

## Sommario

Uso di espressioni e di variabili .....	1
Uso di espressioni e di variabili: Introduzione .....	1
Uso delle espressioni in una routine di misurazione .....	2
Visualizzazione dei valori delle espressioni .....	2
Come conservare solo i valori delle espressioni .....	2
Uso delle espressioni con la diramazione .....	3
Uso delle espressioni nelle operazioni sui file .....	3
Creazione di espressioni con il generatore di espressioni .....	3
Creazione di un'espressione attraverso la digitazione .....	4
Creazione di un'espressione con il generatore di espressioni.....	4
Verifica della correttezza delle espressioni .....	5
Tipo di elemento di un'espressione .....	6
ID .....	6
Estensione .....	7
Seconda prolunga .....	9
Pulsante Aggiungi .....	9
Casella di modifica .....	10
Riquadro Descrizione .....	10
Uso di variabili con espressioni.....	10
Assegnazione di valori alle variabili tramite la finestra di dialogo Assegnazione ...	11
Informazioni sui componenti delle espressioni .....	12
Tipi di operandi.....	12

Valori letterali .....	12
Riferimenti .....	13
Variabili .....	20
Strutture .....	24
Puntatori.....	26
Array .....	28
Operatori delle espressioni .....	42
Priorità.....	44
Funzioni .....	44
Forzatura degli operandi .....	93
Espressioni con ID .....	96
Accesso alle proprietà di un oggetto Rapporto .....	100
Accesso alle informazioni da cerchio di scansione minimo costruito .....	104

# Uso di espressioni e di variabili

---

## Uso di espressioni e di variabili: Introduzione

Un'espressione è una condizione definita dall'utente ed utilizzata con i comandi del controllo di flusso di PC-DMIS. Con le istruzioni di controllo flusso, è possibile testare tali condizioni nella routine di misurazione. È possibile determinare l'operazione che verrà eseguita da PC-DMIS se una condizione viene soddisfatta o meno.

Le espressioni sono importanti per far sì che PC-DMIS svolga operazioni specifiche. L'uso delle espressioni insieme ai comandi di controllo del flusso consente di sfruttare meglio le potenti funzionalità di PC-DMIS.

Questo capitolo spiega come creare e usare le espressioni nella finestra di modifica di PC-DMIS. Quando si lavora con le espressioni è necessario porre la finestra di modifica nella modalità di comando. Ciò consente di vedere direttamente il codice della finestra di modifica.

In questo capitolo verranno trattati i seguenti argomenti principali:

- Uso delle espressioni in una routine di misurazione
- Creazione di espressioni con il generatore di espressioni
- Uso di variabili con espressioni
- Informazioni sui componenti delle espressioni
- Accesso alle proprietà oggetto di un rapporto
- Accesso alle informazioni da cerchio minimo costruito mediante scansione



Per informazioni sulle espressioni di rapporto, vedere "Informazioni sulle espressioni di rapporto" nel capitolo "Rapporto dei risultati di misura".

## Uso delle espressioni in una routine di misurazione

La finestra di modifica di PC-DMIS permette di inserire espressioni nella maggior parte dei campi modificabili. I campi modificabili sono quelli solitamente evidenziati in giallo quando si preme il tasto di tabulazione all'interno della finestra di modifica in modalità di comando. I campi che modificano il tipo di elemento non possono contenere espressioni.



La casella di un elemento automatico che specifica il tipo di elemento automatico, ad esempio Punto di superficie, Cerchio automatico, Asola rotonda automatica e così via, non può contenere espressioni.

Negli argomenti secondari di questo argomento vengono fornite informazioni complete sulle espressioni disponibili.

## Visualizzazione dei valori delle espressioni

Per visualizzare il valore di un'espressione, posizionare il cursore sull'espressione desiderata e lasciarlo in tale posizione per almeno un secondo. L'espressione verrà valutata e verrà visualizzata, insieme al relativo valore corrente, in una piccola finestra popup di colore giallo immediatamente sotto il cursore.

## Come conservare solo i valori delle espressioni

Per risolvere immediatamente un'espressione nella finestra di modifica, conservando solo il valore procedere come segue.

1. Selezionare il testo dell'espressione nella finestra di modifica.
2. Anteporre al testo dell'espressione il carattere ` (accento grave).



Si supponga di inserire l'espressione ``1/7` in un campo numerico. L'espressione viene risolta immediatamente e nel campo viene inserito soltanto il valore (**0.143**).

## Uso delle espressioni con la diramazione

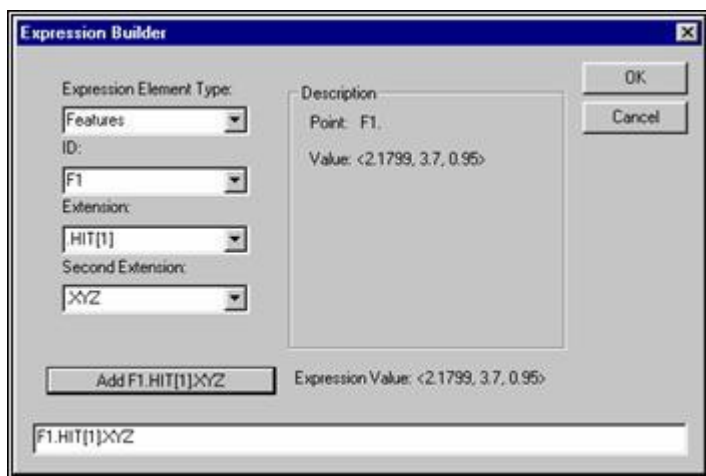
I comandi di controllo di flusso usano espressioni per determinare il flusso di esecuzione della routine. Per informazioni sull'esecuzione delle diramazioni vedere il capitolo "Esecuzione di diramazioni mediante il controllo del flusso".

## Uso delle espressioni nelle operazioni sui file

Per la scrittura o la lettura dei dati in un file di dati esterno spesso vengono utilizzate variabili e altre espressioni che permettono di gestire e memorizzare o visualizzare efficacemente i dati. Per ulteriori informazioni vedere il capitolo "Uso di input/output di file".

---

## Creazione di espressioni con il generatore di espressioni



*Finestra di dialogo Generatore di espressioni*

PC-DMIS consente di creare le espressioni nella finestra di modifica digitandole semplicemente oppure utilizzando l'interfaccia della finestra di dialogo **Generatore di**

**espressioni.** L'opzione di menu **Modifica | Espressione** visualizza la finestra di dialogo **Generatore di espressioni**.

In questa finestra di dialogo è possibile creare un'espressione e inserirla in un campo modificabile. La finestra di dialogo **Generatore di espressioni** viene visualizzata anche premendo il tasto funzione F2 quando il cursore è posizionato su un campo che può contenere espressioni.

Nella finestra di dialogo **Generatore di espressioni** è riportato l'elenco di tutti i tipi di operatori e di funzioni disponibili per le espressioni.

## Creazione di un'espressione attraverso la digitazione

Per creare un'espressione digitandola direttamente nella finestra di modifica procedere come segue.

1. Aprire la finestra di modifica (**Visualizza | Finestra di modifica**).
2. Attivare la modalità Comando per la finestra di modifica.
3. Premere il tasto di tabulazione per spostare il cursore su un campo modificabile nel quale si desidera inserire l'espressione. I campi evidenziati in giallo sono considerati "modificabili".
4. Inserire l'espressione.

## Creazione di un'espressione con il generatore di espressioni



Perché l'opzione Espressione sia abilitata è necessario trovarsi in modalità Comando.

Per immettere un'espressione utilizzando la finestra di dialogo Generatore di espressioni (**Modifica | Espressione**):

1. Aprire la finestra di modifica (**Visualizza | Finestra di modifica**).
2. Portare la finestra di modifica in modalità Comando (**Visualizza | Modalità comando**).
3. Spostare il cursore su un campo modificabile nel quale si desidera inserire l'espressione.
4. Premere il tasto funzione F2 con il cursore posizionato su un campo in cui è possibile inserire espressioni. Verrà visualizzata la finestra di dialogo

**Generatore di espressioni.** La finestra di dialogo **Generatore di espressioni** contiene l'elenco di tutti i tipi di operatori, operandi e funzioni. In questa finestra di dialogo è possibile visualizzare i seguenti elementi:

- Tipi di espressioni disponibili
  - Variabili
  - Funzionalità
  - Dimensioni
  - Allineamenti
  - Comments
5. Selezionare il tipo di elemento dell'espressione nel primo elenco a discesa. In base alla selezione effettuata, vengono visualizzate ulteriori caselle combinate.
  6. Selezionare l'ID desiderato nell'elenco a discesa **ID**.
  7. Selezionare una prolunga nell'elenco a discesa **Prolunga**.
  8. Selezionare un'altra prolunga nell'elenco a discesa **Seconda prolunga**. Se l'espressione è valida, il pulsante **Aggiungi** viene attivato.
  9. Fare clic sul pulsante **Aggiungi**. L'espressione viene visualizzata nella casella di modifica.
  10. Fare clic sul pulsante **OK**. A questo punto, l'espressione viene visualizzata nel punto in cui è posizionato il cursore all'interno della finestra di modifica.



È possibile aprire la finestra di dialogo **Generatore di espressioni** anche da queste altre finestre di dialogo:

- La finestra di dialogo **Espressione If** - Selezionare **Inserisco | Comando di controllo di flusso | If Goto**. Fare clic sul pulsante **Espressione**.
- La finestra di dialogo **Assegnazione** - Selezionare **Inserisci | Assegnazione**. Fare clic sul pulsante **Assegna a** o **Assegna da**.

Una volta creata, l'espressione viene inserita automaticamente nella successiva posizione valida nella finestra di modifica.

## Verifica della correttezza delle espressioni

Quando il cursore viene spostato in un campo diverso da quello in cui è stata aggiunta l'espressione, PC-DMIS tenta di verificare la correttezza dell'espressione. Se l'espressione risulta non corretta, viene visualizzato un messaggio di errore indicante un

numero non valido oppure il testo dell'espressione viene visualizzato rosso. Anche il testo delle espressioni che fanno riferimento ad oggetti inesistenti verrà visualizzato in rosso.

Poiché il test per la correttezza dell'espressione viene eseguito nel momento in cui si esce dal campo, un campo che diventa rosso a causa di un riferimento a un oggetto non esistente, (ad es. Cerchio1.X) resterà rosso anche se viene aggiunto il nuovo oggetto (ad es. CERCHIO1). In altre parole, il campo rimane rosso fino a quando non viene verificata di nuovo la correttezza dell'espressione.

Per eseguire nuovamente la verifica della correttezza dell'espressione, effettuare le seguenti operazioni.

1. Spostare il cursore sul campo dell'espressione.
2. Premere il tasto F2 . Viene visualizzata di nuovo la finestra di dialogo **Generatore di espressioni**. Tutte le modifiche all'espressione vengono visualizzate nella casella di modifica.
3. Premere INVIO per chiudere la finestra di dialogo.

## Tipo di elemento di un'espressione

L'elenco a discesa Tipo di elemento espressione disponibile nella finestra di dialogo **Generatore di espressioni (Modifica | Espressione)** riporta i tipi di elemento che è possibile inserire nelle espressioni. Queste sono le seguenti.

- Funzioni
- Operatori
- Allineamenti
- Comments
- Dimensioni
- Funzionalità
- Variabili

## ID

L'elenco a discesa **ID** della finestra di dialogo **Generatore di espressioni (Modifica | Espressione)** contiene gli elementi disponibili in base al tipo di elemento dell'espressione selezionato nell'elenco a discesa **Tipo di elemento di un'espressione**.





L'elenco delle voci disponibili dipende dall'elemento selezionato:

- Quando si seleziona **Funzioni e operatori** nell'elenco a discesa **Tipo di elemento di un'espressione**, l'elenco a discesa **ID** contiene le funzioni e gli operatori disponibili.
- Quando si seleziona **Elementi** nell'elenco a discesa **Tipo di elemento nell'espressione**, nell'elenco a discesa **ID** mostra gli ID degli elementi presenti nella routine di misurazione.

## Estensione

L'elenco a discesa **Estensione** della finestra di dialogo **Generatore di espressioni (Modifica | Espressione)** diventa disponibile quando l'elemento scelto nell'elenco a discesa **ID** richiede l'aggiunta di una estensione per definire un elemento dell'espressione utilizzabile. L'elenco a discesa **Estensione** mostra le estensioni disponibili in base alla voce selezionata nell'elenco a discesa **ID**.



Si supponga di selezionare un elemento nell'elenco a discesa **ID**. Nell'elenco a discesa **Estensione** PC-DMIS mostra le estensioni che è possibile usare come dati di riferimento dell'elemento selezionato (ad esempio "X", "Y", "Z", "Diametro", "Lunghezza" e così via).

Le possibili estensioni riguardano questi tipi di dati teorici o misurati:

### Misurati

- AllTutti i valori dell'elemento sono assegnati alla variabile. Vedere l'esempio seguente.
- X – I valori X misurati dei punti
- Y – I valori Y misurati dei punti
- Z – I valori Z misurati dei punti
- XYZ – I valori XYZ misurati dei punti
- I – I valori I misurati dei punti
- J – I valori J misurati dei punti
- K – I valori K misurati dei punti

- IJK – I valori IJK misurati dei punti

## Teorici

- TX – I valori teorici X dei punti
- TY – I valori teorici Y dei punti
- TZ – I valori teorici Z dei punti
- TXYZ – I valori teorici XYZ dei punti
- TI – I valori teorici I dei punti
- TJ – I valori teorici J dei punti
- TK – I valori teorici K dei punti
- TIJK – I valori teorici IJK dei punti



Si supponga che questo frammento di codice sia parte di una routine di misurazione:

```
F1 =PUNTO/GENERICO,DIPENDENTE,CARTESIANO,$
```

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

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

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

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

```
ASSEGNA/MIO_ELEM=F1.ALL
```

```
ASSEGNA/V1=MIO_ELEM.X
```

```
ASSEGNA/V2=MIO_ELEM.TX
```

Quando la routine di misurazione esegue il frammento di codice precedente, PC-DMIS assegna le variabili con questi valori:

```
V1 = 7.98
```

```
V2 = 8
```

## Seconda prolunga

L'elenco a discesa **Seconda prolunga** è disponibile solo se l'elemento scelto nell'elenco a discesa **Prolunga** richiede l'aggiunta di una seconda estensione per formare un elemento dell'espressione utilizzabile.



Si supponga di voler fare riferimento al valore nominale dell'asse X della dimensione "D1". Ecco cosa fare.

1. Scegliere **D1** nell'elenco a discesa **ID**.
2. Selezionare **X** nell'elenco a discesa **Prolunga**.
3. Selezionare **Nom** nell'elenco a discesa **Seconda prolunga**.

## Pulsante Aggiungi

Il pulsante **Aggiungi** diventa disponibile ogni volta che si seleziona da un elenco a discesa un elemento dell'espressione utilizzabile o completo. Il pulsante visualizza il testo da aggiungere all'espressione.

Ad esempio, se si scelgono per l'espressione i seguenti elementi:

- Dimensioni nell'elenco **Tipo di elemento espressione**
- D1 dall'elenco **ID**
- X nell'elenco **Estensione**
- Nom dall'elenco **Seconda estensione**

il pulsante **Aggiungi** sarà abilitato e si avrà il seguente testo: **Aggiungi D1.X.NOM.**

Quando si fa clic sul pulsante **Aggiungi**, il testo viene visualizzato nella casella di modifica in fondo alla finestra di dialogo.



Quando si fa clic sul pulsante **OK**, PC-DMIS aggiunge il testo presente nella casella di modifica nel campo dell'espressione nel punto in cui si trova il cursore all'interno della finestra di modifica. Se si seleziona una voce nel campo dell'espressione all'interno della finestra di modifica e il testo da aggiungere contiene delle parentesi, l'elemento selezionato verrà inserito all'interno delle parentesi del testo aggiunto.

## Casella di modifica

Nella parte inferiore della finestra di dialogo **Generatore di espressioni (Modifica | Espressione)** c'è una casella di modifica che mostra l'espressione corrente. È possibile immettere l'espressione direttamente in questa casella oppure selezionare il pulsante **Aggiungi**.

## Riquadro Descrizione

La finestra di dialogo **Generatore di espressioni (Modifica | Espressione)** contiene anche un riquadro **Descrizione** che fornisce informazioni sugli elementi selezionati negli elenchi a discesa. Accanto al pulsante **Aggiungi** è disponibile un campo contenente il valore assunto al momento dall'espressione.



Le espressioni con un valore pari a 0 non sono valide.

---

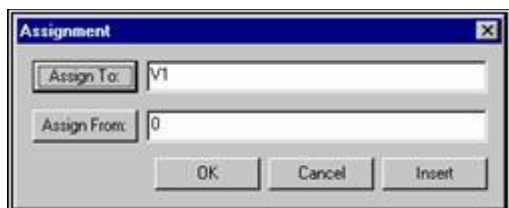
## Uso di variabili con espressioni

Le variabili sono oggetti che contengono valori. Fanno riferimento a operandi di tipo numero intero, numero reale, stringa o punto. Le variabili sono essenziali per l'uso delle espressioni. Ad una variabile sono associati un nome e un valore. Il nome serve per accedere al valore della variabile. Il nome è costante, mentre il valore può essere modificato. Per assegnare un valore a una variabile, si può usare il comando [ASSEGNA/](#).

Ad esempio, l'istruzione [ASSEGNA/V1=2](#) crea una variabile il cui nome è V1 e il cui valore è 2. [ASSEGNA/V2=V1+2](#) accede al valore di V1. Se V1 ha ancora il valore 2 al momento dell'esecuzione di questa istruzione di assegnazione, il valore di V2 sarà 4.

Per ulteriori informazioni sulle variabili, vedere "Variabili".

## Assegnazione di valori alle variabili tramite la finestra di dialogo Assegnazione



*Finestra di dialogo Assegnazione*

L'opzione del menu **Inserisci | Assegnazione** consente di visualizzare la finestra di dialogo **Assegnazione**. Usare questa finestra di dialogo per assegnare un valore ad una variabile o ai dati di un elemento, una dimensione o un allineamento della routine di misurazione. L'uso del comando di assegnazione richiede una conoscenza di base delle espressioni di PC-DMIS.

Pulsante **Assegna a** - Questo pulsante indica la variabile che riceve il valore calcolato nella casella **Assegna da**. Le informazioni selezionate mediante il pulsante **Assegna a** vengono inserite nella casella **Assegna a**. Questo valore può essere il nome di una variabile oppure un riferimento ai dati di un elemento, una dimensione o un allineamento.

*Il risultato della risoluzione di un'espressione matematica relativa ad un valore viene indicato mediante il termine "valutato".*

Pulsante **Assegna da** - Questo pulsante inserisce il valore assegnato nella casella **Assegna da**. Se questa casella contiene un'espressione, l'espressione viene valutata durante l'esecuzione e il relativo risultato o il valore del calcolo viene assegnato all'oggetto indicato nella casella **Assegna a**.

Pulsante **Inserisci** - Questo pulsante inserisce un comando di assegnazione nella routine di misurazione mantenendo aperta la finestra di dialogo **Assegnazione**. In questo modo sarà possibile inserire una serie di comandi di assegnazione senza dover chiudere la finestra di dialogo.

### **Argomenti correlati:**

Informazioni sui componenti delle espressioni

Accesso alle proprietà oggetto di un rapporto

---

# Informazioni sui componenti delle espressioni

Le espressioni presentano questi tipi di operandi:

- Numeri interi
- Numeri reali
- Stringhe
- punti
- Puntatori degli elementi
- Array
- Funzioni

Questi operandi verranno illustrati dettagliatamente nelle sezioni successive.

## Tipi di operandi

Gli operandi possono esistere sotto forma di:

- Valori Letterali
- Riferimenti
- Variabili
- Strutture
- Puntatori
- Array

## Valori letterali

**\*Interi:** 1, -6, 209

**Reali:** 1, -6, 2,4, -0,1, 345.6789

**Stringhe:** "Ciao", "47", "CERCHIO 1"

**Punti:** non è possibile usare una rappresentazione letterale per i punti. Tuttavia, è possibile creare punti da altre rappresentazioni letterali mediante la funzione MPOINT: ad esempio, `MPOINT(0,0,1)`, `MPOINT(2.2, 3.1, 4.0)`.

**Puntatore:** è il nome di un elemento racchiuso tra parentesi graffe: {CIR1}, {LIN2}, {F3}

**Array:** non è possibile utilizzare una rappresentazione letterale per gli array. Tuttavia, è possibile creare gli array da altri valori letterali utilizzando la funzione ARRAY: ad esempio, `ARRAY(3, 5, 6)`, `ARRAY("Hello", 2.3, 9)`. Tali funzioni creano tre array di elementi con gli elementi numeri interi 3, 5 e 6 nel primo esempio e l'elemento stringa "Hello", l'elemento doppio 2,3 e l'elemento numero intero 9 nel secondo esempio.

**Funzioni:** non è possibile utilizzare una rappresentazione letterale per le funzioni. Le funzioni possono essere definite mediante la parola chiave `FUNZIONE` a cui si accede tramite gli ID delle variabili. Ad esempio, `ASSEGNA/Aggiungi2=FUNCTION+2` definisce una funzione che accetta un solo argomento a cui aggiunge 2. La variabile `Aggiungi2` viene assegnata alla funzione. È possibile richiamare la funzione usando la variabile `Aggiungi2` nel modo seguente. `ASSEGNA/Risultato= Aggiungi2(5)`. Alla variabile `Risultato` viene assegnato il valore 7.



I valori letterali numerici vengono interpretati come numeri reali, a meno che l'operatore o la funzione non implicino l'uso di numeri interi. Ad esempio l'espressione `10/8` viene valutata come 1.25 e non come 1. Inoltre, è possibile eseguire la divisione discreta anche usando gli operatori di forzatura degli operandi. L'espressione `INT(10) / INT(8)` viene valutata 1.

## Riferimenti

I riferimenti riguardano dati di altri oggetti di una routine di misurazione. Essi utilizzano l'ID di un oggetto della routine di misurazione, seguito da un punto e un'estensione riferita ai dati dell'oggetto.



Se `CERCHIO1` era il nome di un cerchio misurato nella routine di misurazione, `CERCHIO1.X` farà riferimento al valore misurato del componente X di `CERCHIO1`. Tutti i riferimenti vengono valutati in coordinate del pezzo relative all'allineamento corrente.

