela Edição. Um menu de atalho aparecerá.
2. No menu de atalho, selecione **Alterar exibição pop-up** e, em seguida, **Informações sobre tipo de dados**.
3. Posicione o mouse sobre um campo de dados da janela Edição. São exibidos a descrição, o número e o índice do tipo desse item de dados.



O tipo de descrição pode ser diferente para idiomas diferentes. Se a sua rotina de medição é executada em uma versão de PC-DMIS executada em um idioma diferente, use o número do tipo.



Informações sobre o tipo de dados de amostra mostrando (A) Descrição do tipo (B) Identificador da sequência do tipo, (C) Número do tipo e (D) Índice do tipo

Segundo campo — Índice de tipo

O segundo campo é o índice do tipo, indicado como (D) na imagem acima. Esse campo é normalmente igual a zero, a não ser que haja mais ocorrências do mesmo tipo de campo no mesmo comando, como vários campos de DIRETÓRIO mostrados na imagem acima. O valor correto para esse campo pode ser obtido da mesma maneira, conforme descrito no primeiro campo.

Terceiro campo — Ponteiro de comando

O terceiro campo é um ponteiro do comando. Ele aponta para o comando que contém o campo a partir do qual o texto está sendo obtido. Você pode usar tanto a notação de ponteiro de comando (ou seja, {F15}) para especificar esse campo ou usar a expressão GETCOMMAND, como mostrado abaixo:



`ATRIBUIR/V1=GETTEXT("Método de melhor ajuste", 0, {F15})` - Esse comando atribui a V1 o valor atual do método de melhor ajuste do elemento F15.

`ATRIBUIR/V2=GETCOMMAND("Comentário", "TOPO", 1)` - V2 é atribuída um ponteiro ao primeiro comentário a partir da parte superior da rotina de medição.

`ATRIBUIR/V3=GETTEXT("Tipo de comentário", 1, V2)` - V3 é atribuído o valor do campo de alternância Tipo de comentário. Se o primeiro comentário da rotina de medição for um comentário a ser exibido para o operador, o valor de V3 será a sequência "OPER".

Consulte as "Funções de ponteiro" para obter mais informações sobre a expressão GETCOMMAND usada para configurar um ponteiro a um comando.

GETTEXTEX

Essa função retorna o texto atual de um campo de dados específicos:

`GETTEXTEX(<Sequência ou Inteiro>, <Inteiro>, <Sequência>, <Ponteiro>)`

Essa função tem quatro campos.

Primeiro campo — Número ou descrição do campo de dados

O primeiro campo poderá ser tanto uma descrição de sequência do campo de dados, indicado pelo item (A) na imagem abaixo.



Se você usar o identificador da sequência em vez do identificador numérico (Item (A) na imagem abaixo), o PC-DMIS o converte automaticamente para o valor numérico correto.

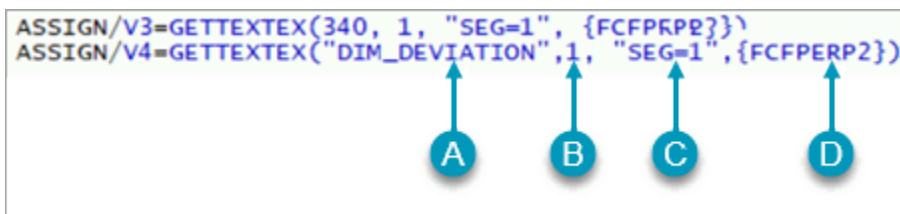
Por exemplo, se você passa no identificador de sequência "DIM_DEVIATION", internamente, o PC-DMIS o converte para o valor numérico 340. Quando então você passa o ponteiro do mouse sobre o comando na janela Edição, o comando pop-up mostra a sequência de texto e também o valor do identificador numérico. Ou sejam nesse exemplo, se você passa o cursor sobre o comando na janela Edição, o comando pop-up mostra (DIM_DEVIATION - 340), 1, SEG=1.

Você também pode passar o cursor somente sobre o valor numérico, se conhecido.

Para obter esses valores:

1. Coloque o PC-DMIS no modo Comando e clique com o botão direito do mouse em qualquer lugar da janela Edição. Um menu de atalho aparecerá.
2. No menu de atalho, selecione **Alterar exibição pop-up** e, em seguida, **Informações sobre tipo de dados**.
3. Posicione o mouse sobre um campo de dados da janela Edição. São exibidos a descrição, o número e o índice do tipo desse item de dados. Passe o cursor sobre um Tipo_D estendido para mostrar a sequência de conteúdo depois da vírgula.

O tipo de descrição pode ser diferente para idiomas diferentes. Se a sua rotina de medição é executada em uma versão de PC-DMIS executada em um idioma diferente, use o número do tipo.



Informações sobre o tipo de dados de amostra mostrando (A) Identificador de sequência de tipo ou numérico, (B) Índice de tipo, (C) Sequência de conteúdo e (D) Ponteiro de comando

Segundo campo - Índice de tipo

O segundo campo é o índice do tipo, indicado como B na imagem acima. Esse campo é normalmente igual a zero, a não ser que haja mais ocorrências do mesmo tipo de campo no mesmo comando, como vários campos de DIRETÓRIO mostrados na imagem acima. O valor correto para esse campo pode ser obtido da mesma maneira descrita para o primeiro campo.

Terceiro campo - Sequência de conteúdo

O terceiro campo é a sequência de conteúdo do TIPO_D estendido, indicado como C na imagem acima.

Quarto campo - Ponteiro de comando

O quarto campo é um ponteiro de comando, indicado como D na imagem acima. Ele aponta para o comando do qual a expressão extrai dados. Você pode usar tanto a notação de ponteiro de comando (ou seja, {FCFPERP2}) quanto a expressão GETCOMMAND, como mostrado abaixo:

```
ASSIGN/V1=GETTEXT("DIM_DEVIATION",1,"SEG=1",{FCFPERP2})
```

- Esse comando atribui V1 como o valor atual de desvio para o elemento 1, segmento 1, dimensão FCFPERP2.

Consulte as "Funções de ponteiro" para obter mais informações sobre a expressão GETCOMMAND usada para configurar um ponteiro a um comando.



A função GETTEXT adiciona suporte aos Tipos_D estendidos que contêm uma sequência CONTENT. Atualmente, somente os comandos de tolerância geométrica do PC-DMIS usam Tipos_D estendidos.

GETPROGRAMINFO

Essa função retorna as informações da rotina de medição baseadas nos parâmetros transferidos em: `GETPROGRAMINFO(<Sequência>, <Sequência opcional>)`

Essa função tem no máximo duas sequências como parâmetros. Para a maioria dos itens, você somente necessita do primeiro parâmetro. Os campos de sequência não são sensíveis a maiúsculas e minúsculas.

Primeiro campo—Sequência

O primeiro campo é uma entrada de sequência detalhando as informações a retornar.

CADMODELFILE - Retorna o caminho completo para o nome do arquivo do modelo do CAD que você importou na rotina de medição.

CADMODELFILENAME - Retorna somente o nome do modelo do CAD (não o caminho) que você importou na rotina de medição.

DATE - Retorna a data atual.

DRAWING - Do mesmo modo que REVISION, retorna o número de revisão conforme definido no cabeçalho.

ELAPSEDTIME - Retorna o tempo passado desde o início da execução.

FILENAME - Retorna o nome do arquivo da rotina de medição (.prg).

NUMMEAS - Retorna o número de dimensões executadas.

NUMOOT - Retorna o número de dimensões fora da tolerância executadas.

PARTNAME - Retorna o nome da peça conforme definido no cabeçalho da rotina de medição.

PARTPATH - Retorna o caminho completo para o arquivo da rotina de medição.

PCDMISVERSION - Retorna um valor de sequência da versão real do software do PC-DMIS instalado.

PRGSCHEMA - Retorna um inteiro do número do esquemático do PC-DMIS do arquivo da rotina de medição. Este é um valor interno usado pelo PC-DMIS para indicar os comandos e as opções que são colocados em série.

PRGVERSION - Retorna um valor de sequência do número da versão do PC-DMIS do arquivo da rotina de medição. Você pode salvar um arquivo de rotina de medição para ser compatível com uma versão específica. Para mais informações, consulte "Salvar como" no capítulo "Uso de opções básicas de arquivo".

ARQUIVOSONDA - Retorna o nome do arquivo de sonda atualmente em uso.

REPORTNAME - Retorna o nome do arquivo de saída atual.

REVISION - Retorna o número de revisão conforme definido no cabeçalho.

SERIALNUM - Retorna o número de série conforme definido no cabeçalho.

SEQNUM - Do mesmo modo que STATSCOUNT, essa sequência retorna a contagem estatística atual.

SHRINK - Retorna o fator da escala global.

STATSCOUNT - Retorna a contagem estatística atual.

TEMP - Retorna a temperatura para a segunda sequência de entrada opcional. Consulte "Segundo campo—Sequência opcional" abaixo.

TIME - Retorna a hora atual.

IDPONTA - Retorna o nome da ponta atualmente em uso.

Segundo campo—Sequência opcional

O segundo campo é uma entrada de sequência opcional. É necessário somente se TEMP for usado no primeiro campo de entrada. As possíveis sequências abaixo resultam do comando Compensação de temperatura. Para obter mais informações, consulte "Compensação de temperatura" no capítulo "Configuração de preferências".

HIGH_THRESHOLD - Retorna a temperatura limite superior

LOW_THRESHOLD - Retorna a temperatura limite inferior

REF_TEMP - Retorna a temperatura de referência

TEMPP - Retorna a temperatura para o sensor da peça

TEMPX - Retorna a temperatura para o sensor do eixo X

TEMPY - Retorna a temperatura para o sensor do eixo Y

TEMPZ - Retorna a temperatura para o sensor do eixo Z

Exemplo



```
$$ NO, Este código exemplo exibe o número de dimensões  
totais e o número de dimensões fora da tolerância.  
ATRIBUIR/V1=GETPROGRAMINFO("MÉDNÚM")  
ATRIBUIR/V2=GETPROGRAMINFO("NÚMFORATOL")  
COMENTÁRIO/RELATÓRIO  
"Dimensões totais: "+V1  
"Fora da tolerância total: "+V2  
$$ NÃO, Este exemplo de código retorna a temperatura no  
eixo do sensor Z.  
ATRIBUIR/V3=GETPROGRAMINFO("TEMP", "TEMPZ")  
COMENTÁRIO/RELATÓRIO  
"Temperatura no eixo Z: "+V3
```

GETTRACEVALUE

Ler valor de rastreamento: *GETTRACEVALUE(<sequência>)*

Essa função usa um parâmetro de sequência único. Ele retorna um valor de um comando *CAMPORASTREAMENTO* na rotina de medição.

