

Inhaltsverzeichnis

Verwenden von Ausdrücken und Variablen	1
Verwenden von Ausdrücken und Variablen: Einführung.....	1
Einsatz von Ausdrücken in einer Messroutine	2
Anzeigen von Ausdruckswerten	2
Nur Ausdruckswerte beibehalten	2
Verwenden von Ausdrücken mit Verzweigung.....	3
Verwenden von Ausdrücken mit Datei-Eingabe/-Ausgabe.....	3
Erstellen von Ausdrücken mit dem Ausdruckerstellungsprogramm.....	3
Erstellen eines Ausdrucks über die Tastatur	4
Erstellen eines Ausdrucks mit dem Ausdruckerstellungsprogramm.....	4
Prüfen der Korrektheit des Ausdrucks.....	5
Typ des Ausdruckelements	6
ID	6
Verlängerung.....	7
Zweite Erweiterung	9
Schaltfläche "Hinzufügen"	10
Bearbeitungsfeld	11
Bereich "Beschreibung"	11
Verwenden von Variablen mit Ausdrücken	11
Zuweisen von Werten zu Variablen mit Hilfe des Dialogfelds "Zuweisung"	12
Ausdruckskomponenten	13
Arten von Operanden.....	13

Literale	13
Verweise	14
Variablen	21
Strukturen	25
Zeiger	27
Arrays.....	29
Operatoren für Ausdrücke	42
Priorität.....	44
Funktionen	44
Erzwungene Umwandlung von Operanden.....	93
ID-Ausdrücke	96
Zugreifen auf die Objekteigenschaften eines Protokolls	99
Aufrufen von Informationen aus einem erstellten Kreisminimum eines Scans	103

Verwenden von Ausdrücken und Variablen

Verwenden von Ausdrücken und Variablen: Einführung

Ein Ausdruck ist eine benutzerdefinierte Bedingung, die in Verbindung mit PC-DMIS-Befehlen zur Ablaufsteuerung verwendet wird. Mit dem Einsatz von Ablaufsteuerungs-Anweisungen können Sie diese Bedingungen in Ihrer Messroutine testen. Sie können jeweils festlegen, welche Aktion in PC-DMIS stattfinden soll, wenn die Bedingung zutrifft bzw. nicht zutrifft.

Ausdrücke spielen eine wichtige Rolle, wenn Sie bestimmte Aufgaben durch PC-DMIS ausführen lassen möchten. Durch die Verwendung von Ausdrücken in Verbindung mit Befehlen zur Programmablaufsteuerung können Sie die leistungsstarken Funktionen von PC-DMIS noch besser nutzen.

In diesem Abschnitt wird erklärt, wie Ausdrücke innerhalb des Bearbeitungsfensters von PC-DMIS erstellt und angewendet werden. Beim Arbeiten mit Ausdrücken sollten Sie das Bearbeitungsfenster in den Befehlsmodus versetzen. Dadurch können Sie den Code im Bearbeitungsfenster direkt einsehen.

In diesem Abschnitt werden folgende Themen behandelt:

- Einsatz von Ausdrücken in einer Messroutine
- Erstellen von Ausdrücken mit dem Ausdruckerstellungsprogramm
- Verwenden von Variablen mit Ausdrücken
- Ausdruckskomponenten
- Zugreifen auf die Objekteigenschaften eines Protokolls
- Aufrufen von Informationen aus einem erstellten Kreisminimum eines Scans



Weitere Angaben zum Protokollieren von Ausdrücken finden Sie im Thema "Hinweise zu Protokollausdrücke" im Abschnitt "Protokollieren von Messergebnissen".

Einsatz von Ausdrücken in einer Messroutine

Im Bearbeitungsfenster von PC-DMIS können in den meisten editierbaren Feldern Ausdrücke verwendet werden. Bei editierbaren Feldern handelt es sich in der Regel um die Felder, die gelb markiert werden, wenn Sie im Befehlsmodus des Bearbeitungsfensters die Tabulatortaste drücken. In Feldern, die den Typ eines Elements ändern, sind keine Ausdrücke zulässig.



Im Feld eines Auto-Elements, das den Typ des Auto-Elements angibt (z.B. Flächenpunkt, Auto-Kreis, Auto-Langloch, usw.), sind keine Ausdrücke zulässig.

Die folgenden Unterthemen enthalten eine ausführliche Beschreibung der verfügbaren Ausdrücke.

Anzeigen von Ausdruckswerten

Wenn Sie den Wert eines Ausdrucks sehen möchten, setzen Sie den Mauszeiger mindestens eine Sekunde lang über den Ausdruck. Der Ausdruck wird berechnet, und unmittelbar unterhalb des Mauszeigers wird ein kleines gelbes Feld eingeblendet, das den Ausdruck und den ermittelten Wert anzeigt.

Nur Ausdruckswerte beibehalten

So lösen Sie einen Ausdruck im Bearbeitungsfenster sofort, so dass nur der Wert beibehalten wird:

1. Markieren Sie den Ausdruckstext im Bearbeitungsfenster.
2. Setzen Sie das Zeichen ` (accent grave) vor den Ausdruckstext.



Sie geben den Ausdruck ``1/7` in einem numerischen Feld ein. Der Ausdruck wird sofort gelöst, und nur der Wert (**0.143**) wird in das Feld übernommen.

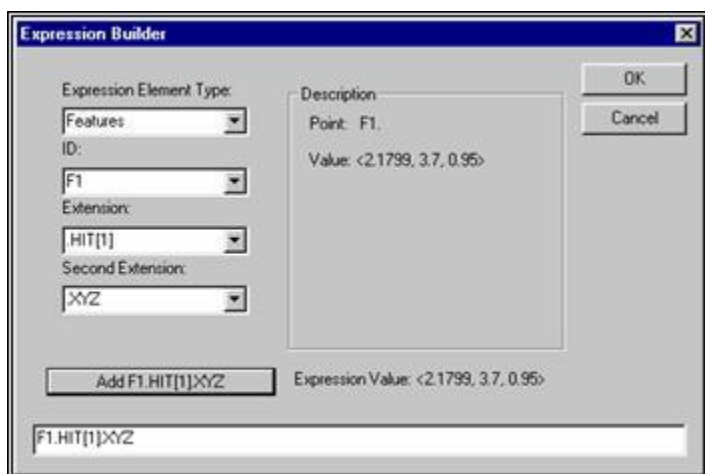
Verwenden von Ausdrücken mit Verzweigung

Die Befehle der Ablaufsteuerung verwenden Ausdrücke zur Bestimmung des Ablaufs der Ausführung der Messroutine. Unabhängig davon, ob die Verzweigung stattfindet oder nicht sollten Sie im Thema „Verzweigen mit Hilfe der Ablaufsteuerung“ nachschlagen.

Verwenden von Ausdrücken mit Datei-Eingabe/-Ausgabe

Beim Schreiben von Daten in eine externe Datendatei oder Lesen von Daten aus einer externen Datendatei werden häufig Variablen und andere Ausdrücke verwendet, um diese Daten effizient zu verwalten, zu speichern und anzuzeigen. Siehe auch unter "Verwenden der Dateieingabe / -ausgabe".

Erstellen von Ausdrücken mit dem Ausdruckerstellungsprogramm



Dialogfeld Ausdrucks-Generator

In PC-DMIS können Sie im Bearbeitungsfenster über die Tastatur Ausdrücke erstellen und hinzufügen, oder Sie können die komfortable Benutzeroberfläche des Dialogfelds **Ausdrucks-Generator** verwenden. Die Menüoption **Bearbeiten | Ausdruck** öffnet das Dialogfeld **Ausdrucks-Generator**.

In diesem Dialogfeld können Sie einen Ausdruck erstellen und in ein editierbares Feld einfügen. Das Dialogfeld **Ausdrucks-Generator** wird auch geöffnet, wenn der Cursor in einem Feld positioniert ist, in dem Ausdrücke zulässig sind, und Sie die Taste F2 drücken.

Das Dialogfeld **Ausdrucks-Generator** enthält alle Operatoren und Funktionen, die in Ausdrücken zulässig sind.

Erstellen eines Ausdrucks über die Tastatur

So erstellen Sie einen Ausdruck, indem Sie ihn über die Tastatur direkt in das Bearbeitungsfenster eingeben:

1. Öffnen Sie das Bearbeitungsfenster (**Ansicht | Bearbeitungsfenster**).
2. Versetzen Sie das Bearbeitungsfenster in den Befehlsmodus.
3. Drücken Sie die TABULATOR-TASTE, um den Cursor in ein editierbares Feld zu setzen, in das Sie den Ausdruck einfügen möchten. Bei den editierbaren Feldern handelt es sich um die gelb markierten Felder.
4. Geben Sie den Ausdruck ein.

Erstellen eines Ausdrucks mit dem Ausdruckerstellungsprogramm



Sie müssen sich im Befehlsmodus befinden, um die Ausdrucksoption nutzen zu können.

So geben Sie mit dem Dialogfeld Ausdrucks-Generator (**Bearbeiten | Ausdruck**) einen Ausdruck ein:

1. Öffnen Sie das Bearbeitungsfenster (**Ansicht | Bearbeitungsfenster**).
2. Versetzen Sie das Bearbeitungsfenster in den Befehlsmodus (**Ansicht | Befehlsmodus**).
3. Setzen Sie den Cursor in ein editierbares Feld, in das Sie den Ausdruck einfügen möchten.
4. Drücken Sie die Taste F2, wenn der Cursor in einem Feld positioniert ist, das Ausdrücke zulässt. Das Dialogfeld **Ausdrucks-Generator** wird angezeigt. Das Dialogfeld **Ausdrucks-Generator** enthält alle Typen von Operatoren, Operanden und Funktionen. Auf folgende Elemente kann über dieses Dialogfeld zugegriffen werden:
 - Verfügbare Ausdruckstypen

Erstellen von Ausdrücken mit dem Ausdruckerstellungsprogramm

- Variablen
 - Elemente
 - Merkmale
 - Ausrichtungen
 - Kommentare
5. Wählen Sie den Typ des Ausdruckelements aus der ersten Auswahlliste aus. Je nach ausgewähltem Typ werden unterschiedliche Kombinationsfelder angezeigt.
 6. Wählen Sie aus der Auswahlliste **ID** die gewünschte ID aus.
 7. Wählen Sie aus der Auswahlliste **Erweiterung** eine Erweiterung aus.
 8. Wählen Sie aus der Auswahlliste **Zweite Erweiterung** eine weitere Erweiterung aus. Wenn der Ausdruck eingesetzt werden kann, wird die Schaltfläche **Hinzufügen** verfügbar.
 9. Klicken Sie auf **Hinzufügen**. Der Ausdruck wird in einem Bearbeitungsfeld angezeigt.
 10. Klicken Sie auf **OK**. Der Ausdruck wird jetzt an der aktuellen Cursorposition im Bearbeitungsfenster angezeigt.



Sie können das Dialogfeld **Ausdruckerstellungsprogramm** auch über folgende Dialogfelder aufrufen:

- Das Dialogfeld **If-Ausdruck** - Wählen Sie die Option **Einfügen | Programmablaufsteuerungsbefehl | If Goto**. Klicken Sie auf die Schaltfläche **Ausdruck**.
- Über das Dialogfeld **Zuweisung** - Wählen Sie **Einfügen | Zuweisung**. Klicken Sie entweder auf die Schaltfläche **Zuweisen zu** oder **Zuweisen von**.

Sobald der Ausdruck erstellt wurde, fügt PC-DMIS ihn automatisch an der nächstmöglichen Position im Bearbeitungsfenster ein.

Prüfen der Korrektheit des Ausdrucks

Wenn der Cursor das Feld verlässt, in das ein Ausdruck eingegeben wurde, prüft PC-DMIS den Ausdruck auf Korrektheit. Wenn es im geprüften Ausdruck ein Problem gibt, wird eine Fehlermeldung mit dem Hinweis auf einen ungültigen Wert eingeblendet oder der Ausdruckstext wird rot angezeigt. Ausdrücke, die sich auf nicht existierende Objekte beziehen, werden ebenfalls rot angezeigt.

Da der Ausdruck beim Verlassen des Feldes auf seine Korrektheit geprüft wird, bleibt ein Feld, das aufgrund eines Bezugs auf ein nicht vorhandenes Objekt (z.B. KREIS1.X) rot dargestellt wurde, sogar dann noch rot, wenn das neue Objekt (z.B. KREIS1) hinzugefügt wurde. Das Feld wird so lange rot angezeigt, bis der Ausdruck erneut auf Korrektheit geprüft wird.

So prüfen Sie die Korrektheit des Ausdrucks erneut:

1. Bewegen Sie den Cursor in das Feld des Ausdrucks.
2. Drücken Sie die Taste F2. Das Dialogfeld **Ausdruckerstellungsprogramm** wird erneut aufgerufen. Die am Ausdruck vorgenommenen Änderungen werden im Bearbeitungsfeld angezeigt.
3. Klicken Sie auf die EINGABETASTE, um das Dialogfeld zu schließen.

Typ des Ausdruckelements

In der Auswahlliste Typ des Ausdruckelements des Dialogfeldes **Ausdrucks-Generator (Bearbeiten | Ausdruck)** werden die verschiedenen Typen von Ausdruckelementen aufgeführt, die in Ausdrücke eingefügt werden können. Dazu gehören:

- Funktionen
- Operatoren
- Ausrichtungen
- Kommentare
- Merkmale
- Elemente
- Variablen

ID

In der Auswahlliste **ID** des Dialogfeldes **Ausdrucks-Generator (Bearbeiten | Ausdruck)** werden die Elemente aufgeführt, die für den in der Auswahlliste **Typ des Ausdruckelements** ausgewählten Typ des Ausdruckelements verfügbar sind.



Die Liste der verfügbaren Elemente hängt von dem ausgewählten Ausdruckselement ab:

- Wenn in der Auswahlliste **Typ des Ausdruckelements** der Eintrag **Funktionen und Operatoren** ausgewählt wurde, enthält die Auswahlliste **ID** eine Liste der verfügbaren Funktionen und Operatoren.
- Wenn in der Auswahlliste **Typ des Ausdruckelements** der Eintrag **Elemente** ausgewählt wurde, enthält die Auswahlliste **ID** eine Liste der Elemente in der Messroutine.

Verlängerung

Die Auswahlliste **Erweiterung** im Dialogfeld **Ausdrucks-Generator (Bearbeiten | Ausdruck)** wird verfügbar, wenn dem in der Auswahlliste **ID** ausgewählten Element eine Erweiterung hinzugefügt werden muss, damit es als Ausdruckelement eingesetzt werden kann. In der Auswahlliste **Erweiterung** werden die Erweiterungen angezeigt, die für den in der Auswahlliste **ID** ausgewählten Eintrag verfügbar sind.



Angenommen, Sie haben in der Auswahlliste **ID** ein Element ausgewählt. PC-DMIS zeigt die Liste möglicher Erweiterungen, die zum Verweisen auf Daten dieses Elements eingesetzt werden können (z. B. "X", "Y", "Z", "Durchm", "Länge" etc.), werden daraufhin in der Auswahlliste **Erweiterung** aufgeführt.

Mögliche Erweiterungen beinhalten die folgenden gemessenen oder theoretischen Datentypen:

Istwert

- All – Alle Werte des Elements werden der Variablen zugeordnet. Siehe unten stehendes Beispiel.
- X – Gemessene X-Werte der Messpunkte
- Y – Gemessene Y-Werte der Messpunkte
- Z – Gemessene Z-Werte der Messpunkte
- XYZ – Gemessene XYZ-Werte der Messpunkte
- I – Gemessene I-Werte der Messpunkte
- J – Gemessene J-Werte der Messpunkte
- K – Gemessene K-Werte der Messpunkte
- IJK – Gemessene IJK-Werte der Messpunkte

Theo

- TX – Theoretische X-Werte der Messpunkte
- TY – Theoretische Y-Werte der Messpunkte
- TZ – Theoretische Z-Werte der Messpunkte
- TXYZ – Theoretische XYZ-Werte der Messpunkte
- TI – Theoretische I-Werte der Messpunkte
- TJ – Theoretische J-Werte der Messpunkte
- TK – Theoretische K-Werte der Messpunkte
- TIJK – Theoretische IJK-Werte der Messpunkte



Nehmen wir an, dieses Codeschnipsel ist Teil einer Messroutine:

```
F1=BENUTZERDEFINIERT/PUNKT,ABHÄNGIG,KARTESISCH,$
```

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

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

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

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

```
ZUWEISEN/MYFEATURE=F1.ALL
```

```
ZUWEISEN/V1=MYFEATURE.X
```

```
ZUWEISEN/V2=MYFEATURE.TX
```

Wenn die Messroutine den obigen Codeausschnitte abschließt, weist PC-DMIS den Variablen diese Werte zu:

```
V1 = 7.98
```

```
V2 = 8
```

Zweite Erweiterung

Die Auswahlliste **Zweite Erweiterung** wird nur dann verfügbar, wenn der in der Auswahlliste **Erweiterung** gewählte Eintrag den Zusatz einer zweiten Erweiterung erforderlich macht, um ein brauchbares Ausdruckselement zu formen.



Sie verweisen auf den Nennwert der X-Achsenlage eines Merkmals namens "D1". Setzen Sie folgendes Verfahren ein:

1. Wählen Sie **D1** aus der Auswahlliste **ID** aus.
2. Wählen Sie **X** aus der Auswahlliste **Erweiterung** aus.
3. Wählen Sie **Nom.** aus der Auswahlliste **Zweite Erweiterung** aus.

Schaltfläche "Hinzufügen"

Jedesmal, wenn Sie ein brauchbares oder vollständiges Ausdruckselement aus der Liste auswählen, wird die Schaltfläche **Hinzufügen** verfügbar. Wenn Sie auf diese Schaltfläche klicken, wird der Text angezeigt, der dem Ausdruck hinzugefügt werden soll.

Wenn Sie z. B. die folgenden Elemente für Ihren Ausdruck gewählt haben:

- Merkmale aus der Liste **Typ des Ausdruckelements**
- D1 aus der Liste **ID**
- X aus der Liste **Erweiterung**
- Nom aus der Liste **Zweite Erweiterung**

In diesem Fall wird die Schaltfläche **Hinzufügen** aktiviert und zeigt folgenden Text an: **D1.X.NOM** hinzufügen.

Wenn Sie auf die Schaltfläche **Hinzufügen** klicken, wird der Text dann im Bearbeitungsfeld unten im Dialogfeld eingeblendet.



Wenn Sie auf die Schaltfläche **OK** klicken, fügt PC-DMIS den Text aus dem Bearbeitungsfeld in das Bearbeitungsfenster an der aktuellen Cursorposition ein. Wenn Sie einen Eintrag aus dem Ausdrucksfeld des Bearbeitungsfensters auswählen und der hinzuzufügende Text Klammern enthält, wird das ausgewählte Element in die Klammern des hinzugefügten Textes gesetzt.

Bearbeitungsfeld

Unten im Dialogfeld **Ausdrucks-Generator (Bearbeiten | Ausdruck)** wird in einem Bearbeitungsfeld der aktuelle Ausdruck angezeigt. Der Ausdruck kann direkt in dieses Feld eingegeben werden, Sie können jedoch auch die Schaltfläche **Hinzufügen** verwenden.

Bereich "Beschreibung"

Das Dialogfeld **Ausdrucks-Generator (Bearbeiten | Ausdruck)** enthält auch einen **Beschreibungsbereich**. Hier werden Informationen zu den aus der Auswahlliste ausgewählten Einträgen angezeigt. Darüber hinaus wird in einem Feld neben der Schaltfläche **Hinzufügen** der aktuelle Wert des Ausdrucks angezeigt.



Ungültige Ausdrücke haben den Wert 0.

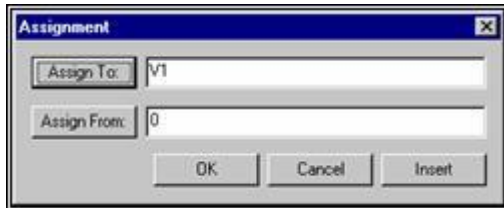
Verwenden von Variablen mit Ausdrücken

Variablen sind Objekte, die Werte enthalten. Variablen beziehen sich auf reelle, Ganzzahl-, Zeichenfolgen- oder Punkt-Operanden. Variablen sind bei der Verwendung von Ausdrücken äußerst wichtig. Eine Variable hat einen Namen und einen Wert. Der Name dient zum Zugriff auf den Wert der Variablen. Er bleibt unverändert, während sich der Wert ändern kann. Über den Befehl **ZUWEISEN/** können Sie einer Variablen einen Wert zuweisen.

Mit der Anweisung **ZUWEISEN/V1=2** wird beispielsweise eine Variable namens V1 mit einem Wert von 2 erstellt. **ZUWEISEN/V2=V1+2** greift auf den Wert von V1 zu. Hätte V1 bei der Ausführung dieser Zuweisungsanweisung immernoch den Wert 2, dann würde V2 den Wert 4 haben.

Weitere Informationen zu Variablen finden Sie unter "Variablen".

Zuweisen von Werten zu Variablen mit Hilfe des Dialogfelds "Zuweisung"



Dialogfeld Zuweisung

Die Menüoption **Einfügen | Zuweisung** blendet das Dialogfeld **Zuweisung** ein. Mit Hilfe dieses Dialogfelds können Sie einer Variablen oder einem Datenelement eines Elements, Merkmals oder Ausrichtung einer Messroutine einen Wert zuweisen. Für die Verwendung des Befehls Zuweisung wird ein grundlegendes Verständnis von PC-DMIS-Ausdrücken vorausgesetzt.

Schaltfläche **Zuweisen zu** - Damit können Sie angeben, welche Variable den im Feld **Zuweisen von** errechneten Wert erhalten soll. Die mit der Schaltfläche **Zuweisen zu** ausgewählten Daten werden in das Feld **Zuweisen zu** gesetzt. Bei diesem Wert kann es sich um den Namen einer Variablen oder um einen Verweis auf ein Datenelement eines Elements, Merkmals oder einer Ausrichtung handeln.

Unter dem Begriff "ausgewertet" versteht man das Ergebnis, das man bei Lösen des mathematischen Ausdrucks eines Werts erhält.

Schaltfläche **Zuweisen von** - Damit können Sie den zuzuweisenden Wert in das Feld **Zuweisen von** platzieren. Enthält dieses Feld einen Ausdruck, so wird dieser bei der Programmausführung ausgewertet und das Ergebnis oder der Wert der Berechnung wird dem im Feld **Zuweisen zu** angegebenen Objekt zugewiesen.

Schaltfläche **Einfügen** - Diese Schaltfläche fügt einen Zuweisungsbefehl in die Messroutine ein, wobei das Dialogfeld **Zuweisung** geöffnet bleibt. Eine Reihe von Zuweisungsbefehlen kann eingefügt werden, ohne dabei das Dialogfeld schließen zu müssen.

Verwandte Themen:

Ausdruckskomponenten

Zugreifen auf die Objekteigenschaften eines Protokolls

Ausdruckskomponenten

Ausdrücke haben folgende Arten von Operanden:

- Ganzzahlen
- Reelle Zahlen
- Zeichenfolgen
- Punkte
- Elementzeiger
- Arrays
- Funktionen

Diese werden im Folgenden näher beschrieben.

Arten von Operanden

Die Operanden können vorliegen als:

- Literale
- Referenzen
- Variablen
- Strukturen
- Zeiger
- Arrays

Literale

***Ganzzahlen:** 1, -6, 209

Reelle Zahlen: 1, -6, 2,4, -0,1, 345.6789

Zeichenfolgen: "Hallo Welt", "47", "KREIS 1"

Punkte: Für Punkte ist keine Literal Darstellung verfügbar. Punkte können jedoch mit der Funktion MPOINT aus anderen Literalen erstellt werden. Beispiel: `MPOINT(0,0,1)`, `MPOINT(2.2, 3.1, 4.0)`.

Dateizeiger: Der Name eines Elements in geschwungenen Klammern:
`{KREIS1}`, `{LINIE2}`, `{F3}`

Arrays: Für Arrays ist keine Literaldarstellung verfügbar. Arrays können jedoch mit der Funktion ARRAY aus anderen Literalen erstellt werden: Beispiel `ARRAY(3, 5, 6)`, `ARRAY("Hallo", 2.3, 9)`. Diese Funktionen erstellen 3 Element-Arrays mit den Ganzzahlelementen 3, 5 und 6 im ersten Beispiel und dem Zeichenfolgeelement "Hallo", dem Doppelement 2.3 und dem Ganzzahlelement 9 im zweiten Beispiel.

Funktionen: Für Funktionen ist keine Literaldarstellung verfügbar. Der Zugriff auf Funktionen erfolgt über Variablen-IDs. Ein Beispiel:
`ZUWEISEN/Add2=FUNCTION(X, X+2)` definiert eine Funktion, mit der einem Argument 2 hinzugefügt wird. Der Variablen Add2 wird die Funktion zugewiesen. Die Funktion kann mit der Variablen Add2 wie folgt aufgerufen werden.
`ZUWEISEN/Ergebnis=Add2(5)`. Dem Ergebnis wird der Wert 7 zugewiesen.



Numerische Literale werden als reelle Zahlen interpretiert, sofern der Operator bzw. die Funktion keine Ganzzahlen vorschreibt. Beispiel: Der Ausdruck `10 / 8` ergibt 1.25 anstatt 1. Beachten Sie außerdem, dass eine ganzzahlige Teilung auch über die Operanden-Umwandlungsoperatoren möglich ist. Der Ausdruck `INT(10) / INT(8)` ergibt den Wert 1.

