

Sommario

Uso del file di Input/Output	1
Uso del file di Input/Output: Introduzione	1
Informazioni di base sul file I/O	2
Uso della finestra di dialogo I/O file	4
Apertura di un file per la lettura o la scrittura dei dati	4
Esempio di codice per il comando di apertura di un file	6
Chiusura di un file al termine delle operazioni di lettura e scrittura	6
Esempio di codice per il comando di chiusura di un file	7
Letture di un carattere da un file	8
Esempio di codice per il comando Leggi carattere	9
Letture di una riga da un file	11
Esempio di codice per il comando Leggi riga	12
Letture di un blocco di testo da un file	18
Esempio di codice per il comando Leggi blocco	19
Letture di un testo fino a un delimitatore	22
Esempio di codice per il comando Leggi fino a	23
Scrittura di un carattere in un file	25
Esempio di codice per il comando Scrivi carattere	26
Scrittura di una riga in un file	28
Esempio di codice per il comando Scrivi riga	29
Scrittura di un blocco di testo in un file	30
Esempio di codice per il comando Scrivi blocco	32

Posizionamento del puntatore all'inizio di un file.....	33
Esempio di codice per il comando di riposizionamento all'inizio	34
Salvataggio della posizione corrente del puntatore di un file	35
Esempio di codice per il comando Salva posizione file	36
Richiamo della posizione salvata del puntatore di un file.....	38
Esempio di codice per il comando Richiama posizione file	39
Copia di un file.....	40
Esempio di codice per il comando Sposta file	41
Spostamento di un file	43
Esempio di codice per lo spostamento di un file	44
Eliminazione di un file	46
Esempio di codice per il comando Elimina file	47
Verifica dell'esistenza di un file	48
Esempio di codice per il comando File esistente.....	49
Visualizzazione della finestra di dialogo di un file	49
Esempio di codice per il comando Finestra file	51
Controllo di fine file o fine linea.....	52
Esempio di codice per EOF e EOL	52

Uso del file di Input/Output

Uso del file di Input/Output: Introduzione

In questo capitolo vengono spiegate le modalità di input e output delle informazioni da e in una routine di misurazione. Le opzioni disponibili nel menu permettono di aprire i file in modalità di lettura o di scrittura. Sarà possibile scrivere i dati nei file o leggerli da essi. I comandi di I/O dei file permettono di leggere e acquisire da file esterni i dati che devono essere usati in una routine di misurazione. Inoltre, usando questi comandi sarà possibile riscrivere sui file esterni le informazioni sulle misure e le tolleranze. Questi comandi permettono di eseguire anche altre operazioni sui file.

In questo capitolo vengono fornite informazioni dettagliate sulle operazioni di I/O oltre a esempi pratici delle varie operazioni. Questi esempi contengono gli elementi descritti nei capitoli "Esecuzione della diramazione utilizzando il controllo del flusso" e "Uso di espressioni e di variabili".

In questa sezione vengono descritti i seguenti argomenti:

- Informazioni di base sul file I/O
- Uso della finestra di dialogo I/O file
- Apertura di un file per la lettura o la scrittura dei dati
- Chiusura di un file al termine delle operazioni di lettura e scrittura
- Lettura di un carattere da File
- Lettura di una riga da un file
- Lettura di un blocco di testo da un file
- Lettura di un testo fino a un delimitatore
- Scrittura di un carattere in un file
- Scrittura di una riga in un file
- Scrittura di un blocco di testo in un file
- Posizionamento del puntatore all'inizio di un file
- Salvataggio della posizione corrente del puntatore di un file
- Richiamo della posizione salvata del puntatore di un file
- Copia di un File
- Spostamento file
- Eliminazione file
- Verifica dell'esistenza di un file
- Visualizzazione della finestra di dialogo di un file

- Controllo della fine di un file o di una riga

Comandi nella modalità di comando dopo i commenti

Poiché molti degli esempi di codice in questo capitolo usano comandi `COMMENT` digitati, considerare quanto segue.



Per immettere ulteriori comandi di PC-DMIS in modalità di comando dopo aver inserito un commento di PC-DMIS, si deve prima premere *due volte* il tasto Invio dopo il comando `COMMENT`. Questo comunicherà a PC-DMIS che non si desidera aggiungere testo al commento, ma si è pronti ad aggiungere un nuovo comando.

Informazioni di base sul file I/O

Verifica dell'esistenza di un file:

prima di eseguire le operazioni di input e output su un file, è possibile verificare l'effettiva esistenza del file. Questa operazione dovrebbe essere probabilmente inserita in un ciclo IF/THEN in modo che se la verifica ha esito negativo è possibile informare l'utente. Prima di eseguire un'operazione di scrittura su un file, è necessario creare tale file nell'ambiente Windows.

Vedere "Verifica dell'esistenza di un file".

Apertura e chiusura dei file:

prima di eseguire le operazioni di lettura o di scrittura nei file, è necessario aprire tali file nei processi di sistema. È possibile eseguire questa operazione assegnando il file a una variabile denominata puntatore del file. Quando si apre un file, è possibile specificare se si desidera aprire tale file in lettura, in scrittura (sovrascrittura) o come allegato. Quando il file è aperto, è possibile eseguire le operazioni di lettura o scrittura. Una volta terminato di utilizzare un file chiudere il puntatore; in questo modo il file viene chiuso e risulta accessibile ad altri processi del sistema. Non è possibile aprire file che sono già aperti da un altro processo.

Vedere "Apertura di un file per la lettura o la scrittura dei dati" e "Chiusura di un file al termine delle operazioni di lettura e scrittura".

Puntatori e posizioni dei file:

I puntatori sono variabili che rimandano a un file. In essi sono memorizzati il nome e la posizione di un file aperto e vengono pertanto utilizzati per eseguire le operazioni di

lettura e scrittura sul file in questione. Quando un file è aperto e per esso è stato impostato un puntatore, tale puntatore funziona come il cursore di un elaboratore di testi. Tramite i puntatori è possibile conoscere la posizione effettiva in cui si stanno eseguendo le operazioni di lettura e scrittura all'interno del file.

- Ad esempio, se si desidera allegare dati a un file, la posizione ideale del puntatore è alla fine del file.
- Se invece si desidera eseguire operazioni di lettura o di sovrascrittura, il puntatore dovrebbe essere posizionato all'inizio di un file.

Uso dei delimitatori durante le operazioni di scrittura e di lettura

Quando si scrivono dei dati, si consiglia di utilizzare delimitatori per separare i singoli dati. Questo facilita la lettura dei dati in una routine di misurazione. Un delimitatore può essere costituito da un qualsiasi carattere o da una stringa di caratteri. Ad esempio, si supponga di avere un punto denominato PNT1 e dei valori X,Y e Z misurati pari a 2.5,4.3,6.1. È possibile scrivere facilmente questi valori separati da una virgola in un file di dati con un codice simile al seguente:



```
FILE/WRITELINE,FPTR,PNT1.X + "," + PNT1.Y + "," +  
PNT1.Z
```

Durante la lettura dei dati, è possibile separare i dati in entrata in base a un delimitatore specificato e inserire i dati in variabili per una modifica successiva. Si supponga ad esempio di voler leggere i valori X, Y e Z elencati in precedenza. I valori devono trovarsi su una sola riga di testo, ad esempio: 2.5,4.3,6.1. È possibile separare il testo in corrispondenza della virgola e posizionare quei valori in corrispondenza delle variabili utilizzando una riga di codice simile alla seguente:

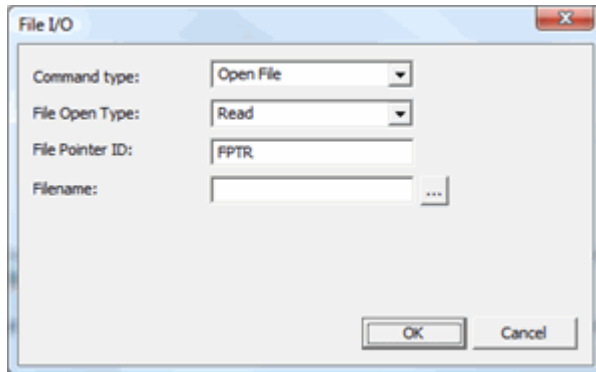


```
V1=FILE/LEGGI RIGA,FPTR,{ValX}+"," +{ValY}+"," +{ValZ}
```

Quindi, è possibile usare nella routine di misurazione `ValX`, `ValY` e `ValZ` come normali variabili. Ne risulta `ValX = 2.5`, `ValY = 4.3` e `ValZ = 6.1`

Uso della finestra di dialogo I/O file

Tutti i Comandi di I/O file vengono inizialmente inseriti nella routine di misurazione selezionando l'opzione del menu dei file I/O appropriata (selezionare **Inserisci | Comando di I/O file** dal menu). Se un comando esiste nella finestra di modifica, premendo F9 sul comando si apre la finestra di dialogo **I/O file** associata.



Finestra di dialogo File I/O

Questa finestra di dialogo fornisce un modo visivo per modificare il comando *attuale* di input/output di un file. In alternativa, è possibile modificare un comando nella finestra di modifica con le tecniche illustrate nel capitolo “Uso della finestra di modifica”.

Questa finestra di dialogo non consente di inserire *nuovi* comandi di I/O file. Questa operazione deve essere eseguita selezionando la voce di menu appropriata oppure digitando i comandi direttamente nella finestra di modifica.

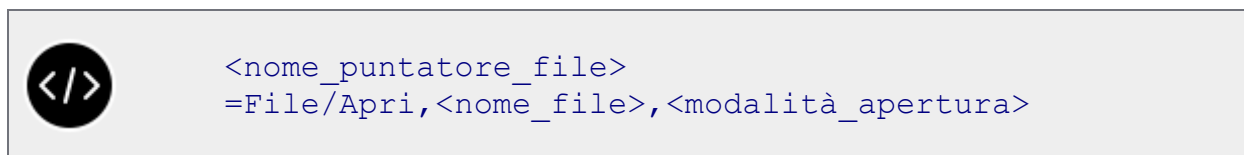
Apertura di un file per la lettura o la scrittura dei dati

La voce del menu **Inserisci | Comando di I/O file | Apri file** consente di inserire nella finestra di modifica un comando che apre un file presente sul computer durante l'esecuzione della routine di misurazione.

È possibile aprire i file per aggiungere e salvare le informazioni oppure semplicemente per visualizzarle.

Uso dei file di input e output

La sintassi di questo comando nella finestra di modifica è la seguente:



Descrizioni di alcuni dei componenti di questo comando.

