Table des matières

Utilisation d'entrées/sorties de fichier 1
Utilisation d'entrées / sorties de fichier : Introduction 1
Compréhension des concepts d'E/S de fichier de base 2
Utilisation de la boîte de dialogue Fichier E/S 3
Ouverture d'un fichier en lecture ou en écriture 4
Fermeture d'un fichier ouvert après lecture ou écriture 5
Lecture d'un caractère dans un fichier 6
Lecture d'une ligne dans un fichier 8
Lecture d'un bloc de texte dans un fichier 13
Lecture d'un texte jusqu'à un délimiteur 16
Écriture d'un caractère dans un fichier 18
Écriture d'une ligne dans un fichier 20
Écriture d'un bloc de texte dans un fichier 22
Positionnement d'un pointeur de fichier au début d'un fichier 24
Enregistrement de la position courante d'un pointeur de fichier 26
Rappel d'une position de pointeur de fichier enregistrée 28
Copie d'un fichier 29
Déplacement d'un fichier 31
Suppression d'un fichier 33
Vérification de l'existence d'un fichier 34
Affichage d'une boîte de dialogue de fichier 36
Vérification de la fin d'un fichier ou d'une ligne 37

Utilisation d'entrées/sorties de fichier

Utilisation d'entrées / sorties de fichier : Introduction

Ce chapitre explique comment lire et écrire des informations dans vos routines de mesure. Les options de menu disponibles permettent d'ouvrir des fichiers en mode lecture ou écriture. Des données peuvent ensuite être lues ou écrites dans ces fichiers. Les commandes de fichier E/S permettent de lire des données de fichiers externes qui seront utilisées dans une routine de mesure. De plus, des informations de mesure et de tolérance peuvent être écrites dans les fichiers à l'aide de ces commandes. Vous pouvez également effectuer d'autres opérations sur les fichiers à l'aide de ces commandes.

Ce chapitre détaille ces opérations d'E/S sur les fichiers et comprend des exemples fonctionnels pour chacune d'entre elles. Ces exemples utilisent des notions décrites dans les chapitres « Branchement à l'aide du contrôle de flux » et « Utilisation d'expressions et de variables ».

Remarque : Lorsque le bouclage ou branchement apparaît dans les exemples de code, la mise en retrait sert à améliorer la lisibilité des instructions applicables à chaque condition. Dans le code affiché dans la fenêtre Édition, vous ne voyez apparaître aucune mise en retrait.

Les rubriques principales décrites dans cette section sont :

- Compréhension des concepts d'E/S de fichier de base
- Utilisation de la boîte de dialogue Fichier E/S
- Ouverture d'un fichier en lecture ou en écriture
- Fermeture d'un fichier ouvert après lecture ou écriture
- Lecture d'un caractère dans un fichier
- Lecture d'une ligne dans un fichier
- Lecture d'un bloc de texte dans un fichier
- Lecture d'un texte jusqu'à un délimiteur
- Écriture d'un caractère dans un fichier
- Écriture d'une ligne dans un fichier
- Écriture d'un bloc de texte dans un fichier
- Positionnement d'un pointeur de fichier au début d'un fichier
- Enregistrement de la position courante d'un pointeur de fichier
- Rappel d'une position de pointeur de fichier enregistrée
- Copie d'un fichier

- Déplacement d'un fichier
- Suppression d'un fichier
- Vérification de l'existence d'un fichier
- Affichage d'une boîte de dialogue de fichier
- Vérification de la fin d'un fichier ou d'une ligne

Compréhension des concepts d'E/S de fichier de base

Vérification de l'existence d'un fichier :
pour toutes les opérations d'E/S d'un fichier, vous voulez probablement vérifier d'abord que ce fichier existe bien. Ceci peut être effectué dans une boucle IF / THEN pour pouvoir, si la vérification échoue, en avertir l'utilisateur. Quand vous écrivez dans un fichier, vous devez d'abord créer ce fichier dans l'environnement Windows.

Voir « Vérification de l'existence d'un fichier ».

Ouverture et fermeture de fichiers :
avant d'utiliser les opérations de lecture et d'écriture dans des fichiers, vous devez les ouvrir dans vos processus système. Pour ce faire, affectez le fichier à une variable appelée pointeur de fichier. Lors de l'ouverture d'un fichier, vous pouvez spécifier si celui-ci est ouvert pour lecture, écriture (écrasement) ou ajout. Une fois ouvert, vous pouvez lire ou écrire dedans. Lorsque vous avez terminé de travailler sur un fichier, vous devez en fermer le pointeur ; cette opération ferme le fichier et permet à un autre processus système d'y accéder. Vous ne pouvez pas ouvrir des fichiers déjà ouverts par un autre processus.

Voir « Ouverture d'un fichier en lecture ou en écriture » et «Fermeture d'un fichier ouvert après lecture ou écriture ».

Pointeurs et positions de fichiers :
Les pointeurs de fichiers sont des variables qui pointent un fichier. Ils mémorisent l'emplacement et le nom du fichier ouvert et servent à lire ou à écrire dans celui-ci. Une fois le fichier ouvert et affecté à un pointeur de fichier, le pointeur se comporte comme un curseur dans un traitement de texte. Il indique votre position de lecture ou d'écriture dans le fichier.

- Si vous ajoutez des données à un fichier, le pointeur de fichier se trouve normalement à la fin.
- Si vous lisez ou écrasez un fichier, le pointeur doit habituellement commencer au début du fichier.

Utilisation de délimiteurs en écriture ou en lecture

Lorsque vous écrivez des données, pensez à employer des délimiteurs pour les séparer. La lecture ultérieure des données dans une routine de mesure s'en trouve simplifiée. Un délimiteur peut être tout caractère ou toute chaîne de caractères. Prenez par exemple un point, nommé PNT1, avec les valeurs mesurées pour X, Y et Z équivalant à `2,5,4,3,6,1`. Vous pouvez écrire

ces valeurs dans un fichier de données en les séparant par une virgule, à l'aide d'un code comme suit :

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

Lorsque vous lisez des données, vous pouvez séparer les données entrantes par un délimiteur et les placer dans des variables en vue d'une utilisation ultérieure. Imaginez par exemple vouloir lire les mêmes valeurs X, Y et Z ci-dessus. Ces valeurs occupent une ligne de texte comme celle-ci : `2.5,4.3,6.1`. Vous pouvez diviser le texte aux virgules et placer les valeurs dans les variables correspondantes par une ligne de code comme suit :

```
V1=FICHIER/LIRELIGNE,FPTR,{ValX}+","+{ValY}+","+{ValZ}
```

Vous pouvez ensuite utiliser `ValX`, `ValY` et `ValZ` comme des variables normales dans votre routine de mesure. Résultats : ValX = 2.5, ValY = 4.3 et ValZ = 6.1.

Utilisation de la boîte de dialogue Fichier E/S

Toutes les commandes de fichier E/S sont à l'origine insérées dans la routine de mesure en sélectionnant l'option de menu de fichier E/S appropriée (sélectionnez **Insérer | Commande fichier E/S**). Une fois la commande insérée dans la fenêtre de modification, vous pouvez appuyer sur la touche F9 pour accéder à la boîte de dialogue Fichier E/S associée.

Exemple de boîte de dialogue Fichier E/S

Cette boîte de dialogue permet d'éditer la commande de fichier E/S *actuelle* de façon visuelle. Vous pouvez également modifier une commande dans la fenêtre de modification, à l'aide des techniques présentées au chapitre « Utilisation de la fenêtre de modification ».

Vous ne devez pas utiliser cette boîte de dialogue pour insérer de *nouvelles* commandes d'E/S de fichier. Cette opération doit se faire en sélectionnant l'option de menu appropriée ou en entrant les commandes directement dans la fenêtre de modification.

Ouverture d'un fichier en lecture ou en écriture

L'option de menu **Insérer | Commande fichier E/S | Ouvrir le fichier** permet de placer une commande dans la fenêtre de modification afin d'ouvrir un fichier durant l'exécution de la routine de mesure.

Les fichiers peuvent être ouverts afin de visualiser des informations ou pour ajouter et enregistrer des informations.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
<nom_pointeur_fichier> =Fichier/Ouvrir,<nom_fichier>,<mode_ouverture>
```

Descriptions de certains composants de cette commande :

<filepointername>
L'ID du pointeur de fichier choisi par l'utilisateur et servant à accéder au fichier ouvert. Cet ID fait référence au fichier ouvert dans les autres commandes de fichier E/S.

<filename>
Nom du fichier sur le disque à ouvrir.

<openmode>
Mode dans lequel le fichier doit être ouvert. Les fichiers peuvent être ouverts dans les modes suivants : Lire, Écrire ou Ajouter.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Ouvrir le fichier.
3. Appuyez sur F9.

Exemple de code pour Ouvrir le fichier

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Le code suivant ouvre un fichier nommé TEST.TXT en lecture, écriture et ajout ; il stocke son nom dans un pointeur de fichier nommé FPTR.