<sequência> representa uma sequência com maiúsculas e minúsculas do nome do rastreamento cujo valor você deseja retornar.

Se você tem vários campos de rastreamento com o mesmo nome de rastreamento, essa função retorna o valor do campo de rastreamento mais recente acima dessa função. Se um campo de rastreamento não contém um valor, essa função retorna um valor de 0. No exemplo abaixo, "Operador" é o nome do campo de rastreamento na rotina de medição:



```
ATRIBUIR/V2=GETTRACEVALUE("Operador")
```

INDEX

Local da subsequência: *INDEX(<Sequência>, <Sequência>)*

Essa função retorna o local da segunda sequência dentro da primeira sequência. A primeira letra da sequência é 1. Um valor de retorno de zero indica que a subsequência não foi encontrada na sequência.

Para ver um exemplo dessa função, consulte o tópico Exemplo de código para a linha de leitura no capítulo "Utilização do arquivo de entrada/saída".

LASTEXECUTIONTIME

Último tempo de execução formatado: *LASTEXECUTIONTIME()*

Esta função retorna o último tempo de execução que o PC-DMIS gravou e armazenou no arquivo *<nome da rotina de medição>.MiniRoutines.xml*. O último tempo de execução aparece na caixa de diálogo **Execução**. O tempo é retornado no formato "hh:mm:ss".

LEFT

Número de caracteres à esquerda de uma sequência: *LEFT(<Sequência>, <n>)*

Essa função retorna uma sequência que consiste de um número de caracteres mais à esquerda, especificados pela segunda expressão (n) a partir da sequência especificada na primeira expressão (Sequência).

A primeira expressão (Sequência) é forçada a digitar em sequência. A segunda expressão (n) é forçada a digitar em inteiro.

Para ver um exemplo dessa função, consulte o tópico Exemplo de código para a linha de leitura no capítulo "Utilização do arquivo de entrada/saída".

LEN

Comprimento da sequência: *LEN(<Sequência>)*

Essa função retorna o número de caracteres da sequência.

LOWERCASE

Cria uma sequência em letras minúsculas: *LOWERCASE(<Sequência>)*

Essa função retorna uma sequência que é o equivalente em letras minúsculas da sequência da expressão.

MID

n caracteres no meio de uma sequência: *MID(<Sequência>, <Inteiro>, <Inteiro opcional>)*

Retorna uma subsequência que consiste em caracteres da sequência especificada no primeiro parâmetro, começando na posição especificada pelo segundo parâmetro para um comprimento de n caracteres, conforme especificado pelo terceiro parâmetro. Se o terceiro parâmetro não for fornecido, o resto da sequência é retornado.

Para um exemplo desta função, consulte o tópico Código exemplo para a linha de leitura no capítulo "Utilização do arquivo de entrada / saída".

ORD

Conversão de ordinais: *ORD(<Sequência>)*

Retorna o valor inteiro de ASCII da primeira letra da sequência.

PCDMISAPPLICATIONPATH

Exibição de caminho completo: *PCDMISAPPLICATIONPPATH()*

Essa função retorna o valor da sequência contendo o caminho completo para o diretório do aplicativo onde o PC-DMIS está instalado. O diretório contém os arquivos executáveis principais e de outros programas necessários para executar o PC-DMIS.

PCDMISUSERHIDDENATAPATH

Exibição de caminho completo: *PCDMISUSERHIDDENATAPATH()*

Essa função retorna o valor da sequência contendo o caminho completo para o diretório oculto de dados do usuário usado pelo PC-DMIS. Consulte "Compreensão dos locais de arquivos" para obter os arquivos contidos neste diretório.

PCDMISUSERVISIBLEATAPATH

Exibição de caminho completo: *PCDMISUSERHIDDENATAPATH()*

Essa função retorna o valor da sequência contendo o caminho completo para o diretório visível de dados do usuário usado pelo PC-DMIS. Consulte "Compreensão dos locais de arquivos" para obter os arquivos contidos neste diretório.

PCDMISSYSTEMHIDDEN DATAPATH

Exibição de caminho completo: PCDMISSYSTEMHIDDEN DATAPATH()

Essa função retorna o valor da sequência contendo o caminho completo para o diretório oculto de dados do sistema usado pelo PC-DMIS. Consulte "Compreensão dos locais de arquivos" para obter os arquivos contidos neste diretório.

PCDMISSYSTEMVISIBLE DATAPATH

Exibição de caminho completo: PCDMISSYSTEMVISIBLE DATAPATH()

Essa função retorna o valor da sequência contendo o caminho completo para o diretório visível de dados do sistema usado pelo PC-DMIS. Consulte "Compreensão dos locais de arquivos" para obter os arquivos contidos neste diretório.

PCDMISSYSTEMREPORTINGPATH

Exibição de caminho completo: PCDMISSYSTEMREPORTINGPATH()

Essa função retorna o valor da sequência contendo o caminho completo para o diretório de relatórios usado pelo PC-DMIS. Este diretório contém os gabaritos de rótulos e relatório usados pela janela Relatório.

RIGHT

n caracteres à direita da sequência: *RIGHT*(<Sequência>, <Inteiro>)

Essa função retorna uma sequência que consiste nos n caracteres à direita especificados pelo inteiro a partir da sequência.

SYSTEMDATE

Data do sistema: SYSTEMDATE(<Sequência de formato de data>)

Essa função retorna a sequência formatada em data com os detalhes da data atual preenchidos. Por exemplo, o comando SYSTEMDATE("MM'/'dd'/'yy") retorna a sequência "02/12/14" se a data atual for 12 de fevereiro de 2014.

Use os elementos da sequência a seguir para criar a sequência de data. Os elementos devem coincidir maiúsculas e minúsculas, como mostrado a seguir (MM em vez de mm). Os caracteres que não são datas (tais como espaços) que aparecem entre os elementos da sequência do formato de data são exibidos na sequência de saída no mesmo local da sequência de entrada. Os caracteres da

sequência de entrada delimitados por aspas simples são exibidos no mesmo local na sequência de saída sem as aspas simples.

d - Dia do mês como dígitos. Sem zero à esquerda para datas de um dígito.

dd – Dia do mês como dígitos. Com zero à esquerda para datas de um dígito.

ddd – Abreviatura com três letras para o dia da semana.

dddd – Nome completo do dia atual da semana.

M – Mês como dígitos, sem zero à esquerda para meses de um dígito.

MM – Mês como dígito, com zero à esquerda para meses de um dígito.

MMM – Mês como abreviação de três letras.

MMMM – Nome completo do mês.

y – Ano como dígitos, sem zeros à esquerda para anos de um dígito.

yy – Ano como dígitos, com zero à esquerda para anos de um dígito.

yyyy – Ano representado por quatro dígitos.

SYSTEMTIME

Hora do sistema formatada: *SYSTEMTIME(<Sequência do formato de hora>)*

Essa função retorna a sequência formatada em horário com os detalhes do horário atual preenchidos. Por exemplo, o comando `SYSTEMTIME("hh:mm:ss tt")` retorna o tempo em uma sequência formatada, como "11:29:40 PM".

Use os elementos da sequência a seguir para criar a sequência de hora. Os elementos devem coincidir maiúsculas e minúsculas, como mostrado a seguir (**tt** em vez de **TT**). Os caracteres que não são horas (tais como espaços) que aparecem entre os elementos da sequência do formato de hora serão exibidos na sequência de saída no mesmo local da sequência de entrada. Os caracteres da sequência de entrada delimitados por aspas simples serão exibidos no mesmo local na sequência de saída sem as aspas simples.

h - Horas sem zero à esquerda para horas de um dígito; relógio de 12 horas

hh - Horas com um zero à esquerda para horas de um dígito; relógio de 12 horas

H - Horas sem zero à esquerda para horas de um dígito; relógio de 24 horas

HH - Horas com um zero à esquerda para horas de um dígito; relógio de 24 horas

m - Minutos sem zero à esquerda para minutos de um dígito

mm - Minutos com zero à esquerda para minutos de um dígito

s - Segundos sem zero à esquerda para segundos de um dígito

ss - Segundos com zero à esquerda para segundos de um dígito

t - Sequência do indicador de hora de um caractere, tal como A ou P

tt - Sequência do indicador de hora de vários caracteres, tais como AM ou PM

SYSTIME

Hora do sistema: *SYSTIME()*

Essa função retorna uma sequência com a hora atual do sistema. Essa função é diferente da função *SYSTEMTIME* descrita acima. Ela retorna automaticamente o dia, a data e a hora, seguidos pelo ano.

Exemplo: "Qua Fevereiro 12 13:50:21 2014"



A sequência retornada, mostrando a hora atual do sistema, é ajustada para as definições do fuso horário local.

UPPERCASE

Cria uma sequência em letras maiúsculas: *UPPERCASE(<Sequência>)*

Essa função retorna uma sequência que é o equivalente em letras maiúsculas da sequência.

Funções matemáticas

ABS

Valor absoluto: *ABS(<Duplo>)*

Retorna o valor absoluto do valor de entrada.

EXP

Exponencial: *EXP(<Duplo>)*

Retorna o exponencial da expressão.

LOG

Log Base 10: *LOG(<Duplo>)*

Retorna o logaritmo na base 10 da expressão.

LN

Log natural: *LN(<Duplo>)*

Retorna o logaritmo natural da expressão.

ROUND

Arredondamento: *ROUND(<Duplo>)*

Retorna o valor da entrada, arredondado para o inteiro mais próximo.

SQRT

Raiz quadrada: *SQRT(<Duplo>)*

Retorna a raiz quadrada da entrada.

Funções de trigonometria

Por padrão, todas as funções de trigonometria recebem e retornam radianos. Se desejar valores em graus, use a função RAD2DEG descrita abaixo.

ACOS

Arco cosseno: *ACOS(<Duplo>)*

Retorna o arco cosseno da expressão. Por exemplo, *ACOS(5,0)* retorna o valor 0. Em geral, *ACOS(<expressão>)* retorna o arco cosseno do valor da expressão.

ASIN

Arco seno: *ASIN(<Duplo>)*

Retorna o arco seno do valor da entrada.

ATAN

Arco tangente: *ATAN(<Duplo>)*

Retorna o arco tangente do valor da entrada.

COS

Cosseno: *COS(<Duplo>)*

Retorna o cosseno do valor da entrada.

DEG2RAD

Graus para radianos: *DEG2RAD(<Duplo>)*

Retorna a entrada dividida por 360 e multiplicada por 2π . Isso realiza a conversão de graus para radianos.

RAD2DEG

Radianos para graus: *RAD2DEG(<Double>)*

Retorna a entrada multiplicada por 360 e dividida por 2π . Isso realiza a conversão de radianos para graus.

SIN

Seno: *SIN(<Duplo>)*

Retorna o seno do valor da entrada.