## Riferimenti di tipo in doppia precisione

Di seguito sono indicate le espressioni di riferimento disponibili.

## Esempi di estensioni valide per i riferimenti agli elementi di tipo in doppia precisione

Formato: <ID elemento>.<estensione> -> `CERCHIO1.X`

`CERCHIO1.X` valore X misurato di `CERCHIO1`

`CERCHIO1.Y` valore Y misurato di `CERCHIO1`

`CERCHIO1.Z` valore Z misurato di `CERCHIO1`

`CERCHIO1.TX` valore X teorico (nominale) di `CERCHIO1`

`CERCHIO1.TY` valore Y teorico (nominale) di `CERCHIO1`

`CERCHIO1.TZ` valore Z teorico (nominale) di `CERCHIO1`

`LINEA1.SX` valore X misurato del punto iniziale della `LINEA1`

`LINEA1.SY` valore Y misurato del punto iniziale della `LINEA1`

`LINEA1.SZ` valore Z misurato del punto iniziale della `LINEA1`

`LINEA1.TSX` valore X teorico del punto iniziale della `LINEA1`

`LINEA1.TY` valore Y teorico del punto iniziale della `LINEA1`

`LINEA1.TSZ` valore Z teorico del punto iniziale della `LINEA1`

`LINEA1.EX` valore X misurato del punto finale della `LINEA1`

`LINEA1.EY` valore Y misurato del punto finale della `LINEA1`

`LINEA1.EZ` valore Z misurato del punto finale della `LINEA1`

`LINEA1.TEX` valore X teorico del punto finale della `LINEA1`

`LINEA1.TEY` valore Y teorico del punto finale della `LINEA1`

`LINEA1.TEZ` valore Z teorico del punto finale della `LINEA1`

`PUNTO.I` Componente I misurato del vettore del `PUNTO`

`PUNTO.J` Componente J misurato del vettore del `PUNTO`

`PUNTO.K` Componente K misurato del vettore del `PUNTO`

`PUNTO.TI` Componente I teorico del vettore del `PUNTO`

`PUNTO.TJ` Componente J teorico del vettore del `PUNTO`

`PUNTO.TK` Componente K teorico del vettore del `PUNTO`

`ELEM1.TYP` Il tipo di elemento (ad esempio, cerchio, alloggiamento, cono). È possibile usarlo per modificare il tipo di un elemento generico (`ASSEGNA/Gen1.TIPO = Elem1.TIPO`).

`ELEM1.ALL` Fa riferimento a tutti gli elementi dell'elemento. Consente di copiare informazioni in un elemento generico. (`ASSEGNA/Gen1.TUTTI = Elem1.TUTTI`).



### **Vettore di superficie**

BORDO.SURFI  
BORDO.SURFJ  
BORDO.SURFK  
BORDO.TSURFI  
BORDO.TSURFJ  
BORDO.TSURFK

### **Vettore angolo**

CER.ANGI  
CER.ANGJ  
CER.ANGK  
CER.TANGI  
CER.TANGJ  
CER.TANGK

### **Raggio**

CERCHIO1.R  
CERCHIO1.TR  
CERCHIO1.RAD  
CERCHIO1.TRAD  
CERCHIO1.RADIUS  
CERCHIO1.PR – Raggio polare  
CERCHIO1.TPR – Raggio polare teorico  
CERCHIO1.TRADIUS (sono significativi solo i primi caratteri)

### **Diametro**

CERCHIO1.D  
CERCHIO1.TD  
CERCHIO1.DIAM  
CERCHIO1.TDIAM  
CERCHIO1.DIAMETER  
CERCHIO1.TDIAMETER (sono significativi solo i primi caratteri)

### **Angolo**

CONO.A  
CONO.TA  
CONO.ANG  
CONO.TANG  
CONO.ANGLE

CONO.TANGLE

CONO.PA – Angolo polare

CONO.TPA – Angolo polare teorico (sono significativi solo i primi caratteri)

### Lunghezza

LINEA.L

LINEA.TL

LINEA.LEN

LINEA.TLEN

LINEA.LENGTH

LINEA.TLENGTH (sono significativi solo i primi caratteri)

### Altezza

CILINDRO.PH – Altezza polare

CILINDRO.TPH – Altezza polare teorica

### Raggio, angolo, altezza

PUNTO.RAH – Punto con raggio, angolo e altezza misurati

PUNTO.TRAH – Punto con raggio, angolo e altezza teorici

### Area

BLOB1.AREA - Restituisce il valore dell'area misurato per un elemento Blob.

BLOB1.TAREA - Restituisce il valore dell'area teorica per un elemento Blob.

Ad esempio, `ASSEGNA/V1=BLOB1.AREA` restituisce il valore dell'area misurato dell'elemento BLOB1 e lo assegna alla variabile V1.

Al momento, soltanto l'elemento Blob funziona con queste estensioni di area. Per informazioni sull'elemento Blob, vedere "Vision Blob" nella documentazione di "PC-DMIS Vision".

### Esempi di estensioni valide per i riferimenti a dimensioni di tipo in doppia precisione

Formato: <ID dimensione>.<ASSI>.<elemento dimensione> -> `DIM1.X.NOM`

DIM1.X.NOM	Il valore nominale per la posizione dell'asse X di DIM1
DIM1.X.MEAS	Il valore misurato per la posizione dell'asse X di DIM1
DIM1.X.MAX	La deviazione massima per l'asse X di DIM1
DIM1.X.MIN	La deviazione minima per la posizione dell'asse X di DIM1

DIM1.X.PTOL	Il valore di tolleranza positiva per la posizione dell'asse X di DIM1
DIM1.X.MTOL	Il valore di tolleranza negativa per la posizione dell'asse X di DIM1
DIM1.X.DEV	La deviazione per la posizione dell'asse X di DIM1
DIM1.X.OUTTOL	Il valore fuori tolleranza per la posizione dell'asse X di DIM1
DIM1.Y.NOM	Il valore nominale per la posizione dell'asse Y di DIM1
DIM1.Z.DEV	La deviazione per la posizione dell'asse Z di DIM1
DIM3.PA.MEAS	Il valore misurato per la posizione dell'angolo polare di DIM3
DIM4.M.PTOL	Il valore di tolleranza positivo per l'asse M di DIM4
DIM4.PTOL	Il valore di tolleranza positivo per l'asse M di DIM4 (vedere la nota in "Assi validi" più avanti).
DIM5.BTOL	Il valore di tolleranza bonus in cui DIM5 è una posizione.

#### Assi validi:

X, Y, Z, D, R, A, T, V, L, PR, PA, M, PD, RS, RT, S, H, DD, DF, TP



\* le dimensioni che per definizione hanno solo un asse (come rotondità, concentricità e così via) possono omettere il qualificatore dell'asse. Se si usa il qualificatore dell'asse, tutti questi tipi di dimensioni con un solo asse usano il qualificatore dell'asse M ad eccezione delle dimensioni degli angoli in 2D e 3D che usano il qualificatore dell'asse A.

#### Esempi di estensioni valide per i riferimenti all'allineamento di tipo in doppia precisione

Formato: <ID allineamento>.<Assi o origine dell'allineamento>.<Componente degli assi o dell'origine dell'allineamento> -> [A1.ORIGIN.X](#)

A1.ORIGIN.X	Componente X dell'origine misurata dell'allineamento A1
A2.ORIGIN.Y	Componente Y dell'origine misurata dell'allineamento A2
A1.ORIGIN.Z	Componente Z dell'origine misurata dell'allineamento A1
A1.XAXIS.I	Componente I dell'asse X misurato dell'allineamento A1

A1.YAXIS.J	Componente J dell'asse Y misurato dell'allineamento A1
A1.ZAXIS.K	Componente K dell'asse Z misurato dell'allineamento A1
A1.CORIGIN.X	Componente X dell'origine basata sui dati teorici dell'allineamento A1 (C sta per CAD)
A1.CXAXIS.J	Componente J dell'asse X basato sui dati teorici dell'allineamento A1 (C sta per CAD)

## Riferimenti di tipo punto

Di seguito sono indicate le espressioni di riferimento disponibili.

### Esempi di estensioni valide per i riferimenti agli elementi di tipo punto

Formato: <ID elemento>.<estensione> -> CIRCLE1.XYZ

CIRCLE1.XYZ	Baricentro misurato di CERCHIO1
CIRCLE1.TXYZ	Baricentro teorico di CERCHIO1
LINE1.SXYZ	Punto iniziale misurato di LINEA1
LINE1.TSXYZ	Punto iniziale teorico di LINEA1
LINE1.EXYZ	Punto finale misurato di LINEA1
LINE1.TEXYZ	Punto finale teorico di LINEA1

CIRCLE1.IJK	Vettore misurato di CERCHIO1
CIRCLE1.TIJK	Vettore teorico di CERCHIO1
EDGE.SURFIJK	Vettore superficie misurato di BORDO
EDGE.TSURFIJK	Vettore superficie teorico di BORDO
EDGE.TSURFIJK	Vettore angolo misurato di CERAUTO1
AUTOCIR1.TANGIJK	Vettore angolo teorico di CERAUTO1

### Esempi di estensioni valide per i riferimenti agli allineamenti di tipo punto

Formato: <ID allineamento>.<asse o origine dell'allineamento> -> A1.XAXIS

A1.ORIGIN	Origine misurata dell'allineamento A1
A1.XAXIS	Asse X misurato dell'allineamento A1
A1.YAXIS	Asse Y misurato dell'allineamento A1
A1.ZAXIS	Asse Z misurato dell'allineamento A1

A1.CORIGIN	Origine teorica dell'allineamento A1
A1.CXAXIS	Asse X teorico dell'allineamento A1
A1.CYAXIS	Asse Y teorico dell'allineamento A1
A1.CZAXIS	Asse Z teorico dell'allineamento A1

## Riferimenti di tipo stringa

I riferimenti ai commenti sono gli unici oggetti tipo stringa. Consentono di fare riferimento solo ai commenti di tipo INPUT oppure SÌ/NO. Per identificare questo tipo di commenti è possibile usare il relativo ID.

Formato: <ID commento>.INPUT -> [C1.INPUT](#)

C1.INPUT - Il valore di input (dall'operatore) per il commento C1

I tipi di commento SÌ/NO consentono di impostare l'input sulla stringa affermativa o negativa appropriata, in base al linguaggio corrente di PC-DMIS. Nella versione italiana di PC-DMIS, se l'operatore seleziona il pulsante che indica affermazione la stringa viene impostata su "SÌ". Se l'operatore seleziona il pulsante di negazione, la stringa viene impostata su "NO". Quando si confrontano le stringhe per la verifica della selezione dei commenti "SÌ" o "NO", il software distingue tra lettere maiuscole e minuscole. Pertanto, il confronto con le parole "sì" o "no" fallisce se l'input del commento SÌ/NO è impostato su "Sì" o "NO".

## Variabili

Le variabili possono coincidere con uno dei sette tipi di operandi: numero intero, numero reale, stringa, punto, puntatore elemento, array e funzione. Le variabili vengono create mediante l'istruzione [ASSEGNA](#), che assegna anche loro un valore e un tipo.

L'ID della variabile può essere costituito da una qualsiasi stringa alfanumerica che non inizi con un carattere numerico. Nell'ID di una variabile è possibile usare caratteri sottolineati, tranne che nel primo carattere.



Fintanto che la routine di misurazione rimane aperta, PC-DMIS salva i valori delle variabili tra un'esecuzione e l'altra. Questo significa che quando l'esecuzione termina, PC-DMIS usa i valori come sono alla fine dell'esecuzione della routine quando la si riavvia. Questo comportamento potrebbe non essere sempre desiderabile. Se si desiderano valori freschi delle variabili, cancellare i valori con istruzioni di assegnazione all'inizio della routine. Per esempio se per una variabile si usa il valore V1 in alcuni calcoli, si può usare `ASSEGNA/V1=0` per cancellare il valore precedente della variabile.



Se la finestra di modifica è aperta, PC-DMIS indica il valore assunto dalla variabile ogni volta che il cursore viene posizionato sull'apposito campo. I valori delle variabili vengono modificati durante il flusso dell'esecuzione. Per ottenere il valore assunto al momento da una determinata variabile, posizionare il puntatore del mouse su di essa.

```
ASSEGNA/ V1=2.2+2
```

La variabile V1 è un numero reale il cui valore è 4,2

```
ASSEGNA/ VAR1=CIRCLE1.X
```

La variabile VAR1 è un numero reale con un valore uguale al valore misurato di CIRCLE1.X al momento dell'assegnazione.

```
ASSEGNA/ MYVAR=LINE1.XYZ
```

La variabile MYVAR è un punto con lo stesso valore del centro misurato di LINE1 al momento dell'assegnazione.

```
ASSEGNA/SVAR="Ciao"
```

La variabile SVAR è costituita da una stringa il cui valore è "Ciao".

In questi esempi alle variabili vengono assegnati dei valori. Una volta assegnato un valore a una variabile, è possibile usare le variabili come un operando in un qualsiasi campo delle espressioni.

Qui V1 è usato in un campo numerico. È usata come valore del comando di avvicinamento:

```
ASSEGNA/V1=1/3AVVICINAMENTO/V1
```



Poiché è possibile usare le espressioni in molti campi modificabili, anche la seguente espressione è valida ed ha lo stesso effetto: AVVICINAMENTO / 1/3.

È possibile fare riferimento a ciascuna componente delle variabili di tipo punto mediante l'uso della rispettiva notazione dell'estensione utilizzata per i riferimenti.

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

V1 è una variabile di tipo punto il cui valore è 3, 4, 5

```
ASSEGNA/ XVAR=V1.X
```

XVAR è una variabile di tipo double con valore 3

```
ASSEGNA/YVAR=V1.Y
```

YVAR è una variabile di tipo double con valore 4

```
ASSEGNA/IVAR=V1.I
```

IVAR è di tipo double con valore 3

```
ASSEGNA/REDUNVAR=V1.XYZ
```

REDUNVAR è una variabile di tipo punto con il valore 3, 4, 5

Le estensioni seguenti si equivalgono. Vengono fornite entrambe per chiarire il significato di un'espressione in una routine di misurazione.

V1 è una variabile di tipo punto.

V1.X è uguale a V1.I

V1.Y è uguale a V1.J

V1.Z è uguale a V1.K

V1.XYZ è uguale a V1.IJK e V1 senza alcuna estensione.

Se il valore di una variabile tipo stringa è uguale all'ID di un elemento, di una dimensione o di un allineamento, è possibile usare tale variabile come oggetto di riferimento:

```
ASSEGNA/V1="CERCHIO1"
```

È possibile utilizzare i seguenti operandi, i quali vengono considerati validi solo se esiste un elemento denominato CERCHIO1.



## Uso di espressioni e di variabili

V1.X - Il valore X misurato di CIRCLE1  
V1.TX - Il valore X teorico di CIRCLE1  
V1.Diametro - Il diametro misurato di CIRCLE1  
V1.Raggio - Il raggio misurato di CIRCLE1

Questo tipo di indirizzamento indiretto delle variabili stringa è disponibile solo su un livello. Di seguito viene riportato un esempio non valido:



```
ASSEGNA/V1="CIRCLE1"  
ASSEGNA/V2="V1"
```

v2.X - Viene valutata 0 invece del valore misurato di CIRCLE1.X.



Il riferimento V2.X *non* viene visualizzato in rosso come errore, anche se un'espressione che lo precede lo imposta come stringa. Il motivo per cui non viene segnalato come errore è che il flusso di esecuzione della routine di misurazione non è noto fino al momento dell'esecuzione.

Tuttavia, se si utilizzano le parentesi graffe, i comandi validi sono i seguenti:



```
ASSEGNA/V1=CIRCLE1" ASSEGNA/V2="V1"
```

V2.X - Dà il valore di CIRCLE1.X.

Si prenda in considerazione il seguente esempio:



```
ASSEGNA/V1="CIRCLE1" ASSEGNA/V2="V1"  
IF/CIRCLE1.X>CIRCLE1.TX,GOTO,L2 L1=ETICHETTA/  
ASSEGNA/V3=V2.X  
GOTO/ETICHETTA,L3 L2=ETICHETTA /  
ASSEGNA/V2=MPOINT(2,5,7)  
GOTO/ETICHETTA,L1 L3=ETICHETTA/
```

Se durante l'esecuzione della routine il valore di CIRCLE1.X è maggiore del valore di CIRCLE1.TX, l'espressione V2.X sarà valida e restituirà 2. Altrimenti, l'espressione V2.X restituirà 0 poiché il valore di V2 al momento di ASSEGNA per V3 è la stringa "V1". In questi casi, il programmatore del pezzo ha il compito di accertarsi che le espressioni si comportino come ci si aspetta.



È possibile usare quasi tutti i riferimenti agli elementi a sinistra dell'istruzione di *assegnazione* per immettere un valore nei dati misurati o teorici di un elemento. L'unica eccezione è rappresentata dai singoli componenti I, J, K dei vettori. In tal caso, è necessario assegnare l'intero vettore in una sola volta, mediante un'espressione valutata che lo valuta come punto. Quando vengono inseriti nei dati dell'elemento, i dati del vettore vengono normalizzati.



```
ASSEGNA/CIRCLE1.I=2-non consentito
ASSEGNA/CIRCLE1.IJK=PUNTOM(2,0,0)-consentito(il vettore
è normalizzato su 1,0,0)
```

Per informazioni sull'uso di variabili nelle dimensioni, vedere l'argomento "Dimensionamento delle variabili" nel capitolo "Uso delle dimensioni Legacy".

## Strutture

È possibile usare un tipo di variabili chiamate *strutture* che consentono di attribuire estensioni ad una variabile per identificarne un elemento secondario. Ciò si può vedere in questo frammento di codice:



```
ASSEGNA/V1.ALTEZZA=6
ASSEGNA/V1.LARGHEZZA=4.3
ASSEGNA/V1.MODALITÀ="CIRCOLARE"
ASSEGNA/V1.PUNTO
= PUNTOM(100.3,37.5,63.1)
```

Dove:

- `V1` è la struttura.
- `HEIGHT`, `WIDTH`, `MODE` e `POINT` sono i sotto elementi della struttura

## Regole per le strutture

- Come le variabili, le strutture non devono essere necessariamente dichiarate.
- Gli elementi secondari di una struttura possono essere costituiti da uno dei seguenti tipi di variabile:
  - Numero intero

## Uso di espressioni e di variabili

- Doppia precisione
- Punto
- Puntatore all'elemento
- Funzione
- Array
- Struttura

Ad esempio, gli elementi struttura possono essere array e gli elementi array possono essere strutture. Si possono vedere le espressioni valide in questo frammento di codice:

```
ASSEGNA/AUTO.PORTA.SINISTRA[2].QUADRANTE[3].GIUNTO[5].HIT[4]
=MPUNTO(558.89,910.12,42.45)

COMMENTO/OPER,"Posizione Z attuale:
"+AUTO.PORTA.SINISTRA[2].QUADRANTE[3].GIUNTO[5].HIT[4].Z

ASSEGNA/GIUNTOATT.PORTA.SINISTRA[2].QUADRANTE[3].GIUNTO[5]

COMMENTO/OPER,"Prossimo punto: "+GIUNTOATT.HIT[4]
```

## Strutture con variabili di tipo Punto

Se una variabile è di tipo Punto, l'utente può ancora usare le estensioni .X, .Y, .Z, .I, .J e .K per arrivare ai singoli elementi del punto. È possibile anche usare tutte le estensioni di questo esempio nelle loro strutture senza necessariamente doverle usarle come elementi Punto come mostrato qui:



```
ASSEGNA/V1.X="Una
stringa" ASSEGNA/V1.Y=ARRAY(1,3,5,9,7)
ASSEGNA/V1.Z=PUNTOM(3,5,7)
```

COMMENT/REPT,V1.X	L'output è "Una stringa"
COMMENT/REPT,V1.Y[2]	L'output è 3, il secondo elemento dell'array.

COMMENT/REPT,V1.Z.Y	L'output è 5, il valore Y di MPOINT.
---------------------	--------------------------------------

Combinando le strutture alle funzionalità del linguaggio delle espressioni PC-DMIS, è possibile ottenere riferimenti di struttura dinamica, come mostrato di seguito:



```

ASSEGNA/DYNAMICSTRUCT=FUNZIONE ( (X,Y) ,X.Y)
C1=COMMENTO/INPUT,Immettere nella voce
ASSEGNA/TESTSTR=C1.INPUT

ASSEGNA/ANTERIORE=SINISTRO (TESTSTR, INDICE (TESTSTR, "."
) -1)
ASSEGNA/POSTERIORE=MID (TESTSTR, INDICE (TESTSTR, "." )

ASSEGNA/RISULTATO=STRUTDINAMICA (ANTERIORE, POSTERIORE)

```

Questa parte dell'esempio richiede di fornire un riferimento alla variabile, divide il riferimento al primo '.' e quindi assegna **RESULT** in modo che sia uguale al riferimento utilizzando la funzione **DYNAMICSTRUCT**.

Pertanto se è stato immesso **V1.Y[4]** per la variabile **C1.INPUT**, il valore di **RESULT** terminerà con 9 (il quarto elemento dell'array assegnato a **V1.Y**).

La valutazione del tempo di memorizzazione delle espressioni è stata migliorata per mostrare in modo preciso tutti gli elementi di una struttura o di un array.

## Puntatori

*I puntatori vengono anche definiti "Puntatori degli elementi". Per ulteriori informazioni, vedere nel glossario la voce "Puntatori degli elementi".*

I puntatori rappresentano un modo semplice per fare riferimento a un elemento mediante una variabile o per trasferire gli oggetti mediante un comando di richiamo di una subroutine. I puntatori sono simili all'indirizzamento indiretto mediante i nomi delle stringhe. Tuttavia, hanno il vantaggio di poter essere usati con le subroutine. A differenza delle stringhe, i puntatori trasferiti come argomenti di una subroutine consentono di modificare direttamente l'oggetto indicato dalla subroutine stessa. I puntatori non vengono usati nelle espressioni complesse. Se sono usati nelle espressioni complesse, i puntatori vengono valutati zero.

Si prendano in considerazione i seguenti esempi.