```
FPTR=FICHIER/OUVRIR,C:\PCDMISW\TEST.TXT,LIRE
FPTR=FICHIER/OUVRIR,C:\PCDMISW\TEST.TXT,ÉCRIRE
FPTR=FICHIER/OUVRIR,C:\PCDMISW\TEST.TXT,AJOUTER
```

Sachez que vous pouvez utiliser un commentaire d'entrée pour prendre le chemin complet comme entrée et l'utiliser dans une commande FICHIER/OUVRIR. Vous pouvez faire de même en utilisant la commande FICHIER/DIALOG. Prenez les exemples suivants :

```
C1=COMMENTAIRE/ENTRÉE,Tapez le chemin complet et le nom du fichier.
V1=FICHIER/DIALOG, CHOISISSEZ LE FICHIER À OUVRIR
FPTR=FILE/OPEN,C1.INPUT,READ
FPTR=FICHIER/OUVRIR,V1,LIRE
```

Voir « Affichage d'une boîte de dialogue de fichier ».

Fermeture d'un fichier ouvert après lecture ou écriture

L'option de menu **Insérer | Commande fichier E/S | Fermer fichier** permet d'insérer une commande dans la fenêtre Édition afin de fermer un fichier durant l'exécution de la routine de mesure. La fermeture de fichiers ouverts libère des ressources et enregistre toutes les modifications apportées.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
Fichier/Fermer, <nom_pointeur_fichier>,<mode_fermeture>
```

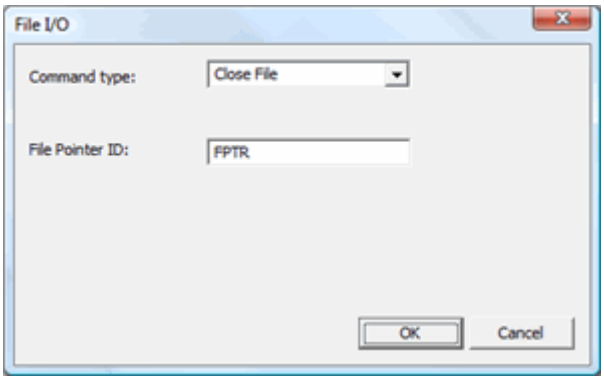
Descriptions de certains composants de cette commande :

<filepointername>
ID utilisé pour identifier le fichier et créé quand le fichier est ouvert.

<closemode>
Ce paramètre a deux options : KEEP ou DELETE. Avec KEEP, PC-DMIS ferme simplement le fichier défini dans le pointeur de fichier. Avec SUPPRIMER, PC-DMIS ferme le fichier et le supprime.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Fermer le fichier.
3. Appuyez sur F9.



Exemple de code pour Fermer le fichier

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Le code suivant ferme le fichier affecté au pointeur de fichier FPTR :

```
FICHIER/FERMER,FPTR,GARDER
```

Ce code, qui utilise le paramètre SUPPRIMER, ferme et supprime le fichier attribué à FPTR :

```
FICHIER/FERMER,FPTR,SUPPRIMER
```

Lecture d'un caractère dans un fichier

L'option de menu **Insérer | Commande fichier E/S | Commandes de lecture | Lire le caractère** place une commande dans la fenêtre de modification afin de lire un caractère à partir du fichier spécifié par le champ nom_pointeur_fichier (voir syntaxe ci-dessous) et attribue ce caractère à la variable spécifiée dans le champ de nom de variable.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
<nom_variable> = Fichier/LireCaractère,<nom_pointeur_fichier>
```

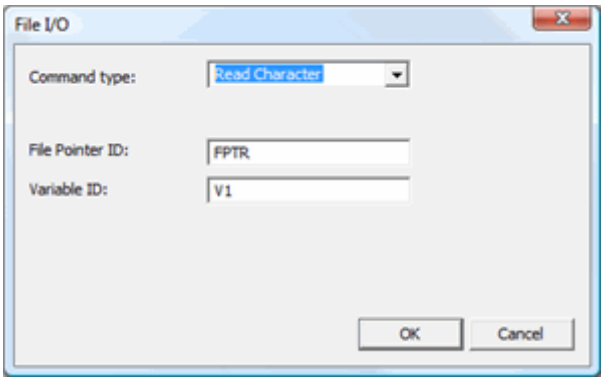
Descriptions de certains composants de cette commande :

<filepointername>
ID utilisé pour ouvrir le fichier.

<varname>
Nom de la variable devant contenir ce caractère.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Lire le caractère.
3. Appuyez sur F9.



Exemple de code pour Lire le caractère

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Observez l'exemple suivant qui lit un caractère à la fois dans une ligne de fichier de données jusqu'à ce qu'il rencontre un espace.

```
V1=FICHIER/EXISTE,test.txt
SI/V1<>0
COMMENTAIRE/OPER,Lecture dans le fichier de données possible. Cliquez sur OK pour
continuer.
ASSIGN/V3 = ""
FPTR=FICHIER/OUVERT,D:\Fichiers programme\pcdmis35\test.txt,LIRE
FAIRE/
V2=FICHIER/LIRE_CARACTÈRE,FPTR
ASSIGN/V3 = V3+V2
JUSQU'À/V2 == " "
FICHIER/FERMER,FPTR
COMMENTAIRE/OPER,« Le premier mot d'une ligne de texte du fichier est : » + V3
END_IF/
ELSE/
COMMENT/OPER,Lecture dans le fichier de données impossible. La routine va se terminer.
ALLERÀ/FIN
END_ELSE/
FIN=ÉTIQUETTE/
ROUTINE/END
```

Explication du code

V1=FICHIER/EXISTE
Cette ligne vérifie l'existence du fichier spécifié. Le fichier doit être placé dans le répertoire de PC-DMIS pour que ce code fonctionne ; dans le cas contraire, la ligne contenant le fichier doit également contenir son chemin complet. V1 reçoit le résultat de la vérification du fichier. Il s'agit d'une valeur non nulle s'il existe, nulle dans le cas contraire.

IF/V1<>0
Cette ligne prend la valeur V1 et vérifie s'il s'agit d'une valeur non nulle. Si tel est le cas, un commentaire apparaît pour indiquer que le processus de lecture est prêt à commencer. Si elle est nulle en revanche, la routine de mesure se termine.

ASSIGN/V3 = ""
Cette ligne crée une chaîne vide et l'affecte à V3. Le code utilise cette variable pour bâtir une chaîne à partir des caractères individuels lus. Si vous ne créez pas la chaîne vide, la valeur par défaut de V3 est 0.

FPTR=FICHIER/OUVRIR
Cette ligne ouvre le fichier spécifié en lecture et l'affecte au pointeur de fichier par défaut FPTR.

FAIRE
Cette ligne commence une boucle FAIRE/JUSQU'À. Elle délimite le code FICHIER/LIRE_CARACTÈRE pour lire continuellement un caractère à la fois. La boucle s'arrête lorsqu'elle lit un espace.

V2=FICHIER/LIRE_CARACTÈRE,FPTR
Cette ligne lit un caractère dans le fichier ouvert lié au pointeur de fichier FPTR. Le caractère est stocké dans la variable V2.

ASSIGN/V3 = V3+V2
Cette ligne utilise la variable V3 vide, concatène la chaîne V3 avec V2, puis réaffecte la valeur à V3. Les exécutions successives de la boucle DO/UNTIL ajoutent chacune un caractère à V3.

JUSQU'À/V2 == " "
Cette ligne termine la boucle FAIRE / JUSQU'À lorsque le code FICHIER/LIRE_CARACTÈRE rencontre un espace dans le fichier ouvert.

FICHIER/FERMER,FPTR
Cette ligne ferme le fichier de données ouvert, auquel les autres processus du système peuvent ainsi à nouveau accéder. Le reste du code termine l'exécution et affiche dans un commentaire à l'opérateur le premier mot lu dans le fichier de données.

Lecture d'une ligne dans un fichier

L'option **Insérer | Commande fichier E/S | Commandes de lecture | Lire la ligne** place une commande dans la fenêtre Édition afin de lire une ligne à partir du fichier spécifié durant l'exécution. Cette commande définit la variable spécifiée par l'ID de variable à 1 (vrai) ou 0 (faux) pour indiquer le succès (vrai) ou l'échec (faux) de l'appel. L'expression requise par cette

commande peut être utilisée pour délimiter la ligne lue et pour renseigner automatiquement les variables et références avec des données lues dans le fichier. Les informations sont lues à partir du fichier d'entrée jusqu'au caractère de retour à la ligne suivante.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
<nom_variable> = Fichier/LireLigne,<nom_pointeur_fichier>,<expr>
```

Descriptions de certains composants de cette commande :

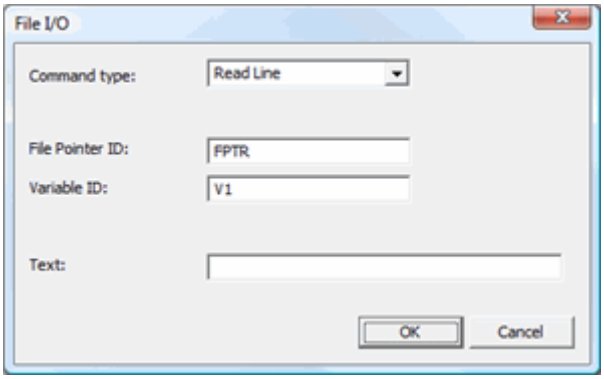
<varname>
Nom de la variable qui contiendra le résultat indiquant le succès ou l'échec de la commande ReadLine. Elle renvoie OK ou EOF.

<filepointername>
Nom spécifié pour le pointeur de fichier à l'ouverture du fichier.

<expr>
La ou les variables de destination des données d'entrée. Les données d'entrée peuvent être délimitées par un texte afin de faciliter l'analyse des lignes entrantes de données. Les références de Variables et d'éléments doivent être placées entre accolades.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Lire la ligne.
3. Appuyez sur F9.



