

Inhaltsverzeichnis

Verwenden der Datei-Eingabe/-Ausgabe	1
Verwenden der Datei-Eingabe/-Ausgabe: Einführung	1
Grundlegende Datei-I/O-Konzepte	2
Verwenden des Dialogfeldes "Datei I/O"	4
Öffnen einer Datei zum Lesen oder Schreiben	4
Beispielcode für "Datei öffnen"	5
Schließen einer geöffneten Datei nach dem Lesen oder Schreiben	6
Beispielcode für "Datei schließen"	7
Lesen eines Zeichens aus einer Datei	8
Beispielcode für "Zeichen lesen"	9
Lesen einer Zeile aus einer Datei	11
Beispielcode für "Zeile lesen"	12
Lesen eines Textblocks aus einer Datei	18
Beispielcode für "Block lesen"	19
Lesen von Text bis zu einem Begrenzer	22
Beispielcode für "Lesen bis zu"	23
Schreiben eines Zeichens in eine Datei	25
Beispielcode für "Zeichen schreiben"	26
Schreiben einer Zeile in eine Datei	28
Beispielcode für "Zeile schreiben"	30
Schreiben eines Textblocks in eine Datei	31
Beispielcode für "Block schreiben"	33

Positionieren eines Dateizeigers am Anfang einer Datei	34
Beispielcode für "Zum Start zurück"	35
Speichern der aktuellen Position eines Dateizeigers	36
Beispielcode für "Dateiposition speichern"	37
Aufrufen der gespeicherten Position eines Dateizeigers	39
Beispielcode für "Dateiposition abrufen"	40
Kopieren einer Datei	41
Beispielcode für "Datei kopieren"	42
Verschieben einer Datei	44
Beispielcode für "Datei verschieben"	45
Löschen einer Datei	47
Beispielcode für "Datei löschen"	48
Prüfen des Vorhandenseins einer Datei	49
Beispielcode für "Datei vorhanden"	50
Anzeigen eines Dateidialogfelds	50
Beispielcode für "Dateidialogfeld"	51
Prüfen auf das Ende einer Datei oder Ende einer Zeile	52
Beispielcode für EOF und EOL	53

Verwenden der Datei-Eingabe/-Ausgabe

Verwenden der Datei-Eingabe/-Ausgabe: Einführung

In diesem Abschnitt wird beschrieben, wie Daten in Messroutinen eingegeben und daraus ausgegeben werden. Die verfügbaren Menüoptionen ermöglichen das Öffnen von Dateien im Lese- oder Schreibmodus. Daten können dann entweder aus diesen Dateien gelesen oder in sie geschrieben werden. Mit den Datei I/O-Befehlen können Daten aus externen Dateien eingelesen und in einer Messroutine verwendet werden. Mess- und Toleranzdaten können mit diesen Befehlen auch wieder in externe Dateien geschrieben werden. Außerdem können mit Hilfe dieser Befehle weitere Dateivorgänge durchgeführt werden.

Dieser Abschnitt enthält detaillierte Beschreibungen der I/O-Operationen sowie Anwendungsbeispiele für die einzelnen Operationen. Die Beispiele verwenden Elemente, die in den Abschnitten "Verzweigen mit Hilfe der Ablaufsteuerung" und "Verwenden von Ausdrücken und Variablen" beschrieben sind.

Dieser Abschnitt enthält die folgenden Hauptthemen:

- Grundlegende Datei-I/O-Konzepte
- Verwenden des Dialogfeldes "Datei I/O"
- Öffnen einer Datei zum Lesen oder Schreiben
- Schließen einer geöffneten Datei nach dem Lesen oder Schreiben
- Lesen eines Zeichens aus einer Datei
- Lesen einer Zeile aus einer Datei
- Lesen eines Textblocks aus einer Datei
- Lesen von Text bis zu einem Begrenzer
- Schreiben eines Zeichens in eine Datei
- Schreiben einer Zeile in eine Datei
- Schreiben eines Textblocks in eine Datei
- Positionieren eines Dateizeigers am Anfang einer Datei
- Speichern der aktuellen Position eines Dateizeigers
- Aufrufen der gespeicherten Position eines Dateizeigers

- Kopieren einer Datei
- Verschieben einer Datei
- Löschen einer Datei
- Prüfen des Vorhandenseins einer Datei
- Anzeigen eines Dateialogfelds
- Prüfen auf das Ende einer Datei oder Ende einer Zeile

Befehlsmodus-Befehle nach Kommentaren

Da viele der Codebeispiele in diesem Kapitel getippte **KOMMENTAR**-Befehle verwenden, sollten Sie Folgendes beachten:



Nachdem Sie einen PC-DMIS-Kommentar eingefügt haben, müssen Sie, um weitere PC-DMIS-Befehle im Befehlsmodus einzugeben, zunächst *zweimal* die Eingabetaste nach dem Befehl **KOMMENTAR** drücken. Dadurch wird PC-DMIS mitgeteilt, dass Sie dem Kommentar keinen Text mehr hinzufügen möchten, sondern bereit sind, einen neuen Befehl hinzuzufügen.

Grundlegende Datei-I/O-Konzepte

Prüfen des Vorhandenseins einer Datei:

Bei allen Datei-I/O-Operationen ist es sinnvoll, zunächst das Vorhandensein der Datei zu überprüfen. Dies könnte in Form einer IF/THEN-Schleife erfolgen, so dass der Benutzer bei einem negativen Prüfungsergebnis benachrichtigt wird. Beim Schreiben in eine Datei muss die Datei zuerst in der Windows-Umgebung erstellt werden.

Weitere Informationen hierzu finden Sie unter "Prüfen des Vorhandenseins einer Datei".

Öffnen und Schließen von Dateien:

Bei Operationen, die aus einer Datei lesen oder in eine Datei schreiben, müssen Sie die Datei zuerst für die Systemprozesse öffnen. Hierfür weisen Sie der Datei eine Variable, den sogenannten Dateizeiger, zu. Beim Öffnen einer Datei können Sie angeben, ob die Datei zum Lesen oder Schreiben (Überschreiben) oder zum Anhängen geöffnet wird. Nachdem die Datei geöffnet wurde, ist das Lesen oder Schreiben möglich. Wenn Sie die Arbeit an der Datei beendet haben, sollten Sie den Dateizeiger schließen. Dadurch wird die Datei geschlossen, und sie kann von anderen Systemprozessen verwendet werden. Dateien, die bereits von einem anderen Prozess geöffnet sind, können nicht geöffnet werden.

Weitere Informationen hierzu finden Sie unter "Öffnen einer Datei zum Lesen oder Schreiben" und unter "Schließen einer geöffneten Datei nach dem Lesen oder Schreiben".

Dateizeiger und Positionen:

Dateizeiger sind Variablen, die auf eine Datei verweisen. Sie speichern den Namen und den Speicherort einer geöffneten Datei und dienen zum Lesen aus oder Schreiben in diese Datei. Sobald eine Datei geöffnet und auf einen Dateizeiger eingestellt ist, verhält sich der Zeiger wie ein Cursor in einem Textverarbeitungsprogramm. Er zeigt an, an welcher Position in der Datei Sie gerade lesen oder schreiben.

- Wenn Sie Daten an eine Datei anhängen, sollte sich der Dateizeiger meist am Ende der Datei befinden.
- Wenn Sie eine Datei lesen oder überschreiben, sollte sich der Dateizeiger am Anfang der Datei befinden.

Verwenden von Begrenzern beim Lesen oder Schreiben

Beim Schreiben von Daten sollten Sie Begrenzer verwenden, um Datenteile voneinander zu trennen. Dies vereinfacht die spätere Einlesung der Daten in die Messroutine. Ein Begrenzer kann jedes Zeichen oder eine Zeichenfolge sein. Angenommen, Sie haben einen Punkt namens PKT1 mit den gemessenen X-,Y- und Z-Werten 2.5, 4.3 und 6.1. Sie können diese Werte durch Kommata voneinander getrennt (der Begrenzer ist in diesem Fall ein Komma) in eine Datendatei schreiben. Beispiel:



```
DATEI/ZEILE_SCHREIB,FPTR,PKT1.X + "," + PKT1.Y + ","  
+ PKT1.Z
```

Beim Lesen von Daten können Sie die hereinkommenden Daten auf Basis eines festgelegten Begrenzers trennen und zur späteren Bearbeitung in Variablen einfügen. Angenommen, Sie möchten dieselben X-,Y- und Z-Werte wie oben aufgeführt einlesen. Die Werte sollten in einer einzelnen Textzeile stehen: 2.5,4.3,6.1. Sie können den Text bei den Kommata trennen und die Werte mit Hilfe einer Codezeile in die entsprechenden Variablen einfügen. Ein Beispiel:

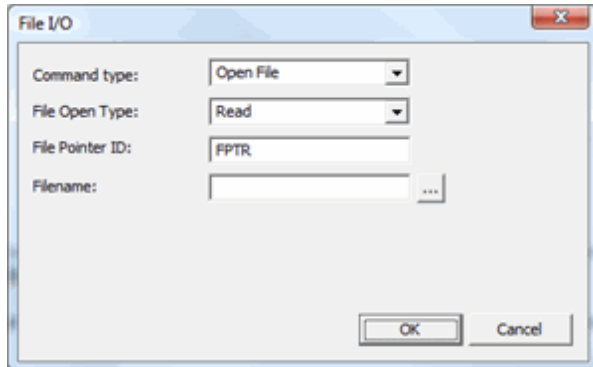


```
V1=DATEI/ZEILE_LESEN,FPTR,{WertX}+","+"{WertY}+","+"{We  
rtZ}
```

Anschließend können Sie `WertX`, `WertY` und `WertZ` als normale Variablen in der Messroutine verwenden. Ergebnis: `ValX = 2.5`, `ValY = 4.3`, und `ValZ = 6.1`.

Verwenden des Dialogfeldes "Datei I/O"

Alle Datei I/O-Befehle werden zuerst durch Auswahl der entsprechenden Datei I/O-Menüoption in die Messroutine eingefügt (wählen Sie **Einfügen | Datei I/O-Befehl** vom Menü). Nachdem der Befehl im Bearbeitungsfenster existiert, drücken Sie auf den Befehl "F9", um auf das zugehörige Dialogfeld **Datei I/O** zuzugreifen.



Dialogfeld E/A-Datei

Dieses Dialogfeld bietet einfach eine visuelle Art und Weise, den *aktuellen* Datei I/O-Befehl zu bearbeiten. Sie können aber auch einen Befehl innerhalb des Bearbeitungsfensters ändern, indem Sie die im Abschnitt "Verwenden des Bearbeitungsfensters" behandelten Methoden anwenden.

Sie sollten dieses Dialogfeld dazu verwenden, *neue* Datei I/O-Befehle einzufügen. Hierfür wählen Sie die entsprechende Menüoption aus oder geben die Befehle direkt ins Bearbeitungsfenster ein.

Öffnen einer Datei zum Lesen oder Schreiben

Die Menüoption **Einfügen | Datei I/O-Befehl | Datei öffnen** ermöglicht es Ihnen, einen Befehl in das Bearbeitungsfenster einzufügen, der während der Ausführung der Messroutine eine Datei von Ihrem Computer öffnet.

Dateien können geöffnet werden, um Daten einfach anzuzeigen, hinzuzufügen oder zu speichern.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:

Öffnen einer Datei zum Lesen oder Schreiben



`<Dateizeigername> =Datei/Öffnen,<Dateiname>,<Öffnen-Modus>`

Beschreibungen für einige Komponenten dieser Befehle:

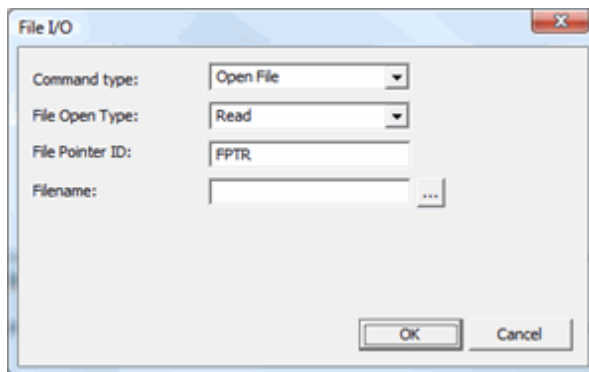
<Dateizeigername> - Die vom Benutzer gewählte ID des Dateizeigers, der zum Zugriff auf die geöffnete Datei verwendet wird. Diese ID dient zum Verweis auf die geöffnete Datei in anderen Datei I/O-Befehlen.