### Esempio dell'uso del puntatore:

In questo esempio, `V1` è definito come un puntatore che punta a `CIR1`:

```
ASSEGNA/V1={CIR1}
```

In questo esempio, a `DIST` viene assegnato il valore della distanza tra `CIR1` e l'origine:

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

È possibile anche inserire un'espressione tra parentesi graffe per ottenere un puntatore all'elemento. Attualmente gli esempi riportati di seguito sono tutti metodi validi per ottenere il puntatore all'elemento `CER1`:

```
ASSIGN/FEATCOUNT = 1 (ASSEGNA/CONT_ELEM = 1)
```

```
ASSIGN/V1={"CIR" + FEATCOUNT} (ASSEGNA/V1={"CER" +  
CONT_ELEM})
```

Assegna l'espressione `"CER1"` a `V1`.

```
ASSEGNA/V2="CER1"
```

```
ASSEGNA/V3={V2}
```

Assegna l'espressione `"CER1"` della variabile `V2` alla variabile `V3`.

```
C1=COMMENTO/INPUT, Immettere il nome di un elemento.
```

```
ASSIGN/V4={C1.INPUT} (ASSEGNA/V4={C1.INPUT})
```

Inserisce il nome dell'elemento `C1.INPUT` e lo posiziona nella variabile, `V4`.

### Esempio di subroutine:

Nella routine chiamante:

```
CS1 = CALLSUB/SUB.PRG,CHANGEX, {CER1}
```

Nella subroutine:

```
GEN1=GENERICO/ELEMENTO
```

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

(Durante l'esecuzione, quando il codice passa CER1 alla routine, CER1 prende il posto di GEN1)

```
ARG1.X=5
```

(Imposta il valore X misurato di CER1 su 5)

```
END/SUBROUTINE
```

### Esempio di espressione complessa:

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

{CER1} viene valutato come zero e l'intera espressione viene valutata come 2.

## Array

Sono disponibili tre tipi di array: array di elementi, di punti e di variabili.



Anche se gli array multidimensionali sono visualizzati come tali nel software, in realtà è possibile utilizzarli soltanto come unidimensionali, a meno che non si anteponga agli array un comando INDICI ARRAY (vedere l'argomento "Oggetto Indici array").

## Array di elementi

Quando si misura un elemento più di una volta durante l'esecuzione di una routine di misurazione a causa di un qualche tipo di ciclo, il software crea automaticamente un array di elementi. Il numero di elementi presenti nell'array è uguale al numero di esecuzioni dell'elemento.



Se un oggetto Cerchio misurato si trova in un ciclo eseguito cinque volte, viene creato un array costituito da cinque cerchi misurati. Se l'ID del cerchio misurato è "CER1", è possibile usare l'espressione dell'array per accedere alle singole istanze del cerchio misurato. Per indicare l'istanza desiderata, usare parentesi quadre, come qui:

```
ASSEGNA/V1=CER1 [3] .X
```

A V1 viene assegnato il valore X misurato della terza istanza di CER1.



Quando esiste un array di istanze di un determinato elemento ma non viene utilizzata la notazione di array in un riferimento a quell'elemento, viene utilizzata l'istanza più recente. Nell'esempio precedente, il riferimento `CER1 .X` sarebbe uguale a `CER1 [5] .X`, in quanto la quinta istanza rappresenta l'istanza più recente dell'oggetto.

All'interno delle parentesi quadre dell'espressione di un array si possono usare espressioni.

`CER1 [3] .X` e `CER1 [2+1] .X` sarebbero pertanto equivalenti.

Il prossimo esempio usa due blocchi di comandi di ciclo While/End While. Il primo blocco esegue CER1 cinque volte. Il secondo blocco usa la variabile V1 all'interno di parentesi quadre come `CER1 [V1] .XYZ`, per inviare alla finestra del rapporto il baricentro misurato in ciascuna delle cinque esecuzioni:



```


        ASSIGN/V1=1
        WHILE/V1<6
CIR1      =FEAT/CIRCLE,CARTESIAN,IN,LEAST_SQR
          THEO/<40,30,-4.824>,<0,0,1>,30
          ACTL/<40.002,29.991,-4.836>,<0,0,1>,29.982
          MEAS/CIRCLE,4,ZPLUS
          HIT/BASIC,NORMAL,<41.984,44.868,-
2.885>,<-0.132272,-0.9912135,0>,<41.972,44.85,-2.891>,USE
THEO=YES
          HIT/BASIC,NORMAL,<51.721,39.36,-5.094>,<-
0.781412,-0.6240155,0>,<51.706,39.375,-5.107>,USE
THEO=YES
          HIT/BASIC,NORMAL,<54.792,32.491,-5.44>,<-
0.9861119,-0.1660821,0>,<54.775,32.474,-5.453>,USE
THEO=YES
          HIT/BASIC,NORMAL,<52.526,21.748,-
5.879>,<-0.8350841,0.5501223,0>,<52.537,21.764,-
5.893>,USE THEO=YES
          ENDMEAS/
          ASSIGN/V1=V1+1
        END_WHILE/
        ASSIGN/V1=1
        WHILE/V1<6
          COMMENT/REPT,
            "Centroid of CIR1, instance #" + V1
          CIR1[V1].XYZ
          COMMENT/REPT,
            -----
          ASSIGN/V1=V1+1
        END_WHILE/

```





Ecco l'output generato nella finestra di modifica:

	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>  
-----

Gli array sono disponibili anche per le dimensioni e gli allineamenti eseguiti più volte durante una stessa esecuzione del programma. Pertanto, `Dim1[2].Nom` e `Allin1[4].Origin` sarebbero disponibili se la dimensione "Dim1" e l'allineamento "Allin1" sono stati eseguiti rispettivamente almeno due e quattro volte.

Se il riferimento all'array di elementi non rientra nei limiti, (per esempio se l'utente chiede `CER1[2.5]` o `CER1["Ciao"]`) viene restituito l'elemento con il limite maggiore o minore. Se `CER1` aveva 3 istanze, l'espressione `CER1[4]` o maggiore restituirebbe il valore di `CER1[3]` e `CER1[0]` o minore restituirebbe il valore di `CER1[1]`. Poiché viene forzata la conversione di tutte le espressioni comprese tra parentesi in numeri interi, 2,5 diventerebbe 2 e "Ciao" diventerebbe 0.

## Oggetto Indici array

In base all'impostazione predefinita, gli array di elementi sono sempre unidimensionali. Se è più comodo trattare un array di elementi come array multidimensionale, è possibile farlo con un oggetto Indici array.

L'oggetto Indici array consente di specificare il limite superiore ed il limite inferiore per le dimensioni di più array.

- Quando si impostano i limiti superiore ed inferiore della prima dimensione, PC-DMIS crea un array bidimensionale, nel quale la prima dimensione è vincolata e la seconda non lo è.

- Quando si impostano i limiti superiore e inferiore delle prime due dimensioni di un array, PC-DMIS crea un array tridimensionale. L'ultima dimensione è sempre non vincolata.



Si supponga che l'elemento F1 si trovi in un ciclo WHILE nidificato. Il ciclo WHILE interno viene eseguito cinque volte, mentre il ciclo WHILE esterno viene eseguito tre volte. Al termine dell'esecuzione, l'elemento F1 è stato eseguito 15 volte; pertanto, esistono 15 istanze di F1.

Si consideri il seguente esempio di segmento di una routine di misurazione:

```
INDICI_ARRAY/1..5,..
```

```
ASSEGNA/V1 = 1
```

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

```
ASSEGNA/V2=1
```

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

```
  F1 =ELEM/PUNTO,RETT
```

```
  TEOR/V2,V1,0,0,0,1
```

```
  REALE/1,1,0,0,0,1
```

```
  MIS/PUNTO,1
```

```
  PUNTO/BASE,V2,V1,0,0,0,1,1,1,0
```

```
  FINE_MIS/
```

```
  ASSEGNA/V2=V2+1
```

```
  COMMENTO/REP,"Location of F1["+V2+", "+V1+"]  
  : "+F1[V2,V1].XYZ
```

```
END_WHILE/
```

```
ASSEGNA/V1=V1+1
```

```
END_WHILE/
```

Questa sezione di codice crea una griglia 3 x 5 costituita da 15 punti misurati.

Il comando Indici array ha limitato la prima dimensione dell'array di elementi da 1 a 5 inclusi. Quindi, nel rapporto di ispezione, anziché di apparire come F1[1] – F1[15], gli oggetti appariranno come F1[1, 1] – F1[5, 3], in maniera più coerente con la disposizione degli elementi. Si noti che il commento fa riferimento anche all'array di elementi che utilizza una sintassi di array bi-dimensionale.

Per inserire un oggetto Indici array in una routine di misurazione, procedere come segue.

1. Utilizzando la tastiera, digitare "**Array**" in una riga vuota della finestra di modifica.
2. Premere il tasto di tabulazione sulla tastiera.



Se si deseleziona la casella di opzione **Visualizza parentesi per array di elementi**, il nome dell'elemento non viene visualizzato tra parentesi. Vedere la descrizione "Visualizza parentesi per array di elementi" nell'argomento "Opzioni di Impostazione: scheda Impostazione ID" del capitolo "Impostazione delle preferenze".

## Array di punti

I punti di un determinato elemento sono disponibili sotto forma di array, ed è possibile accedervi mediante espressioni che utilizzano la sintassi di array <ID elemento>.Hit[<espressione array>].<estensione> o <ID elemento>.RawHit[<espressione array>].<estensione>. Durante l'esecuzione della compensazione del tastatore Hit restituisce i dati compensati del tastatore. RawHit restituisce sempre dati non compensati. Estensioni valide sono X, Y, Z, I, J, K, TX, TY, TZ, TI, TJ, TK, XYZ, TXYZ, IJK e TIJK

```
Circle1.Hit[1].XYZ
```

Il centro misurato (compensato dal tastatore) del punto 1 di "Circle1".

```
Circle1.Hit[2].IJK
```

Il vettore misurato del punto 2 di "Circle1"

I dati dei punti sono disponibili per tutti gli oggetti contenenti punti, indipendentemente dal fatto che i punti effettivi vengano o meno visualizzati nella finestra di modifica. Pertanto, è possibile ottenere punti per le scansioni e gli elementi automatici.

### Uso di array di punti per definire input per elementi costruiti

Si possono usare array di punti per definire input per gli elementi costruiti.

## Uso di espressioni e di variabili

Per gli elementi costruiti, quando si usano i metodi `feature.HIT[start..end]` o `feature.HITS[start..end]`, le proprietà iniziali e finali sono facoltative:

- se il valore iniziale non è definito, PC-DMIS presume che sia pari a "1";
- se il valore finale non è definito, PC-DMIS presume che sia pari al numero totale di punti dell'elemento.

È lo come se si fosse immesso il valore di `"feature.NUMHITS"`.



Questi esempi mostrano tre modi differenti di usare gli stessi punti per il cerchio CIR1.

### Esempio 1



In questo esempio i valori iniziale e finale sono definiti esplicitamente:



```
CIR3      =FEAT/CIRCLE,CARTESIAN,IN,NO
          THEO/<20.97629,22.90352,0>,<0,0,-
1>,20.97629
          ACTL/<20.96578,22.89023,-0.01243>,<0,0,-
1>,20.96578
          CONSTR/CIRCLE,REV,CIR1.HIT[1..CIR1.NUMHITS]
```

**Esempio 2**

In questo esempio è definito solo il valore iniziale "1". Poiché il valore finale non è definito, PC-DMIS definisce come valore finale "CIR1.NUMHITS":



```
CIR3      =FEAT/CIRCLE,CARTESIAN,IN,NO
           THEO/<20.97629,22.90352,0>,<0,0,-
1>,20.97629
           ACTL/<20.96578,22.89023,-0.01243>,<0,0,-
1>,20.96578
           CONSTR/CIRCLE,REV,CIR1.HIT[1..]
```

**Esempio 3**

In questo esempio, non sono definiti né il valore iniziale né il valore finale. Perciò, PC-DMIS definisce "1" come valore iniziale e "CIR1.NUMHITS" come valore finale:



```
CIR3      =FEAT/CIRCLE,CARTESIAN,IN,NO
           THEO/<20.97629,22.90352,0>,<0,0,-
1>,20.97629
           ACTL/<20.96578,22.89023,-0.01243>,<0,0,-
1>,20.96578
           CONSTR/CIRCLE,REV,CIR1.HIT[. .]
```

Le sezioni che seguono descrivono alcune funzioni supplementari che sono utili per trovare il punti minimo e massimo di una scansione.

**Assegnare un intervallo di punti a un array**

Un intervallo di punti può essere assegnato ad un array usando la sintassi seguente:

<ID elemento>.<Tipo di Punto>[<Num Iniziale>...<Num. Finale>].<Estensione>

dove

<ID Elemento> è il nome dell'elemento

<Hitttype> può essere la parola "HIT" per i dati compensati o "RAWHIT" per i dati non compensati. Se la compensazione tastatore è disabilitata, il valore restituito non è mai compensato.

<Num Iniziale> è un'espressione che identifica il valore del primo indice dell'intervallo di punti

<Num finale> è un'espressione che identifica il valore del secondo indice dell'intervallo di punti

<Estensione> identifica il tipo dei dati. Le possibili estensioni includono i tipi di dati teorici o misurati elencati nella sezione successiva.



### **Esempio di come usare in un'espressione i valori impliciti iniziale e finale.**

Si supponga di voler trovare tutti i punti massimi che definiscono un piano (PLN1). Questo si può scrivere nel modo seguente:

```
PLN1.HIT[1..PLN1.NUMHITS]
```

Un modo più breve per farlo è quello di usare i valori iniziale e finale dedotti per i punti. Lo stesso codice si può scrivere nel modo seguente:

```
PLN1.HIT[...]
```

In questo caso, poiché non abbiamo definito i valori iniziale e finale dei punti, PC-DMIS presume che il valore iniziale sia 1 e il valore finale sia NUMHITS.

## **Misurata**

- X – I valori X misurati dei punti
- Y – I valori Y misurati dei punti
- Z – I valori Z misurati dei punti
- XYZ – I valori XYZ misurati dei punti
- I – I valori I misurati dei punti
- J – I valori J misurati dei punti
- K – I valori K misurati dei punti
- IJK – I valori IJK misurati dei punti

## **Teorici**

- TX – I valori teorici X dei punti

- TY – I valori teorici Y dei punti
- TZ – I valori teorici Z dei punti
- TXYZ – I valori teorici XYZ dei punti
- TI – I valori teorici I dei punti
- TJ – I valori teorici J dei punti
- TK – I valori teorici K dei punti
- TIJK – I valori teorici IJK dei punti

Per esempio:

```
ASSEGNA/V1 = SCAN1.HIT[1..10].X
```

V1 è assegnato a un array di 10 valori che sono i valori di X misurati dai primi 10 punti di SCAN1.

```
ASSEGNA/V2 = SCAN1.HIT[1..SCAN1.NUMHITS].XYZ
```

V2 è assegnato a un array di punti da ognuno dei baricentri dei punti nella scansione.

## Ordinamento degli array

PC-DMIS consente di ordinare gli array in modo crescente o decrescente. Le due espressioni seguenti accettano un array e restituiscono un array ordinato:

Per ordinare in ordine *ascendente* utilizzare:

```
SORTUP(<array>)
```

Per ordinare in ordine *discendente* utilizzare:

```
SORTDOWN(<array>)
```

Per esempio:

```
ASSEGNA/V1 = ARRAY(5,8,3,9,2,6,1,7)
```

V1 è assegnato all'array "5,8,3,9,2,6,1,7"

```
ASSEGNA/V2=SORTUP(V1)
```

V2 conterrà i valori dell'array in ordine crescente: "1,2,3,5,6,7,8,9"

```
ASSEGNA/V3=SORTDOWN(V1)
```

V3 conterrà i valori dell'array in ordine decrescente: "9,8,7,6,5,3,2,1"

## Come restituire il valore maggiore o minore degli indici di un array.

È possibile inserire un array in una funzione ed ottenerne il numero di indice dell'elemento con il valore massimo o minimo utilizzando le funzioni seguenti:



## Uso di espressioni e di variabili

Per ottenere l'indice dell'elemento con il valore *massimo*, utilizzare:

```
MAXINDEX(<array>)
```

Per ottenere l'indice dell'elemento con il valore *minimo*, utilizzare:

```
MININDEX(<array>)
```

Per esempio:

```
ASSEGNA/V1 = ARRAY(5,8,3,9,2,6,1,7)
```

V1 è assegnato all'array "5,8,3,9,2,6,1,7"

```
ASSEGNA/V2=MAXINDEX(V1)
```

V2 conterrà il valore dell'indice dell'array 4. Il valore effettivo degli elementi di quell'array è 9.

```
ASSEGNA/V3=MININDEX(V1)
```

V3 conterrà il valore dell'indice dell'array 7. Il valore effettivo degli elementi di quell'array è 1.

È possibile utilizzare i valori di indice restituiti per ottenere il valore effettivo dell'elemento dell'array.

### **Come ottenere i valori di indice dato di un array, dopo l'ordinamento.**

È possibile inserire un array in una funzione, ordinare i valori dell'array in modo crescente o decrescente e poi restituire i valori dell'indice utilizzando le seguenti funzioni:

Per restituire le posizioni di indice dell'array in base ai relativi valori ordinati dal maggiore al minore, utilizzare:

```
MAXINDICES(<array>)
```

Per restituire le posizioni di indice dell'array in base ai relativi valori ordinati dal maggiore al minore, utilizzare:

```
MININDICES(<array>)
```

Per esempio:

```
ASSEGNA/V1=ARRAY(4,8,2,9,5,7)
```

V1 è assegnato all'array "4,8,2,9,5,7"

```
ASSEGNA/V2=MAXINDICES(V1)
```

V2 conterrà un array con i seguenti valori: "4,2,6,5,1,3"

```
ASSEGNA/V3=MININDICES (V1)
```

V3 conterrà un array con i seguenti valori: "3,1,5,6,2,4"

### Esempio d'uso di funzioni su array, per trovare il Massimo e Minimo Punto di una Scansione

Lo scopo principale delle funzioni viste sopra, che agiscono su array di punti, è di fornire in modo semplice il minimo ed il massimo punto di una scansione.

Per dimensionare il punto da SCAN1 che ha il valore X massimo misurato, è possibile utilizzare la seguente espressione:



```
ASSEGNA/MIND_MAX_PUNTI=MAXINDEX(SCAN1.HIT[1..SCAN1.NUMHITS].X)
D1 = POSIZIONE DELL'ELEMENTO SCAN1.HIT[IND_MAX_PUNTI]
```

Per individuare i tre punti più elevati nell'asse Z di SCAN2, utilizzare la seguente espressione:



```
ASSEGNA/MI=MAXINDICES(SCAN2.HIT[1..SCAN2.NUMHITS].Z)
ASSEGNA/TREPUNTI=ARRAY(SCAN2.HIT[MI[1]].XYZ, SCAN2.HIT[MI[2]].XYZ, SCAN2.HIT[MI[3]].XYZ)
```

## Array di variabili

Gli array delle variabili non devono essere dichiarati. Gli array di variabili vengono creati quando l'espressione a destra dell'istruzione di assegnazione viene valutata come un array o quando la parte sinistra dell'istruzione fa riferimento a un elemento presente in un array di variabili.

```
Assegna/V1=Array(3, 4, 5, 6, 7)
```

Crea l'array di 5 elementi e lo assegna a V1

```
Assegna/V2=V1[3]
```

Assegna a V2 il valore del terzo elemento nell'array di V1: 5

```
ASSEGNA/V1[4] = 23
```

Imposta il 4 elemento dell'array V1 su 23

Gli array vengono creati e allocati in modo dinamico. Pertanto, è possibile creare un array specificandone il riferimento a sinistra di un'istruzione di assegnazione.

## Uso di espressioni e di variabili

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

Crea l'array in modo dinamico, con il 5 elemento impostato su 8

Quando si fa riferimento a un elemento dell'array a cui non è mai stato assegnato un valore, l'espressione dell'array viene valutata come 0.

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

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

V4 è impostata a 8.

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

Se il sesto elemento di V3 non è mai stato impostato, V5 viene impostata su 0.

Come negli altri tipi di array, è possibile utilizzare le espressioni all'interno delle parentesi quadre.

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

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

V4 è impostata a 8.

Gli array di variabili possono essere costituiti da più dimensioni.

```
ASSEGNA/V6=Array(Array(4,7,2),Array(9,2,6))
```

V6 viene impostata su un array di dimensioni 2 per 3, dove V6[1, 1] uguale a 4, V6[1, 2] uguale a 7, V6[1, 3] uguale a 2, V6[2, 1] uguale a 9, V6[2,2] uguale a 2 e V6[2,3] uguale a 6.

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

V7 è impostata a 9.

Gli array di variabili possono essere costituiti da indici negativi:

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

Il -il 3° indice dell'array V8 è impostato su 5.

L'assegnazione dell'array sovrascrive i valori precedenti:

```
ASSEGNA/V8="Ciao"
```

La variabile V8 equivale alla stringa "Ciao".

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

V8 non è più una stringa, ma un array, nel quale il valore del secondo elemento è pari a 5.

```
ASSEGNA/V8=9
```

V8 non è più un array, ma un numero intero il cui valore è 9.

Gli array possono essere costituiti da più tipi di elementi:

```
Assegna/V9=Array("Ciao", 3, 2.9, {ELEM1})
```

Crea un array V9 con 4 elementi. Il primo elemento è una stringa, il secondo un numero intero, il terzo un numero reale e il quarto è un puntatore a ELEM1.

È possibile aumentare le dimensioni degli array per includervi più elementi:



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

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

La prima istruzione crea un array iniziale V10 con 3 elementi (3,1 e 5). La seconda quindi aumenta V10 di un elemento e dà il valore 7 all'elemento finale.

## Operatori delle espressioni

Di seguito vengono elencati gli operatori di base disponibili in PC-DMIS:

**+** Somma: *<Espressione> + <Espressione>*

Somma le due espressioni. Nel caso delle stringhe, queste vengono concatenate.

**-** Sottrazione: *<Espressione> - <Espressione>*

Sottrae la seconda espressione dalla prima.

**\*** Moltiplicazione: *<Espressione> \* <Espressione>*

Moltiplica le due espressioni.

**/** Divisione: *<Espressione> / <Espressione>*

Divide la prima espressione per la seconda.