TAN

Tangente: *TAN(<Duplo>)*

Retorna a tangente do valor da entrada.



As funções em que a entrada está fora do intervalo (como ACOS, ASIN, LOG, LN, SQRT etc. que fariam o computador travar) retornam 0.

Funções de pontos

ANGLEBETWEEN

Ângulo entre: *ANGLEBETWEEN(<vetor>, <vetor>)*

Retorna o valor do ângulo entre os dois vetores em graus. Os dois parâmetros devem ser expressões que avaliam para um tipo de vetor. Para obter o vetor de um elemento, por exemplo, você precisa usar a ID do elemento seguida pela extensão .IJK. Você pode ver isso no resumo do código abaixo:



```
F1      =GENÉRICO/PONTO,DEPENDENTE,CARTESIANO,$
        NOM/XYZ,<3,3,3>,$
        MED/XYZ,<3,3,3>,$
        NOM/IJK,<1,0,0>,$
        MED/IJK,<1,0,0>

F2      =GENÉRICO/PONTO,DEPENDENTE,CARTESIANO,$
        NOM/XYZ,<10,10,10>,$
        MED/XYZ,<10,10,10>,$
        NOM/IJK,<0,0,1>,$
        MED/IJK,<0,0,1>
        ATRIBUIR/V1=F1.IJK
        ATRIBUIR/V2=F2.IJK
        ATRIBUIR/V3=ANGLEBETWEEN(V1,V2)
        COMENT/OPER,NÃO,TELA INTEIRA=NÃO,AUTO-
CONTINUAR=NÃO,
        "O ângulo entre "+V1+" e "+V2+" é: "+V3
```

CROSS

Produto vetorial: *CROSS(<Ponto>, <Ponto>)*

O valor de retorno é do tipo ponto e é o vetor da unidade na direção do produto vetorial da primeira e da segunda expressões.

DELTA

Deslocamento vetorial: *DELTA*(<Ponto>, <Ponto>, <Duplo>)

A função pega a primeira expressão (ponto) e calcula um novo ponto na direção da segunda expressão (vetor) em um deslocamento da terceira expressão. Por exemplo, *DELTA* (*MPOINT* (0, 0, 0) , *MPOINT* (1, 0, 0) , 10) retorna o ponto 10,0,0.

DOT

Produto escalar: *DOT*(<Ponto>, <Ponto>)

Retorna o produto escalar dos dois pontos (vetores).

UNIT

Vetor unitário: *UNIT*(<Ponto>)

Retorna o ponto dividido pelo seu comprimento. Por exemplo, *UNIT* (*MPOINT* (0, 0, 0)) retorna o ponto 0,0,1.

MPOINT

Coerção de ponto: *MPOINT*(<Expressão1>, <Expressão2>, <Expressão3>)

Força as três expressões para um tipo Ponto, como mostrado neste resumo de código:

```
ATRIBUIR/V1=MPOINT (2.5, 3.6, 4)
```

Em que:

$$V1.X = 2,5$$

$$V1.Y = 3,6$$

$$V1.Z = 4,0$$

Consulte "Coerção de ponto".

Funções de apontadores

DIST2D

2d Distância: DIST2D(<ELEM1>, <ELEM2>, <ELEM3>)



Os elementos devem estar dentro dos suportes curvos.

DIST2D: 2d Distância: DIST2D(<ELEM1>, <ELEM2>, <ELEM3>) Calcula a distância entre os dois primeiros argumentos no comando (ELEM1 e ELEM2), perpendicular ao terceiro argumento (ELEM3).

- Se o terceiro argumento for um plano, o PC-DMIS calcula a distância entre os dois primeiros argumentos perpendicularmente ao plano.
- Se o terceiro argumento for uma linha ou um cilindro, o PC-DMIS calcula a distância entre os dois primeiros argumentos perpendicularmente ao terceiro argumento no plano de trabalho ativo.

Por exemplo, se você tiver o plano XY como seu terceiro argumento, ele terá um vetor Z+ (0,0,1) e a distância relatada estará apenas no eixo Z.

Exemplo



```
ATRIBUIR/V3=DIST2D({CIR1},{CIR2},{PLN1})  
COMENT/OPER,NÃO,TELA INTEIRA=NÃO,AUTO  
CONTINUAR=NÃO,  
V3
```

DIST3D

3D Distância: DIST3D(<ELEM1>, <ELEM2>)
Calcula a distância 3D entre o Elem1 e o Elem2.

Os elementos devem estar dentro dos suportes curvos.

Exemplo

```
ATRIBUIR/V3=DIST3D({CIR1},{CIR2})
COMENT/OPER,NÃO,TELA INTEIRA=NÃO,AUTO
CONTINUAR=NÃO,
V3
```

GETCOMMAND

Obtém um ponteiro para o comando especificado pelos parâmetros:

```
GETCOMMAND(<Inteiro ou sequência>, <Sequência>, <Inteiro>)
```

Campo de Informações do Primeiro Comando de Parâmetro

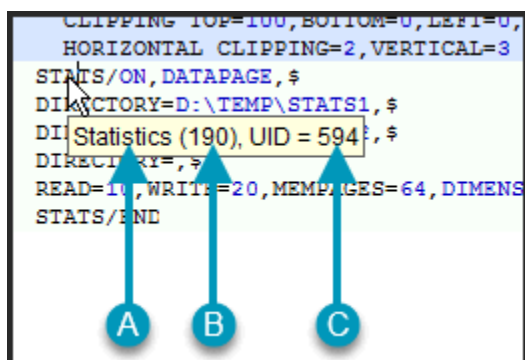
O primeiro parâmetro é um campo de informações sobre o comando. Ele especifica o tipo de comando a ser pesquisado. A seguir estão algumas das opções que podem ser transmitidas:

- Uma seqüência de descrição do comando. Consulte (A) no gráfico abaixo.
- Um número do tipo de comando. Consulte (B) no gráfico abaixo.
- Um identificador de número exclusivo. Consulte (C) no gráfico abaixo.

Se a ID exclusiva do comando for transmitida, nenhum outro argumento será necessário.

Para obter a seqüência de descrição do comando, o número do tipo de comando e o identificador de número exclusivo do comando:

1. Clique com o botão direito do mouse na Janela Edição
2. Escolha **Alterar exibição pop-up | Informações sobre o comando** (o PC-DMIS deve estar no modo Comando).
3. Posicione o mouse sobre o comando desejado. O PC-DMIS exibe a descrição do comando, o número do tipo e o identificador de número exclusivo daquele comando no pop-up.



- A. String de descrição de comando
- B. Número do tipo de comando
- C. Identificador numérico único (UID)

Segundo Parâmetro - Pesquisar Direção

O segundo parâmetro é a direção da pesquisa. Os valores válidos incluem:

Valor	Descrição
Acima	Esse valor significa que a pesquisa deve começar no comando atual e prosseguir para cima.
Abaixo	Esse valor significa que a pesquisa deve começar no comando atual e prosseguir para baixo.
TOP	Esse valor significa que a pesquisa deve começar no início da rotina de medição em uma direção descendente.
PARTE INFERIOR	Esse valor significa que a pesquisa deve começar com o último objeto da rotina de medição em uma direção ascendente.

Terceiro Parâmetro - Qual Instância Localizar

O terceiro parâmetro indica qual instância do comando deverá ser localizada se várias instâncias do mesmo comando estiverem na rotina de medição.



Se a rotina de medição tem duas ocorrências de comando ESTATS/LIG e você deseja obter um ponteiro para a segunda ocorrência a partir da parte superior, passe "2" como o terceiro parâmetro e "TOPO" como o segundo parâmetro, conforme mostrado aqui.

```
ATRIBUIR/V1=GETCOMMAND("Estatísticas","TOP",2)
```

Você pode usar a função GETCOMMAND para produzir o terceiro parâmetro para a função da sequência GETTEXT. Consulte "Funções de sequências" para obter informações sobre GETTEXT.

LEN

Contagem de loop do ponteiro: *LEN(<PONTEIRO>)*

Retorna o número de vezes que um ponteiro esteve em um loop. Por exemplo, se um elemento CIR1 está em um loop que faz iterações 10 vezes, você pode armazenar quantas vezes o CIR1 foi medido em uma variável utilizando uma declaração

ATRIBUIR como esta: *ATRIBUIR/V1=LEN({CIR1})*

Funções de matrizes

MATRIZ

Criar matriz: *MATRIZ(<EXPRESSÃO1>, <EXPRESSÃO2>, <EXPRESSÃO3>, ...)*

Cria um objeto de matriz com os elementos de matriz indicados pelos parâmetros da expressão. Os elementos de matriz são numerados com um índice de base 1.

MÉDIA

Média de elementos de matriz: *AVERAGE(<MATRIZ>)*

Retorna o valor médio dos elementos da matriz.

EQUAL

Comparação de matriz elemento por elemento: *EQUAL(<MATRIZ>, <MATRIZ>)*

Compara as duas matrizes, elemento por elemento, para determinar se elas contêm os mesmos elementos. Se as duas matrizes não forem do mesmo tamanho, ou se algum dos elementos de uma das matrizes não coincidir com o elemento correspondente da outra matriz, a função retorna o valor 0. Caso contrário, a função retorna 1.

LEN

Contagem de elementos da matriz: *LEN(<MATRIZ>)*

Retorna o número de elementos da matriz.

MAX

Maior elemento da matriz: *MAX(<MATRIZ>)*

Retorna o maior elemento da matriz. Os itens da matriz são comparados numericamente ou alfabeticamente.

MIN

Menor elemento da matriz: *MIN(<MATRIZ>)*

Retorna o menor elemento da matriz. Os itens da matriz são comparados numericamente ou alfabeticamente.

SOMA

Soma de elementos de matriz: *SOMA(<MATRIZ>)*

Retorna a soma dos elementos na matriz.

Funções diversas

ARCSEGMENTENDINDEX

Essa função retorna o número do índice do ponto final de um segmento de arco especificado a partir de uma varredura: *ARCSEGMENTENDINDEX(<ID>, <índice>, <tol1>, <tol2>)*

<ID> – O primeiro parâmetro é um valor de sequência da ID da varredura da qual essa função obtém o número de índice do ponto final do arco. Esse parâmetro pode ser a ID entre aspas ou qualquer expressão que, quando forçada para o tipo de sequência, acaba sendo a ID de uma varredura.

<índice> - O segundo parâmetro é o número do índice para o arco a partir do qual você deseja obter o número do ponto final. Esse é um valor baseado em um. Por exemplo, o número do índice do arco seria 3 se você desejar o número do ponto final do terceiro arco na varredura.

<tol1> – O terceiro parâmetro é a tolerância geral do elemento. É um erro máximo de forma utilizado para dividir a varredura em linhas e arcos.