Verweise

Verweise nehmen Bezug auf Datenelemente von anderen Objekten in einer Messroutine. Verweise basieren auf der ID eines Objekts in der Messroutine, gefolgt von einem Punkt und einer Erweiterung, die auf das Datenelement des Objekts verweist.



Wenn KREIS1 der Name eines gemessenen Kreises in der Messroutine ist, dann verweist `KREIS1.X` auf den gemessenen Wert der Komponente X von KREIS1. Alle Verweise werden in Werkstückkoordinaten relativ zur aktuellen Ausrichtung ermittelt.

Verweise vom Typ "Double"

Es sind die folgenden Bezugsausdrücke verfügbar:

Beispiele für gültige Erweiterungen für Elementverweise vom Typ "Double"

Format: `<Element-ID>.<Erweiterung>` -> `KREIS1.X`

Ausdruckskomponenten

KREIS1.X gemessener X-Wert von KREIS1
KREIS1.Y gemessener Y-Wert von KREIS1
KREIS1.Z gemessener Z-Wert von KREIS1

KREIS1.TX theoretischer X-Wert (Nennwert) von KREIS1
KREIS1.TY theoretischer Y-Wert (Nennwert) von KREIS1
KREIS1.TZ theoretischer Z-Wert (Nennwert) von KREIS1.

GERADE1.SX Gemessener X-Wert des Startpunkts von GERADE1
GERADE1.SY Gemessener Y-Wert des Startpunkts von GERADE1
GERADE1.SZ Gemessener Z-Wert des Startpunkts von GERADE1

GERADE1.TSX Theoretischer X-Wert des Startpunkts von GERADE1
GERADE1.TSY Theoretischer Y-Wert des Startpunkts von GERADE1
GERADE1.TSZ Theoretischer Z-Wert des Startpunkts von GERADE1

GERADE1.EX Gemessener X-Wert des Endpunkts von GERADE1
GERADE1.EY Gemessener Y-Wert des Endpunkts von GERADE1
GERADE1.EZ Gemessener Z-Wert des Endpunkts von GERADE1

GERADE1.TEX Theoretischer X-Wert des Endpunkts von GERADE1
GERADE1.TEY Theoretischer Y-Wert des Endpunkts von GERADE1
GERADE1.TEZ Theoretischer Z-Wert des Endpunkts von GERADE1

PUNKT.I Gemessene I-Komponente des Vektors für PUNKT
PUNKT.J Gemessene J-Komponente des Vektors für PUNKT
PUNKT.K Gemessene K-Komponente des Vektors für PUNKT

PUNKT.TI Theoretische I-Komponente des Vektors für PUNKT
PUNKT.TJ Theoretische J-Komponente des Vektors für PUNKT
PUNKT.TK Theoretische K-Komponente des Vektors für PUNKT

ELEMENT1.TYP Der Elementtyp (z.B. Kreis, Langloch, Kegel). Damit können Sie beispielsweise den Typ eines allgemeinen Elements ändern
(ZUWEISEN/Allg1.TYP=Elem1.TYP).

ELEM1.ALL nimmt auf alle Elemente des Elements Bezug. Diese Art von Bezug eignet sich zum Kopieren von Informationen in ein allgemeines Element.
(ZUWEISEN/Allg1.ALLE=Elem1.ALLE)

Oberflächenvektor

KANTE.SURFI
KANTE.SURFJ
KANTE.SURFK
KANTE.TSURFI
KANTE.TSURFJ

KANTE.TSURFK

Winkelvektor

KREIS.ANGI
KREIS.ANGJ
KREIS.ANGK
KREIS.TANGI
KREIS.TANGJ
KREIS.TANGK

Radius

KREIS1.R
KREIS1.TR
KREIS1.RAD
KREIS1.TRAD
KREIS1.RADIUS
KREIS1.PR – Polarradius
KREIS1.TPR – Theoretischer Polarradius
KREIS1.TRADIUS (Nur die ersten Zeichen, TRAD, sind von Bedeutung)

Durchm.

KREIS1.D
KREIS1.TD
KREIS1.DIAM
KREIS1.TDIAM
KREIS1.DIAMETER
KREIS1.TDIAMETER (nur die ersten Zeichen, TDURCHM, sind von Bedeutung)

Winkel

KEGEL.A
KEGEL.TA
KEGEL.ANG
KEGEL.TANG
KEGEL.ANGLE
KEGEL.TANGLE
KEGEL.PA – Polarwinkel
KEGEL.TPA – Theoretischer Polarwinkel (Nur die ersten Zeichen sind von Bedeutung)

Länge

GERADE.L
GERADE.TL
GERADE.LEN
GERADE.TLEN
GERADE.LENGTH
GERADE.TLENGTH (nur die ersten Zeichen, TLÄNG, sind von Bedeutung)

Höhe

ZYLINDER.PH – Polarhöhe

ZYLINDER.TPH – Theoretische Polarhöhe

Radius, Winkel, Höhe

PUNKT.RAH – Punkt mit Gemessenem Radius, Winkel und Höhe

PUNKT.TRAH – Punkt mit theoretischem Radius, Winkel und Höhe

Fläche

SPRENKEL1.AREA - Gibt den gemessenen Flächenwert für ein Sprengel-Element zurück.

SPRENKEL1.TAREA - Gibt den theoretischen Flächenwert für ein Sprengel-Element zurück.

Beispiel: `ZUWEISEN/V1=SPRENKEL1.AREA` gibt den gemessenen Flächenwert des SPRENKEL1-Elements zurück und weist diesen der Variable V1 zu.

Momentan funktioniert nur das Sprengel-Element mit dieser Flächenerweiterung.

Ausführliche Informationen zum Sprengel-Element finden Sie im Abschnitt "Vision Sprengel" in Ihrer Dokumentation über "PC-DMIS-Vision".

Bespiele für gültige Erweiterungen für Merkmalsverweise vom Typ "Double"

Format: <Merkmal-ID>.<ACHSE>.<Merkmalelement> -> `MERKM1.X.NOM`

<code>MERKM1.X.NOM</code>	Nennwert für die X-Achsenlage von MERKM1
<code>MERKM1.X.MEAS</code>	Gemessener Wert für die X-Achsenlage von MERKM1
<code>MERKM1.X.MAX</code>	Maximale Abweichung für die X-Achsenlage von MERKMAL1
<code>MERKM1.X.MIN</code>	Minimale Abweichung für die X-Achsenlage von MERKMAL1
<code>MERKM1.X.PTOL</code>	Plus-Toleranzwert für die X-Achsenlage von MERKMAL1
<code>MERKM1.X.MTOL</code>	Minus-Toleranzwert für die X-Achsenlage von MERKMAL1
<code>MERKM1.X.DEV</code>	Abweichung auf der X-Achsenlage von MERKMAL1
<code>MERKM1.X.OUTTOL</code>	Außerhalb-Toleranz-Wert auf der X-Achsenlage von MERKMAL1
<code>MERKM1.Y.NOM</code>	Nennwert für die Y-Achsenlage von MERKMAL1
<code>MERKM1.Z.DEV</code>	Abweichung auf der Z-Achsenlage von MERKMAL1
<code>MERKM3.PA.MEAS</code>	Gemessener Wert für die Polarwinkellage von MERKMAL3
<code>MERKM4.M.PTOL</code>	Plus-Toleranzwert für die M-Achsenlage von MERKMAL4

<code>MERKM4 . PTOL</code>	Plus-Toleranzwert für die M-Achsenlage von MERKMAL4 (*siehe Hinweis unter "Gültige Achsen" unten).
<code>MERKM5 . BTOL</code>	Bonus-Toleranzwert, wobei MERKMAL5 eine Position ist.

Gültige Achsen:

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



* Bei Merkmalen, die per Definition nur eine Achse haben (z.B. Rundheit, Konzentrität, usw.), kann der Achsenkennzeichner weggelassen werden. Wenn Sie den Achsenkennzeichner verwenden, sollten Sie beachten, dass alle Merkmalstypen (die nur eine Achse haben) den M-Achsenkennzeichner verwenden – ausgenommen davon sind 2D- und 3D-Winkelabmessungen, bei denen der A-Achsenkennzeichner verwendet wird.

Beispiele für gültige Erweiterungen für Ausrichtungsverweise vom Typ "Double"

Format: <Ausrichtungs-ID>.<Ausrichtungsachse oder Nullpunkt>.<Ausrichtungsachse- oder Nullpunkt-Komponente> -> `A1 . NULLPUNKT . X`

<code>A1 . ORIGIN . X</code>	X-Ausrichtungskomponente des gemessenen Nullpunkts von A1
<code>A2 . ORIGIN . Y</code>	Y-Ausrichtungskomponente des gemessenen Nullpunkts von A2
<code>A1 . ORIGIN . Z</code>	Z-Ausrichtungskomponente des gemessenen Nullpunkts von A1
<code>A1 . XAXIS . I</code>	I-Ausrichtungskomponente der gemessenen X-Achse von A1
<code>A1 . YAXIS . J</code>	J-Ausrichtungskomponente der gemessenen Y-Achse von A1
<code>A1 . ZAXIS . K</code>	K-Ausrichtungskomponente der gemessenen Z-Achse von A1
<code>A1 . CORIGIN . X</code>	X-Ausrichtungskomponente des Nullpunkts von A1 auf Basis der theoretischen Daten (C steht für CAD)
<code>A1 . CXAXIS . J</code>	J-Ausrichtungskomponente der X-Achse von A1 auf Basis der theoretischen Daten (C steht für CAD)

Verweise vom Typ "Punkt"

Es sind die folgenden Bezugsausdrücke verfügbar:

Beispiele für gültige Erweiterungen für Elementverweise vom Typ "Punkt"

Format: <Element-ID>.<Erweiterung> -> KREIS1.XYZ

KREIS1.XYZ	Gemessener Flächenmittelpunkt von KREIS1
KREIS1.TXYZ	Theoretischer Flächenmittelpunkt von KREIS1
LINE1.SXYZ	Gemessener Anfangspunkt von GERADE1
LINE1.TSXYZ	Theoretischer Anfangspunkt von GERADE1
LINE1.EXYZ	Gemessener Endpunkt von GERADE1
LINE1.TEXYZ	Theoretischer Endpunkt von GERADE1
CIRCLE1.IJK	Gemessener Vektor von KREIS1
CIRCLE1.TIJK	Theoretischer Vektor von KREIS1
EDGE.SURFIJK	Gemessener Oberflächenvektor von KANTE
EDGE.TSURFIJK	Theoretischer Oberflächenvektor von KANTE
AUTOCIR1.ANGIJK	Gemessener Winkelvektor von AUTOKREIS1

<code>AUTOCIR1.TANGIJK</code>	Theoretischer Winkelvektor von AUTOKREIS1
-------------------------------	---

Beispiele für gültige Erweiterungen für Ausrichtungsverweise vom Typ "Punkt"

Format: <Ausrichtungs-ID>.<Ausrichtungsachse oder Nullpunkt> -> A1.XAXIS

<code>A1.ORIGIN</code>	Gemessener Nullpunkt von Ausrichtung A1
<code>A1.XAXIS</code>	Gemessene X-Achse von Ausrichtung A1
<code>A1.YAXIS</code>	Gemessene Y-Achse von Ausrichtung A1
<code>A1.ZAXIS</code>	Gemessene Z-Achse von Ausrichtung A1
<code>A1.CORIGIN</code>	Theoretischer Nullpunkt von Ausrichtung A1
<code>A1.CXAXIS</code>	Theoretische X-Achse von Ausrichtung A1
<code>A1.CYAXIS</code>	Theoretische Y-Achse von Ausrichtung A1
<code>A1.CZAXIS</code>	Theoretische Z-Achse von Ausrichtung A1

Verweise vom Typ "Zeichenfolge"

Verweise auf Kommentare sind die einzigen Objekte vom Typ "Zeichenfolge". Über Verweise kann nur auf EINGABE-Kommentare oder JA/NEIN-Kommentare Bezug

genommen werden. Diese Kommentartypen verfügen über eine ID, über die sie identifiziert werden können.

Format: <Kommentar-ID>.INPUT -> `C1.INPUT`

C1.INPUT – Der (vom Bediener eingegebene) Eingabewert für Kommentar C1.

Kommentare vom Typ JA/NEIN setzen die Eingabe unter Berücksichtigung der aktuellen Spracheinstellung von PC-DMIS auf die jeweilige "Ja"- bzw. "Nein"-Zeichenfolge. In der englischen Ausgabe von PC-DMIS kann der Bediener die Zeichenfolge mit der Schaltfläche "JA" auf "JA". Mit der Schaltfläche "Nein" kann der Bediener die Zeichenfolge auf "Nein". Beachten Sie, dass bei Vergleichen mit diesen Zeichenfolgen zwischen Groß- und Kleinschreibung unterschieden wird. Wenn Sie daher die gesetzten Zeichenfolgen mit "ja" bzw. "nein" vergleichen, erhalten Sie immer ein falsches Ergebnis, selbst wenn der Kommentartyp auf "JA" bzw. "NEIN" gesetzt ist.

Variablen

Variablen gehören einem von sieben Operandentypen an: Ganzzahl, reell, Zeichenfolge, Punkt, Elementzeiger, Array und Funktion. Variablen kommen zustande und erhalten ihre Werte und den Typ über die Anweisung `ZUWEISEN`.

Die Variablen-ID kann aus einer beliebigen alphanumerischen Zeichenfolge bestehen, die nicht mit einem numerischen Zeichen beginnt. Sie können Unterstriche in der Variablen-ID verwenden, solange der Unterstrich nicht das erste Zeichen ist.



Solange die Messroutine geöffnet bleibt, speichert PC-DMIS Variablenwerte zwischen den Ausführungsläufen. Das heißt, wenn die Ausführung beendet ist, verwendet PC-DMIS die Werte als Ende der Routine, wenn Sie die Routine neu starten. Sie können dieses Verhalten wünschen oder nicht. Wenn Sie frische Variablenwerte wünschen, ist es eine gute Praxis, die Werte mit ZUWEISEN-Anweisungen am Anfang der Routine zu löschen. Wenn Sie z. B. einen Variablenwert V1 in einigen Zahlenberechnungen verwenden, können Sie `ZUWEISEN/V1=0` verwenden, um diese Variable zu löschen.



Wenn das Bearbeitungsfenster aktiviert ist, zeigt PC-DMIS den aktuellen Wert einer Variablen an, sobald der Cursor in das entsprechende Feld gesetzt wird. Die Werte von Variablen ändern sich während der Ausführung je nach Ablauf des Programms. Positionieren Sie den Mauszeiger über einer beliebigen Variable, um den aktuellen Wert dieser Variable zu sehen.

```
ZUWEISEN/V1=2.2+2
```

Die Variable V1 ist eine reelle Zahl mit dem Wert 4.2

```
ZUWEISEN/VAR1=KREIS1.X
```

Die Variable VAR1 ist eine reelle Zahl, deren Wert dem zum Zeitpunkt der Zuweisung gemessenen Wert von KREIS1.X entspricht.

```
ZUWEISEN/MEINEVAR=GERADE1.XYZ
```

Die Variable MEINEVAR ist ein Punkt, dessen Wert dem zum Zeitpunkt der Zuweisung gemessenen Flächenmittelpunkt von GERADE1 entspricht.

```
ZUWEISEN/SVAR="Hallo Welt"
```

Die Variable SVAR ist eine Zeichenfolge mit dem Wert "Hallo Welt".

In diesen Beispielen sind die Variablen zugewiesene Werte. Wenn einer Variablen ein Wert zugewiesen wurde, kann sie in beliebigen Ausdrucksfeldern als Operand verwendet werden.

Hier wird V1 in einem numerischen Feld verwendet. Sie wird als Wert des Anfahrabstands für den Befehl ANFAHRWEG verwendet:

```
ZUWEISEN/V1=1/3ANFAHRWEG/V1
```



In den meisten editierbaren Feldern können Ausdrücke verwendet werden. Auch der folgende Ausdruck ist daher gültig und zeigt dieselbe Wirkung: ANFAHRWEG / 1/3.

Auf Variablenkomponenten des Typs "Punkt" kann über die für Verweise verwendete Punkt-Erweiterungs-Schreibweise einzeln Bezug genommen werden.

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

Die Variable V1 ist vom Typ "Punkt" und hat einen Wert von 3, 4, 5.

Ausdruckskomponenten

`ZUWEISEN/XVAR=V1.X`

XVAR ist vom Typ "Double" und hat einen Wert von 3.

`ZUWEISEN/YVAR=V1.Y`

YVAR ist vom Typ "Double" und hat einen Wert von 4.

`ZUWEISEN/IVAR=V1.I`

XVAR ist vom Typ "Double" und hat einen Wert von 3.

`ZUWEISEN/REDUNVAR=V1.XYZ`

REDUNVAR ist vom Typ "Punkt" und hat einen Wert von 3, 4, 5.

Die folgenden Erweiterungen sind gleichwertig. Beide Beispiele dienen dazu, die Bedeutung von Ausdrücken in Messroutinen zu verdeutlichen.

Angenommen, V1 ist vom Typ "Punkt".

V1.X entspricht V1.I

1.Y entspricht V1.J

V1.Z entspricht V1.K

V1.XYZ entspricht V1.IJK und V1 ohne Erweiterung.

Wenn einer Variablen vom Typ "Zeichenfolge" ein Wert zugewiesen wurde, der dem Namen der ID eines Elements, Merkmals oder einer Ausrichtung entspricht, so kann diese Variable als Bezugsobjekt verwendet werden:

`ZUWEISEN/V1="KREIS1"`

Die folgenden Operanden sind möglich und gültig, vorausgesetzt, es ist ein Element mit dem Namen KREIS1 vorhanden.

V1.X - der gemessene X-Wert von KREIS1

V1.TX - der theoretische X-Wert von KREIS1

V1.Diameter (Durchmesser) - der gemessene Durchmesser von KREIS1

V1.Radius - der gemessene Radius von KREIS1

Diese Art eines bei Zeichenfolgen-Variablen möglichen Indirektverweises ist auf eine Ebene von Indirektverweisen beschränkt. Die folgende Kombination funktioniert nicht.



```
ASSIGN/V1="CIRCLE1"  
ASSIGN/V2="V1"
```

V2.X - Dies ergibt den Wert 0 und nicht den aktuell gemessenen Wert von KREIS1.X.



Der Verweis V2.X wird *nicht* als Fehler durch roten Text gekennzeichnet, obwohl ein darüber stehender Ausdruck seinen Typ auf "Zeichenfolge" setzt. Der Grund dafür, dass es nicht als Fehler gekennzeichnet werden kann, ist, dass der Ablauf der Messroutine bis zur Ausführungszeit nicht bekannt ist.

Wenn Sie jedoch geschwungene Klammern verwenden, funktioniert es folgendermaßen:



```
ZUWEISEN/V1={KREIS1} ZUWEISEN/V2={V1}
```

V2.X - Dadurch erhalten Sie den Wert von KREIS1.X.

Sehen Sie sich das folgende Beispiel an:



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

Sobald während der Ausführung der Messroutine der Wert von KREIS1.X größer als der Wert von KREIS1.TX ist, ist der Ausdruck V2.X gültig und wird mit 2 wiedergegeben. Andernfalls wird der Ausdruck V2.X mit 0 wiedergegeben, da der Wert von V2 zum Zeitpunkt der ZUWEISUNG von V3 die Zeichenfolge "V1" ist. Der Programmierer des Werkstückprogramms muss dafür sorgen, dass die Ausdrücke in allen Fällen brauchbare Ergebnisse liefern.



Beinahe alle Elementverweise können auf der linken Seite der **Zuweisungsanweisung** (ZUWEISEN) verwendet werden, um einem gemessenen oder theoretischen Datenelement eines Elements einen Wert zuzuweisen. Die einzige Ausnahme bilden die einzelnen I-, J- und K-Komponenten von Vektoren. Um Vektoren Werte zuzuweisen, muss der gesamte Vektor als Einheit mit einem Ausdruck zugewiesen werden, der als Punkt ausgewertet wird. Vektordaten werden normalisiert, wenn sie in die Vektordatenelemente des Elements eingegeben werden.



```
ASSIGN/CIRCLE1.I=2-illegal  
ASSIGN/CIRCLE1.IJK=MPOINT(2,0,0)-legal (vector  
is normalized to 1,0,0)
```

Weitere Informationen zur Anwendung von Variablen innerhalb von Merkmalen finden Sie im Thema "Merkmale für Variablen erstellen" im Abschnitt "Anwenden von V3.7-kompatiblen Merkmalen".

Strukturen

Mit Hilfe von *Strukturen* können Sie Variablen mit Erweiterungen versehen, um ein Unterelement dieser Variablen zu kennzeichnen. Sie können dies im untenstehenden Codeausschnitt sehen:



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

Wobei:

- `V1` ist die Struktur.
- `HÖHE`, `BREITE`, `MODUS`, und `PUNKT` sind Unterelemente der Struktur.

Regeln für Strukturen

- Ebenso wie Variablen müssen auch Strukturen nicht deklariert werden.
- Unterelemente einer Struktur können aus folgenden Variablentypen bestehen:
 - Integer (Ganzzahl)
 - Double (Verdoppelung)
 - Punkt
 - Elementzeiger
 - Funktion
 - Array
 - Struktur

So können Sie beispielsweise Strukturelemente haben, die Arrays sind, und Array-Elemente, die Strukturen sind. Sie können die gültigen Ausdrücke im untenstehenden Codeausschnitt sehen:

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


```
COMMENT/OPER,"Aktuelle Z-Position:  
"+CAR.LEFTSIDE.DOOR[2].QUADRANT[3].JOINT[5].HIT[4].Z
```

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


```
COMMENT/OPER,"Nächster Messpunkt: "+CURRENTJOINT.HIT[4]
```

Strukturen mit Variablen vom Typ "Punkt"

Wenn eine Variable den Typ "Punkt" aufweist, können Sie die Erweiterungen .X, .Y, .Z, .I, .J und .K dennoch verwenden, um einzelne Elemente des Punkts zu erhalten. Sie können auch alle Erweiterungen aus diesem Beispiel in deren Strukturen einsetzen, ohne sie dabei als Punktelemente verwenden zu müssen, wie hier dargestellt:

	<pre>ASSIGN/V1.X="Some string" ASSIGN/V1.Y=ARRAY(1,3,5,9,7) ASSIGN/V1.Z=MPOINT(3,5,7)</pre>
KOMMENTAR/PROTOK,V1.X	Die Ausgabe lautet "Eine Zeichenfolge".
KOMMENTAR/PROTOK,V1.Y[2]	Die Ausgabe lautet "3", das zweite Element des Arrays.
KOMMENTAR/PROTOK,V1.Z.Y	Die Ausgabe ist "5", der Y-Wert von MPOINT.

Durch das Kombinieren von Strukturen mit den Funktionen der PC-DMIS-Ausdruckssprache sind dynamische Strukturverweise, wie sie hier gezeigt werden, möglich:

	<pre>ASSIGN/DYNAMICSTRUCT=FUNCTION((X,Y),X.Y) C1=COMMENT/INPUT,Please enter in item ASSIGN/TESTSTR=C1.INPUT ASSIGN/FRONT=LEFT(TESTSTR,INDEX(TESTSTR,".")-1) ASSIGN/BACK=MID(TESTSTR,INDEX(TESTSTR,".")) ASSIGN/RESULT=DYNAMICSTRUCT(FRONT,BACK)</pre>
---	--

Ausdruckskomponenten

In diesem Teil des Beispiels werden Sie aufgefordert, einen Variablenverweis einzugeben, dann wird der Verweis am ersten '.' geteilt und mit Hilfe der Funktion `DYNAMISCHESTRUKT` festgelegt, dass `ERGEBNIS` diesem Verweis entspricht.

Wenn Sie also beispielsweise `V1.Y[4]` für die Variable `C1.INPUT` eingeben, erhält `ERGEBNIS` schließlich den Wert 9 (das vierte Element des Arrays, das `V1.Y` zugewiesen ist).

Die Lernzeitauswertung der Ausdrücke wurde dahingehend verbessert, dass nun alle Elemente einer Struktur oder eines Arrays genau angezeigt werden.

Zeiger

Zeiger sind auch als "Elementzeiger" bekannt. Weitere Hinweise dazu finden Sie unter dem Begriff "Elementzeiger" im Glossar.

Mit Zeigern können Sie auf eine Variable auf ein Element verweisen oder Objekte mit dem Befehl "Unterprogramm aufrufen" übergeben. Zeiger funktionieren ähnlich wie ein Indirektverweis über Zeichenfolgenamen. Die Vorteile von Zeigern zeigen sich jedoch bei Unterprogrammen. Im Gegensatz zu Zeichenfolgen lassen Zeiger, die als Argument eines Unterprogramms übergeben werden, eine direkte Modifizierung des Objekts zu, auf das das Unterprogramm verweist. Zeiger werden nicht in komplexen Ausdrücken verwendet. Wird ein Zeiger in einem komplexen Ausdruck verwendet, ergibt er den Wert 0.

Beachten Sie die folgenden Beispiele.