**^** Elevazione a potenza: *<Espressione> ^ <Espressione>*

Eleva la prima espressione alla potenza della seconda espressione.

Uso di espressioni e di variabili

% Modulo: *<Espressione> % <Espressione>*

Restituisce il resto della divisione di un'espressione per l'altra espressione.

- Elemento opposto *-<Espressione>*

Restituisce l'elemento opposto dell'espressione.

! Not logico: *!<Espressione>*

Restituisce il NOT logico dell'espressione.

== Uguale a: *<Espressione> == <Espressione>*

Viene valutato come 1 se le espressioni sono uguali. Altrimenti viene valutato come 0. (Due segni di uguale (==) sono usati per distinguere questo operatore dall'operatore di assegnazione = nell'istruzione di assegnazione).

<> Non uguale a: *<Espressione> <> <Espressione>*

Viene valutato come 1 se le espressioni non sono uguali. Altrimenti viene valutato come 0.

> Maggiore di: *<Espressione> > <Espressione>*

Viene valutato come 1 se la prima espressione è maggiore della seconda. Altrimenti viene valutato come 0.

>= Maggiore di o uguale a: *<Espressione> >= <Espressione>*

Viene valutato come 1 se la prima espressione è maggiore della seconda o uguale ad essa. Altrimenti viene valutato come 0.

< Inferiore a: *<Espressione> < <Espressione>*

Viene valutato come 1 se la prima espressione è inferiore alla seconda. Altrimenti viene valutato come 0.

<= Inferiore a o uguale a: *<Espressione> <= <Espressione>*

Viene valutato come 1 se la prima espressione è inferiore alla seconda o uguale ad essa. Altrimenti viene valutato come 0.

E Logico E: *<Espressione> E <Espressione>*

Viene valutato come 1 se entrambe le espressioni sono valutate diverse da 0. Altrimenti viene valutato come 0.

**O** Logico O: *<Espressione> O <Espressione>*

Viene valutato come 1 se una delle due espressioni è valutata diversa da 0. Altrimenti viene valutato come 0.

**()** Parentesi: *( <Espressione> )*

Assegna la priorità di valutazione all'espressione presente all'interno delle parentesi.

## Priorità

Le espressioni vengono valutate in base alla priorità indicata di seguito (partendo dalla precedenza maggiore alla priorità minore):

Priorità maggiore

- Operandi
- (menu unario), !, (), funzioni (come ABS, COS, STR, LEN, CROSS, e così via.)
- ^
- \*, /, %
- +, -
- ==, <>, <, <=, >, >=
- AND
- OR

Priorità minore

## Funzioni

Le funzioni sono espressioni specifiche di PC-DMIS o espressioni definite dall'utente che accettano parametri e restituiscono i risultati. I parametri vengono sostituiti nell'espressione prima che tale espressione venga valutata.

## Elenco delle funzioni

Nel seguente elenco sono contenute in ordine alfabetico tutte le funzioni disponibili per il linguaggio delle espressioni di PC-DMIS.

- ABS (matematica)

## Uso di espressioni e di variabili

- ACOS (matematica)
- ANGLEBETWEEN (punto)
- ARCSEGMENTENDINDEX (varie)
- ARCSEGMENTSTARTINDEX (varie)
- ARRAY (array)
- ASIN (matematica)
- ATAN (matematica)
- CHR (stringa)
- CONCAT (stringa)
- COS (matematica)
- CROSS (punto)
- DEG2RAD (matematica)
- DELTA (punto)
- DIST2D (puntatore)
- DIST3D (puntatore)
- DOT (punto)
- ELEMENT (stringa)
- EOF (varie)
- EOL (varie)
- EQUAL (array)
- EQUAL (stringa)
- EXP (matematica)
- FORMAT (stringa)
- FUNCTION (funzione)
- GETCOMMAND (puntatore)
- GETPROGRAMINFO (stringa)
- GETROTABDATA (varie)
- GETSETTING (stringa)
- GETTEXT (stringa)
- GETTRACEVALUE (stringa)
- IF (varie)
- INDEX (stringa)
- ISIOCHANNELSET (varie)
- LEFT (stringa)
- LEN (array)
- LEN (puntatore)
- LEN (stringa)
- LINESEGMENTENDINDEX (varie)
- LINESEGMENTSTARTINDEX (varie)
- LN (matematica)

- LOG (matematica)
- LOWERCASE (stringa)
- MAX (array)
- MID (stringa)
- MIN (array)
- MPOINT (punto)
- ORD (stringa)
- PCDMISAPPLICATIONPATH (stringa)
- PCDMISUSERHIDDENATAPATH (stringa)
- PCDMISUSERVISIBLEDATAPATH (stringa)
- PCDMISSYSTEMHIDDENATAPATH (stringa)
- PCDMISSYSTEMVISIBLEDATAPATH (stringa)
- PCDMISSYSTEMREPORTINGPATH (stringa)
- PROBEDATA (varie)
- QUALTOOLDATA (varie)
- RAD2DEG (matematica)
- RIGHT (stringa)
- ROUND (matematica)
- SETROTABDATA (varie)
- SIN (matematica)
- SQRT (matematica)
- SYSTEMDATE (stringa)
- SYSTEMTIME (stringa)
- SYSTIME (stringa)
- TAN (matematica)
- TUTORELEMENT (varie)
- UNIT (punto)
- UPPERCASE (stringa)

## Funzioni su stringhe

Con le stringhe di testo vengono utilizzate le seguenti funzioni:

### **CHR**

Conversione caratteri: *CHR(<Intero>)*

Questa funzione restituisce una stringa costituita dal carattere corrispondente al valore decimale ASCII.



## CONCAT

Questa funzione concatena tutte le stringhe specificate nelle espressioni da 1 a N in un'unica stringa: `CONCAT (<espressione1>, <espressione2>, & <espressioneN>)`

## ELAPSEDEXECUTIONTIME

Tempo di esecuzione trascorso formattato: *ELAPSEDEXECUTIONTIME()*

Questa funzione restituisce il tempo trascorso dal momento in cui è iniziata l'esecuzione della routine o della mini routine di misurazione. Il tempo di esecuzione trascorso è il tempo passato durante la parte DCC dell'esecuzione; non registra il tempo speso nelle pause in attesa dell'attenzione richiesta dall'utente. Tali pause comprendono pause nell'esecuzione durante l'esecuzione dei commenti o dei messaggi di PC-DMIS e i messaggi di errore che possono comunque arrestare l'esecuzione.

Si può registrare in ogni momento di tempo di esecuzione della routine o della mini routine di misurazione assegnando la funzione a una variabile, come:



```
ASSEGNA/V1=ELAPSEDEXECUTIONTIME ()
```

L'ora è restituita nel formato predefinito "hh:mm:ss". Si può misurare il tempo di esecuzione trascorso anche in altri formati:

- Usare `ASSIGN/V1=FORMAT(ELAPSEDEXECUTIONTIME(),"hh:mm:ss")` o `ASSIGN/V1=ELAPSEDEXECUTIONTIME()` per ottenere il tempo in ore, minuti e secondi.
- Usare `ASSIGN/V1=FORMAT(ELAPSEDEXECUTIONTIME(),"mm:ss")` per ottenere il tempo in minuti e secondi.
- Usare `ASSIGN/V1=FORMAT(ELAPSEDEXECUTIONTIME(),"ss")` per ottenere il tempo in secondi.

## ELEMENTO

Posizione stringa secondaria delimitata: *ELEMENT(<intero>, <stringa1>, <stringa2>)*

Questa funzione restituisce l'n-esima sottostringa (elemento) della stringa2 utilizzando stringa1 come testo di delimitazione che divide gli elementi in stringa2.



Si supponga che la stringa 2 sia composta da "6, 12, 8, 4, 5" e che la stringa 1 sia composta da una virgola ",". I cinque elementi che possono essere recuperati singolarmente con il comando sono "6", "12", "8", "4" e "5".

## EQUAL

Confronto di stringhe senza distinzione tra maiuscole e minuscole:  
**EQUAL(<stringa1>, <stringa2>)**

Questa funzione confronta due stringhe (ignorando la distinzione tra maiuscole e minuscole) per determinare se sono identiche. Restituisce 1 se le stringhe sono uguali e 0 se le stringhe sono diverse.

## FORMAT

Formato: **FORMAT(<stringa>, <intero, doppio o punto>)**

Questa funzione accetta due espressioni e restituisce una stringa formattata, come quando si utilizza la funzione *sprintf* in C++.

- L'espressione 1 deve essere una *stringa* e contiene uno o tre specificatore di formato. Se è di tipo diverso, il valutatore dell'espressione tenta di forzarla a stringa. La stringa deve contenere *un solo* specificatore di formato se l'Espressione 2 è contiene valori interi o in doppia precisione e *tre* specificatori di formato (vedere il paragrafo seguente) se l'Espressione 2 è di tipo.
- L'Espressione 2 deve essere del tipo *intero*, *doppia precisione* o *punto*. Se si usa un tipo diverso, il valore dell'espressione è 0.

### Specificatore di formato per la funzione FORMAT:

Lo specificatore di formato deve avere la stessa sintassi usata dalla funzione *sprintf* utilizzata nel linguaggio di programmazione C++.

Uno specificatore di formato contiene campi obbligatori e campi opzionali, ed ha la seguente sintassi:

**%[flag] [larghezza] [.precisione] tipo**

Ciascun campo dello specificatore di formato è un carattere singolo, oppure un numero che indica una particolare opzione di formato. Il più semplice specificatore di formato usa soltanto il simbolo di percentuale seguito da un carattere (per es.

%d). Se il segno di percentuale è seguito da un carattere che non ha il significato di campo di formato, tale carattere viene copiato in STDOUT(output standard). Per esempio, per stampare il carattere percento, si componga %%.

Il flag, la larghezza e la precisione sono campi opzionali che precedono il tipo e gestiscono altri aspetti del formato. Tali valori vengono descritti più avanti.

### flag

Questi *caratteri facoltativi* gestiscono l'allineamento nell'output, la stampa dei segni, degli spazi, dei punti decimali e dei prefissi per numeri ottali o esadecimali. Uno specificatore di formato può presentare più di un flag.

Ecco i flag possibili:

–

**Significato:** allinea a sinistra il risultato all'interno dell'ampiezza di campo indicata.

**Impostazione predefinita:** Allineamento a destra.

+

**Significato:** se il valore in output ha un segno, un segno + oppure - lo precede.

**Impostazione predefinita:** il segno compare soltanto nel caso di numeri con segno meno (–).

0

**Significato:** se la larghezza è preceduta da 0, vengono aggiunti zeri fino a completare la larghezza. Se c'è uno zero ed un –, lo zero viene ignorato. Se 0 è specificato in un formato intero (i, u, x, X, o, d), lo 0 viene ignorato.

**Impostazione predefinita:** Nessun riempimento.

spazio (' ')

**Significato:** anteporre al valore dell'output uno spazio se tale valore ha un segno positivo; lo spazio viene ignorato se sono presenti sia uno spazio che i segni +.

**Impostazione predefinita:** Nessuno spazio.

#

**Significato 1:** quando è usato con i tipi o, x, oppure X, il flag # precede rispettivamente qualsiasi output non nullo con 0, 0x o 0X.

**Impostazione predefinita 1:** Senza prefisso.

**Significato 2:** quando viene utilizzato con il tipo e, E o f, l'indicatore numerico forza il valore di output a contenere un punto decimale in tutti i casi.

**Impostazione predefinita 2:** Il punto decimale è presente soltanto se seguito da almeno un numero.

**Significato 3:** Quando utilizzato con il formato g o G, l'indicatore numerico forza il valore di output a contenere un punto decimale in tutti i casi ed evitare il troncamento degli zero finali.

**Impostazione predefinita 3:** Il punto decimale viene visualizzato solo se seguito da cifre. Gli zeri finali vengono troncati. È ignorato se usato insieme a d, i oppure u.

### larghezza

Questo secondo campo opzionale, o argomento, gestisce il numero minimo di caratteri da stampare. È un numero intero decimale non negativo.

- Se il numero dei caratteri dell'output è minore della larghezza specificata, vengono aggiunti spazi alla sinistra o alla destra dei

caratteri — a seconda che sia specificato o meno il flag — di allineamento a sinistra — fino al raggiungimento della dimensione minima impostata.

- Se il valore della larghezza è preceduto da uno 0, vengono aggiunti zeri fino a raggiungere la larghezza minima (non è utile per i numeri allineati a sinistra).
- La definizione della larghezza non causa mai il troncamento di un valore. Se il numero dei caratteri di output è maggiore della dimensione specificata, oppure se non se ne definisce alcun valore, vengono stampati tutti i caratteri, in base alla precisione definita di seguito.

### **precisione**

Questo terzo campo o argomento facoltativo specifica il numero di caratteri da stampare, il numero di cifre decimali, oppure il numero di cifre significative. A differenza della definizione di larghezza, la precisione può causare sia il troncamento dell'output, sia l'arrotondamento, nel caso di un valore in virgola mobile. È un intero non negativo, preceduto da un punto (.).

### **tipo**

Questo carattere è obbligatorio e determina se l'argomento associato è un intero, un double, oppure un punto. L'elenco dei tipi disponibili contiene:

**d** - intero decimale con segno

**i** - intero decimale con segno

**o** - intero ottale senza segno

**u** - intero decimale senza segno

**x** - intero esadecimale senza segno, con la codifica "abcdef"

**X** - intero esadecimale senza segno, con la codifica "ABCDEF"

**e** - valore esponenziale in doppia precisione [-]d.dddd e [sign]ddd

**E** - come e, eccetto per l'uso di E prima dell'esponente

**f** - in doppia precisione con la forma [-]dddd.dddd

**g** - assume il formato e, oppure f, a seconda di qual è il più compatto

**G** - come g, eccetto per l'uso di G prima dell'esponente

### Esempio di FORMAT

Questo esempio riporta diverse istruzioni che utilizzano la funzione FORMAT all'interno di una routine di misurazione:

ASSEGNA/V1=PROBEDATA("SCOSTAMENTO")	V1 diventa di tipo punto e rappresenta gli scostamenti del tastatore corrente. Usando i valori nella routine di misurazione utilizzata per questo esempio, V1 diventa: <-1.8898, 1.8898, 5.704>
ASSEGNA/V3=FORMAT("%.5f, %.5f, %.5f", V1)	V3 diventa una stringa. La stringa viene formattata usando l'oggetto punto della variabile V1. V3 ora contiene - 1.88976, 1.88976, 5.70403
ASSEGNA/V4=1.123456789	V4 diventa in doppia precisione.
ASSEGNA/V5=FORMAT("%.5f", V4)+FORMAT("%.6f", V4)+FORMAT("%.7f", V4)+FORMAT("%.8f", V4)	V5 diventa di tipo stringa con valore: 1.12346 1.123457 1.1234568 1.12345679
ASSEGNA/V6A="Il valore di V4 è: "+FORMAT("%.8f", V4)	V6A diventa di tipo stringa con il valore: Il valore di V4 è: 1.12345679
ASSEGNA/V6B=FORMAT("Il valore di V4 è: %.8f", V4)	Il risultato dell'espressione resta lo stesso di V6A sopra riportato.
ASSEGNA/V7=4444	V7 diventa in doppia precisione, poiché tutti i numeri si suppongono in doppia precisione, a meno che siano forzati ad interi.
ASSEGNA/V8=FORMAT("%o",INT(V7))	V8 diventa di tipo stringa con il valore: 10534
ASSEGNA/V9=FORMAT("%u",INT(-1))	V9 diventa di tipo stringa con il valore: 4294967295

ASSEGNA/V10=FORMAT("%x",INT(2143))	V10 diventa di tipo stringa con il valore: 85f
ASSEGNA/V11=FORMAT("%X",INT(9567))	V11 diventa di tipo stringa con il valore: 255F
ASSEGNA/V12=FORMAT("%e",0.0005432)	V12 diventa di tipo stringa con valore: 5.432000e-004
ASSEGNA/V13=FORMAT("%E",145.3421)	V13 diventa di tipo stringa con il valore: 1.453421E+002
ASSEGNA/V14=FORMAT(",%6d,",INT(1))	V14 diventa di tipo stringa con il valore: , 1,
ASSEGNA/V15=FORMAT(",%-6d,",INT(1))	V15 diventa di tipo stringa con il valore: , 1 ,

## GETSETTING

Questa funzione consente di ottenere diverse impostazioni di PC-DMIS, in base al parametro inserito nella stringa.

GETSETTING(<stringa>)

Si possono utilizzare i parametri di stringa seguenti:

- "DCC Mode" – Restituisce 1 se PC-DMIS si trova in modalità DCC, altrimenti restituisce 0.
- "Manual Mode" – Restituisce 1 se PC-DMIS si trova in modalità manuale, altrimenti restituisce 0.
- "Current Alignment" – Restituisce una stringa dell'allineamento corrente.
- "Current Workplane" – Restituisce una stringa del piano di lavoro corrente.
- "Workplane Value" – Restituisce un valore numerico del piano di lavoro corrente.
- "PreHit" – Restituisce il valore di avvicinamento in uso come numero in doppia precisione.
- "Retract" – Restituisce il valore di ritrazione corrente come numero in doppia precisione.
- "Check" – Restituisce il valore di controllo corrente come numero di in doppia precisione.

- "Touch Speed" – Restituisce il valore della velocità di contatto corrente come numero in doppia precisione.
- "Move Speed" – Restituisce il valore della velocità di movimento corrente come numero in doppia precisione.
- "Fly Mode" – Restituisce 1 se PC-DMIS utilizza la modalità Fly, altrimenti restituisce 0.
- "Ph9 present" – Restituisce 1 se Ph9/Ph10 è presente, altrimenti restituisce 0.
- "Manual CMM" – Restituisce 1 se la CMM è una CMM manuale, altrimenti restituisce 0.
- "LangStr(<Numero o ID>)" – Restituisce una stringa dalle risorse di PC-DMIS nella lingua corrente, da un numero di ID risorsa oppure da uno degli ID seguenti:

"Sì", "No", "Oper", "Rapp", "Input", "Doc", "SiNo", "Lettura", "Interno", "Esterno", "Rett ", "Pol ", "Out", "In", "Quad\_min", "Sep\_Min", "Max\_Isc", "Min\_Circos", "Rag\_Fisso", "Piano di lavoro", "AsseX", "AsseY", "AsseZ", "X+", "X-", "Y+", "Y-", "Z+", "Z-", "Punto", "Piano", "Linea", "Cerchio", "Sfera", "Cilindro", "Asola\_Rotonda", "Asola\_Quadrata", "Cono" o "Nessuno".

Se il valore utilizzato è un numero positivo, PC-DMIS estrae la stringa dal file resource.dll. Se si utilizza un numero negativo, PC-DMIS estrae la stringa dal file strings.dll (tabella delle stringhe).

- "Extended Sheet Metal" – Restituisce 1 se la casella di opzione **Mostra opzioni per lamiera estese** è selezionata nella finestra di dialogo **Opzioni di impostazione**, altrimenti restituisce 0.
- "LastHitMove(X)" – Restituisce il valore X del comando HIT /BASIC o MOVE/POINT più recente. PC-DMIS deve trovarsi in modalità DCC.
- "LastHitMove(Y)" – Restituisce il valore Y del comando HIT/BASIC o MOVE/POINT più recente. Perché questo parametro funzioni, PC-DMIS deve trovarsi in modalità DCC.
- "LastHitMove(Z)" – Restituisce il valore Z del comando HIT/BASIC o MOVE/POINT più recente. Perché questo parametro funzioni, PC-DMIS deve trovarsi in modalità DCC.

Per determinare se PC-DMIS si trova in modalità manuale o DCC, si può usare la funzione GETSETTING come mostrato qui sotto:

`ASSEGNA/DCCMODEVAR=GETSETTING("DCC Mode")` - Assegna alla variabile DCCMODEVAR il valore 1 se PC-DMIS si trova in modalità DCC, altrimenti assegna 0.

`ASSEGNA/MANMODEVAR=ETSETTING("Manual Mode")` - Assegna alla variabile MANMODEVAR il valore 1 se PC-DMIS si trova in modalità manuale, altrimenti assegna 0.

Si può usare la funzione GETSETTING anche per determinare il piano di lavoro in uso come mostrato qui sotto:

`ASSEGNA/WORKPLANE_ID=GETSETTING("Current Workplane")` - Assegna alla variabile WORKPLANE\_ID il valore della stringa del piano di lavoro corrente (Z+, Z- ecc.).

`ASSEGNA/WORKPLANE_VALUE=GETSETTING("Workplane Value")` - Assegna alla variabile WORKPLANE\_VALUE un valore numerico che rappresenta il piano di lavoro. Ai piani di lavoro sono associati i seguenti valori: Z+ = 0, Z- = 3, X+ = 1, X- = 4, Y+ = 2 o Y- = 5.

## GETTEXT

Questa funzione restituisce il testo esistente nel campo di dati specificato:  
`GETTEXT(<stringa o intero>, <intero>, <puntatore>)`

Questa funzione dispone di tre campi.

### Primo campo - Numero o descrizione di un campo dati

Il primo campo può essere la descrizione di una stringa del campo di dati, indicata da una (A) nell'immagine sotto riportata oppure il numero del campo di dati, indicato da una (C) nella stessa immagine.



La stringa (B) nell'immagine seguente non è usata in questa funzione ma a volte viene usata in espressioni di automazione o nei rapporti.

Per avere questi valori:

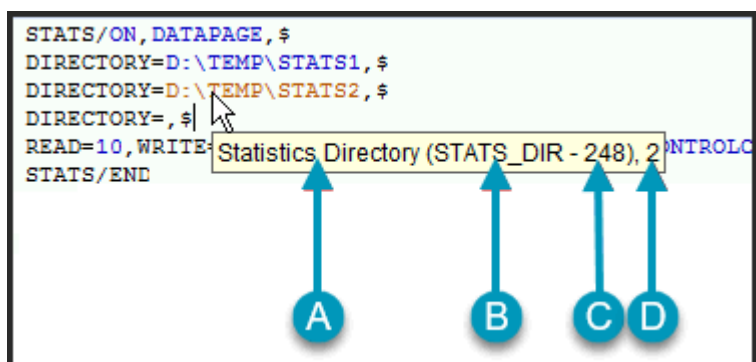
1. Attivare la modalità Comando di PC-DMIS. Fare clic con il pulsante destro del mouse in un punto qualsiasi nella finestra di modifica. Verrà visualizzato un menu di scelta rapida.
2. Dal menu di scelta rapida selezionare **Modifica visualizzazione popup**, quindi **Informazioni su tipo dati**.