<Dateiname> - Der Dateiname der zu öffnenden Datenträgerdatei.

<Öffnen-Modus> - Der Modus, in dem die Datei geöffnet werden sollte. Dateien können in den folgenden Modi geöffnet werden: Lesen, Schreiben und Anhängen.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Datei öffnen".
3. Drücken Sie F9.



Beispielcode für "Datei öffnen"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Dieser Code öffnet eine Datei mit dem Namen TEST.TXT zum Lesen, Schreiben und Anhängen. Er speichert den Dateinamen zu einem Dateizeiger namens `FPTR`.



```
FPTR=DATEI/ÖFFNEN,C:\PCDMISW\TEST.TXT,LESEN
FPTR=DATEI/ÖFFNEN,C:\PCDMISW\TEST.TXT,SCHREIBEN
FPTR=DATEI/ÖFFNEN,C:\PCDMISW\TEST.TXT,ANHÄNGEN
```

Beachten Sie, dass Sie für die Eingabe des vollständigen Pfads einen Eingabekommentar verwenden und den Pfad dann in einen `DATEI/ÖFFNEN`-Befehl einsetzen können. Dasselbe ist auch bei einem `DATEI/DIALOGFENSTER`-Befehl möglich. Sehen Sie sich diese Beispiele an:



```
C1=KOMMENTAR/EINGABE,Geben Sie den vollständigen Pfad und
den Namen der Datei ein.
V1=DATEI/DIALOGFENSTER, WÄHLEN SIE EINE DATEI ZUM ÖFFNEN
AUS.
FPTR=DATEI/ÖFFNEN,C1.INPUT,LESEN
FPTR=DATEI/ÖFFNEN,V1,LESEN
```

Weitere Informationen hierzu finden Sie unter "Anzeigen eines Dateidialogfelds".

Schließen einer geöffneten Datei nach dem Lesen oder Schreiben

Mit der Menüoption **Einfügen | Datei I/O-Befehl | Datei schließen** können Sie einen Befehl in das Bearbeitungsfenster einfügen, durch den eine geöffnete Datei während der Ausführung der Messroutine geschlossen wird. Durch das Schließen von Dateien werden die von den geöffneten Dateien beanspruchten Ressourcen wieder freigegeben, und alle an der Datei vorgenommenen Änderungen müssen auf dem Datenträger gespeichert werden.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:

```
Datei/Schließen, <Dateizeigername>,<Schließmodus>
```

Beschreibungen für einige Komponenten dieser Befehle:

<Dateizeigername>

Die zur Kennung der Datei verwendete ID. Sie wird beim Öffnen der Datei erstellt.

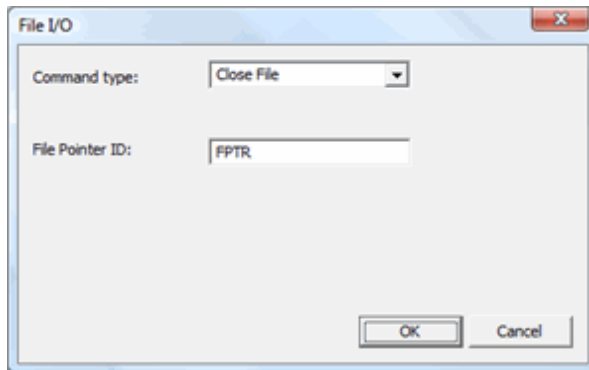
<Schließmodus>

Dieser Parameter verfügt über zwei Optionen: **BEIBEHALTEN** und **LÖSCHEN**. Durch die Option **BEIBEHALTEN** wird die im Dateizeiger definierte Datei einfach nur geschlossen. Durch die Option **LÖSCHEN** wird die Datei zuerst geschlossen und anschließend gelöscht.

Schließen einer geöffneten Datei nach dem Lesen oder Schreiben

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Datei schließen".
3. Drücken Sie F9.



Beispielcode für "Datei schließen"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Dieser Code schließt die Datei, die dem Dateizeiger `FPTR` zugewiesen ist.



```
DATEI/SCHLIESSEN, FPTR, BEIBEHALTEN
```

Unter Verwendung des Parameters `LÖSCHEN` schließt und löscht dieser Code die Datei, die `FPTR` zugeordnet ist:



```
DATEI/SCHLIESSEN, FPTR, LÖSCHEN
```

Lesen eines Zeichens aus einer Datei

Die Menüoption **Einfügen | Datei I/O-Befehl | Lesebefehle | Zeichen lesen** platziert einen Befehl in das Bearbeitungsfenster, durch den ein Einzelzeichen aus der im Dateizeiger-Namensfeld (siehe Syntax weiter unten). Der Befehl weist dieses Zeichen der Variablen zu, die in dem Variablennamensfeld angegeben ist.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



```
<Variablenname> =  
Datei/Zeichen_lesen,<Dateizeigername>
```

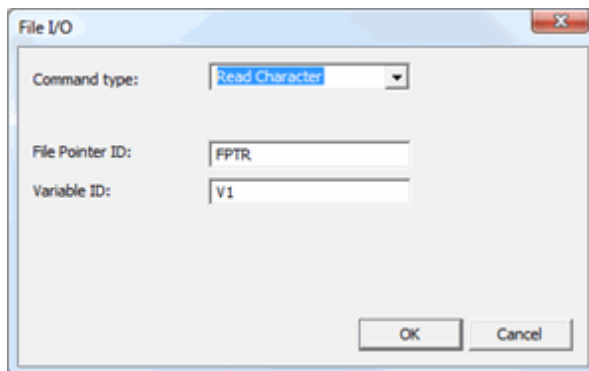
Beschreibungen für einige Komponenten dieser Befehle:

<Dateizeigername> - Die ID zum Öffnen der Datei.

<Variablenname> - Der Name der Variablen, die dieses Zeichen trägt.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Zeichen lesen".
3. Drücken Sie F9.



Beispielcode für "Zeichen lesen"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Sehen Sie sich dieses Beispiel an, in dem aus einer Zeile einer Datendatei immer ein Zeichen gelesen wird, bis ein Leerzeichen vorkommt.



```
V1=DATEI/VORHANDEN,test.txt
IF/V1<>0
    KOMMENTAR/BEDIENER,Lesen aus der Datei möglich.
    Zum Fortfahren auf 'OK' klicken.
    ZUWEISEN/V3=""
    FPTR=DATEI/ÖFFNEN,D:\Programmdateien\pcdmis35\tes
    t.txt,LESEN
    DO/
        V2=DATEI/ZEICH_LESEN,FPTR
        ZUWEISEN/V3=V3+V2
    UNTIL/V2==" "
    DATEI/SCHLIESSEN,FPTR
    KOMMENTAR/BEDIENER,"Das erste Wort aus einer
    Textzeile der Datei lautet: "+V3
END_IF/
ELSE/
    KOMMENTAR/BEDIENER,Lesen aus der Datei nicht
    möglich. Routine wird jetzt beendet.
    GEHZU/ENDE
END_ELSE/
ENDE=MARKE/
ROUTINE/ENDE
```

Erläuterung des Codes

V1=DATEI/VORHANDEN

Diese Zeile überprüft, ob die angegebene Datei vorhanden ist. Die Datei muss sich im Verzeichnis befinden, in dem PC-DMIS installiert ist, damit dieser Code funktioniert. Andernfalls muss die Zeile, welche die Datei enthält, auch den vollständigen Pfad der Datei enthalten. **v1** erhält das Ergebnis der Dateiprüfung. Ist die Datei vorhanden, ist der Wert ungleich Null. Andernfalls lautet er 0.

IF/V1<>0

Diese Zeile nimmt den Wert von `V1` und prüft, ob die Auswertung einen Wert ungleich Null ergibt. Ist dies der Fall, wird in einem Kommentar gemeldet, dass der Lesevorgang beginnen kann. Wenn gleich Null, wird die Messroutine beendet.

ZUWEISEN/V3=""

Diese Zeile erstellt eine leere Zeichenfolge und weist diese `V3` zu. Der Code verwendet diese Variable, um aus den einzeln eingelesenen Zeichen eine Zeichenfolge zu erstellen. Wenn Sie die leere Zeichenfolge nicht erstellen, weist `V3` den Standardwert 0 auf.

FPTR=DATEI/ÖFFNEN

Diese Zeile öffnet die angegebene Datei zum Lesen und weist sie dem Standarddateizeiger `FPTR` zu.

DO

Diese Zeile beginnt eine `DO/UNTIL`-Schleife. Sie begrenzt den `DATEI/ZEICH_LESEN`-Code, so dass die Zeichen fortlaufend einzeln eingelesen werden. Die Schleife wird beendet, wenn ein Leerzeichen eingelesen wird.

V2=DATEI/ZEICH_LESEN,FPTR

Diese Zeile liest ein Zeichen aus der geöffneten Datei ein, die mit dem Dateizeiger `FPTR` verbunden ist. Das Zeichen wird in der Variablen `V2` gespeichert.

ZUWEISEN/V3=V3+V2

Diese Zeile verwendet die leere Variable `V3` und verkettet die Zeichenfolge von `V3` mit `V2` und weist den Wert wieder der `V3` zu. Auf diese Weise wird `V3` bei jeder weiteren Ausführung der `DO/UNTIL`-Schleife ein weiteres Zeichen hinzugefügt.

UNTIL/V2==" "

Diese Zeile beendet die `DO/UNTIL`-Schleife, sobald der `DATEI/ZEICH_LESEN`-Code ein Leerzeichen in der geöffneten Datei entdeckt.

DATEI/SCHLIESSEN,FPTR

Diese Zeile schließt die geöffnete Datendatei, wodurch andere Systemprozesse wieder auf die Datei zugreifen können. Der restliche Code wird beendet und zeigt das erste Wort aus der Datendatei in einem Bedienerkommentar an.

Lesen einer Zeile aus einer Datei

Die Menüoption **Einfügen | Datei I/O-Befehl | Lesebefehle | Zeile lesen** platziert einen Befehl in das Bearbeitungsfenster, durch den während der Programmausführung eine Zeile aus der angegebenen Datei gelesen wird. Dieser Befehl setzt die durch die Variablen-ID angegebene Variable auf 1 (wahr) oder 0 (falsch) und zeigt damit den Erfolg (wahr) bzw. das Fehlschlagen (falsch) des Aufrufs an. Der von diesem Befehl angeforderte Ausdruck kann verwendet werden, um die eingelesene Zeile zu begrenzen und um Variablen und Verweise automatisch durch die aus der Datei eingelesenen Daten aufzufüllen. Aus der Eingabedatei werden Daten bis zum nächsten Zeilenumbruch-Zeichen eingelesen.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



```
<Variablenname> =  
Datei/Zeile_lesen,<Dateizeigername>,<Ausdruck>
```

Beschreibungen für einige Komponenten dieser Befehle:

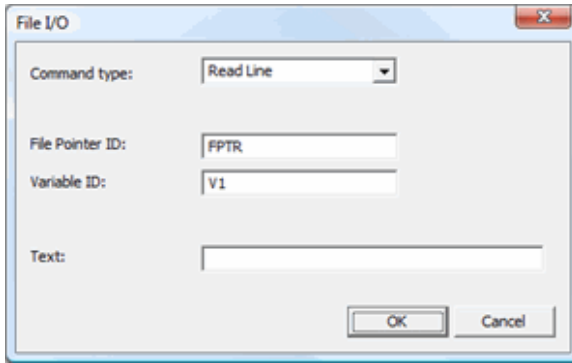
<Variablenname> - Der Name der Variablen, die als Ergebnis den Erfolg oder das Fehlschlagen des Befehls "Zeile lesen" anzeigt. Sie gibt "OK" oder "EOF" zurück.