<tol2> – O quarto parâmetro é a tolerância aprimorada. Geralmente, essa tolerância mais forte é utilizada para projetar pontos a partir do final do elemento até que o erro da forma do segmento esteja dentro dessa tolerância.

Depois de ter os índices Inicial e Final de um arco, é possível usar esses pontos dentro de um elemento construído para construir um elemento de arco separado. Consulte "Exemplo de um elemento Linha criado a partir de um segmento de varredura" para ver um exemplo similar.

ARCSEGMENTSTARTINDEX

Essa função retorna o número do índice do ponto de início de um segmento de arco especificado a partir de uma varredura: `ARCSEGMENTSTARTINDEX(<ID>, <índice>, <tol1>, <tol2>)`.

`<ID>` – O primeiro parâmetro é um valor de sequência da ID da varredura da qual essa função obtém o número de índice do ponto inicial do arco. Esse parâmetro pode ser a ID entre aspas ou qualquer expressão que, quando forçada para o tipo de sequência, acaba sendo a ID de uma varredura.

`<índice>` - O segundo parâmetro é o número do índice para o arco a partir do qual você deseja obter o número do ponto inicial. Esse é um valor baseado em um. Por exemplo, o número do índice do arco seria 3 se você desejar o número do ponto inicial do terceiro arco na varredura.

`<tol1>` – O terceiro parâmetro é a tolerância geral do elemento. É um erro máximo de forma utilizado para dividir a varredura em linhas e arcos.

`<tol2>` – O quarto parâmetro é a tolerância aprimorada. Geralmente, essa tolerância mais forte é utilizada para projetar pontos a partir do final do elemento até que o erro da forma do segmento esteja dentro dessa tolerância.

Há dois parâmetros adicionais, os quais controlam se um segmento de arco identificado em uma varredura é aceitável. Eles podem ser modificados apenas com o Editor de configurações PC-DMIS. Qualquer segmento de arco com raio menor que o valor para a entrada `MinimumArcSegmentRadiusInMM` é rejeitado. O valor padrão desse parâmetro é 2 mm. De forma similar, qualquer segmento de arco com o raio maior que o valor da entrada `MaximumArcSegmentRadiusInMM` é rejeitado. O valor padrão para este parâmetro é de 2000 mm (provavelmente não será necessário alterar este valor).

Depois de ter os índices Inicial e Final de um arco, é possível usar esses pontos dentro de um elemento construído para construir um elemento de arco separado. Consulte "Exemplo de um elemento Linha criado a partir de um segmento de varredura" para ver um exemplo similar.

EOF e EOL

Para mais informações sobre essas funções, consulte "Verificação do final de um arquivo ou do final de uma linha" no capítulo "Uso de entrada/saída de arquivo".

FUNCTION

Cria uma função: `FUNCTION((<PARAM1>, <PARAM2>...), <EXPRESSÃO>)`

Cria uma função que usa o número de parâmetros indicado pela lista de parâmetros e substitui esses parâmetros na expressão.

- O primeiro item ao usar a palavra-chave `FUNCTION` é a lista de parâmetros.
- A lista consiste em nomes de parâmetros separados por vírgulas.
- A lista de parâmetros também é cercada por parênteses.
- O segundo item é a expressão.
- A expressão contém os nomes dos parâmetros onde os parâmetros devem ser substituídos quando a função é chamada.

Consulte o tópico "Exemplo de função genérica" para ver um exemplo.

GETROTABDATA

Essa função retorna os valores de centro, a posição angular e o vetor para a mesa rotatória especificada.

`GETROTABDATA(<PARÂMETRO>[,<MESA>])`

Essa função retorna os valores para as seguintes configurações:

- Mesa rotatória simples
- Mesas rotatórias duplas (independentes)
- Mesas Rotatórias empilhadas

Os dados retornados pela função correspondem aos dados na caixa de diálogo **Configuração da mesa rotatória (Editar | Preferências | Configurar mesa rotatória)**. Para mais informações sobre essa caixa de diálogo, consulte "Definição da mesa rotatória".

CENTRO

- "CENTRO" - Retorna o centro XYZ atual da mesa rotatória.
- "CENTRO","V" - Retorna o centro XYZ atual para a mesa rotatória V em configuração de mesas duplas ou mesas empilhadas.
- "CENTRO","W" - Retorna o centro XYZ atual para a mesa rotatória W em configuração de mesas duplas ou mesas empilhadas.

Exemplos:

<code>ATRIBUIR/V1=GETROTABDATA ("CENTRO")</code>	V1 é definido para o centro XYZ atual da mesa rotatória.
<code>ATRIBUIR/V1=OBTERDADOSMESAROT ("CENTRO", "V")</code>	V1 é definido para o centro XYZ atual da mesa rotatória V.
<code>ATRIBUIR/V1=OBTERDADOSMESAROT ("CENTRO", "W")</code>	V1 é definido para o centro XYZ atual da mesa rotatória W.

POSIÇÃO ANGULAR

- "ÂNGULO" - Retorna a posição angular atual da mesa rotatória.
- "ÂNGULO", "V" - Retorna a posição angular atual para a mesa rotatória V em configuração de mesas duplas ou mesas empilhadas.
- "ÂNGULO", "W" - Retorna a posição angular atual para a mesa rotatória W em configuração de mesas duplas ou mesas empilhadas.

Exemplos:

<code>ATRIBUIR/V2=GETROTABDATA ("ÂNGULO")</code>	V2 é definido para a posição angular atual da mesa rotatória.
<code>ATRIBUIR/V2=GETROTABDATA ("ÂNGULO", "V")</code>	V2 é definido para a posição angular atual da mesa rotatória V.
<code>ATRIBUIR/V2=GETROTABDATA ("ÂNGULO", "W")</code>	V2 é definido para a posição angular atual da mesa rotatória W.

VETOR

- "VECTOR" - Retorna o vetor IJK atual da mesa rotatória.
- "VECTOR", "V" - Retorna o vetor IJK atual para a mesa rotatória V em configuração de mesas duplas ou mesas empilhadas.

Uso de expressões e variáveis

- "VECTOR","W" - Retorna o vetor IJK atual para a mesa rotatória W em configuração de mesas duplas ou mesas empilhadas.

Exemplos:

<code>ATRIBUIR/V3=GETROTABDATA ("VECTOR")</code>	V3 é definido para o vetor IJK atual da mesa rotatória.
<code>ATRIBUIR/V3=GETROTABDATA ("VECTOR", "V")</code>	V3 é definido para o vetor IJK atual da mesa rotatória V.
<code>ATRIBUIR/V3=GETROTABDATA ("VECTOR", "W")</code>	V3 é definido para o vetor IJK atual da mesa rotatória W.



O argumento [MESA] é opcional. Se você não especifica a mesa V ou W, o PC-DMIS toma uma das seguintes ações:

- Se você está usando uma configuração de mesa única ou mesas empilhadas, ele retorna os valores para a mesa rotatória W.
- Se você está usando uma configuração de mesas duplas, ele retorna os valores para a mesa rotatória que está ativada na barra de ferramentas **Mesa rotatória ativa**. Para mais informações sobre a barra de ferramentas, consulte "Barra de ferramentas Mesa rotatória ativa".

O PC-DMIS possui duas definições de mesa interna para acomodar as configurações de mesas duplas e mesas empilhadas. Para uma configuração de mesa única, a segunda definição de mesa não é realmente usada. Mas porque a definição existe internamente, não ocorrerá um erro se você especificar a mesa V em uma configuração de mesa única; contudo, isso não é recomendado. Os valores retornados pela função geralmente não são úteis, pois a mesa não existe realmente.

IF

Cálculo da expressão condicional: IF(<EXPRESSÃO1>, <EXPRESSÃO2>, <EXPRESSÃO3>)

Se a expressão1 for verdadeira (não-zero), essa função retorna o valor de expressão2; caso contrário, ela retorna o valor de expressão3.

ISIOCHANNELSET

Essa expressão usa dois parâmetros. O primeiro parâmetro indica qual canal E/S será marcado (o intervalo de números disponíveis está baseado na máquina que está sendo usada). O segundo parâmetro determina se o software consulta a máquina Braço1 ou Braço2. Se o segundo parâmetro é configurado para 1 (um), ele consulta o controlador do Braço2. Se o segundo parâmetro não estiver presente (ou configurado para zero), o Canal IO consulta o controlador do Braço1. O controlador do Braço1 é a única opção se você não estiver no modo de vários braços.



Se for fornecido um tipo de dados de sonda, uma ID de ponta, um nome de arquivo de sonda ou um número de canal inválido, a expressão é avaliada para 0.

Exemplo:

```
ATRIBUIR/V4=ISIOCHANNELSET (3, 0)
```

V4 será igual a 1 (avaliado como verdadeiro) quando o canal for definido, caso contrário, ele será igual a 0 (avaliado como falso).

LINESEGMENTENDINDEX

Essa função retorna o número do índice do ponto final de um segmento de linha especificado a partir de uma varredura: LINESEGMENTENDINDEX(<ID>, <index>, <tol1>, <tol2>).

<ID> – O primeiro parâmetro é um valor de sequência da ID da varredura da qual essa função obtém o número de índice do ponto final do segmento de linha. Esse parâmetro pode ser a ID entre aspas ou qualquer expressão que, quando forçada para o tipo de sequência, acaba sendo a ID de uma varredura.

<índice> – O segundo parâmetro é o número do índice para o segmento de linha a partir do qual você deseja obter o número do ponto final. Esse é um valor baseado em um. Por exemplo, o número do índice do segmento de linha seria 3 se você deseja o número do ponto final da terceira linha na varredura.

<tol1> – O terceiro parâmetro é a tolerância geral do elemento. É um erro máximo de forma utilizado para dividir a varredura em linhas e arcos.

<tol2> – O quarto parâmetro é a tolerância aprimorada. Geralmente, essa tolerância mais forte é utilizada para projetar pontos a partir do final do elemento até que o erro da forma do segmento esteja dentro dessa tolerância.

Depois de ter os índices Inicial e Final de um segmento de linha, é possível usar esses pontos dentro de um elemento construído para construir um elemento de linha

separado. Consulte "Exemplo de um elemento Linha criado a partir de um segmento de varredura" para ver um exemplo.

LINESEGMENTSTARTINDEX

Retorna o número do índice do ponto de início de um segmento de linha especificado a partir de uma varredura: `LINESEGMENTSTARTINDEX(<ID>, <índice>, <tol1>, <tol2>)`.

<ID> – O primeiro parâmetro é um valor de sequência da ID da varredura da qual essa função obtém o número de índice do ponto inicial do segmento de linha. Essa pode ser a ID entre aspas ou qualquer expressão que, quando forçada para o tipo sequência, acaba por se tornar a ID de uma varredura.