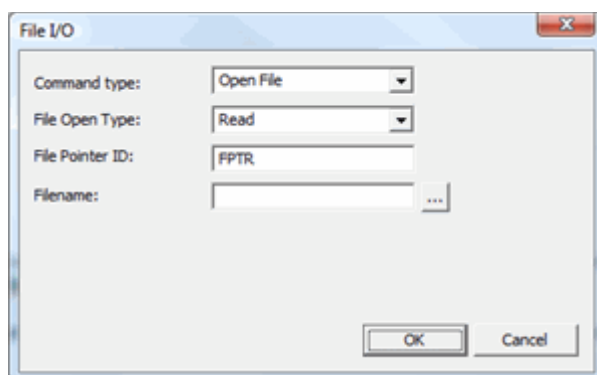
<nome_puntatore_file> - È l'ID definito dall'utente del puntatore del file utilizzato per accedere al file aperto. Questo ID viene utilizzato per fare riferimento al file aperto in altri comandi file I/O.

<nome_file> - È il nome del file da aprire.

<modalità_apertura> - È la modalità in cui il file deve essere aperto. I file possono essere aperti nelle seguenti modalità: Lettura, Scrittura, Aggiunta.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sull'opzione Apri file.
3. Premere F9.



Esempio di codice per il comando di apertura di un file



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Questo codice apre un file chiamato TEST.TXT per lettura, scrittura e aggiunta. Memorizza il nome del file in un puntatore chiamato **FPTR**.



```
FPTR=FILE/APRI,C:\PCDMISW\TEST.TXT,LEGGI
FPTR=FILE/APRI,C:\PCDMIWS\TEST.TXT,SCRIVI
FPTR=FILE/APRI,C:\PCDMIWS\TEST.TXT,AGGIUNGI
```

È possibile utilizzare un commento di input prendendo il percorso completo come input e utilizzandolo nel comando **FILE/APRI**. È possibile eseguire la stessa operazione utilizzando il comando **FILE/FINESTRA**. Fare riferimento agli esempi seguenti:



```
C1=COMMENTO/INPUT,Digitare il nome e il percorso
completi del file.
V1=FILE/FINESTRA, SCEGLIERE UN FILE DA APRIRE
FPTR=FILE/OPEN,C1.INPUT,READ
FPTR=FILE/APRI,V1,LEGGI
```

Vedere "Visualizzazione della finestra di dialogo di un file".

Chiusura di un file al termine delle operazioni di lettura e scrittura

L'opzione del menu **Inserisci | Comando di I/O file | Chiudi file** consente di inserire nella finestra di modifica un comando che causa la chiusura di un file aperto durante l'esecuzione della routine di misurazione. La chiusura dei file consente di rendere disponibili risorse di memoria utilizzate per i file aperti e di salvare sul disco rigido le modifiche apportate ai file.

La sintassi di questo comando nella finestra di modifica è la seguente:

```
File/Chiudi, <nome_puntatore_file>,<modalità_chiusura>
```

Descrizioni di alcuni dei componenti di questo comando.

<nome_puntatore_file>

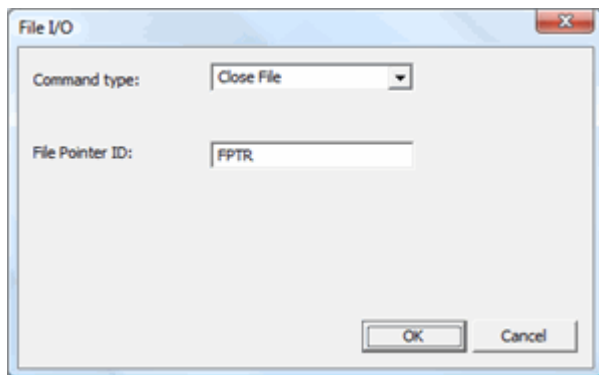
È l'ID creato al momento dell'apertura del file e utilizzato per identificare il file stesso.

<modalitàchiusura>

Questo parametro ha due opzioni, KEEP o DELETE. Se si sceglie MANTIENI, PC-DMIS chiude semplicemente il file indicato nel puntatore del file. Se si sceglie ELIMINA, PC-DMIS chiude il file e lo elimina.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando Chiudi file.
3. Premere F9.



Esempio di codice per il comando di chiusura di un file



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Il codice seguente consente di chiudere il file assegnato al puntatore del file **FPTR**:



```
FILE/CHIUDI,FPTR,MANTIENI
```

Questo codice, che usa il parametro `ELIMINA`, chiude ed elimina il file assegnato a `FPTR`:



`FILE/CHIUDI, FPTR, ELIMINA`

Lettura di un carattere da un file

L'opzione del menu **Inserisci | Comando file I/O | Comandi di lettura | Caratteri di lettura** inserisce un comando nella finestra di modifica che legge un singolo carattere dal file specificato dal campo `nome_puntatore_file` (vedere la sintassi di seguito). Quindi, assegna tale carattere alla variabile specificata nel campo del nome della variabile.

La sintassi di questo comando nella finestra di modifica è la seguente:



```
<nome_variabile> =  
File/Leggi_Caratt,<nome_puntatore_file>
```

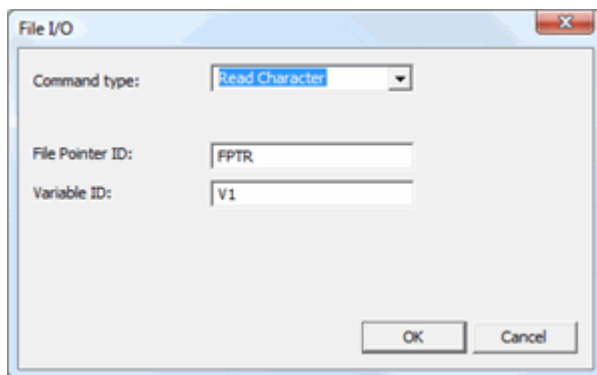
Descrizioni di alcuni dei componenti di questo comando.

<nome_puntatore_file> - È l'ID utilizzato per aprire il file.

<nome_variabile> - È il nome della variabile che contiene il carattere.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando di lettura del carattere.
3. Premere F9.



Esempio di codice per il comando Leggi carattere



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Nell'esempio riportato di seguito le righe di un file di dati vengono lette un carattere alla volta finché non viene rilevato il primo spazio.



```
V1=FILE/ ESISTE,test.txt
IF/V1<>0
    COMMENTO/OPER,Lettura da un file di dati
    possibile. Fare clic su OK per continuare.
    ASSEGNA/V3 = ""
    FPTR=FILE/APRI,D:\Programmi\pcdmis35\test.txt,LE
    GGI
    DO/
        V2=FILE/LEGGI CARATTERE,FPTR
        ASSEGNA/V3 = V3+V2
    UNTIL/V2 == " "
    FILE/CHIUDI,FPTR
    COMMENTO/OPER,"La prima parola di una riga di
    testo del file è: " + V3
END_IF/
ELSE/
    COMMENTO/OPER,Impossibile leggere da un file di
    dati. La routine si chiuderà.
    GOTO/FINE
END ELSE/
FINE=ETICHETTA/
ROUTINE/FINE
```

Spiegazione del codice

V1=FILE/ESISTE

Questa riga verifica l'esistenza del file specificato. Per fare in modo che questo codice funzioni, il file deve trovarsi nella stessa directory di PC-DMIS; in caso contrario la riga contenente il file deve includere il percorso completo del file. **V1** riceve il risultato del controllo del file. Se il file esiste viene restituito un valore diverso da zero, in caso contrario viene restituito un valore uguale a zero.

IF/V1<>0

Questa riga registra il valore della variabile **V1** e verifica se il valore restituito è diverso da zero. In quest'ultimo caso, viene visualizzato un commento indicante che il processo di lettura può essere avviato. Se il valore è uguale a zero, la routine di misurazione si chiude.

ASSEGNA/V3 = ""

Questa riga crea una stringa vuota e la assegna a **V3**. Il codice usa questa variabile per creare una stringa dalla lettura dei singoli caratteri. Se non viene creata alcuna stringa vuota, alla variabile **V3** viene assegnato il valore predefinito 0.

FPTR=FILE/APRI

Questa riga apre il file specificato per la lettura e lo assegna al puntatore del file predefinito **FPTR**.

DO

Questa riga avvia un ciclo **DO / UNTIL**. A tale ciclo è associato il codice **FILE/LEGGI CARATTERE** in modo che i caratteri vengano letti uno alla volta, continuamente. Il ciclo si interrompe ogni volta che viene rilevato uno spazio tra i caratteri.

V2=FILE/LEGGI CARATTERE,FPTR

Questa riga legge un carattere dal file aperto collegato al puntatore del file **FPTR**. Il carattere viene memorizzato nella variabile, **V2**.

ASSEGNA/V3 = V3+V2

Questa riga utilizza la variabile **V3** vuota, concatena la stringa **V3** con **V2**, quindi riassegna il valore a **V3**. Quindi, alle successive esecuzioni del ciclo **DO/UNTIL**, uno o più caratteri saranno aggiunti a **V3**.

```
UNTIL/v2 == " "
```

Questa riga arresta il ciclo DO/UNTIL quando il codice `FILE/LEGGI CARATTERE` rileva uno spazio nel file aperto.

```
FILE/CHIUDI , FPTR
```

Questa riga chiude il file di dati aperto, rendendolo accessibile agli altri processi del sistema. La parte rimanente del codice completa l'esecuzione e viene visualizzata la prima parola del file di dati in un commento dell'operatore.

Lettura di una riga da un file

L'opzione di menu **Inserisci | Comando file I/O | Comandi di lettura | Leggi riga** consente di inserire nella finestra di modifica un comando per la lettura di una riga dal file specificato durante l'esecuzione. Questo comando imposta la variabile specificata dal relativo ID su 1 (vero) o su 0 (falso) per indicare che la chiamata è stata eseguita correttamente (vero) o meno (falso). L'espressione richiesta da questo comando può essere utilizzata per delimitare la riga di lettura e completare automaticamente le variabili e i riferimenti con i dati letti dal file. Le informazioni vengono lette dal file di input fino al successivo carattere di ritorno a capo.

La sintassi di questo comando nella finestra di modifica è la seguente:



```
<nome_variabile> =  
File/Leggi_Riga,<nome_puntatore_file>,<espressione>
```

Descrizioni di alcuni dei componenti di questo comando.

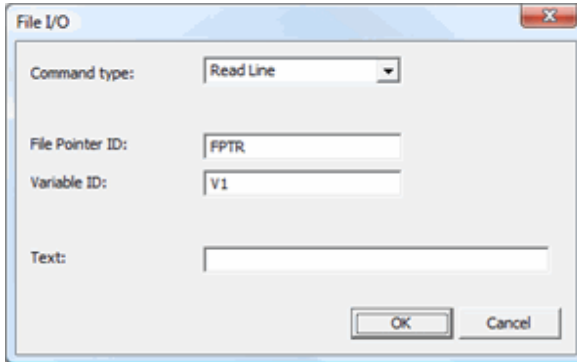
<nome_variabile> - È il nome della variabile che contiene il risultato del comando di lettura della riga. Viene restituito "OK" o "EOF".

<nome_puntatore_file> - È il nome del puntatore del file specificato al momento dell'apertura del file.

<espressione> - È la variabile di destinazione dei dati di input. I dati di input possono essere delimitati da testo per facilitare l'analisi delle righe di dati acquisite. Le variabili e gli elementi di riferimento devono essere racchiusi tra parentesi graffe.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando di lettura della linea.
3. Premere F9.