3. Posizionare il mouse su un campo dati nella finestra di modifica. Verranno visualizzati la descrizione, il numero e l'indice del tipo di quei dati.



La descrizione del tipo può essere differente a seconda della lingua. Se la routine di misurazione è eseguita su una versione di PC-DMIS che usa una lingua diversa dall'inglese, usare il numero del tipo.



*Esempio di informazioni sul tipo di dati contenenti (A) la descrizione del tipo, (B) l'identificativo della stringa del tipo, (C) il numero del tipo e (D) l'indice del tipo*

## Secondo campo - Indice del tipo

Il secondo campo corrisponde all'indice del tipo, indicato come (D) nell'immagine precedente. Questo campo di solito è zero a meno che non siano presenti altre istanze dello stesso tipo di campo nello stesso comando, come nel caso dei campi DIRECTORY mostrati nella precedente immagine. Il valore corretto per questo campo è ottenibile nello stesso modo del valore per il primo campo.

## Terzo campo - Puntatore di comando

Il terzo campo corrisponde al puntatore di un comando. Punta al comando contenente il campo dal quale viene ottenuto il testo. Per specificare questo campo si può usare la notazione del puntatore dei comandi (cioè {{F15}}) o l'espressione GETCOMMAND come mostrato qui:



`ASSEGNA/V1=GETTEXT("Tipo di matematica best-fit", 0, {F15})` Questo comando assegna a V1 il valore del tipo di formula best fit dell'elemento F15.

`ASSEGNA/V2=GETCOMMAND("Commento", "TOP", 1)` - A V2 è assegnato un puntatore al primo commento a partire dall'inizio della routine di misurazione.

`ASSEGNA/V3=GETTEXT("Tipo commento", 1, V2)` - A V3 è assegnato il valore riportato nel campo di selezione Tipo commento. Se il primo commento nella routine di misurazione deve essere visualizzato all'operatore, il valore di V3 sarà la stringa "OPER".

Fare riferimento a "Funzioni del puntatore" per informazioni sull'espressione `GETCOMMAND` utilizzata per impostare un puntatore su un comando.

## **GETTEXT**

Questa funzione restituisce il testo esistente nel campo dati specificato:

`GETTEXT(<stringa o numero intero>, <numero intero>, <stringa del contenuto>, <puntatore del comando>)`

Questa funzione ha quattro campi.

### **Primo campo - Numero o descrizione di un campo dati**

Il primo campo può essere una stringa di descrizione oppure il valore numerico del campo dati; è indicata da (A) nell'immagine seguente.



Se invece dell'identificativo numerico si usa la stringa (elemento (A) nell'immagine seguente), PC-DMIS la convertirà automaticamente nel valore numerico corretto.

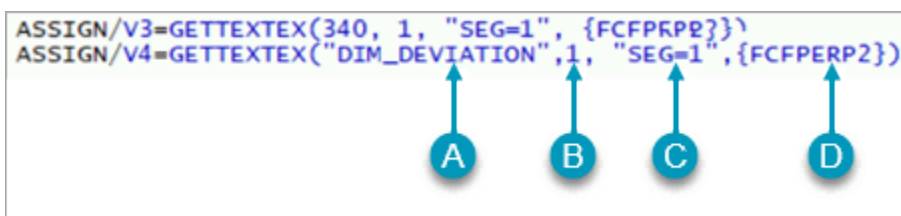
Per esempio se si usa l'identificativo della stringa "DIM\_DEVIATION", PC-DMIS lo convertirà internamente nel valore numerico 340. Quando poi si passa con il puntatore del mouse sul comando nella finestra di modifica, il comando a comparsa mostra la stringa di testo nonché il valore del suo identificativo numerico. Quindi, se in questo esempio si passa con il puntatore del mouse sul comando nella finestra di modifica, il comando a comparsa mostrerà  
(DIM\_DEVIATION - 340), 1, SEG=1.

Si può anche usare semplicemente al valore numerico se lo si conosce.

Per avere questi valori:

1. Portare PC-DMIS nella modalità di comando e quindi fare clic con il pulsante destro del mouse in un punto qualsiasi della finestra di modifica. Verrà visualizzato un menu di scelta rapida.
2. Dal menu di scelta rapida selezionare **Modifica visualizzazione popup**, quindi **Informazioni su tipo dati**.
3. Posizionare il mouse su un campo dati nella finestra di modifica. Verranno visualizzati la descrizione, il numero e l'indice del tipo di quei dati. Passare con il puntatore del mouse su un Tipo\_D esteso per mostrare il contenuto della stringa dopo i due punti.

La descrizione del tipo può essere differente a seconda della lingua. Se la routine di misurazione è eseguita su una versione di PC-DMIS che usa una lingua diversa dall'inglese, usare il numero del tipo.



*Esempio di informazioni sul tipo di dati che mostra (A) la stringa o l'identificativo numerico del tipo, (B) l'indice del tipo, (C) la stringa del contenuto e (D) il puntatore del comando*

## Secondo campo - Indice del tipo

Il secondo campo corrisponde all'indice del tipo, indicato con B nell'immagine precedente. Questo campo di solito è zero a meno che non siano presenti altre istanze dello stesso tipo di campo nello stesso comando, come nel caso dei campi DIRECTORY mostrati nella precedente immagine. Si può ottenere il valore corretto per questo campo nello stesso modo del valore per il primo campo.

## Terzo campo—Stringa del contenuto

Il terzo campo è la stringa del Tipo\_D esteso indicata con C nell'immagine precedente.

## Quarto campo —Puntatore del comando

Il quarto campo è il puntatore del comando, indicato con D nella figura precedente. Punta al comando da cui l'espressione estrae i dati. Si può usare la notazione del puntatore dei comandi (cioè {FCFPERP2}) o l'espressione GETCOMMAND come mostrato qui:

```
ASSEGNA/V1=GETTEXTEX ("DEV_DIAMETRO",1,"SEG=1",{FCFPERP2}
) - Questo comando assegna a V1 il valore attuale della deviazione ricavata
da elemento 1, segmento 1, dimensione FCFPERP2.
```

Fare riferimento a "Funzioni del puntatore" per informazioni sull'espressione GETCOMMAND utilizzata per impostare un puntatore su un comando.



La funzione GETTEXTEX aggiunge supporto per i tipi\_D estesi che contengono una stringa CONTENT. Attualmente, solo i comandi di tolleranza geometrica di PC-DMIS usano i tipi\_D estesi.

## GETPROGRAMINFO

Questa funzione restituisce le informazioni nella routine di misurazione basate sui parametri inoltrati: `GETPROGRAMINFO(<stringa>, <stringa facoltativa>)`

Questa funzione ha al massimo due stringhe come parametri. Nella maggior parte dei casi servirà solo il primo parametro. I campi delle stringhe non distinguono tra maiuscole e minuscole.

## Primo campo—Stringa

Questo campo contiene una stringa di input che descrive l'informazione restituita.

**CADMODELFILE** - Restituisce il percorso completo del nome del file del modello CAD importato nella routine di misurazione.

**CADMODELNAME** - Restituisce solo il nome (non il percorso) del file del modello CAD importato nella routine di misurazione.

**DATE** - Restituisce la data corrente.

**DRAWING** - Come REVISION, restituisce il numero di revisione come definito nell'intestazione.

**ELAPSEDTIME** - Restituisce il tempo trascorso dall'inizio dell'esecuzione.

**FILENAME** - Restituisce il nome del file della routine di misurazione.

**NUMMEAS** - Restituisce il numero di dimensioni eseguite.

**NUMOOT** - Restituisce il numero di dimensioni fuori tolleranza eseguite.

**PARTNAME** - Restituisce il nome del pezzo come definito nell'intestazione della routine di misurazione.

**PARTPATH** - Restituisce il percorso completo del file della routine di misurazione.

**PCDMISVERSION** - Restituisce un valore della stringa della versione correntemente installata del software PC-DMIS.

**PRGSCHEMA** - Restituisce un intero del numero di schema di PC-DMIS del file della routine di misurazione. Questo è un valore interno utilizzato da PC-DMIS per indicare i comandi e le opzioni che sono serializzati.

**PRGVERSION** - Restituisce la stringa del valore del numero di versione di PC-DMIS del file della routine di misurazione. È possibile salvare un file della routine di misurazione in modo che sia compatibile con una versione specifica. Per maggiori informazioni, vedere "Salva con nome" nel capitolo "Uso delle opzioni base del menu File".

**PROBEFILE** - Restituisce il nome del file del tastatore in uso.

**REPORTNAME** - Restituisce il nome file di output.

**REVISION** - Restituisce il numero di revisione come definito nell'intestazione.

**SERIALNUM** - Restituisce il numero di serie come definito nell'intestazione.

**SEQNUM** - Come STATSCOUNT, anche questa stringa restituisce il numero di statistiche correnti.

**SHRINK** - Restituisce il fattore di scala globale.

**STATSCOUNT** - Restituisce il numero di statistiche correnti.

**TEMP** - Restituisce la temperatura per la seconda stringa di input facoltativa. Vedere "Secondo campo—Stringa opzionale" di seguito.

**TIME** - Restituisce l'ora corrente.

**TIPID** - Restituisce il nome del file della punta in uso.

### Secondo campo—Stringa opzionale

Il secondo campo è una stringa di input opzionale. Serve solo se si usa TEMP nel primo campo di input. Le possibili stringhe seguenti provengono dal comando di compensazione della temperatura. Per ulteriori informazioni, vedere "Compensazione della temperatura" nel capitolo "Impostazione delle preferenze".

**HIGH\_THRESHOLD** - Restituisce la soglia superiore della temperatura

**LOW\_THRESHOLD** - Restituisce la soglia inferiore della temperatura

**REF\_TEMP** - Restituisce la temperatura di riferimento

**TEMPP** - Restituisce la temperatura del sensore del pezzo

**TEMPX** - Restituisce la temperatura del sensore dell'asse X

**TEMPY** - Restituisce la temperatura del sensore dell'asse Y

**TEMPZ** - Restituisce la temperatura del sensore dell'asse Z

## Esempio



```
$$ NO, Questo esempio di codice mostra il numero di  
dimensioni totali e il numero delle dimensioni fuori  
tolleranza..
```

```
ASSEGNA/V1=GETPROGRAMINFO("NUMMEAS")
```

```
ASSEGNA/V2=GETPROGRAMINFO("NUMOOT")
```

```
COMMENTO/REP
```

```
"Dimensioni totali: "+V1
```

```
"Fuori tolleranza totale: "+V2
```

```
$$ NO, Questo esempio di codice restituisce la  
temperatura del sensore sull'asse Z.
```

```
ASSEGNA/V3=GETPROGRAMINFO("TEMP", "TEMPZ")
```

```
COMMENTO/REP
```

```
"Temperatura sull'asse Z: "+V3
```

## GETTRACEVALUE

Leggi valore traccia: *GETTRACEVALUE(<stringa>)*

Questa funzione ha un unico parametro. Restituisce il valore del comando `TRACEFIELD` nella routine di misurazione.

*<stringa>* è una stringa che distingue tra maiuscole e minuscole del nome della traccia di cui si desidera ottenere il valore.

Se si hanno più campi di tracciatura con il nome della stessa traccia, questa funzione restituisce il valore del campo di tracciatura più recente sopra la funzione. Se il campo di tracciatura non contiene un valore, questa funzione restituisce 0. Nell'esempio seguente "Operatore" è il nome del campo di tracciatura nella routine di misurazione:



```
ASSEGNA/V2=GETTRACEVALUE("Operatore")
```

## INDEX

Posizione stringa secondaria: *INDEX(<stringa>, <stringa>)*

Questa funzione restituisce la posizione della seconda stringa all'interno della prima stringa. La prima lettera della stringa è 1. Se non è possibile individuare la sottostringa all'interno della stringa principale, viene restituito il valore 0.

Per un esempio di questa funzione, vedere l'argomento "Esempio di codice per il comando Leggi riga" nel capitolo "Uso di input/output di file".

## **LASTEXECUTIONTIME**

Ora dell'ultima esecuzione formattata: *LASTEXECUTIONTIME()*

Questa funzione restituisce l'ora dell'ultima esecuzione registrata da PC-DMIS e memorizzata nel file *<nome della routine di isurazione>.MiniRoutines.xml*. L'ora dell'ultima esecuzione appare nella finestra di dialogo **Esecuzione**. L'ora è restituita nel formato "hh:mm:ss".

## **LEFT**

Numero di caratteri a sinistra della stringa: *LEFT(<stringa>, <n>)*

Questa funzione restituisce una stringa costituita dal numero di caratteri più a sinistra specificati dalla seconda espressione (n) presenti nella stringa specificata nella prima espressione (stringa).

La prima espressione (Stringa) è forzata al tipo Stringa. La seconda espressione (n) è forzata al tipo Intero.

Per un esempio di questa funzione, vedere l'argomento "Esempio di codice per il comando Leggi riga" nel capitolo "Uso di input/output di file".

## **LEN**

Lunghezza della stringa: *LEN(<stringa>)*

Questa funzione restituisce il numero di caratteri della stringa.

## **LOWERCASE**

Crea una stringa in minuscolo: *LOWERCASE(<stringa>)*

Questa funzione restituisce una stringa che è l'equivalente della stringa in caratteri minuscoli.

## **MID**

Gli n caratteri centrali di una stringa: *MID(<stringa>, <intero>, <intero facoltativo>)*



Questa funzione restituisce una sottostringa costituita dai caratteri della stringa specificata nel primo parametro a partire dalla posizione specificata dal secondo parametro e lunga gli *n* caratteri specificati dal terzo parametro. Se il terzo parametro non viene specificato, viene restituita la parte restante della stringa.

Per un esempio di questa funzione, vedere l'argomento "Esempio di codice per il comando Leggi riga" nel capitolo "Uso di input/output di file".

### **ORD**

Conversione ordinali: *ORD(<stringa>)*

Questa funzione restituisce il valore intero ASCII della prima lettera della stringa (0-255).

### **PCDMISAPPLICATIONPATH**

Visualizzazione del percorso completo: *PCDMISAPPLICATIONPPATH()*

Questa funzione restituisce il valore della stringa contenente il percorso completo della directory dell'applicazione in cui è installato PC-DMIS. Questa directory contiene l'eseguibile principale e gli altri file del programma necessari per eseguire PC-DMIS.

### **PCDMISUSERHIDDENDATAPATH**

Visualizzazione del percorso completo: *PCDMISUSERHIDDENDATAPATH()*

Questa funzione restituisce il valore della stringa contenente il percorso completo della directory nascosta dei dati dell'utente usata da PC-DMIS. Per informazioni sui file contenuti in questa directory, vedere "Informazioni sulla posizione dei file".

### **PCDMISUSERVISIBLEDATAPATH**

Visualizzazione del percorso completo: *PCDMISUSERHIDDENDATAPATH()*

Questa funzione restituisce il valore della stringa contenente il percorso completo della directory visibile dei dati dell'utente usata da PC-DMIS. Per informazioni sui file contenuti in questa directory, vedere "Informazioni sulla posizione dei file".

### **PCDMISSYSTEMHIDDENDATAPATH**

Visualizzazione del percorso completo: *PCDMISSYSTEMHIDDENDATAPATH()*

Questa funzione restituisce il valore della stringa contenente il percorso completo della directory nascosta dei dati del sistema usata da PC-DMIS. Per informazioni sui file contenuti in questa directory, vedere "Informazioni sulla posizione dei file".

### **PCDMISSYSTEMVISIBLEDATAPATH**

Visualizzazione del percorso completo: PCDMISSYSTEMVISIBLEDATAPATH()

Questa funzione restituisce il valore della stringa contenente il percorso completo della directory visibile dei dati del sistema usata da PC-DMIS. Per informazioni sui file contenuti in questa directory, vedere "Informazioni sulla posizione dei file".

### **PCDMISSYSTEMREPORTINGPATH**

Visualizzazione percorso completo: PCDMISSYSTEMREPORTINGPATH()

Questa funzione restituisce il valore della stringa contenente il percorso completo della directory dei rapporti usata da PC-DMIS. Questa directory contiene i modelli di rapporto ed etichetta utilizzati nella finestra Rapporto.

### **RIGHT**

Gli n a destra della stringa: *RIGHT*(<stringa>, <intero>)

Questa funzione restituisce una stringa costituita dagli n caratteri più a destra specificati dal numero intero nella stringa.

### **SYSTEMDATE**

Data del sistema: SYSTEMDATE(<stringa formato data>)

Questa funzione restituisce la stringa formattata della data contenente i dettagli della data immessa. Ad esempio, il comando SYSTEMDATE("MM'/'dd'/'yy") restituirà la stringa "02/12/14" se la data corrente è il 12 febbraio 2014.

Utilizzare i seguenti elementi per creare la stringa della data. È necessario usare le lettere maiuscole e minuscole come mostrato di seguito (MM al posto di mm). I caratteri non relativi alla data (ad esempio gli spazi) presenti tra gli elementi che costituiscono la stringa in formato data, mantengono nella stringa di output la stessa posizione che avevano in quella di input. I caratteri della stringa di input compresi tra virgolette singole vengono visualizzati nella stessa posizione all'interno della stringa di output, ma senza virgolette.

**g** - È il giorno del mese espresso in cifre. Senza lo zero iniziale per le date composte da una sola cifra.

**dd** – È il giorno del mese espresso in cifre. Con lo zero iniziale per le date composte da una sola cifra.

**ddd** – Abbreviazione di tre lettere del giorno della settimana.

**dddd** – Nome completo del giorno della settimana.

**M** – È il mese espresso in cifre, senza zeri iniziali per i mesi composti da una sola cifra.

**MM** – È il mese espresso in cifre, con uno zero iniziale per i mesi composti da una sola cifra.

**MMM** – Abbreviazione di tre lettere del nome del mese.

**MMMM** – Nome completo del mese.

**y** – È l'anno espresso in cifre, senza zeri iniziali per gli anni composti da una sola cifra.

**yy** – È l'anno espresso in cifre, con uno zero iniziale per gli anni composti da una sola cifra.

**yyyy** – È l'anno espresso in quattro cifre.

## **SYSTEMTIME**

Ora formattata del sistema: *SYSTEMTIME(<stringa formato ora>)*

Questa funzione restituisce la stringa formattata dell'ora contenente i dettagli dell'ora immessa. Ad esempio, il comando `SYSTEMTIME("hh:mm:ss tt")` restituisce l'ora in una stringa formattata, ad esempio "11:29:40 PM".

Utilizzare i seguenti elementi della stringa per creare la stringa relativa all'ora. È necessario rispettare le indicazioni relative all'uso delle maiuscole e minuscole riportate di seguito (**tt** anziché **TT**). I caratteri non relativi all'ora (ad esempio gli spazi) presenti tra gli elementi che costituiscono la stringa in formato ora, mantengono nella stringa di output la stessa posizione che avevano in quella di input. I caratteri della stringa di input compresi tra virgolette singole vengono visualizzati nella stessa posizione all'interno della stringa di output, ma senza virgolette.

**h** - Ora espressa in formato 12 ore, senza lo zero iniziale in caso di cifra singola.

**hh** - Ora espressa in formato 12 ore, con lo zero iniziale in caso di cifra singola

**H** - Ora espressa in formato 24 ore, senza lo zero iniziale in caso di cifra singola

**HH** - Ora espressa in formato 12 ore, con lo zero iniziale in caso di cifra singola

**m** - Minuti senza zero iniziale per i minuti a una sola cifra

**mm** - Minuti con lo zero iniziale per i minuti a singola cifra

**s** - Secondi senza zero iniziale per i secondi a una sola cifra

**ss** - Secondi con lo zero iniziale per i secondi a una sola cifra

**t** - Stringa identificativa dell'ora composta da un carattere, ad esempio A o P

**tt** - Stringa identificativa dell'ora composta da più caratteri, ad esempio AM o PM

## **SYSTIME**

Ora del sistema: *SYSTIME()*

Questa funzione restituisce una stringa con l'ora del sistema. Questa funzione è diversa dalla funzione SYSTEMTIME descritta in precedenza. Restituisce automaticamente il giorno, la data e l'ora, seguiti dall'anno.

Esempio: "Mer febbraio 12 13:50:21 2014"



La stringa restituita relativa all'ora del sistema varia in base alle impostazioni del fuso orario locale.

## **UPPERCASE**

Crea una stringa in caratteri maiuscoli: *UPPERCASE(<stringa>)*

Questa funzione restituisce una stringa che è l'equivalente della stringa in caratteri maiuscoli.

## Funzioni matematiche

### **ABS**

Valore assoluto: *ABS(<Doppio>)*

Restituisce il valore assoluto di input.

### **ESP**

Esponenziale: *EXP(<Doppio>)*

Restituisce l'esponente dell'espressione.

### **LOG**

Logaritmo in base 10: *LOG(<Doppio>)*

Restituisce il logaritmo in base 10 dell'espressione.

### **LI**

Logaritmo naturale: *LN(<Doppio>)*

Restituisce il logaritmo naturale dell'espressione.

### **ROUND**

Arrotondamento: *ROUND(<Doppio>)*

Restituisce l'input arrotondato al numero intero più vicino.

### **SQRT**

Radice quadrata: *SQRT(<Doppio>)*

Restituisce la radice quadrata dell'input.

## Funzioni di trigonometria

Per impostazione predefinita, ogni funzione trigonometrica accetta e restituisce i radianti. Se si desidera lavorare con i valori in gradi, usare la funzione RAD2DEG descritta di seguito.

**ACOS**

Arco Coseno: *ACOS(<Doppio>)*

Restituisce l'arco coseno dell'espressione. Ad esempio, *ACOS(5.0)* restituisce 0. In generale, *ACOS(<espressione>)* restituisce l'arco coseno del valore dell'espressione.

**ASIN**

Arco Seno: *ASIN(<Doppio>)*

Restituisce l'arco seno dell'input.

**ATAN**

Arco tangente: *ATAN(<Doppio>)*

Restituisce l'arco tangente del valore di input.

**COS**

Coseno: *COS(<Doppio>)*

Restituisce il coseno dell'input.