<Dateizeigername> - Der beim Öffnen der Datei für den Dateizeiger angegebene Name.

<Ausdruck> - Die Zielvariable(n) der Eingabedaten. Eingabedaten können durch Text begrenzt werden und erlauben somit ein einfacheres Analysieren der eingehenden Datenzeilen. Variablen- und Elementverweise sollten in geschwungene Klammern gesetzt werden.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Zeile lesen".
3. Drücken Sie F9.



Beispielcode für "Zeile lesen"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Sehen Sie sich dieses Beispiel an, in dem aus einer Datendatei immer eine Zeile gelesen wird, bis der Befehl `DATEI/ZEILE_LESEN` eine leere Zeile vorfindet. Die Messroutine zeigt den gefundenen Textblock an und wird dann beendet.

```

.
.
.
V1      =DATEI/EXISTIERT,D:\HEXAGON\PCDMIS FILES\BASIC_SCRIPTS\TEST.TXT

WENN/V1<>0

    KOMMENTAR/BEDIEN,NEIN,VOLLBILD=NEIN,AUTO-FORTFAHREN=NEIN,OVC=NEIN,

    Kann aus der Datei lesen. Zum Fortfahren auf 'OK' klicken.

    ZUWEISEN/V3=""

FPTR    =DATEI/ÖFFNEN,D:\HEXAGON\PCDMIS FILES\BASIC_SCRIPTS\TEST.TXT,LESEN

    LOS/

V2      =DATEI/ZEILE_LESEN,FPTR,{LINE}

```

Lesen einer Zeile aus einer Datei

```
ZUWEISEN/V3=V3+ZEILE

KOMMENTAR/BEDIEN, NEIN, VOLLBILD=NEIN, AUTO-
FORTFAHREN=NEIN, OVC=NEIN,

    "Der aktuelle Wert der Variable V3 ist: "+V3

BIS/V2=="EOF"

DATEI/SCHLIESSEN, FPTR, LÖSCHEN

KOMMENTAR/BEDIEN, NEIN, VOLLBILD=NEIN, AUTO-FORTFAHREN=NEIN, OVC=NEIN,

    "Der Textblock lautet wie folgt: "+V3

END_IF/

ELSE/

    KOMMENTAR/BEDIEN, NEIN, VOLLBILD=NEIN, AUTO-FORTFAHREN=NEIN, OVC=NEIN,

    Konnte nicht aus der Datendatei gelesen werden. Routine wird jetzt
beendet.

    GOTO/END

END_ELSE/

END    =LABEL/

.

.

.
```

Erläuterung des Codes

Ein Großteil dieses Codes entspricht dem Code, der unter "Beispielcode für 'Zeichen lesen'" erläutert wird. Hier ist nur Code aufgelistet, der sich von diesem unterscheidet.

DO

Diese Zeile beginnt eine DO/UNTIL-Schleife. Sie begrenzt den `DATEI/ZEILE_LESEN`-Code, so dass die Zeilen fortlaufend einzeln eingelesen werden. Die Schleife wird beendet, wenn das Ende der Datei erreicht wird.

```
V2=DATEI/ZEILE_LESEN, FPTR, {ZEILE}
```

Diese Zeile liest den gesamten Text ein, bis sie auf einen Zeilenumbruch stößt. Im Gegensatz zu DATEI/ZEICH_LESEN, wo der Text in `V2` gespeichert wird, geht dieser Code anders vor.

- In diesem Fall gibt `V2` zwei Werte zurück: entweder "OK" oder "EOF". "OK" wird angezeigt, wenn weitere Dateien eingelesen werden können. "EOF", wenn das Ende der Datei erreicht ist.
- Der Code `{ZEILE}` ist eine vom Benutzer eingegebene Variable, die den Text speichert. Die Variable steht in geschwungenen Klammern, um PC-DMIS mitzuteilen, dass es sich um eine Variable und nicht um einen Teil des begrenzenden Textes handelt. Ohne geschwungene Klammern würde PC-DMIS in der Datei "ZEILE" nach einer Zeichenfolge suchen und nur den Text nach "ZEILE" und vor dem Zeilenumbruch zurückgeben.

ZUWEISEN/V3=V3+ZEILE

Diese Zeile verwendet die leere Variable `V3` und verkettet die Zeichenfolge von `V3` mit `ZEILE`. Anschließend weist sie den Wert wieder `V3` zu. Auf diese Weise wird `V3` bei jeder weiteren Ausführung der DO/UNTIL-Schleife eine weitere Zeile hinzugefügt.

UNTIL/V2=="EOF"

Diese Zeile testet die Bedingung für die DO/UNTIL-Schleife. Sobald der DATEI/ZEILE_LESEN-Code das Ende der Datei erreicht, wird die Schleife beendet. Wenn die Messroutine die Schleife beendet hat, wird die Ausführung des restlichen Codes beendet und der gesamte Codeblock in einem Bedienerkommentar angezeigt.



`Ergebnis=Datei/Zeile_Lesen,F1,"Werkstück_ID :"+{V1}` - Dies hat zur Folge, dass sämtlicher Text, der in der eingelesenen Zeile nach "Werkstück-ID:" steht, V1 zugewiesen wird. Die Zeile wird in diesem Fall aus der Datei eingelesen, die mit F1 als Dateizeigernamen geöffnet wurde. Das Ergebnis des Einlesens (Erfolg oder Fehlschlag) wird dann in der Variablen "Ergebnis" gespeichert.

```
Datei/Zeile_lesen,F1,"Lage:"+{VARX}+", "+{VARY}+", "+{VARZ}+",  
"+{VARI}+", "+{VARJ}+", "+{VARK}
```

```
ZUWEISEN/KREIS1.XYZ=MPOINT (VARX,VARY,VARZ)
```

```
ZUWEISEN/KREIS1.IJK=MPOINT (VARI,VARJ,VARK)
```

Die drei Befehlszeilen weiter oben werden in von Kommata getrenntem Text nach der Zeichenfolge "Lage:" eingelesen und die XYZ-Werte und die IJK-Werte von KREIS1 werden abgespeichert.

`Datei/Zeile_lesen,F1,"Wert Nr. "+Schleifenvar.+" :"+{Var2}` - Diese Befehlszeile bewirkt, dass Var2 mit dem nach dem Doppelpunkt erscheinenden Text ausgefüllt wird. Die in diesem Beispiel angegebene Variable "Schleifenvar." steht nicht in geschweiften Klammern und trägt folglich zum begrenzenden Text bei.

Beispielcode mit Zahlen, die voranstehende Nullen enthalten

Wenn die Datei, die eingelesen wird, Zahlenzeilen enthält, werden Sie beobachten, dass PC-DMIS voranstehende Nullen ignoriert. Enthält die Zeile beispielsweise den Wert 005450, dann würde dieser Wert streng als Zahl eingelesen und der Wert 5450 zurückgegeben, wobei die beiden voranstehenden Nullen ignoriert würden. Diese Vorgehensweise ist jedoch nicht immer zufriedenstellend.

Wenn Sie zum Beispiel über eine Textdatei verfügen, die von einer externen Software zum Lesen von Barcode erstellt wurde und die die beiden nachfolgenden Datenzeilen enthält:

290291143;582750;0010

291143;5827;0010

, dann können Sie einen so einfachen Code verwenden, um die Zahlenwerte zwischen den Semikolons zu erhalten:



```

ZUWEISEN/ERSTER_WERT=0
ZUWEISEN/ZWEITER_WERT=0
ZUWEISEN/DRITTER_WERT=0
ZUWEISEN/ZEILENANZ=1
FPTR=DATEI/ÖFFNEN,D:\TEMP\CODES.TXT,LESEN
DO/
  INLINE
  =DATEI/ZEILE_LESEN,FPTR,{ERSTER_WERT}+";"+{ZWEITER_WERT}+";"+{DRITTER_WERT}
  KOMMENTAR/BEDIENER,NEIN,"ZEILENNUMMER: "+ZEILENANZ
    , "Erster Wert: "+ERSTER_WERT
    , "Zweiter Wert: "+ZWEITER_WERT
    , "Dritter Wert: "+DRITTER_WERT
  UNTIL/INLINE=="EOF"
DATEI/SCHLIESSEN,FPTR,BEIBEHALTEN

```

Obwohl hierdurch die Textzeilen erfolgreich analysiert und die Zahlenwerte zurückgegeben werden, wird außerdem jede voranstehende Null aller zurückgegebenen Werte entfernt. Die Variable DRITTER_WERT würde also anstelle von 0010 den Wert 10 enthalten.

Um die voranstehenden Nullwerte beizubehalten, muss die gesamte Zeile wie eine Zeichenfolge behandelt werden, anstatt die Zeichenfolgenfunktionen INDEX, LINKS und MITTE zur Lokalisierung der Semikolon-Positionen in einer Textzeile einzusetzen und so die Zahlenwerte zu erhalten:



```
FPTR=DATEI/ÖFFNEN,D:\TEMP\CODES.TXT,LESEN
ZUWEISEN/ZEILENANZ=1
DO/
ZEILENSTATUS =DATEI/ZEILE_LESEN,FPTR,{ZEILEN_ZF}
ZUWEISEN/ZEILEN_ZF=STR(ZEILEN_ZF)
ZUWEISEN/ERSTER_INDEX=INDEX(ZEILEN_ZF,";")
ZUWEISEN/ERSTER_WERT=STR(LEFT(ZEILEN_ZF,ERSTER_INDEX-1))
ZUWEISEN/RESTL_ZF=STR(MID(ZEILEN_ZF,(ERSTER_INDEX)))
ZUWEISEN/ZWEITER_INDEX=INDEX(RESTL_ZF,";")
ZUWEISEN/ZWEITER_WERT=STR(LEFT(RESTL_ZF,ZWEITER_INDEX-1))
ZUWEISEN/DRITTER_WERT=STR(MID(RESTL_ZF,ZWEITER_INDEX))
KOMMENTAR/BEDIENER,NEIN,"ZEILENNUMMER:
"+ZEILENANZ
    ,"Erster Wert: "+ERSTER_WERT
    ,"Zweiter Wert: "+ZWEITER_WERT
    ,"Dritter Wert: "+DRITTER_WERT
ZUWEISEN/ZEILENANZ=ZEILENANZ+1
UNTIL/ZEILENSTATUS=="EOF"
DATEI/SCHLIESSEN,FPTR,BEIBEHALTEN
```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der weiter oben erläutert wird. Hier wird nur die Erläuterung des Codes, die sich auf die behandelten Zeichenfolgenfunktionen bezieht, aufgelistet.

ZUWEISEN/ERSTER_INDEX=INDEX(ZEILEN_ZF,";")

Diese Zeile lokalisiert die Position des ersten Semikolons in der Zeile und weist es der Variablen `ERSTER_INDEX` zu.

ZUWEISEN/ERSTER_WERT=STR(LEFT(ZEILEN_ZF,ERSTER_INDEX-1))

Diese Zeile weist der Variablen `FIRST_VALUE` die Zeichenfolge zu, die bis zum ersten Semikolon in der Variablen `LINESTR` reicht, dieses jedoch nicht mit einschließt. `ZEILEN_ZF` enthält die gesamte Textzeile.

ZUWEISEN/RESTL_ZF=STR(MID(ZEILEN_ZF,(ERSTER_INDEX)))

Diese Zeile weist der Variablen `RESTL_ZF` (steht für "restliche Zeichenfolge") die Zeichenfolge der übriggebliebenen bzw. restlichen Zeichen zu, beginnend an der Position `ERSTER_INDEX` (der Position des ersten Semikolons) bis zum Ende der Zeile.

```
ZUWEISEN/ZWEITER_INDEX=INDEX(RESTL_ZF,";")
```

Hiermit wird innerhalb der Variablen `RESTL_ZF` nach einem weiteren Semikolon (das zweite Semikolon in der Zeile) gesucht und die Position der Variablen `ZWEITER_INDEX` zugewiesen.

```
ZUWEISEN/ZWEITER_WERT=STR(LEFT(RESTL_ZF,ZWEITER_INDEX-1))
```

Diese Zeile weist der Variablen `ZWEITER_WERT` die Zeichenfolge bis zum ersten Semikolon - das Semikolon wird jedoch nicht mit einbezogen - in der Variablen `RESTL_ZF` zu (das zweite Semikolon in der gesamten Zeile).

```
ZUWEISEN/DRITTER_WERT=STR(MID(RESTL_ZF,ZWEITER_INDEX))
```

Diese Zeile weist der Variablen `DRITTER_WERT` die Zeichenfolge ab der Position `ZWEITER_INDEX` bis zum Zeilenende zu.