Esempio di codice per il comando Leggi riga



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Nell'esempio riportato di seguito le righe di un file di dati vengono lette una alla volta finché il comando `FILE/LEGGI RIGA` non rileva una riga vuota. La routine di misurazione visualizza quindi il blocco di testo trovato e si chiude.

```
.
.
V1      =FILE/EXISTS,D:\HEXAGON\PCDMIS
FILES\BASIC_SCRIPTS\TEST.TXT
      IF/V1<>0
          COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,OVC=NO,
          Able to read from data file. Fare clic su OK per
continuare.
          ASSIGN/V3=""
FPTR    =FILE/OPEN,D:\HEXAGON\PCDMIS
FILES\BASIC_SCRIPTS\TEST.TXT,READ
      DO/
V2      =FILE/READLINE,FPTR,{LINE}
          ASSIGN/V3=V3+LINE
          COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,OVC=NO,
          "The current value of variable V3 is: "+V3
          UNTIL/V2=="EOF"
          FILE/CLOSE,FPTR,DELETE
          COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,OVC=NO,
          "The block of text reads as follows: "+V3
          END_IF/
          ELSE/
          COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,OVC=NO,
          Wasn't able to read from data file. La routine
si chiuderà.
          GOTO/END
          END_ELSE/
END      =LABEL/
.
.
.
```

Spiegazione del codice

Questo codice è molto simile a quello descritto in "Esempio di codice per il comando Leggi carattere". Le spiegazioni riportate di seguito riguardano soltanto la parte di codice specifica per questo esempio.

DO

Questa riga avvia un ciclo DO / UNTIL. A tale ciclo è associato il codice `FILE/LEGGI RIGA` in modo che le righe vengano lette una alla volta, continuativamente. Il ciclo viene interrotto quando viene raggiunta la fine del file.

`V2=FILE/LEGGI RIGA,FPTR,{RIGA}`

Questa riga legge il testo finché non viene rilevato un ritorno a capo. Anziché memorizzare il testo in `V2`, come `FILE/LEGGI CARATTERE`, questo codice prevede un funzionamento diverso.

- `V2` in questo caso restituisce due valori: "OK" o "EOF". "OK" se contiene ancora una riga da leggere. "EOF" se viene raggiunta la fine del file.
- Il codice `{RIGA}` è una variabile inserita dall'utente che consente di memorizzare il testo effettivo. Viene racchiuso tra parentesi graffe per indicare a PC-DMIS che si tratta di una variabile e non di una parte di testo di delimitazione. Senza parentesi graffe, PC-DMIS esegue la ricerca di una stringa di caratteri nel file denominato "RIGA" e restituisce soltanto il testo posizionato dopo "RIGA" e prima del ritorno a capo.

`ASSEGNA/V3 = V3 + RIGA`

Questa riga utilizza la variabile vuota `V3` per concatenare la stringa `V3` con `RIGA`, quindi riassegna il valore concatenato a `V3`. Quindi alle successive esecuzioni del ciclo DO/UNTIL, una o più righe saranno aggiunte a `V3`.

`UNTIL/V2=="EOF"`

Questa riga verifica la condizione del ciclo DO/UNTIL. Quando il codice `FILE/LEGGI RIGA` raggiunge la fine del file, il ciclo viene interrotto. Quando il flusso della routine esce dal ciclo, viene interrotta l'esecuzione del codice rimanente e l'intero blocco di codice viene visualizzato in un commento per l'operatore.



Risultato = File/RLeggiLinea,F1, "ID pezzo:" + {V1} - Fa sì che il testo visualizzato sulla riga dopo il testo "ID pezzo:" sia assegnato a V1. La riga verrà letta dal file aperto utilizzando il puntatore del file F1. Il risultato dell'operazione di lettura (positivo o negativo) verrà memorizzato nella variabile Risultato.

```
File/RLeggiLinea,F1, "Posizione:"+{VARX}+ ", "+{VARY}+  
", "+{VARZ}+ ", "+{VARI}+ ", "+{VARJ}+ ", "+{VARK}
```

```
ASSEGNA/CER1.XYZ = MPNT(VARX, VARY, VARZ)
```

```
ASSEGNA/CER1.IJK = MPNT(VARI, VARJ, VARK)
```

Le precedenti tre righe di comando leggono il testo delimitato da virgole dopo la stringa "Posizione:" e memorizzano i valori nei valori di CER1 X, Y, Z, e I, J, K.

File/LeggiLinea,F1, "Valore # " + varciclo + " : " + {var2} - Fa sì che var2 sia riempita con il testo presente dopo i due punti. In questo esempio, la variabile varciclo non è racchiusa tra parentesi graffe e viene considerata come testo di delimitazione.

Esempio di codice relativo a numeri contenenti zero iniziali

Se il file che si sta leggendo contiene righe di numeri, si noterà che i caratteri precedenti allo zero vengono ignorati. Ad esempio, se una riga contiene il valore 005450, il valore viene letto come un numero e viene restituito il valore 5450, ignorando i due zero precedenti. Questa opzione può essere conservata o annullata.

Si supponga di avere un file di testo creato da un software lettore di codice a barre esterno e contenente due righe di dati:

290291143;582750;0010

291143;5827;0010

È possibile utilizzare un codice semplice come questo per ottenere i valori numerici tra i punti e virgola:



```

ASSEGNA/PRIMO_VALORE=0
ASSEGNA/SECONDO_VALORE=0
ASSEGNA/TERZO_VALORE=0
ASSEGNA/NUMRIGA=1
FPTR =FILE/APRI,D:\TEMP\CODES.TXT,LEGGI
DO/
INRIGA
=FILE/LEGGIRIGA,FPTR,{PRIMO_VALORE}+";"+{SECONDO
_VALORE}+";"+{TERZO_VALORE}
COMMENTO/OPER,NO,"NUMERO RIGA: " + NUMRIGA
,"Primo valore: " + PRIMO_VALORE
,"Secondo Valore: "+SECONDO_VALORE
,"Terzo Valore: "+TERZO_VALORE
UNTIL/INRIGA == "EOF"
FILE/CHIUDI,FPTR,MANTIENI

```

In questo modo, le righe del testo saranno correttamente analizzate e saranno restituiti i valori numeri, ma saranno anche rimossi tutti gli zero precedenti ai valori restituiti. Quindi, la variabile TERZO_VALORE conterrà il valore 10, anziché 0010.

Per conservare i valori degli zero iniziali, è necessario trattare l'intera riga come una stringa e utilizzare invece le funzioni di stringa INDICE, SINISTRA e MEDIO per individuare la posizione dei punti e virgola in una riga di testo e ottenere i valori numerici:



```
FPTR =FILE/APRI,D:\TEMP\CODES.TXT,LEGGI
ASSEGNA/NUMRIGA=1
DO/
STATORIGA =FILE/LEGGIRIGA,FPTR,{STRRIGA}
ASSIGN/LINESTR=STR(LINESTR)
ASSEGNA/PRIMO_INDICE=INDEX (LINESTR,";")
ASSIGN/FIRST_VALUE=STR(LEFT(LINESTR,FIRST_INDEX-
1))
ASSIGN/REMAINSTR=STR(MID(LINESTR,(FIRST_INDEX)))
ASSEGNA/SECONDO_INDICE=INDEX (REMAINSTR,";")
ASSIGN/SECOND_VALUE=STR(LEFT(REMAINSTR,SECOND_INDE
X-1))
ASSIGN/THIRD_VALUE=STR(MID(REMAINSTR,SECOND_INDEX
))
COMMENTO/OPER,NO,"NUMERO RIGA: " + NUMRIGA
,"Primo valore: " + PRIMO_VALORE
,"Secondo Valore: "+SECONDO_VALORE
,"Terzo Valore: "+TERZO_VALORE
ASSEGNA/NUMRIGA=NUMRIGA+1
UNTIL/STATORIGA == "EOF"
FILE/CHIUDI,FPTR,MANTIENI
```

Spiegazione del codice

Questo codice è molto simile a quello descritto più sopra. Saranno elencate di seguito solo le spiegazioni del codice relative alle funzioni di stringa menzionate.

ASSEGNA/PRIMO_INDICE=INDEX (LINESTR,";")

Questa riga individua la posizione del primo punto e virgola sulla riga e lo assegna alla variabile `PRIMO_INDICE`.

ASSIGN/FIRST_VALUE=STR(LEFT(LINESTR,FIRST_INDEX-1))

Questa riga assegna la variabile `PRIMO_VALORE` alla stringa di caratteri fino al primo punto e virgola nella variabile `LINESTR`. `LINESTR` contiene l'intera riga di testo.

ASSIGN/REMAINSTR=STR(MID(LINESTR,(FIRST_INDEX)))

Questa riga assegna la variabile `REMAINSTR` ("remaining string", stringa rimanente) alla stringa di sinistra oltre i caratteri che iniziano alla posizione `PRIMO_INDICE` (la posizione del primo punto e virgola) fino alla fine della riga.

ASSEGNA/SECONDO_INDICE=INDEX (REMAINSTR,";")

Questa riga cerca all'interno della variabile REMAINSTR un altro punto e virgola (il secondo sulla riga) e assegna la posizione alla variabile `SECONDO_INDICE`.

```
ASSIGN/SECONDO_VALUE=STR(LEFT(REMAINSTR,SECONDO_INDEX-1))
```

Questa riga assegna la variabile `SECONDO_VALORE` alla stringa di caratteri fino al primo punto e virgola nella variabile REMAINSTR (il secondo punto e virgola sull'intera riga).

```
ASSIGN/THIRD_VALUE=STR(MID(REMAINSTR,SECONDO_INDEX))
```

Questa riga assegna la variabile `TERZO_VALORE` alla stringa di caratteri che inizia dalla posizione `SECONDO_INDICE` fino alla fine della riga.

Lettura di un blocco di testo da un file

L'opzione del menu **Inserisci | Comando file I/O | Comandi di lettura | Leggi blocco** consente di inserire nella finestra di modifica un comando per la lettura di un blocco di caratteri da un file aperto al momento dell'esecuzione. Il parametro di dimensione indica il numero di caratteri letti.

La sintassi di questo comando nella finestra di modifica è la seguente:



```
<nome_variabile>=File/Leggi_Blocco,<nome_
puntatore_file>,<dimensione>
```

Descrizioni di alcuni dei componenti di questo comando.

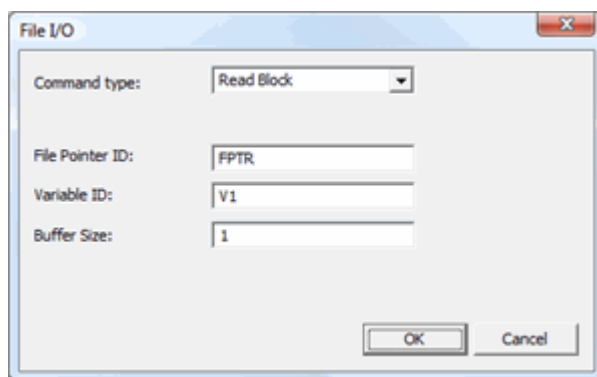
<nome_variabile> - È l'ID della variabile nella quale viene registrato il valore che indica se l'operazione di lettura del blocco è stata eseguita correttamente o meno.