Beispiel: Zeigerverwendung

In diesem Beispiel ist `v1` als Zeiger definiert, der auf `KREIS1` verweist:

```
ZUWEISEN/V1={KREIS1}
```

In diesem Beispiel wird `DIST` der Wert des Abstands von `CIR1` vom Ursprung zugewiesen:

```
ZUWEISEN/ABST=DOUBLE (V1.XYZ)
```

Sie können einen Ausdruck auch zwischen geschwungene Klammern setzen, um einen Elementzeiger zu erhalten. Nun sind alle folgenden Beispiele gültige Methoden, um den Zeiger auf das Element `KREIS1` einzustellen:

```
ZUWEISEN/ELEMENTZÄHLER=1
```

```
ZUWEISEN/V1={"KREIS"+ELEMENTZÄHLER}
```

Weist V1 den Ausdruck "KREIS1" zu.

```
ZUWEISEN/V2="KREIS1"
```

```
ZUWEISEN/V3={V2}
```

Weist Variable V3 den Ausdruck "KREIS1" aus Variable V2 zu.

```
C1=KOMMENTAR/EINGABE,Bitte Elementnamen eingeben.
```

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

Nimmt den Elementnamen aus C1.INPUT und setzt ihn in Variable V4.

Beispiel: Unterprogramm

In der aufrufenden Messroutine:

```
CS1=UNTERPROGAUFR/UNTERPRG,CHANGEX,{KREIS1}
```

In der Unteroutine:

```
GEN1=GENERIC/FEATURE
```

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

(Während der Ausführung, wenn der Code KREIS1 an die Routine übergibt, nimmt KREIS1 den Platz von GEN1 ein)

```
ARG1.X=5
```

(Setzt den gemessenen X-Wert von KREIS1 auf 5)

```
ENDE/UNTERPROGRAMM
```

Beispiel: komplexer Ausdruck

```
ZUWEISEN/V1={KREIS1}+2
```

{KREIS1} ergibt 0 und somit ergibt der gesamte Ausdruck 2.

Arrays

Es sind drei Array-Typen verfügbar: Element-Arrays, Messpunkt-Arrays und Variablen-Arrays.



Achtung: Selbst wenn mehrdimensionale Arrays im Softwareprogramm mehrdimensional dargestellt werden, können diese erst als mehrdimensionale Arrays genutzt werden, wenn ihnen ein ARRAY-INDIZES-Befehl (siehe auch "Array-Indexobjekt:") vorangestellt wird. Vorher sind sie nur eindimensional

Element-Arrays

Wenn ein Element während der Ausführung einer Routine mehr als einmal gemessen wird, weil es sich um eine Art Schleife handelt, erstellt die Software automatisch ein Merkmalfeld. Die Anzahl der Elemente in diesem Array entspricht dabei der Anzahl der Ausführungen dieses Elements.



Wenn sich ein gemessenes Kreismerkmal innerhalb einer while-Schleife befindet, die fünfmal ausgeführt wird, dann existiert ein Array mit fünf gemessenen Kreisen. Lautet die ID des gemessenen Kreises "Kreis1", kann mit einem Array-Ausdruck auf die einzelnen Instanzen des gemessenen Kreisobjekts zugegriffen werden. Sie verwenden eckige Klammern, um die gewünschte Instanz anzugeben, etwa so:

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

V1 wird der gemessene X-Wert der dritten Instanz des CIR1-Kreises zugewiesen.



Wenn für ein bestimmtes Element ein Element-Array vorhanden ist, in einem Verweis jedoch keine Array-Schreibweise verwendet wird, nimmt PC-DMIS die aktuellste Instanz. Im obigen Beispiel würde der Bezug `CIR1.X` dem Bezug `CIR1[5].X` entsprechen, da die fünfte Instanz die aktuellste Instanz des Objekts ist.

Sie können Ausdrücke innerhalb der eckigen Klammern eines Array-Ausdrucks verwenden:

`CIR1[3].X` und `CIR1[2+1].X` sind demzufolge gültige gleichwertige Bezüge.

Das nächste Beispiel verwendet zwei While / End While Schleifen-Befehlsblöcke. Der erste Block führt den CIR1-Kreis fünfmal aus. Der zweite Block verwendet die Variable V1 in eckigen Klammern, also `CIR1[V1].XYZ`, um den gemessenen Schwerpunkt jeder der fünf Ausführungen an das Berichtsfenster zu senden:



```

                                ZUWEISEN/V1=1
                                WHILE/V1<6
KREIS1                          =ELEMENT/KREIS,KARTESISCH,INNEN,KLEINST_QU
ADR                             NENN/<40,30,-4.824>,<0,0,1>,30
                                ISTW/<40.002,29.991,-4.836>,<0,0,1>,29.982
                                MESS/KREIS,4,ZPLUS
                                MESSPKT/BASIS,NORMAL,<41.984,44.868,-
                                2.885>,<-0.132272,-0.9912135,0>,<41.972,44.85,-
                                2.891>,NENN VERWENDEN=JA

```



```

        MESSPKT/BASIS,NORMAL,<51.721,39.36,-
5.094>,<-0.781412,-0.6240155,0>,<51.706,39.375,-
5.107>,NENN VERWENDEN=JA
        MESSPKT/BASIS,NORMAL,<54.792,32.491,-
5.44>,<-0.9861119,-0.1660821,0>,<54.775,32.474,-
5.453>,NENN VERWENDEN=JA
        MESSPKT/BASIS,NORMAL,<52.526,21.748,-
5.879>,<-0.8350841,0.5501223,0>,<52.537,21.764,-
5.893>,NENN VERWENDEN=YES
        ENDMESS/
        ZUWEISEN/V1=V1+1
    END_WHILE/
    ZUWEISEN/V1=1
    WHILE/V1<6
        COMMENT/REPT,
        "Schwerpunkt von KREIS1, Instanz #" + V1
        KREIS1[V1].XYZ
        KOMMENTAR/REPT,
        -----
        ZUWEISEN/V1=V1+1
    END_WHILE/

```



Hier sehen Sie die generierte Ausgabe im Berichtsfenster:

	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>

Arrays sind auch für Merkmale und Ausrichtungen vorhanden, die mehrmals während eines Programmablaufs ausgeführt wurden. Die Ausdrücke `Merkm1[2].Nom` und `Ausrich1[4].Origin` sind gültig, sofern das Merkmal "Merkm1" mindestens zweimal und die Ausrichtung "Ausrich1" mindestens viermal ausgeführt wurden.

Wenn ein Verweis auf ein Element-Array außerhalb der Grenzen liegt (z. B. wenn der Benutzer nach `CIR1[2.5]` oder `CIR1["Hello, World"]` fragt), wird das Element mit der oberen oder unteren Grenze zurückgegeben. Weist `Kreis1` beispielsweise 3 Instanzen auf, gibt `CIR1[4]` und höher `CIR1[3]` zurück und `CIR1[0]` und tiefer würde `CIR1[1]` zurückgeben. Alle Ausdrücke zwischen eckigen Klammern werden als Ganzzahlen interpretiert – 2.5 ergibt demzufolge 2, und "Hallo Welt" ergibt 0.

Array-Indexobjekt

Standardmäßig handelt es sich bei Element-Arrays stets um eindimensionale Arrays. Wenn die Behandlung eines Element-Arrays als mehrdimensionales Array vorteilhafter ist, können Sie zu diesem Zweck das Array-Index-Objekt einsetzen.

Das Array-Indexobjekt ermöglicht Ihnen die Angabe von Ober- und Untergrenzen, um mehrdimensionale Arrays zu erstellen.

- Durch das Festlegen der Ober- und Untergrenze der ersten Dimension wird PC-DMIS ein zweidimensionaler Array erstellen, bei dem die erste Dimension begrenzt und die zweite Dimension unbegrenzt ist.
- Durch Einstellen der Ober- und Untergrenze der ersten zwei Dimensionen eines Arrays erstellt PC-DMIS ein dreidimensionaler Array. Die letzte Dimension ist immer unbegrenzt.



Das Element F1 befindet sich in einer verschachtelten WHILE-Schleife. Die innere WHILE-Schleife wird fünfmal ausgeführt, und die äußere WHILE-Schleife wird dreimal ausgeführt. Wenn die Ausführung abgeschlossen ist, wurde F1 15mal ausgeführt. Somit sind 15 Instanzen von F1 vorhanden.

Sehen Sie sich das Segment einer Messroutine im folgenden Beispiel an:

```
ARRAY-INDIZES/1..5, ..
```

```
ASSIGN/V1=1
```

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

```
    ASSIGN/V2=1
```

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

```
        F1=FEAT/POINT,RECT
```

```
        THEO/V2,V1,0,0,0,1
```

```
        ACTL/1,1,0,0,0,1
```

```
        MESS/PUNKT,1
```

```
        HIT/BASIC,V2,V1,0,0,0,1,1,1,0
```

```
        ENDEMESS/
```

```
        ASSIGN/V2=V2+1
```

```
        COMMENT/REPT,"Location of F1["+V2+", "+V1+"]  
        : "+F1[V2,V1].XYZ
```

```
    END_WHILE/
```

```
    ASSIGN/V1=V1+1
```

```
END_WHILE/
```

Mit diesem Codesegment wird ein 3 x 5-Raster von 15 gemessenen Punkten erstellt.

Der ARRAY-INDIZES-Befehl hat für die erste Dimension des Elements einen Wertebereich von 1 bis einschließlich 5 festgelegt. Demzufolge erscheinen die Objekte auf dem Prüfprotokoll anstelle von F1[1] – F1[15] als F1[1, 1] – F1[5, 3], konsequenter mit dem Layout der Elemente. Beachten Sie, dass auch der Kommentar auf das Element-Array mit einer Syntax für zweidimensionale Arrays verweist.

So fügen Sie ein Array-Indizes-Objekt in eine Messroutine ein:

1. Geben Sie über die Tastatur **"Array"** in einer leeren Zeile im Bearbeitungsfenster ein.
2. Durch Drücken auf die Tabulatortaste auf Ihrer Tastatur.



Wenn das Kontrollkästchen **Klammern für Element-Arrays anzeigen** deaktiviert ist, wird das Element nicht mit dem Namen in Klammern angezeigt. Siehe die Beschreibung von "Klammern für Element-Arrays anzeigen" unter dem Thema "Setup-Optionen, Registerkarte ID-Setup" im Kapitel "Voreinstellungen".

Messpunkt-Arrays

Die Messpunkte eines vorgegebenen Elements sind als Array verfügbar. Sie können auf diese Messpunkte über Ausdrücke mit folgender Array-Syntax zugreifen: <Element-ID>.Hit[<Array-Ausdruck>].<Erweiterung> oder <Element-ID>.RawHit[<Array-Ausdruck>].<Erweiterung>. Hit gibt tasterkompensierte Daten zurück, wenn die Tasterkompensation eingeschaltet ist. RawHit gibt immer Rohdaten (nicht kompensierte Daten) zurück. Gültige Erweiterungen sind X, Y, Z, I, J, K, TX, TY, TZ, TI, TJ, TK, XYZ, TXYZ, IJK und TIJK

```
Kreis1.Hit[1].XYZ
```

Der gemessene Flächenmittelpunkt (tasterkompensiert) von Messpunkt 1 von "Kreis1".

```
Kreis1.Hit[2].IJK
```

Der gemessene Vektor von Messpunkt 2 von "Kreis1".

Messpunktdaten sind für alle Objekte verfügbar, die Messpunkte haben, unabhängig davon, ob die tatsächlichen Messpunkte im Bearbeitungsfenster angezeigt werden oder nicht. Messpunkte können daher von Scans oder Auto-Elementen stammen.

Sie können Messpunkt-Arrays verwenden, um Eingaben für abhängige Elemente zu definieren.

Sie können Messpunkt-Arrays verwenden, um die Eingaben für abhängige Elemente zu definieren.

Wenn Sie bei abhängigen Elementen die Methoden `Element.MESSPUNKT[start..ende]` oder `Element.MESSPUNKTE[start..ende]` verwenden, sind die Eigenschaften `start` und `ende` optional:

- Wenn der Startwert nicht definiert ist, nimmt PC-DMIS den Wert "1" an.
- Wenn der Endwert nicht definiert ist, nimmt PC-DMIS an, dass es sich um die Gesamtzahl der Messpunkte für das Element handelt.

Dies ist dasselbe, wenn Sie `"Element.ANZPKTE"` eingeben.



Diese Beispiele zeigen drei verschiedene Möglichkeiten, dieselben Messpunkte aus KREIS1 zu verwenden:

Beispiel 1

In diesem Beispiel sind die Anfangs- und Endwerte explizit definiert:



```
KREIS3      =ELEMENT/KREIS,KARTESISCH,INNEN,NEIN
            NENNW/<20.97629,22.90352,0>,<0,0,-1>,20.97629
            ISTW/<20.96578,22.89023,-0.01243>,<0,0,-
1>,20.96578
            ABHÄNGIG/KREIS,UMK,KREIS1.MESSPUNKT[1..KREIS1.AN
ZPKTE]
```

Beispiel 2

In diesem Beispiel ist nur der Startwert "1" definiert. Da der Endwert nicht definiert ist, definiert PC-DMIS den Endwert als "KREIS1.ANZPKTE":



```
KREIS3      =ELEMENT/KREIS,KARTESISCH,INNEN,NEIN
            NENNW/<20.97629,22.90352,0>,<0,0,-1>,20.97629
            ISTW/<20.96578,22.89023,-0.01243>,<0,0,-
1>,20.96578
            ABHÄNGIG/KREIS,UMK,KREIS1.MESSPUNKT[1..]
```

Beispiel 3

In diesem Beispiel sind weder der Start- noch der Endwert definiert. Aus diesem Grund definiert PC-DMIS den Startwert als "1" und den Endwert als "KREIS1.ANZPKT":



```
KREIS3      =ELEMENT/KREIS,KARTESISCH,INNEN,NEIN
              NENNW/<20.97629,22.90352,0>,<0,0,-1>,20.97629
              ISTW/<20.96578,22.89023,-0.01243>,<0,0,-
              1>,20.96578
              ABHÄNGIG/KREIS,UMK,KREIS1.MESSPUNKT[...]
```

In den folgenden Abschnitten werden weitere Array-Funktionen behandelt, die zum Auffinden der minimalen oder maximalen Punkte in einem Scan hilfreich sind.

Zuweisen eines Messpunktbereichs zu einem Array.

Mit Hilfe der folgenden Syntax können Sie einem Array einen Messpunktbereich zuweisen:

<Element-ID>.<Messpunkttyp>[<Startnum>..<Endnum>].<Erweiterung>

wobei:

<Element-Id> den Namen des Elements angibt.

<Messpunkttyp> entweder das Wort "HIT" für kompensierte Daten oder "RAWHIT" für unkompensierte Daten enthalten kann. Wird die Tasterkompensation ausgeschaltet, sind wiedergegebene Werte immer unkompensiert.

<Startnum> ist ein Ausdruck, der zur Erkennung des ersten Indexwertes des Messpunktbereichs dient.

<Endnum> ist ein Ausdruck, der zur Erkennung des zweiten Indexwertes des Messpunktbereichs dient.

<Erweiterung> dient zur Erkennung des Datentyps. Zu den möglichen Erweiterungen gehören die im nächsten Abschnitt aufgeführten gemessenen oder theoretischen Datentypen.



Beispiel für die Verwendung impliziter Start- und Endwerte in einem Ausdruck.

Angenommen, Sie wollen alle Messpunkte finden, die eine Ebene (EBENE1) definieren. Sie können es so formulieren:

```
[EBENE1.MESSPUNKT[1..EBENE1.ANZPKTE]
```

Ein kürzerer Weg, dies zu tun, ist die Verwendung abgeleiteter Start- und Endtrefferwerte. Sie können denselben Code auch so schreiben:

```
EBENE1.MESSPKT[...]
```

Da wir in diesem Fall die Start- und Endwerte der Messpunkte nicht definiert haben, nimmt PC-DMIS an, dass der Start-Messpunkt gleich 1 und der End-Messpunkt gleich NUMHITS ist.

Istwert

- X – Gemessene X-Werte der Messpunkte
- Y – Gemessene Y-Werte der Messpunkte
- Z – Gemessene Z-Werte der Messpunkte
- XYZ – Gemessene XYZ-Werte der Messpunkte
- I – Gemessene I-Werte der Messpunkte
- J – Gemessene J-Werte der Messpunkte
- K – Gemessene K-Werte der Messpunkte
- IJK – Gemessene IJK-Werte der Messpunkte

Theo

- TX – Theoretische X-Werte der Messpunkte
- TY – Theoretische Y-Werte der Messpunkte
- TZ – Theoretische Z-Werte der Messpunkte
- TXYZ – Theoretische XYZ-Werte der Messpunkte
- TI – Theoretische I-Werte der Messpunkte
- TJ – Theoretische J-Werte der Messpunkte
- TK – Theoretische K-Werte der Messpunkte
- TIJK – Theoretische IJK-Werte der Messpunkte

Zum Beispiel:

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

V1 wird einem Array aus 10 Werten zugewiesen, welche die gemessenen X-Werte aus den ersten 10 Messpunkten von SCAN1 sind.

`ZUWEISEN/V2=SCAN1.HIT[1..SCAN1.ANZMESSPUNKTE].XYZ`

V2 wird einem Array aus Punkten aus jedem der Flächenmittelpunkte der Messpunkte des Scans zugewiesen.

Arrays sortieren

Arrays können in PC-DMIS in auf- oder absteigender Reihenfolge sortiert werden. Die folgenden Ausdrücke nehmen einen Array an und geben ihn als sortierten Array zurück:

Um in *aufsteigender* Reihenfolge zu sortieren, verwenden Sie:

`SORTUP(<Array>)`

Um in *absteigender* Reihenfolge zu sortieren, verwenden Sie:

`SORTDOWN(<Array>)`

Zum Beispiel:

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

V1 wird die Reihe "5,8,3,9,2,6,1,7" zugewiesen.

`ZUWEISEN/V2=SORTUP(V1)`

V2 wird die Array-Werte in aufsteigender Reihenfolge enthalten: "1,2,3,5,6,7,8,9"

`ZUWEISEN/V3=SORTDOWN(V1)`

V3 wird die Array-Werte in absteigender Reihenfolge enthalten: "9,8,7,6,5,3,2,1"

Die größten oder kleinsten Indexwerte eines Arrays zurückgeben

Sie können einen Array in eine Funktion eingeben und die Indexwerte des Elements, das den größten oder niedrigsten Wert aufweist, mit Hilfe folgender Funktionen zurückgeben:

Um den Indexwert des Elements mit dem *größten* Wert zurückzugeben, verwenden Sie:

`MAXINDEX(<array>)`

Um den Indexwert des Elements mit dem *kleinsten* Wert zurückzugeben, verwenden Sie:

`MININDEX(<array>)`

Zum Beispiel,

Ausdruckskomponenten

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

V1 wird die Reihe "5,8,3,9,2,6,1,7" zugewiesen.

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

V2 wird den Array-Indexwert von 4 halten. Der Istwert dieses Arrays lautet 9.

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

V3 wird den Array-Indexwert von 7 halten. Der Istwert dieses Arrays lautet 1.

Sie können die zurückgegebenen Indexwerte zugrundelegen, um den Istwert des Array-Elements zu ermitteln.

Die größten oder kleinsten sortierten Indexwerte eines Arrays zurückgeben.

Sie können einen Array in eine Funktion eingeben, die Array-Werte in auf- oder absteigender Reihenfolge sortieren lassen und die Indexwerte dann mit Hilfe folgender Funktionen zurückgeben:

Um die Indexpositionen des Arrays zurückzugeben, damit deren Werte in absteigender Reihenfolge sortiert werden, verwenden Sie:

```
MAXINDICES(<Array>)
```

Um die Indexpositionen des Arrays zurückzugeben, damit deren Werte in absteigender Reihenfolge sortiert werden, verwenden Sie:

```
MININDICES(<Array>)
```

Zum Beispiel:

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

V1 wird die Reihe "4,8,2,9,5,7" zugewiesen.

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

V2 wird die Reihe mit diesem Werten halten: "4,2,6,5,1,3".

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

V3 wird die Reihe mit diesem Werten halten: "3,1,5,6,2,4".

Beispiel für die Verwendung von Arrays, um die minimalen und maximalen Punkte in einem Scan zu finden

Die oben behandelten Messpunkt-Arrayfunktionen dienen hauptsächlich dem Zweck, Ihnen ein einfaches Auffinden der minimalen oder maximalen Punkte in einem Scan zu ermöglichen.

Um den Punkt aus SCAN1 mit dem größten gemessenen X-Wert zu bemaßen, könnten Sie folgenden Ausdruck verwenden:



```
ZUWEISEN/MAXPKTINDEX=MAXINDEX (SCAN1.HIT[1..SCAN1.NUMHITS]
.X)
D1=LAGE VON ELEMENT SCAN1.HIT[MAXPKTINDEX]
```

Um die drei höchsten Punkte in der Z-Achse von SCAN2 zu finden, könnten Sie folgenden Ausdruck verwenden:



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

Variablen-Arrays

Variablen-Arrays müssen nicht deklariert werden. Variablen-Arrays entstehen, sobald eine ZUWEISEN-Anweisung ausgeführt wird und der Ausdruck auf der rechten Seite der Anweisung einen Array ergibt oder sich der Ausdruck auf der linken Seite der Anweisung auf ein Element in einem Variablen-Array bezieht.

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

Erstellt einen Array aus 5 Elementen und weist ihn V1 zu.

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

Weist V2 den Wert des dritten Elements im Array V1 zu: 5.

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

Weist dem vierten Element des Arrays V1 den Wert 23 zu.

Arrays werden dynamisch erstellt und zugeordnet. Ein Array kann demzufolge mit einem Array-Bezug auf der linken Seite einer ZUWEISEN-Anweisung erstellt werden.

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

Erstellt dynamisch einen Array und setzt das fünfte Element auf 8.

Wenn auf ein Array-Element Bezug genommen wird, dem noch nie ein Wert zugewiesen wurde, ergibt der Array-Ausdruck den Wert 0.

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

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

V4 wird auf den Wert 8 gesetzt.

Ausdruckskomponenten

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

Wenn dem sechsten Element von V3 noch nie ein Wert zugewiesen wurde, wird V5 auf 0 gesetzt.

Ebenso wie andere Array-Typen können Ausdrücke innerhalb von eckigen Klammern verwendet werden.

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

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

V4 wird auf den Wert 8 gesetzt.

Variablen-Arrays können mehrere Dimensionen haben.

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

V6 wird ein 2-mal-3-dimensionaler Array zugewiesen, wobei gilt: V6[1, 1] = 4, V6[1, 2] = 7, V6[1, 3] = 2, V6[2, 1] = 9, V6[2, 2] = 2 und V6[2, 3] = 6.

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

V7 wird auf den Wert 9 gesetzt.

Variablen-Arrays können negative Indizes haben:

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

Der -3. Index von Array V8 wird auf 5 gesetzt.

Array-Zuweisungen überschreiben vorherige Werte:

```
ZUWEISEN/V8="Hallo"
```

Der Variablen V8 entspricht die Zeichenfolge "Hallo".

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

V8 ist nun nicht mehr vom Typ "Zeichenfolge", sondern vom Typ "Array", dessen zweitem Element der Wert 5 zugewiesen wird.

```
ZUWEISEN/V8=9
```

V8 ist nun nicht mehr vom Typ "Array", sondern vom Typ "Ganzzahl" mit dem Wert 9.

Arrays können aus mehreren Typen erstellt werden:

```
ZUWEISEN/V9=Array("Hallo",3,2.9,{ELEM1})
```

Erstellt Array V9 mit 4 Elementen. Das erste Element ist eine Zeichenfolge, das zweite eine Ganzzahl, das dritte eine reelle Zahl, und das vierte Element ist ein Zeiger auf ELEM1.

Arrays können vergrößert werden, um mehr Elemente aufnehmen zu können:



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

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

Die erste Anweisung erstellt den Anfangsarray V10 mit 3 Elementen (3, 1 und 5). Die zweite Anweisung erhöht das Array in V10 um ein Element und gibt dem letzten Element den Wert 7.

Operatoren für Ausdrücke

PC-DMIS hält folgende Basisoperatoren für Sie bereit:

+ Addition: *<Ausdruck> + <Ausdruck>*

Addiert zwei Ausdrücke. Im Fall von Zeichenfolgen werden diese verkettet.

- Subtraktion: *<Ausdruck> - <Ausdruck>*

Subtrahiert den zweiten Ausdruck vom ersten Ausdruck.

* Multiplikation: *<Ausdruck> * <Ausdruck>*

Multipliziert zwei Ausdrücke.

/ Division: *<Ausdruck> / <Ausdruck>*

Dividiert den ersten Ausdruck durch den zweiten Ausdruck.

^ Potenzierung: *<Ausdruck> ^ <Ausdruck>*

Potenziert den ersten Ausdruck mit dem zweiten Ausdruck.

% Modulo: *<Ausdruck> % <Ausdruck>*

Gibt den Rest aus der Division des ersten Ausdrucks durch den zweiten zurück.

- Additive Umkehrung *-<Ausdruck>*

Gibt die additive Umkehrung des Ausdrucks zurück.

Ausdruckskomponenten

! Logisches Nicht: *!<Ausdruck>*

Gibt die logische Nicht-Funktion des Ausdrucks zurück.

== Gleich zu: *<Ausdruck> == <Ausdruck>*

Wird zu 1 ausgewertet, wenn die Ausdrücke gleich sind. Ergibt andernfalls 0. (Hier werden zwei Gleichheitszeichen verwendet, um den Zuweisungsoperator = in der Zuweisungsanweisung zu unterscheiden).

<> Nicht gleich zu: *<Ausdruck> <> <Ausdruck>*

Ergibt 1, wenn Ausdrücke nicht gleich sind. Ergibt andernfalls 0.

> Größer als: *<Ausdruck> > <Ausdruck>*

Ergibt 1, wenn Ausdruck 1 größer ist als Ausdruck2. Ergibt andernfalls 0.

>= Größer als oder gleich:: *<Ausdruck> >= <Ausdruck>*

Ergibt 1, wenn Ausdruck 1 größer oder gleich Ausdruck 2 ist. Ergibt andernfalls 0.