<índice> – O segundo parâmetro é o número do índice para o segmento de linha a partir do qual você deseja obter o número do ponto inicial. Esse é um valor baseado em um. Por exemplo, o número do índice do segmento de linha seria 3 se você deseja o número do ponto inicial da terceira linha na varredura.

<tol1> – O terceiro parâmetro é a tolerância geral do elemento. É um erro máximo de forma utilizado para dividir a varredura em linhas e arcos.

<tol2> – O quarto parâmetro é a tolerância aprimorada. Geralmente, essa tolerância mais forte é utilizada para projetar pontos a partir do final do elemento até que o erro da forma do segmento esteja dentro dessa tolerância.

Há um parâmetro adicional, que controla se um segmento de linha identificado em uma varredura é aceitável. Ele pode ser modificado apenas com o Editor de configurações do PC-DMIS. Qualquer segmento de linha menor do que o valor da entrada `MinimumLineSegmentLengthInMM` é rejeitado. O valor padrão desse parâmetro é 2 mm.

Depois de ter os índices Inicial e Final de um segmento de linha, é possível usar esses pontos dentro de um elemento construído para construir um elemento de linha separado. Consulte "Exemplo de um elemento Linha criado a partir de um segmento de varredura" para obter um exemplo.

PROBEDATA

Essa função retorna dados sobre a sonda atual ou especificada:

`PROBEDATA(<TIPODADOSSONDAOPC>, <IDPONTAOPC>, <NOMEARQUIVOOPC>)`

Essa função usa até três parâmetros opcionais. Você precisa colocar vírgulas entre os parâmetros apenas se usar mais de um parâmetro. Não é necessário usar vírgulas entre parâmetros vazios. Por exemplo, para obter o diâmetro da sonda atual, você usaria `ATRIBUIR/V1=PROBEDATA("DIAM")`.

OPTPROBEDATATYPE - Parâmetro opcional que especifica quais dados da sonda a expressão deve retornar. Se esse parâmetro não for fornecido, a ID da ponta é retornada. Esse parâmetro é do tipo sequência. Qualquer expressão que calcule uma expressão de sequência válida pode ser colocada no local da primeira expressão. As expressões de sequência válidas (não sensíveis a maiúsculas e minúsculas) do

primeiro parâmetro incluem o seguinte. São expressões de sequência e devem estar entre aspas duplas:

"A" - Ângulo A da ponta. Retorna tipo duplo.

"B" - Ângulo B da ponta. Retorna tipo duplo.

"C" - O ângulo C de uma cabeça de CW43 (sonda leve). Retorna tipo inteiro.

"Date" - Data em que a ponta foi qualificada pela última vez. Retorna tipo sequência.

"Diam" ou **"Diameter"** - Diâmetro da ponta medida. São necessárias as quatro primeiras letras "Diam", mas pode-se incluir mais letras até o nome completo de "Diameter". Retorna tipo duplo.

"ID" - ID da ponta. Parâmetro padrão. Retorna tipo sequência.

"Offset" - Deslocamento X,Y,Z da ponta medida. Retorna tipo ponto.

"PrbRdv" - O desvio radial da sonda. Retorna tipo duplo.

"Rotation" - Essa é a rotação sobre o vetor da ponta, em radianos. Retorna tipo duplo.

"Standarddeviation" - O desvio padrão da sonda. Retorna tipo duplo.

"Thick" ou **"Thickness"** - Espessura medida da ponta. As primeiras cinco letras são requeridas "Thick", mas poderia incluir mais letras até o nome completo de "Thickness". Retorna tipo duplo.

"Time" - Hora em que a ponta foi qualificada pela última vez. Retorna tipo sequência.

"Vector" - Vetor da ponta. Retorna tipo ponto.



Adicionar um **"T"** na frente de **"Diâmetro"**, **"Deslocamento"** ou **"Espessura"** retorna as informações teóricas (por exemplo, **TDIÂMETRO**, **TDESLOCAM** e **TESPESSURA**).

OPTTIPID - Esse parâmetro opcional especifica a ponta a ser usada ao obter os dados da sonda especificados na primeira expressão. Se não forem fornecidos, a ponta atual será usada. Esse parâmetro deve ser tipo sequência.

OPTPROBEFILENAME - Esse parâmetro opcional especifica o nome do arquivo da sonda a ser usado na obtenção dos dados da sonda. Se usso não for fornecido, o arquivo da sonda atual é usado.

Exemplos:

```
ATRIBUIR/V1=PROBEDATA()
```

V1 é definido para a ID de ponta atual (isto é, "T1A0B0")

<code>ATRIBUIR/V2=PROBEDATA("TDESLOCAMENTO","T1A45B0")</code>	V2 é definido para o deslocamento de sonda teórica para a ponta T1A45B0
<code>ATRIBUIR/V3=PROBEDATA("Data","T1A90B90","MEUSONDA")</code>	V3 é definido para uma seqüência que representa a data em que a ponta T1A90B90 do arquivo de sonda MINSENSOR foi qualificado pela última vez.

QUALTOOLDATA

Essa função retorna dados sobre a ferramenta de calibração atual ou especificada. Ela tem esta sintaxe:

`QUALTOOLDATA(<INFOFERRAM>, <IDFERRAM>, <NÚMEROFACE>)`

Essa função usa até três parâmetros. Ela precisa pelo menos de um parâmetro para retornar algum dado:

O primeiro parâmetro, <TOOLINFO>, é uma *seqüência* que especifica o tipo de informação que retornará sobre da ferramenta de calibração. Se não passar este parâmetro, esta função retorna o nome da ferramenta de calibração atual ou especificada.

- **"CTE"** ou **"COEFFICIENTOFHERMALEXPANSION"** - Qualquer uma dessas seqüências retorna o coeficiente de expansão térmica como um valor duplo.
- **"DIAM"** - Essa seqüência retorna o diâmetro da ferramenta como valor duplo.
- **"ID"** - Essa seqüência retorna o nome da ferramenta como um valor de seqüência.
- **"LENGTH"** - Essa seqüência atua da mesma forma que o "DIAM". Também retorna o diâmetro da ferramenta como valor duplo.

- **"OVERRIDEIJK"** - Essa sequência retorna o vetor pesquisar substituir IJK como um valor ponto.
- **"POLYDIAM"** - Essa sequência retorna o diâmetro da face do poliedrico especificado como um valor duplo.
- **"POLYIJK"** - Essa sequência retorna o vetor IJK da face do poliedrico especificado como um valor de ponto.
- **"POLYXYZ"** - Essa sequência retorna o centro XYZ da face do poliedrico especificado como um valor de ponto.
- **"SHANKIJK"** - Essa sequência retorna o vetor IJK vector da haste como um valor de ponto.
- **"TIPO"** – Essa sequência retorna o tipo de ferramenta como um valor inteiro (0 para esfera, 1 para esfera Braço2, 2 para poliedro, 3 para poliedro Braço2).
- **"WIDTH"** - Esse parâmetro não é mais usado.
- **"XYZ"** - Essa sequência retorna o local XYZ da ferramenta como um valor de ponto.

O segundo parâmetro, *<IDFERRAM>*, é uma *sequência* que especifica o nome da ferramenta de calibração sobre a qual o usuário gostaria de receber informações. Se você não passar esse parâmetro, o PC-DMIS assume que deseja informações sobre a ferramenta de calibração atual. A sequência não é sensível a maiúsculas e minúsculas.

O terceiro parâmetro, *<NÚMEROFACE>*, somente é necessário quanto você trabalha com uma ferramenta de calibração poliédrica e somente quando o primeiro parâmetro for "POLYXYZ", "POLYIJK", or "POLYDIAM. Este é um valor inteiro que especifica a face da ferramenta poliédrica para utilizar com a finalidade de obter os dados.



A ferramenta de calibração não possui uma configuração global que se aplica automaticamente a todas as sondas em uma rotina de medição. Ao calibrar uma sonda pela primeira vez, você precisa selecionar a ferramenta de calibração a ser usada. O PC-DMIS salva as informações da ferramenta de calibração para cada sonda. Quando você recalibrar a mesma sonda, o PC-DMIS irá usar a mesma ferramenta de calibração.

Exemplos:

<pre>ATRIBUIR/VDIAM=QUALTOOLDATA("DIAM","ESFERA_1_IN")</pre>	<p>Fornece à variável VDIAM o diâmetro da ferramenta SPHERE_1_IN.</p>
--	---

<code>ATRIBUIR/VID=QUALTOOLDATA ("ID")</code>	Fornece à variável VID o nome da ferramenta atual.
<code>ATRIBUIR/VTYPE=QUALTOOLDATA ("TIPO")</code>	Fornece à variável VTYPE o tipo da ferramenta atual.
<code>ATRIBUIR/VPOLYDIAM=QUALTOOLDATA ("DIAMPOLI", "TESTEPOLI", 3)</code>	Fornece à variável VPOLYDIAM o diâmetro da face 3 na ferramenta de poliedro, POLYTEST.

SETROTABDATA

Essa função define o centro ou vetor para o novo valor de entrada:

`SETROTABDATA(<PARÂMETRO>,<NovoValor>[,<MESA>])`

. Essa função funciona nas seguintes configurações:

- Mesa rotatória simples
- Mesas rotatórias duplas (independentes)
- Mesas Rotatórias empilhadas

CENTRO

- "CENTRO",<NovoValor> - Define o centro XYZ da mesa rotatória para o novo valor.
- "CENTRO",<NovoValor>,"V" - Define o centro XYZ para a mesa rotatória V em configuração de mesas duplas ou mesas empilhadas para o novo valor.
- "CENTRO",<NovoValor>,"W" - Define o centro XYZ para a mesa rotatória W em configuração de mesas duplas ou mesas empilhadas para o novo valor.

Exemplos:

<code>ATRIBUIR/V1=SETROTABDATA("CENTRO",NovoValor)</code>	V1 é um código de retorno (1=sucesso, 0=falha).
<code>ATRIBUIR/V1=SETROTABDATA("CENTRO",NovoValor,"V")</code>	V1 é um código de retorno (1=sucesso, 0=falha).
<code>ATRIBUIR/V1=SETROTABDATA("CENTRO",NovoValor,"W")</code>	V1 é um código de retorno (1=sucesso, 0=falha).

VETOR

- "VETOR",<NovoValor> - Define o vetor IJK da mesa rotatória para o novo valor.
- "VETOR",<NovoValor>,"V" - Define o IJKZ para a mesa rotatória V em configuração de mesas duplas ou mesas empilhadas para o novo valor.
- "VETOR",<NovoValor>,"W" - Define o IJKZ para a mesa rotatória W em configuração de mesas duplas ou mesas empilhadas para o novo valor.