Lesen eines Textblocks aus einer Datei

Die Menüoption **Einfügen | Datei I/O-Befehl | Lesebefehle | Block lesen** platziert einen Befehl in das Bearbeitungsfenster, durch den während der Programmausführung ein Zeichenblock aus einer geöffneten Datei gelesen wird. Die Anzahl der eingelesenen Zeichen wird durch den Parameter "Größe" angegeben.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



```
<Variablenname>=Datei/Block_lesen,<Dateizeigername>,<Größe>
```

Beschreibungen für einige Komponenten dieser Befehle:

<Variablenname> - Eine Variablen-ID der Variablen, die den Wert erhält, der den Erfolg oder Fehlschlag des Block-Einlesevorgangs anzeigt.

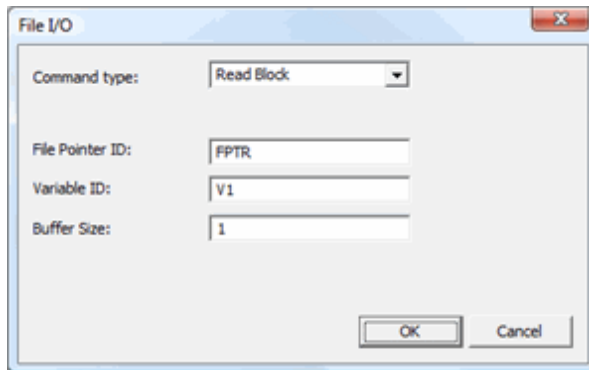
<Dateizeigername> - Der beim Öffnen der Datei für den Dateizeiger angegebene Name.

<Größe> - Die Anzahl der zu lesenden Zeichen.

Lesen eines Textblocks aus einer Datei

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Block lesen".
3. Drücken Sie F9.



Beispielcode für "Block lesen"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Angenommen, Sie haben mehrere externe Datendateien, die verschiedene Werkstückdaten enthalten. Darin geben jeweils die ersten Zeichen an, wofür die Datei bestimmt ist. In diesem Fall können Sie den Befehl `Datei/Block_Lesen` verwenden, um nur die ersten Zeichen jeder Datei zu lesen, bevor Sie sich dazu entschließen, alle Zeilen einzulesen und zu verarbeiten. Sehen Sie sich den folgenden Code an:

```

C3=KOMMENTAR/EINGABE,Geben Sie den Namen des
,Dateicodes ein, nach dem gesucht werden soll.
ZUWEISEN/BLOCKSIZE (BLOCKGRÖSSE) = LEN(C3.INPUT)
ZUWEISEN/FILECODE (DATEICODE) = C3.INPUT
DO/
  C1=KOMMENTAR/EINGABE,Geben Sie den vollständigen
  Pfad,
  ,den Dateinamen und die Erweiterung
  ,für die zu verarbeitende Datei ein.
  ,Drücken Sie [Q], um das Programm zu beenden.
  IF/C1.INPUT== "Q" ODER C1.INPUT=="q"
    KOMMENTAT/BEDIENER,Sie haben sich für die
    Beendigung des Programms entschieden.
    Messroutine wird jetzt beendet.
    GEHZU/ENDE
  END_IF/
  V1=DATEI/VORHANDEN,C1.INPUT
  IF/V1<>0
    KOMMENTAR/BEDIENER,"Datendatei [" + C1.INPUT
    + "]" ist vorhanden. Zum Fortfahren auf 'OK'
    klicken."
    FPTR=DATEI/ÖFFNEN,C1.INPUT,LESEN
    V2=DATEI/BLOCK_LESEN,FPTR,BLOCKSIZE
    (BLOCKGRÖSSE)
    DATEI/SCHLIESSEN,FPTR
    IF/V2<>FILECODE (DATEICODE)
      KOMMENTAR/BEDIENER,"Der Dateicode von ["
      + V2 + "]" stimmt nicht mit"
      ,"dem FILECODE (DATEICODE) von [" +
      FILECODE (DATEICODE) + "]" überein."
    END_IF/
    KOMMENTAR/BEDIENER,"Datei [" + C1.INPUT + "]"
    stimmt damit überein."
    ,"Der Dateicode von [" + V2 + "]" stimmt mit"
    ,"dem FILECODE (DATEICODE) von [" + FILECODE
    (DATEICODE) + "]" überein."
    KOMMENTAR/BEDIENER,Die Routine verarbeitet
    dann die Datei.
  END_IF/
  ELSE/
    KOMMENTAR/BEDIENER,"Datendatei
    [ "+C1.INPUT+" ] ist nicht vorhanden.
    Versuchen Sie es erneut mit einer
    vorhandenen Datei."
    GEHZU/ENDE
  END_ELSE/

```



```
UNTIL/V2==FILECODE (DATEICODE)  
ENDE=MARKE/  
ROUTINE/ENDE
```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Zeichen lesen'" oder unter "Beispielcode für 'Zeile lesen'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

```
ZUWEISEN/BLOCKSIZE (BLOCKGRÖSSE) = LEN(C3.INPUT)
```

Diese Zeile erstellt eine benutzerdefinierte Variable namens `BLOCKSIZE` (`BLOCKGRÖSSE`), die eine Ganzzahl enthält, die der Anzahl der gefundenen Zeichen in `C3.INPUT` entspricht. Diese Variable wird als Größe für den Zeichenblock verwendet, der eingelesen werden soll.

```
ZUWEISEN/FILECODE (DATEICODE) = C3.INPUT
```

Diese Zeile erstellt die Variable `FILECODE` (`DATEICODE`) und gibt ihr den Wert aus `C3.INPUT`.

```
C1=KOMMENTAR/EINGABE
```

Dieser Kommentar speichert den vom Benutzer eingegebenen vollständigen Pfad in der Variable `C1.INPUT`.

```
V1=DATEI/VORHANDEN,C1.INPUT
```

Diese Zeile prüft das Vorhandensein der Datei, die im Kommentar C1 definiert wurde.

```
DO/
```

Diese Zeile beginnt eine DO/UNTIL-Schleife. Sie begrenzt den Codeblock, der dem Benutzer die Eingabe einer Datei zum Lesen ermöglicht. Die Schleife wird so lange fortgesetzt, bis der Text, welcher der Variablen `FILECODE` (`DATEICODE`) zugewiesen wurde, dem ausgelesenen Text entspricht.

```
V2=DATEI/BLOCK_LESEN,FPTR,BLOCKSIZE (BLOCKGRÖSSE)
```

Diese Zeile liest jene Anzahl von Zeichen, die der Ganzzahl in der Variablen `BLOCKSIZE` (`BLOCKGRÖSSE`) entspricht. Der Text wird dann in der Variablen `V2` gespeichert.

IF/V2FILECODE

Diese Zeile beginnt einen IF/END IF-Codeblock, der testet, ob der Text in der Variablen **V2** dem in der Variablen **FILECODE** (DATEICODE) gespeicherten Text entspricht. Stimmt er damit überein, wird die Ausführung der Messroutine fortgesetzt. Andernfalls wird dem Benutzer in einer Meldung mitgeteilt, dass die beiden Codes nicht übereinstimmen.

UNTIL/V2==FILECODE (DATEICODE)

Diese Zeile prüft die Bedingung der DO/UNTIL-Schleife, um festzustellen, ob der Text in der Variable **V2** mit jenem in der Variable **FILECODE** (DATEICODE) übereinstimmt. Wird die Anweisung als falsch ausgewertet, wird die **DO**-Schleife erneut ausgeführt, so dass der Benutzer einen anderen Dateinamen auswählen kann. Wird die Anweisung als wahr ausgewertet, wird die Schleife beendet, und die Messroutine meldet, dass die Codes übereinstimmen. PC-DMIS kann dann mit dem Lesen aller Datenzeilen der angegebenen Datei fortfahren.

Lesen von Text bis zu einem Begrenzer

Die Menüoption **Einfügen | Datei I/O-Befehl | Lesebefehle | Lesen bis zu** platziert einen Befehl in das Bearbeitungsfenster, durch den während der Programmausführung der gesamte Text 'bis zu' einem der gegebenen Begrenzer aus der angegebenen Datei gelesen wird. Der gesamte durch diesen Befehl gelesene Text wird in der angegebenen Zielvariable platziert. Der Befehl stellt den Textlesevorgang ein, wenn PC-DMIS folgendes antrifft:

- Definierte Begrenzer
- Zeilenumbrüche
- Zeilenvorschubzeichen

Bei Erreichen des Dateiendes wird die Zielvariable auf "EOF" (End of File, Dateiende) gesetzt.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



```
<Variablenname> =  
DATEI/LESEN_BIS,<Dateizeigername>,<Begrenzer>
```

Beschreibungen für einige Komponenten dieser Befehle:

<Variablenname> - Dies ist der Name der Zielvariablen.

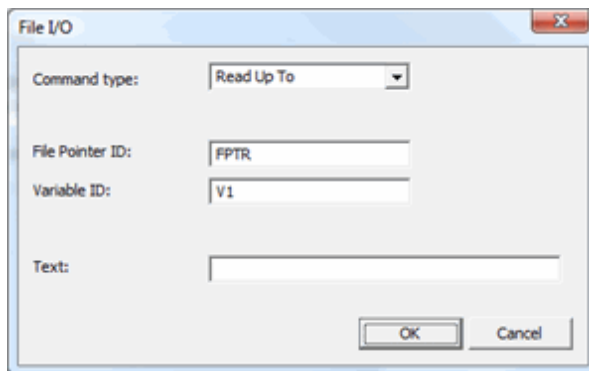
Lesen von Text bis zu einem Begrenzer

<Dateizeigername> - Der beim Öffnen der Datei für den Dateizeiger angegebene Name.

<Begrenzer> - Dies ist eine Zeichenfolge mit Null oder mehr Begrenzungszeichen.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster, indem Sie **Bearbeitungsfenster** wählen.
2. Setzen Sie den Cursor auf den Befehl **DATEI / LESEN**.
3. Drücken Sie F9. Das Dialogfeld **Datei I/O** wird geöffnet.



In dem nun eingeblendeten Dialogfeld:

1. Geben Sie in das Feld **Variablen-ID** den Namen der Variablen ein, welche die eingelesenen Informationen aufnehmen wird.
2. Geben Sie in das Feld **Dateizeiger-ID** den Namen des Dateizeigers ein.
3. Geben Sie in das Feld **Text** den Begrenzer ein (der von Ihnen ausgewählte Begrenzer muss in Anführungszeichen gesetzt werden).
4. Klicken Sie auf **OK**.

Beispielcode für "Lesen bis zu"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Gesetzt den Fall, im Verzeichnis D:\Temp befindet sich eine Textdatei namens "muster.txt", deren erste Zeile folgende Informationen enthält:



```
KREIS1:2.54:KREIS
```

So verwenden Sie den Befehl "Lesen bis zu" bei dieser Datei:

1. Fügen Sie in das Bearbeitungsfenster einen `DATEI/ÖFFNEN`-Befehl ein.
2. Wählen Sie einen beliebigen Dateizeigernamen aus, um den Datei/Öffnen-Befehl zu benennen. In diesem Beispiel wird als Dateizeigernamen "Muster" verwendet.