<nome_puntatore_file> - È il nome del puntatore del file specificato al momento dell'apertura del file.

<dimensione> - È il numero di caratteri da leggere.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando di lettura del blocco.
3. Premere F9.



Esempio di codice per il comando Leggi blocco



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Si supponga di disporre di alcuni file di dati esterni contenenti diversi dati relativi al pezzo e che tali file siano definiti dai primi caratteri dei rispettivi nomi. È possibile utilizzare il comando [File/Leggi blocco](#) per leggere unicamente i primi caratteri prima di decidere se leggere ed elaborare tutte le righe. Si prenda in considerazione il codice seguente:

```

C3=COMMENTO/INPUT,Digitare il nome
,del codice del file da ricercare.
ASSEGNA/DIM BLOCCO = LEN(C3.INPUT)
ASSEGNA/CODICE FILE = C3.INPUT
DO/
    C1=COMMENTO/INPUT,Digitare il percorso completo,
    ,il nome e l'estensione del file da
    ,elaborare.
    ,Per uscire dal programma, digitare [E].
    IF/C1.INPUT=="Q" O C1.INPUT=="q"
        COMMENTO/OPER,Si è deciso di uscire. La
        routine ora si chiuderà.
        GOTO/FINE
    END_IF/
    V1=FILE/EXISTS,C1.INPUT
    IF/V1<>0
        COMMENTO/OPER,"Il file di dati [" + C1.INPUT
        + "] esiste. Fare clic su OK per
        continuare."
        FPTR=FILE/OPEN,C1.INPUT,READ
        V2=FILE/LEGGI BLOCCO,FPTR, BLOCKSIZE (DIM
        BLOCCO)
        FILE/CHIUDI,FPTR
        IF/V2<>FILECODE (CODICE FILE)
            COMMENTO/OPER,"Il codice del file della
            variabile [" + V2 + "] non corrisponde"
            ,"al codice del file [" + FILECODE
            (CODICE FILE) + "].
        END_IF/
        COMMENTO/OPER,"Il file [" + C1.INPUT + "]
        corrisponde."
        ,"Il codice del file della variabile [" + V2
        + "] corrisponde"
        ,"al codice del file [" + FILECODE (CODICE
        FILE) + "].
        COMMENTO/OPER,la routine elabora il file.
    END_IF/
    ELSE/
        COMMENTO/OPER,"Il file di dati [" + C1.INPUT
        + "] non esiste. Riprovare usando un file di
        dati esistente."
        GOTO/FINE
    END ELSE/
    UNTIL/V2==FILECODE (CODICE FILE)
    FINE=ETICHETTA/
    ROUTINE/FINE

```



Spiegazione del codice

Parte di questo codice è simile a quello descritto in "Esempio di codice per il comando Leggi carattere" oppure in "Esempio di codice per il comando Leggi riga".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

```
ASSEGNA/DIM BLOCCO = LEN(C3.INPUT)
```

Questa riga crea una variabile definita dall'utente denominata `DIM BLOCCO` che contiene un numero intero uguale al numero di caratteri trovati in `C3.INPUT`. Questo valore costituisce la dimensione del blocco di caratteri da leggere.

```
ASSEGNA/CODICE FILE = C3.INPUT
```

Questa riga crea la variabile `CODICE FILE` e le assegna il valore di `C3.INPUT`.

```
C1=COMMENTO/INPUT
```

Questo commento memorizza il percorso completo immesso dall'utente nella variabile `C1.INPUT`.

```
V1=FILE/EXISTS,C1.INPUT
```

Questa riga verifica l'esistenza del nome del file definito nel commento C1.

```
DO/
```

Questa riga avvia un ciclo DO / UNTIL. Associa il blocco di codice che consente all'utente di specificare un file per la lettura dei dati. L'esecuzione del ciclo proseguirà finché il testo assegnato alla variabile `CODICE FILE` non corrisponde al testo letto dal file.

```
V2=FILE/LEGGI BLOCCO,FPTR, BLOCKSIZE (DIM BLOCCO)
```

Questa riga legge un numero di caratteri equivalente al valore intero contenuto nella variabile `DIM BLOCCO`. Il testo viene quindi memorizzato nella variabile `V2`.

```
IF/V2<>FILECODE (CODICE FILE)
```

Questa riga inizia un blocco di codice IF/END IF che verifica se il testo contenuto nella variabile `V2` corrisponde al testo memorizzato nella variabile `CODICE FILE`. Se i testi coincidono, l'esecuzione della routine continua. Altrimenti, viene visualizzato un messaggio che informa che i due codici non coincidono.

```
UNTIL/V2==FILECODE (CODICE FILE)
```

Questa riga controlla la condizione del ciclo DO/UNTIL per verificare che la variabile **V2** corrisponda al testo contenuto nella variabile **CODICE FILE**. Se l'istruzione viene valutata come falsa, il ciclo **DO** viene eseguito di nuovo, consentendo all'utente di scegliere un nome file diverso. Se l'istruzione viene valutata come vera, il ciclo viene interrotto e viene visualizzato un messaggio indicante che è stata rilevata una corrispondenza. PC-DMIS potrebbe pertanto continuare a leggere ciascuna riga di dati dal file di dati specificato.

Lettura di un testo fino a un delimitatore

L'opzione del menu **Inserisci | Comando file I/O | Comandi di lettura | Leggi fino a** consente di inserire nella finestra di modifica un comando per la lettura durante l'esecuzione di tutto il testo fino a uno dei delimitatori indicati dal file specificato. Il testo letto da questo comando viene inserito nella variabile di destinazione specificata. La lettura del testo viene interrotta quando vengono rilevati i seguenti elementi:

- Delimitatori definiti
- Ritorni a capo
- Caratteri di avanzamento riga

Se viene raggiunta la fine del file, la variabile di destinazione verrà impostata su "EOF" (fine del file).

La sintassi di questo comando nella finestra di modifica è la seguente:



```
<nome_variabile> = FILE/LEGGI  
FINOA,<nome_puntatore_file>,<delimitatori>
```

Descrizioni di alcuni dei componenti di questo comando.

<nome_variabile> - È il nome della variabile di destinazione.

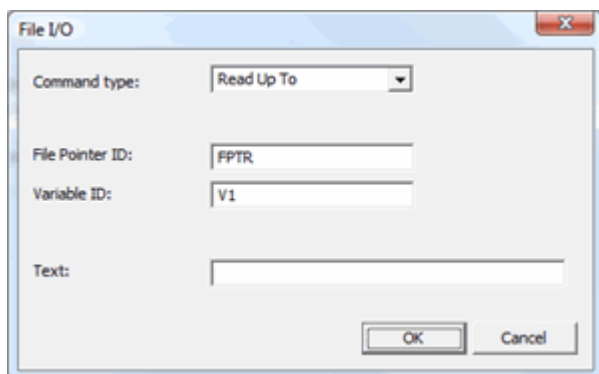
<nome_puntatore_file> - È il nome del puntatore del file specificato al momento dell'apertura del file.

<delimitatori> - È una stringa che contiene zero o più caratteri delimitatori.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

Uso dei file di input e output

1. Selezionare **Finestra di modifica** per aprire la finestra di modifica.
2. Posizionare il cursore sul comando `FILE/LEGGI`.
3. Premere F9. Viene visualizzata la finestra di dialogo **File I/O**.



Una volta visualizzata la finestra di dialogo:

1. Immettere il nome della variabile che riceverà le informazioni di lettura nella casella **ID variabile**.
2. Digitare il nome del puntatore file nella casella **ID puntatore file**.
3. Immettere il delimitatore nella casella **Testo** (assicurandosi di racchiudere il delimitatore scelto tra virgolette).
4. Fare clic sul pulsante **OK**.

Esempio di codice per il comando Leggi fino a



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Si consideri questo esempio in cui nella directory D:\Temp si ha un file di testo denominato "esempio.txt" che contiene nella prima riga:




```
CER1:2.54:CERCHIO
```

Per utilizzare il comando Leggi fino a su questo file, effettuare le seguenti operazioni:

1. Inserire un comando `FILE/APRI` nella finestra di modifica.


- Utilizzare il nome di un puntatore file desiderato per il comando di apertura file. In questo esempio, il nome del puntatore file è "esempio".

Il comando di apertura file è simile a quello riportato di seguito:



```
ESEMPIO      =FILE/APRI,D:\TEMP\ESEMPIO.TXT,LEGGI
```

Ora, usare i comandi di PC-DMIS Leggi fino a, per definire alcune variabili che richiamano segmenti di dati diversi. Questo esempio usa le seguenti variabili che cercano come delimitatore un carattere":" (senza le virgolette).




```
ETICHETTA V   =FILE/LEGGI_FINOA,ESEMPIO,:
VALORE V      =FILE/LEGGI_FINOA,ESEMPIO,:
TIPO V°       =FILE/LEGGI_FINOA,ESEMPIO,:
```

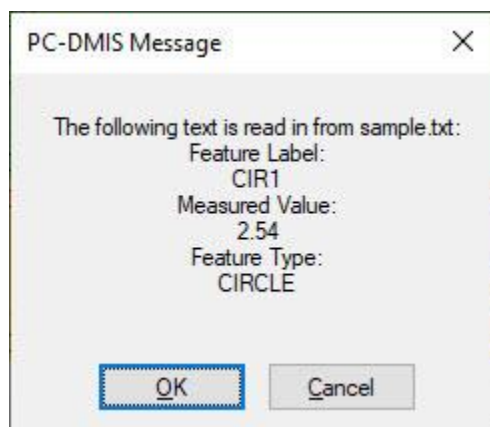
Quindi, quando esegue queste linee di codice, PC-DMIS imposta in queste variabili i seguenti valori:

- ETICHETTA_V = CER1
- VALORE V = 2,54
- TIPO V = CERCHIO

Per visualizzare questa procedura durante l'esecuzione, è possibile usare un commento dell'operatore simile a quello riportato di seguito:



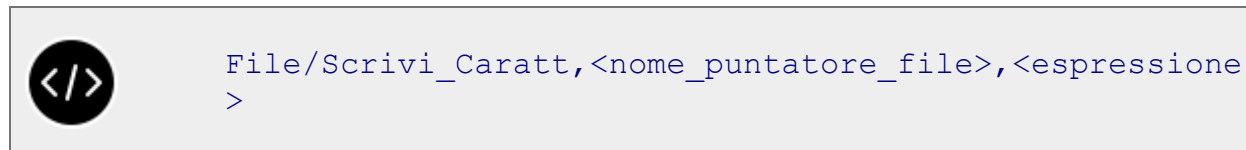
```
COMMENTO/OPER,NO,SCHERMO INTERO=NO,AUTO-
CONTINUA=NO,OVC=NO,
Il seguente testo è letto da sample.txt:
Etichetta elemento:
ETICHETTA_V
Valore misurato:
VALORE_V
Tipo di elemento:
TIPO_V
```



Scrittura di un carattere in un file

L'opzione del menu **Inserisci | Comando I/O file | Comandi di scrittura | Scrivi carattere** consente di inserire nella finestra di modifica un comando che esegue l'output di un singolo carattere su un file nel computer al momento dell'esecuzione.