< Kleiner als: *<Ausdruck> < <Ausdruck>*

Ergibt 1, wenn Ausdruck 1 kleiner ist als Ausdruck2. Ergibt andernfalls 0.

<= Kleiner als oder gleich: *<Ausdruck> <= <Ausdruck>*

Ergibt 1, wenn Ausdruck 1 kleiner oder gleich Ausdruck 2 ist. Ergibt andernfalls 0.

UND Logisches Und: *<Ausdruck> UND <Ausdruck>*

Wird zu 1 ausgewertet, wenn beide Ausdrücke nicht zu 0 ausgewertet werden. Ergibt andernfalls 0.

ODER Logisches Oder: *<Ausdruck> ODER <Ausdruck>*。

Ergibt 1, wenn beide Ausdrücke nicht 0 ergeben. Ergibt andernfalls 0.

() Klammern: *(<Ausdruck>)*

Gibt dem Ausdruck innerhalb der Klammern Priorität bei der Auswertung.

Priorität

Ausdrücke werden anhand folgender Prioritäten ausgewertet. (Die Liste beginnt mit der höchsten Priorität und endet mit der Niedrigsten.)

Höchste Priorität

- Operanden
- (monadisches Minus), !, (), Funktionen (i.e. ABS, COS, STR, LEN, CROSS usw.)
- ^
- *, /, %
- +, -
- ==, <>, <, <=, >, >=
- UND
- ODER

Niedrigste Priorität

Funktionen

Bei Funktionen handelt es sich um PC-DMIS-spezifische Ausdrücke oder benutzerdefinierte Ausdrücke, die Parameter annehmen und Ergebnisse zurückgeben. Die Parameter werden im Ausdruck eingesetzt, bevor der Ausdruck ausgewertet wird.

Liste "Funktionen"

Die folgende Liste in alphabetischer Reihenfolge enthält alle Funktionen, die der Ausdruckssprache von PC-DMIS zur Verfügung stehen.

- ABS (mathematisch)
- ACOS (mathematisch)
- ANGLEBETWEEN (Punkt)
- ARCSEGMENTENDINDEX (Verschiedenes)
- ARCSEGMENTSTARTINDEX (Verschiedenes)
- ARRAY (Array)
- ASIN (mathematisch)
- ATAN (mathematisch)
- CHR (Zeichenfolge)
- CONCAT (string)
- COS (mathematisch)
- CROSS (Punkt)
- DEG2RAD (mathematisch)

Ausdruckskomponenten

- DELTA (Punkt)
- DIST2D (Dateizeiger)
- DIST3D (Dateizeiger)
- DOT (Punkt)
- ELEMENT (Zeichenfolge)
- EOF (Verschiedenes)
- EOL (Verschiedenes)
- EQUAL (Array)
- EQUAL (Zeichenfolge)
- EXP (mathematisch)
- FORMAT (Zeichenfolge)
- FUNCTION (Funktion)
- GETCOMMAND (Dateizeiger)
- GETPROGRAMINFO (Zeichenfolge)
- GETROTABDATA (Verschiedenes))
- GETSETTING (Zeichenfolge)
- GETTEXT (Zeichenfolge)
- GETTRACEVALUE (Zeichenfolge)
- IF (Verschiedenes)
- INDEX (Zeichenfolge)
- ISIOCHANNELSET (Verschiedenes)
- LEFT (Zeichenfolge)
- LEN (Array)
- LEN (Zeiger)
- LEN (Zeichenfolge)
- LINESEGMENTENDINDEX (Verschiedenes)
- LINESEGMENTSTARTINDEX (Verschiedenes)
- LN (mathematisch)
- LOG (mathematisch)
- LOWERCASE (Zeichenfolge)
- MAX (Array)
- MID (Zeichenfolge)
- MIN (Array)
- MPOINT (Punkt)
- ORD (Zeichenfolge)
- PCDMISAPPLICATIONPATH (Zeichenfolge)
- PCDMISUSERHIDDEN DATAPATH (Zeichenfolge)
- PCDMISUSERVISIBLE DATAPATH (Zeichenfolge)
- PCDMISSYSTEMHIDDEN DATAPATH (Zeichenfolge)
- PCDMISSYSTEMVISIBLE DATAPATH (Zeichenfolge)

- PCDMISSYSTEMREPORTINGPATH (Zeichenfolge)
- PROBEDATA (verschiedenes)
- QUALTOOLDATA (verschiedenes)
- RAD2DEG (mathematisch)
- RIGHT (Zeichenfolge)
- ROUND (mathematisch)
- SETROTABDATA (verschiedenes)
- SIN (mathematisch)
- SQRT (mathematisch)
- SYSTEMDATE (Zeichenfolge)
- SYSTEMTIME (Zeichenfolge)
- SYSTIME (Zeichenfolge)
- TAN (mathematisch)
- TUTORELEMENT (verschiedenes)
- UNIT (Punkt)
- UPPERCASE (Zeichenfolge)

Zeichenfolgenfunktionen

Folgende Funktionen werden zusammen mit Textzeichenfolgen verwendet:

CHR

Zeichenkonvertierung: *CHR(<Ganzzahl>)*

Diese Funktion gibt eine Zeichenfolge zurück, die aus dem Zeichen besteht, das dem angegebenen ASCII-Dezimalwert entspricht.

CONCAT

Diese Funktion verkettet alle in den Ausdrücken 1 bis N angegebenen Zeichenketten zu einer einzigen Zeichenkette: `CONCAT (<Ausdruck1>, <Ausdruck2>, & <AusdruckN>)`

ELAPSEDEXECUTIONTIME

Formatierte, verstrichene Ausführungszeit: *ELAPSEDEXECUTIONTIME()*

Mit dieser Funktion wird die Zeit angezeigt, die seit der Ausführung der Messroutine oder Mini-Routine vergangen ist. Die vergangene Ausführungszeit ist die Zeit der Ausführung des CNC-Teils der Ausführung. Die Zeit für vom Benutzer

benötigter Pausen wird nicht hinzugefügt. Zu solchen Pausen gehören Ausführungspausen während der Ausführung von Kommentaren oder PC-DMIS-Meldungen sowie Fehlermeldungen, die die Ausführung ganz abbrechen können.

Die verstrichene Ausführungszeit kann zu jedem beliebigen Zeitpunkt in der Messroutine oder Miniroutine aufgezeichnet werden, indem die Funktion einer Variablen zugewiesen wird, wie beispielsweise so:



```
ZUWEISEN/V1=ELAPSEDEXECUTIONTIME()
```

Das Standardformat der zurückgegebenen Zeit ist "hh:mm:ss". Sie können die verstrichene Ausführungszeit auch in anderen Formaten messen:

- Verwenden Sie
ASSIGN/V1=FORMAT(ELAPSEDEXECUTIONTIME(),"hh:mm:ss") oder
ASSIGN/V1=ELAPSEDEXECUTIONTIME(), um die Zeit in Stunden, Minuten und Sekunden zu erhalten.
- Verwenden Sie
ASSIGN/V1=FORMAT(ELAPSEDEXECUTIONTIME(),"mm:ss"), um die Zeit in Minuten und Sekunden zu erhalten.
- Verwenden Sie
ASSIGN/V1=FORMAT(ELAPSEDEXECUTIONTIME(),"ss"), um die Zeit in Sekunden zu erhalten.

ELEMENT

Lage begrenzter Teilzeichenfolgen: *ELEMENT(<Ganzzahl>, <Zeichenfolge1>, <Zeichenfolge2>)*

Diese Funktion gibt die n-te (n = <Ganzzahl>) Teilzeichenfolge in Zeichenfolge2 zurück, wobei Zeichenfolge1 als begrenzender Text dient, durch den die Elemente in Zeichenfolge2 unterteilt werden.



Angenommen Zeichenfolge2 ist "6, 12, 8, 4, 5", und Zeichenfolge1 besteht aus ",". Dann lauten die 5 Elemente, die mit dem ELEMENT-Befehl einzeln extrahiert werden können "6", "12", "8", "4" und "5".

EQUAL

Zeichenfolgevergleich ohne Berücksichtigung von Groß-/Kleinschreibung:
EQUAL(<Zeichenfolge1>, <Zeichenfolge2>)

Diese Funktion vergleicht zwei Zeichenfolgen (ignoriert Groß-/Kleinschreibung), um zu bestimmen, ob sie gleich sind. Sie gibt eine auf den Wert 1 gesetzte Ganzzahl zurück, wenn die Zeichenfolgen gleich sind, bzw. 0, wenn sie nicht gleich sind.

FORMAT

Format: **FORMAT(<Zeichenfolge>, <Ganzzahl, Gleitkommazahl oder Punkt>)**

Diese Funktion nimmt zwei Ausdrücke und gibt eine mit der *sprintf*-Funktion in C++ vergleichbare, formatierte Zeichenfolge zurück.

- Ausdruck 1 sollte vom Typ *Zeichenfolge* sein und einen oder drei Formatspezifizierer enthalten. Wenn es sich um einen anderen Typ handelt, versucht der Ausdrucksauswerter die Umwandlung in eine Zeichenfolge zu erzwingen. Die Zeichenfolge sollte *einen* Formatspezifizierer enthalten, wenn Ausdruck 2 vom Typ "Ganzzahl" oder "Double" ist und *drei* Formatspezifizierer (siehe die folgenden Abschnitte), wenn es sich bei Ausdruck 2 um den Typ "Punkt" handelt.
- Ausdruck 2 muss vom Typ *Ganzzahl*, *Double* oder *Punkt* sein. Wird ein anderer Typ verwendet, hat der Ausdruck den Wert 0.

Formatspezifizierer für die Funktion FORMAT:

Die Formatspezifizierer sollten dieselbe Syntax haben, wie die in der *sprintf*-Funktion der Programmiersprache C++ verwendeten Formatspezifizierer.

Ein Formatspezifizierer besteht aus optionalen und obligatorischen Feldern und hat folgende Syntax:

%[Attribute] [Breite] [.Genauigkeit] Typ

Jedes Feld des Formatspezifizierers besteht entweder aus einem Einzelzeichen oder einer Zahl, die eine bestimmte Formatoption bedeuten. Der einfachste Formatspezifizierer basiert einfach auf einem Prozentzeichen und einem Schriftzeichen (beispielsweise %d). Wenn nach dem Prozentzeichen ein Schriftzeichen folgt, das als Formatfeld bedeutungslos ist, wird das Zeichen nach STDOUT kopiert. Um beispielsweise ein Prozentzeichen-Schriftzeichen auszugeben, verwenden Sie %%.

Die optionalen Felder Attribut, Breite und Genauigkeit, die vor dem Schriftzeichen erscheinen, steuern andere Aspekte der Formatierung. Sie werden nachfolgend beschrieben.

Attribute

Diese *optionalen Zeichen* steuern die Ausgabeanpassung und den Ausdruck von Zeichen, Leerstellen, Dezimalpunkten und Oktal-/Hexadezimalvorzeichen. Ein Formatspezifizierer kann mehrere Attribute aufweisen.

Folgende Attribute sind zulässig:

–

Bedeutung: Linksausrichtung des Ergebnisses an der vorgegebenen Feldbreite.

Standard: Rechtsausrichtung.

+

Bedeutung: Stellen Sie dem Ausgabewert ein Zeichen (+ oder –) voran, wenn es sich beim Ausgabewert um einen mit einem Vorzeichen versehenen Wert handelt.

Standard: Vorzeichen erscheinen nur bei Werten mit negativen Vorzeichen (–).

0

Bedeutung: Steht als Vorzeichen vor der Breite eine 0, werden Nullen hinzugefügt, bis die Mindestbreite erreicht wird. Erscheint eine 0 und –, dann wird die 0 ignoriert. Wird der Wert 0 mit einem Ganzzahlformat (i, u, x, X, o, d) definiert, wird der Wert 0 ignoriert.

Standard: Kein Auffüllen.

Leerzeichen (' ')

Bedeutung: Ein Leerzeichen vor den Ausgabewert setzen, wenn der Ausgabewert vorzeichenbehaftet und positiv ist. Das Leerzeichen wird ignoriert, wenn sowohl Leerzeichen als auch + Attribute erscheinen.

Standard: Kein Leerzeichen.

#

Bedeutung 1: In Kombination mit den Typen o, x oder X, versieht das # Attribut jeden Ausgabewert, der ungleich Null ist, mit dem Vorzeichen 0, 0x beziehungsweise 0X.

Standard 1: Kein Vorzeichen.

Bedeutung 2: In Kombination mit den Typen e, E oder f, erzwingt das # Attribut in allen Fällen ein Dezimalkomma in jedem Ausgabewert.

Standard 2: Das Dezimalkomma erscheint nur vor Ziffern.

Bedeutung 3: In Kombination mit dem Format g oder G erzwingt das # Attribut in allen Fällen ein Dezimalkomma in jedem Ausgabewert und verhindert das Abschneiden der hinteren Nullstellen.

Standard 3: Das Dezimalkomma erscheint nur vor Ziffern. Die hinteren Nullstellen werden entfernt. Wird in Kombination mit d, i oder u ignoriert.

breite

Mit diesem zweiten optionalen Feld oder Argument wird die Mindestanzahl der auszugebenden Zeichen gesteuert. Es handelt sich bei diesem Wert um eine nicht negative Dezimalzahl (Ganzzahl).

- Ist die Zahl der Zeichen im Ausgabewert geringer als die vorgegebene Breite, werden rechts oder links neben den Werten so viele Leerstellen eingefügt — abhängig davon, ob das Attribut – (für die Linksausrichtung) angegeben ist — bis die Mindestbreite erreicht ist.
- Steht als Vorzeichen vor der Breite eine 0, werden Nullen hinzugefügt, bis die Mindestbreite erreicht wird (was bei linksausgerichteten Zahlen störend ist).
- Die Definition der Breite ist niemals Ursache für ein Abschneiden der Werte. Wenn die Zeichenanzahl im Ausgabewert die vorgegebene Breite überschreitet, oder wenn keine Breite vorgegeben wurde, werden alle Zeichen des Werts gedruckt (je nach dem, wie die weiter unten aufgeführten Genauigkeitsvorgaben sind).

GENAUIGKEIT

Dieses dritte optionale Feld oder Argument steuert die Anzahl der auszugebenden Zeichen, die Anzahl der Dezimalstellen oder die Anzahl signifikanter Ziffern. Anders als bei der Definition der Breite kann die Genauigkeitsdefinition entweder zu einem Abschneiden des Ausgabewerts oder zur Abrundung eines Gleitpunktwerts führen. Es handelt sich bei diesem Wert um eine nicht negative Dezimalzahl (Ganzzahl) nach einem Punkt (.).

Typ

Dieses obligatorische Zeichen bestimmt, ob es sich bei dem zugehörigen Argument um den Typ "Ganzzahl", "Double" oder "Punkt" handelt. Die Liste verfügbarer Typen beinhaltet:

d - vorzeichenbehaftete Dezimalganzzahl

i - vorzeichenbehaftete Dezimalganzzahl

o - vorzeichenlose Oktalganzzahl

u – vorzeichenlose Dezimalganzzahl

x - vorzeichenlose HexGanzzahl, in Kombination mit "abcdef"

x - vorzeichenlose HexGanzzahl, in Kombination mit "ABCDEF"

e - doppelt genauer Wert in Exponentialform [-]d.dddd e [sign]ddd

E - wie "e", nur dass E zur Einführung des Exponenten verwendet wird

f - doppelt genauer Wert in der Form [-]dddd.dddd

g - formatiert entweder in das Format e oder f, je nachdem, welches Format kompakter ist

G - wie "g", nur dass E zur Einführung des Exponenten verwendet wird

Beispiel für die FORMAT-Funktion

Dieses Beispiel zeigt verschiedene Anweisungen, welche die FORMAT-Funktion in einer Messroutine verwenden:

ZUWEISEN/V1=PROBEDATA ("VERSATZ")	V1 wird zum Typ "Punkt", der die Versätze des aktuellen Tasters darstellt. Unter Zugrundelegung der Werte aus der für dieses Beispiel verwendeten Messroutine, wird V1 zu: <-1.8898, 1.8898, 5.704>
ZUWEISEN/V3=FORMAT ("% .5f, % .5f, % .5f", V1)	V3 wird zum Typ "Zeichenfolge". Die Zeichenfolge wird auf Basis des Punktoobjekts der Variablen V1 formatiert. V3 enthält nun: - 1.88976, 1.88976, 5.70403
ZUWEISEN/V4=1,123456789	V4 wird zum Typ "Double".
ZUWEISEN/V5=FORMAT ("% .5f", V4)+FORMAT ("% .6f", V4)+FORMAT ("% .7f", V4)+FORMAT ("% .8f", V4)	V5 wird zum Typ "Zeichenfolge" mit dem folgenden Wert: 1.12346 1.123457 1.1234568 1.12345679
ZUWEISEN/V6A="V4 wird der Wert: "+FORMAT ("% .8f", V4)	V6 wird zum Typ "Zeichenfolge" mit dem Wert: V4 wird der Wert: 1.12345679
ZUWEISEN/V6B=FORMAT("V4 wird der Wert: % .8f", V4)	Das Ergebnis des Ausdrucks bleibt dasselbe wie für V6A oben.

ZUWEISEN/V7=4444	V7 wird zum Typ "Double", da alle Zahlen als doppelte Werte interpretiert werden, solange nicht die Umwandlung in Ganzzahlen erzwungen wird.
ZUWEISEN/V8=FORMAT ("%o", INT (V7))	V8 wird zum Typ "Zeichenfolge" mit dem folgenden Wert: 10534
ZUWEISEN/V9=FORMAT ("%u", INT (-1))	V9 wird zum Typ "Zeichenfolge" mit dem folgenden Wert: 4294967295
ZUWEISEN/V10=FORMAT ("%x", INT (2143))	V10 wird zum Typ "Zeichenfolge" mit dem folgenden Wert: 85f
ZUWEISEN/V11=FORMAT ("%X", INT (9567))	V11 wird zum Typ "Zeichenfolge" mit dem folgenden Wert: 255F
ZUWEISEN/V12=FORMAT ("%e", 0.0005432)	V12 wird zum Typ "Zeichenfolge" mit dem folgenden Wert: 5.432000e-004
ZUWEISEN/V13=FORMAT ("%E", 145.3421)	V13 wird zum Typ "Zeichenfolge" mit dem folgenden Wert: 1.453421E+002
ZUWEISEN/V14=FORMAT (" , %6d , " , INT (1))	<div>  <p>V14 wird zum Typ "Zeichenfolge" mit dem folgenden Wert: , 1 ,</p> </div>
ZUWEISEN/V15=FORMAT (" , %-6d , " , INT (1))	<div>  <p>V15 wird zum Typ "Zeichenfolge" mit dem folgenden Wert: ,1 ,</p> </div>

GETSETTING

Mit dieser Funktion können Sie verschiedene PC-DMIS-Einstellungen unter Zugrundelegung der eingefügten Zeichenfolgenparameter zurückgeben.

GETSETTING(<Zeichenfolge>)

Folgende Zeichenfolgenparameter stehen zur Auswahl:

- "DCC Mode" – Wenn sich PC-DMIS im CNC-Modus befindet, wird eine 1 zurückgegeben; ansonsten eine 0.
- "Manual Mode" – Wenn sich PC-DMIS im manuellen Modus befindet, wird eine 1 zurückgegeben; ansonsten eine 0.
- "Current Alignment" – Gibt eine Zeichenfolge der aktuellen Ausrichtung zurück.
- "Current Workplane" – Gibt eine Zeichenfolge der aktuellen Arbeitsebene zurück.
- "Workplane Value" – Gibt einen numerischen Wert der aktuellen Arbeitsebene zurück.
- "PreHit" – Gibt den aktuellen Anfahrabstand als eine doppelt genaue Zahl zurück.
- "Retract" – Gibt den aktuellen Rückfahrabstand als eine doppelt genaue Zahl zurück.
- "Check" – Gibt den aktuellen Prüfwert als eine doppelt genaue Zahl zurück.
- "Touch Speed" – Gibt den aktuellen Messgeschwindigkeitswert als eine doppelt genaue Zahl zurück.
- "Move Speed" – Gibt den aktuellen Bewegungsgeschwindigkeitswert als eine doppelt genaue Zahl zurück.
- "Fly Mode" – Gibt den Wert 1 zurück, wenn sich PC-DMIS im Fly-Modus befindet, andernfalls "0".
- "Ph9 present" – Gibt den Wert 1 zurück, wenn eine DSE vorhanden ist, andernfalls 0.
- "Manual CMM" – Gibt den Wert 1 zurück, wenn es sich um ein manuelles KMG handelt, andernfalls "0".
- "LangStr(<Nummer oder ID>)" – Gibt eine aus einer Zahl oder einer der folgenden IDs bestehende Zeichenfolge der PC-DMIS-Ressourcen in der aktuellen Sprache wieder:

"Ja", "Nein", "Bediener", "Protok", "Eingabe", "Dok", "JaNein", "Auslesen", "Intern", "Extern", "Kart", "Polr", "Außen", "Innen", "Kleinste_Quad", "Min", "Max", "Hüllkr", "Fester_Rad", "Arbeitsebene", "X-Achse", "Y-Achse", "Z-Achse", "Xplus", "Xminus", "YPlus", "YMinus", "ZPlus", "ZMinus", "Punkt",

"Ebene", "Linie", "Kreis", "Kugel", "Zylinder", "Langloch", "Rechteckloch", "Kegel" oder "Keins".

Wenn der Wert, den Sie verwenden, eine positive Zahl ist, extrahiert PC-DMIS die Zeichenfolge aus der Datei "resource.dll". Verwenden Sie eine negative Zahl, extrahiert PC-DMIS die Zeichenfolge aus der Datei "strings.dll" (Tabelle der Zeichenfolgen).

- "Extended Sheet Metal" – Gibt den Wert 1 zurück, wenn das Kontrollkästchen **Erweiterte Blechoptionen einblenden** im Dialogfeld **Setup-Optionen** aktiviert ist, andernfalls 0.
- "LastHitMove(X)" – Gibt den Wert X des allerletzten Befehls MESSPUNKT /BASIC oder BEWEGEN/PUNKT zurück. Hierfür muss PC-DMIS in den CNC-Modus versetzt werden.
- "LastHitMove(Y)" – Gibt den Wert Y des allerletzten Befehls MESSPUNKT /BASIC oder BEWEGEN/PUNKT zurück. Damit dieser Parameter funktioniert, muss sich PC-DMIS im CNC-Modus befinden.
- "LastHitMove(Z)" – Gibt den Wert Z des allerletzten Befehls MESSPUNKT /BASIC oder BEWEGEN/PUNKT zurück. Damit dieser Parameter funktioniert, muss sich PC-DMIS im CNC-Modus befinden.

Sie können die Funktion GETSETTING verwenden, um festzustellen, ob sich PC-DMIS im manuellen oder im DCC-Modus befindet (siehe unten):

`ZUWEISEN/DCCMODEVAR=GETSETTING("DCC Modus")` - Dies weist der Variablen DCCMODEVAR den Wert 1 zu, wenn sich PC-DMIS im CNC-Modus befindet, andernfalls den Wert 0.

`ZUWEISEN/MANMODEVAR=GETSETTING("Manueller Modus")` - Dies weist der Variablen MANMODEVAR den Wert 1 zu, wenn sich PC-DMIS im Manuellen Modus befindet, andernfalls den Wert 0.

Sie können die Funktion GETSETTING verwenden, um die aktuelle Arbeitsebene zu bestimmen (siehe unten):

`ZUWEISEN/ARBEITSEBENE_ID = GETSETTING("Aktuelle Arbeitsebene")` - Dies weist der Variablen ARBEITSEBENE_ID den Zeichenfolgenwert der aktuellen Arbeitsebene (ZPLUS, ZMINUS etc.) zu.

`ZUWEISEN/ARBEITSEBENE_WERT=GETSETTING("Arbeitsebenenwert")` - Dies weist der Variablen ARBEITSEBENE_WERT einen numerischen Wert für die Arbeitsebene zu. Folgende Werte sind mit den Arbeitsebenen

verknüpft: ZPLUS = 0, ZMINUS = 3, XPLUS = 1, XMINUS = 4, YPLUS = 2
oder YMINUS = 5.

GETTEXT

Diese Funktion gibt den aktuellen Text vom angegebenen Datenfeld zurück:
`GETTEXT(<Zeichenfolge oder Ganzzahl>, <Ganzzahl>, <Zeiger>)`

Diese Funktion besitzt drei Felder.

Erstes Feld—Datenfeldnummer oder Beschreibung

Das erste Feld kann entweder eine beschreibende Zeichenfolge des Datenfeldes, in der Abbildung unten angegeben durch (A), oder die Datenfeldnummer, in der Abbildung unten angegeben durch (C), sein.



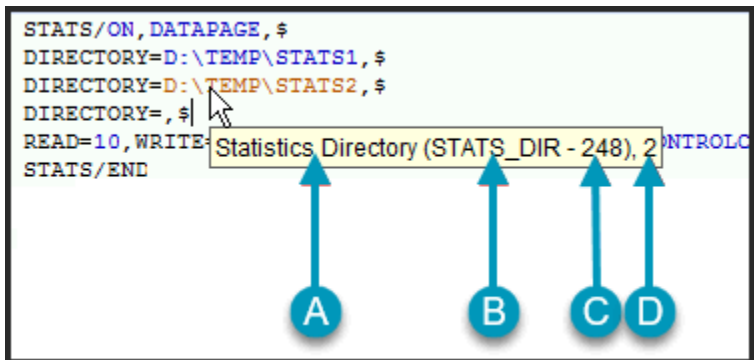
Element (B) in der unten stehenden Abbildung wird in dieser Funktion nicht verwendet, kommt aber von Zeit zu Zeit bei der Automation oder bei Protokollausdrücken zum Einsatz.