Exemple de code pour Lire la ligne

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Observez l'exemple suivant qui lit une ligne à la fois dans un fichier de données jusqu'à ce que la commande FILE/READ_LINE rencontre une ligne vide. La routine de mesure affiche ensuite le bloc de texte obtenu et se termine.

```
V1=FICHIER/EXISTE,test.txt
SI/V1<>0
COMMENTAIRE/OPER,Lecture dans le fichier de données possible. Cliquez sur OK pour
continuer.
ASSIGN/V3 = ""
FPTR=FICHIER/OUVERT,D:\Fichiers programme\pcdmis35\test.txt,LIRE
FAIRE/
V2=FICHIER/LIRELIGNE,FPTR,{LIGNE}
ASSIGN/V3 = V3 + LIGNE
COMMENTAIRE/OPER,« La valeur actuelle de la variable V3 est :
,V3
JUSQU'À/V2 == « FDF »
FICHIER/FERMER,FPTR
COMMENTAIRE/OPER,« Le bloc de texte contient : »
,V3
END_IF/
ELSE/
COMMENT/OPER,Lecture dans le fichier de données impossible. La routine va se terminer.
ALLERÀ/FIN
END_ELSE/
FIN=ÉTIQUETTE/
ROUTINE/END
```

Explication du code

Une partie importante de ce code est semblable à celui décrit dans « Exemple de code pour Lire le caractère ». Seul le code spécifique à cet exemple est listé ici.

FAIRE
Cette ligne commence une boucle FAIRE/JUSQU'À. Elle délimite le code `FICHIER/LIRE_LIGNE` pour lire de façon continue une ligne à la fois. La boucle s'arrête lorsqu'elle atteint la fin du fichier.

V2=FICHIER/LIRELIGNE,FPTR,{LIGNE}
Cette ligne lit tout le texte jusqu'au prochain retour chariot. Au lieu de stocker le texte dans V2, comme le ferait FICHIER/LIRE_CARACTÈRE, ce code agit différemment.

- V2, dans ce cas, renvoie deux valeurs : soit « OK » soit « FDF ». Elle renvoie « OK » s'il reste des lignes à lire. Elle renvoie « FDF » si la fin du fichier est atteinte.
- Le code `{LIGNE}` est une variable entrée par l'utilisateur et qui stocke le texte réel. Elle est entre accolades pour indiquer à PC-DMIS qu'il s'agit d'une variable et non d'une partie de texte délimité. Sans les accolades, PC-DMIS rechercherait dans le fichier une chaîne de caractères nommée « LIGNE » et ne renverrait que le texte après « LIGNE » et avant le retour chariot suivant.

`ASSIGN/V3 = V3+LINE`
Cette ligne utilise la variable V3 vide, concatène la chaîne V3 avec `LINE` puis réaffecte la valeur obtenue à V3. Les exécutions successives de la boucle DO/UNTIL ajoutent donc chacune une ligne à V3.

`UNTIL/V2 == "FDF"`
Cette ligne teste la condition de la boucle DO / UNTIL. Lorsque le code `FILE/READLINE` rencontre la fin du fichier, la boucle s'arrête. Lorsque le cours de la routine sort de la boucle, le reste du code termine l'exécution et affiche le bloc de code complet dans un commentaire à l'opérateur.

Autres exemples :
la ligne de commande
`Result = File/ReadLine,F1, "Part ID :" + {V1}` fait apparaître un texte à la ligne après le texte "Part ID:" à attribuer à V1. La ligne est lue à partir du fichier ouvert en utilisant F1 comme nom de pointeur de fichier. Le résultat de la lecture (succès ou échec) est stocké dans la variable Résultat.

Les lignes de commande
```
File/ReadLine,F1, "Emplacement:"+{VARX}+ ","+{VARY}+ ","+{VARZ}+
","+{VARI}+ ","+{VARJ}+ ","+{VARK}
ASSIGN/CIR1.XYZ = MPOINT(VARX, VARY, VARZ)
ASSIGN/CIR1.IJK = MPOINT(VARI, VARJ, VARK)
```
lisent les chaînes séparées par des virgules après la chaîne "Emplacement:" et stocke les valeurs dans les valeurs X, Y, Z, et I, J, K de CIR1.

La ligne de commande
`File/ReadLine,F1, "Value # " + loopvar + " : " + {var2}` reseigne var2 avec le texte apparaissant après le signe deux-points. La variable loopvar dans cet exemple n'est pas entre accolades et sert à délimiter le texte.

Exemple de code concernant des numéros commençant par des zéros

Si le fichier que vous lisez contient des lignes de numéros, vous verrez que PC-DMIS ignore les zéros du début. Par exemple, si votre ligne contient une valeur de 005450, PC-DMIS lit cette valeur strictement comme un nombre et retourne la valeur de 5450, en ignorant les deux zéros du début. Cela peut vous convenir ou ne pas vous convenir.

Supposez que vous ayez un fichier de texte créé par un logiciel de lecteur de code barre externe qui contient ces deux lignes de données :

```
290291143;582750;0010

291143;5827;0010
```

Vous pourriez utiliser un code simple comme celui-ci pour mettre les valeurs des numéros entre les deux points :

```
ASSIGN/PREMIÈRE_VALEUR=0
ASSIGN/SECONDE_VALEUR=0
```

```
        ASSIGN/TROISIÈME_VALEUR=0
        ASSIGN/NUMLIGNE=1
FPTR=FILE/OPEN,D:\TEMP\CODES.TXT,READ
        FAIRE/
INLINE=FILE/READLINE,FPTR,{FIRST_VALUE}+";"+{SECOND_VALUE}+";"+{THIRD_VALUE}
        COMMENTAIRE/OPER,NON,"LIGNE NUMÉRO: " + NUMLIGNE
                ,« Première Valeur : » + PREMIÈRE_VALEUR
                ,« Seconde Valeur : »+ SECONDE_VALEUR
                ,« Troisième Valeur : » + TROISIÈME_VALEUR
        JUSQU'À/ENLINE == « FDF »
        FICHIER/FERMER,FPTR,GARDER
```

Alors que le code analyse avec succès les lignes de texte et rétablit les valeurs des numéros, il enlève aussi tous les zéros précédant pour toute valeur qu'il rétablit. Ainsi, la variable TROISIÈME_VALEUR contient une valeur de 10, au lieu de 0010.

Pour garder les zéros qui précèdent, vous devez traiter toute la ligne comme une chaîne et utiliser à la place les fonctions de chaîne INDEX, GAUCHE et MILIEU pour déterminer la position des points-virgules dans une ligne de texte et obtenir la valeur des numéros :

```
FPTR=FILE/OPEN,D:\TEMP\CODES.TXT,READ
        ASSIGN/NUMLIGNE=1
        FAIRE/
LINESTATUS=FILE/READLINE,FPTR,{LINESTR}
        ASSIGN/LINESTR=STR(LINESTR)
        ASSIGN/FIRST_INDEX=INDEX (LINESTR,";")
        ASSIGN/FIRST_VALUE=STR(LEFT(LINESTR,FIRST_INDEX-1))
        ASSIGN/REMAINSTR=STR(MID(LINESTR,(FIRST_INDEX)))
        ASSIGN/SECOND_INDEX=INDEX (REMAINSTR,";")
        ASSIGN/SECOND_VALUE=STR(LEFT(REMAINSTR,SECOND_INDEX-1))
        ASSIGN/THIRD_VALUE=STR(MID(REMAINSTR,SECOND_INDEX))
        COMMENTAIRE/OPER,NON,"LIGNE NUMÉRO: " + NUMLIGNE
                ,« Première Valeur : » + PREMIÈRE_VALEUR
                ,« Seconde Valeur : »+ SECONDE_VALEUR
                ,« Troisième Valeur : » + TROISIÈME_VALEUR
        ASSIGN/NUMLIGNE=NUMLIGNE+1
        JUSQU'À/STATUTLIGNE == « FDF »
        FICHIER/FERMER,FPTR,GARDER
```

Explication du code

Une partie importante de ce code est semblable à ce qui est expliqué au-dessus. Seules les explications de code spécifiques aux fonctions de chaîne sont listées ici.

`ASSIGN/FIRST_INDEX=INDEX (LINESTR,";")`
Cette ligne recherche la position du premier point-virgule et lui attribue la variable FIRST_INDEX.

ASSIGN/FIRST_VALUE=STR(LEFT(LINESTR,FIRST_INDEX-1))
Cette ligne assigne à la variable FIRST_VALUE la chaîne de caractères jusqu'à, sans l'inclure, le premier point-virgule dans la variable LINESTR. LINESTR contient la ligne entière de texte.

ASSIGN/REMAINSTR=STR(MID(LINESTR,(FIRST_INDEX)))
Cette ligne assigne à la variable REMAINSTR (correspondant à "chaîne restante") la chaîne de caractères restants de la position FIRST_INDEX (la position du premier point-virgule) jusqu'à la fin de la ligne.

ASSIGN/SECOND_INDEX=INDEX (REMAINSTR,";")
Cette ligne recherche dans la variable REMAINSTR un autre point-virgule (le second dans la ligne) et attribue la position de la variable SECOND_INDEX.

ASSIGN/SECOND_VALUE=STR(LEFT(REMAINSTR,SECOND_INDEX-1))
Cette ligne assigne à la variable SECOND_VALUE la chaîne de caractères jusqu'à, mais pas y compris, le premier point-virgule dans la variable REMAINSTR (le second point-virgule dans la ligne entière).