La sintassi di questo comando nella finestra di modifica è la seguente:



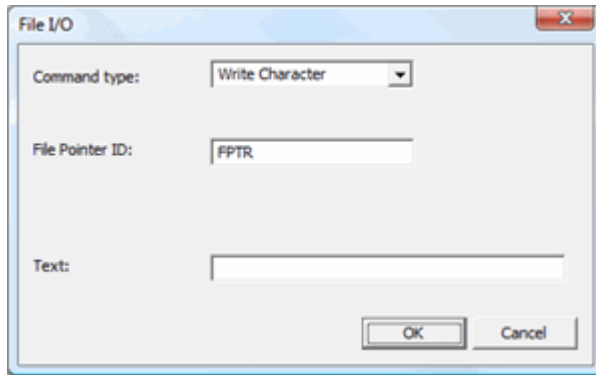
Descrizioni di alcuni dei componenti di questo comando.

<nome_puntatore_file> - È il nome del puntatore del file specificato al momento dell'apertura del file.

<espressione> È il carattere da scrivere nel file. Se l'espressione viene valutata come avente più caratteri, viene scritto soltanto il primo carattere.

Per accedere alla finestra di dialogo associata a questo comando File I/O, procedere come segue.

1. Aprire la finestra di modifica.
2. Posizionare il cursore sul comando di scrittura del carattere.
3. Premere F9.



Esempio di codice per il comando Scrivi carattere



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Questo codice scrive una stringa specificata dall'utente in un file di dati, un carattere alla volta.



```
C1=COMMENTO/INPUT,Digitare il nome del file in cui
scrivere
,(specificare il percorso completo).
FPTR=FILE/OPEN,C1.INPUT,WRITE
C2=COMMENTO/INPUT,Digitare il testo da inviare al
file.
,Tale testo verrà inviato alla stringa un carattere
,alla volta.
ASSEGNA/CONT=0
ASSEGNA/LUNGHEZZA = LEN(C2.INPUT)
DO/
    ASSEGNA/SCRIVIQUESTO = MID(C2.INPUT,CONT,1)
    FILE/SCRIVI CARATTERE,FPTR,WRITETHIS
    (SCRIVIQUESTO)
    ASSEGNA/CONT=CONT + 1
UNTIL/CONT == LUNGH
```

Spiegazione del codice

Parte di questo codice è simile a quello descritto in "Esempio di codice per il comando Leggi carattere" oppure in "Esempio di codice per il comando Leggi riga".

Uso dei file di input e output

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

```
FPTR=FILE/OPEN,C1.INPUT,WRITE
```

Questa riga apre il file specificato nel commento C1, lo rende disponibile per la scrittura e lo assegna al puntatore del file `FPTR`. Tutti i dati contenuti nel file vengono sovrascritti se il puntatore del file è posizionato all'inizio del file di dati.

```
ASSEGNA/CONT=0
```

Questa riga assegna alla variabile `CONT` definita dall'utente il valore zero. Ciò consente di avviare il ciclo per la stampa della stringa un carattere alla volta.

```
ASSEGNA/LUNGHEZZA = LEN(C2.INPUT)
```

Questa riga utilizza la funzione `LEN()` per restituire la lunghezza di una stringa. Questa funzione considera come unico parametro la stringa. Esegue il conteggio dei caratteri nella stringa (inclusi gli spazi) e restituisce il valore intero corrispondente. Questo risultato costituisce il valore della variabile `LUNGHEZZA` definita dall'utente.

```
DO/
```

Questa riga avvia un ciclo `DO / UNTIL`. Il codice tra le istruzioni `DO` e `UNTIL` viene eseguito finché il valore della condizione del ciclo non viene valutata come vera.

```
ASSEGNA/SCRIVIQUESTO = MID(C2.INPUT,CONT,1)
```

Questa riga crea una variabile definita dall'utente denominata `WRITETHIS (SCRIVI QUESTO)`, quindi utilizza la funzione `CENTR()` per restituire un carattere secondario dalla stringa `C2.INPUT` e assegnarla a `WRITETHIS (SCRIVI QUESTO)`.

`MID()` utilizza tre parametri.

- Parametro 1: è la stringa da cui si ottengono i valori. In questo caso viene utilizzato `C2.INPUT`.
- Parametro 2: è la posizione della stringa da cui leggere il carattere. Il primo carattere in una stringa rappresenta la posizione 0, il secondo la posizione 1, il terzo la posizione 2 e così via. In questo caso viene utilizzata la variabile `CONT`.
- Parametro 3: è il numero di caratteri a partire dalla posizione del secondo parametro da leggere. In questo caso, viene utilizzato il valore 1 (il codice esemplificativo proposto scrive un solo carattere alla volta, non è dunque necessario ottenerne di più).

```
FILE/SCRIVI CARATTERE,FPTR,WRITETHIS (SCRIVIQUESTO)
```

Questa riga scrive il carattere memorizzato nella variabile `WRITETHIS` (SCRIVI QUESTO) nel file indicato dal puntatore del file, `FPTR`.

ASSEGNA/CONT=CONT +1

Questa riga legge il valore `CONT` corrente, lo incrementa di un'unità, quindi inserisce di nuovo il nuovo valore in `CONT`.

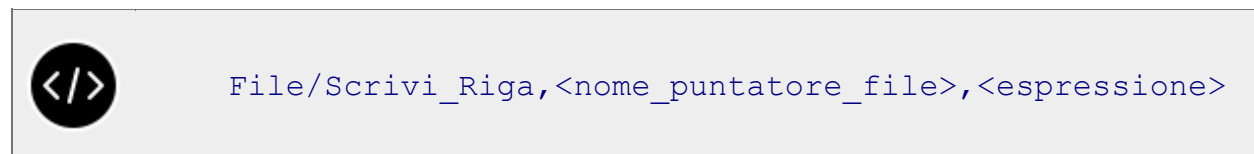
UNTIL/CONT == LUNGH

Questa riga verifica la condizione del ciclo `DO/UNTIL`. In questo caso, il ciclo continua a incrementare la variabile `CONT` finchè non raggiunge lo stesso valore della variabile `LUNGH`. Al termine di questa operazione il ciclo viene interrotto e la routine viene chiusa.

Scrittura di una riga in un file

L'opzione del menu **Inserisci | Comando file I/O | Comandi di scrittura | Scrivi riga** consente di inserire nella finestra di modifica un comando che esegue l'output di una linea di testo su un file nel computer al momento dell'esecuzione. Usare la sintassi dell'espressione per eseguire l'output su un file delle variabili e delle informazioni nella routine di misurazione. Alla fine del testo viene automaticamente aggiunto un carattere di ritorno a capo.

La sintassi di questo comando nella finestra di modifica è la seguente:



Descrizioni di alcuni dei componenti di questo comando.

<nome_puntatore_file> È il nome del puntatore del file specificato al momento dell'apertura del file.

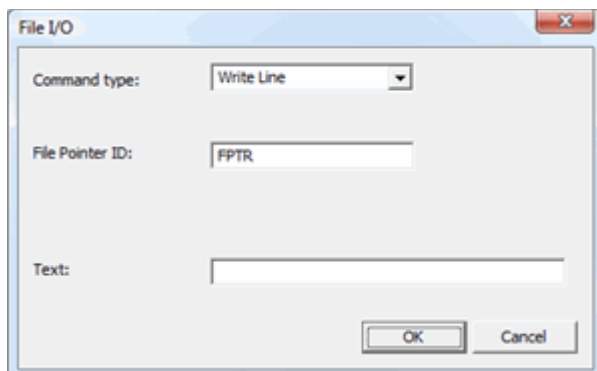
<espressione> - È il testo da scrivere nel file. In questo campo è possibile utilizzare delle espressioni.

Per accedere alla finestra di dialogo associata a questo comando File I/O, procedere come segue.

1. Aprire la finestra di modifica.

Uso dei file di input e output

2. Posizionare il cursore sul comando di scrittura della riga.
3. Premere F9.



Esempio di codice per il comando Scrivi riga



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Si supponga di dover esportare alcuni valori XYZ misurati in un file di dati. Il codice seguente consente di inserire l'etichetta di un elemento e un file di dati, nonché di inviare i dati X,Y e Z relativi a quell'elemento in un file di dati.



```
C1=COMMENTO/INPUT,Digitare l'etichetta dell'elemento
,da utilizzare.
C2=COMMENTO/INPUT,Digitare il nome del file in cui
scrivere
,(specificare il percorso completo).
FPTR=FILE/OPEN,C2.INPUT,APPEND
ASSEGNA/NOME ELEM = C1.INPUT
ASSEGNA/TUTTI I VALORI =NOME ELEM.X+", "+NOME
ELEM.Y+", "+NOME ELEME.Z
COMMENTO/OPER,"Il testo da scrivere è: "+TUTTI I
VALORI
FILE/SCRIVI RIGA,FPTR,ALLVALS (TUTTI VALORI)
FILE/CHIUDI,FPTR
```

Spiegazione del codice

Parte di questo codice è simile a quello descritto in "Esempio di codice per il comando Leggi carattere" oppure in "Esempio di codice per il comando Leggi riga".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

```
FPTR=FILE/OPEN,C2.INPUT,APPEND
```

Questa riga apre il file specificato nel commento C2, lo rende disponibile per la scrittura e lo assegna al puntatore del file `FPTR`. Se invece si desidera modificare il parametro `AGGIUNGI` in `SCRIVI`, il contenuto esistente nel file di dati viene sovrascritto.

```
ASSEGNA/NOME ELEM = C1.INPUT
```

Questa riga assegna la stringa dell'etichetta dell'elemento da `C1.INPUT` alla variabile definita dall'utente, `NOME ELEM`.

```
ASSEGNA/TUTTI I VALORI=NOME ELEM.X+", "+NOME ELEM.Y+", "+ NOME  
ELEM.Z
```

Questa riga assegna la variabile definita dall'utente `TUTTI I VALORI` al valore di `NOME ELEM.X`, `NOME ELEM.Y`, `NOME ELEM.Z`, in altre parole contiene i valori X, Y e Z dell'etichetta dell'elemento immesso nel comando di input C1.

```
FILE/SCRIVI RIGA,FPTR,ALLVALS (TUTTI VALORI)
```

Questa riga scrive i valori contenuti in `TUTTI I VALORI` nel file specificato dal puntatore del file, `FPTR`.