**DEG2RAD**

Gradi in radianti: *DEG2RAD(<Doppio>)*

Restituisce l'input diviso per 360 e moltiplicato per  $2\pi$ . Converte il valore da gradi a radianti.

**RAD2DEG**

Radianti in gradi: *RAD2DEG(<Doppio>)*

Restituisce l'input moltiplicato per 360 e diviso per  $2\pi$ . Converte il valore da radianti a gradi.

**SIN**

Seno: *SIN(<Doppio> )*

Restituisce il seno dell'input.

## TAN

Tangente: *TAN(<Doppio>)*

Restituisce la tangente dell'input.



Le funzioni in cui l'input non è compreso nell'intervallo (come ACOS, ASIN, LOG, LN, SQRT e così via, che provocherebbero un arresto anomalo del sistema) restituiscono 0.

## Funzioni Punto

### ANGLEBETWEEN

Angolo tra: *ANGLEBETWEEN(<vettore>, <vettore>)*

Restituisce il valore in gradi dell'angolo tra i due vettori. I due parametri devono essere espressioni che valutano rispetto a un tipo di vettore. Per ottenere il vettore da un elemento, ad esempio, sarà necessario usare l'ID dell'elemento seguito dall'estensione .IJK. Questo si può vedere nel frammento di codice seguente.



```
F1      =GENERICO/PUNTO,DIPENDENTE,CARTESIANO,$
        NOM/XYZ,<3,3,3>,$
        MIS/XYZ,<3,3,3>,$
        NOM/IJK,<1,0,0>,$
        MIS/IJK,<1,0,0>

F2      =GENERICO/PUNTO,DIPENDENTE,CARTESIANO,$
        NOM/XYZ,<10,10,10>,$
        MIS/XYZ,<10,10,10>,$
        NOM/IJK,<0,0,1>,$
        MIS/IJK,<0,0,1>
        ASSEGNA/V1=F1.IJK
        ASSEGNA/V2=F2.IJK
        ASSEGNA/V3=ANGLEBETWEEN(V1,V2)
        COMMENTO/OPER,NO,INTERO=NO,CONTINUAZIONE
AUTOMATICA=NO,
        "L'angolo tra "+V1+" e "+V2+" è "+V3
```

**CROSS**

Prodotto incrociato: *CROSS*(<punto>, <punto>)

Il valore restituito è di tipo punto ed è il vettore unitario nella direzione del prodotto incrociato della prima e della seconda espressione.

**DELTA**

Scostamento vettore: *DELTA*(<punto>, <punto>, <doppio>)

La funzione prende la prima espressione (Punto) e calcola un nuovo punto nella direzione della seconda espressione (vettore) distanziato in base al valore della terza espressione. Ad esempio, *DELTA*(*MPOINT*(0,0,0), *MPOINT*(1,0,0), 10) restituisce il punto 10,0,0.

**DOT**

Prodotto punto: *DOT*(<punto>, <punto>)

Restituisce il punto risultante dal prodotto dei due punti (vettori).

**UNIT**

Vettore unità: *UNIT*(<punto>)

Restituisce il punto diviso per la sua lunghezza. Ad esempio, *UNIT*(*MPOINT*(0,0,0)) restituisce il punto 0,0,1.

**MPOINT**

Forzatura nel tipo punto: *MPOINT*(<espressione1>, <espressione2>, <espressione3>)

Forza le tre espressioni sul tipo Punto come mostrato in questo frammento di codice:

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

Dove:

V1.X = 2.5

V1.Y = 3.6

V1.Z = 4.0

Vedere "Forzatura nel tipo punto".



## Funzioni puntatori

### ***DIST2D***

Distanza 2D: DIST2D(<ELEM1>, <ELEM2>, <ELEM3>)



L'elemento deve essere compreso tra parentesi graffe.

Calcola la distanza tra i primi due argomenti del comando (Elem1 e Elem2), perpendicolare al terzo argomento (Elem3).

- Se il terzo argomento è un piano, PC-DMIS calcola la distanza tra i primi due argomenti perpendicolarmente al piano.
- Se il terzo argomento è una linea o un cilindro, PC-DMIS calcola la distanza tra i primi due argomenti perpendicolarmente al terzo argomento nel piano di lavoro attivo.

Ad esempio, se il piano XY è il terzo argomento e ha un vettore Z+ (0,0,1) e la distanza riportata è solo nell'asse Z.

### **Esempio**



```
ASSEGNA/V3=DIST2D ({CER1},{CER2},{PIANO1})  
COMMENTO/OPER,NO,SCHERMO  
INTERO=NO,CONTINUAZIONE AUTOMATICA=NO,  
V3
```

### ***DIST3D***

Distanza 3D: DIST3D(<ELEM1>, <ELEM2>)

Calcola la distanza 3D tra Feat1 e Feat2.

L'elemento deve essere compreso tra parentesi graffe.

**Esempio**

```
ASSEGNA/V3=DIST3D({CER1},{CER2})
COMMENTO/OPER,NO,SCHERMO
INTERO=NO,CONTINUAZIONE AUTOMATICA=NO,
V3
```

**GETCOMMAND**

Ottiene un puntatore per il comando specificato dai parametri: `GETCOMMAND(<intero o stringa>, <stringa>,<intero>`

**Primo parametro - Campo Informazioni su comando**

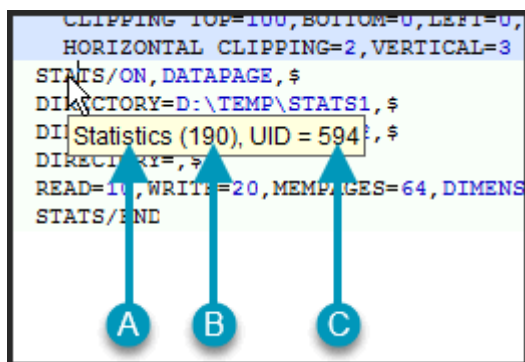
Il primo parametro è il campo delle informazioni sul comando. Specifica il tipo di comando da cercare. Possono essere trasferiti i seguenti elementi:

- Una stringa di descrizione del comando. Vedere (A) nel grafico riportato di seguito.
- Il numero del tipo di comando. Vedere (B) nel grafico riportato di seguito.
- L'identificatore univoco. Vedere (C) nel grafico riportato di seguito.

Se viene trasferito l'ID univoco del comando, non sono necessari altri argomenti.

Per ottenere la stringa di descrizione, il numero del tipo e l'identificatore del comando, attenersi alla seguente procedura:

1. Fare clic con il pulsante destro del mouse nella finestra di modifica.
2. Scegliere **Modifica visualizzazione popup | Informazioni su comando** (PC-DMIS deve essere in modalità Comando).
3. Posizionare il mouse sul comando desiderato. PC-DMIS visualizza la descrizione, il numero del tipo e l'identificatore del comando nel popup.



- A. Stringa di descrizione del comando
- B. Numero del tipo di comando
- C. Identificativo univoco (UID)

### Secondo parametro - Direzione della ricerca

Il secondo parametro è la direzione della ricerca. I valori validi includono:

Valore	Descrizione
SU	Questo valore indica che la ricerca deve partire dal comando corrente e procedere verso l'alto.
GIÙ	Questo valore indica che la ricerca deve partire dal comando corrente e procedere verso il basso.
INIZIO	Questo valore indica che la ricerca deve partire dall'inizio della routine di misurazione e procedere verso il basso.
FINE	Questo valore indica che la ricerca deve partire dall'ultimo oggetto nella routine di misurazione e risalire verso l'inizio.

### Terzo parametro - Istanza da trovare

Il terzo parametro indica quale istanza del comando deve essere trovata se sono presenti più istanze dello stesso comando nella routine di misurazione.



Se la routine di misurazione contiene due istanze di un comando STAT/ON e si desidera ottenere un puntatore verso la seconda istanza a partire dall'inizio, viene passato "2" come terzo parametro e "TOP" come secondo parametro come mostrato qui.

```
ASSEGNA/V1=GETCOMMAND("Statistiche","SUPERIORE",2)
```

La funzione GETCOMMAND si può usare per fornire il terzo parametro alla funzione stringa GETTEXT. Vedere "Funzioni stringa" per informazioni su GETTEXT.

### LEN

Numero di loop del contatore: *LEN(<POINTER>)*

Restituisce il numero di volte che un puntatore si trova in un loop. Ad esempio, se l'elemento CIR1 si trova in un ciclo che viene ripetuto 10 volte, sarà possibile memorizzare il numero di volte che CIR1 è stato misurato in una variabile mediante un'istruzione *ASSEGNA* come la seguente: *ASSEGNA/V1=LEN({CIR1})*

## Funzioni array

### ARRAY

Crea Array: *ARRAY(<ESPRESSIONE1>, <ESPRESSIONE2>, <ESPRESSIONE3>, ...)*

Crea un oggetto con elementi array indicati dai parametri dell'espressione. Gli elementi dell'array sono numerati con un indice di base 1.

### AVERAGE

Elementi medi dell'array: *AVERAGE(<ARRAY>)*

Restituisce il valore medio degli elementi dell'array.

### EQUAL

Confronto di ciascun elemento degli array: *EQUAL(<ARRAY>, <ARRAY>)*

Confronta ciascun elemento di due array per determinare se gli array contengono gli stessi elementi. Se le dimensioni dei due array sono differenti o se non vi è

corrispondenza tra due elementi di due array, la funzione restituisce il valore zero 0. Altrimenti, viene restituito il valore 1.

## **LEN**

Conteggio degli elementi di array: *LEN(<ARRAY>)*

restituisce il numero di elementi dell'array.

## **MAX**

Elemento maggiore dell'array: *MAX(<ARRAY>)*

Restituisce l'elemento più grande nell'array. Gli elementi dell'array vengono confrontati numericamente o alfabeticamente.

## **MIN**

Elemento minore dell'array: *MIN(<ARRAY>)*

Restituisce l'elemento più piccolo nell'array. Gli elementi dell'array vengono confrontati numericamente o alfabeticamente.

## **SUM**

Somma gli elementi dell'array: *SUM(<ARRAY>)*

restituisce la somma degli elementi dell'array.

## **Funzioni varie**

### **ARCSEGMENTENDINDEX**

Questa funzione restituisce il numero indice del punto finale di un segmento di arco specificato ricavato da una scansione: *ARCSEGMENTENDINDEX(<ID>, <indice>, <tol1>, <tol2>)*

*<ID>* - Il primo parametro è una stringa con l'ID della scansione in cui questa funzione estrae il numero di indice del punto finale per l'arco. Questo parametro può essere l'ID tra virgolette oppure qualsiasi espressione che quando deve essere una stringa diventa l'ID di una scansione.

*<indice>* - Il secondo parametro è il numero indice per l'arco dal quale si desidera ottenere il numero del punto finale. È un valore su base uno. Ad esempio, il numero di

indice dell'arco è 3 se si desidera il numero del punto finale per il terzo arco nella scansione.

**<tol1>** - Il terzo parametro è la tolleranza generale dell'elemento. È un errore massimo di forma utilizzato per dividere la scansione in linee ed archi.

**<tol2>** - Il quarto parametro è la tolleranza massima. Di solito, questa tolleranza stringente viene utilizzata per rilasciare i punti da una delle estremità dell'elemento finché l'errore di forma del segmento non si trova all'interno di questa tolleranza. Una volta ottenuti gli indice iniziale e finale per un arco, è possibile utilizzare tali punti all'interno di un elemento costruito per generare un elemento arco separato. Per un esempio simile, vedere "Esempio di un elemento linea creato da un segmento di scansione".

## ARCSEGMENTSTARTINDEX

Questa funzione restituisce il numero indice del punto iniziale del segmento di un arco specificato ricavato da una scansione: **ARCSEGMENTSTARTINDEX(<ID>, <indice>, <tol1>, <tol2>)**.

**<ID>** - Il primo parametro è una stringa con l'ID della scansione in cui questa funzione estrae il numero di indice del punto iniziale dell'arco. Questo parametro può essere l'ID tra virgolette oppure qualsiasi espressione che quando deve essere una stringa diventa l'ID di una scansione.

**<indice>** - Il secondo parametro è il numero indice dell'arco da cui si desidera ottenere il numero del punto iniziale. È un valore su base uno. Ad esempio, il numero di indice dell'arco è 3 se si desidera il numero del punto iniziale per il terzo arco della scansione.

**<tol1>** - Il terzo parametro è la tolleranza generale dell'elemento. È un errore massimo di forma utilizzato per dividere la scansione in linee ed archi.

**<tol2>** - Il quarto parametro è la tolleranza massima. Di solito, questa tolleranza stringente viene utilizzata per rilasciare i punti da una delle estremità dell'elemento finché l'errore di forma del segmento non si trova all'interno di questa tolleranza.

Esistono due parametri aggiuntivi che controllano se è possibile accettare un segmento arco identificato. Possono essere modificati solo con l'Editor impostazioni di PC-DMIS.

Qualsiasi segmento di arco con un raggio minore del valore della voce

**MinimumArcSegmentRadiusInMM** viene rifiutato. Il valore predefinito di questo parametro è 2 mm. Allo stesso modo, viene rifiutato anche qualsiasi segmento di arco con un raggio maggiore del valore della voce **MaximumArcSegmentRadiusInMM**. Il valore predefinito per questo parametro è 2000 mm (di solito non è necessario modificarlo).

Una volta ottenuti gli indice iniziale e finale per un arco, è possibile utilizzare tali punti all'interno di un elemento costruito per generare un elemento arco separato. Per un esempio simile, vedere "Esempio di un elemento linea creato da un segmento di scansione".

## **EOF e EOL**

Per informazioni su queste funzioni, vedere "Controllo di fine file o fine linea" nel capitolo "Uso di input/output di file".

## **FUNCTION**

Crea una funzione: `FUNCTION((<PARAM1>, <PARAM2>...), <ESPRESSIONE>)`

Crea una funzione che utilizza il numero di parametri indicato dall'elenco di parametri e li sostituisce nell'espressione.

- Quando si utilizza la parola chiave `FUNCTION`, l'elenco dei parametri rappresenta il primo elemento della funzione.
- L'elenco contiene i nomi dei parametri separati da virgole.
- L'elenco dei parametri è racchiuso tra parentesi.
- Il secondo elemento è rappresentato dall'espressione.
- L'espressione contiene i nomi dei parametri che devono essere sostituiti quando la funzione viene richiamata.

Per un esempio vedere l'argomento "Esempio di funzione generica".

## **GETROTABDATA**

Questa funzione restituisce i valori di centro, posizione angolare e vettori della tavola rotante specificata

`GETROTABDATA(<PARAMETRO>[, <TAVOLA>])`

La funzione restituisce i valori delle seguenti configurazioni:

- Tavola rotante singola
- Due tavole rotanti indipendenti
- Tavole rotanti sovrapposte

I dati che la funzione restituisce corrispondono ai dati nella finestra di dialogo

**Impostazione tavola rotante (Modifica | Preferenze | Impostazione tavola rotante).**

Per ulteriori informazioni sull'uso di questa finestra di dialogo, vedere "Definizione della tavola rotante".

## **CENTRO**

- "CENTER" - Restituisce i valori XYZ della posizione del centro.

- "CENTER","V" - Restituisce i valori XYZ della posizione del centro della tavola rotante V nella configurazione con due tavole o con tavole sovrapposte.
- "CENTER","W" - Restituisce i valori XYZ della posizione del centro della tavola rotante W nella configurazione con due tavole o con tavole sovrapposte.

Esempi:

<code>ASSEGNA/V1=GETROTABDATA ("CENTER")</code>	V1 è impostato sul valore attuale XYZ del centro della tavola rotante.
<code>ASSIGN/V1=GETROTABDATA ("CENTER", "V")</code>	V1 è impostato sul valore attuale XYZ del centro della tavola rotante V.
<code>ASSIGN/V1=GETROTABDATA ("CENTER", "W")</code>	V1 è impostato sul valore attuale XYZ del centro della tavola rotante W.

## POSIZIONE ANGOLARE

- "ANGLE" - Restituisce la posizione angolare della tavola rotante.
- "ANGLE","V" - Restituisce la posizione angolare della tavola rotante V nella configurazione con due tavole o con tavole sovrapposte.
- "ANGLE","W" - Restituisce la posizione angolare della tavola rotante W nella configurazione con due tavole o con tavole sovrapposte.

Esempi:

<code>ASSEGNA/V2=GETROTABDATA ("ANGLE")</code>	V2 è impostata sulla posizione angolare della tavola rotante
<code>ASSEGNA/V2=GETROTABDATA ("ANGLE", "V")</code>	V2 è impostata sulla posizione angolare della tavola rotante V.
<code>ASSEGNA/V2=GETROTABDATA ("ANGLE", "W")</code>	V2 è impostata sulla posizione angolare della tavola rotante W.



## VETTORE

- "VECTOR" - Restituisce il vettore IJK della tavola rotante.
- "VECTOR","V" - Restituisce il vettore IJK della tavola rotante V nella configurazione con due tavole o con tavole sovrapposte.
- "VECTOR","W" - Restituisce il vettore IJK della tavola rotante W nella configurazione con due tavole o con tavole sovrapposte.

Esempi:

<code>ASSEGNA/V3=GETROTABDATA ("VECTOR")</code>	V3 è impostata sul vettore IJK della tavola rotante.
<code>ASSEGNA/V3=GETROTABDATA ("VECTOR", "V")</code>	V3 è impostata sul vettore IJK della tavola rotante V.
<code>ASSEGNA/V3=GETROTABDATA ("VECTOR", "W")</code>	V3 è impostata sul vettore IJK della tavola rotante W.



L'argomento [TAVOLA] è facoltativo. Se non si specifica la tavola V o W, PC-DMIS si comporta in uno dei due modi seguenti.

- Se si usa una configurazione con una tavola singola o con tavole sovrapposte, restituisce i valori della tavola rotante W.
- Se si usa una configurazione con due tavole, restituisce i valori della tavola rotante attivata per prima nella barra degli strumenti **Tavola rotante attiva**. Per ulteriori informazioni sulla barra degli strumenti, vedere "Barra degli strumenti Tavola rotante attiva".

PC-DMIS ha due definizioni interne della tavole per le configurazioni con due tavole e tavole sovrapposte. Nelle configurazioni con una sola tavola, la definizione della seconda tavola non viene usata. Poiché questa esiste internamente, non si verificherà un errore se si specifica la tavola V in una configurazione con una sola tavola; tuttavia non si consiglia di procedere in questo modo. I valori restituiti dalla funzione in generale non saranno utili poiché la tavola non esiste realmente.

**IF**

Valutazione dell'espressione condizionale: IF(<EXPRESSION1>, <EXPRESSION2>, <EXPRESSION3>)

Se l'espressione 1 viene valutata come vera, ovvero con un valore diverso da zero, questa funzione restituisce il valore dell'espressione 2; altrimenti restituisce il valore dell'espressione 3.

**ISIOCHANNELSET**

Questa espressione usa due parametri. Il primo parametro indica quale canale di I/O sarà controllato (l'intervallo di numeri disponibile si basa sulla macchina da utilizzare). Il secondo parametro determina se il software eseguirà la query della macchina con Braccio1 o Braccio2. Se il secondo parametro è impostato su 1, eseguirà la query del controller del Braccio2. Se il secondo parametro non è presente (oppure è impostato su zero), il canale di IO eseguirà la query del controller del Braccio1. Il controller del Braccio1 è l'unica opzione se non è attiva la modalità bracci multipli.



Se il tipo di dati del tastatore, l'ID della punta, il nome del file del tastatore o il numero di canale forniti non sono validi, l'espressione restituisce 0.

**Esempio:**

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

V4 viene valutata come 1 (vera) se il canale è impostato; in caso contrario, viene valutata come 0 (falsa).

**LINESEGMENTENDINDEX**

Questa funzione restituisce il numero indice del punto finale del segmento di una linea specificato da una scansione: LINESEGMENTENDINDEX(<ID>, <indice>, <tol1>, <tol2>).

<ID> - Il primo parametro è un valore della stringa dell'ID della scansione in cui questa funzione estrae il numero di indice del punto finale per il segmento linea. Questo parametro può essere l'ID tra virgolette oppure qualsiasi espressione che quando deve essere una stringa diventa l'ID di una scansione.

<indice> - Il secondo parametro è il numero indice del segmento linea da cui si desidera ottenere il numero del punto finale. È un valore su base uno. Ad esempio, il numero di indice del segmento linea è 3 se si desidera il numero del punto finale per la terza linea della scansione.

<tol1> - Il terzo parametro è la tolleranza generale dell'elemento. È un errore massimo di forma utilizzato per dividere la scansione in linee ed archi.

**<tol2>** - Il quarto parametro è la tolleranza massima. Di solito, questa tolleranza stringente viene utilizzata per rilasciare i punti da una delle estremità dell'elemento finché l'errore di forma del segmento non si trova all'interno di questa tolleranza. Una volta ottenuti gli indici iniziale e finale per un segmento linea, è possibile utilizzare quei punti all'interno di un elemento costruito per generare un elemento linea separato. Per un esempio, vedere "Esempio di un elemento linea creato da un segmento di scansione".

### LINESEGMENTSTARTINDEX

Restituisce il numero indice del punto iniziale del segmento di una linea specificato da una scansione: LINESEGMENTSTARTIND(<ID>, <indice>, <tol1>, <tol2>).

**<ID>** - Il primo parametro è un valore della stringa dell'ID di scansione in cui questa funzione estrae il numero di indice del punto iniziale per il segmento linea. Può essere l'ID tra virgolette oppure qualsiasi espressione che quando deve essere una stringa diventa l'ID di una scansione.

**<indice>** - Il secondo parametro è il numero indice del segmento linea da cui estrarre il numero del punto iniziale. È un valore su base uno. Ad esempio, il numero indice del segmento di linea è 3 se si desidera il numero del punto iniziale per la terza linea della scansione.

**<tol1>** - Il terzo parametro è la tolleranza generale dell'elemento. È un errore massimo di forma utilizzato per dividere la scansione in linee ed archi.

**<tol2>** - Il quarto parametro è la tolleranza massima. Di solito, questa tolleranza stringente viene utilizzata per rilasciare i punti da una delle estremità dell'elemento finché l'errore di forma del segmento non si trova all'interno di questa tolleranza. C'è un ulteriore parametro che controlla se un segmento linea identificato in una scansione è accettabile. Può essere modificato solo con l'Editor impostazioni PC-DMIS. Qualsiasi segmento di una linea di lunghezza inferiore al valore della voce `MinimumLineSegmentLengthInMM` viene rifiutato. Il valore predefinito di questo parametro è 2 mm.