ASSIGN/THIRD_VALUE=STR(MID(REMAINSTR,SECOND_INDEX))
Cette ligne assigne à la variable THIRD_VALUE le chaîne de caractères allant de la position SECOND_INDEX à la fin de la ligne.

Lecture d'un bloc de texte dans un fichier

L'option **Insérer | Commande fichier E/S | Commandes de lecture | Lire le bloc** place une commande dans la fenêtre Édition afin de lire un bloc de caractères à partir d'un fichier ouvert durant l'exécution. Le nombre de caractères lus est spécifié par le paramètre de taille.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
<nom_variable>=Fichier/Lire_Bloc,<nom_pointeur_fichier>,<taille>
```

Descriptions de certains composants de cette commande :

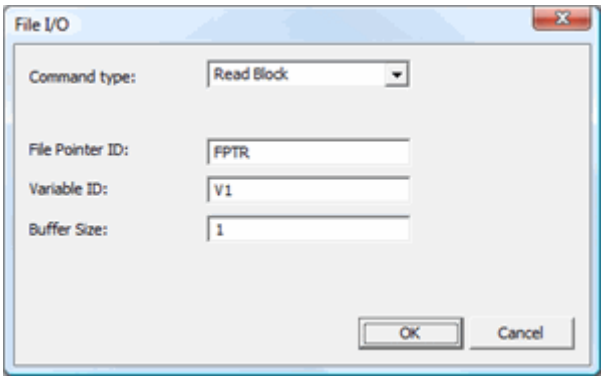
<varname>
ID de la variable qui reçoit la valeur indiquant le succès ou l'échec de l'opération de lecture de bloc.

<fptrname>
Nom spécifié pour le pointeur de fichier à l'ouverture du fichier.

<size>
Nombre de caractères à lire.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Lire le bloc.
3. Appuyez sur F9.



Exemple de code pour Lire le bloc

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Imaginez que vous avez plusieurs fichiers de données externes contenant diverses données de pièces et que les premiers caractères du fichier indiquent l'objectif du fichier. Vous pouvez utiliser la commande `Fichier/Lire_Bloc` pour ne lire que les premiers caractères avant de décider de poursuivre la lecture et de traiter toutes les lignes. Prenez le code suivant :

```
C3=COMMENTAIRE/ENTRÉE,Entrez le nom du
,code fichier à rechercher.
ASSIGN/BLOCKSIZE = LEN(C3.INPUT)
ASSIGN/FILECODE = C3.INPUT
FAIRE/
C1=COMMENTAIRE/ENTRÉE,Entrez le chemin complet,
,nom de fichier et extension au
,fichier à traiter.
,Tapez [Q] pour quitter.
IF/C1.INPUT== "Q" OR C1.INPUT=="q"
COMMENT/OPER,Vous avez choisi de sortir. Fin de la routine.
ALLERÀ/FIN
END_IF/
V1=FILE/EXISTS,C1.INPUT
SI/V1<>0
COMMENT/OPER,"Le fichier de données [" + C1.INPUT + "] existe. Cliquez sur OK pour
continuer ».
FPTR=FILE/OPEN,C1.INPUT,READ
V2=FICHIER/LIRE_BLOC,FPTR,TAILLEBLOC
FICHIER/FERMER,FPTR
SI/V2<>CODEFICHIER
COMMENTAIRE/OPER,« Le code du fichier de [" + V2 + "] ne correspond pas »
,« au CODE FICHIER de [« + CODEFICHIER + »] ».
END_IF/
JUSQU'À/V2==CODEFICHIER
```

```
COMMENT/OPER,"Le fichier [" + C1.INPUT + "] correspond."
,« Le code du fichier de [« + V2 + »] correspond »
,« au CODE FICHIER de [« + CODEFICHIER + »] ».
COMMENTAIRE/OPER,La routine peut donc traiter le fichier.
END_IF/
ELSE/
COMMENT/OPER,"Le fichier de données [" + C1.INPUT + "] n'existe pas. Veuillez réessayer
avec un fichier de données existant ».
ALLERÀ/FIN
END_ELSE/
FIN=ÉTIQUETTE/
ROUTINE/END
```

Explication du code

Une partie de ce code est semblable à celui décrit dans « Exemple de code pour Lire le caractère » ou dans « Exemple de code pour Lire la ligne ».

Seules les explications spécifiques à cet exemple figurent ici.

ASSIGN/BLOCKSIZE = LEN(C3.INPUT)
Cette ligne crée une variable `BLOCKSIZE` contenant un entier égal au nombre de caractères dans `C3.INPUT`. Elle est utilisée comme taille du bloc de caractères à lire.

`ASSIGN/FILECODE = C3.`**`INPUT`**
Cette ligne crée la variable `FILECODE` et lui assigne la valeur `C3.INPUT`.

C1=COMMENT/INPUT
Ce commentaire stocke le chemin d'accès complet entré par l'utilisateur dans la variable `C1.INPUT`.

`V1=FILE/EXISTS,C1.`**`INPUT`**
Cette ligne vérifie l'existence du nom de fichier défini dans le commentaire C1.

FAIRE/
Cette ligne commence une boucle FAIRE/JUSQU'À. Elle délimite le bloc de code permettant à l'utilisateur de saisir le nom du fichier à lire. Elle poursuit la boucle jusqu'à ce que le texte affecté à la variable `CODEFICHIER` corresponde au texte lu dans le fichier.

V2=FICHIER/LIRE_BLOC,FPTR,TAILLEBLOC
Cette ligne lit le nombre de caractères égal à l'entier contenu dans la variable `TAILLEBLOC`. Le texte est ensuite stocké dans la variable V2.

SI/V2<>FILECODE
Cette ligne commence un bloc de code IF / END IF qui teste la correspondance entre le texte de la variable V2 et le texte stocké dans la variable `FILECODE`. S'ils correspondent, la routine continue à s'exécuter. Sinon, il affiche un message indiquant que les deux codes ne correspondent pas.

JUSQU'À/V2==FILECODE
Cette ligne vérifie la condition de la boucle DO / UNTIL pour vérifier la correspondance entre

le texte de la variable `V2` et le texte stocké dans la variable `FILECODE`. Si l'instruction est fausse, la boucle `DO` s'exécute encore pour permettre à l'utilisateur de choisir un autre nom de fichier. Si l'instruction est vraie, en revanche la boucle se termine et la routine affiche un message signalant la correspondance. PC-DMIS peut alors continuer à lire chaque ligne de données dans le fichier de données spécifié.

Lecture d'un texte jusqu'à un délimiteur

L'option de menu **Insérer | Commande fichier E/S | Commandes de lecture | Lire jusqu'à** place une commande dans la fenêtre de modification afin de lire tout le texte ‘jusqu'à' l'un des délimiteurs du fichier spécifié durant l'exécution. Tout texte lu par cette commande est placé dans la variable de destination spécifiée. La commande arrête de lire le texte lorsque PC-DMIS rencontre ce qui suit :

- des délimiteurs spécifiés
- des retours de chariot
- des caractères de saut de ligne.

Si la fin du fichier est atteinte, la variable de destination est définie comme EOF (End of File, ou fin de fichier).

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
<nom_variable> = FICHIER/LIRE_JUSQU'À,<nom_pointeur_fichier>,<délimiteurs>
```

Descriptions de certains composants de cette commande :

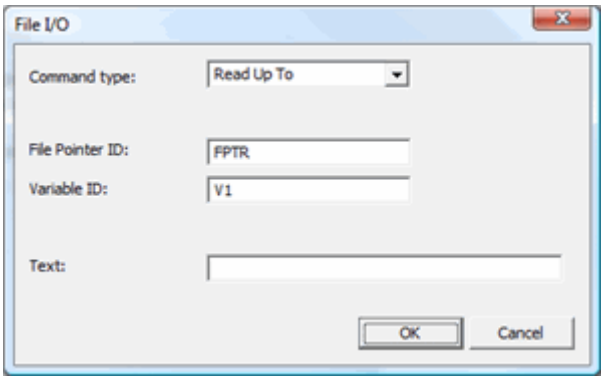
<varname>
Nom de la variable de destination.

<fptrname>
Nom spécifié pour le pointeur de fichier à l'ouverture du fichier.

<delimiters>
Chaîne qui contient aucun ou plusieurs caractères de délimiteurs.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Sélectionnez **Fenêtre Édition** pour ouvrir la fenêtre Édition.
2. Placez votre curseur sur la commande `FICHIER/LIRE`.
3. Appuyez sur F9. La boîte de dialogue **Fichier E/S** s'ouvre.



Lorsque la boîte de dialogue s'affiche :

1. Tapez le nom de la variable qui doit recevoir les informations lues dans la zone **ID de variable**.
2. Tapez le nom du pointeur de fichier dans la case **ID pointeur de fichier**.
3. Entrez le délimiteur dans la case **Texte** (n'oubliez pas de mettre le délimiteur choisi entre guillemets).
4. Cliquez sur **OK**.

Exemple de code pour Lire jusqu'à

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Exemple: Imaginez que vous avez un fichier texte appelé « exemple.txt » dans votre répertoire c:\temp qui contient cette information à la première ligne.

root:x:0:0:root:/root:/bin/bash

Pour utiliser la commande Lire jusqu'à dans ce fichier :

1. Insérez une commande `FICHIER/OUVRIR` dans la fenêtre Édition.
2. Utilisez un nom de pointeur de fichier de votre choix dans la commande Ouvrir le fichier. Cet exemple utilise « exemple » comme nom du pointeur de fichier.