Der Datei/Öffnen-Befehl sollte in etwa folgendermaßen aussehen:

```
MUSTER      =DATEI/ÖFFNEN,C:\TEMP\MUSTER.TXT,LESEN
```

Sie können nun mit den "Lesen bis zu"-Befehlen von PC-DMIS Variablen definieren, durch die unterschiedliche Datensegmente aufgerufen werden. Im folgenden Beispiel suchen folgende Variablen nach einem Doppelpunkt ":" als Begrenzer (ohne die Anführungszeichen).

```
V_LABEL      =DATEI/LESEN_BIS,MUSTER, :
V_WERT       =DATEI/LESEN_BIS,MUSTER, :
V_TYP        =DATEI/LESEN_BIS,MUSTER, :
```

Wenn PC-DMIS diese Zeilen ausführt, setzt es diese Variablen auf diese Werte:

- V_LABEL = KREIS1
- V_WERT = 2.54
- V_TYP = KREIS

Über einen Bedienerkommentar wie im folgenden Beispiel können Sie veranlassen, dass diese Ergebnisse während der Ausführung auf dem Bildschirm angezeigt werden:

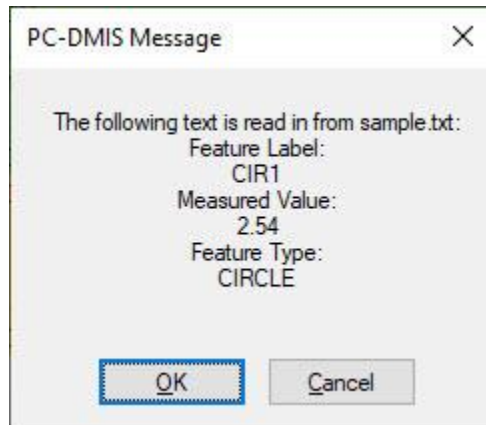
```
KOMMENTAR/OPER,NEIN,VOLLBILD=NEIN,AUTO-WEITER=NEIN,OVC=NEIN,
```

```
Der folgende Text wird aus Muster.txt eingelesen:
```

```
Elementbezeichnung:
```

Schreiben eines Zeichens in eine Datei

```
V_LABEL  
  
Messwert:  
  
V_WERT  
  
Elementtyp:  
  
V_TYP
```



Schreiben eines Zeichens in eine Datei

Die Menüoption **Einfügen | Datei I/O-Befehle | Schreibbefehle | Zeichen schreiben** fügt einen Befehl in das Bearbeitungsfenster ein, bei dessen Ausführung ein Einzelzeichen in eine Datenträgerdatei ausgegeben wird.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



`Datei/Zeichen_schreiben,<Dateizeigername>,<Ausdruck>`

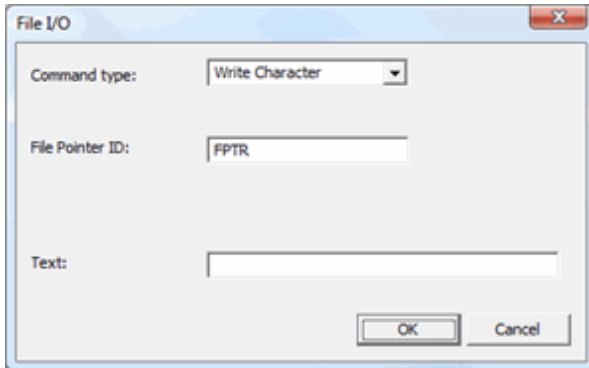
Beschreibungen für einige Komponenten dieser Befehle:

<Dateizeigername> - Dies ist der Name des Dateizeigers, der beim Öffnen der Datei angegeben wurde.

<Ausdruck> - Das in die Datei zu schreibende Zeichen. Wenn die Auswertung des Ausdrucks mehr als ein Zeichen ergibt, wird lediglich das erste Zeichen geschrieben.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Zeichen schreiben".
3. Drücken Sie F9.



Beispielcode für "Zeichen schreiben"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Sehen Sie sich den folgenden Code an, der eine vom Benutzer vorgegebene Zeichenfolge jeweils pro Zeichen in eine Datendatei schreibt.

Schreiben eines Zeichens in eine Datei



```
C1=KOMMENTAR/EINGABE,Geben Sie den Namen der Datei
ein, in
,die geschrieben werden soll (einschl. des
vollständigen Pfads).
FPTR=DATEI/ÖFFNEN,C1.INPUT,SCHREIBEN
C2=KOMMENTAR/EINGABE,Geben Sie einen Text ein, der
an die Datei gesendet werden soll.
,Dadurch wird die Zeichenfolge zeichenweise
,an die Datei gesendet.
ASSIGN/COUNT=0
ZUWEISEN/LÄNGE=LEN(C2.INPUT)
DO/
    ZUWEISEN/WRITETHIS=MID(C2.INPUT,ZÄHLER,1)
    DATEI/ZEICH_SCHREIBEN,FPTR,WRITETHIS
    ASSIGN/COUNT=COUNT + 1
UNTIL/ZÄHLER==LÄNGE
```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Zeichen lesen'" oder unter "Beispielcode für 'Zeile lesen'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

FPTR=DATEI/ÖFFNEN,C1.INPUT,SCHREIBEN

Diese Zeile öffnet die im Kommentar C1 angegebene Datei zum Anhängen und weist sie dem Dateizeiger **FPTR** zu. Alle Daten in dieser Datei werden überschrieben, wenn der Dateizeiger auf den Anfang der Datei zeigt.

ASSIGN/COUNT=0

Diese Zeile weist der benutzerdefinierten Variable **COUNT** (ANZAHL) den Wert 0 zu. Sie wird für die Schleifen verwendet, um die Zeichenfolge jeweils pro Zeichen auszugeben.

ZUWEISEN/LÄNGE=LEN(C2.INPUT)

Diese Zeile verwendet die Funktion **LEN()**, um die Länge der Zeichenfolge zurückzugeben. Diese Funktion verwendet einen Parameter, die Zeichenfolge. Sie zählt die Anzahl der Zeichen in der Zeichenfolge (einschl. Leerzeichen) und gibt diese Anzahl in Form eines Ganzzahlwerts aus. In diesem Fall enthält die benutzerdefinierte Variable **LÄNGE** diesen Wert.

DO/

Diese Zeile beginnt eine DO/UNTIL-Schleife. Der Code zwischen der DO- und der UNTIL-Anweisung wird ausgeführt, bis die Schleifenbedingung als wahr ausgewertet wird.

ZUWEISEN/WRITETHIS=MID (C2.INPUT, ZÄHLER, 1)

Diese Zeile erstellt eine benutzerdefinierte Variable namens WRITETHIS und verwendet die Funktion MID (), um ein Zeichen der Zeichenfolge C2.INPUT an WRITETHIS auszugeben.

MID () verwendet drei Parameter.

- Parameter 1: ist die Zeichenfolge, aus der Werte abgerufen werden. In diesem Fall wird C2.INPUT verwendet.
- Parameter 2: ist die Position in der Zeichenfolge, an der das Zeichen entnommen wird. Das erste Zeichen in einer Zeichenfolge befindet sich an Position 0, das zweite an Position 1, das dritte an Position 2 usw. In diesem Fall wird die Variable ZÄHLER verwendet.
- Parameter 3: ist die Anzahl der Zeichen ab der Position des zweiten Parameters, die übernommen werden sollen. In diesem Fall wird der Wert 1 verwendet. (Da das Beispiel immer nur ein Zeichen schreibt, gibt es keinen Grund, mehr Zeichen zu übernehmen.)

DATEI/ZEICH_SCHREIBEN, FPTR, WRITETHIS

Diese Zeile schreibt das in der Variable WRITETHIS gespeicherte Zeichen in die Datei, die durch den Dateizeiger FPTR angegeben wird.

ASSIGN/COUNT=COUNT+1

Diese Zeile nimmt den aktuellen Wert aus ZÄHLER, erhöht ihn um 1 und platziert dann den neuen Wert wieder in ZÄHLER.

UNTIL/ZÄHLER==LÄNGE

Diese Zeile testet die Bedingung für die DO/UNTIL-Schleife. In diesem Fall erhöht die Schleife die Variable ZÄHLER so lange, bis diese denselben Wert aufweist wie die Variable LÄNGE. Dann werden die Schleife und die Messroutine beendet.

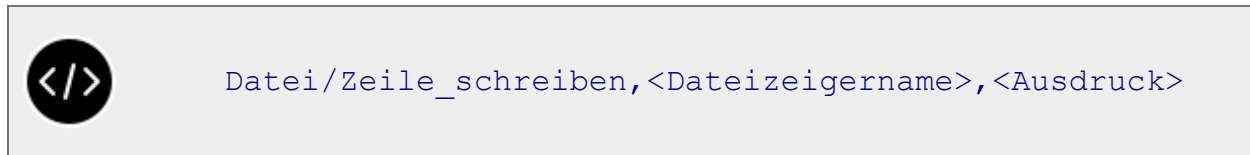
Schreiben einer Zeile in eine Datei

Die Menüoption **Einfügen | Datei I/O-Befehle | Schreibbefehle | Zeile schreiben** fügt einen Befehl in das Bearbeitungsfenster ein, bei dessen Ausführung eine Zeile in eine

Schreiben einer Zeile in eine Datei

Datenträgerdatei ausgegeben wird. Zur Ausgabe von Variablen und Daten der Messroutine in eine Datei sollten Sie die Ausdruckssyntax verwenden. An den in die Datei geschriebenen Text wird automatisch ein Zeilenwechsel angefügt.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



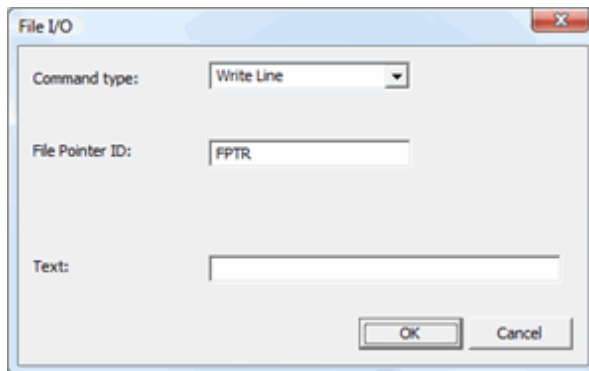
Beschreibungen für einige Komponenten dieser Befehle:

<Dateizeigername> - Der Name des Dateiverweises, der beim Öffnen der Datei angegeben wird.

<Ausdruck> - Der in die Datei zu schreibende Text. In diesem Feld können Ausdrücke verwendet werden.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Zeile schreiben".
3. Drücken Sie F9.



Beispielcode für "Zeile schreiben"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Angenommen, Sie möchten einige gemessene XYZ-Werte in eine Datendatei exportieren. Der folgende Code ermöglicht Ihnen die Eingabe eines Elementetiketts und einer Datendatei und das anschließende Senden der X-,Y- und Z-Daten für dieses Element an eine Datendatei.



```
C1=KOMMENTAR/EINGABE,Geben Sie die Beschriftung des
,zu verwendenden Elements ein.
C2=KOMMENTAR/EINGABE,Geben Sie den Namen der Datei
ein, in
, die geschrieben werden soll (einschl. des
vollständigen Pfads).
FPTR=DATEI/ÖFFNEN,C2.INPUT,ANHÄNGEN
ZUWEISEN/ELEMENTNAME=C1.INPUT
ZUWEISEN/ALLVALS
(ALLE_WERTE)=ELEMENTNAME.X+", "+ELEMENTNAME.Y+", "+ELEM
ENTNAME.Z
KOMMENTAR/BEDIENER,"Der zu schreibende Text lautet:
"+ALLVALS (ALLE_WERTE)
DATEI/ZEILE_SCHREIB,FPTR,ALLVALS (ALLE_WERTE)
DATEI/SCHLIESSEN,FPTR
```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Zeichen lesen'" oder unter "Beispielcode für 'Zeile lesen'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

FPTR=DATEI/ÖFFNEN,C2.INPUT,ANHÄNGEN

Diese Zeile öffnet die im Kommentar C2 angegebene Datei zum Anhängen und weist sie dem Dateizeiger **FPTR** zu. Wenn Sie **ANHÄNGEN** in **SCHREIBEN** ändern, wird der vorhandene Inhalt der Datendatei überschrieben.

ZUWEISEN/ELEMENTNAME=C1.INPUT

Schreiben eines Textblocks in eine Datei

Diese Zeile weist der benutzerdefinierten Variable `ELEMENTNAME` die Zeichenfolge des Elementetiketts aus `C1.INPUT` zu.