Una volta ottenuti gli indici iniziale e finale per un segmento linea, è possibile utilizzare quei punti all'interno di un elemento costruito per generare un elemento linea separato. Vedere "Esempio di un elemento linea creato da un segmento scansione" per un esempio.

### PROBEDATA

Questa funzione restituisce i dati relativi al tastatore attuale o specificato:

PROBEDATA(<OPTPROBEDATATYPE>, <OPTTIPID>, <OPTPROBEFILENAME>)

Questa funzione accetta fino a tre parametri facoltativi. Se si usano più parametri, basta separarli con una virgola. Non occorre inserire virgole tra parametri vuoti. Ad esempio, per ottenere il diametro del tastatore in uso, basta usare

`ASSEGNA/V1=PROBEDATA ("DIAM")`.

**OPTPROBEDATATYPE** - Parametro facoltativo che consente di specificare il tipo di dati del tastatore restituiti dall'espressione. Se questo parametro non viene specificato,

viene restituito l'ID della punta corrente. Si tratta di un parametro di tipo stringa. È possibile inserire nella posizione della prima espressione tutte le espressioni valutate come stringa valida. Vengono di seguito riportate le espressioni di tipo stringa valide per il primo parametro, per le quali non è necessario rispettare le maiuscole e le minuscole. Si tratta di espressioni di tipo stringa che devono essere racchiuse tra virgolette:

**"A"** - È l'angolo A della punta. Restituisce un valore in doppia precisione.

**"B"** - È l'angolo B della punta. Restituisce un valore in doppia precisione.

**"C"** - È l'angolo C della testata di un testa CW43 light. Restituisce un intero.

**"Date"** - È la data dell'ultima qualifica della punta. Restituisce un valore di tipo stringa.

**"Diam"** o **"Diameter"** È il diametro misurato della punta. Le prime quattro lettere "Diam" sono obbligatorie, ma è possibile includere anche più lettere fino ad avere il nome completo "Diameter". Restituisce un valore in doppia precisione.

**"ID"** - È l'ID della punta. Parametro predefinito. Restituisce un valore di tipo stringa.

**"Offset"** - È lo scostamento X,Y,Z misurato della punta. Viene restituito un valore di tipo punto.

**"PrbRdv"** - È la deviazione radiale del tastatore. Restituisce un valore in doppia precisione.

**"Rotation"** - È la rotazione espressa in radianti intorno alla punta del vettore. Restituisce un valore in doppia precisione.

**"Standarddeviation"** - È la deviazione radiale del tastatore. Restituisce un valore in doppia precisione.

**"Thick"** o **"Thickness"** - È lo spessore misurato della punta. Oltre alle prime cinque lettere obbligatorie "Thick", è possibile aggiungerne altre fino ad avere il nome completo "Thickness". Restituisce un valore in doppia precisione.

**"Time"** - È l'ora dell'ultima qualificazione della punta. Restituisce un valore di tipo stringa.

**"Vector"** - È il vettore della punta. Viene restituito un valore di tipo punto.



L'aggiunta di una **"T"** davanti a **"Diameter"**, **"Offset"**, o **"Thickness"** restituisce le informazioni teoriche (ad esempio, **TDIAMETER**, **TOFFSET**, e **TTHICKNESS**).

**OPTTIPID** - Questo parametro facoltativo specifica la punta da utilizzare durante l'ottenimento dei dati del tastatore specificati nella prima espressione. Se questo parametro non viene specificato, viene utilizzata la punta corrente. È opportuno utilizzare dati di tipo stringa.

**OPTPROBEFILENAME** - Questo parametro facoltativo specifica il nome del file del tastatore da usare per ottenere i dati del tastatore. Se questo parametro non viene specificato, sarà utilizzato il file del tastatore corrente.

### Esempi:

<code>ASSEGNA/V1=PROBEDATA()</code>	V1 viene impostata sull'ID della punta corrente, ovvero "T1A0B0".
<code>ASSEGNA/V2=PROBEDATA("TOFFSET", "T1A45B0")</code>	V2 viene impostata sullo scostamento teorico del tastatore per la punta T1A45B0.
<code>ASSEGNA/V3=PROBEDATA("Data", "T1A90B90", "TAST")</code>	V3 viene impostata su una stringa che rappresenta la data dell'ultima calibrazione della punta T1A90B90 del file del tastatore TAST.

### QUALTOOLDATA

Questa funzione restituisce i dati relativi all'utensile di calibrazione attuale o specificato. La sintassi di questa funzione è la seguente:

`QUALTOOLDATA(<TOOLINFO>, <TOOLID>, <FACENUMBER>)`

Questa funzione accetta fino a tre parametri. È necessario che almeno un parametro restituisca dei dati:

Il primo parametro, <TOOLINFO>, è una *stringa* che specifica il tipo di informazione da restituire, riguardo all'utensile di calibrazione. Se questo parametro non viene passato, la funzione restituisce il nome dell'utensile corrente o specificato.

- **"CTE"** o **"COEFFICIENTOF THERMALEXPANSION"** - L'una o l'altra di queste stringhe restituisce il valore del coefficiente di dilatazione termica in doppia precisione.
- **"DIAM"** - Questa stringa restituisce il diametro dell'utensile come valore in doppia precisione.
- **"ID"** - Questa stringa restituisce il nome dell'utensile come stringa.
- **"LENGTH"** - Questa stringa funziona come "DIAM". Inoltre, restituisce il diametro dell'utensile in doppia precisione.
- **"OVERRIDEIJK"** - Questa stringa restituisce il vettore IJK di sovrascrittura ricerca come valore di un punto.

- **"POLYDIAM"** - Questa stringa restituisce il valore in doppia precisione del diametro della faccia poliedrica specificata.
- **"POLYIJK"** - Questa stringa restituisce il vettore IJK della faccia poliedrica specificata come valore di un punto.
- **"POLYXYZ"** - Questa stringa restituisce il centro XYZ della faccia poliedrica specificata come valore di un punto.
- **"SHANKIJK"** - Questa stringa restituisce il vettore IJK dello stelo come valore di un punto.
- **"TYPE"** - Questa stringa restituisce il tipo di utensile come valore intero nel modo seguente: 0 per una sfera, 1 per una sfera con braccio2, 2 per un poliedro, 3 per un poliedro con braccio2.
- **"WIDTH"** - Questo parametro non è più utilizzato.
- **"XYZ"** - Questa stringa restituisce la posizione XYZ dell'utensile come valore di un punto.

Il secondo parametro, *<TOOLID>*, è una *stringa* che specifica il nome dell'utensile di calibrazione per il quale si chiedono informazioni. Se questo parametro non viene inviato, PC-DMIS presume che si vogliano informazioni riguardo all'utensile di calibrazione in uso. La stringa contiene indifferentemente maiuscole e minuscole.

Il terzo parametro, *<FACENUMBER>*, serve solo se si lavora con un utensile di calibrazione a forma di poliedro e solo se il primo parametro è "POLIXYZ", "POLIJK" o "POLIDIAM". È un valore intero che specifica la faccia dell'utensile a forma di poliedro da usare per ottenere dati.



L'utensile di calibrazione non ha un'impostazione generale che si applica automaticamente a tutti i tastatori in una routine di misurazione. Quando si calibra un tastatore per la prima volta, occorre scegliere l'utensile di calibrazione da usare. PC-DMIS salva queste informazioni sull'utensile di calibrazione per ogni tastatore. Quando si ricalibra lo stesso tastatore, PC-DMIS utilizzerà lo stesso utensile.

### Esempi:

ASSEGNA/VDIAM=QUALTOOLDATA("DIAM","SFERA_1_IN")	Assegna alla variabile VDIAM il diametro dell'utensile SFERA_1_IN.
ASSEGNA/VID = QUALTOOLDATA("ID")	Assegna alla variabile VID

	il nome dell'utensile corrente.
ASSEGNA/VTYPE=QUALTOOLDATA ("TYPE")	Assegna alla variabile VTYPE il tipo dell'utensile corrente.
ASSEGNA/VPOLYDIAM=QUALTOOLDATA ("POLYDIAM", "POLYTEST", 3)	Assegna alla variabile VPOLYDIAM il diametro della faccia 3 dell'utensile poliedrico, POLYTEST.

## SETROTABDATA

Questa funzione imposta il centro o il vettore del nuovo input:  
 SETROTABDATA(<PARAMETRO>,<nuovo valore>[,<TAVOLA>])  
 . Funziona nelle seguenti configurazioni:

- Tavola rotante singola
- Due tavole rotanti indipendenti
- Tavole rotanti sovrapposte

## CENTRO

- "CENTER",<NewValue> - Imposta i nuovi valori di XYZ del centro della tavola rotante.
- "CENTER",<NewValue>,"V" - Imposta i nuovi valori di XYZ del centro della tavola rotante V nella configurazione con due tavole o con tavole sovrapposte.
- "CENTER",<NewValue>,"W" - Imposta i nuovi valori di XYZ del centro della tavola rotante V nella configurazione con due tavole o con tavole sovrapposte.

## Esempi:

ASSEGNA/V1=SETROTABDATA ("CENTER",NewValue)	V1 è un codice di restituzione (1 =
---	-------------------------------------

	successo, 0 = insuccesso)
ASSEGNA/V1=SETROTABDATA ("CENTER",NewValue,"V")	V1 è un codice di restituzione (1 = successo, 0 = insuccesso)
ASSEGNA/V1=SETROTABDATA ("CENTER",NewValue,"W")	V1 è un codice di restituzione (1 = successo, 0 = insuccesso)

## VETTORE

- "VECTOR",<NewValue> - Imposta i nuovi valori del vettore IJK della tavola rotante.
- "VECTOR",<NewValue>,"V" - Imposta i nuovi valori del vettore IJK della tavola rotante V nella configurazione con due tavole o con tavole sovrapposte.
- "VECTOR",<NewValue>,"W" - Imposta i nuovi valori del vettore IJK della tavola rotante W nella configurazione con due tavole o con tavole sovrapposte.

## Esempi:

ASSEGNA/V2=SETROTABDATA ("VECTOR",NewValue)	V2 è un codice di restituzione (1 = successo, 0 = insuccesso)
ASSEGNA/V2=SETROTABDATA ("VECTOR",NewValue,"V")	V2 è un codice di restituzione (1 = successo, 0 = insuccesso)
ASSEGNA/V2=SETROTABDATA ("VECTOR",NewValue,"W")	V2 è un codice di restituzione (1 = successo, 0 = insuccesso)





L'argomento [TAVOLA] è facoltativo. Se non si specifica la tavola V o W, PC-DMIS si comporta in uno dei due modi seguenti.

- Se si usa una configurazione con una tavola singola o con tavole sovrapposte, imposta il nuovo valore della tavola rotante W.
- Se si usa una configurazione con due tavole, imposta i valori della tavola rotante attivata per prima nella barra degli strumenti **Tavola rotante attiva**. Per ulteriori informazioni sulla barra degli strumenti, vedere "Barra degli strumenti Tavola rotante attiva".

PC-DMIS ha due definizioni interne della tavole per le configurazioni con due tavole e tavole sovrapposte. Nelle configurazioni con una sola tavola, la definizione della seconda tavola non viene usata. Poiché questa esiste internamente, non si verificherà un errore se si specifica la tavola V in una configurazione con una sola tavola; tuttavia non si consiglia di procedere in questo modo. I valori impostati dalla funzione in generale non saranno utili poiché la tavola non esiste realmente.

## TUTORELEMENT

Questa funzione accetta un parametro che può essere un numero o una stringa (una stringa è rappresentata dall'ID di un elemento).

TUTORELEMENT(<PARAMETRO>)

La funzione è utilizzata insieme al tipo di variabile, *strutture*. Per informazioni sulla struttura e sugli elementi secondari vedere "Strutture".

### Esempi:

ASSEGNA/E=TUTORELEMENT (1)	Crea una Struttura Tutor Element
ASSEGNA/WM=TUTORELEMENT (n)	Per i numeri superiori a 1, crea un array di <i>n</i> strutture di elementi Tutor
ASSEGNA/CIR1E=TUTORELEMENT ("CIR1")	Copia i dati dall'elemento CER1 nelle strutture di elementi Tutor.

La struttura TutorElement presenta i seguenti elementi secondari:

Elemento secondario	Descrizione
ID	Stringa dell'ID dell'elemento
TIPO	INTERO (TIPOF)
X, Y, Z	Valori delle coordinate X, Y, Z
RP	Raggio polare
AP	Angolo polare
CX	I
CY	J
CZ	K
DM	Diametro 1
DM2	Diametro 2
DS	Distanza dall'origine
A	Angolo
AXY	Angolo nel piano XY
AYZ	Angolo nel piano YZ
AZX	Angolo nel piano ZX
F	Errore di forma
SDEV	Deviazione standard
TP	Posizione

## Esempi di funzione

Di seguito sono riportati esempi diversi di funzioni che possono aiutare a creare ed utilizzare funzioni personalizzate:

- Esempio di funzione generica
- Esempio di funzioni trasferite come variabili

## Uso di espressioni e di variabili

- Esempio di funzione contenente più parametri
- Esempio di funzioni derivanti da altre funzioni
- Esempio di funzioni definite come elementi di un array
- Esempio di funzioni definite in modo iterativo

### Esempio di funzione generica

```
ASSIGN/MYFUNC=FUNCTION ( (X,Y,Z) , X*3+Y*2+Z)
```

Crea una funzione definita dall'utente e l'assegna alla variabile MYFUNC. La funzione accetta tre parametri, X,Y e Z.

- X viene moltiplicato per 3.
- Y viene moltiplicato per 2.
- Z contiene il valore inviato.

Il totale di  $X + Y + Z$  è ciò che viene restituito quando in valori sono inoltrati a una funzione, come riportato di seguito:

```
ASSIGN/V1=MYFUNC (7,2,5)
```

Assegna a V1 il valore 30 valutando i parametri inoltrati alla funzione MYFUNC(7,2,5).

- 7 è il parametro e sostituisce tutte le ricorrenze di X presenti nella sezione dell'espressione relativa alla definizione della funzione. In questo modo,  $X*3$  diventa  $7*3$  o 21.
- 2 sostituisce le ricorrenze di Y. Pertanto  $Y*2$  diventa  $2*2$  o 4.
- 5 sostituisce le ricorrenze di Z.

I valori vengono quindi sommati ( $21 + 4 + 5$ ) e inviati a V1.

### Esempio di funzioni trasferite come variabili

È possibile trasferire le funzioni come variabili. Il seguente esempio di basa sull'esempio di funzione generica precedente:

```
ASSEGNA/NUOVAFUNZ=MIAFUNZ
```

Imposta la variabile NUOVAFUNZ in modo che corrisponda alla funzione MIAFUNZ.

```
ASSEGNA/V3 = NUOVAFUNZ (12, 2, 3)
```

Assegna a V3 il valore 43 dalle espressioni valutate nella funzione  $(36 + 4 + 3)$ .

### Esempio di funzione contenente più parametri

Le funzioni possono contenere più parametri.

```
ASSEGNA/AGGIUNGIERADDOPPIA=FUNCTION ( (A,B) , 2 * (A+B) )
```

Crea una funzione e la assegna alla funzione AGGIUNGIERADDOPPIA. I due parametri inclusi nella funzione vengono sommati ed il relativo risultato moltiplicato per 2.

```
ASSEGNA/V2=AGGIUNGIERADDOPPIA (4, 5)
```

Assegna a V2 il valore 18. I parametri 4 e 5 vengono sostituiti nella parte di espressione della funzione, diventando quindi  $2*(4+5)$ .

### Esempio di funzioni derivanti da altre funzioni

Le funzioni possono creare altre funzioni.

```
ASSEGNA/COMPONI=FUNCTION ( (F, G) , FUNCTION ( (X) , G (F (X)) ) )
```

Assegna COMPONI in modo che sia una funzione che utilizza due funzioni come parametri e crea una nuova funzione che utilizza le due funzioni.

```
ASSEGNA/AGGIUNGI2=FUNCTION ( (X) , X+2)
```

Assegna AGGIUNGI2 in modo che sia una funzione che aggiunge due ai parametri inoltrati.

```
ASSEGNA/AGGIUNGI3=FUNCTION ( (X) , X+3)
```

Assegna AGGIUNGI3 in modo che sia una funzione che aggiunge tre al parametro inoltrato.

```
ASSEGNA/AGGIUNGI5=COMPONI (AGGIUNGI2, AGGIUNGI3)
```

Assegna AGGIUNGI5 in modo che sia una funzione composta dalle funzioni AGGIUNGI2 e AGGIUNGI3.

```
ASSEGNA/V5=AGGIUNGI5 (3)
```

Assegna V5 in modo che abbia il valore V8.

### Esempio di funzioni definite come elementi di un array

Le funzioni possono essere definite come elementi di un array.

## Uso di espressioni e di variabili

```
ASSEGNA/UNARRAY=ARRAY(3, FATTORIALE, "Hello World", ADD5)
```

Assegna UNARRAYY in modo che sia un array di 4 elementi: un numero (3), una funzione (FATTORIALE), una stringa, ("Hello World") e una funzione (Add5).

```
ASSEGNA/V6=UNARRAY[2] (4)
```

La funzione FATTORIALE è il secondo elemento di UNARRAY. Il parametro 4 viene trasferito in questa funzione e il risultato (24) viene assegnato alla variabile V6.

```
ASSEGNA/V7=UNARRAY[2] (UNARRAY[4] (UNARRAYY[1]))
```

Dall'interno all'esterno: il primo elemento di UNARRAY (3) viene trasferito alla funzione del quarto elemento dell'array (Add5). Il risultato (8 ) viene trasferito alla funzione del secondo elemento dell'array (FATTORIALE) ed assegnato a V7. V7 riceve il valore 40320.

## Esempio di funzioni definite in modo iterativo

È possibile definire le funzioni in modo iterativo, ovvero autoreferenziale.

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

Crea una funzione detta fattoriale che utilizza un parametro. Se il parametro è minore o uguale a 1 restituirà 1, altrimenti restituirà X moltiplicato per FACTORIAL di X-1.

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

Assegna a V4 il valore 120 (5\*4\*3\*2\*1).

## Esempio di un elemento linea creato da un segmento scansione

In questo argomento viene fornito un esempio di utilizzo del linguaggio delle espressioni di PC-DMIS, in particolare per le funzioni del segmento linea, per esportare i numeri del punto di inizio e di fine per segmenti linea entro una scansione, quindi per creare un proprio elemento linea utilizzando i punti estratti all'interno di un elemento costruito. I principi illustrati in questo esempio possono essere utilizzati anche per creare un segmento arco da una scansione.

Si supponga che la routine di misurazione disponga di un elemento scansione denominato SCAN1 simile al seguente:



```

SCAN1 =ELEM/SCAN,APERTALINEAE,MOSTRA
PUNTI=NO,MOSTRATUTTIPARAM=SÌ
MOD ESEC=RIAPPRENDI, MOD NOM=TROVA
NOMINALI,PIANOSICUREZZA=NO,PUNTO SINGOLO=NO,SPESSORE=0
TROVANOMINALI=5,SOLOSELEZIONATI=NO,USABESTFIT=NO,COMPTA
STATORE=SÌ,MOVIMENTO DI SICUREZZA=NO,DISTANZA=0,CAD
Compensazione=NO
DIR1=VARIABILE,
TIPOPUNTO=VETTORE
VETTINIZ=0,-1,0
VETTDIR=1,0,0
PIANOTAGLIO=0,0,1
VETTFINE=0,-1,0
VETTPIANO=-1,0,0
PUNTO1=100,0,-5
PUNTO2=70,0,-5
MIS/SCAN
SCANSIONEBASE/LINEA,MOSTRAPUNTI=NO,MOSTRATUTTIPARAM=SÌ
<100,0,-5>,<70,0,-5>,VetTag=0,0,1,VetDir=1,0,0
VetIniz=0,-1,0,VetFine=0,-1,0,SPESSORE=0
FILTRO/FILTRONULLO,
MODALITÀ ESEC=RIAPPRENDI
BORDO/PIANO,<70,0,-5>,VetPian=-1,0,0,Incroci=2
TIPOPUNTO/VETTORE
MODALITÀ NOMINALI=TROVANOM,5
FINE SCAN
FINEMIS/

```

Per creare una riga da questa scansione sarà necessario utilizzare le funzioni LINESEGMENTSTARTINDEX e LINESEGMENTENDINDEX per estrarre i dati, nel seguente modo:




```

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

```

In questo modo si indica a PC-DMIS di individuare la scansione denominata "SCAN1" e dal primo segmento riga estrarre i valori dell'indice iniziale e finale compresi nelle tolleranze definite. Quindi, assegnare tali valori di indice alle variabili denominate INDICEINIZIORIGA e INDICEFINERIGA.

Una volta assegnati i valori dell'indice iniziale e finale per il segmento riga alle variabili, è possibile utilizzare tali variabili in una riga costruita, nel seguente modo:



```

LIN4 =ELEM/LINEA,RETT,ILLIM
      TEOR/100.225,0,-5.011,1,0,0
      REALE/100.225,-0.005,-5.011,1,-0.0000388,0
      CONSTR/LINE,BF,2D,SCN1.HIT[LINESTARTINDEX..LINEENDINDEX],,
      RIMOZ_PUNTI_ISOLATI/OFF,3
      FILTRO/OFF,LUNGHEZZAONDA=0
    
```

Nel codice evidenziato nell'elemento riga sopra riportato, PC-DMIS utilizza i numeri iniziale e finale estratti dalla scansione per creare l'elemento:

`SCN1.HIT[LINESTARTINDEX..LINEENDINDEX]`

## Forzatura degli operandi

È possibile eseguire la forzatura degli operandi convertendoli in tipi diversi, mediante l'uso degli operatori di forzatura:

### Forzatura nel tipo intero

`INT(<espressione>)` - Forza il valore dell'espressione nel tipo intero

<code>INT(4)</code>	L'espressione viene valutata come 4.
<code>INT(4.5)</code>	L'espressione viene valutata come 4.
<code>INT("Hello World")</code>	L'espressione viene valutata come 0.
<code>INT("2")</code>	L'espressione viene valutata come 2.
<code>INT("2.2")</code>	L'espressione viene valutata come 2.
<code>INT("3 Blind Mice")</code>	L'espressione viene valutata come 3.
<code>INT("The 3 Blind Mice")</code>	L'espressione viene valutata come 0.
<code>INT("3, 4, 5")</code>	L'espressione viene valutata come 3.
<code>INT(MPOINT(0, 0, 1))</code>	L'espressione viene valutata come distanza del punto dall'origine. In questo caso, viene valutata come 1.
<code>INT(MPOINT(3, 4, 5))</code>	La distanza viene valutata 7.0711. Questa espressione viene valutata come 7.