La commande Ouvrir le fichier doit ressembler à ce qui suit :

```
EXEMPLE =FICHIER/OUVRIR,C:\TEMP\EXEMPLE.TXT,LIRE
```

Au moyen des commandes Lire jusqu'à de PC-DMIS, définissez des variables qui appellent des segments de données différents. Cet exemple utilise les variables suivantes pour chercher un « : » (sans les guillemets) comme délimiteur.

```
USERNAME =FILE/READ_UPTO,EXEMPLE, :
MOTDEPASSE =FICHIER/LIRE_JUSQU'À,EXEMPLE, :
UTILISATEUR =FICHIER/LIRE_JUSQU'À,EXEMPLE,:
```

Lorsque PC-DMIS exécute ces lignes, il définit donc

nomutilisateur = racine

motdepasse = x

utilisateur = 0

Pour afficher ces données pendant l'exécution, vous pouvez utiliser un commentaire d'opérateur tel que celui-ci :

```
COMMENTAIRE/OPER,Le texte suivant est lu dans exemple.txt
,Nomutilisateur:
,NOMUTILISATEUR
,Motdepasse:
,MOTDEPASSE
,Utilisateur:
,UTILISATEUR
```

Écriture d'un caractère dans un fichier

L'option de menu **Insérer | Commande de fichier E/S | Commandes d'écriture | Écrire un caractère** permet d'insérer une commande dans la fenêtre de modification afin d'écrire un seul caractère dans un fichier lors de l'exécution.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
Fichier/Écrire_caractère,<nom_pointeur_fichier>,<expr>
```

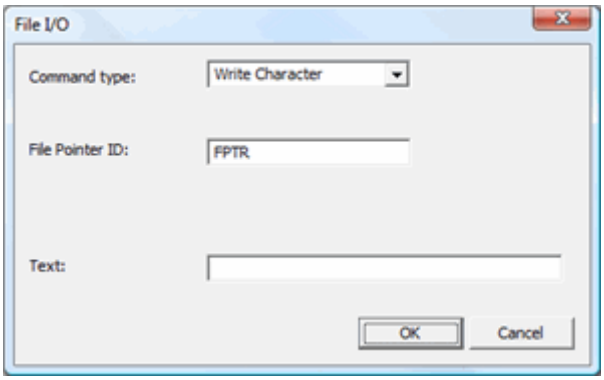
Descriptions de certains composants de cette commande :

<fptrname>
Nom spécifié pour le pointeur de fichier à l'ouverture du fichier.

<expr>
Caractère à écrire dans le fichier. Si l'expression donne plusieurs caractères, seul le premier est écrit.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Écrire un caractère.
3. Appuyez sur F9.



Exemple de code pour Écrire un caractère

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Observez le code suivant qui écrit caractère par caractère une chaîne provenant de l'utilisateur dans un fichier de données.

```
C1=COMMENTAIRE/ENTRÉE,Tapez le nom du fichier à écrire
,dans (inclure le chemin complet).
FPTR=FILE/OPEN,C1.INPUT,WRITE
C2=COMMENTAIRE/ENTRÉE,Tapez ce que vous voulez envoyer au fichier.
, La chaîne sera envoyée un caractère
,à la fois.
ASSIGN/COMPTE = 0
ASSIGN/LENGTH = LEN(C2.INPUT)
FAIRE/
ASSIGN/WRITETHIS = MID(C2.INPUT,COUNT,1)
FICHIER/ÉCRIRE_CARACTÈRE,FPTR,ÉCRIRECECI
ASSIGN/COMPTE = COMPTE + 1
JUSQU'À/COMPTE == LONGUEUR
```

Explication du code

Une partie de ce code est semblable à celui décrit dans « Exemple de code pour Lire le caractère » ou dans « Exemple de code pour Lire la ligne ».

Seules les explications spécifiques à cet exemple figurent ici.

`FPTR=FILE/OPEN,C1.INPUT,WRITE`
Cette ligne ouvre le fichier indiqué dans le commentaire C1 pour écriture et l'attribue au pointeur de fichier `FPTR`. Toutes les données de ce fichier sont écrasées si le pointeur du fichier démarre au début du fichier de données.

ASSIGN/COMPTE = 0
Cette ligne affecte la valeur zéro à une variable COMPTE définie par l'utilisateur. Elle sert à effectuer des boucles pour écrire la chaîne caractère par caractère.

ASSIGN/LENGTH = LEN(C2.INPUT)
Cette ligne utilise la fonction LEN( ) pour renvoyer la longueur d'une chaîne. Cette fonction prend un paramètre, la chaîne. Elle compte le nombre de caractères dans la chaîne (espaces compris) et renvoie la valeur entière correspondante. Dans le cas présent, la variable LENGHT définie par l'utilisateur contient cette valeur.

FAIRE/
Cette ligne commence une boucle FAIRE/JUSQU'À. Le code entre les instructions FAIRE et JUSQU'À sont exécutées jusqu'à ce que la condition de la boucle soit vraie.

ASSIGN/WRITETHIS = MID(C2.INPUT,COUNT,1)
Cette ligne crée une variable définie par l'utilisateur et appelée WRITETHIS, et elle prend la fonction MID( ) pour renvoyer un caractère de sous-chaîne de la chaîne C2.INPUT et l'attribuer à WRITETHIS. MID( ) accepte trois paramètres.

- Paramètre 1 : il s'agit de la chaîne dont s'obtiennent les valeurs. C2.INPUT est dans ce cas utilisé.
- Paramètre 2 : il s'agit de la position dans la chaîne où prendre le caractère. Le premier caractère de la chaîne est à la position 0, le deuxième à la position 1, le troisième à la position 2, et ainsi de suite. La variable COMPTE est utilisée dans le cas présent.
- Paramètre 3: il s'agit du nombre de caractères à prendre à partir de la position du second paramètre à saisir. Dans le cas présent, la valeur 1 est utilisée (l'exemple n'écrit qu'un seul caractère à la fois, il n'y a donc aucune raison d'en prendre plus).

FILE/ÉCRIRE_CARACTÈRE,FPTR,ÉCRIRECECI
Cette ligne écrit le caractère stocké dans la variable ÉCRIRECECI dans le fichier spécifié par le pointeur de fichier FPTR.

ASSIGN/COMPTE = COMPTE + 1
Cette ligne prend la valeur COMPTE courante, l'incrémente de un et renvoie la nouvelle valeur dans COMPTE.

UNTIL/COUNT == LENGTH
Cette ligne teste la condition de la boucle DO / UNTIL. Dans le cas présent, la boucle poursuit l'incrémentation de la variable COUNT jusqu'à ce qu'elle contienne la même valeur que la variable LENGTH. Puis la boucle se termine, mettant ainsi fin à la routine.

Écriture d'une ligne dans un fichier

L'option de menu **Insérer | Commande de fichier E/S | Commandes d'écriture | Écrire une ligne** permet d'insérer une commande dans la fenêtre Édition afin d'écrire une ligne dans un fichier lors de l'exécution. Utilisez la syntaxe d'expression pour écrire des variables et des informations de routine de mesure dans un fichier. Un retour à la ligne est automatiquement ajouté au texte écrit.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
Fichier/ÉcrireLigne,<nom_pointeur_fichier>,<expr>
```

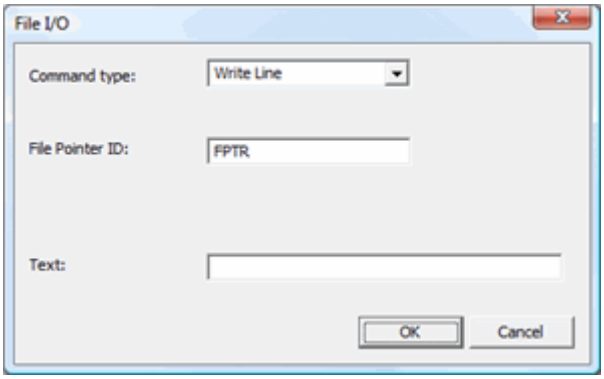
Descriptions de certains composants de cette commande :

<fptrname>
Nom spécifié pour la référence de fichier à l'ouverture du fichier.

<expr>
Texte à écrire dans le fichier. Des expressions peuvent être utilisées dans cette zone.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Écrire une ligne.
3. Appuyez sur F9.



Exemple de code pour Écrire une ligne

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Imaginez vouloir exporter des valeurs mesurées XYZ dans un fichier de données. Le code suivant vous permet d'entrer une étiquette d'élément et un fichier de données, puis d'envoyer les données X, Y et Z de cet élément à ce fichier.

```
C1=COMMENTAIRE/ENTRÉE,Tapez l'étiquette de l'élément
,à utiliser.
C2=COMMENTAIRE/ENTRÉE,Tapez le nom du fichier à écrire
,dans (inclure le chemin complet).
FPTR=FILE/OPEN,C2.INPUT,APPEND
ASSIGN/FEATNAME = C1.INPUT
ASSIGN/ALLVALS = FEATNAME.X + "," + FEATNAME.Y + "," + FEATNAME.Z
COMMENTAIRE/OPER,« Le texte à écrire est : » + TTESVALS
FICHIER/ÉCRIRELIGNE,FPTR,TTESVALS
```

```
FICHIER/FERMER,FPTR
```

Explication du code

Une partie de ce code est semblable à celui décrit dans « Exemple de code pour Lire le caractère » ou dans « Exemple de code pour Lire la ligne ».