Exemplos:

<code>ATRIBUIR/V2=SETROTABDATA("VETOR",NovoValor)</code>	V2 é um código de retorno (1=sucesso, 0=falha).
<code>ATRIBUIR/V2=SETROTABDATA("VETOR",NovoValor,"V")</code>	V2 é um código de retorno (1=sucesso, 0=falha).
<code>ATRIBUIR/V2=SETROTABDATA("VETOR",NovoValor,"W")</code>	V2 é um código de retorno (1=sucesso, 0=falha).



O argumento [MESA] é opcional. Se você não especifica a mesa V ou W, o PC-DMIS toma uma das seguintes ações:

- Se você está usando uma configuração de mesa única ou mesas empilhadas, é definido um novo valor para a mesa rotatória W.
- Se você está usando uma configuração de mesas duplas, são definidos novos valores para a mesa rotatória que está ativada na barra de ferramentas **Mesa rotatória ativa**. Para mais informações sobre a barra de ferramentas, consulte "Barra de ferramentas Mesa rotatória ativa".

O PC-DMIS possui duas definições de mesa interna para acomodar as configurações de mesas duplas e mesas empilhadas. Para uma configuração de mesa única, a segunda definição de mesa não é realmente usada. Mas porque a definição existe internamente, não ocorrerá um erro se você especificar a mesa V em uma configuração de mesa única; contudo, isso não é recomendado. Os valores definidos pela função geralmente não são úteis, pois a mesa não existe realmente.

TUTORELEMENT

Essa função utiliza um parâmetro, seja um número ou uma sequência (uma sequência seria a ID de um elemento).

TUTORELEMENT(<PARÂMETRO>)

Essa função trabalha com o tipo de variável, *Estruturas*. Consulte "Estruturas" para ver explicações da estrutura e dos subelementos.

Exemplos:

ATRIBUIR/E=TUTORELEMENT (1)	Cria uma única Estrutura de elemento do tutor.
ATRIBUIR/WM=TUTORELEMENT (n)	Para qualquer número acima de 1, isso cria uma matriz de n estruturas de elemento do tutor.
ATRIBUIR/CIR1E=TUTORELEMENT ("CIR1")	Copia dados do elemento CIR1 para as Estruturas de elemento do tutor.

Atualmente, a estrutura TutorElement possui os seguintes subelementos:

Subelemento	Descrição
ID	Seqüência da ID do elemento.
TIPO	INTEIRO (FTIPO)
X, Y, Z	Valores das coordenadas X, Y e Z.
RP	Raio polar
AP	Ângulo polar
CX	I
CY	J
CZ	K
DM	Diâmetro 1
DM2	Diâmetro 2
DS	Distância da origem
A	Ângulo
AXY	Ângulo no plano XY
AYZ	Ângulo no plano YZ
AZX	Ângulo no plano ZX
F	Erro na forma
SDEV	Desvio padrão
PR	Posição

Exemplos de funções

A seguir estão alguns exemplos diferentes de funções que podem ajudá-lo a criar e usar suas próprias funções:

- Exemplo de função genérica

Uso de expressões e variáveis

- Exemplo de funções passadas como variáveis
- Exemplo de função com vários parâmetros
- Exemplo de funções criando outras funções
- Exemplo de funções como itens de uma matriz
- Exemplo de funções definidas recursivamente

Exemplo de função genérica

`ATRIBUIR/MINHAFUNÇÃO=FUNÇÃO ((X,Y,Z) ,X*3+Y*2+Z)`

- Cria uma função definida pelo usuário e a atribui à variável MINHAFUNÇÃO. A função usa três parâmetros, X, Y e Z.

- X é multiplicado por 3.
- Y é multiplicado por 2.
- Z contém o valor passado.

O total de $X + Y + Z$ é o que é retornado quando valores são passados para a função, como mostrado aqui:

`ATRIBUIR/V1=MINHAFUNÇÃO (7,2,5)`

- Atribui a V1 o valor 30, por meio da avaliação dos parâmetros passados para a função MINHAFUNÇÃO(7,2,5).

- 7 é o parâmetro e é substituído onde há um X na parte da definição da função da expressão. Assim, $X*3$ torna-se $7*3$, ou 21.
- 2 é substituído onde Y ocorre. Assim, $Y*2$ torna-se $2*2$, ou 4.
- 5 é substituído onde Z ocorre.

Em seguida, os valores são todos somados ($21 + 4 + 5$) e passados para V1.

Exemplo de funções passadas como variáveis

As funções podem ser passadas como variáveis. O exemplo a seguir é construído no Exemplo de função genérica acima:

`ASSIGN/NEWFUNC=MYFUNC`

Define as variáveis NEWFUNC para que tenham a mesma função que MYFUNC tem.

```
ASSIGN/V3=NEWFUNC (12, 2, 3)
```

Atribui V3 para ter o valor 43 a partir das expressões avaliadas dentro da função (36 + 4 + 3).

Exemplo de função com vários parâmetros

As funções podem ter vários parâmetros:

```
ATRIBUIR/ADDANDDOUBLE=FUNÇÃO ( (A, B) , 2 * (A+B) )
```

Cria uma função e a atribui à variável ADDANDDOUBLE. A função usa dois parâmetros, soma-os e então multiplica o resultado por 2.

```
ASSIGN/V2=ADDANDDOUBLE (4, 5)
```

Atribui a V2 o valor de 18. Os parâmetros 4 e 5 são substituídos na parte da função da expressão, ficando portanto 2*(4+5).

Exemplo de funções criando outras funções

As funções podem criar outras funções.

```
ASSIGN/COMPOSE=FUNCTION ( (F, G) , FUNCTION ( (X) , G (F (X) ) ) )
```

Atribui COMPOSE para ser uma função que usa duas funções como parâmetros e cria uma nova função usando as duas funções.

```
ASSIGN/ADD2=FUNCTION ( (X) , X+2)
```

Atribui ADD2 para ser uma função que adiciona dois ao parâmetro passado.

```
ASSIGN/ADD3=FUNCTION ( (X) , X+3)
```

Atribui ADD3 para ser uma função que adiciona três ao parâmetro passado.

```
ASSIGN/ADD5=COMPOSE (ADD2, ADD3)
```

Atribui ADD5 para ser uma função formada das funções ADD2 e ADD3.

```
ASSIGN/V5=ADD5 (3)
```

Atribui a V5 o valor V8.

Exemplo de funções como itens de uma matriz

As funções podem ser membros de uma matriz.

```
ASSIGN/ANARRAY=ARRAY (3, FACTORIAL, "Olá Pessoal", ADD5)
```

Atribui ANARRAY para ser uma matriz de 4 elementos: um número (3), uma função (FACTORIAL), uma sequência, ("Olá Pessoal") e uma função (Add5).

Uso de expressões e variáveis

```
ASSIGN/V6=ANARRAY[2](4)
```

O segundo elemento de ANARRAY é a função FACTORIAL. O parâmetro 4 é passado para essa função e o resultado de 24 é atribuído à V6.

```
ASSIGN/V7=ANARRAY[2](ANARRAY[4](ANARRAY[1]))
```

De dentro para fora: O primeiro elemento de ANARRAY (3) é passado para a função do elemento (Add5) da quarta matriz. O resultado, 8, é passado para a função do segundo elemento da matriz (FATORIAL) e atribuído à V7. V7 recebe o valor 40320.

Exemplo de funções definidas recursivamente

As funções podem ser definidas recursivamente, ou seja, podem ser definidas para chamar a si próprias.

```
ASSIGN/FACTORIAL=FUNCTION((X), IF(X<=1, 1, X*FACTORIAL(X-1)))
```

Cria uma função chamada de fatorial que admite um parâmetro. Se o parâmetro for menor ou igual a 1, é avaliada para 1; caso contrário, é avaliada para X multiplicado pelo FACTORIAL de X-1.

```
ASSIGN/V4=FACTORIAL(5)
```

Atribui a V4 o valor de 120 (5*4*3*2*1).

Exemplo de um elemento Linha criado a partir de um segmento de varredura

Esse tópico fornece um exemplo de como usar a linguagem de expressão do PC-DMIS, especialmente as funções de segmento de linha, para exportar números de pontos inicial e final para segmentos de linha dentro de uma varredura e, em seguida, criar seu próprio elemento de linha utilizando os pontos extraídos dentro de um elemento construído. Também é possível usar os mesmos princípios abrangidos neste exemplo para criar um segmento de arco a partir de uma varredura.

Suponha que a rotina de medição tenha um elemento de varredura denominado SCN1 que seja semelhante ao seguinte:



```
SCN1=ELEM/VARRED, LINEARABERTO, MOSTRAR  
TOQUES=NÃO, MOSTRARTODOSPARÂM=SIM  
MODOS EXEC=REAPRENDER, MODOS NOMS=LOC  
NOMS, PLANASEG=NÃO, PONTO ÚNICO=NÃO, ESPESSURA=0  
LOCNOMS=5, SOMENTESELECIONADO=NÃO, USARMELHORA  
JUSTE=NÃO, COMPSONDA=SIM, MOVIMENTO  
EVITAR=NÃO, DISTÂNCIA=0, Compensação CAD=NÃO  
DIR1=VARIÁVEL,
```

```

TIPOTOQUE=VETOR
INITVEC=0,-1,0
DIRVEC=1,0,0
VETCORTE=0,0,1
VETFINAL=0,-1,0
PLANEVEC=-1,0,0
POINT1=100,0,-5
POINT2=70,0,-5
MED/VARRED
VARREDBÁSICA/LINHA,MOSTRAR
TOQUES=NÃO,MOSTRARTODOSPÂM=SIM
<100,0,-5>,<70,0,-5>,VetCorte=0,0,1,VetDir=1,0,0
VetInic=0,-1,0,VetFinal=0,-1,0,ESPESSURA=0
FILTRO/FILTRONULO,
MODO EXEC=REAPRENDER
FRONTEIRA/PLANO,<70,0,-5>,VetPlano=-
1,0,0,Cruzamentos=2
TIPOTOQUE/VETOR
MODO NOMS=LOCNOMS,5
FIMVARRED
ENDMEAS/

```

Para criar uma linha a partir dessa varredura, será necessário usar as funções LINESEGMENTSTARTINDEX e LINESEGMENTENDINDEX para obter os dados, da seguinte maneira:



```

ASSIGN/LINESTARTINDEX=LINESEGMENTSTARTINDEX("SCN1",1,
0.4,0.1)
ASSIGN/LINEENDINDEX=LINESEGMENTENDINDEX("SCN1",1,0.4,
0.1)

```

Isso instrui o PC-DMIS a ir para a varredura denominada "SCN1" e a partir de seu primeiro segmento de linha, obter os valores de índice inicial e final que estão dentro das tolerâncias definidas. Em seguida, atribui esses valores de índice a variáveis denominadas LINESTARTINDEX e LINEENDINDEX.