So erhalten Sie diese Werte:

1. Versetzen Sie PC-DMIS in den Befehlsmodus. Klicken Sie mit der rechten Maustaste auf eine beliebige Stelle im Bearbeitungsfenster. Es erscheint ein Kontextmenü.
2. Wählen Sie im Kontextmenü **Wechsle Popup-Anzeige** und dann **Datentypangaben** aus.
3. Setzen Sie die Maus auf ein Datenfeld im Bearbeitungsfenster. Es erscheinen die Datentypbeschreibung und -nummer sowie der Datentypindex des betreffenden Datenelements.



Die Typbeschreibung kann je nach Sprache verschieden sein. Wenn die Messroutine auf einer PC-DMIS-Version ausgeführt wird, die auf eine andere Sprache gesetzt ist, benutzen Sie besser die Typnummer.



Beispiel-Datentypangaben, die folgendes darstellen: (A) Typbeschreibung (B) Typ-Zeichenfolgenkennung, (C) Typnummer und (D) Typindex

Zweites Feld—Typ Index

Das zweite Feld ist der Typindex, der in der obigen Abbildung als (D) angegeben wird. Dieses Feld ist normalerweise Null, es sei denn, Sie verfügen über mehrere Instanzen desselben Feldtyps im selben Befehl wie beispielsweise mehrere VERZEICHNIS-Felder, dargestellt in der oben stehenden Abbildung. Der richtige Wert lässt sich mit derselben wie für das erste Feld beschriebenen Methode feststellen.

Drittes Feld—Befehlszeiger

Das dritte Feld ist ein Befehlszeiger. Es verweist auf den Befehl, der das Feld enthält, aus dem der Text stammt. Sie können entweder die Befehlszeiger-Notation (d. h. {F15}), um dieses Feld zu spezifizieren, oder Sie können den Ausdruck GETCOMMAND verwenden, wie in diesem Beispiel gezeigt:



`ZUWEISEN/V1=GETTEXT("Besteinpassungs-Berechnungstyp",0,{F15})` - Dieser Befehl weist V1 den aktuellen Wert des Umschaltfeldes "Besteinpassungs-Berechnungstyp" von Element F15 zu.

`ZUWEISEN/V2=GETCOMMAND("Kommentar","TOP",1)` - V2 wird ein Zeiger zum ersten Kommentar vom Anfang der Messroutine aus gesehen zugewiesen.

`ZUWEISEN/V3=GETTEXT("Kommentartyp",1,V2)` - V3 wird der Wert des Umschaltfeldes "Kommentartyp" zugewiesen. Ist der erste Kommentar in der Messroutine ein dem Bediener anzuzeigender Kommentar, erhält V3 als Wert die Zeichenfolge "BEDIENER".

Weitere Informationen zum Ausdruck `GETCOMMAND`, der zur Einstellung eines Zeigers auf einen Befehl verwendet wird, finden Sie unter "Zeigerfunktionen".

GETTEXT

Diese Funktion gibt den aktuellen Text vom angegebenen Datenfeld zurück:

`GETTEXT(<Zeichenfolge oder Ganzzahl>, <Ganzzahl>,
<Zeichenfolge>, <Zeiger>)`

Diese Funktion besitzt vier Felder.

Erstes Feld—Datenfeldnummer oder Beschreibung

Das erste Feld kann entweder eine Beschreibung des Datenfelds (Zeichenfolge) oder die Datenfeldnummer sein, wie in Punkt (A) in der Abbildung unten angegeben.



Wenn Sie anstelle des numerischen Bezeichners (Punkt (A) in der Abbildung unten) den Zeichenfolge-Bezeichner verwenden, wandelt PC-DMIS diesen automatisch in den richtigen numerischen Wert um.

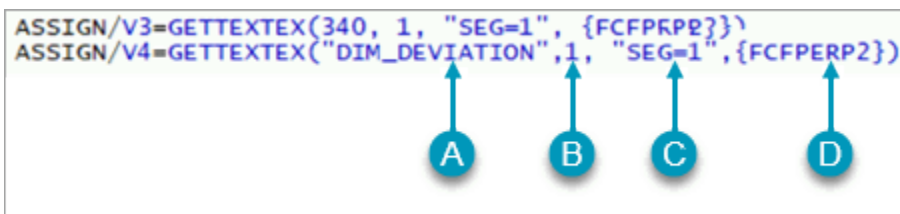
Wenn Sie z. B. die Zeichenfolge "DIM_DEVIATION" übergeben, wandelt PC-DMIS ihn intern in den numerischen Wert 340 um. Wenn Sie dann mit dem Mauszeiger über den Befehl im Bearbeitungsfenster fahren, zeigt der Popup-Befehl sowohl den Textstring als auch den tatsächlichen numerischen Bezeichnerwert an. Wenn Sie in diesem Beispiel mit dem Mauszeiger über den Befehl im Bearbeitungsfenster fahren, zeigt das Befehls-Popup `(DIM_DEVIATION - 340), 1, SEG=1` an.

Sie können auch nur den numerischen Wert übergeben, wenn dieser bekannt ist.

So erhalten Sie diese Werte:

1. Versetzen Sie PC-DMIS in den Befehlsmodus und klicken Sie dann mit der rechten Maustaste auf eine beliebige Stelle im Bearbeitungsfenster. Es erscheint ein Kontextmenü.
2. Wählen Sie im Kontextmenü **Wechsle Popup-Anzeige** und dann **Datentypangaben** aus.
3. Setzen Sie die Maus auf ein Datenfeld im Bearbeitungsfenster. Es erscheinen die Datentypbeschreibung und -nummer sowie der Datentypindex des betreffenden Datenelements. Bewegen Sie den Mauszeiger über einen erweiterten D_Type, um die Inhaltszeichenfolge nach dem Doppelpunkt anzuzeigen.

Die Typbeschreibung kann je nach Sprache verschieden sein. Wenn die Messroutine auf einer PC-DMIS-Version ausgeführt wird, die auf eine andere Sprache gesetzt ist, benutzen Sie besser die Typnummer.



Beispiel für Datentypinformationen mit (A) Zeichenfolge oder numerischer Kennung, (B) Typ-Index, (C) Inhalts-String und (D) Befehlszeiger

Zweites Feld - Typindex

Das zweite Feld ist der Typindex, der in der obigen Abbildung als (B) angegeben wird. Dieses Feld ist normalerweise Null, es sei denn, Sie verfügen über mehrere Instanzen desselben Feldtyps im selben Befehl wie beispielsweise mehrere VERZEICHNIS-Felder, dargestellt in der oben stehenden Abbildung. Der richtige Wert lässt sich mit derselben wie für das erste Feld beschriebenen Methode feststellen.

Drittes Feld—Inhaltszeichenfolge

Das dritte Feld ist die Inhaltszeichenfolge des erweiterten D_TYPE, im obigen Bild als C gekennzeichnet.

Viertes Feld—Befehlszeiger

Das vierte Feld ist ein Befehlszeiger, in der obigen Abbildung mit D bezeichnet. Er verweist auf den Befehl, aus dem der Ausdruck Daten extrahiert. Sie können entweder die Befehlszeiger-Notation (d. h. {FCFPERP2}) oder den Ausdruck GETCOMMAND verwenden, wie in diesem Beispiel gezeigt:

```
ZUWEISEN/V1=GETTEXT("DIM_DEVIATION",1,"SEG=1",{FCFPERP2}) - Dieser Befehl weist V1 den aktuellen Wert der Abweichung von Element 1, Segment 1, Merkmal FCFPERP2 zu.
```

Weitere Informationen zum Ausdruck GETCOMMAND, der zur Einstellung eines Zeigers auf einen Befehl verwendet wird, finden Sie unter "Zeigerfunktionen".



Die Funktion GETTEXT fügt Unterstützung für die Erweiterten DTypes hinzu, die eine Inhaltszeichenfolge enthalten. Derzeit verwenden nur die PC-DMIS-Befehle für geometrische Toleranzen Erweiterte DTypes.

GETPROGRAMINFO

Diese Funktion gibt die Informationen der Messroutine basierend auf den Parametern im folgenden Ausdruck zurück:

```
GETPROGRAMINFO(<Zeichenfolge>, <Optionale Zeichenfolge>)
```

Diese Funktion verwendet max. zwei Parameter. Für die meisten Elemente benötigen Sie lediglich den ersten Parameter. Bei diesen Zeichenfolge wird nicht zwischen Groß- und Kleinbuchstaben unterschieden.

Erstes Feld—Zeichenfolge

Das erste Feld ist eine Zeichenfolge, die bestimmt, was zurückgegeben wird.

CADMODELFILE - Gibt den vollständigen Pfad zum Dateinamen des CAD-Modells zurück, das Sie in die Messroutine importiert haben.

CADMODELFILENAME - Liefert nur den Namen des CAD-Modells (nicht den Pfad), das Sie in die Messroutine importiert haben.

DATE - Gibt das aktuelle Datum zurück.

DRAWING - Wie REVISION, gibt auch die Revisionsnummer gemäß der Überschrift zurück.

ELAPSEDTIME - Gibt die Zeit zurück, die seit dem Start der Ausführung vergangen ist.

FILENAME - Gibt den Dateinamen (*.PRG) der Messroutine zurück.

NUMMEAS - Gibt die Anzahl der ausgeführten Merkmale zurück.

NUMOOT - Gibt die Anzahl der ausgeführten Elemente außerhalb der Toleranz zurück.

PARTNAME - Gibt den Werkstücknamen aus der Kopfzeile der Messroutine zurück.

PARTPATH - Gibt den vollständigen Pfad der Datei der Messroutine zurück.

PCDMISVERSION - Gibt eine Zeichenfolge mit der momentan installierten PC-DMIS-Version zurück.

PRGSCHEMA - Gibt eine Ganzzahl mit der PC-DMIS-Schemanummer der Messroutine zurück. Hierbei handelt es sich um einen internen Wert, den PC-DMIS nutzt, um die serialisierte Befehle und Optionen zu markieren.

PRGVERSION - Gibt eine Zeichenfolge mit der PC-DMIS-Versionsnummer der Messroutine zurück. Somit können Sie eine Messroutine zum Zwecke der Kompatibilität in einer bestimmten Version speichern. Siehe auch „Speichern unter“ im Abschnitt „Verwenden von grundlegenden Dateioptionen“.

PROBEFILE - Gibt den Namen der aktuell verwendeten Tasterdatei zurück.

REPORTNAME - Gibt den Namen der aktuellen Ausgabedatei zurück.

REVISION - Gibt die Revisionsnummer gemäß der Überschrift zurück.

SERIALNUM - Gibt die Seriennummer gemäß der Überschrift zurück.

SEQNUM - Wie STATSCOUNT, gibt auch den aktuellen Statistikstand zurück.

SHRINK - Gibt den globalen Maßstab zurück.

STATSCOUNT - Gibt den aktuellen Statistikstand zurück.

TEMP - Gibt die Temperatur für die optionale, zweite Eingabezeichenfolge zurück. See "Zweites Feld—Optionale Zeichenfolge" weiter unten.

TIME - Gibt die aktuelle Zeit zurück.

TIPID - Gibt den Namen des aktuell verwendeten Tastspitze zurück.

Zweites Feld—Optionale Zeichenfolge

Das zweite Feld ist eine optionale Zeichenfolgeeingabe. Es wird nur benötigt, wenn TEMP im ersten Eingabefeld verwendet wird. Die folgenden möglichen Zeichenfolgen stammen aus dem Befehl Temperaturkompensation. Weitere Informationen finden Sie unter "Temperaturkompensation" im Abschnitt "Einstellungen".

HIGH_THRESHOLD - Gibt den oberen Grenzwert der Temperaturkompensation zurück

LOW_THRESHOLD - Gibt den unteren Grenzwert der Temperaturkompensation zurück

REF_TEMP - Gibt die Referenztemperatur zurück

TEMPP - Gibt die Temperatur am Werkstücksensor zurück.

TEMPX - Gibt die Temperatur am X-Achsen-Sensor zurück

TEMPY - Gibt die Temperatur am Y-Achsen-Sensor zurück

TEMPZ - Gibt die Temperatur am Z-Achsen-Sensor zurück

Beispiel

```

$$ NO, Dieses Programmbeispiel zeigt die Anzahl der
Merkmale und die Anzahl der Merkmal außerhalb der
Toleranz an.
    ZUWEISEN/V1=GETPROGRAMINFO("NUMMEAS")
    ZUWEISEN/V2=GETPROGRAMINFO("NUMOOT")
    KOMMENTAR/PROT
    "Merkmale Total: "+V1
    "Außerhalb Toleranz Total: "+V2
$$ NO, Dieses Programmbeispiel gibt die Temperatur am Z-
Achsen sensor zurück.
    ZUWEISEN/V3=GETPROGRAMINFO("TEMP", "TEMPZ")
    KOMMENTAR/PROT
    "Temperatur Z-Achse: "+V3

```

GETTRACEVALUE

Überwachungswert lesen: *GETTRACEVALUE(<Zeichenfolge>)*

Diese Funktion verwendet einen einzelnen Zeichenfolge-Parameter. Sie gibt einen Wert aus einem **TRACEFIELD**-Befehl in der Messroutine zurück.

<Zeichenfolge> steht für eine Zeichenfolge des Überwachungsnamens, dessen Wert zurückgegeben werden soll.

Wenn Sie mehrere Überwachungsfelder mit dem gleichen Namen besitzen, gibt diese Funktion den Wert für das neueste Überwachungsfeld über dieser Funktion zurück. Wenn ein Überwachungsfeld keinen Wert enthält, gibt diese Funktion den Wert 0 zurück. Im folgenden Beispiel lautet "Bediener" der Name des Überwachungsfeldes in der Messroutine:



```
ZUWEISEN/V2=GETTRACEVALUE("Bediener")
```

INDEX

Lage Teilzeichenfolge: *INDEX(<Zeichenfolge>, <Zeichenfolge>)*

Diese Position gibt die zweite Zeichenfolge innerhalb der ersten Zeichenfolge zurück. Der erste Buchstabe der Zeichenfolge ist 1. Ein Rückgabewert von 0 zeigt an, dass die Teilzeichenfolge nicht in der Zeichenfolge gefunden wurde.

Ein einfaches Beispiel dieser Funktion finden Sie im Thema "Beispielcode für 'Zeile lesen'" im Abschnitt "Verwenden der Dateieingabe / -ausgabe".

LASTEXECUTIONTIME

Formatierte letzte Ausführungszeit: *LASTEXECUTIONTIME()*

Mit dieser Funktion wird die letzte Ausführungszeit, die von PC-DMIS aufgezeichnet und in der Datei *<Name der Messroutine>.MiniRoutines.xml* gespeichert wurde, zurückgegeben. Die letzte Ausführungszeit erscheint im Dialogfeld **Ausführung**. Der Zeitraum wird im Format "hh:mm:ss" angegeben.

LINKS

Linke Anzahl an Zeichen von Zeichenfolge: *LEFT(<Zeichenfolge>, <n>)*

Diese Funktion gibt eine Zeichenfolge zurück und extrahiert dabei die durch den zweiten Ausdruck (n) definierte Anzahl von Zeichen, von links nach rechts aus dem ersten Ausdruck (Zeichenfolge).

Der erste Ausdruck (Zeichenfolge) wird zum Typ Zeichenfolge gezwungen. Der zweite Ausdruck (n) wird zum Typ Ganzzahl gezwungen.

Ein einfaches Beispiel dieser Funktion finden Sie im Thema "Beispielcode für 'Zeile lesen'" im Abschnitt "Verwenden der Dateieingabe / -ausgabe".

LEN

Länge der Zeichenfolge: *LEN(<Zeichenfolge>)*

Diese Funktion gibt die Anzahl von Zeichen in der Zeichenfolge zurück.

LOWERCASE

Zeichenfolge mit Kleinschreibung erstellen: *LOWERCASE(<Zeichenfolge>)*

Diese Funktion gibt eine Zeichenfolge zurück, die das Äquivalent der Ausdruckszeichenkette in Kleinbuchstaben ist.

MID

Mittlere n Zeichen einer Zeichenfolge: *MID(<Zeichenfolge>, <Ganzzahl>, <Optionale Ganzzahl>)*

Diese Funktion gibt eine Teilzeichenfolge zurück und extrahiert dabei die durch den dritten Parameter definierte Anzahl von n Zeichen aus der durch den ersten

Parameter definierten Zeichenfolge, beginnend an der durch den zweiten Parameter definierten Position. Wenn der dritte Parameter nicht definiert wird, werden alle Zeichen ab der Position bis zum Ende der Zeichenfolge zurückgegeben.

Ein einfaches Beispiel dieser Funktion finden Sie im Thema "Beispielcode für 'Zeile lesen'" im Abschnitt "Verwenden der Dateieingabe / -ausgabe".

ORD

Ordinalkonvertierung: *ORD(<Zeichenfolge>)*

Diese Funktion gibt den Ganzzahl-ASCII-Wert des ersten Zeichens der Zeichenfolge (0-255) zurück.

PCDMISAPPLICATIONPATH

Kompletter Pfad: *PCDMISAPPLICATIONPATH()*

Diese Funktion gibt eine Zeichenfolge mit dem kompletten Pfad des Installationsverzeichnisses von PC-DMIS zurück. Dieses Verzeichnis enthält das Hauptprogramm und andere notwendige Programme für den Betrieb von PC-DMIS.

PCDMISUSERHIDDENDATAPATH

Anzeige Kompletter Pfad: *PCDMISUSERHIDDENDATAPATH()*

Diese Funktion gibt eine Zeichenfolge mit dem kompletten Pfad des Verzeichnisses mit den verborgenen Benutzerdaten von PC-DMIS zurück. Siehe Abschnitt "Dateispeicherorte verstehen" für Dateien, die dieses Verzeichnis enthält.

PCDMISUSERVISIBLEDATAPATH

Anzeige Kompletter Pfad: *PCDMISUSERHIDDENDATAPATH()*

Diese Funktion gibt eine Zeichenfolge mit dem kompletten Pfad des Verzeichnisses mit den sichtbaren Benutzerdaten von PC-DMIS zurück. Siehe Abschnitt "Dateispeicherorte verstehen" für Dateien, die dieses Verzeichnis enthält.

PCDMISSYSTEMHIDDENDATAPATH

Anzeige Kompletter Pfad: *PCDMISSYSTEMHIDDENDATAPATH()*

Diese Funktion gibt eine Zeichenfolge mit dem kompletten Pfad des Verzeichnisses mit den verborgenen Systemdaten von PC-DMIS zurück. Siehe Abschnitt "Dateispeicherorte verstehen" für Dateien, die dieses Verzeichnis enthält.

PCDMISSYSTEMVISIBLEDATAPATH

Anzeige Kompletter Pfad: PCDMISSYSTEMVISIBLEDATAPATH()

Diese Funktion gibt eine Zeichenfolge mit dem kompletten Pfad des Verzeichnisses mit den sichtbaren Systemdaten von PC-DMIS zurück. Siehe Abschnitt "Dateispeicherorte verstehen" für Dateien, die dieses Verzeichnis enthält.

PCDMISSYSTEMREPORTINGPATH

Anzeige Kompletter Pfad: PCDMISSYSTEMREPORTINGPATH()

Diese Funktion gibt eine Zeichenfolge mit dem kompletten Pfad des Protokollverzeichnisses von PC-DMIS zurück. Dieses Verzeichnis beinhaltet die Protokoll- und Bezeichnungsvorlagen des Protokollfensters.

RIGHT

Rechte n Zeichen der Zeichenfolge: *RIGHT*(<Zeichenfolge>, <Ganzzahl>)

Diese Funktion gibt eine Zeichenfolge zurück und extrahiert dabei die durch die Ganzzahl definierte Anzahl von Zeichen, von rechts nach links aus der Zeichenfolge.

SYSTEMDATE

Systemdatum: SYSTEMDATE(<Datenformat-Zeichenfolge>)

Diese Funktion gibt die datumsformatierte Zeichenfolge mit den aktuellen Datumsdetails zurück. Der Befehl SYSTEMDATE("MM/'dd'/yy") gibt die Zeichenfolge "02/12/14" zurück, wenn das aktuelle Datum der 12. Februar 2014 ist.

Verwenden Sie die folgenden Zeichenfolgenelemente, um die Datumszeichenfolge zu erstellen. Die unten definierte Groß-/Kleinschreibung der Elemente muss eingehalten werden (MM anstelle von mm). Nicht zum Datum gehörende Zeichen (z.B. Leerstellen zwischen Datumsformat-Zeichenfolge-Elementen) erscheinen in der Ausgabezeichenfolge an derselben Position wie in der Eingabezeichenfolge. Zeichen in der Eingabezeichenfolge, die von einfachen

Anführungszeichen umschlossen sind, erscheinen in der Ausgabezeichenfolge an derselben Position, jedoch ohne die Anführungszeichen.

d – Tag des Monats als Ziffer. Ohne führende Null bei einstelligen Werten.

dd – Tag des Monats in Ziffern. Die führende Null wird für die Einzelziffer-Darstellung des Datums verwendet.

ddd – Abkürzung mit drei Buchstaben für den Wochentag.

dddd – Vollständiger Name für den aktuellen Wochentag.

M – Monat als Ziffern ohne führende Nullen für die Einzelziffer-Darstellung von Monaten.

M – Monat als Ziffern ohne führende Null für die Einzelziffer-Darstellung von Monaten.

MMM – Monat als Abkürzung mit drei Buchstaben.

MMMM – Vollständiger Name des Monats.

y – Jahr als Ziffern ohne führende Nullen für die Einzelziffer-Darstellung der Jahre.

yy – Jahr als Ziffern mit führenden Nullen für die Einzelziffer-Darstellung der Jahre.

jjjj – Das mit vier Ziffern dargestellte Jahr.

SYSTEMTIME

Formatierte Systemzeit: *SYSTEMTIME(<Zeitformat-Zeichenfolge>)*

Diese Funktion gibt die zeitformatierte Zeichenfolge mit den aktuellen Zeitdetails zurück. Beispiel: *SYSTEMTIME ("hh:mm:ss tt")* gibt die Zeit in einer formatierten Zeichenfolge zurück, z. B. "11:29:40 PM".

Verwenden Sie die folgenden Zeichenfolgenelemente, um die Zeitzeichenfolge zu erstellen. Die unten angegebene Groß-/Kleinschreibung der Elemente muss eingehalten werden (**tt** anstelle von **TT**). Nicht zur Zeit gehörende Zeichen (z.B. Leerstellen zwischen Zeitformat-Zeichenfolge-Elementen) erscheinen in der Ausgabezeichenfolge an derselben Position wie in der Eingabezeichenfolge. Zeichen in der Eingabezeichenfolge, die von einfachen Anführungszeichen umschlossen sind, erscheinen in der Ausgabezeichenfolge an derselben Position, jedoch ohne die Anführungszeichen.

h – Stunden ohne führende Null bei einstelligen Werten; 12-Stunden-Format.

hh - Stunden mit führender Null für Einzelziffern-Stunden; 12-Stunden-Uhr

H - Stunden mit keiner führenden Null für Einzelziffern-Stunden; 24-Stunden-Uhr

HH - Stunden mit führender Null für Einzelziffern-Stunden; 24-Stunden-Format

m - Minuten ohne führender Null für Einzelziffer-Darstellung der Minuten

mm - Minuten ohne führender Null für Einzelziffer-Darstellung der Minuten

s - Sekunden ohne führender Null für Einzelziffer-Darstellung der Sekunden

ss - Sekunden mit führender Null für Einzelziffer-Darstellung der Sekunden

t - Zeichenfolge mit einem Zeichen zur Zeitmarkierung, wie beispielsweise A oder P

tt - Zeichenfolge mit mehreren Zeichen zur Zeitmarkierung, wie beispielsweise AM oder PM

SYSTIME

Systemzeit: *SYSTIME()*

Diese Funktion gibt eine Zeichenfolge mit der aktuellen Systemzeit zurück. Diese Funktion unterscheidet sich von der oben beschriebenen SYSTEMTIME-Funktion. Sie gibt Tag, Datum, Uhrzeit und Jahr automatisch aus.

Beispiel: "Mit 12. Februar 13:50:21 2014"



Die zurückgegebene Zeichenfolge für die aktuelle Systemzeit ist an die lokale Zeitzoneneinstellung angepasst.

UPPERCASE

Erstellt Zeichenfolge in Großschreibung: *UPPERCASE(<Zeichenfolge>)*

Diese Funktion gibt eine Zeichenfolge zurück, die das Äquivalent der Ausdruckszeichenkette in Großbuchstaben ist.

Mathematische Funktionen

ABS

Absolutwert: *ABS(<Double>)*

Gibt den Absolutwert des Eingabewertes zurück.

EXP

Exponentialwert: *EXP(<Double>)*

Gibt den Exponentialwert des Ausdrucks zurück.

LOG

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

Gibt den Zehnerlogarithmus des Ausdrucks zurück.

LN

Natürlicher Logarithmus: *LN(<Double>)*

Gibt den natürlichen Logarithmus des Ausdrucks zurück.

ROUND

Runden: *ROUND(<Double>)*

Gibt den auf die nächste Ganzzahl gerundeten Wert zurück.

SQRT

Quadratwurzel: *SQRT(<Double>)*

Gibt die Quadratwurzel der Eingabe zurück.