Scrittura di un blocco di testo in un file

L'opzione del menu **Inserisci | Comando file I/O | Comandi di scrittura | Scrivi blocco** consente di inserire nella finestra di modifica un comando che esegue l'output di un blocco di testo su un file nel computer al momento dell'esecuzione. Usare la sintassi dell'espressione per eseguire l'output su un file delle variabili e delle informazioni nella routine di misurazione. Diversamente dal comando di scrittura di una riga, la scrittura di un blocco *non* aggiunge un ritorno a capo alla fine.

La sintassi di questo comando nella finestra di modifica è la seguente:



```
File/Scrivi_blocco,<nome_puntatore_file>,<espressione>
```

Descrizioni di alcuni dei componenti di questo comando.

<nome_puntatore_file> È il nome del puntatore del file specificato al momento dell'apertura del file.

<espressione> - È il testo da scrivere nel file. In questo campo è possibile utilizzare delle espressioni.



Diversamente dal comando di scrittura di una riga, la scrittura di un blocco *non* aggiunge un ritorno a capo alla fine. Tuttavia, se si desidera posizionare del testo in una nuova riga all'interno del blocco di testo, è possibile inserire manualmente un ritorno a capo e un avanzamento riga utilizzando il codice CHR(10) fuori dalla stringa racchiusa tra virgolette, come in questo esempio: FILE/WRITEBLOCK,FPTR, "CHR(10) inserisce il testo...

```
FILE/SCRIVIBLOCCO,FPTR, "CHR(10) inserisce il testo... " +  
CHR(10) + " ...su una nuova riga."
```

Nel file di output sarà restituito il seguente risultato:

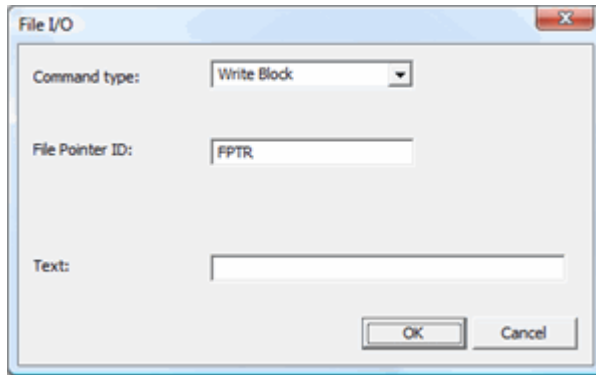
```
CHR(10) inserisce il testo...
```

```
...su una nuova riga.
```

Tenere presente che se CHR(10) è racchiuso tra virgolette il testo effettivo di CHR(10) viene inviato al file.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando Scrivi blocco.
3. Premere F9.



Esempio di codice per il comando Scrivi blocco



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Il codice seguente scrive tutte le informazioni inserite in un commento di input, aggiungendo i due punti (:) come delimitatore.



```
C1=COMMENTO/INPUT,Digitare una stringa. PC-DMIS
aggiunge i due punti come delimitatore e scrive la
stringa nel file desiderato.
C2=COMMENTO/INPUT,Digitare il nome del file in cui
scrivere
,(specificare il percorso completo).
FPTR=FILE/OPEN,C2.INPUT,APPEND
ASSEGNA/SCRIVIQUESTO = C1.INPUT + ":"
COMMENTO/OPER,"Il testo da scrivere è:
"+SCRIVIQUESTO
FILE/SCRIVI RIGA,FPTR,WRITETHIS (SCRIVIQUESTO)
FILE/CHIUDI,FPTR
```

Spiegazione del codice

Parte di questo codice è simile a quello descritto in "Esempio di codice per il comando Leggi carattere" oppure in "Esempio di codice per il comando Leggi riga".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

FPTR=FILE/OPEN,C2.INPUT,APPEND

Uso dei file di input e output

Questa riga apre il file specificato nel commento C2, lo rende disponibile per la scrittura e lo assegna al puntatore del file `FTPTR`.

```
ASSEGNA/SCRIVIQUESTO = C1.INPUT + ":"
```

Questa riga aggiunge due punti (:) al testo contenuto in `C1.INPUT` e assegna la nuova stringa alla variabile definita dall'utente `WRITETHIS` (SCRIVI QUESTO).

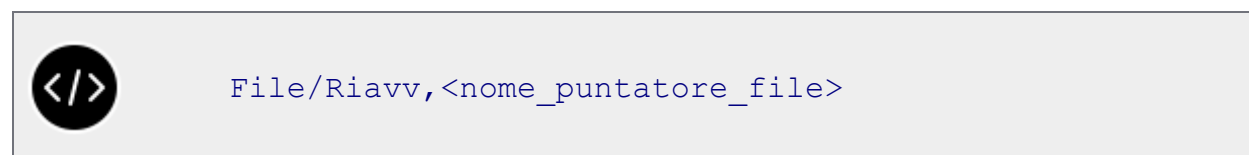
```
FILE/SCRIVI RIGA,FTPTR,WRITETHIS (SCRIVIQUESTO)
```

Questa riga scrive i valori contenuti in `SCRIVIQUESTO` nel file specificato dal puntatore del file, `FTPTR`. In un secondo momento sarà possibile leggere il testo dal file utilizzando i due punti come delimitatore.

Posizionamento del puntatore all'inizio di un file

L'opzione di menu **Inserisci | Comando di I/O file | Comandi di posizione | Riposiziona all'inizio** consente di inserire nella finestra di modifica un comando per posizionare il puntatore del file all'inizio del flusso.

La sintassi di questo comando nella finestra di modifica è la seguente:

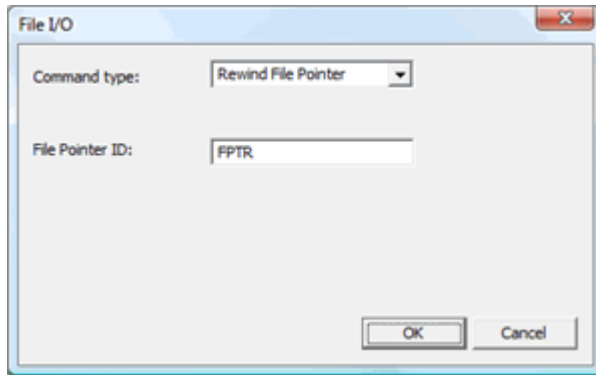


Descrizioni di alcuni dei componenti di questo comando.

<nome_puntatore_file> - È il nome del puntatore da riposizionare all'inizio del file.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sull'opzione Riposiziona all'inizio.
3. Premere F9.



Esempio di codice per il comando di riposizionamento all'inizio



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Nell'esempio proposto il codice legge i dati da un file esterno una riga alla volta. Al termine di ogni riga, è possibile iniziare di nuovo e leggere il file dall'inizio. Di seguito viene descritto l'uso del comando `FILE/RIAVV..`



```

C1=COMMENTO/INPUT,Digitare il file di lettura.
,(specificare il percorso completo)
V1=FILE/EXISTS,C1.INPUT
IF/V1<>0
  DO/
    FPTR=FILE/OPEN,C1.INPUT,READ
    C2=COMMENTO/SÌNO,Si desidera effettuare la
    lettura dall'inizio?
    IF/C2.INPUT == "YES"
      FILE/RIAVV,FPTR
    END_IF/
    V2=FILE/LEGGI RIGA,FPTR,{RIGA}
    COMMENTO/OPER,"La riga corrente è: " + RIGA
  UNTIL/V2=="EOF"
END_IF/
FILE/CHIUDI,FPTR
COMMENTO/OPER,Uscita dalla routine.

```

Spiegazione del codice

Parte di questo codice è simile a quello descritto in "Esempio di codice per il comando Leggi carattere" oppure in "Esempio di codice per il comando Leggi riga".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

C2=COMMENTO/SÍNO

Questa riga richiede se si desidera iniziare la lettura del file dall'inizio. Memorizza la risposta SÍ/NO nella variabile, `C2.INPUT`.

IF/C2.INPUT == "YES"

Questa riga inizia un blocco IF/END IF. Essa verifica se la condizione di `C2.INPUT` ha il valore SÌ. Se la condizione viene valutata vera, PC-DMIS esegue le righe in base all'istruzione `IF`. Se la condizione viene valutata falsa, PC-DMIS esegue il codice in base all'istruzione `END IF`.

FILE/RIAVV,FPTR

Questa riga riposiziona il puntatore all'inizio del file di dati.

END_IF/

Questa riga termina il blocco di codice IF / END IF.

Salvataggio della posizione corrente del puntatore di un file

L'opzione di menu **Inserisci | Comando I/O file | Comandi di posizione | Salva posizione file** inserisce un comando nella finestra di modifica che salva la posizione corrente di un puntatore file all'interno del flusso di file. È possibile richiamare successivamente la posizione salvata utilizzando l'apposito comando.

La sintassi di questo comando nella finestra di modifica è la seguente:



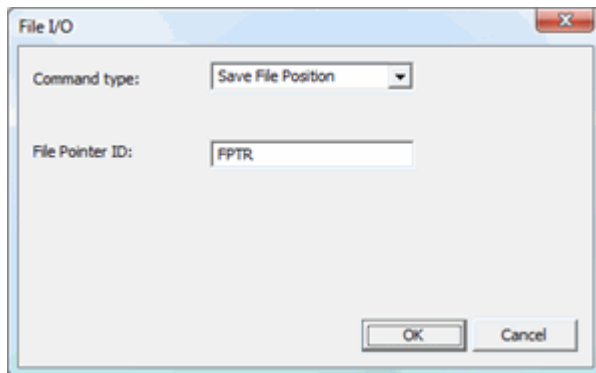
File/Salva_Pos,<nome_puntatore_file>

Descrizioni di alcuni dei componenti di questo comando.

<nome_puntatore_file> - È il nome del puntatore del file del quale si desidera salvare la posizione.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando di salvataggio della posizione del file.
3. Premere F9.



Esempio di codice per il comando Salva posizione file



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Nell'esempio proposto il codice legge i dati da un file esterno una riga alla volta. Dopo ogni riga, è possibile salvare la posizione del file, per richiamarla in un secondo momento. Di seguito viene descritto l'uso del comando `FILE/SALVA POS..`



```
C1=COMMENTO/INPUT,Digitare il file di lettura.
,(specificare il percorso completo)
V1=FILE/EXISTS,C1.INPUT
IF/V1<>0
    DO/
        FPTR=FILE/OPEN,C1.INPUT,READ
        C2=COMMENTO/SÌNO,Si desidera salvare il file
        e richiamarlo in seguito? Il ciclo termina.
        IF/C2.INPUT == "YES"
            FILE/SALVA POS,FPTR
            GOTO/QUITciclo (INTERROMPI ciclo)
        END_IF/
        V2=FILE/LEGGI RIGA,FPTR,{RIGA}
        COMMENTO/OPER,"La riga corrente è: " + RIGA
    UNTIL/V2=="EOF"
END_IF/
FILE/CHIUDI,FPTR
QUITciclo=ETICHETTA/
COMMENTO/OPER,Lettura interrotta.
ROUTINE/FINE
```

Spiegazione del codice

Questo codice è simile a quello descritto in "Esempio di codice per il comando di riposizionamento all'inizio".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

C2=COMMENTO/SÌNO

Questa riga richiede se si desidera memorizzare la posizione del file corrente e uscire dal ciclo. Memorizza la risposta SÌ/NO nella variabile, `C2.INPUT`.