Seules les explications spécifiques à cet exemple figurent ici.

`FPTR=FILE/OPEN,C2.INPUT,APPEND`
Cette ligne ouvre le fichier indiqué dans le commentaire C2 et l'attribue au pointeur de fichier `FPTR`. SI vous remplacez `AJOUT` par `ÉCRIRE`, le contenu existant dans le fichier de données est écrasé.

`ASSIGN/FEATNAME = C1.INPUT`
Cette ligne attribue la chaîne de l'étiquette de l'élément `C1.INPUT` à la variable définie par l'utilisateur, `FEATNAME`.

`ASSIGN/ALLVALS=FEATNAME.X+","+FEATNAME.Y+","+ FEATNAME.Z`
Cette ligne attribue à la variable `ALLVALS` la valeur `FEATNAME.X,FEATNAME.Y,FEATNAME.Z` ; elle inclut ainsi les valeurs X, Y et Z de l'étiquette de l'élément dans la commentaire d'entrée C1.

`FILE/ÉCRIRELIGNE,FPTR,TTESVALS`
Cette ligne écrit les valeurs contenues dans `TTESVALS` dans le fichier spécifié par le pointeur de fichier FPTR.

Écriture d'un bloc de texte dans un fichier

L'option de menu **Insérer | Commande de fichier E/S | Commandes d'écriture | Écrire un bloc** permet d'insérer une commande dans la fenêtre Édition afin d'écrire un bloc dans un fichier lors de l'exécution. Utilisez la syntaxe d'expression pour écrire des variables et des informations de routine de mesure dans un fichier. À la différence de la commande d'écriture d'une ligne, l'écriture d'un bloc n'ajoute *pas* de retour à la ligne à la fin.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
Fichier/ÉcrireBloc,<nom_pointeur_fichier>,<expr>
```

Descriptions de certains composants de cette commande :

<fptrname>
Nom spécifié pour la référence de fichier à l'ouverture du fichier.

<expr>
Texte à écrire dans le fichier. Des expressions peuvent être utilisées dans cette zone.

Retour à la ligne et saut de ligne : contrairement à la commande d'écriture d'une ligne, l'écriture d'un bloc n'ajoute *pas* de retour à la ligne à la fin. Cependant, si vous devez placer une

nouvelle ligne dans votre bloc de texte, vous pouvez insérer manuellement un retour à la ligne et un saut de ligne à l'aide du code CHR(10) à l'extérieur de votre chaîne citée, comme le montre cet exemple :

```
FILE/WRITEBLOCK,FPTR, "CHR(10) insère du texte... " + CHR(10) + "
...sur une nouvelle ligne."
```

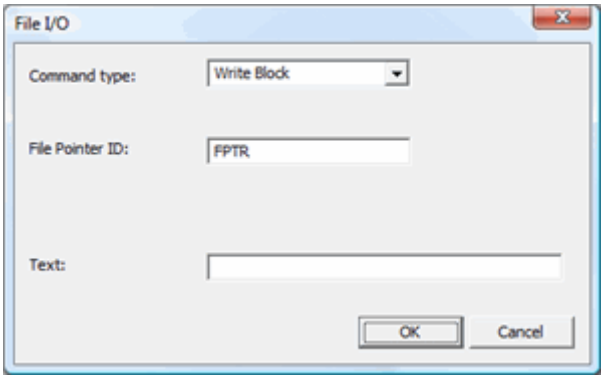
Ceci donne ce résulta dans votre fichier de sortie :

```
CHR(10) insère le texte...
...sur une nouvelle ligne.
```

Si CHR(10) est mis entre guillemets, le texte de CHR(10) est envoyé au fichier.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Écrire un bloc.
3. Appuyez sur F9.



Exemple de code pour Écrire un bloc

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Le code suivant écrit ce que l'utilisateur saisit dans un commentaire d'entrée, en ajoutant deux points qui servent de délimiteur.

```
C1=COMMENTAIRE/ENTRÉE,Tapez une chaîne. PC-DMIS ajoute deux points (comme délimiteur) et
écrit la chaîne dans le fichier de votre choix.
C2=COMMENTAIRE/ENTRÉE,Tapez le nom du fichier à écrire
,dans (inclure le chemin complet).
FPTR=FILE/OPEN,C2.INPUT,APPEND
ASSIGN/WRITETHIS = C1.INPUT + ":"
COMMENTAIRE/OPER,« Le texte à écrire est : » + ÉCRIRECECI
FICHIER/ÉCRIRELIGNE,FPTR,ÉCRIRECECI
FICHIER/FERMER,FPTR
```

Explication du code

Une partie de ce code est semblable à celui décrit dans « Exemple de code pour Lire le caractère » ou dans « Exemple de code pour Lire la ligne ».

Seules les explications spécifiques à cet exemple figurent ici.

`FPTR=FILE/OPEN,C2.INPUT,APPEND`
Cette ligne ouvre le fichier indiqué dans le commentaire C2 et l'attribue au pointeur de fichier `FPTR`.

`ASSIGN/WRITETHIS = C1.INPUT + ":"`
Cette ligne ajoute un signe deux-points au texte figurant dans `C1.INPUT` et attribue la nouvelle chaîne à la variable `WRITETHIS` définie par l'utilisateur.

`FICHIER/ÉCRIRELIGNE,FPTR,ÉCRIRECECI`
Cette ligne écrit les valeurs contenues dans `ÉCRIRECECI` dans le fichier spécifié par le pointeur de fichier FPTR. Vous pouvez ensuite lire le texte contenu dans le fichier en utilisant les deux points comme délimiteur.

Positionnement d'un pointeur de fichier au début d'un fichier

L'option de menu **Insérer | Commande fichier E/S | Commandes de position | Rembobiner jusqu'au début** permet d'insérer une commande dans la fenêtre Édition afin de positionner le pointeur de fichier au début du fichier.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

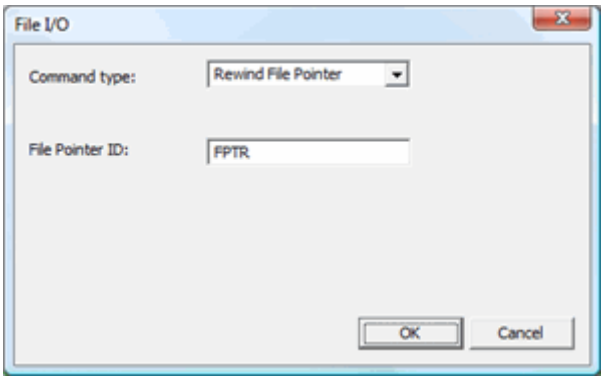
```
Fichier/Rembobiner,<nom_pointeur_fichier>
```

Descriptions de certains composants de cette commande :

<fptrname>
Nom du pointeur de fichier à repositionner au début du fichier.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Rembobiner jusqu'au début.
3. Appuyez sur F9.



Exemple de code pour Rembobiner jusqu'au début

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Observez l'exemple suivant qui lit les données d'un fichier externe ligne par ligne. Après chaque ligne, vous pouvez recommencer et lire à partir du début du fichier. Ceci démontre l'utilisation de la commande `FICHIER/REMBOBINER`.

```
C1=COMMENTAIRE/ENTRÉE,Veuillez taper le nom d'un fichier dans lequel lire.
,(inclure le chemin complet).
V1=FILE/EXISTS,C1.INPUT
SI/V1<>0
FAIRE/
FPTR=FILE/OPEN,C1.INPUT,READ
C2=COMMENTAIRE/OUINON,Voulez-vous lire depuis le début ?
IF/C2.INPUT == "YES"
FICHIER/REMBOBINER,FPTR
END_IF/
V2=FICHIER/LIRELIGNE,FPTR,{LIGNE}
COMMENTAIRE/OPER,« La ligne courante est : » + LIGNE
JUSQU'A/V2==« FDF »
END_IF/
FICHIER/FERMER,FPTR
COMMENT/OPER,Sortie de la routine.
```

Explication du code

Une partie de ce code est semblable à celui décrit dans « Exemple de code pour Lire le caractère » ou dans « Exemple de code pour Lire la ligne ».

Seules les explications spécifiques à cet exemple figurent ici.

C2=COMMENT/YESNO
Cette ligne vous demande si vous voulez commencer la lecture au début du fichier. Elle stocke la réponse YES/NO dans la variable, `C2.INPUT`.

IF/C2.INPUT == "YES"
Cette ligne commence un bloc IF / END IF. Elle teste la condition de `C2.INPUT` avec la valeur YES. Si la condition est vraie, PC-DMIS exécute les lignes après l'instruction SI. Si la condition est fausse, PC-DMIS exécute le code après l'instruction FIN SI.

FICHIER/REMBOBINER,FPTR
Cette ligne ramène le pointeur du fichier au début du fichier de données.

FIN_SI/
Cette ligne sort du bloc de code SI / FIN SI.

Enregistrement de la position courante d'un pointeur de fichier

L'option **Insérer | Commande fichier E/S | Commandes de position | Enregistrer la position du fichier** permet d'insérer une commande dans la fenêtre de modification, afin d'enregistrer la position actuelle d'un pointeur de fichier dans le flux du fichier. La position enregistrée peut être rappelée ultérieurement à l'aide de la commande Rappeler la position du fichier.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

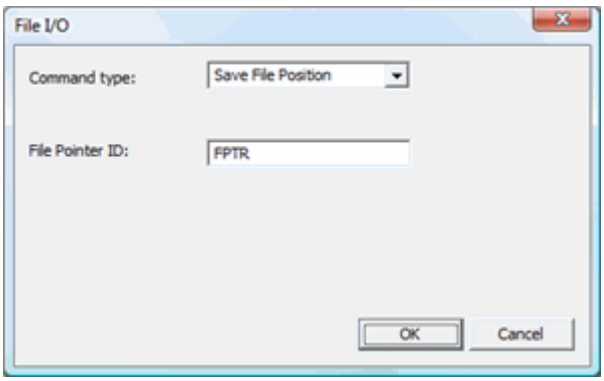
```
Fichier/EnregistrerPosition,<nom_pointeur_fichier>
```

Descriptions de certains composants de cette commande :

<fptrname>
Nom du pointeur de fichier dont la position de fichier est enregistrée.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Enregistrer la position du fichier.
3. Appuyez sur F9.



Exemple de code pour Enregistrer la position du fichier

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Observez l'exemple suivant qui lit les données d'un fichier externe ligne par ligne. Après chaque ligne, vous avez la possibilité d'enregistrer la position dans le fichier afin de la rappeler plus tard. Ceci illustre l'utilisation de la commande `FICHIER/ENREGISTRER_POSITION`.