Trigonometrische Funktionen

Jede der Trigonometriefunktionen unterstützt standardmäßig Eingaben und Ausgaben in Radianten. Verwenden Sie die nachfolgend beschriebene Funktion RAD2DEG, wenn Sie Werte in Grad angeben möchten.

ACOS

Arkuskosinus: *ACOS(<Double>)*

Gibt den Arkuskosinus des Ausdrucks zurück. Beispiel: *ACOS(5.0)* gibt beispielsweise 0 zurück. *ACOS(<Ausdruck>)* gibt grundsätzlich den Arkuskosinus des Ausdruckswerts zurück.

ASIN

Arkussinus: *ASIN(<Double>)*

Gibt den Arkussinus für den Eingabewert zurück.

ATAN

Arkustangens: *ATAN(<Double>)*

Gibt den Arkustangens für den Eingabewert zurück.

COS

Kosinus: *COS(<Double>)*

Gibt den Kosinus für den Eingabewert zurück.

DEG2RAD

Grad in Radianen: *DEG2RAD(<Double>)*

Gibt die Eingabe geteilt durch 360 und multipliziert mit 2π zurück. Damit werden Grad in Radianen umgerechnet.

RAD2DEG

Radianen in Grad: *RAD2DEG(<Double>)*

Gibt die Eingabe dividiert durch 360 und multipliziert mit 2π zurück. Damit werden Radianen in Grad umgerechnet.

SIN

Sinus: *SIN(<Double>)*

Gibt den Sinus für den Eingabewert zurück.

TAN

Tangente: *TAN(<Double>)*

Gibt den Tangens für den Eingabewert zurück.



Funktionen, bei denen die Eingabe außerhalb der Toleranz liegt (bei ACOS, ASIN, LOG, LN, SQRT usw. würde der Computer abstürzen), geben 0 zurück.

Punktfunktionen**ANGLEBETWEEN**

Zwischenwinkel: *ANGLEBETWEEN(<Vektor>, <Vektor>)*

Gibt den Wert des Winkels in Grad zwischen den beiden Vektoren zurück. Die beiden Parameter müssen Ausdrücke sein, die auf einen Vektortyp ausgewertet werden. Um den Vektor von einem Element zu erhalten, müssen Sie beispielsweise die Element-ID, gefolgt von der Erweiterung .IJK, verwenden. Sie können dies im untenstehenden Codeausschnitt sehen:



```
F1      =GENERIC/POINT,DEPENDENT,CARTESIAN,$
        NOM/XYZ,<3,3,3>,$
        MEAS/XYZ,<3,3,3>,$
        NOM/IJK,<1,0,0>,$
        MEAS/IJK,<1,0,0>

F2      =GENERIC/POINT,DEPENDENT,CARTESIAN,$
        NOM/XYZ,<10,10,10>,$
        MEAS/XYZ,<10,10,10>,$
        NOM/IJK,<0,0,1>,$
        MEAS/IJK,<0,0,1>
        ASSIGN/V1=F1.IJK
        ASSIGN/V2=F2.IJK
        ASSIGN/V3=ANGLEBETWEEN(V1,V2)
        COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,
        "The angle between "+V1+" and "+V2+" is: "+V3
```

CROSS

Kreuzprodukt: *CROSS(<Punkt>, <Punkt>)*

Der Rückgabewert ist vom Typ Punkt und ist der Einheitsvektor in Richtung des Kreuzprodukts des ersten und zweiten Ausdrucks.

DELTA

Vektorversatz: *DELTA*(<Punkt>, <Punkt>, <Double>)

Die Funktion nimmt den ersten Ausdruck (Punkt) und berechnet einen neuen Punkt in der Richtung des zweiten Ausdrucks (Vektor) mit einem Versatz gemäß dem dritten Ausdruck. *DELTA* (*MPOINT* (0,0,0) , *MPOINT* (1,0,0) , 10) gibt beispielsweise den Punkt 10,0,0 zurück.

DOT

Skalarprodukt: *DOT*(<Punkt>, <Punkt>)

Gibt das Skalarprodukt der beiden Punkte (Vektoren) zurück.

UNIT

Einheitsvektor: *UNIT*(<Punkt>)

Gibt den Einheitsvektor dividiert durch seine Länge zurück. *UNIT* (*MPOINT* (0,0,0)) gibt beispielsweise den Punkt 0,0,1 zurück.

MPOINT

Erzwungene Umwandlung in einen Punkt: *MPOINT*(<Ausdruck1>, <Ausdruck2>, <Ausdruck3>)

Wandelt die drei Ausdrücke in den Typ "Punkt" um.

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

Wobei:

$$V1.X = 2.5$$

$$V1.Y = 3.6$$

$$V1.Z = 4.0$$

Siehe "Erzwungene Umwandlung in einen Punkt".

Zeigerfunktionen

DIST2D

2D-ABSTAND: *DIST2D*(<ELEM1>, <ELEM2>, <ELEM3>)



Die Elemente müssen innerhalb geschwungener Klammern stehen.

Berechnet den Abstand zwischen den ersten beiden Argumenten im Befehl (ELEM1 und ELEM2), im rechten Winkel zum dritten Argument (ELEM3).

- Handelt es sich beim dritten Argument um eine Ebene, dann berechnet PC-DMIS den Abstand zwischen den ersten beiden Argumenten im rechten Winkel zu dieser Ebene.
- Handelt es sich beim dritten Argument um eine Gerade oder um einen Zylinder, dann berechnet PC-DMIS den Abstand zwischen den ersten beiden Argumenten im rechten Winkel zum dritten Argument in der aktiven Arbeitsebene.

Angenommen, Sie haben die XY-Ebene als drittes Argument, dann weist diese einen 'Z+'-Vektor (0,0,1) auf und der protokollierte Abstand wird entlang der Z-Achse sein.

Beispiel



```
ASSIGN/V3=DIST2D({CIR1},{CIR2},{PLN1})
COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,
V3
```

DIST3D

3D-ABSTAND: DIST3D(<ELEM1>, <ELEM2>)

Berechnet den 3D-Abstand zwischen ELEM1 und ELEM2.

Die Elemente müssen innerhalb geschwungener Klammern stehen.

Beispiel



```
ASSIGN/V3=DIST3D({CIR1},{CIR2})
COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,
V3
```

GETCOMMAND

Bezieht einen Zeiger zu dem durch die Parameter angegebenen Befehl:

```
GETCOMMAND(<Ganzzahl oder Zeichenfolge>, <Zeichenfolge>,
<Ganzzahl>
```

Erster Parameter - Befehlsangabefeld

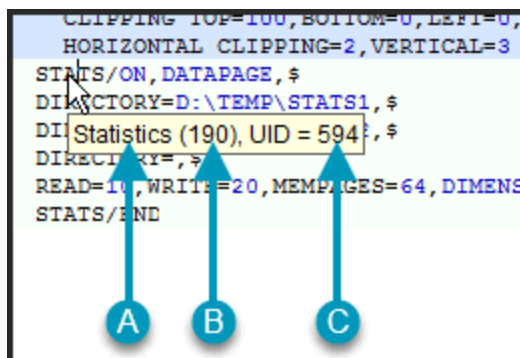
Der erste Parameter ist das Befehlsangabefeld. Es gibt den zu suchenden Befehlstyp an. Es kann Folgendes übergeben werden:

- Eine den Befehl beschreibende Zeichenfolge. Siehe (A) in der unten stehenden Grafik.
- Eine Nummer des Befehlstyps. Siehe (B) in der unten stehenden Grafik.
- Die eindeutige Nummernkennung. Siehe (C) in der unten stehenden Grafik.

Wird die eindeutige ID des Befehls übergeben, sind keine weiteren Argumente erforderlich.

So erhalten Sie die den Befehl beschreibende Zeichenfolge, die Nummer des Befehlstyps und die eindeutige Nummernkennung des Befehls:

1. Klicken Sie mit der rechten Maustaste auf eine Stelle im Bearbeitungsfenster.
2. Wählen Sie **Wechsle Popup-Anzeige | Befehlsangaben** (PC-DMIS muss sich im Befehlsmodus befinden).
3. Setzen Sie den Mauszeiger auf den gewünschten Befehl. PC-DMIS zeigt im Popup die Befehlsbeschreibung, Typnummer und eindeutige Nummernkennung des betreffenden Befehls.



- A. Eine den Befehl beschreibende Zeichenfolge
- B. Eine Nummer des Befehlstyps
- C. Die eindeutige Nummernkennung (UID)

Zweiter Parameter - Suchrichtung

Der zweite Parameter gibt die Suchrichtung an. Zulässige Werte sind:

Wert	Beschreibung
NACH OBEN	Dieser Wert bedeutet, dass die Suche bei dem aktuellen Befehl beginnen und nach oben weiterverlaufen soll.
NACH UNTEN	Dieser Wert besagt, dass die Suche bei dem aktuellen Befehl beginnen und nach unten weiterverlaufen soll.
OBEN	Dieser Wert besagt, dass die Suche am Anfang der Messroutine beginnen und nach unten weiterverlaufen soll.
UNTEN	Dieser Wert besagt, dass die Suche bei dem letzten Objekt in der Messroutine beginnen und nach oben weiterverlaufen soll.

Dritter Parameter – Zu suchende Instanz

Der dritte Parameter gibt an, welche Instanz des Befehls gefunden werden sollte, wenn mehrere Instanzen desselben Befehls in der Messroutine vorhanden sind.



Angenommen, die Messroutine besitzt zwei Instanzen des Befehls STATS/ON und Sie möchten einen Zeiger zur zweiten Instanz von oben erhalten. In diesem Fall würde wie nachstehend veranschaulicht als dritter Parameter "2" übergeben werden und "TOP" als zweiter Parameter übergeben werden.

```
ZUWEISEN/V1=GETCOMMAND("Statistiken","TOP",2)
```

Sie können die Funktion GETCOMMAND verwenden, um den dritten Parameter für die Zeichenfolgenfunktion GETTEXT bereitzustellen. Weitere Informationen zu GETTEXT finden Sie unter "Zeichenfolgenfunktionen"

LEN

Zeigerschleifenzähler: *LEN(<ZEIGER>)*

Gibt aus, wie oft ein Zeiger in einer Schleife war. Ein Beispiel: Wenn sich das Element KREIS1 in einer Schleife befindet, die 10 Mal wiederholt wird, können Sie in einer Variable speichern, wie oft KREIS1 gemessen wurde. Hierfür verwenden Sie eine ZUWEISEN-Anweisung, die wie folgt aussieht: `ZUWEISEN/V1=LEN({KREIS1 })`

Array-Funktionen

ARRAY

Array erstellen: *ARRAY(<AUSDRUCK1>, <AUSDRUCK2>, <AUSDRUCK3>, ...)*

Erstellt ein Array-Objekt mit Array-Elementen, die durch Ausdrucksparameter angegeben werden. Die Array-Elemente sind mit dem Basisindex 1 nummeriert.

AVERAGE

Durchschnitt Array-Elemente: *DURCHSCHNITT(<ARRAY>)*

Gibt den Durchschnittswert der Elemente im Array zurück.

EQUAL

Array-Vergleich Element für Element: *VERGLEICH(<ARRAY>, <ARRAY>)*

Vergleicht die beiden Arrays Element um Element, um zu bestimmen, ob sie dieselben Elemente enthalten. Wenn die beiden Arrays nicht gleich groß sind oder mindestens ein Element in einem Array nicht mit dem entsprechenden Element des anderen Arrays übereinstimmt, gibt die Funktion 0 zurück. Andernfalls liefert sie den Wert 1.

LEN

Array-Elementzähler: *LEN(<ARRAY>)*

Gibt die Anzahl der Elemente im Array zurück.

MAX

Größtes Array-Element: *MAX(<ARRAY>)*

Gibt das größte Element des Arrays zurück. Elemente im Array werden numerisch oder alphabetisch verglichen.

MIN

Kleinstes Array-Element: *MIN(<ARRAY>)*

Gibt das kleinste Element des Arrays zurück. Elemente im Array werden numerisch oder alphabetisch verglichen.

SUM

Summe der Array-Elemente: *SUM(<ARRAY>)*

Gibt die Summe der Elemente im Array zurück.

Gemischte Funktionen

ARCSEGMENTENDINDEX

Diese Funktion gibt den Indexwert des Endpunktes eines vorgegebenen Bogensegments aus einem Scan zurück: *ARCSEGMENTENDINDEX(<ID>, <index>, <tol1>, <tol2>)*

<ID> – Der erste Parameter ist ein Zeichenfolgenwert der ID des Scans, aus dem diese Funktion den Indexwert des Endpunktes für den Bogen extrahiert. Diese Parameter können entweder die ID in Anführungszeichen, oder ein beliebiger Ausdruck sein, der bei Umwandlung in den Typ "Zeichenfolge" als die ID eines Scans interpretiert wird.

<index> – Der zweite Parameter ist der Indexwert für den Bogen, aus dem Sie den Endpunkt erhalten. Hierbei handelt es sich um einen mit "1" beginnenden Indexwert. Wenn Sie beispielsweise den Endpunkt für den dritten Bogen des Scans erhalten möchten, dann wäre der Indexwert "3".

<tol1> – Der dritte Parameter ist der allgemeine Elementtoleranzwert. Dies ist der maximale Formfehler, der dazu verwendet wird, den Scan in Linien und Bögen zu unterteilen.

<tol2> – Der vierte Parameter ist der verfeinernde Toleranzwert. Im Allgemeinen wird dieser engere Toleranzwert dazu verwendet, Punkte von jedem Ende des Elements so lange auszulassen, bis sich der Formfehler des Segments innerhalb dieser Toleranz befindet.

Sobald Sie die Anfangs- und Endindexwerte für einen Bogen erhalten, können Sie diese Punkte innerhalb eines abhängigen Elements dazu verwenden, ein separates Bogenelement zu erstellen. Ein ähnliches Beispiel finden Sie unter "Beispiel eines Geradenelements, das aus einem Scansegment erstellt wurde".

ARCSEGMENTSTARTINDEX

Diese Funktion gibt den Indexwert des Anfangspunktes eines angegebenen Bogensegments aus einem Scan zurück: *ARCSEGMENTSTARTINDEX(<ID>, <index>, <tol1>, <tol2>)*.

<ID> – Der erste Parameter ist ein Zeichenfolgenwert der ID des Scans, aus dem diese Funktion den Indexwert des Anfangspunktes für den Bogen extrahiert. Diese Parameter können entweder die ID in Anführungszeichen, oder ein beliebiger Ausdruck sein, der bei Umwandlung in den Typ "Zeichenfolge" als die ID eines Scans interpretiert wird.

<index> – Der zweite Parameter ist der Indexwert für den Bogen, aus dem Sie den Anfangspunkt erhalten. Hierbei handelt es sich um einen mit "1" beginnenden

Indexwert. Wenn Sie beispielsweise den Anfangspunktwert für den dritten Bogen des Scans erhalten möchten, dann wäre der Indexwert "3".

`<tol1>` – Der dritte Parameter ist der allgemeine Elementtoleranzwert. Dies ist der maximale Formfehler, der dazu verwendet wird, den Scan in Linien und Bögen zu unterteilen.

`<tol2>` – Der vierte Parameter ist der verfeinernde Toleranzwert. Im Allgemeinen wird dieser engere Toleranzwert dazu verwendet, Punkte von jedem Ende des Elements so lange auszulassen, bis sich der Formfehler des Segments innerhalb dieser Toleranz befindet.

Es gibt zwei zusätzliche Parameter, die prüfen, ob ein identifiziertes Bogensegment in einem Scan akzeptabel ist. Diese Parameter können nur im PC-DMIS-Einstellungseditor geändert werden. Jedes Bogensegment mit einem Radius kleiner als der Wert des Eintrages `MinimumArcSegmentRadiusInMM` wird abgelehnt. Der Standardwert für diesen Parameter lautet 2mm. Ebenso wird jedes Bogensegment mit einem Radius größer als der Wert des Eintrages `MaximumArcSegmentRadiusInMM` abgelehnt. Der Standardwert für diesen Parameter lautet 2000 mm (eine Änderung dieses Werts sollte überflüssig sein).

Sobald Sie die Anfangs- und Endindexwerte für einen Bogen erhalten, können Sie diese Punkte innerhalb eines abhängigen Elements dazu verwenden, ein separates Bogenelement zu erstellen. Ein ähnliches Beispiel finden Sie unter "Beispiel eines Geradenelements, das aus einem Scansegment erstellt wurde".

EOF und EOL

Weitere Informationen zu diesen Funktionen finden Sie unter "Das Ende einer Datei oder Zeile prüfen" im Abschnitt "Verwenden der Datei-Eingabe/-Ausgabe".

FUNCTION

Erzeugt eine Funktion: `FUNCTION((<PARAM1>, <PARAM2>...), <AUSDRUCK>)`

Erstellt eine Funktion, die die in der Parameterliste angegebenen Parameter in den Ausdruck einsetzt.

- Das erste Element bei der Verwendung des Schlüsselworts `FUNCTION` ist die Parameterliste.
- Die Liste besteht aus kommagetrennten Parameternamen.
- Die Parameterliste steht auch in Klammern.
- Das zweite Element ist der Ausdruck.
- Der Ausdruck enthält die Parameternamen dort, wo die Parameter beim Aufrufen der Funktion eingesetzt werden sollen.

Ein Beispiel finden Sie im Abschnitt "Beispiel für allgemeine Funktionen".

GETROTABDATA

Diese Funktion gibt den Mittelpunkt, die Winkelposition und die Vektorwerte für den bestimmten Drehtisch in der folgenden Konfiguration zurück.

GETROTABDATA(<PARAMETER>[, <TISCH>])

Die Funktion gibt die Werte für die folgenden Konfigurationen zurück:

- Einzelner Drehtisch
- Doppelte Drehtische (unabhängig)
- Gestapelter Drehtisch

Die durch diese Funktion gelieferten Daten entsprechen den Daten im Dialogfeld **Drehtisch einrichten (Bearbeiten | Einstellungen | Drehtisch einrichten)**. Weitere Informationen über dieses Dialogfeld finden Sie unter "Definieren des Drehtisches".

ZENTRIERT

- "CENTER" - Gibt den aktuellen XYZ-Mittelpunkt des Tisches zurück.
- "CENTER", "V" - Gibt den aktuellen XYZ-Mittelpunkt für den Drehtisch V in einer Konfiguration mit doppelten oder gestapelten Drehtischen zurück.
- "CENTER", "W" - Gibt den aktuellen XYZ-Mittelpunkt für den Drehtisch W in einer Konfiguration mit doppelten oder gestapelten Drehtischen zurück.

Beispiele:

<code>ZUWEISEN/V1=GETROTABDATA ("CENTER")</code>	V1 wird auf den aktuellen XYZ-Mittelpunkt des Drehtisches gesetzt.
<code>ZUWEISEN/V1=GETROTABDATA ("CENTER", "V")</code>	V1 wird auf den aktuellen XYZ-Mittelpunkt des Drehtisches V gesetzt.
<code>ZUWEISEN/V1=GETROTABDATA ("CENTER", "W")</code>	V1 wird auf den aktuellen XYZ-Mittelpunkt des Drehtisches W gesetzt.

WINKELPOSITION

- "ANGLE" - Gibt die aktuelle Winkelposition des Drehtisches zurück.

Ausdruckskomponenten

- "ANGLE","V" - Gibt die aktuelle Winkelposition für den Drehtisch V in einer Konfiguration mit doppelten oder gestapelten Drehtischen zurück.
- "ANGLE","W" - Gibt die aktuelle Winkelposition für den Drehtisch W in einer Konfiguration mit doppelten oder gestapelten Drehtischen zurück.

Beispiele:

<code>ZUWEISEN/V2=GETROTABDATA ("ANGLE")</code>	V2 wird auf die aktuelle Winkelposition des Drehtisches gesetzt.
<code>ZUWEISEN/V2=GETROTABDATA ("ANGLE", "V")</code>	V2 wird auf die aktuelle Winkelposition für den Drehtisch V gesetzt.
<code>ZUWEISEN/V2=GETROTABDATA ("ANGLE", "W")</code>	V2 wird auf die aktuelle Winkelposition für den Drehtisch W gesetzt.

VEKTOR

- "VECTOR" - Gibt den aktuellen IJK-Vektor des Drehtisches zurück.
- "VECTOR","V" - Gibt den aktuellen IJK-Vektor für den Drehtisch V in einer Konfiguration mit doppelten oder gestapelten Drehtischen zurück.
- "VECTOR","W" - Gibt den aktuellen IJK-Vektor für den Drehtisch W in einer Konfiguration mit doppelten oder gestapelten Drehtischen zurück.

Beispiele:

<code>ZUWEISEN/V3=GETROTABDATA ("VECTOR")</code>	V3 wird auf den aktuellen IJK-Vektor des Drehtisches gesetzt.
<code>ZUWEISEN/V3=GETROTABDATA ("VECTOR", "V")</code>	V3 wird auf den aktuellen IJK-Vektor für den Drehtisch V gesetzt.
<code>ZUWEISEN/V3=GETROTABDATA ("VECTOR", "W")</code>	V3 wird auf den aktuellen IJK-Vektor für den Drehtisch W gesetzt.



Das Argument [TISCH] ist optional. Wenn Sie weder Drehtisch V oder W bestimmen, macht PC-DMIS Folgendes:

- Wenn Sie einen einzelnen oder gestapelten Drehtisch verwenden, werden die Werte für den Drehtisch W zurückgegeben.
- Wenn Sie einen doppelten Drehtisch verwenden, werden die Werte für den auf der Symbolleiste **Aktueller Drehtisch** aktivierten Drehtisch zurückgegeben. Weitere Informationen finden Sie unter "Symbolleiste "Aktueller Drehtisch"".

PC-DMIS besitzt zwei interne Tischdefinitionen, um doppelte und gestapelte Tischkonfigurationen zu berücksichtigen. Für einen Einzeltisch wird die zweite Tischdefinition nicht verwendet. Da es sich um eine interne Definition handelt, wird kein Fehler auftreten, wenn Sie für einen Einzeltisch den Tisch V bestimmen, jedoch wird dies nicht empfohlen. Die Werte, die diese Funktion zurückgibt, sind allgemein nützlich, da der Tisch nicht wirklich existiert.

IF

Bedingte Ausdrucksauswertung: IF(<AUSDRUCK1>, <AUSDRUCK2>, <AUSDRUCK3>)

Wenn AUSDRUCK1 wahr (Ergebniswert nicht 0) ist, gibt diese Funktion den Wert von AUSDRUCK2 zurück; im anderen Fall liefert sie den Wert von AUSDRUCK3.

ISIOCHANNELSET

Dieser Ausdruck verarbeitet zwei Parameter. Der erste Parameter gibt an, welcher I/O-Kanal überprüft wird (der Bereich der verfügbaren Nummern richtet sich nach der jeweils verwendeten Maschine). Der zweite Parameter bestimmt, ob die Software die Arm1- oder die Arm2-Maschine abfragt. Ist für den zweiten Parameter 1 (Eins) festgelegt, wird die Arm2-Steuereinheit abgefragt. Ist kein zweiter Parameter vorhanden (oder lautet dessen Einstellung Null), dann fragt der I/O-Kanal die Arm1-Steuereinheit ab. Außerhalb des Mehrarmbetriebs ist die Arm1-Steuereinheit die einzig verfügbare Option.



Wenn einer der angegebenen Parameter (Tasterdatentyp, Tastspitzen-ID, Tasterdateiname oder Kanalnummer) ungültig ist, gibt der Ausdruck den Wert 0 zurück.

Beispiel:

<p>ZUWEISEN/V4 = ISIOCHANNELSET (3, 0)</p>	<p>V4 entspricht 1 (dem Wert "wahr"), wenn der Kanal festgelegt ist, andernfalls ergibt V4 den Wert 0 ("falsch").</p>
--	---

LINESEGMENTENDINDEX

Diese Funktion gibt den Indexwert des Endpunktes eines vorgegebenen Liniensegments aus einem Scan zurück: LINESEGMENTENDINDEX(<ID>, <index>, <tol1>, <tol2>).

<ID> – Der erste Parameter ist ein Zeichenfolgenwert der ID des Scans, aus dem diese Funktion den Indexwert des Endpunktes für das Liniensegment extrahiert. Diese Parameter können entweder die ID in Anführungszeichen, oder ein beliebiger Ausdruck sein, der bei Umwandlung in den Typ "Zeichenfolge" als die ID eines Scans interpretiert wird.

<index> – Der zweite Parameter ist der Indexwert für das Liniensegment, aus dem Sie den Endpunkt erhalten. Hierbei handelt es sich um einen mit "1" beginnenden Indexwert. Wenn Sie beispielsweise den Endpunkt für die dritte Linie des Scans erhalten möchten, dann wäre der Indexwert des Liniensegments "3".

<tol1> – Der dritte Parameter ist der allgemeine Elementtoleranzwert. Dies ist der maximale Formfehler, der dazu verwendet wird, den Scan in Linien und Bögen zu unterteilen.

<tol2> – Der vierte Parameter ist der verfeinernde Toleranzwert. Im Allgemeinen wird dieser engere Toleranzwert dazu verwendet, Punkte von jedem Ende des Elements so lange auszulassen, bis sich der Formfehler des Segments innerhalb dieser Toleranz befindet.

Sobald Sie die Anfangs- und Endindexwerte für ein Liniensegment erhalten, können Sie diese Punkte innerhalb eines abhängigen Elements dazu verwenden, ein separates Linienelement zu erstellen. Ein Beispiel finden Sie unter "Beispiel eines Geradenelements, das aus einem Scansegment erstellt wurde".