FILE/SALVA POS,FPTR

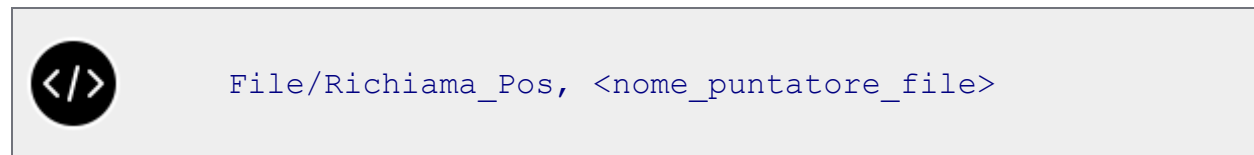
Questa riga memorizza la posizione del puntatore del file nel flusso del file.

Quando in una routine di misurazione si apre lo stesso file con un puntatore che ha lo stesso nome, è possibile richiamare la posizione salvata del file e riprendere la lettura dal punto in cui era stata interrotta. Per continuare questo esempio, vedere l'argomento "Esempio di codice per il comando Richiama posizione file".

Richiamo della posizione salvata del puntatore di un file

L'opzione del menu **Inserisci | Comando file I/O | Comandi di posizione | Richiama posizione file** consente di inserire nella finestra di modifica un comando che richiama la posizione di un file salvata in precedenza. Usare il comando Salva posizione file per salvare una posizione in un file aperto.

La sintassi di questo comando nella finestra di modifica è la seguente:

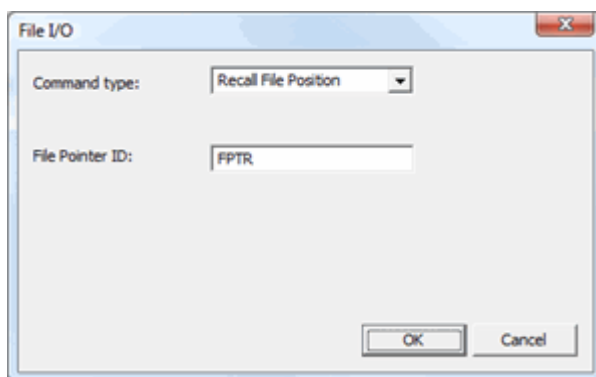


Descrizioni di alcuni dei componenti di questo comando.

<nome_puntatore_file> - È il nome del puntatore del file del quale si desidera richiamare la posizione.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando di richiamo della posizione del file.
3. Premere F9.



Esempio di codice per il comando Richiama posizione file



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Questo esempio apre un file precedentemente chiuso, usa un puntatore del file precedente e richiama la posizione salvata del puntatore del file memorizzato. Quindi, legge i dati da quella posizione. Questo codice mostra l'uso del comando `FILE/SALVA POS..` Segue l'esempio di codice mostrato nell'argomento "Esempio di codice per il comando Salva posizione file".



```
COMMENTO/OPER,la routine richiamerà ora la posizione
salvata del file.
FPTR=FILE/OPEN,C1.INPUT,READ
FILE/RIAVV,FPTR
COMMENTO/OPER,Per verificare se il file è stato
riavvolto.
,La prima riga viene letta per verificare l'avvenuto
riavvolgimento.
V3=FILE/LEGGI RIGA,FPTR,{RIGA}
COMMENTO/OPER,"La prima riga è:
,RIGA
FILE/RIAVV,FPTR
FILE/RICHIAMA POS,FPTR
COMMENTO/OPER,La posizione del file precedentemente
salvata è stata richiamata.
,Vengono ora stampati i dati sulla riga nella
posizione salvata.
V4=FILE/LEGGI RIGA,FPTR,{SALVATA}
COMMENTO/OPER,Il testo nella posizione salvata è:
,SALVATA
```

Spiegazione del codice

Questo codice è simile a quello descritto in "Esempio di codice per il comando di riposizionamento all'inizio".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

`FILE/RICHIAMA_POS,FPTR`

Questa riga richiama la posizione salvata del puntatore del file nel flusso per il puntatore denominato `FPTR`.

```
V4=FILE/LEGGI RIGA,FPTR,{SALVATA}
```

Questa riga legge la riga successiva alla posizione salvata del puntatore del file e la assegna alla variabile definita dall'utente `SALVATA`. La variabile viene quindi stampata accanto al commento dell'operatore.

Copia di un file

L'opzione del menu **Inserisci | Comando di I/O file| Copia file** consente di inserire nella finestra di modifica un comando che causa la copiatura di un file durante l'esecuzione.

La sintassi di questo comando nella finestra di modifica è la seguente:



```
File/Copia,<nome_file_origine>,<nome_file_
destinazione>,<modalità_sostituzione>
```

Descrizioni di alcuni dei componenti di questo comando.

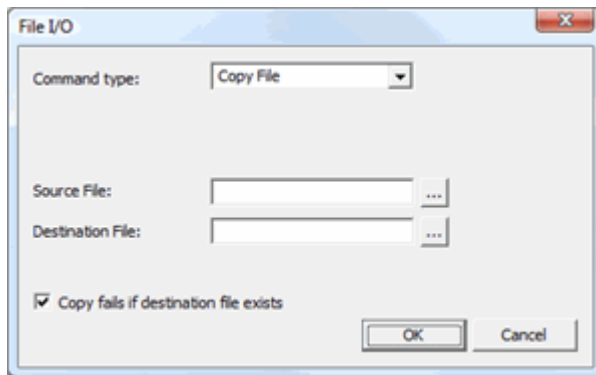
<nome_file_origine> - È il nome del file di origine (file del quale viene eseguita la copia).

<nome_file_destinazione> - È il nome del file di destinazione (file sul quale viene eseguita la copia).

<modalità_sostituzione> - È l'operazione da effettuare se il file di destinazione esiste già. Se la destinazione esiste le due modalità sono sovrascritte e non hanno effetto.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando Copia file.
3. Premere F9.



Esempio di codice per il comando Sposta file



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Il codice seguente richiede il nome di un file del quale eseguire la copia, nonché di una directory e di un file di destinazione nei quali eseguire la copia.



```

C1=COMMENTO/INPUT,Inserire un file di cui si
desidera eseguire la copia.
,Specificare il percorso completo del file.
C2=COMMENTO/INPUT,Digitare il nome del file di
destinazione.
,Specificare il percorso completo del file.
V1=FILE/EXISTS,C1.INPUT
IF/V1<>0
    COMMENTO/OPER,File da copiare esistente. Inizio
    della copia.
    FILE/COPY,C1.INPUT,C2.INPUT,FAIL_IF_DEST_EXISTS
    V2=FILE/EXISTS,C2.INPUT
    IF/V2==0
        COMMENT/OPER,"Il file non esiste in: " +
        C2.INPUT
        ,Fine copia.
        ROUTINE/FINE
    END_IF/
ELSE/
    COMMENTO/OPER,Copia file eseguita.
    ROUTINE/FINE
END ELSE/
END_IF/
COMMENT/OPER,File da copiare inesistente.

```

Spiegazione del codice

Parte di questo codice è simile a quello descritto in "Esempio di codice per il comando Leggi carattere" oppure in "Esempio di codice per il comando Leggi riga".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

C1=COMMENTO/INPUT

Questa riga legge il percorso completo del file da copiare e lo inserisce nella variabile `C1.INPUT`.

C2=COMMENTO/INPUT

Questa riga legge il percorso completo del file di destinazione e lo inserisce nella variabile `C2.INPUT`.

FILE/COPY,C1.INPUT,C2.INPUT,FAIL_IF_DEST_EXISTS

Questa riga copia il file originale in un file di destinazione. Questo comando utilizza tre parametri.

- Il parametro 1 è `C1.INPUT`. Si tratta del percorso completo del file da copiare.
- Il parametro 2 è `C2.INPUT` o il percorso completo del file di destinazione.
- Il parametro 3, in questo caso specifico, annulla la procedura FILE/COPIA se rileva la presenza di un file con lo stesso nome del file di destinazione. È possibile impostare tale parametro in modo che sovrascriva i file con lo stesso nome esistenti.

Comandi nella modalità di comando dopo i commenti

Spostamento di un file

L'opzione del menu **Inserisci | Comando di I/O file | Sposta file** consente di inserire nella finestra di modifica un comando che causa lo spostamento di un file durante l'esecuzione.

La sintassi di questo comando nella finestra di modifica è la seguente:



```
File/Sposta,<nome_file_precedente>,<nome_nuovo_file>
```

Descrizioni di alcuni dei componenti di questo comando.

<nome_file_precedente> - È il nome e la posizione del file.

<nome_nuovo_file> - È il nome e la posizione del nuovo file.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando Muovi file.
3. Premere F9.



Esempio di codice per lo spostamento di un file



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Il codice seguente chiede il nome del file da spostare, nonché la directory e il nome del file verso cui eseguire lo spostamento. Quindi sposta il file.



```
C1=COMMENTO/INPUT,Digitare un file da spostare.
,Specificare il percorso completo del file.
C2=COMMENTO/INPUT,Digitare il nome del file di
destinazione.
,Specificare il percorso completo del file.
V1=FILE/EXISTS,C1.INPUT
IF/V1<>0
    COMMENTO/OPER,File da spostare esistente. Inizio
    dello spostamento del file.
    FILE/MOVE,C1.INPUT,C2.INPUT
    V2=FILE/EXISTS,C2.INPUT
    IF/V2==0
        COMMENT/OPER,"Il file non esiste in: " +
        C2.INPUT
        ,Lo spostamento non è stato eseguito in modo
        adeguato.
        ROUTINE/FINE
    END_IF/
ELSE/
    COMMENTO/OPER,Spostamento file eseguito.
    ROUTINE/FINE
END ELSE/
END_IF/
COMMENTO/OPER,File originale inesistente. Ripetere
l'operazione.
```

Spiegazione del codice

Gran parte di questo codice è simile a quanto descritto in "Esempio di codice per file".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

FILE/MOVE,C1.INPUT,C2.INPUT

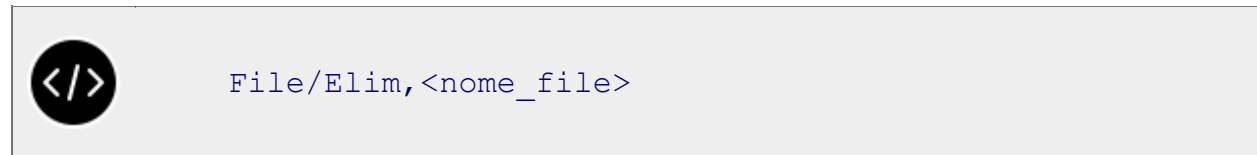
Questa riga copia il file originale in un file di destinazione. Questo comando utilizza due parametri.