`ZUWEISEN/ALLVALS`

```
(ALLE_WERTE)=ELEMENTNAME.X+" , "+ELEMENTNAME.Y+" , "+ELEMENTNAME.Z
```

Diese Zeile gibt der benutzerdefinierten Variable `ALLVALS` (`ALLE_WERTE`) den Wert von `ELEMENTNAME.X`, `ELEMENTNAME.Y`, `ELEMENTNAME.Z`. Das heißt, sie enthält nun die X-, Y- und Z-Werte des Elementetiketts, das im Eingabekommentar C1 eingegeben wurde.

`DATEI/ZEILE_SCHREIB,FPTR,ALLVALS (ALLE_WERTE)`

Diese Zeile schreibt die Werte aus `ALLVALS` in die vom Dateizeiger `FPTR` angegebene Datei.

Schreiben eines Textblocks in eine Datei

Die Menüoption **Einfügen | Datei I/O-Befehle | Schreibbefehle | Block schreiben** fügt einen Befehl in das Bearbeitungsfenster ein, bei dessen Ausführung ein Textblock in eine Datenträgerdatei ausgegeben wird. Zur Ausgabe von Variablen und Daten der Messroutine in eine Datei sollten Sie die Ausdruckssyntax verwenden. Im Gegensatz zum Befehl „Zeile schreiben“ wird beim Befehl „Block schreiben“ *kein* Zeilenwechsel am Ende angehängt.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



```
Datei/Block_schreib,<Dateizeigername>,<Ausdruck>
```

Beschreibungen für einige Komponenten dieser Befehle:

<Dateizeigername> - Der Name des Dateiverweises, der beim Öffnen der Datei angegeben wird.

<Ausdruck> - Der in die Datei zu schreibende Text. In diesem Feld können Ausdrücke verwendet werden.



Im Gegensatz zum Befehl „Zeile schreiben“ wird beim Befehl „Block schreiben“ *kein* Zeilenwechsel am Ende angehängt. Wenn Sie jedoch Text in eine neue Zeile in Ihrem Textblock platzieren möchten, können Sie einen Zeilenwechsel und Zeilenvorschub manuell einfügen, indem Sie den CHR(10)-Code außerhalb der in Anführungszeichen stehenden Zeichenfolge verwenden, wie im nachfolgenden Beispiel veranschaulicht:

```
DATEI/BLOCK_SCHREIB,FPTR, "CHR(10) fügt Text ein... " + CHR(10)
+ " ...in eine neue Zeile."
```

Dies würde in der Ausgabedatei zu folgendem Ergebnis führen:

```
CHR(10) Fügt Text ein...
```

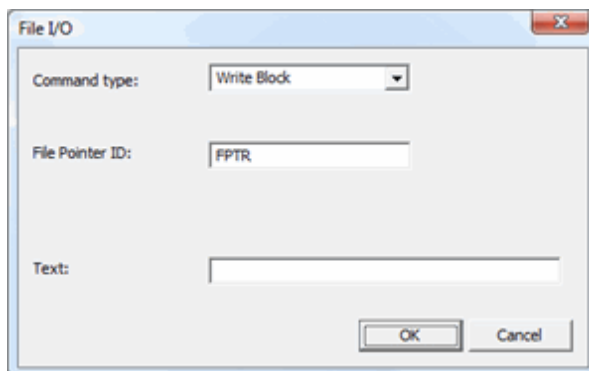


```
...in eine neue Zeile.
```

Beachten Sie, dass der tatsächliche Text aus CHR(10) an die Datei gesendet wird, wenn CHR(10) in Anführungszeichen steht.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Block schreiben".
3. Drücken Sie F9.



Beispielcode für "Block schreiben"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Der folgende Code schreibt jeden beliebigen Text, den der Benutzer eingibt, in einen Eingabekommentar und hängt dann einen Doppelpunkt als Begrenzer an.



```
C1=KOMMENTAR/EINGABE,Geben Sie eine Zeichenfolge
ein. PC-DMIS hängt dann einen Doppelpunkt (als
Begrenzer) an die Zeichenfolge an, und schreibt sie
in eine Datei Ihrer Wahl.
C2=KOMMENTAR/EINGABE,Geben Sie den Namen der Datei
ein, in
, die geschrieben werden soll (einschl. des
vollständigen Pfads).
FPTR=DATEI/ÖFFNEN,C2.INPUT,ANHÄNGEN
ZUWEISEN/WRITETHIS=C1.INPUT+":"
KOMMENTAR/BEDIENER,"Der zu schreibende Text lautet:
"+WRITETHIS
DATEI/ZEILE_SCHREIB,FPTR,WRITETHIS
DATEI/SCHLIESSEN,FPTR
```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Zeichen lesen'" oder unter "Beispielcode für 'Zeile lesen'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

FPTR=DATEI/ÖFFNEN,C2.INPUT,ANHÄNGEN

Diese Zeile öffnet die im Kommentar C2 angegebene Datei zum Anhängen und weist sie dem Dateizeiger **FPTR** zu.

ZUWEISEN/WRITETHIS=C1.INPUT+":"

Diese Zeile hängt einen Doppelpunkt an den Text in **C1.EINGABE** an und weist die neue Zeichenfolge der benutzerdefinierten Variable **WRITETHIS** zu.

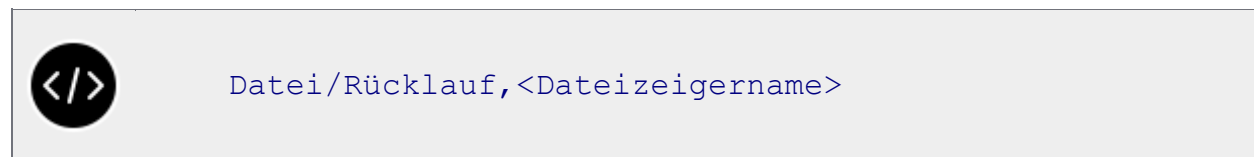
DATEI/ZEILE_SCHREIB,FPTR,WRITETHIS

Diese Zeile schreibt die Werte aus `WRITETHIS` in die vom Dateizeiger `FTPR` angegebene Datei. Später können Sie Text aus dieser Datei einlesen und dabei den Doppelpunkt als Begrenzer verwenden.

Positionieren eines Dateizeigers am Anfang einer Datei

Die Menüoption **Einfügen | Datei I/O-Befehl | Positionierungsbefehle | Zum Start zurück** fügt einen Befehl im Bearbeitungsfenster ein, der den Dateizeiger am Anfang des Dateistroms positioniert.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:

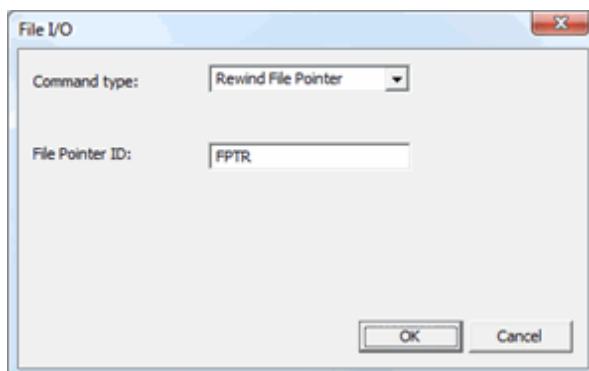


Beschreibungen für einige Komponenten dieser Befehle:

<Dateizeigername> - Der Name des Dateizeigers zur Neupositionierung am Anfang der Datei.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Zum Start zurück".
3. Drücken Sie F9.



Beispielcode für "Zum Start zurück"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Sehen Sie sich dieses Beispiel an, das Daten aus einer externen Datei zeilenweise einliest. Nach jeder Zeile haben Sie die Möglichkeit, von vorne zu beginnen und den Anfang der Datei einzulesen. Das Beispiel beschreibt die Verwendung des Befehls `DATEI/RÜCKLAUF`.



```
C1=KOMMENTAR/EINGABE,Geben Sie eine Datei ein, aus
der gelesen werden soll.
,(Vollständigen Pfad angeben.)
V1=DATEI/VORHANDEN,C1.INPUT
IF/V1<>0
DO/
    FPTR=DATEI/ÖFFNEN,C1.INPUT,LESEN
    C2=KOMMENTAR/JANEIN,Möchten Sie mit dem
    Einlesen am Dateianfang beginnen?
    IF/C2.INPUT == "JA"
        DATEI/RÜCKLAUF,FPTR
    END_IF/
    V2=DATEI/ZEILE_LESEN,FPTR,{ZEILE}
    KOMMENTAR/BEDIENER,"Die aktuelle Zeile
    lautet: " + ZEILE
    UNTIL/V2=="EOF"
END_IF/
DATEI/SCHLIESSEN,FPTR
KOMMENTAR/BEDIENER,Die Routine wird beendet.
```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Zeichen lesen'" oder unter "Beispielcode für 'Zeile lesen'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

`C2=KOMMENTAR/JANEIN`

Diese Zeile fragt, ob Sie mit dem Lesen am Dateianfang beginnen möchten. Die JA/NEIN-Antwort wird in der Variablen `C2.INPUT` gespeichert.

```
IF/C2.INPUT == "JA"
```

Diese Zeile beginnt einen IF/END IF-Block. Sie testet die Bedingung der Variablen `C2.INPUT`, die den Wert JA enthält. Ist die Bedingung wahr, führt PC-DMIS die Zeilen nach der `IF`-Anweisung aus. Ist die Bedingung falsch, führt PC-DMIS die Zeilen nach der `END_IF`-Anweisung aus.

```
DATEI/RÜCKLAUF,FPTR
```

Diese Zeile setzt den Dateizeiger wieder an den Anfang der Datei.

```
END_IF/
```

Diese Zeile beendet den IF/END IF-Codeblock.

Speichern der aktuellen Position eines Dateizeigers

Die Menüoption **Einfügen | Datei I/O-Befehl | Positionierungsbefehle | Dateiposition speichern** fügt einen Befehl im Bearbeitungsfenster ein, durch den die aktuelle Position eines Dateizeigers innerhalb des Dateistroms gespeichert wird. Die Speicherposition kann später über den Befehl "Dateiposition abrufen" erneut abgerufen werden.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



```
Datei/Pos_speichern,<Dateizeigername>
```

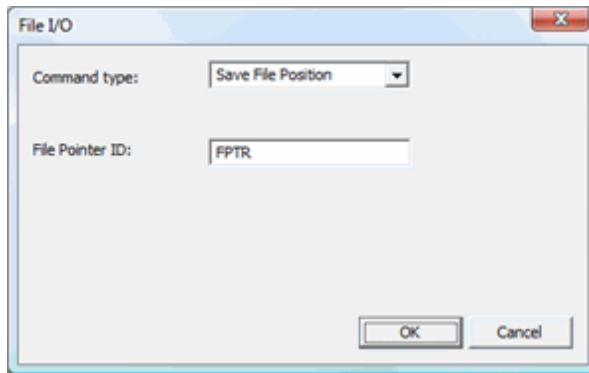
Beschreibungen für einige Komponenten dieser Befehle:

<Dateizeigername> - Der Name des Dateizeigers, dessen Dateiposition gespeichert wird.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Dateiposition speichern".
3. Drücken Sie F9.