Depois de ter atribuído a variáveis os valores de índice inicial e final para o segmento de linha, será possível usar essas variáveis em uma linha construída, desta maneira:



```

LIN4=ELEM/LINHA,CARTES,NAODELIM
TEÓR/100.225,0,-5.011,1,0,0
REAL/100.225,-0.005,-5.011,1,-0.0000388,0
CONSTR/LINHA,BF,2D,SCN1.HIT[LINESTARTINDEX..LINEENDIN
DEX],,

```

```
REMOÇÃO_TESTEMUNHO/DESL, 3  
FILTRO/DESL, COMPRIMENTOONDA=0
```

Observe que no código realçado do elemento de linha acima, o PC-DMIS usa os números inicial e final obtidos da varredura para criar o elemento:

```
SCN1.HIT[LINESTARTINDEX..LINEENDINDEX]
```

Coerção de operandos

Os operandos podem ser forçados a outros tipos usando um dos operadores de coerção:

Coerção de inteiros

INT(<Expressão>) - Força o valor da expressão para tipo inteiro.

INT(4)	É avaliado para 4
INT(4,5)	É avaliado para 4
INT("Olá pessoal")	É avaliado para 0
INT("2")	É avaliado para 2
INT("2,2")	É avaliado para 2
INT("3 Blind Mice")	É avaliado para 3
INT("The 3 Blind Mice")	É avaliado para 0
INT("3, 4, 5")	É avaliado para 3
INT(MPOINT(0, 0, 1))	Calcula para a distância do ponto a partir da origem, nesse caso 1
INT(MPOINT(3, 4, 5))	Distância resulta em 7.0711. Essa expressão resulta em 7.

Coerção de duplos

DOUBLE(<Expressão>) - Força o valor da expressão para tipo duplo

DOUBLE (4)	Calcula para 4,0
DOUBLE (4, 5)	Calcula para 4,5
DOUBLE ("Uma sequência")	Calcula para 0,0
DOUBLE ("3, 5")	Calcula para 3,5
DOUBLE ("3, 5 pol")	Calcula para 3,5
DOUBLE ("O círculo mede 3,5 pol. de diâmetro")	Calcula para 0,0
DOUBLE (MPOINT (0, 0, 1))	Calcula para 1,0
DOUBLE (MPOINT (3, 4, 5))	Calcula para 7.0711

Coerção de sequências

STR(<Expressão>) - Força o valor da expressão para tipo sequência

STR(4)	É avaliado para "4"
STR(4,5)	É avaliado para "4,5"
STR("Olá pessoal")	É avaliado para "Olá pessoal"
STR(MPOINT(3,4,5))	É avaliado para "3, 4, 5"

Coerção de pontos

MPOINT(<Expressão1>, <Expressão2>, <Expressão3>) - Força os valores das expressões para tipo ponto após forçar cada expressão para tipo duplo.

MPOINT (1, 1, 1)	É avaliado para ponto 1,0,1.0,1.0
------------------	-----------------------------------

<code>MPOINT (1,1, 1,1, 1,1)</code>	É avaliado para ponto 1,1, 1,1, 1,1
<code>MPOINT ("1", "1", "1")</code>	É avaliado para ponto 1,0,1.0,1.0
<code>MPOINT (3, 4,5, "5,6")</code>	É avaliado para ponto 3,0, 4,5, 5,6
<code>MPOINT (MPOINT (1, 0, 0) , MPONTO (0,1,0) , MPOINT (3,4,5))</code>	É avaliado para 1.0, 1.0, 7.0711

Coerção de operandos e expressões do tipo misto

O calculador de expressões força automaticamente as variáveis para expressões de tipo misto. Se o resultado de uma expressão não for o esperado devido à coerção automática, o uso de operadores de coerção em alguns casos produz o resultado desejado. A seguir são apresentados exemplos de coerções automáticas em expressões de tipo misto.

"CIR" + 1

É avaliado para "CIR1"

"2" + 2

É avaliado para 4

"O valor de 2+2 é " + 2 + 2

avalia que "O valor de 2+2 é 22". Isso porque o PC-DMIS avalia as expressão da esquerda para a direita, e a parte esquerda da expressão é uma sequência.

"O valor de 2+2 é " + (2 + 2)

É avaliado para "O valor de 2+2 é 4"

LINE1.XYZ > 2

É avaliado para 1 se a distância do centróide de LINHA1 a partir da origem for maior do que 2

LINE1.XYZ > LINE2.XYZ

É avaliado para 1 se o centróide de LINHA1 estiver mais distante da origem do que o centróide de LINHA2

LINE1.XYZ = LINE2.XYZ

É avaliado para 1 se os centróides de LINHA1 e de LINHA2 forem iguais (não ocorre coerção nesse caso)

DOUBLE(LINE1.XYZ) = DOUBLE(LINE2.XYZ)

É avaliado para 1 se os centróides estiverem numa mesma distância da origem

11% 3.1

É avaliado para 2 (% é o operador do módulo designado para trabalhar com inteiros. Ele retorna o restante a partir da divisão discreta. $11\%3 = 2$.)

CÍRCULO1.TOQUE [3.2].X

É avaliado para o valor de X medido do terceiro toque de Círculo1. O argumento 3.2 é forçado automaticamente para um inteiro com o valor 3.

Expressões de ID

Muitos dos comandos do PC-DMIS usam IDs de elementos como parâmetros. Por exemplo, os elementos construídos usam IDs para indicar que os elementos devem ser usados como entradas para o elemento construído. As expressões de ID permitem ao usuário referir-se a uma ocorrência específica de um elemento, a um grupo de elementos de nomes parecidos, a uma ocorrência de um elemento dentro de uma chamada para uma subrotina ou a um elemento em uma rotina de medição externa.

ID da matriz de elementos

Use uma ID de matriz de elementos para se referir a uma ocorrência específica de um elemento ou a um intervalo de ocorrências de elementos. Por exemplo, se o elemento "Círculo1" estivesse localizado em um loop while que executasse um loop cinco vezes, então cinco ocorrências do círculo existiriam na saída do loop. Para se referir a uma ocorrência específica das cinco ocorrências de "Círculo1", use a sintaxe de matriz de elementos descrita em "Matrizes de elementos:" onde "Círculo1[1]" pode ser referido à primeira instância, "Círculo1[2]" pode ser referido à segunda instância, etc. Para se referir a um intervalo de ocorrências, use a notação ... "Círculo1[1..3]" refere-se da primeira à terceira ocorrências de Círculo1. "Círculo1[3..5]" refere-se da terceira à quinta ocorrências de Círculo1. "Círculo1[1..5]" refere-se da primeira à quinta ocorrências de Círculo1. Ao referir-se a um intervalo de elementos, o conjunto é tratado e se comporta como um conjunto construído.

Curingas de IDs

Use Curingas de IDs para se referir a um conjunto de elementos de nomes parecidos. Os dois caracteres curinga são "*" e "?". (Consulte Seleção de elementos usando correspondência de metacaracteres no capítulo Edição da exibição do CAD para obter informações adicionais.)

O caractere de asterisco "*" refere-se a 0 ou mais ocorrência de qualquer caractere. Para se referir a um conjunto de todos os elementos que começam com as letras "CIR", use a ID de expressão "CIR*". Essa sintaxe cria um conjunto de elementos que inclui todos os elementos com IDs que começam com "CIR", por exemplo "CÍRCULO1", "CÍRCULO2", "CIR3" ou "CIR".



Se CIR3 tem várias execuções, somente a medição mais recente é utilizada. Para obter diferentes ocorrências das execuções, você poderia usar a seguinte expressão: CIR?[1..3]

O caractere de interrogação "?" refere-se a uma única ocorrência de qualquer caractere.



A expressão de ID "MEU???1" cria um conjunto de elementos de seis caracteres de comprimento, começando com "MEU" e terminando com "1", tais como "MEUCIR1", "MEUCON1", "MEULIN1" ou "MEUELEM21".

IDs para elementos dentro de subrotinas, scripts do Basic ou rotinas de medição externas

Subrotinas podem estar localizadas dentro da rotina de medição atual ou uma rotina de medição externa. Quando a subrotina está localizada na mesma rotina de medição que a chamada para a subrotina, a sintaxe de ID de matriz de elemento explicada em "Matrizes de elemento:" pode ser usada para fazer referência a instâncias individuais de um elemento criado na subrotina. Entretanto, quando a subrotina está localizada em uma rotina de medição externa, a seguinte sintaxe pode ser usada para se referir a quaisquer elementos criados na subrotina: "<ID de chamar sub>:<IDElem>".

Por exemplo, se um elemento chamado "F1" estivesse localizado em uma subrotina externa que fosse chamada de um comando de chamada de subrotina com a ID "CS1", então a expressão de ID "CS1:F1" poderia ser usada para se referir àquele elemento.



Este exemplo simplesmente ilustra o uso da sintaxe CS1.F1 e não deve ser utilizado.

Rotina 1: PLUS1.PRG

```
SUB-ROTINA/MAIS1, A1 = 0, A2 = 0, A3 = 0

PNT1=ELEM/PONTO,CARTES

TEÓR /A1+1,A2+1,A3+1,0,0,1

REAL/3,1,1,0,0,1

MED/PONTO,1

TOQUE /BÁSICO,A1+1,A2+1,A3+1,0,0,1,0,0,0

FIMMED//

FIMSUB/
```

Rotina 2: TEST.PRG

```
CS1 =CHAMARSUB/MAIS1,D:\V30\WINDEBAG\MAIS1.PRG: 3,3,3,,

DIM D1= LOCALIZAÇÃO DO PONTO CS1:F1 UNIDADES=IN , $

GRÁFICO=DESATIVADO TEXTO=DESATIVADO MÚLT=10.00 SAÍDA=AMBOS

AX NOMINAL +TOL -TOL MED MÁX MIN DESV TOLEXT

X 3,0000 0,0000 0,0000 3,0000 3,0000 3,0000 0,0000 0,0000

----#----

FIM DA DIMENSÃO D1
```

Os scripts do Basic criam e excluem objetos dinamicamente. Use a sintaxe "<ID script do Basic>:<ID elem>" para se referir a um elemento criado por um script básico. Por exemplo, se um script básico com ID "BS1" cria um elemento com ID "F2", use a expressão de ID "BS1:F2" para se referir àquele elemento.

Você pode usar o comando de anexar para anexar rotinas externas ao PC-DMIS. Para se referir a elementos na rotina de medição anexada, utilize a seguinte sintaxe: "<Anexar ID Rotina>:<ID Elem>". Para se referir ao elemento "F3" na rotina de medição anexada com a ID "ENGRENAGEM1", utilize a expressão, "ENGRENAGEM1:F3". (Para mais informações, consulte "Anexação de uma rotina de medição externa" no capítulo "Adição de elementos externos".)