- Il parametro 1 è `C1.INPUT`. Si tratta del percorso completo del file da spostare.
- Il parametro 2 è `C2.INPUT` o il percorso completo del file di destinazione.

Eliminazione di un file

L'opzione del menu **Inserisci | Comando di I/O file | Elimina file** consente di inserire nella finestra di modifica un comando la cui esecuzione elimina un file.

La sintassi di questo comando nella finestra di modifica è la seguente:

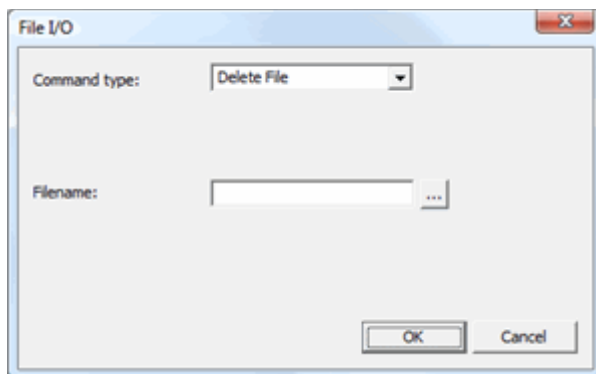


Descrizioni di alcuni dei componenti di questo comando.

<nome_file> - È il nome del file da eliminare.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando Elimina file.
3. Premere F9.



Esempio di codice per il comando Elimina file



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Il codice seguente richiede il nome di un file, quindi lo elimina.



```
C1=COMMENTO/INPUT,Digitare un file da eliminare.
,Specificare il percorso completo del file.
V1=FILE/EXISTS,C1.INPUT
IF/V1<>0
    COMMENTO/OPER,File esistente. Pronto per
    l'eliminazione.
    FILE/DELETE,C1.INPUT
    V2=FILE/ESISTE,
    IF/V2==0
        COMMENTO/OPER,Eliminazione del file
        eseguita.
        ROUTINE/FINE
    END_IF/
ELSE/
    COMMENTO/OPER,File ancora esistente.
    ROUTINE/FINE
END ELSE/
END_IF/
COMMENTO/OPER,File da eliminare inesistente.
Scegliere un file esistente.
```

Spiegazione del codice

Gran parte di questo codice è simile a quanto descritto in "Esempio di codice per lo spostamento di un file".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

FILE/ELIMINA,C1.INPUT Questa riga elimina il file specificato. Questo comando utilizza un unico parametro, ovvero il nome del file da eliminare. In questo caso, C1.INPUT.

Verifica dell'esistenza di un file

L'opzione di menu **Inserisci | Comando di I/O file | File esistente** consente di inserire nella finestra di modifica un comando che verifica l'esistenza di un file durante l'esecuzione, inserendo il risultato nella variabile indicata.

La sintassi di questo comando nella finestra di modifica è la seguente:



```
<nome_variabile> = File/Esistente,<nome_file>
```

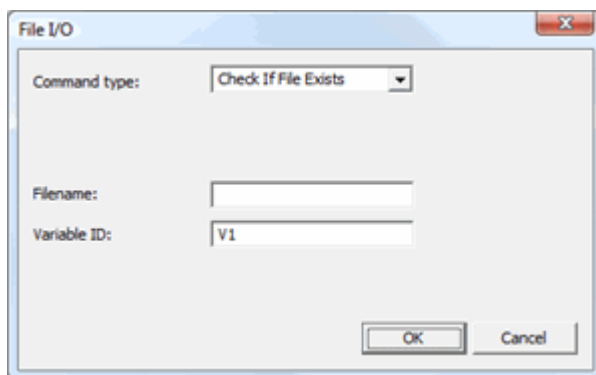
Descrizioni di alcuni dei componenti di questo comando.

<nome_variabile> - È il nome della variabile impostata in base al risultato della verifica eseguita. La variabile viene impostata su 1 se il file esiste e su 0 se il file non esiste.

<nome_file> - È il nome del file del quale viene verificata l'esistenza sul disco.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando File esistente.
3. Premere F9.



Esempio di codice per il comando File esistente



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Il codice seguente richiede il nome di un file, quindi ne verifica l'esistenza.



```
C1=COMMENTO/INPUT,Digitare un file da controllare.  
V1=FILE/EXISTS,C1.INPUT  
IF/V1<>0  
    COMMENTO/OPER,File esistente.  
END_IF/  
ELSE/  
    COMMENTO/OPER,Il file none esiste  
END ELSE/
```

Spiegazione del codice

Parte di questo codice è simile a quello descritto in "Esempio di codice per il comando Leggi carattere" oppure in "Esempio di codice per il comando Leggi riga".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

V1=FILE/EXISTS,C1.INPUT

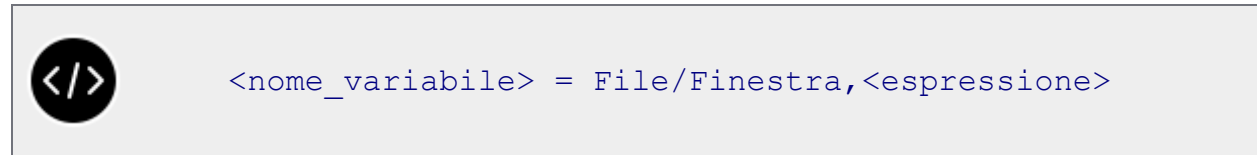
Questa riga verifica l'esistenza del file specificato. Per fare in modo che questo codice funzioni, il file deve trovarsi nella stessa directory di PC-DMIS; in caso contrario la riga contenente il file deve includere il percorso completo del file. **V1** riceve il risultato del controllo del file. Se il file esiste viene restituito un valore diverso da zero, in caso contrario viene restituito un valore uguale a zero.

Visualizzazione della finestra di dialogo di un file

L'opzione del menu **Inserisci | Comandi di I/O file | Finestra di dialogo File** inserisce nella finestra di modifica un comando che visualizza una finestra di dialogo **Apri** durante l'esecuzione. Questa permette all'operatore di scegliere il nome di un file

durante l'esecuzione. Il nome scelto per il file viene così memorizzato nella variabile specificata.

La sintassi di questo comando nella finestra di modifica è la seguente:



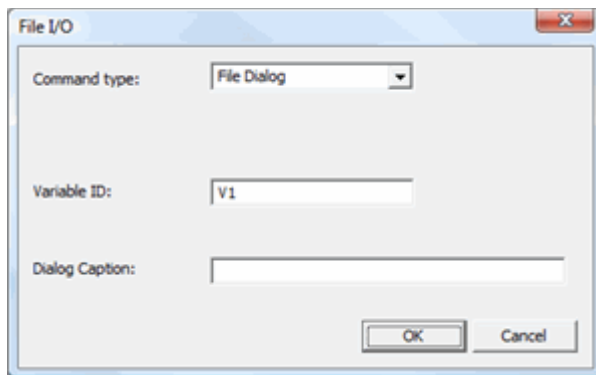
Descrizioni di alcuni dei componenti di questo comando.

<nome_variabile> - È il nome della variabile che conterrà il nome del file scelto dall'utente nella finestra di dialogo File.

<espressione> - Questo è il testo che verrà visualizzato sulla barra del titolo della finestra di dialogo del file.

Per accedere alla finestra di dialogo associata all'opzione File I/O, effettuare le seguenti operazioni:

1. Aprire la finestra di Modifica.
2. Posizionare il cursore sul comando Finestra di dialogo file.
3. Premere F9.



Esempio di codice per il comando Finestra file



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Il codice seguente sottomete una finestra di dialogo che consente di selezionare il file da eliminare.



```
V1=FILE/FINESTRA,Scegliere un file da eliminare.
V1=FILE/ESISTE,V1
IF/V2<>0
    COMMENTO/OPER,File esistente. Pronto per
    l'eliminazione.
    FILE/ELIMINA,V1
    V3=FILE/ESISTE,
    IF/V3==0
        COMMENTO/OPER,Eliminazione del file
        eseguita.
        ROUTINE/FINE
    END_IF/
ELSE/
    COMMENTO/OPER,File ancora esistente.
    ROUTINE/FINE
END ELSE/
END_IF/
COMMENTO/OPER,File da eliminare inesistente.
Scegliere un file esistente.
```

Parte di questo codice è simile a quello descritto in "Esempio di codice per il comando Leggi carattere" oppure in "Esempio di codice per il comando Leggi riga".

Le spiegazioni riportate di seguito sono specifiche per questo esempio.

V1=FILE/Finestra,Scegliere un file da eliminare

Questa riga visualizza una finestra di dialogo nel cui titolo viene indicato di scegliere un file da eliminare. Quando si individua il file e si fa clic su **Apri**, PC-DMIS restituisce V1, ovvero il percorso completo del file selezionato. La rimanente parte della routine elimina il file selezionato.

Controllo di fine file o fine linea

PC-DMIS consente di eseguire il controllo di Fine file utilizzando le funzioni `EOF` o `EOL` inserite in una condizione.

`EOF` è l'acronimo di END OF FILE (FINE DEL FILE). Questa funzione impiega un puntatore a file di tipo stringa. Il quale, inserito opportunamente in una condizione, verifica se il puntatore a file abbia raggiunto la fine del file specificato. In tal caso, la funzione restituisce TRUE.

`EOL` è l'acronimo di END OF LINE (fine linea). Questa funzione impiega un puntatore a file di tipo stringa. Quando è inserita opportunamente in una istruzione condizionale, verifica se il puntatore a file abbia raggiunto la fine di una linea. In tal caso, la funzione restituisce TRUE. Funziona particolarmente bene all'interno di un ciclo.

La sintassi di questo comando nella finestra di modifica è la seguente:

```
EOF(<puntatorefile>) o EOL(<puntatorefile>)
```

Descrizioni di alcuni dei componenti di questo comando.

<puntatore_file> - Questo è il nome del puntatore al file da verificare.

Esempio di codice per EOF e EOL



L'esempio di codice riportato di seguito deve essere immesso in modalità Comando della finestra di modifica e non all'interno della finestra di dialogo **I/O file**.

Il codice seguente apre test.txt e legge il file. Finché il carattere di fine file non viene raggiunto (indicato dal codice `WHILE/!EOF`), PC-DMIS legge il file, carattere per carattere, assegnando ciascun carattere a V1.

Se PC-DMIS raggiunge la fine di una riga del file, visualizza l'ultimo carattere in quella riga.

Questo si ripete fino a quando PC-DMIS raggiunge la fine del file. PC-DMIS a questo punto visualizza il testo "Raggiuntala fine del file...".

Uso dei file di input e output



```
FPTR=FILE/APRI,D:\temp\test.txt,LEGGI
WHILE/!EOF("FPTR")
V1=FILE/LEGGI CARATTERE,FPTR
IF/EOL("FPTR")
COMMENTO/OPER,NO,"Fine della riga raggiunta.
L'ultimo carattere è:"
,V1
END_IF/
END_WHILE/
COMMENTO/OPER,NO,"Fine del file raggiunta..."
```