## Forzatura in doppia precisione

**DOUBLE(<Espressione>)** - Esegue la forzatura del valore dell'espressione sul tipo double

<code>DOUBLE (4)</code>	L'espressione viene valutata come 4.0.
<code>DOUBLE (4.5)</code>	L'espressione viene valutata come 4.5.
<code>DOUBLE ("A String")</code>	L'espressione viene valutata come 0,0.
<code>DOUBLE ("3.5")</code>	L'espressione viene valutata come 3,5.
<code>DOUBLE ("3.5 pollici")</code>	L'espressione viene valutata come 3.5.
<code>DOUBLE ("Il diametro del cerchio misura 3,5 pollici")</code>	L'espressione viene valutata come 0.0.
<code>DOUBLE (MPOINT (0, 0, 1))</code>	L'espressione viene valutata come 1.0.
<code>DOUBLE (MPOINT (3, 4, 5))</code>	L'espressione viene valutata come 7.0711

## Forzatura nel tipo stringa

**STR(<espressione>)** - Forza il valore dell'espressione sul tipo stringa

<code>STR(4)</code>	L'espressione viene valutata come "4".
<code>STR(4.5)</code>	L'espressione viene valutata come "4,5".
<code>STR("Hello World")</code>	L'espressione viene valutata come "Ciao".



STR(MPOINT(3,4,5))	L'espressione viene valutata come "3, 4, 5".
--------------------	--

## Forzatura nel tipo punto

MPOINT(<espressione1>, <espressione2>, <espressione3>) - Esegue la forzatura dei valori delle espressioni sul tipo punto dopo aver costretto ogni espressione al tipo double.

PUNTOM(1, 1, 1)	L'espressione viene valutata come punto 1,0 , 1,0 ,1,0.
PUNTOM(1.1, 1.1, 1.1)	L'espressione viene valutata come punto 1,1, 1,1, 1,1.
MPOINT("1", "1", "1")	L'espressione viene valutata come punto 1,0 , 1,0 ,1,0.
PUNTOM(3, 4.5, "5.6")	Viene valutata come punto 3.0, 4.5, 5.6
PUNTOM(PUNTOM(1, 0, 0), MPOINT(0,1,0), PUNTOM(3,4,5))	Viene valutata come 1.0, 1.0, 7.0711

## Espressioni di forzatura degli operandi e di tipo misto

Lo strumento di valutazione delle espressioni esegue automaticamente la forzatura delle variabili nelle espressioni di tipo misto. Se il risultato di un'espressione non è quello atteso a causa della forzatura automatica, in alcuni casi è possibile ottenere il risultato desiderato mediante l'uso degli operatori di forzatura. Vengono di seguito riportati alcuni esempi di forzatura automatica in espressioni di tipo misto.

### "CER" + 1

L'espressione viene valutata come "CER1".

### **"2" + 2**

L'espressione viene valutata come 4

### **"Il valore di 2+2 è " + 2 + 2**

L'espressione viene valutata come "Il valore di 2+2 è 4". Questo perché PC-DMIS valuta le espressioni da sinistra a destra, e la parte sinistra dell'espressione è una stringa.

### **"Il valore di 2+2 è " + (2 + 2)**

L'espressione viene valutata come "Il valore di 2+2 è 4"

### **LINEA1.XYZ > 2**

Viene valutata come 1 se la distanza del baricentro di LINEA1 dall'origine è maggiore di 2.

### **LINEA1.XYZ > LINEA2.XYZ**

L'espressione viene valutata come 1 se il baricentro di LINEA1 è maggiore della distanza dall'origine del baricentro della LINEA2.

### **LINEA1.XYZ = LINEA2.XYZ**

L'espressione viene valutata come 1 se i baricentri di LINEA1 e LINEA2 corrispondono, in questo caso, la forzatura non viene eseguita.

### **DOPPIO(LINEA1.XYZ) = DOPPIO(LINEA2.XYZ)**

L'espressione viene valutata come 1 se i baricentri si trovano alla stessa distanza dall'origine.

### **11% 3.1**

L'espressione viene valutata come 2 (% rappresenta l'operatore del modulo utilizzato con i numeri interi. Restituisce il resto di una divisione discreta.  $11\%3 = 2$ .)

### **CERCHIO1.PUNTO [3.2].X**

Viene valutata sul valore X misurato del terzo punto di Cerchio1. L'argomento 3.2 viene forzato automaticamente a un valore intero 3.

## **Espressioni con ID**

Molti comandi di PC-DMIS utilizzano come parametri gli identificativi (ID) degli elementi. Ad esempio, gli ID vengono utilizzati per indicare gli elementi da utilizzare come input negli elementi costruiti. Le espressioni con gli ID consentono all'utente di fare riferimento ad una specifica istanza di un elemento, a un gruppo di elementi con nomi simili, a un'istanza di un elemento in una chiamata ad una subroutine o ad un elemento di una routine di misurazione esterna.

## ID di array di elementi

Utilizzare un ID di array elementi per fare riferimento a una specifica istanza di un elemento o ad un intervallo di istanze dell'elemento. Ad esempio, se l'elemento "Cerchio1" viene posizionato in un ciclo WHILE eseguito cinque volte, all'uscita dal ciclo saranno state create cinque istanze del cerchio. Per fare riferimento ad una singola istanza di "Cerchio1", utilizzare la sintassi relativa agli array descritta in "Array di elementi:" dove "Cerchio1[1]" fa riferimento alla prima istanza, "Cerchio1[2]" alla seconda istanza e così via.

Per fare riferimento a un intervallo di istanze utilizzare la notazione ... "Cerchio1[1..3]" fa riferimento alle istanze di Cerchio1 comprese tra la 1° e la 3°. "Cerchio1[3..5]" fa riferimento alle istanze di Cerchio1 comprese tra la 3° e la 5°. "Cerchio1[1..5]" fa riferimento alle istanze di Cerchio1 comprese tra la 1° e la 5°. Quando si fa riferimento a un intervallo di elementi, l'insieme delle istanze viene considerato costruito e si comporta come tale.

## Caratteri jolly degli ID

Usare i caratteri jolly degli ID per fare riferimento ad un insieme di elementi dal nome simile. I due caratteri jolly sono "\*" e "?". (Per ulteriori informazioni, vedere "Selezione degli elementi mediante i metacaratteri" nel capitolo "Modifica della visualizzazione CAD").

Il carattere asterisco "\*" si riferisce a 0 o più istanze di un carattere qualsiasi. Per riferirsi all'insieme di tutti gli elementi che iniziano con le lettere "CER", usare l'espressione di ID "CER\*". Questa sintassi crea un insieme di tutti gli elementi i cui ID iniziano con "CER", ad esempio "CERCHIO1", "CERCHIO2", "CER3" o "CER".



Se CER3 viene eseguito più volte, viene usata solo la misura più recente. Per ottenere le varie istanze delle esecuzioni, è possibile usare la seguente espressione: CER?[1..3].

Il carattere punto interrogativo '?' si riferisce a una singola istanza di qualsiasi carattere.



L'espressione ID "MIO????1" crea un insieme di elementi costituiti da sei caratteri, che iniziano con "MIO" e terminano con "1", ad esempio "MIOCER1", "MIOCON1", "MIOLIN1" o "MIOFT21".

## **ID di elementi all'interno di subroutine, script in Basic o routine esterne**

Le subroutine possono trovarsi all'interno della routine di misurazione attuale o di una routine di misurazione esterna. Quando la subroutine si trova nella stessa routine come chiamata della subroutine, per fare riferimento alle singole istanze di un elemento creato nella subroutine è possibile usare la sintassi relativa all'ID degli array degli elementi descritta in "Array di elementi. Tuttavia, se la subroutine si trova in una routine di misurazione esterna, per fare riferimento agli elementi creati nella subroutine è possibile usare la seguente sintassi: "<ID Sub Call>:<ID elemento>".

Ad esempio, è possibile fare riferimento a un elemento denominato "F1" posizionato in una subroutine esterna chiamata da un comando Call Sub con l'ID "CS1", utilizzando l'espressione ID "CS1:F1".



Questo esempio ha il solo scopo di mostrare l'uso della sintassi CS1.F1 e non è destinato a un uso effettivo.

### Routine 1: PLUS1.PRG

```
SUBROUTINE/PLUS1, A1 = 0, A2 = 0, A3 = 0

F1 =ELEM/PUNTO,RETT

THEO /A1+1,A2+1,A3+1,0,0,1

REALE/3,1,1,0,0,1

MIS/PUNTO,1

PUNTO /BASE,A1+1,A2+1,A3+1,0,0,1,0,0,0

FINE_MIS/

ENDSUB/
```

### Routine 2: TEST.PRG

```
CS1 =CALLSUB/PLUS1,D:\V30\WINDEB\PLUS1.PRG: 3,3,3,,

DIM D1= POSIZIONE DEL PUNTO CS1:F1 UNITÁ=IN,$

GRAFICO=OFF TESTO=OFF MULT=10.00 OUTPUT=ENTRAMBI

ASSE NOMINALE +TOL -TOL MIS MAX MIN DEV OUTTOL

X 3.0000 0.0000 0.0000 3.0000 3.0000 3.0000 0.0000 0.0000

----#----

FINE DIMENSIONE D1
```

Gli script in Basic consentono di creare ed eliminare gli oggetti in modo dinamico. Per fare riferimento a un elemento creato mediante uno script in Basic, utilizzare la sintassi "<ID Basic Script>:<ID elemento>". Ad esempio, utilizzare l'espressione ID "BS1:F2" per fare riferimento a un elemento con ID "F2" creato mediante uno script in Basic con ID "BS1".

Per aggiungere routine esterne a PC-DMIS è possibile usare il comando di aggiunta. Per fare riferimento ad elementi contenuti nella routine aggiunta, usare la sintassi seguente: "<Attach Routine ID>:<Feat ID>". Pertanto, l'espressione "GEAR1:F3" consente di fare riferimento all'elemento "F3" della routine di misurazione con l'ID "GEAR1". (Per ulteriori informazioni vedere "Aggiunta di una routine di misurazione esterna" nel capitolo "Aggiunta di elementi esterni").

## Combinazione di espressioni con ID

È possibile combinare espressioni di ID di array, espressioni di ID con caratteri jolly e subroutine esterne, script Basic e espressioni di ID di routine di misurazione esterne. Ad esempio, per fare riferimento alla terza istanza di tutti gli elementi che iniziano con le lettere "CER" presenti in una routine di misurazione esterna aggiunta con l'ID "BOLTPAT", è possibile usare l'espressione dell'ID "`BOLTPAT:CER*[3]`".

Inoltre, è possibile usare le espressioni degli ID in espressioni normali. Quindi, il baricentro misurato dell'insieme di elementi sopra citato può essere assegnato alla variabile con la seguente espressione:

```
ASSEGNA/V1=BOLTPAT:CER*[3].XYZ
```

Inoltre, poiché è possibile utilizzare le espressioni ID in espressioni normali. Quindi, il baricentro misurato dell'insieme di elementi sopra citato può essere assegnato alla variabile con la seguente espressione:

```
ASSEGNA/V1=BOLTPAT:CER*[3].XYZ
```

---

## Accesso alle proprietà di un oggetto Rapporto

È possibile creare i propri modelli di etichette e rapporti personalizzati. PC-DMIS li usa per visualizzare i dati nel rapporto all'interno della finestra Rapporto (vedere **Visualizza | Finestra Rapporto**). Tali modelli vengono creati utilizzando gli editor dei modelli. Gli Editor si servono di un'interfaccia tipo Visual Basic che si può usare per inserire, riposizionare e dimensionare componenti speciali denominati "oggetti".

Ciascun oggetto è composto da "proprietà" che definiscono le modalità di visualizzazione e le informazioni che contiene. Alcune di tali proprietà sono in comune con tutti gli altri oggetti, altre sono in comune solo con gli oggetti correlati e altre sono univoche di un oggetto specifico.

Il linguaggio delle espressioni di PC-DMIS può eseguire una query sul rapporto caricato e memorizzare in una variabile i valori delle proprietà di un particolare oggetto. Può ottenere valori di tipo Stringa, Intero e Reale utilizzando la seguente sintassi:

## Sintassi di query per la proprietà



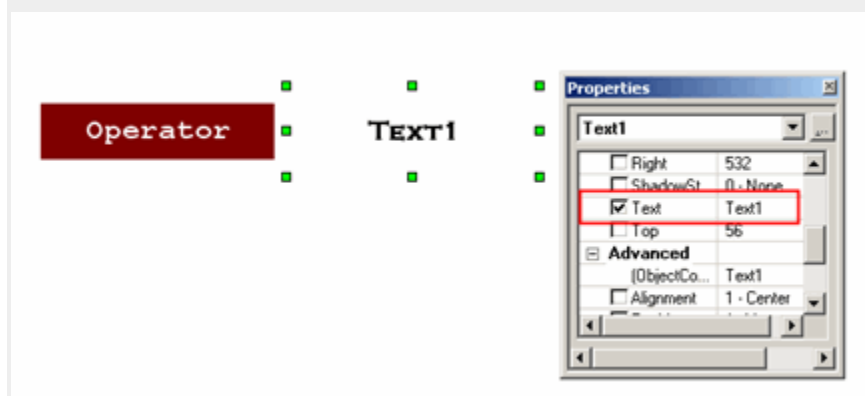
```
ASSEGNA/V1 = Rapporto.<Nome oggetto>.<Nome proprietà>
```

Rapporto indica il rapporto caricato. <Nome oggetto> è il nome univoco dell'oggetto. <Nome proprietà> è il nome di una proprietà dell'oggetto.

## Esempio



Si supponga, ad esempio, che il modello del rapporto contenga un oggetto testo denominato "Text1" che si desidera usare nel rapporto finale per visualizzare il nome dell'operatore. PC-DMIS memorizza la stringa effettiva di caratteri che rappresenta il nome dell'operatore nella proprietà **Testo** dell'oggetto. Per impostazione predefinita, la proprietà del testo (testo visualizzato) inizialmente ha il valore "Text1" (vedere la figura seguente). Poiché si tratta di una proprietà assegnata dall'utente, il valore cambia quando l'operatore immette il nome durante l'esecuzione.



*Finestra di dialogo Proprietà che mostra un oggetto selezionato e la proprietà di cui eseguire la query*

Per utilizzare il codice Expression Language per eseguire la query della proprietà "Testo" dell'oggetto Testo e ottenere i dati digitati, utilizzare il seguente comando:

```
ASSEGNA/V1 = Rapporto.Testo1.Text
```

In questo codice:

"Rapporto" indica di cercare nel rapporto caricato nella finestra Rapporto.

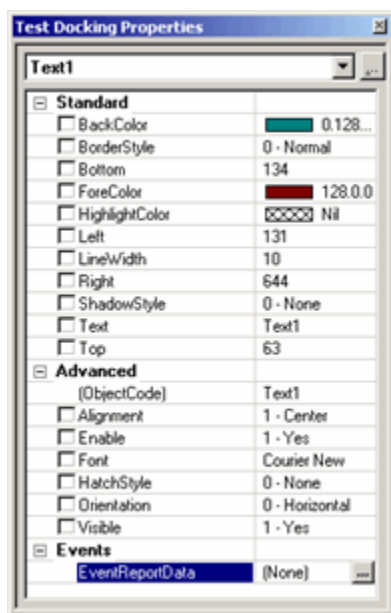
"Testo1" indica di cercare l'oggetto denominato "Testo1".

"Text" indica di cercare la proprietà "Text" nell'oggetto. Il valore della proprietà "Text" viene inviato tramite la variabile V1 che può essere manipolata o visualizzata con il linguaggio delle espressioni di PC-DMIS.



## Ricerca delle proprietà

Per trovare le proprietà associate a un determinato oggetto, accedere al modello del rapporto nell'Editor dei modelli dei rapporti (**File | Rapporto | Modifica | Modello rapporto**), selezionare l'oggetto e fare clic con il pulsante destro del mouse sull'oggetto per visualizzare la relativa pagina delle proprietà.



*La pagina delle proprietà di un oggetto Testo.*

La pagina delle proprietà contiene due colonne. La colonna sinistra mostra il nome della proprietà. La colonna destra ne mostra il valore attuale. Assicurarsi di utilizzare il nome esatto della proprietà nel codice dell'espressione.



Quando si esegue una query dei valori della proprietà, è possibile che alcune proprietà restituiscano valori numerici apparentemente inutili. In genere, ciò si verifica quando la proprietà dispone di una serie di opzioni disponibili. PC-DMIS restituisce un valore interno della proprietà selezionata che non è relativo alla proprietà visualizzata.

Per esempio, la proprietà **Orientation** dell'oggetto **Text** ha i seguenti valori:

- 0 - Orizzontale
- 1 - Verticale su
- 2 - Verticale giù

Tuttavia, se si ottiene il valore usando il linguaggio delle espressioni di PC-DMIS, il software restituirà quanto segue:

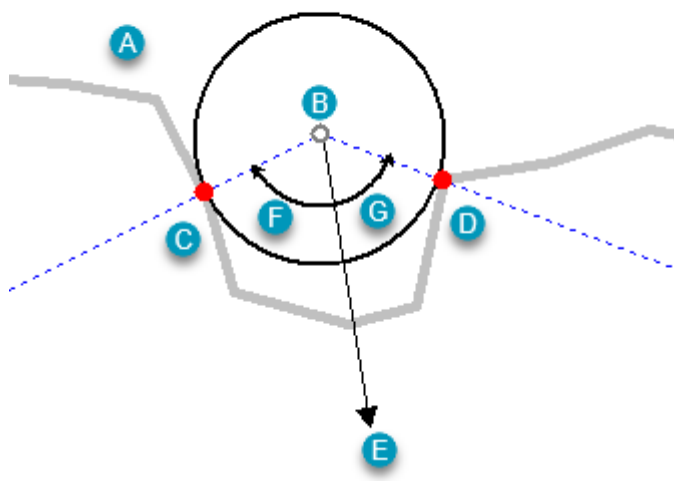
- 0 (Orizzontale)
- 900 (Verticale su)
- -900 (Verticale giù)

Questo può comportare diversi tentativi ed errori per determinare quale dei valori restituiti corrisponde al valore visualizzato sul foglio delle proprietà.

## Accesso alle informazioni da cerchio di scansione minimo costruito

È possibile usare le espressioni di PC-DMIS per estrarre informazioni da un elemento Cerchio costruito con un certo raggio in un punto minimo lungo una scansione lineare. Per informazioni più dettagliate, vedere l'argomento "Costruzione di un cerchio nel punto minimo di una scansione" nel capitolo "Costruzione di nuovi elementi a partire da elementi esistenti".

Quando si costruisce un elemento cerchio di una scansione minima, il cerchio utilizza un vettore (chiamato vettore inferiore) per raggiungere la linea di scansione. Raggiunge la linea in due soli punti chiamati punti di contatto (PUNTOCONTATTO1 e PUNTOCONTATTO2). PC-DMIS utilizzerà questi punti per determinare gli angoli dal vettore inferiore a questi punti di contatto (ANGOLOCONTATTO1 e ANGOLOCONTATTO2). Ad esempio, considerare il diagramma:



*A - La linea di scansione sulla quale viene costruito il cerchio.*

*B - La posizione XYZ finale del baricentro del cerchio.*

*C - Il punto di contatto alla sinistra del Vettore inferiore. Viene denominato CONTACTPOINT1.*

*D - Il punto di contatto alla destra del Vettore inferiore. Viene denominato CONTACTPOINT2.*

*E - Il Vettore inferiore.*

*F - L'angolo dal vettore inferiore a CONTACTPOINT1. Viene denominato CONTACTANGLE1.*

*G - L'angolo dal vettore inferiore a CONTACTPOINT2. Viene denominato CONTACTANGLE2.*

Le espressioni dettagliate di seguito funzionano solo con questo tipo di cerchio costruito. È possibile utilizzare anche CONTACTPOINT2 nella sintassi sotto riportata per restituire informazioni equivalenti utilizzando il secondo punto di contatto.

`ASSEGNA/V1=CER1.CONTACTPOINT1.XYZ`

Restituisce le informazioni sul punto XYZ relative al primo punto di contatto del cerchio con la linea, CONTACTPOINT1.

`ASSEGNA/V1=CIR1.CONTACTPOINT1.X`

Restituisce le informazioni X per CONTACTPOINT1.

`ASSEGNA/V1=CIR1.CONTACTPOINT1.Y`

Restituisce le informazioni Y per CONTACTPOINT1.

`ASSEGNA/V1=CIR1.CONTACTPOINT1.Z`

Restituisce le informazioni Z per CONTACTPOINT1.

`ASSEGNA/V1=CIR1.CONTACTPOINT1.IJK`

Restituisce il vettore IJK da CONTACTPOINT1 al centro del cerchio.

`ASSEGNA/V1=CIR1.CONTACTPOINT1.I`

Restituisce il valore I dal vettore precedente CONTACTPOINT1 IJK.

`ASSEGNA/V1=CIR1.CONTACTPOINT1.J`

Restituisce il valore J dal vettore precedente CONTACTPOINT1 IJK.

`ASSEGNA/V1=CIR1.CONTACTPOINT1.K`

Restituisce il valore K dal vettore CONTACTPOINT1 IJK.

`ASSEGNA/V1=CIR1.CONTACTANGLE1`

Restituisce l'angolo dal vettore inferiore a CONTACTANGLE1.

`ASSEGNA/V1=CIR1.CONTACTANGLE2`

Restituisce l'angolo dal vettore inferiore a CONTACTANGLE2.

`ASSEGNA/V1=CIR1.CONTACTANGLE`

Restituisce la somma dei valori assoluti di CONTACTANGLE1e CONTACTANGLE2.  
Non può superare i 180 gradi.