Combinações de expressões de ID

Você pode usar Expressões de IDs de matrizes, Expressões de IDs curingas e Expressões de IDs de subrotinas externas, scripts básicos e de rotinas de medição externas em combinação. Por exemplo, para se referir à terceira ocorrência de todos os elementos que começam com as letras "CIR" em uma rotina de medição externa anexada à ID "BOLTPAT", use a expressão de ID "BOLTPAT:CIR*[3]".

Além disso, você pode usar as Expressões de ID em expressões regulares. Portanto, o centróide medido do conjunto de elementos acima poderia ser atribuído à variável com a seguinte expressão:

```
ATRIBUIR/V1=BOLTPAT:CIR*[3].XYZ
```

Além disso, as Expressões de ID podem ser usadas em expressões regulares.

Portanto, o centróide medido do conjunto de elementos acima poderia ser atribuído à variável com a seguinte expressão:

```
ATRIBUIR/V1=BOLTPAT:CIR*[3].XYZ
```

Acesso a propriedades de objeto de um relatório

Você pode criar seu próprio relatório personalizado e modelos de rótulos. O PC-DMIS usa-os para exibir dados de relatório em uma janela Relatório (consulte **Visualizar | Janela Relatório**). Você usa editores de modelo para criar os modelos. Os editores utilizam uma interface como a do Visual Basic que permite inserir, relocalizar e dimensionar componentes especiais chamados de "objetos".

Cada objeto consiste em "propriedades" que definem como aparece e informações que contém. Algumas dessas propriedades são comuns a todos os outros objetos, algumas são comuns apenas a objetos relacionados e outras são exclusivas a esse objeto específico.

A linguagem de expressão do PC-DMIS pode consultar o Relatório carregado atual e armazenar valores de propriedade de um objeto específico em uma variável. Ela pode obter valores do tipo Sequência, Inteiro e Real usando essa sintaxe:

Sintaxe de consulta de propriedade



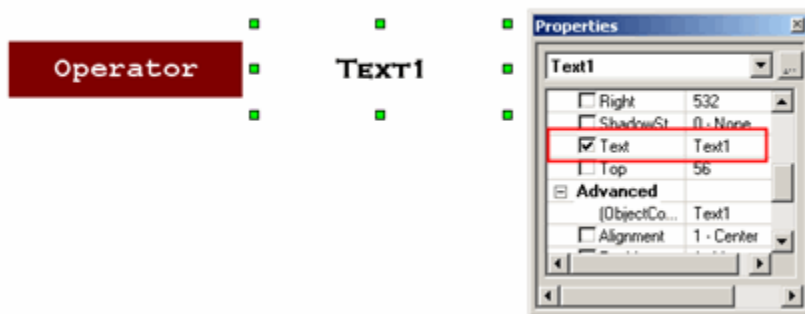
```
ATRIBUIR/V1=Relatório.<Nome do objeto>.<Nome da
propriedade>
```

O relatório é uma referência para o relatório carregado atualmente. <Nome do objeto> é o nome específico do objeto. <Nome da propriedade> é um nome de propriedade válido para esse objeto.

Exemplo



Suponha que seu modelo de relatório tenha um objeto de texto chamado "Texto1" que você deseja usar no relatório final para exibir o nome do operador. O PC-DMIS armazena a sequência de caracteres real que representa o nome do operador na propriedade **Texto** do objeto. Por padrão, a propriedade de texto (texto exibido) inicialmente tem o valor de "Texto1" (veja a figura abaixo). Por ser uma propriedade atribuída pelo usuário, o valor da propriedade muda quando você digita o nome durante a execução.



Caixa de diálogo Propriedades que mostra um objeto selecionado e a propriedade a consultar

Para usar o código de linguagem de expressão para consultar essa propriedade "Texto" do objeto Texto e obter os dados digitados, você usaria o comando a seguir:

```
ATRIBUIR/V1=Relatório.Texto1.Text
```

Nesse código:

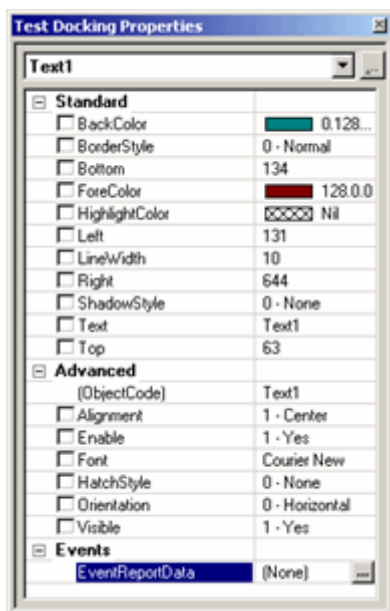
"Relatório" informa o código a ser examinado no relatório carregado na janela Relatório.

"Texto1" instrui a procurar o objeto chamado "Texto1".

"Text" instrui a procurar a propriedade "Text" dentro desse objeto. Em seguida, o valor da propriedade "Text" é passado para a variável V1, que pode então ser mais manipulada ou exibida com o uso da linguagem de expressão do PC-DMIS.

Detecção de propriedades

Para detectar as propriedades que estão associadas com um objeto específico, acesse o modelo Relatório no editor de gabarito Relatório (**Arquivo | Relatórios | Editar | Modelo de relatório**), selecione o objeto e clique com o botão direito do mouse para exibir sua folha de propriedades.



Folha de propriedades para um objeto Texto

A folha de propriedades contém duas colunas. A coluna da esquerda exibe o nome da propriedade. A coluna da direita exibe o valor atual. Certifique-se de que o nome da propriedade exato no código de expressão.



Quando você consulta valores da propriedade, pode perceber que algumas propriedades retornam um valor numérico aparentemente inútil. Geralmente, isso acontece quando a propriedade tem uma lista definida de opções disponíveis. O PC-DMIS retorna um valor de intervalo para a propriedade selecionada que não está relacionado à propriedade exibida.

Por exemplo, o objeto **Texto** tem uma propriedade **Orientação** com esses valores:

- 0 - Horizontal
- 1 - Vertical para cima
- 2 - Vertical para baixo

Contudo, se você obtiver o valor usando uma linguagem de expressão do PC-DMIS, o software irá retornar o seguinte:

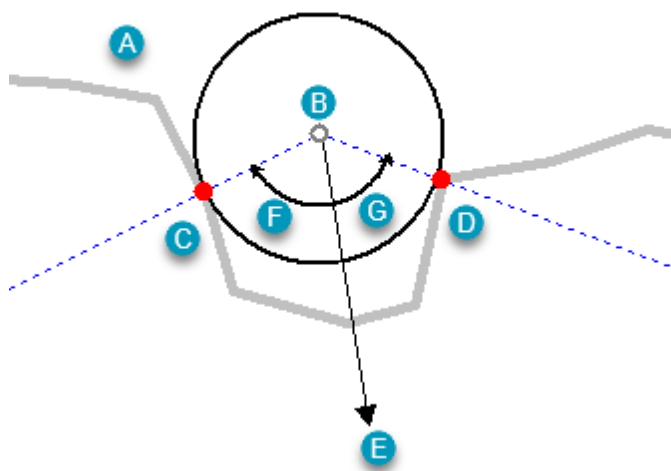
- 0 (para Horizontal)
- 90 (para Vertical para cima)
- -900 (para Vertical para baixo)

Pode ser necessário passar por um processo de tentativa e erro para determinar quais valores retornados correspondem ao valor exibido na folha de propriedade.

Acessando informações de varredura construída círculos mínimos

Você pode usar expressões do PC-DMIS para trazer informações de um elemento círculo que é construído com um determinado raio em um ponto mínimo ao longo de uma varredura linear. Para obter mais informações, consulte o tópico Construção de um círculo em um ponto mínimo de varredura no capítulo Construção de novos elementos a partir de elementos existentes.

Quando você constroi um elemento circular mínimo de varredura, o círculo em última análise utiliza um vetor (nomeado de vetor para baixo) para encontrar a linha de varredura. Ele somente encontra a linha em dois lugares chamados de pontos de contato (CONTACTPOINT1 e CONTACTPOINT2). O PC-DIMS pode usar esses pontos para determinar os ângulos do vetor para baixo para esses pontos de contato (CONTACTANGLE1 e CONTACTANGLE2). Por exemplo, considere o diagrama:



A - A linha de varredura para a qual o círculo é construído.

B - A posição final XYZ do centróide do círculo.

C - O ponto de contato à esquerda do Vetor para baixo. É denominado CONTACTPOINT1.

D - O ponto de contato à direita do Vetor para baixo. É denominado CONTACTPOINT2.

E - O Vetor para baixo.

F - O ângulo do Vetor para Baixo para o CONTACTPOINT1. É denominado CONTACTANGLE1.

G - O ângulo do Vetor para Baixo para o CONTACTPOINT2. É denominado CONTACTANGLE2.

As expressões detalhadas abaixo funcionam apenas com este tipo de elemento círculo construído. Pode-se também usar o CONTACTPOINT2 na sintaxe abaixo para retornar a informação equivalente usando em seu lugar o segundo ponto de contato.

`ATRIBUIR/V1=CIR1.CONTACTPOINT1.XYZ`

Retorna as informações de ponto XYZ para o primeiro ponto de contato do círculo com a linha, CONTACTPOINT1.

`ATRIBUIR/V1=CIR1.CONTACTPOINT1.X`

Retorna as informações de X para CONTACTPOINT1.

`ATRIBUIR/V1=CIR1.CONTACTPOINT1.Y`

Retorna as informações de Y para CONTACTPOINT1.

`ATRIBUIR/V1=CIR1.CONTACTPOINT1.Z`

Retorna as informações de Z para CONTACTPOINT1.

`ATRIBUIR/V1=CIR1.CONTACTPOINT1.IJK`

Retorna o vetor IJK do CONTACTPOINT1 para o centróide do círculo.

`ATRIBUIR/V1=CIR1.CONTACTPOINT1.I`

Retorna o valor de I do vetor CONTACTPOINT1 IJK acima.

`ATRIBUIR/V1=CIR1.CONTACTPOINT1.J`

Retorna o valor de J do vetor CONTACTPOINT1 IJK acima.

`ATRIBUIR/V1=CIR1.CONTACTPOINT1.K`

Retorna o valor de K do vetor CONTACTPOINT1 IJK acima.

`ATRIBUIR/V1=CIR1.CONTACTANGLE1`

Retorna o ângulo do Vetor para Baixo para o CONTACTPOINT1.

`ATRIBUIR/V1=CIR1.CONTACTANGLE2`

Retorna o ângulo do Vetor para Baixo para o CONTACTPOINT2.

`ATRIBUIR/V1=CIR1.ÂNGULOCONTATO`

Retorna a soma dos valores absolutos do ÂNGULOCONTATO1 e do ÂNGULOCONTATO2. Não deve ser maior do que 180 graus.