Speichern der aktuellen Position eines Dateizeigers



Beispielcode für "Dateiposition speichern"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Sehen Sie sich dieses Beispiel an, das Daten aus einer externen Datei zeilenweise einliest. Nach jeder Zeile haben Sie die Möglichkeit, die Dateiposition zu speichern und später wieder aufzurufen. Das Beispiel beschreibt die Verwendung des Befehls `DATEI/POS_SPEICHERN`.



```

C1=KOMMENTAR/EINGABE,Geben Sie eine Datei ein, aus
der gelesen werden soll.
,(Vollständigen Pfad angeben.)
V1=DATEI/VORHANDEN,C1.INPUT
IF/V1<>0
    DO/
        FPTR=DATEI/ÖFFNEN,C1.INPUT,LESEN
        C2=KOMMENTAR/JANEIN,Möchten Sie die
        Dateiposition speichern und später aufrufen?
        Die Schleife wird beendet.
        IF/C2.INPUT == "JA"
            DATEI/POS_SPEICHERN,FPTR
            GOTO/QUITLOOP (SCHLEIFE_BEENDEN)
        END_IF/
        V2=DATEI/ZEILE_LESEN,FPTR,{ZEILE}
        KOMMENTAR/BEDIENER,"Die aktuelle Zeile
        lautet: " + ZEILE
        UNTIL/V2=="EOF"
    END_IF/
    DATEI/SCHLIESSEN,FPTR
    QUITLOOP (SCHLEIFE_BEENDEN)=MARKE/
    KOMMENTAR/BEDIENER,Der Lesevorgang wurde beendet.
    ROUTINE/ENDE

```

Erläuterung des Codes

Dieser Code entspricht dem Code, der unter "Beispielcode für 'Zum Start zurück'" beschrieben wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

C2=KOMMENTAR/JANEIN

Diese Zeile fragt, ob Sie die aktuelle Dateiposition speichern und die Schleife beenden möchten. Die JA/NEIN-Antwort wird in der Variablen `C2.INPUT` gespeichert.

DATEI/POS_SPEICHERN,FPTR

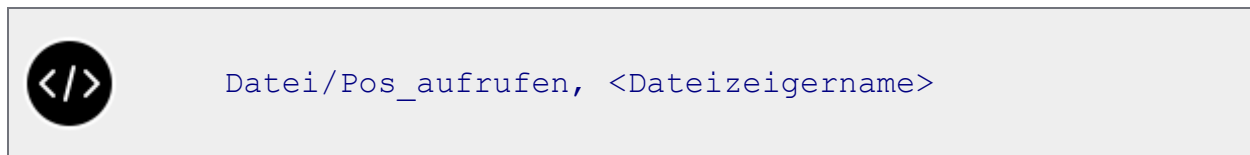
Diese Zeile speichert die Position des Dateizeigers im Dateistrom.

Wenn Sie die Datei mit demselben Dateizeigernamen und in derselben Messroutine wieder öffnen, können Sie eine gespeicherte Dateiposition aufrufen und von dort aus weiterlesen. Eine Fortsetzung dieses Beispiels finden Sie unter „Beispielcode für 'Dateiposition aufrufen'".

Aufrufen der gespeicherten Position eines Dateizeigers

Die Menüoption **Einfügen | Datei I/O-Befehl | Positionierungsbefehle | Dateiposition abrufen** fügt einen Befehl im Bearbeitungsfenster ein, durch den eine zuvor gespeicherte Dateiposition abgerufen wird. Verwenden Sie den Befehl "Dateiposition speichern", um eine Position in einer geöffneten Datei zu speichern.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:

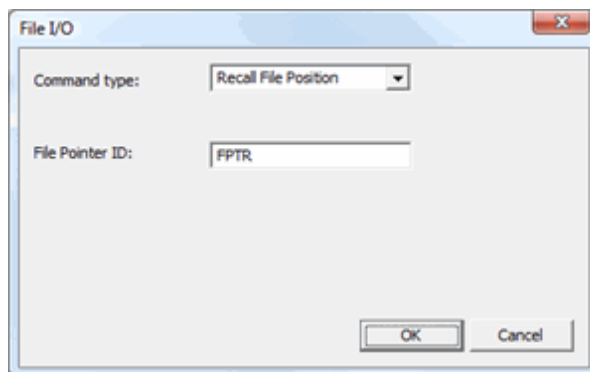


Beschreibungen für einige Komponenten dieser Befehle:

<Dateizeigername> - Der Name des Dateizeigers, dessen Position abgerufen wird.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Dateiposition aufrufen".
3. Drücken Sie F9.



Beispielcode für "Dateiposition abrufen"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

In diesem Beispiel wird eine zuvor gespeicherte Datei geöffnet, ein vorhandener Dateizeiger verwendet und die gespeicherte Position des Dateizeigers abgerufen. Anschließend werden die Daten aus dieser Position eingelesen. Dieser Code beschreibt die Verwendung des Befehls `DATEI/POS_AUFRUFEN`. Er führt den Beispielcode im Abschnitt Beispielcode für "Dateiposition speichern" fort.



```
KOMMENTAR/BEDIENER,Die Routine ruft nun die
gespeicherte Dateiposition auf.
FPTR=DATEI/ÖFFNEN,C1.INPUT,LESEN
DATEI/RÜCKLAUF,FPTR
KOMMENTAR/BEDIENER,Die Datei wurde zum Testen
zurückgespult.
,Zum Testen des Rücklaufs wird nun die erste Zeile
eingelesen.
V3=DATEI/ZEILE_LESEN,FPTR,{ZEILE}
KOMMENTAR/BEDIENER,Die aktuelle Zeile lautet:
,ZEILE
DATEI/RÜCKLAUF,FPTR
DATEI/POS_AUFRUFEN,FPTR
KOMMENTAR/BEDIENER,Die zuvor gespeicherte
Dateiposition wurde aufgerufen.
,Die Daten der Zeile an der gespeicherten Position
werden nun ausgegeben.
V4=DATEI/ZEILE_LESEN,FPTR,{GESPEICHERT}
KOMMENTAR/BEDIENER,Der Text an der gespeicherten
Position lautet:
,GESPEICHERT
```

Erläuterung des Codes

Dieser Code entspricht dem Code, der unter "Beispielcode für 'Zum Start zurück'" beschrieben wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

`DATEI/POS_AUFRUFEN,FPTR`

Kopieren einer Datei

Diese Zeile ruft die gespeicherte Dateizeigerposition im Dateistrom für den Dateizeiger `FPTR` auf.

```
V4=DATEI/ZEILE_LESEN,FPTR,{GESPEICHERT}
```

Diese Zeile liest die nächste Zeile nach der gespeicherten Dateizeigerposition ein und weist sie der benutzerdefinierten Variable `GESPEICHERT` zu. Die Variable wird dann im nächsten Bedienerkommentar ausgegeben.

Kopieren einer Datei

Die Menüoption **Einfügen | Datei I/O-Befehl | Datei kopieren** fügt einen Befehl in das Bearbeitungsfenster ein, bei dessen Ausführung ein Dateikopiervorgang stattfindet.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



```
Datei/Kopieren,<Quelldateiname>,<Zieldateiname>,<Ersetzen-Modus>
```

Beschreibungen für einige Komponenten dieser Befehle:

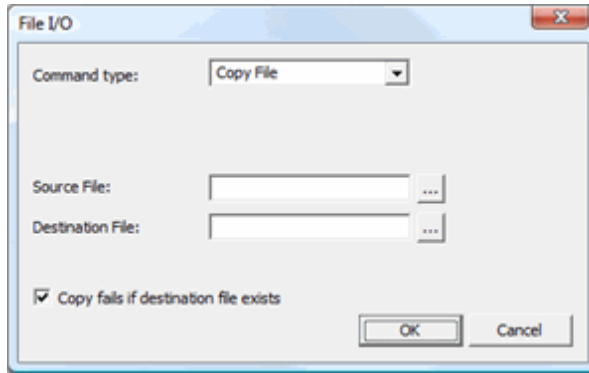
<Quelldateiname> - Der Name der Quelldatei (Datei, aus der kopiert wird)

<Zieldateiname> - Der Name der Zieldatei (Datei, in die kopiert wird).

<Ersetzen-Modus> - Die zu treffende Maßnahme, wenn die Zieldatei bereits existiert. Wenn das Ziel bereits existiert, lauten die beiden Modi "Überschreiben" und "Fehler".

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Datei kopieren".
3. Drücken Sie F9.



Beispielcode für "Datei kopieren"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Der folgende Code fragt nach der zu kopierenden Datei sowie nach einem Zielverzeichnis und der Datei, in die kopiert werden soll.

Kopieren einer Datei



```
C1=KOMMENTAR/EINGABE,Geben Sie die zu kopierende
Datei ein.
,(Vollständigen Dateipfad angeben.)
C2=KOMMENTAR/EINGABE,Geben Sie einen Namen für die
Zieldatei ein.
,(Vollständigen Dateipfad angeben.)
V1=DATEI/VORHANDEN,C1.INPUT
IF/V1<>0
    KOMMENTAR/BEDIENER,Die zu kopierende Datei ist
    vorhanden. Der Kopiervorgang beginnt.
    DATEI/KOPIEREN,C1.INPUT,C2.INPUT,FEHLER_WENN_ZIEL
    _VORHANDEN
    V2=DATEI/VORHANDEN,C2.INPUT
    IF/V2==0
        KOMMENTAR/BEDIENER,"Datendatei ist nicht
        vorhanden in: " + C2.INPUT.
        ,Kopiervorgang beendet.
        ROUTINE/ENDE
    END_IF/
ELSE/
    KOMMENTAR/BEDIENER,Kopiervorgang abgeschlossen.
    ROUTINE/ENDE
END_ELSE/
END_IF/
KOMMENTAR/BEDIENER,Die zu kopierende Datei ist nicht
vorhanden.
```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Zeichen lesen'" oder unter "Beispielcode für 'Zeile lesen'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

C1=KOMMENTAR/EINGABE

Diese Zeile nimmt den vollständigen Pfad der zu kopierenden Datei und platziert ihn in der Variablen `C1.INPUT`.

C2=KOMMENTAR/EINGABE

Diese Zeile nimmt den vollständigen Pfad der Zieldatei und platziert ihn in der Variablen `C2.INPUT`.

DATEI/KOPIEREN,C1.INPUT,C2.INPUT,FEHLER_WENN_ZIEL_VORHANDEN

Diese Zeile kopiert die Originaldatei in eine Zieldatei. Dieser Befehl verwendet drei Parameter.

- Parameter 1 ist `C1.INPUT`. Dies ist der vollständige Pfad zur Datei, die kopiert werden soll.
- Parameter 2 ist `C2.INPUT` oder der vollständige Pfad zur Zieldatei.
- Parameter 3 bricht in diesem Fall den DATEI/KOPIEREN-Vorgang ab, wenn bereits eine Datei mit demselben Zieldateinamen vorhanden ist. Sie können diesen Parameter so einstellen, dass eine vorhandene gleichnamige Datei überschrieben wird.

Befehlsmodus-Befehle nach Kommentaren

Verschieben einer Datei

Die Menüoption **Einfügen | Datei I/O-Befehl | Datei verschieben** fügt einen Befehl in das Bearbeitungsfenster ein, bei dessen Ausführung ein Dateischiebevorgang stattfindet.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



`Datei/Verschieben,<alterDateiname>,<neuerDateiname>`

Beschreibungen für einige Komponenten dieser Befehle:

<alterDateiname> - Die Speicherposition und der Name der Datei.

<neuerDateiname> - .Die neue Speicherposition und der Name der Datei.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Datei verschieben".
3. Drücken Sie F9.

Verschieben einer Datei



Beispielcode für "Datei verschieben"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Der folgende Code fragt nach der zu verschiebenden Datei, nach dem Zielverzeichnis und dem Namen der Zieldatei. Der Code führt dann die Dateiverschiebung durch.



```

C1=KOMMENTAR/EINGABE,Geben Sie die zu verschiebende
Datei ein.
,(Vollständigen Dateipfad angeben.)
C2=KOMMENTAR/EINGABE,Geben Sie einen Namen für die
Zieldatei ein.
,(Vollständigen Dateipfad angeben.)
V1=DATEI/VORHANDEN,C1.INPUT
IF/V1<>0
    KOMMENTAR/BEDIENER,Die zu verschiebende Datei
    ist vorhanden. Die Dateiverschiebung beginnt.
    DATEI/VERSCHIEBEN,C1.INPUT,C2.INPUT
    V2=DATEI/VORHANDEN,C2.INPUT
    IF/V2==0
        KOMMENTAR/BEDIENER,"Datendatei ist nicht
        vorhanden in: " + C2.INPUT.
        ,Die Verschiebung wurde nicht ordnungsgemäß
        durchgeführt.
        ROUTINE/ENDE
    END_IF/
ELSE/
    KOMMENTAR/BEDIENER,Verschiebungsvorgang
    abgeschlossen.
    ROUTINE/ENDE
END_ELSE/
END_IF/
KOMMENTAR/BEDIENER,Die Originaldatei ist nicht
vorhanden. Versuchen Sie es erneut.