LINESEGMENTSTARTINDEX

Gibt den Indexwert des Anfangspunktes eines angegebenen Liniensegments aus einem Scan zurück: LINESEGMENTSTARTINDEX(<ID>, <index>, <tol1>, <tol2>).

<ID> – Der erste Parameter ist ein Zeichenfolgenwert der ID des Scans, aus dem diese Funktion den Indexwert des Anfangspunktes für das Liniensegment extrahiert. Hierbei kann es sich entweder um die ID in Anführungszeichen, oder um einen beliebigen Ausdruck handeln, der bei Umwandlung in den Typ "Zeichenfolge" als die ID eines Scans interpretiert wird.

<index> – Der zweite Parameter ist der Indexwert für das Liniensegment, aus dem Sie den Anfangspunkt erhalten. Hierbei handelt es sich um einen mit "1" beginnenden Indexwert. Wenn Sie beispielsweise den Anfangspunkt für die dritte Linie des Scans erhalten möchten, dann wäre der Indexwert des Liniensegments "3".

<tol1> – Der dritte Parameter ist der allgemeine Elementtoleranzwert. Dies ist der maximale Formfehler, der dazu verwendet wird, den Scan in Linien und Bögen zu unterteilen.

<tol2> – Der vierte Parameter ist der verfeinernde Toleranzwert. Im Allgemeinen wird dieser engere Toleranzwert dazu verwendet, Punkte von jedem Ende des Elements so lange auszulassen, bis sich der Formfehler des Segments innerhalb dieser Toleranz befindet.

Es gibt zwei zusätzliche Parameter, die prüfen, ob ein identifiziertes Liniensegment in einem Scan akzeptabel ist. Dieser Parameter kann nur im PC-DMIS-Einstellungseditor geändert werden. Jedes Liniensegment, das kürzer als der Wert des Eintrages `MinimumLineSegmentLengthInMM` ist, wird abgelehnt. Der Standardwert für diesen Parameter lautet 2 mm.

Sobald Sie die Anfangs- und Endindexwerte für ein Liniensegment erhalten, können Sie diese Punkte innerhalb eines abhängigen Elements dazu verwenden, ein separates Linienelement zu erstellen. Ein Beispiel hierzu finden Sie unter "Beispiel eines Geradenelements, das aus einem Scansegment erstellt wurde".

PROBEDATA

Diese Funktion gibt Daten zum aktuellen oder angegebenen Taster zurück:

`PROBEDATA(<OPTTASTERDATENTYP>, <OPTTASTSPITZENID>, <OPTTASTERDATEINAME>)`

Diese Funktion verwendet drei optionale Parameter. Sie müssen die Parameter lediglich durch Kommata trennen, wenn Sie mehr als einen Parameter verwenden. Zwischen leeren Parametern müssen keine Kommata gesetzt werden. Um beispielsweise den aktuellen Tasterdurchmesser zu erhalten, verwenden Sie einfach `ASSIGN/V1=PROBEDATA("DIAM")`.

OPTPROBEDATATYPE - Optionaler Parameter, der angibt, welche Tasterdaten der Ausdruck zurückgeben soll. Wird dieser Parameter nicht angegeben, wird die aktuelle Tastspitzen-ID zurückgegeben (Standardparameter). Dieser Parameter gehört dem Typ "Zeichenfolge" an. Jeder Ausdruck, der einen gültigen Zeichenfolgenausdruck ergibt, kann an der Position des ersten Ausdrucks stehen. Zu den gültigen Zeichenfolgen ausdrücken (Groß-/Kleinschreibung ist unerheblich) für den ersten Ausdruck zählen. Hierbei handelt es sich um Zeichenfolgenausdrücke, die in doppelten Anführungszeichen stehen sollten:

"A" - Tastspitzen-A-Winkel. Gibt den Typ "Gleitkommazahl" zurück.

"B" - B-Tastspitzenwinkel. Gibt den Typ "Gleitkommazahl" zurück.

"C" - Der Winkel C eines CW43 light probe head (Tastkopf). Gibt den Typ "Ganzzahl" zurück.

"Date" - Datum, an dem die Tastspitze zuletzt kalibriert wurde. Gibt den Typ "Zeichenfolge" zurück.

"Diam" oder "Diameter" - Gemessener Tastspitzen-Durchmesser. Die ersten vier Zeichen "Diam" sind erforderlich. Die restlichen Zeichen bis zur vollständigen Bezeichnung "Diameter" sind fakultativ. Gibt den Typ "Gleitkommazahl" zurück.

"ID" - Tastspitzen-ID. Standardparameter. Gibt den Typ "Zeichenfolge" zurück.

"Offset" - Gemessener Tastspitzen-X-, -Y-, -Z-Versatz. Gibt den Typ "Punkt" zurück.

"PrbRdv" – Die radiale Abweichung des Tasters. Gibt den Typ "Gleitkommazahl" zurück.

"Rotation" - Dies ist die Drehung um den Spitzenvektor im Bogenmaß. Gibt den Typ "Gleitkommazahl" zurück.

"Standarddeviation" - Die Standardabweichung der Taster. Gibt den Typ "Gleitkommazahl" zurück.

"Thick" oder "Thickness" - Gemessene Tastspitzenstärke. Die ersten fünf Zeichen "Thick" sind erforderlich. Die restlichen Zeichen bis zur vollständigen Bezeichnung "Thickness" sind fakultativ. Gibt den Typ "Gleitkommazahl" zurück.

"Time" - Uhrzeit, zu der die Tastspitze zuletzt kalibriert wurde. Gibt den Typ "Zeichenfolge" zurück.

"Vector" - Tastspitzenvektor. Gibt den Typ "Punkt" zurück.



Hinweis: Durch Voranstellen des Buchstabens **"T"** bei den Funktionen **"Diameter"**, **"Offset"** oder **"Thickness"** werden die theoretischen Informationen (z. B. **TDIAMETER**, **TOFFSET** und **TTHICKNESS**) zurückgegeben.

OPTTIPIID - Dieser optionale Parameter gibt an, mit welcher Tastspitze die mit dem ersten Ausdruck spezifizierten Tasterdaten ermittelt werden sollen. Wenn dieser Parameter nicht angegeben wird, wird die aktuelle Tastspitze verwendet. Dieser Parameter sollte vom Typ "Zeichenfolge" sein.

OPTPROBEFILENAME - Dieser optionale Parameter gibt an, mit welcher Tasterdatei die Tasterdaten ermittelt werden sollen. Wenn dieser Parameter nicht angegeben wird, wird die aktuelle Tasterdatei verwendet.

Beispiele:

ZUWEISEN/V1=PROBEDATA ()	V1 wird die aktuelle Tastspitzen-ID zugewiesen (z.B. "T1A0B0").
ZUWEISEN/V2=PROBEDATA ("TOFFSET", "T1A45B0")	V2 wird der theoretische Tasterversatz für die Tastspitze T1A45B0 zugewiesen.
ZUWEISEN/V3=PROBEDATA ("Date", "T1A90B90", "MEINTASTER")	V3 wird eine Zeichenfolge zugewiesen, die das Datum repräsentiert, an dem die Tastspitze T1A90B90 der Tasterdatei MEINTASTER zuletzt kalibriert wurde.

QUALTOOLDATA

Diese Funktion gibt Daten über den aktuellen oder angegebenen Taster zurück. Die Syntax lautet wie folgt:

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

Dieser Befehl verwendet bis zu drei Parameter. Zur Datenrückgabe ist mindestens ein Parameter erforderlich:

Der erste Parameter, *<TOOLINFO>*, ist eine *Zeichenfolge*, mit der die Art der Informationen, die über das Kalibriernormal zurückzugeben sind, festgelegt wird. Wird dieser Parameter nicht übergeben, gibt diese Funktion den Namen des aktuellen oder angegebenen Kalibriernormals wieder.

- **"CTE"** oder **"COEFFICIENTOFTHERMALEXPANSION"** - Einer dieser beiden Zeichenfolgen gibt den Wärmeausdehnungskoeffizienten als Gleitkommazahl zurück.
- **"DIAM"** - Hiermit wird der Durchmesser des Instruments als doppelter Wert zurückgegeben.
- **"ID"** - Hiermit wird der Name des Instruments als Zeichenfolge zurückgegeben.
- **"LENGTH"** - Erfüllt dieselbe Funktion wie "DURCHM". Gibt auch den Durchmesser des Kalibriernormals als einen doppelten Wert zurück.
- **"OVERRIDEIJK"** - Gibt den mit der Suche übersteuerten IJK-Vektor als Punktwert zurück.
- **"POLYDIAM"** - Gibt den Durchmesser der vorgegebenen polyedrischen Fläche als doppelten Wert zurück.
- **"POLYIJK"** - Gibt den IJK-Vektor der vorgegebenen polyedrischen Fläche als Punktwert zurück.
- **"POLYXYZ"** - Gibt den XYZ-Vektor der vorgegebenen polyedrischen Fläche als Punktwert zurück.
- **"SHANKIJK"** - Gibt den IJK-Vektor des Schafts als Punktwert zurück.
- **"TYP"** – Gibt den Kalibriertyp als einen Ganzzahlenwert zurück (0 für eine Kugel, 1 für eine Arm2-Kugel, 2 für ein Polyeder, 3 für ein Arm2-Polyeder).
- **"WIDTH"** - Dieser Parameter wird nicht mehr verwendet.
- **"XYZ"** - Gibt die XYZ-Position des Instruments als Punktwert zurück.

Der zweite Parameter, *<TOOLID>*, ist eine *Zeichenfolge*, mit der der Name des Kalibriernormals, über das der Bediener Informationen erhalten möchte, festgelegt wird. Wird dieser Parameter nicht übergeben, nimmt PC-DMIS an, dass Informationen über das aktuelle Kalibriernormal gewünscht werden. Bei dieser Zeichenfolge wird nicht zwischen Groß- und Kleinbuchstaben unterschieden. Den dritten Parameter, *<FACENUMBER>*, benötigen Sie nur für die Arbeit mit einem polyedrischen Kalibriernormal und nur dann, wenn der erste Parameter "POLYXYZ", "POLYIJK", or "POLYDURCHM" lautet. Hierbei handelt es sich um einen Ganzzahlwert, der vorgibt, welche Fläche des polyedrischen Kalibriernormals zu verwenden ist, um Daten zu erhalten.



Das Kalibriernormal verfügt nicht über eine globale Einstellung, die automatisch für alle Taster in einer Messroutine gilt. Wenn Sie eine Taster zum ersten Mal kalibrieren, müssen Sie das zu verwendende Kalibriernormal auswählen. PC-DMIS speichert diese Informationen zum Kalibriernormal für jeden Taster. Wenn Sie Taster dieselbe Sonde erneut kalibrieren, verwendet PC-DMIS dasselbe Kalibriernormal.

Beispiele:

<code>ZUWEISEN/VDURCHM=QUALTOOLDATA ("DIAM", "KUGEL_1_INNEN")</code>	Übergibt der Variablen VDURCHM den Durchmesser des Kalibriernormals KUGEL_1_INNEN.
<code>ZUWEISEN/VID=QUALTOOLDATA ("ID")</code>	Übergibt der Variablen VID den Namen des aktuellen Kalibriernormals.
<code>ZUWEISEN/VTYP=QUALTOOLDATA ("TYPE")</code>	Übergibt der Variablen VTYP den aktuellen Kalibriernormaltyp.
<code>ZUWEISEN/VPOLYDURCHM=QUALTOOLDATA ("POLYDIAM", "POLYTEST", 3)</code>	Übergibt der Variablen VPOLYDIAM den Durchmesser von Fläche 3 auf dem polyedrischen Kalibriernormal POLYTEST.

SETROTABDATA

Diese Funktion setzt den Mittelpunkt oder den Vektor auf den neuen Eingabewert:

`SETROTABDATA(<PARAMETER>,<NeuerWert>[,<TISCH>])`

Sie funktioniert mit den folgenden Konfigurationen:

- Einzelner Drehtisch
- Doppelte Drehtische (unabhängig)
- Gestapelter Drehtisch

ZENTRIERT

- "CENTER",<NeuerWert> - Setzt den XYZ-Mittelpunkt des Drehtisch auf den neuen Wert.
- "CENTER",<NeuerWert>,"V" - Setzt den XYZ-Mittelpunkt des Drehtisches V für eine doppelte oder gestapelte Tischkonfiguration auf den neuen Wert.
- "CENTER",<NeuerWert>,"W" - Setzt den XYZ-Mittelpunkt des Drehtisches W für eine doppelte oder gestapelte Tischkonfiguration auf den neuen Wert.

Beispiele:

<code>ZUWEISEN/V1=SETROTABDATA ("CENTER",NeuerWert)</code>	V1 ist ein Rückgabecode (1=Erfolg, 0=Fehler).
<code>ZUWEISEN/V1=SETROTABDATA ("CENTER",NeuerWert,"V")</code>	V1 ist ein Rückgabecode (1=Erfolg, 0=Fehler).
<code>ZUWEISEN/V1=SETROTABDATA ("CENTER",NeuerWert,"W")</code>	V1 ist ein Rückgabecode (1=Erfolg, 0=Fehler).

VEKTOR

- "VEKTOR",<NeuerWert> - Setzt den IJK-Vektor des Drehtisch auf den neuen Wert.
- "VECTOR",<NeuerWert>,"V" - Setzt den IJK-Vektor des Drehtisches V für eine doppelte oder gestapelte Tischkonfiguration auf den neuen Wert.
- "VECTOR",<NeuerWert>,"W" - Setzt den IJK-Vektor des Drehtisches W für eine doppelte oder gestapelte Tischkonfiguration auf den neuen Wert.

Beispiele:

<code>ZUWEISEN/V2=SETROTABDATA ("VECTOR",NeuerWert)</code>	V2 ist ein Rückgabecode (1=Erfolg, 0=Fehler).
<code>ZUWEISEN/V2=SETROTABDATA ("VECTOR",NeuerWert,"V")</code>	V2 ist ein Rückgabecode (1=Erfolg, 0=Fehler).
<code>ZUWEISEN/V2=SETROTABDATA ("VECTOR",NeuerWert,"W")</code>	V2 ist ein Rückgabecode

	(1=Erfolg, 0=Fehler).
--	--------------------------



Das Argument [TISCH] ist optional. Wenn Sie weder Drehtisch V oder W bestimmen, macht PC-DMIS Folgendes:

- Wenn Sie einen einzelnen oder gestapelten Drehtisch verwenden, wird der neue Wert für den Drehtisch W gesetzt.
- Wenn Sie einen doppelten Drehtisch verwenden, wird der neue Wert für den auf der Symbolleiste **Aktueller Drehtisch** aktivierten Drehtisch gesetzt. Weitere Informationen finden Sie unter "Symbolleiste "Aktueller Drehtisch"".

PC-DMIS besitzt zwei interne Tischdefinitionen, um doppelte und gestapelte Tischkonfigurationen zu berücksichtigen. Für einen Einzeltisch wird die zweite Tischdefinition nicht verwendet. Da es sich um eine interne Definition handelt, wird kein Fehler auftreten, wenn Sie für einen Einzeltisch den Tisch V bestimmen, jedoch wird dies nicht empfohlen. Die Werte, die diese Funktion setzt, sind allgemein nützlich, da der Tisch nicht wirklich existiert.

TUTORELEMENT

Diese Funktion verarbeitet einen Parameter, entweder eine Zahl oder eine Zeichenfolge (eine Zeichenfolge wäre beispielsweise die ID eines Elements).

TUTORELEMENT(<PARAMETER>)

Diese Funktion arbeitet mit dem Variablentyp *Strukturen*. Eine Beschreibung der Strukturen und Unterelemente finden Sie unter "Strukturen".

Beispiele:

ZUWEISEN/E=TUTORELEMENT (1)	Erstellt eine einzelne TutorElement-Struktur.
ZUWEISEN/WM=TUTORELEMENT (n)	Erstellt für jede Zahl über 1 einen Array von <i>n</i> TutorElement-Strukturen.

<code>ZUWEISEN/KREIS1E=TUTORELEMENT ("KREIS1")</code>	Kopiert Daten aus dem Element KREIS1 in die TutorElement-Strukturen.
---	--

Die TutorElement-Struktur verfügt derzeit über die folgenden Unterelemente:

Unter-Element	Beschreibung
ID	Zeichenfolge der Element-ID
TYP	GANZZAHL (FTYP)
X, Y, Z	X-, Y- und Z-Koordinatenwerte
PR	Polarradius
PA	Polarwinkel
CX	I
CY	J
CZ	K
DM	Durchmesser 1
DM2	Durchmesser 2
DS	Abstand vom Nullpunkt
A	Winkel
AXY	Winkel in der XY-Ebene
AYZ	Winkel in der YZ-Ebene
AZX	Winkel in der ZX-Ebene

F	Formfehler
SDEV	Standardabweichung
TP	Positionieren

Beispiele für Funktionen

Bei der Erstellung und Verwendung Ihrer eigenen Funktionen könnten folgende unterschiedliche Funktionsbeispiele hilfreich sein:

- Beispiel für allgemeine Funktionen
- Beispiel für als Variable übergebene Funktionen
- Beispiel für Funktionen mit mehreren Parametern
- Beispiel für Funktionen, die andere Funktionen erstellen
- Beispiel für Funktionen als Mitglied eines Arrays
- Beispiel für rekursiv definierte Funktionen

Beispiel für allgemeine Funktionen

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

Erstellt eine benutzerdefinierte Funktion und weist sie der Variablen MEINEFUNK zu. Die Funktion nimmt drei Parameter: X, Y und Z.

- X wird mit 3 multipliziert.
- Y wird mit 2 multipliziert.
- Z enthält einfach den übergebenen Wert.

Sie Summe von $X + Y + Z$ wird zurückgegeben, wenn die Werte mit der Funktion verarbeitet werden. Beispiel:

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

Weist V1 durch Auswerten der an die Funktion MEINEFUNK(7,2,5) übergebenen Parameter den Wert 30 zu.

- 7 ist der Parameter und ersetzt X im Ausdrucksteil der Funktionsdefinition. So wird $X*3$ zu $7*3$ oder zu 21.
- 2 ersetzt Y. So wird $Y*2$ zu $2*2$ oder 4.
- 5 ersetzt Z.

Alle Werte werden nun addiert ($21 + 4 + 5$) und an V1 übergeben.

Beispiel für als Variable übergebene Funktionen

Funktionen können als Variablen übergeben werden. Das folgende Beispiel baut auf das oben stehende Beispiel für allgemeine Funktionen auf:

```
ASSIGN/NEWFUNC=MYFUNC
```

Stellt für die Variable NEWFUNC die Funktion ein, die auch MYFUNC verwendet.

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

Weist V3 den Wert 43 von den ausgewerteten Ausdrücken innerhalb der Funktion ($36 + 4 + 3$) zu.

Beispiel für Funktionen mit mehreren Parametern

Funktionen können über mehrere Parameter verfügen:

```
ASSIGN/ADDANDDOUBLE=FUNCTION ( (A,B) , 2* (A+B) )
```

Erstellt eine Funktion und weist dieser die Variable ADDANDDOUBLE zu. Die Funktion addiert zwei Parameter und multipliziert das Ergebnis mit 2.

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

Weist V2 den Wert 18 zu. Die Parameter 4 und 5 werden im Ausdrucksteil der Funktion eingesetzt. Ergebnis: $2*(4+5)$.

Beispiel für Funktionen, die andere Funktionen erstellen

Funktionen können andere Funktionen erstellen.

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

Definiert COMPOSE als Funktion, die zwei Funktionen als Parameter verwendet, mit denen sie eine neue Funktion erstellt.

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

Definiert ADD2 als Funktion, die 2 zu dem übergebenen Parameter hinzuaddiert.

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

Definiert ADD3 als Funktion, die 3 zu dem übergebenen Parameter hinzuaddiert.

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

Definiert ADD5 als Funktion, die sich aus den Funktionen ADD2 und ADD3 zusammensetzt.

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

Weist V5 den Wert V8 zu.

Beispiel für Funktionen als Mitglied eines Arrays

Funktionen können Mitglieder eines Array sein.

```
ASSIGN/ANARRAY=ARRAY (3, FACTORIAL, "Hallo Welt", ADD5)
```

Definiert ANARRAY als Array aus 4 Elementen: einer Zahl (3), einer Funktion (FACTORIAL), einer Zeichenfolge ("Hallo Welt") und einer Funktion (Add5).

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

Das zweite Element von ANARRAY ist die Funktion FACTORIAL. Der Parameter 4 wird an diese Funktion übergeben, und das Ergebnis von 24 wird V6 zugewiesen.

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

Von innen nach außen: Das erste Element von ANARRAY (3) wird an die Funktion des vierten Array-Elements (Add5) übergeben. Das Ergebnis, 8, wird an die Funktion des zweiten Array-Elements (FACTORIAL) übergeben und V7 zugewiesen. V7 erhält einen Wert von 40320.

Beispiel für rekursiv definierte Funktionen

Funktionen können rekursiv definiert werden. (Das heißt, sie können so definiert werden, dass sie sich selbst aufrufen.)

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

Erstellt ein Faktorprodukt, die einen Parameter aufnimmt. Wenn der Parameter kleiner als oder gleich 1 ist, wird er als 1 ausgewertet, andernfalls wird er als X multipliziert mit dem FACTORIAL von X-1 ausgewertet.

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

Weist V4 den Wert von 120 (5 x 3 x 2 x 1) zu.

Beispiel eines Geradenelements, das aus einem Scansegment erstellt wurde

Dieses Kapitel enthält ein Beispiel zur Anwendung der Ausdruckssprache in PC-DMIS, insbesondere der Funktionen für Geradensegmente, zum Exportieren von Anfangspunkt- und Endpunktnummern für Geradensegmente innerhalb eines Scans. Des Weiteren erhalten Sie Informationen zur Erstellung Ihres eigenen Geradenelements unter Verwendung der extrahierten Punkte innerhalb eines abhängigen Elements. Die in diesem Beispiel erläuterten Richtlinien gelten auch für die Erstellung eines Bogensegments aus einem Scan.

Angenommen, die Messroutine hat ein Scanelement mit Namen SCN1, das so aussieht:



```

SCN1 =ELEM/SCAN,OFFENE_LINIE,MESSPUNKTE
EINBLENDEN=NEIN,ALLEPARAMANZEIGEN=JA
  AUSFÜHRART=LERNEN,
  NENNW_MODUS=NW_SUCHE,SICHERHEITSEBENE=NEIN,
  EINZELPUNKT=NEIN,STÄRKE=0
  NW_SUCHE=5,NURMARKIERTE=NEIN,
  BESTEEINPASSUNGVERWENDEN=NEIN,TASTERKOMP=JA,
  RELATIVBEWEGUNG=NEIN,ABSTAND=0,CAD_Kompensation=NEIN
  RICHT1=VARIABLE,
  MESSPKTART=VEKTOR
  ANFANGSVEKTOR=0,-1,0
  RICHTUNGSVEKTOR=1,0,0
  SCHNITTEBENENVEKTOR=0,0,1
  ENDVEKTOR=0,-1,0
  EBENENVEKTOR=-1,0,0
  PUNKT1=100,0,-5
  PUNKT2=70,0,-5
MESS/SCAN
BASIS_SCAN/LINIE,MESSPUNKTE
EINBLENDEN=NEIN,ALLEPARAMANZEIGEN=JA
<100,0,-5>,<70,0,-
5>,Schnittebenenvektor=0,0,1,Richtungsvektor=1,0,0
Anfangsvektor=0,-1,0,Endvektor=0,-1,0,STÄRKE=0
FILTER/NULLFILTER,
AUSFÜHRART=LERNEN
BEREICHSGRENZE/EBENE,<70,0,-5>,Ebenenvektor=-
1,0,0,Überschreitungen=2
MESSPKTART/VEKTOR
NENNW_MODUS=NW_SUCHE,5
ENDESCAN
ENDEMESS/

```

Zur Erstellung einer LINIEn aus diesem Scan müssen Sie die Funktionen LINESEGMENTSTARTINDEX und LINESEGMENTENDINDEX verwenden, um die Daten auf folgende Weise zu extrahieren:



```

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

```

Dadurch wird PC-DMIS veranlasst, zum Scan "SCN1" vorzurücken und die Anfangs- und Endindexwerte, die sich innerhalb des vorgegebenen Toleranzbereichs befinden, aus dem ersten Geradensegment zu extrahieren. Diese Indexwerte werden dann den Variablen LINIENANFANGSINDEX und LINIENENDINDEX zugewiesen.

Ausdruckskomponenten

Nachdem die Anfangs- und Endindexwerte für das Geradensegment den Variablen zugewiesen wurden, können Sie diese Variablen innerhalb einer abhängigen Geraden so anwenden:



```
LIN4 =ELEM/GERADE,KART,UNBEGR  
NENN/100.225,0,-5.011,1,0,0  
MESS/100.225,-0.005,-5.011,1,-0.0000388,0  
ABHÄNGIG/LINIE,BE,2D,SCN1.HIT[LINESTARTINDEX..LINEEND  
INDEX],,  
AUSREISSER_ENTFERNEN/AUS,3  
FILTER/AUS,WELLENLÄNGE=0
```

Beachten Sie, dass PC-DMIS im hervorgehobenen Code des oben stehenden LINIElements die aus dem Scan extrahierten Anfangs- und Endnummern zur Erstellung des Elements verwendet:

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

Erzwungene Umwandlung von Operanden

Operanden können in andere Typen umgewandelt werden, indem ein beliebiger Umwandlungsoperator verwendet wird:

Erzwungene Umwandlung in eine Ganzzahl

INT(<Ausdruck>) – wandelt den Ausdruckswert in den Typ Ganzzahl um.