```
C1=COMMENTAIRE/ENTRÉE,Veuillez taper le nom d'un fichier dans lequel lire.
,(inclure le chemin complet).
V1=FILE/EXISTS,C1.INPUT
SI/V1<>0
FAIRE/
FPTR=FILE/OPEN,C1.INPUT,READ
C2=COMMENTAIRE/OUINON,Voulez-vous enregistrer la position du fichier et la rappeler plus
tard ? La boucle va se terminer.
IF/C2.INPUT == "YES"
FICHIER/ENREGISTRER_POSITION,FPTR
ALLERÀ/FINBOUCLE
END_IF/
V2=FICHIER/LIRELIGNE,FPTR,{LIGNE}
COMMENTAIRE/OPER,« La ligne courante est : » + LIGNE
JUSQU'A/V2==« FDF »
END_IF/
FICHIER/FERMER,FPTR
FINBOUCLE=ÉTIQUETTE/
COMMENTAIRE/OPER,Vous avez arrêté la lecture.
ROUTINE/END
```

Explication du code

Une partie importante de ce code est semblable à celui décrit dans « Exemple de code pour Rembobiner jusqu'au début ».

Seules les explications spécifiques à cet exemple figurent ici.

`C2=COMMENT/YESNO`
Cette ligne vous demande si vous voulez enregistrer la position courante dans le fichier et sortir de la boucle. Elle stocke la réponse YES/NO dans la variable, `C2.INPUT`.

`FICHIER/ENREGISTRER_POSITION,FPTR`
Cette ligne stocke la position du pointeur de fichier dans le flux du fichier.

Tant que vous ouvrez le même fichier avec le même nom de pointeur de fichier dans la même routine de mesure, vous pouvez rappeler une position de fichier enregistrée et continuer la

lecture là où vous l'avez arrêtée. Pour continuer cet exemple, voir la rubrique "Exemple de code pour rappeler la position d'un fichier".

Rappel d'une position de pointeur de fichier enregistrée

L'option **Insérer | Commande fichier E/S | Commandes de position | Rappeler la position du fichier** permet d'insérer une commande dans la fenêtre Édition afin de rappeler une position de fichier précédemment enregistrée. Utilisez la commande Enregistrer la position du fichier pour enregistrer une position dans un fichier ouvert.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

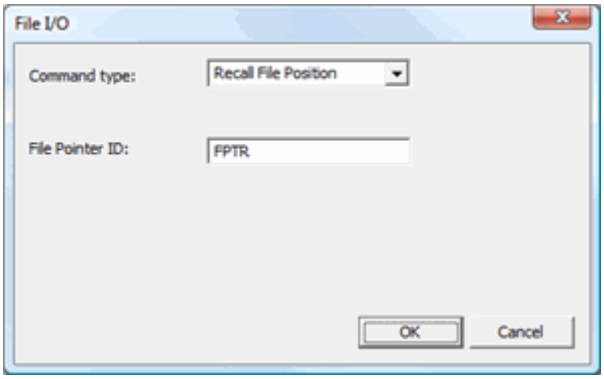
```
Fichier/RappelPosition, <nom_pointeur_fichier>
```

Descriptions de certains composants de cette commande :

<fptrname>
Nom du pointeur de fichier dont la position est rappelée.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Rappeler la position du fichier.
3. Appuyez sur F9.



Exemple de code pour Rappeler la position du fichier

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Cet exemple ouvre un fichier précédemment fermé, utilise un ancien pointeur de fichier et rappelle la position enregistrée du pointeur de fichier. Il lit ensuite les données à partir de cette position. Le code suivant démontre l'utilisation de la commande `FICHIER/RAPPEL_POSITION`

et prolonge l'exemple de code donné dans la rubrique « Exemple de code pour Enregistrer la position du fichier ».

```
COMMENT/OPER,La routine va à présent rappeler la position de fichier enregistrée.
FPTR=FILE/OPEN,C1.INPUT,READ
FICHIER/REMBOBINER,FPTR
COMMENTAIRE/OPER,Le fichier a été rembobiné pour test.
,La première ligne sera lue pour tester le rembobinage.
V3=FICHIER/LIRELIGNE,FPTR,{LIGNE}
COMMENTAIRE/OPER,La première ligne est :
,LIGNE
FICHIER/REMBOBINER,FPTR
FICHIER/RAPPEL_POSITION,FPTR
COMMENTAIRE/OPER,La position de fichier précédemment enregistrée a été rappelée.
,Les données de la ligne de la position enregistrée vont maintenant être imprimées.
V4=FICHIER/LIRELIGNE,FPTR,{ENREGISTRÉ}
COMMENTAIRE/OPER,Le texte à la position enregistrée est :
,ENREGISTRÉ
```

Explication du code

Une partie importante de ce code est semblable à celui décrit dans « Exemple de code pour Rembobiner jusqu'au début ».

Seules les explications spécifiques à cet exemple figurent ici.

`FICHIER/RAPPEL_POSITION,FPTR`
Cette ligne rappelle la position de pointeur de fichier enregistrée dans le flux du fichier pour le pointeur de fichier appelé `FPTR`.

`V4=FICHIER/LIRELIGNE,FPTR,{ENREGISTRÉ}`
Cette ligne lit la ligne après la position de pointeur enregistrée et l'affecte à la variable définie par l'utilisateur `ENREGISTRÉ`. Cette variable est ensuite imprimée dans le commentaire d'opérateur suivant.

Copie d'un fichier

L'option de menu **Insérer | Commande fichier E/S | Copier le fichier** permet d'insérer une commande dans la fenêtre Édition afin de copier un fichier au moment de l'exécution.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
Fichier/Copier,<nom_fichier_source>,<nom_fichier_dest>,<mode_remplacement>
```

Descriptions de certains composants de cette commande :

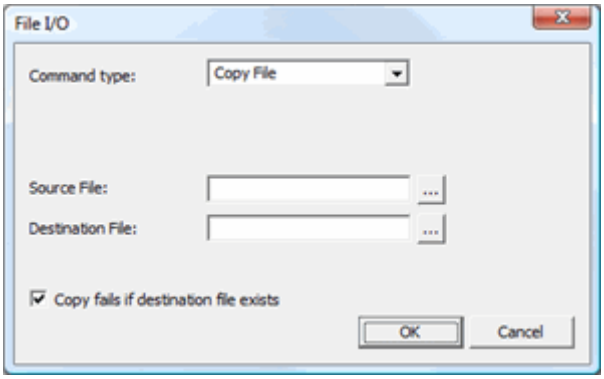
<srcfilename>
Nom du fichier source (à partir duquel la copie est effectuée).

<destfilename>
Nom du fichier de destination (vers lequel la copie est effectuée).

<replacemode>
Action effectuée si le fichier de destination existe déjà. Si la destination existe, les deux modes sont écrasement et échec.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Copier le fichier.
3. Appuyez sur F9.



Exemple de code pour Copier le fichier

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Le code suivant demande le nom d'un fichier à copier et un répertoire de destination pour la copie :

```
C1=COMMENTAIRE/ENTRÉE,Veuillez taper le nom du fichier à copier.
,(inclure le chemin de fichier complet)
C2=COMMENTAIRE/ENTRÉE,Veuillez taper un nom de fichier de destination.
,(inclure le chemin de fichier complet)
V1=FILE/EXISTS,C1.INPUT
SI/V1<>0
COMMENTAIRE/OPER,Le fichier à copier existe. Début de la copie.
FILE/COPY,C1.INPUT,C2.INPUT,FAIL_IF_DEST_EXISTS
V2=FILE/EXISTS,C2.INPUT
SI/V2==0
COMMENT/OPER,"Le fichier existe dans : " + C2.INPUT
,Fin de copie.
ROUTINE/END
END_IF/
```

```
ELSE/
COMMENTAIRE/OPER,Copie de fichier réussie.
ROUTINE/END
END_ELSE/
END_IF/
COMMENTAIRE/OPER,Le fichier à copier n'existe pas.
```

Explication du code

Une partie importante de ce code est semblable à celui décrit dans « Exemple de code pour Lire le caractère » ou dans « Exemple de code pour Lire la ligne ».