```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Datei verschieben'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

DATEI/VERSCHIEBEN,C1.INPUT,C2.INPUT

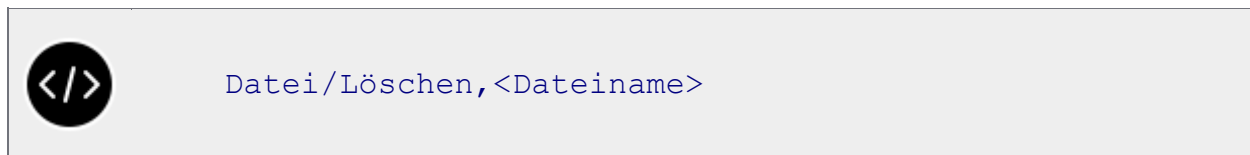
Diese Zeile kopiert die Originaldatei in eine Zieldatei. Dieser Befehl verwendet zwei Parameter.

- Parameter 1 ist **C1.INPUT**. Dies ist der vollständige Pfad zu der zu verschiebenden Datei.
- Parameter 2 ist **C2.INPUT** oder der vollständige Pfad zur Zieldatei.

Löschen einer Datei

Die Menüoption **Einfügen | Datei I/O-Befehl | Datei löschen** fügt einen Befehl in das Bearbeitungsfenster ein, bei dessen Ausführung eine Datei gelöscht wird.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:

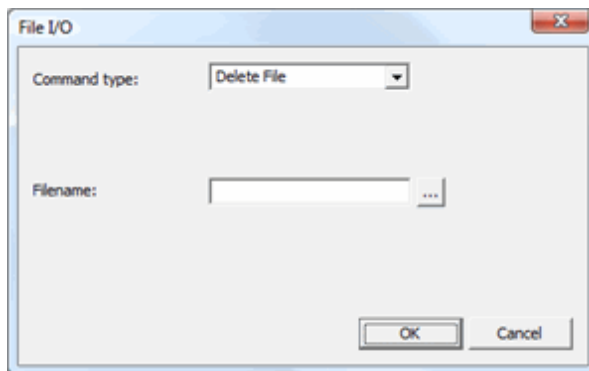


Beschreibungen für einige Komponenten dieser Befehle:

<Dateiname> - Der Name der zu löschenden Datei.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Datei löschen".
3. Drücken Sie F9.



Beispielcode für "Datei löschen"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Der folgende Code fragt nach einem Dateinamen und löscht dann die Datei.



```
C1=KOMMENTAR/EINGABE,Geben Sie die zu löschende
Datei ein.
,(Vollständigen Dateipfad angeben.)
V1=DATEI/VORHANDEN,C1.INPUT
IF/V1<>0
    KOMMENTAR/BEDIENER,Die Datei ist vorhanden.
    Bereit für den Löschvorgang.
    DATEI/LÖSCHEN,C1.INPUT
    V2=DATEI/VORHANDEN,
    IF/V2==0
        KOMMENTAR/BEDIENER,Die Datei wurde gelöscht.
        ROUTINE/ENDE
    END_IF/
ELSE/
    KOMMENTAR/BEDIENER,Die Datei ist noch vorhanden.
    ROUTINE/ENDE
END_ELSE/
END_IF/
KOMMENTAR/BEDIENER,Die zu löschende Datei ist nicht
vorhanden. Wählen Sie eine vorhandene Datei aus.
```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Datei verschieben'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

DATEI/LÖSCHEN,C1.INPUT - Diese Zeile löscht die angegebene Datei. Dieser Befehl verwendet einen Parameter, den Namen der zu löschenden Datei. In diesem Fall ist dies **C1.INPUT**.

Prüfen des Vorhandenseins einer Datei

Die Menüoption **Einfügen | Datei I/O-Befehl | Datei vorhanden** fügt einen Befehl in das Bearbeitungsfenster ein, bei dessen Ausführung das Vorhandensein einer Datei überprüft und die angegebene Variable auf das Ergebnis gesetzt wird.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



```
<Variablenname> = Datei/Vorhanden,<Dateiname>
```

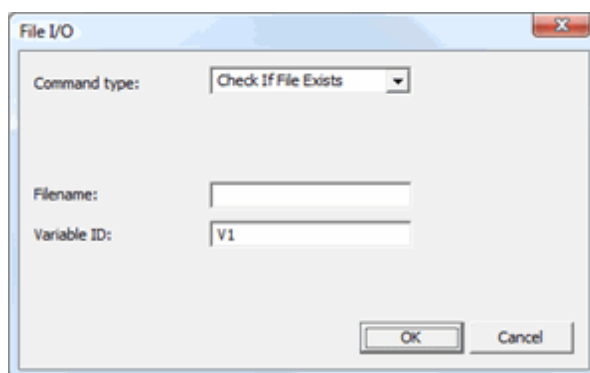
Beschreibungen für einige Komponenten dieser Befehle:

<Variablenname> - Der Name der Variablen, die auf das Ergebnis der durchgeführten Prüfung gesetzt wird. Die Variable wird auf den Wert 1 gesetzt, wenn die Datei existiert und auf 0, wenn sie nicht existiert.

<Dateiname> - Der Name der Datei, die daraufhin überprüft wird, ob sie bereits auf dem Datenträger existiert

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Datei vorhanden".
3. Drücken Sie F9.



Beispielcode für "Datei vorhanden"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Der folgenden Code fragt nach einem Dateinamen und prüft dann, ob die Datei vorhanden ist.



```
C1=KOMMENTAR/EINGABE,Geben Sie die zu überprüfende
Datei ein.
V1=DATEI/VORHANDEN,C1.INPUT
IF/V1<>0
    KOMMENTAR/BEDIENER,Die Datei ist vorhanden.
END_IF/
ELSE/
    KOMMENTAR/BEDIENER,Die Datei ist nicht
    vorhanden.
END_ELSE/
```

Erläuterung des Codes

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Zeichen lesen'" oder unter "Beispielcode für 'Zeile lesen'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

V1=DATEI/VORHANDEN,C1.INPUT

Diese Zeile überprüft, ob die angegebene Datei vorhanden ist. Die Datei muss sich im Verzeichnis befinden, in dem PC-DMIS installiert ist, damit dieser Code funktioniert. Andernfalls muss die Zeile, welche die Datei enthält, auch den vollständigen Pfad der Datei enthalten. **V1** erhält das Ergebnis der Dateiprüfung. Ist die Datei vorhanden, ist der Wert ungleich Null. Andernfalls lautet er 0.

Anzeigen eines Dateidialogfelds

Die Menüoption **Einfügen | Datei I/O-Befehl | Dateidialogfeld** fügt einen Befehl in das Bearbeitungsfenster ein, bei dessen Ausführung das Dialogfeld **Öffnen** eingeblendet wird. Damit hat der Bediener die Möglichkeit, zur Ausführungszeit einen Dateinamen zu wählen. Der gewählte Dateiname wird in der vorgegebenen Variablen gespeichert.

Anzeigen eines Dateidialogfelds

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:



```
<Variablenname> = Datei/Dialogfeld,<Ausdruck>
```

Beschreibungen für einige Komponenten dieser Befehle:

<Variablenname> - Der Name der Variablen, die auf den Namen der Datei, die der Benutzer im Dateidialogfeld gewählt hat, gesetzt wird.

<Ausdruck> - Der Text, der in der Titelleiste des Dateidialogfelds erscheint.

So rufen Sie das mit diesem Datei I/O-Befehl verknüpfte Dialogfeld auf:

1. Öffnen Sie das Bearbeitungsfenster.
2. Setzen Sie den Cursor auf den Befehl "Dateidialogfeld".
3. Drücken Sie F9.



Beispielcode für "Dateidialogfeld"



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Der folgende Code zeigt ein Dialogfeld an, in dem Sie eine zu löschende Datei auswählen können.



```

V1=DATEI/DIALOGFELD,Zu löschende Datei auswählen
V2=DATEI/VORHANDEN,V1
IF/V2<>0
    KOMMENTAR/BEDIENER,Die Datei ist vorhanden.
    Bereit für den Löschvorgang.
    DATEI/LÖSCHEN,V1
    V3=DATEI/VORHANDEN,
    IF/V3==0
        KOMMENTAR/BEDIENER,Die Datei wurde gelöscht.
        ROUTINE/ENDE
    END_IF/
ELSE/
    KOMMENTAR/BEDIENER,Die Datei ist noch vorhanden.
    ROUTINE/ENDE
END_ELSE/
END_IF/
KOMMENTAR/BEDIENER,Die zu löschende Datei ist nicht
vorhanden. Wählen Sie eine vorhandene Datei aus.

```

Dieser Code entspricht weitgehend dem Code, der unter "Beispielcode für 'Zeichen lesen'" oder unter "Beispielcode für 'Zeile lesen'" erläutert wird.

Die hier angeführten Erläuterungen beziehen sich nur auf dieses Beispiel.

V1=DATEI/DIALOGFELD,Zu löschende Datei auswählen

Diese Zeile zeigt ein Dialogfeld mit dem Titel "Zu löschende Datei auswählen" an. Darin können Sie nach einer Datei suchen, und wenn Sie auf **Öffnen** klicken, übergibt PC-DMIS den vollständigen Dateipfad der ausgewählten Datei an V1. Der Rest der Messroutine löscht die ausgewählte Datei.

Prüfen auf das Ende einer Datei oder Ende einer Zeile

In PC-DMIS können Sie das Ende einer Datei prüfen, indem Sie die Funktion **EOF** oder **EOL** in einer Bedingungsprüfung verwenden.

EOF steht für END OF FILE (ENDE DER DATEI). Diese Funktion verwendet einen Dateizeiger mit einer Zeichenfolge. Wird die Funktion richtig innerhalb der Bedingungsanweisung platziert, prüft sie den Dateizeiger daraufhin, ob er das Ende der angegebenen Datei erreicht hat. In diesem Fall gibt die Funktion "wahr" wieder.

Prüfen auf das Ende einer Datei oder Ende einer Zeile

EOL steht für END OF LINE (ENDE DER ZEILE). Diese Funktion verwendet einen Dateizeiger mit einer Zeichenfolge. Wird die Funktion richtig innerhalb der Bedingungsanweisung platziert, prüft sie den Dateizeiger daraufhin, ob er das Zeilenende in der angegebenen Datei erreicht hat. In diesem Fall gibt die Funktion "wahr" wieder. Diese Funktion eignet sich am besten für eine Schleife.

Die im Bearbeitungsfenster angezeigte Befehlssyntax lautet:

`EOF(<Dateizeiger>) oder EOL(<Dateizeiger>)`

Beschreibungen für einige Komponenten dieser Befehle:

<Dateizeiger> - Der Name des Dateizeigers, der geprüft wird.

Beispielcode für EOF und EOL



Der unten stehende Beispielcode sollte im Befehlsmodus des Bearbeitungsfensters eingegeben werden, und nicht innerhalb des Dialogfeldes **Datei I/O**.

Der folgende Code öffnet eine Datei namens test.txt und liest die Datei durch. Solange das Ende der Datei (das durch den Code `WHILE/!EOF` gekennzeichnet ist) nicht erreicht wurde, liest PC-DMIS die Datei zeichenweise durch, und weist V1 ein Zeichen zu.

Hat PC-DMIS das Ende einer Zeile in dieser Datei erreicht, wird das letzte Zeichen dieser Zeile angezeigt.

Dieser Vorgang wird solange wiederholt, bis PC-DMIS das Dateiende erreicht hat. PC-DMIS blendet dann den Text "Dateiende erreicht..." ein.



```
FPTR=DATEI/ÖFFNEN,D:\temp\test.txt,LESEN
WHILE/!EOF("FPTR")
V1=DATEI/ZEICH_LESEN,FPTR
IF/EOL("FPTR")
KOMMENTAR/BEDIENER,NEIN,"Zeilenende erreicht. Das
letzte Zeichen lautet:"
,V1
END_IF/
END_WHILE/
KOMMENTAR/BEDIENER,NEIN,"Dateiende erreicht..."
```