INT(4)	Ergibt 4.
INT(4.5)	Ergibt 4.
INT("Hallo Welt")	Ergibt 0.
INT("2")	Ergibt 2.
INT("2.2")	Ergibt 2.
INT("3 Blinde Mäuse")	Ergibt 3.
INT("Die 3 blinden Mäuse")	Ergibt 0.
INT("3, 4, 5")	Ergibt 3.
INT(MPOINT(0, 0, 1))	Ergibt den Abstand des Punkts vom Nullpunkt, in diesem Fall 1.

<code>INT (MPOINT (3, 4, 5))</code>	Der Abstand wird auf 7,0711 geschätzt. Dieser Ausdruck ergibt 7.
-------------------------------------	--

"Double"-Zwangsumwandlung

DOUBLE(<Ausdruck>) - wandelt den Ausdruckswert in den Typ "Double" um.

<code>DOUBLE (4)</code>	Ergibt 4,0.
<code>DOUBLE (4.5)</code>	Ergibt 4,5.
<code>DOUBLE ("Eine Zeichenfolge")</code>	Ergibt 0,0.
<code>DOUBLE ("3.5")</code>	Ergibt 3,5.
<code>DOUBLE ("3.5 Zoll")</code>	Ergibt 3,5.
<code>DOUBLE ("Der Kreis misst 3.5 Zoll im Durchmesser")</code>	Ergibt 0,0.
<code>DOUBLE (MPOINT (0, 0, 1))</code>	Ergibt 1,0.
<code>DOUBLE (MPOINT (3, 4, 5))</code>	Ergibt 7,0711.

Erzwungene Umwandlung in eine Zeichenfolge

STR(<Ausdruck>) – wandelt den Ausdruckswert in den Typ "Zeichenfolge" um.

<code>STR (4)</code>	Ergibt "4".
<code>STR (4.5)</code>	Ergibt "4.5".
<code>STR ("Hallo Welt")</code>	Ergibt "Hallo Welt".
<code>STR (MPOINT (3, 4, 5))</code>	Ergibt "3, 4, 5".

Erzwungene Umwandlung in einen Punkt

MPOINT(<Ausdruck1>, <Ausdruck2>, <Ausdruck3>) – wandelt Ausdruckswerte in den Typ "Punkt" um – nachdem jeder einzelne Ausdruck in den Typ "Double" umgewandelt wurde.

MPOINT (1, 1, 1)	Ergibt Punkt 1,0, 1,0, 1,0.
MPOINT (1.1, 1.1, 1.1)	Ergibt Punkt 1,1, 1,1, 1,1.
MPOINT ("1", "1", "1")	Ergibt Punkt 1,0, 1,0, 1,0.
MPOINT (3, 4.5, "5.6")	Ergibt Punkt 3,0, 4,5, 5,6
MPOINT (MPOINT (1, 0, 0), MPOINT (0, 1, 0), MPOINT (3, 4, 5))	Ergibt 1.0, 1.0, 7.0711

Operanden-Zwangsumwandlung und Ausdrücke gemischter Art

Der Ausdrucksauswerter wandelt Variablen in Ausdrücken gemischter Art automatisch um. Wenn das Ergebnis eines Ausdrucks wegen der automatisch erzwungenen Umwandlung nicht den Erwartungen entspricht, können Sie das gewünschte Ergebnis unter Umständen durch die Verwendung von Umwandlungsoperatoren erzielen. Die folgenden Beispiele zeigen eine automatisch erzwungene Umwandlung in Ausdrücken gemischter Art.

"KREIS" + 1

Ergibt "KREIS1".

"2" + 2

Ergibt 4.

"Der Wert von 2+2 ist " + (2 + 2)

Ergibt "Der Wert von 2+2 ist gleich 22". Das liegt daran, dass PC-DMIS Ausdrücke von links nach rechts ausgewertet, und der linke Teil des Ausdrucks ist eine Zeichenkette.

"Der Wert von 2+2 ist " + (2 + 2)

Ergibt "Der Wert von 2+2 ist gleich 4"

GERADE1.XYZ > 2

Ergibt 1, wenn der Abstand des Flächenmittelpunkts von GERADE1 zum Nullpunkt größer als 2 ist.

GERADE1.XYZ > GERADE2.XYZ

Ergibt 1, wenn der Abstand des Flächenmittelpunkts von GERADE1 zum Nullpunkt größer als der Abstand des Flächenmittelpunkts von GERADE2 ist.

GERADE1.XYZ = GERADE2.XYZ

Ergibt 1, falls die Flächenmittelpunkte von GERADE1 und GERADE2 gleich sind (in diesem Fall erfolgt keine erzwungene Typumwandlung)

DOUBLE(GERADE1.XYZ) = DOUBLE(GERADE2.XYZ)

Ergibt 1, wenn die Flächenmittelpunkte denselben Abstand zum Nullpunkt aufweisen.

11% 3.1

Ergibt 2 (% ist der für den Einsatz von Ganzzahlen entwickelte Modulo-Operator. Er gibt den Rest aus einer ganzzahligen Division zurück. $11\%3 = 2$.)

KREIS1.MESSPUNKT [3.2].X

Ergibt gemessenen X-Wert des 3. Messwertes auf Kreis1. Das Argument 3.2 wird automatisch in eine Ganzzahl mit dem Wert 3 umgewandelt.

ID-Ausdrücke

Viele PC-DMIS-Befehle verwenden Element-IDs als Parameter. Erstellte Elemente verwenden beispielsweise IDs, um anzugeben, welche Elemente als Eingabe für das erstellte Element zu verwenden sind. ID-Ausdrücke ermöglichen es dem Benutzer, Bezug zu nehmen auf: eine bestimmte Instanz eines Elements, eine Gruppe von Elementen mit ähnlichen Namen, eine Instanz eines Elements innerhalb des Aufrufs eines Unterprogramms oder ein Element in einer externen Messroutine.

Element-Array-ID

Mit Hilfe einer Element-Array-ID können Sie auf eine bestimmte Instanz eines Elements oder einen Bereich von Elementinstanzen Bezug nehmen. Wenn das Element "Kreis1" beispielsweise innerhalb einer WHILE-Schleife steht, die fünfmal ausgeführt wird, existiert beim Verlassen der Schleife ein Array aus fünf Instanzen des Kreises. Um auf eine der fünf Instanzen von "Kreis1" Bezug zu nehmen, verwenden Sie die Element-Array-Syntax gemäß der Beschreibung unter "Element-Arrays"; wonach "Kreis1[1]" auf die erste Instanz und "Kreis1[2]" auf die zweite Instanz usw. Bezug nehmen würde. Für einen Bezug auf einen Instanzenbereich verwenden Sie die Punkt-Punkt-Schreibweise (..). "Kreis1[1..3]" verweist auf den Bereich von der ersten bis zur dritten Instanz von Kreis1. "Kreis1[3..5]" verweist auf den Bereich von der dritten bis zur fünften Instanz von Kreis1. "Kreis1[1..5]" verweist auf den Bereich von der ersten bis

zur fünften Instanz von Kreis1. Wenn sich auf zahlreiche Elemente bezogen wird, wird der Satz als abhängiger Satz behandelt und verhält sich entsprechend.

ID-Stellvertreterzeichen

Mit Hilfe von ID-Stellvertreterzeichen können Sie auf einen Satz ähnlich benannter Elemente Bezug nehmen. Es gibt zwei Stellvertreterzeichen: "*" und "?". (Weitere Informationen hierzu finden Sie unter "Auswählen von Elementen durch Metazeichen-Entsprechung" im Abschnitt "Bearbeiten der CAD-Anzeige".)

Das Sternchen "*" steht für 0 oder mehr Instanzen beliebiger Zeichen. Um auf den Satz von Elementen Bezug zu nehmen, deren IDs mit den Buchstaben "KREIS" beginnen, verwenden Sie den ID-Ausdruck "KREIS*". Mit dieser Syntax wird ein Satz von Elementen erstellt, der alle Elemente umfasst, deren ID mit "KREIS" beginnt, beispielsweise "KREIS1", "KREIS2", "KREIS3" oder "KREIS".



Wenn KREIS3 mehrere Ausführungen hat, wird nur die letzte Messung verwendet. Um auf die verschiedenen Instanzen der Ausführungen Bezug zu nehmen, verwenden Sie einen Ausdruck der folgenden Art: KREIS?[1..3]

Das Fragezeichen '?' nimmt Bezug auf einer einzigen Instanz eines beliebigen Zeichens.



Der ID-Ausdruck "XX???1" erstellt einen Satz von Elementen, deren IDs sechs Zeichen lang sind, mit "XX" beginnen und mit "1" enden, beispielsweise "XXKRE1", "XXKEG1", "XXKIN1" oder "XXYY21".

IDs für Elemente innerhalb von Unterprogrammen, BASIC-Skripts oder externer Messroutinen

Unterprogrammen können sich in der aktuellen Messroutine oder in einer externen Messroutine befinden. Wenn sich die Unterroutine in derselben Messroutine wie der Aufruf der Unterroutine befindet, kann die unter "Element-Arrays:" beschriebene Syntax für Element-Array-IDs verwendet werden, um auf einzelne Instanzen eines in der Unterroutine erstellten Elementes zu verweisen. Befindet sich die Unterroutine dagegen in einer externen Messroutine, kann folgende Syntax zur Bezugnahme auf Elemente verwendet werden, die in der Unterroutine erstellt wurden: "<ID des Unterprogramms aufrufen>:<Element-ID>".

Wenn sich beispielsweise ein Element mit dem Namen "F1" in einem externen Unterprogramm befindet, das mit dem Befehl "AUFR_UNTERPROG" mit der ID "CS1" aufgerufen wurde, dann kann die Bezugnahme auf dieses Element über den ID-Ausdruck "CS1:F1" verwendet werden.



Das folgende Beispiel veranschaulicht lediglich die Verwendung der Syntax "CS1.F1" und ist nicht als tatsächliches Anwendungsbeispiel gedacht.

Routine 1: PLUS1.PRG

```

UNTERPROGRAMM/PLUS1, A1 = 0, A2 = 0, A3 = 0

F1=ELEM/PUNKT,KART

NENN /A1+1,A2+1,A3+1,0,0,1

MESS/3,1,1,0,0,1

MESS/PUNKT,1

MESSPKT /BASIS,A1+1,A2+1,A3+1,0,0,1,0,0,0

ENDEMESS/

ENDE_UNTERPROG/

```

Routine 2: TEST.PRG

```

CS1 =AUFR_UNTERPROG/PLUS1,D:\V30\WINDEBUG\PLUS1.PRG: 3,3,3,,

MERKM D1= LAGE DES PUNKTES CS1:F1 EINHEITEN=ZOLL,$

GRAFIK=OFF TEXT=OFF MULT=10.00 AUSGABE=BEIDE

ACH NENNWERT OTOL UTOL MESS MAX MIN ABW AUSTOL

X 3.0000 0.0000 0.0000 3.0000 3.0000 3.0000 0.0000 0.0000

----#----

ENDE VON MERKMAL D1

```

BASIC-Skripts erstellen und löschen Objekte dynamisch. Verwenden Sie die Syntax "<BASIC-Skript-ID>:<Element-ID>", um auf ein Element zu verweisen, das mit einem BASIC-Skript erstellt wurde. Wenn beispielsweise ein BASIC-Skript mit der ID "BS1" ein Element mit der ID "F2" erstellt, verwenden Sie den ID-Ausdruck "BS1:F2", um auf dieses Element zu verweisen.

Zugreifen auf die Objekteigenschaften eines Protokolls

Externe Messroutinen können mit Hilfe des Befehls ANHÄNGEN an PC-DMIS angehängt werden. Verwenden Sie die folgende Syntax, um auf Elemente in solchen Routinen zu verweisen: "<Routine-ID anhängen>:<Element-ID>". Um auf Element "F3" in der externen Messroutine mit der ID "GEAR1" Bezug zu nehmen, verwenden Sie den Ausdruck "GEAR1:F3". (Weitere Informationen hierzu finden Sie unter "Anhängen einer externen Messroutine" im Abschnitt "Hinzufügen externer Elemente".)

ID-Ausdrucks-Kombinationen

Array-ID-Ausdrücke, Stellvertreterzeichen-ID-Ausdrücke sowie externe Unterprogramm-, Basic-Skript- und externe Messroutine-ID-Ausdrücke können miteinander kombiniert werden. Um beispielsweise auf die dritte Instanz von allen Elementen in einer externen Messroutine mit der ID "BOLZMUST", die mit den Buchstaben "KREIS" beginnen, zu verweisen, verwenden Sie den ID-Ausdruck "`BOLZMUST:KREIS*[3]`".

ID-Ausdrücke können auch in regulären Ausdrücken verwendet werden. Der gemessene Flächenmittelpunkt der obigen Elementgruppe könnte einer Variablen also mit dem folgenden Ausdruck zugewiesen werden:

```
ZUWEISEN/V1=BOLZMUST:KREIS*[3].XYZ
```

ID-Ausdrücke können auch in regulären Ausdrücken verwendet werden. Der gemessene Flächenmittelpunkt der obigen Elementgruppe könnte einer Variablen also mit dem folgenden Ausdruck zugewiesen werden:

```
ZUWEISEN/V1=BOLZMUST:KREIS*[3].XYZ
```

Zugreifen auf die Objekteigenschaften eines Protokolls

Sie können Ihre eigenen benutzerdefinierten Protokolle und Bezeichnungsvorlage erstellen. PC-DMIS verwendet diese, um Protokolldaten in einem Protokollfenster (siehe **Ansicht | Protokollfenster**) anzuzeigen. Sie können bestehende Vorlagen im Protokollvorlagen-Editors modifizieren oder neu erstellen. Der Editor verwendet eine Visual-Basic-ähnliche Oberfläche mit der Sie besondere Komponenten, genannt "Objekte", einfügen, verschieben und in ihrer Größe anpassen können.

Jedes Objekt besteht aus "Eigenschaften", die bestimmen, auf welche Weise dieses Objekt angezeigt wird und welche Informationen es enthält. Einige dieser Eigenschaften gelten für alle Objekte, andere nur für verwandte Objekte und manche treffen auf nur ein einziges Objekt zu.

Die PC-DMIS-Ausdruckssprache ist in der Lage, das aktuell geladene Protokoll zu befragen und Eigenschaftswerte eines bestimmten Objekts in einer Variablen zu speichern. Sie kann Werte vom Typ "Zeichenfolge", "Ganzzahl" und "reelle Zahl" durch folgende Syntax erfassen:

Eigenschaftsanfrage-Syntax



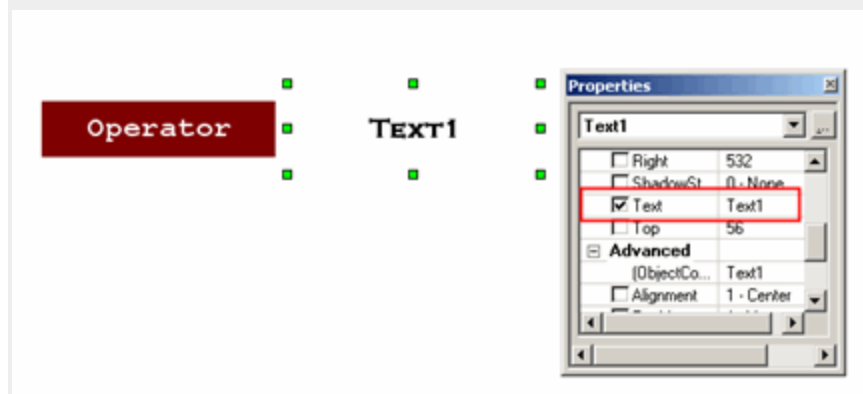
```
ZUWEISEN/V1=Protokoll.<Objektname>.<Eigenschaftsname>
```

Protokoll bezieht sich auf das aktuell geladene Protokoll. <Objektname> ist der eindeutige Name des Objekts. <Eigenschaftsname> ist ein gültiger Eigenschaftsname für das Objekt.

Beispiel



Angenommen, Ihre Protokollvorlage enthält ein Textobjekt namens "Text1", das im Abschlussprotokoll den Namen des Bedieners anzeigen soll. PC-DMIS speichert die tatsächliche Zeichenfolge, die den Namen des Bedieners darstellt, in der **Texteigenschaft** des Objekts. Standardmäßig besitzt die Texteigenschaft (angezeigter Text) den Ausgangswert "Text1" (siehe folgende Abbildung). Da es sich um eine vom Benutzer zugewiesene Eigenschaft handelt, ändert sich der Wert der Eigenschaft, wenn Sie den Namen während der Ausführung eingeben.



Dialogfeld "Eigenschaften", das ein ausgewähltes Objekt und die Abfrageeigenschaft anzeigt

Um mittels des Ausdruckssprache-Codes die "Text"-Eigenschaft dieses Textobjektes anzufragen und die eingegebenen Daten zu erhalten, sähe der von Ihnen hierzu verwendete Befehl folgendermaßen aus:

```
ZUWEISEN/V1=Protokoll1.Text1.Text
```

In diesem Code:

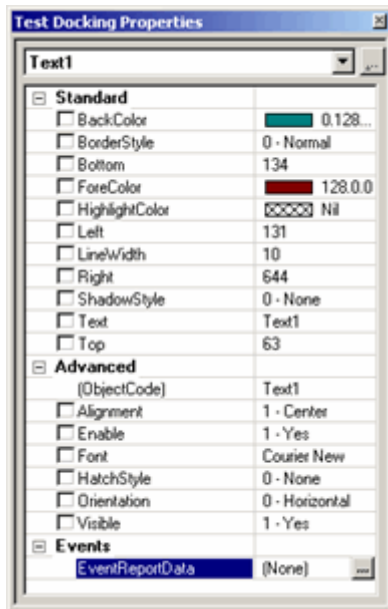
"Protokoll" weist den Code darauf hin, dass in dem Protokollfenster geladene Protokoll anzuschauen.

"Text1" ist eine Anweisung, das Objekt "Text1" zu suchen.

"Text" bedeutet, dass die "Text"-Eigenschaft innerhalb des Objekts gesucht werden soll. Der Wert der "Text"-Eigenschaft wird daraufhin an die V1-Variable übergeben und kann anschließend weiter bearbeitet oder mit Hilfe der PC-DMIS-Ausdruckssprache angezeigt werden.

Suche nach den Eigenschaften

Sie können herausfinden, welche Eigenschaften mit einem bestimmten Objekt verbunden sind, indem Sie auf die Protokollvorlage im Protokollvorlageneditor (**Datei | Protokollieren | Bearbeiten | Protokollvorlage**) zugreifen, das Objekt auswählen und dann mit der rechten Maustaste auf das Objekt klicken, um das Eigenschaftenblatt anzuzeigen.



Eigenschaftenblatt eines Textobjektes

Das Eigenschaftenblatt enthält zwei Spalten. Die linke Spalte zeigt den Eigenschaftsnamen an. Die rechte Spalte zeigt den aktuellen Wert an. Stellen Sie sicher, dass im Ausdruckscode der exakte Eigenschaftsname verwendet wird.



Bei der Abfrage von Eigenschaftswerten könnten Sie feststellen, dass manche Eigenschaften einen scheinbar nutzlosen numerischen Wert zurückgeben. Normalerweise geschieht dies, wenn die Eigenschaft eine Liste verfügbarer Optionen aufweist. PC-DMIS gibt einen internen Wert für die ausgewählte Eigenschaft zurück, die keinen Bezug zur angezeigten Eigenschaft hat.

Das **Text**objekt hat beispielsweise eine Eigenschaft **Ausrichtung** mit folgenden Werten:

- 0 - Horizontal
- 1 - Vertikal Aufwärts
- 2 - Vertikal Abwärts

Wenn Sie jedoch den Wert, der die Ausdruckssprache von PC-DMIS verwendet, erhalten, dann wird stattdessen dies zurückgegeben:

- 0 (für Horizontal)
- 900 (für Vertikal nach oben)
- -900 (für Vertikal nach unten)

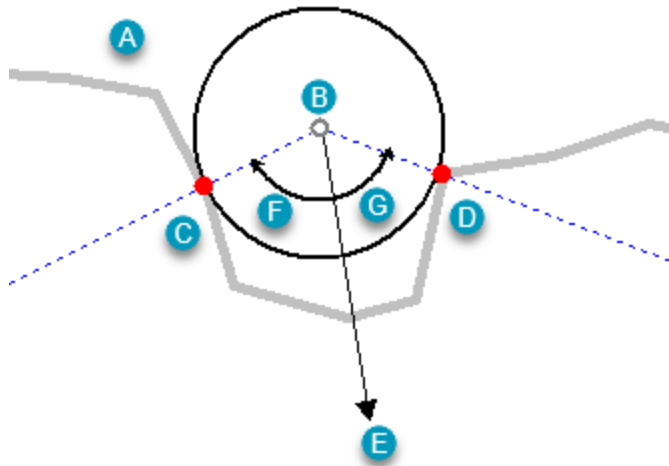
Zur Bestimmung darüber, welche zurückgegebenen Werte dem Wert auf dem Eigenschaftenblatt entsprechen, ist ein systematisches Ausprobieren ("Trial and Error") erforderlich.

Aufrufen von Informationen aus einem erstellten Kreisminimum eines Scans

Mit den Ausdrücken in PC-DMIS können Sie Informationen aus einem Kreiselement, das mit einem vorgegebenen Radius an einem Mindestpunkt entlang eines linearen Scans erstellt wird, extrahieren. Siehe das Thema "Erstellen eines Kreises am minimalen Punkt eines Scans" im Abschnitt "Erstellen von neuen Elementen aus vorhandenen Elementen".

Beim Erstellen eines Kreiselements am minimalen Punkt eines Scans verwendet der Kreis letztendlich einen Vektor (der als 'nach unten verlaufender Vektor' bezeichnet wird), um den Kontakt mit der Scanlinie herzustellen. Die Linie wird nur an zwei Stellen berührt, den sogenannten Kontaktpunkten (KONTAKTPUNKT1 und

KONTAKTPUNKT2). PC-DMIS kann diese Punkte anschließend zur Bestimmung der Winkel vom nach unten verlaufenden Vektor zu diesen Kontaktpunkten (KONTAKTWINKEL1 und KONTAKTWINKEL2) benutzen. Sehen Sie sich das folgende Diagramm an:



A - Die Scanlinie, an der der Kreis erstellt wird.

B - Die endgültige XYZ-Position des Kreismittelpunktes.

C - Der Kontaktpunkt auf der linken Seite des nach unten verlaufenden Vektors. Er wird als CONTACTPOINT1 bezeichnet.

D - Der Kontaktpunkt auf der rechten Seite des nach unten verlaufenden Vektors. Er wird als CONTACTPOINT2 bezeichnet.

E - Nach unten verlaufender Vektor.

F - Der Winkel vom nach unten verlaufenden Vektor zum CONTACTPOINT1. Dieser wird als CONTACTANGLE1 bezeichnet.

G - Der Winkel vom nach unten verlaufenden Vektor zum CONTACTPOINT2. Dieser wird als CONTACTANGLE2 bezeichnet.

Die in der unten stehenden Tabelle beschriebenen Ausdrücke funktionieren nur zusammen mit dieser Art eines erstellten Kreiselements. Sie können in der unten stehenden Syntax auch CONTACTPOINT2 verwenden, um stattdessen äquivalente Angaben unter Verwendung des zweiten CONTACTPOINTes zu erhalten.

`ZUWEISEN/V1=KREIS1.CONTACTPOINT1.XYZ`

Gibt die XYZ-Punktangaben für den ersten Kontaktpunkt des Kreises mit der Linie, CONTACTPOINT1, zurück.

Aufrufen von Informationen aus einem erstellten Kreisminimum eines Scans

`ZUWEISEN/V1=KREIS1.CONTACTPOINT1.X`

Gibt die X-Punktangaben für den ersten CONTACTPOINT des Kreises mit der Linie, CONTACTPOINT1, zurück.

`ZUWEISEN/V1=KREIS1.CONTACTPOINT1.Y`

Gibt die Y-Punktangaben für den ersten CONTACTPOINT des Kreises mit der Linie, CONTACTPOINT1, zurück.

`ZUWEISEN/V1=KREIS1.CONTACTPOINT1.Z`

Gibt die Z-Punktangaben für den ersten CONTACTPOINT des Kreises mit der Linie, CONTACTPOINT1, zurück.

`ZUWEISEN/V1=KREIS1.CONTACTPOINT1.IJK`

Gibt den IJK-Vektor vom CONTACTPOINT1 zum Kreismittelpunkt zurück.

`ZUWEISEN/V1=KREIS1.CONTACTPOINT1.I`

Gibt den I-Wert vom obigen IJK-Vektor von CONTACTPOINT1 zurück.

`ZUWEISEN/V1=KREIS1.CONTACTPOINT1.J`

Gibt den J-Wert vom obigen IJK-Vektor von CONTACTPOINT1 zurück.

`ZUWEISEN/V1=KREIS1.CONTACTPOINT1.K`

Gibt den K-Wert vom obigen IJK-Vektor von CONTACTPOINT1 zurück.

`ZUWEISEN/V1=KREIS1.CONTACTANGLE1`

Gibt den Winkel vom nach unten verlaufenden Vektor zum CONTACTPOINT1 zurück.

`ZUWEISEN/V1=KREIS1.CONTACTANGLE2`

Gibt den Winkel vom nach unten verlaufenden Vektor zum CONTACTPOINT2 zurück.

`ZUWEISEN/V1=KREIS1.CONTACTANGLE`

Gibt die Summe der absoluten Werte von CONTACTANGLE1 und CONTACTANGLE2 zurück. Der Wert sollte 180 Grad nicht übersteigen.