Seules les explications spécifiques à cet exemple figurent ici.

C1=COMMENT/INPUT
Cette ligne prend le chemin complet du fichier à copier et le place dans la variable `C1.INPUT`.

C2=COMMENT/INPUT
Cette ligne prend le chemin complet du fichier de destination et le place dans la variable `C2.INPUT`.

FILE/COPY,C1.INPUT,C2.INPUT,FAIL_IF_DEST_EXISTS
Cette ligne copie le fichier d'origine dans un fichier de destination. Cette commande prend trois paramètres.

- Le paramètre 1 est `C1.ENTRÉE`. Il s'agit du chemin complet du fichier à copier.
- Le paramètre 2 est `C2.ENTRÉE`, soit le chemin complet du fichier de destination.
- Le paramètre 3, dans le cas présent, abandonne la procédure FICHIER/COPIER s'il rencontre un fichier existant portant le même nom que le fichier de destination. Vous pouvez modifier ce paramètre afin d'écraser les fichiers existants portant le même nom.

Déplacement d'un fichier

L'option de menu **Insérer | Commande fichier E/S | Déplacer le fichier** permet d'insérer une commande dans la fenêtre Édition afin de déplacer un fichier au moment de l'exécution.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
Fichier/Déplacer,<nom_ancien_fichier>,<nom_nouveau_fichier>
```

Descriptions de certains composants de cette commande :

<oldfilename>
Emplacement et nom de l'ancien fichier.

<newfilename>
Emplacement et nom du nouveau fichier.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Déplacer le fichier.
3. Appuyez sur F9.

Exemple de code pour Déplacer le fichier

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Le code suivant demande un nom de fichier à déplacer, un répertoire et un nom de fichier dans lesquels faire le déplacement, puis effectue l'opération de déplacement du fichier.

```
C1=COMMENTAIRE/ENTRÉE,Veuillez taper le nom du fichier à déplacer.
,(inclure le chemin de fichier complet)
C2=COMMENTAIRE/ENTRÉE,Veuillez taper un nom de fichier de destination.
,(inclure le chemin de fichier complet)
V1=FILE/EXISTS,C1.INPUT
SI/V1<>0
COMMENTAIRE/OPER,Le fichier à déplacer existe. Début du déplacement du fichier.
FILE/MOVE,C1.INPUT,C2.INPUT
V2=FILE/EXISTS,C2.INPUT
SI/V2==0
COMMENT/OPER,"Le fichier existe dans : " + C2.INPUT
,La commande DÉPLACER n'a pas fonctionné correctement.
ROUTINE/END
END_IF/
ELSE/
COMMENTAIRE/OPER,Déplacement de fichier réussi.
ROUTINE/END
END_ELSE/
```

```
END_IF/
COMMENTAIRE/OPER,Le fichier d'origine n'existe pas. Réessayez.
```

Explication du code

Une partie importante de ce code est semblable à celui décrit dans "Exemple de code pour Copier le fichier".

Seules les explications spécifiques à cet exemple figurent ici.

`FILE/MOVE,C1.INPUT,C2.INPUT`
Cette ligne copie le fichier d'origine dans un fichier de destination. Cette commande prend deux paramètres.

- Le paramètre 1 est `C1.ENTRÉE`. Il s'agit du chemin complet du fichier à déplacer.
- Le paramètre 2 est `C2.ENTRÉE`, soit le chemin complet du fichier de destination.

Suppression d'un fichier

L'option de menu **Insérer | Commande fichier E/S | Supprimer le fichier** permet d'insérer une commande dans la fenêtre Édition afin de supprimer un fichier au moment de l'exécution.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
Fichier/Supprimer,<nom_fichier>
```

Descriptions de certains composants de cette commande :

<filename>
Nom du fichier à supprimer.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Supprimer le fichier.
3. Appuyez sur F9.

Exemple de code pour Supprimer le fichier

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Le code suivant demande un nom de fichier, puis supprime le fichier en question.

```
C1=COMMENTAIRE/ENTRÉE,Veuillez taper le nom du fichier à supprimer.
,(inclure le chemin de fichier complet)
V1=FILE/EXISTS,C1.INPUT
SI/V1<>0
COMMENTAIRE/OPER,Le fichier existe. Prêt pour la suppression.
FILE/DELETE,C1.INPUT
V2=FICHIER/EXISTE,
SI/V2==0
COMMENTAIRE/OPER,Suppression du fichier réussie.
ROUTINE/END
END_IF/
ELSE/
COMMENTAIRE/OPER,Le fichier existe toujours.
ROUTINE/END
END_ELSE/
END_IF/
COMMENTAIRE/OPER,Le fichier à supprimer n'existe pas. Choisissez un fichier qui existe.
```

Explication du code

Une partie importante de ce code est semblable à celui décrit dans "Exemple de code pour Déplacer le fichier".

Seules les explications spécifiques à cet exemple figurent ici.

`FILE/DELETE,C1.INPUT`
Cette ligne supprime le fichier indiqué. Cette commande prend un paramètre, le nom du fichier à supprimer. Dans ce cas, `C1.INPUT`.

Vérification de l'existence d'un fichier

L'option de menu **Insérer | Commande fichier E/S | Occurence de fichier** permet d'insérer une commande dans la fenêtre Édition afin de vérifier l'existence d'un fichier au moment de l'exécution et de définir la variable fournie avec le résultat.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
<nom_variable> = Fichier/Occurence,<nom_fichier>
```

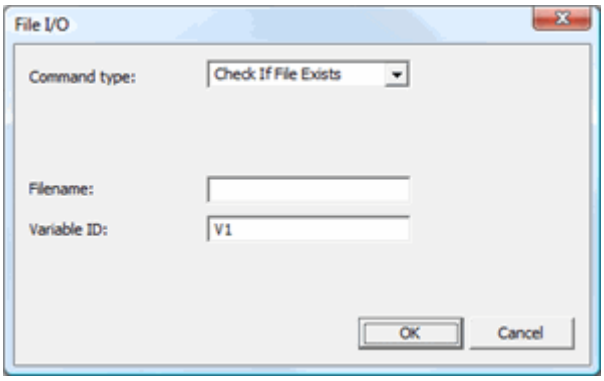
Descriptions de certains composants de cette commande :

<filename>
Nom du fichier dont il faut vérifier la présence sur le disque.

<varname>
Nom de la variable définie avec le résultat du contrôle effectué. La variable est définie à 1 si le fichier existe et à 0 dans le cas contraire.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Occurrence de fichier.
3. Appuyez sur F9.



Exemple de code pour Occurrence de fichier

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Le code suivant demande un nom de fichier, puis en vérifie l'existence.

```
C1=COMMENTAIRE/ENTRÉE,Veuillez taper le nom de fichier à vérifier.
V1=FILE/EXISTS,C1.INPUT
SI/V1<>0
      COMMENTAIRE/OPER,Le fichier existe.
END_IF/
ELSE/
      COMMENTAIRE/OPER,Le fichier n'existe pas.
END_ELSE/
```

Explication du code

Une partie importante de ce code est semblable à celui décrit dans « Exemple de code pour Lire le caractère » ou dans « Exemple de code pour Lire la ligne ».

Seules les explications spécifiques à cet exemple figurent ici.

V1=FILE/EXISTS,C1.INPUT
Cette ligne vérifie l'existence du fichier spécifié. Le fichier doit être placé dans le répertoire de PC-DMIS pour que ce code fonctionne ; dans le cas contraire, la ligne contenant le fichier doit également contenir son chemin complet. V1 reçoit le résultat de la vérification du fichier. Il s'agit d'une valeur non nulle s'il existe, nulle dans le cas contraire.

Affichage d'une boîte de dialogue de fichier

L'option de menu **Insérer | Commande fichier E/S | Boîte de dialogue de fichier** permet d'insérer une commande dans la fenêtre Édition afin d'afficher une boîte de dialogue de fichier au moment de l'exécution. Cette boîte de dialogue de fichier peut être utilisée pour permettre à l'opérateur de la routine de mesure de choisir un nom de fichier au moment de l'exécution. Le nom du fichier choisi est stocké dans la variable spécifiée.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

```
<nom_variable> = Fichier/Dialogue,<expr>
```

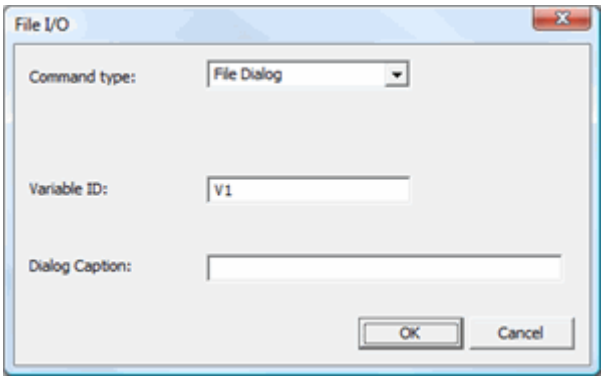
Descriptions de certains composants de cette commande :

<varname>
Nom de la variable définie sur le nom du fichier choisi par l'utilisateur dans la boîte de dialogue de fichier.

<expr>
Texte affiché dans la barre de titre de la boîte de dialogue de fichier.

Pour accéder à la boîte de dialogue associée à cette commande Fichier E/S :

1. Ouvrez la fenêtre de modification.
2. Placez votre curseur sur la commande Boîte de dialogue de fichier.
3. Appuyez sur F9.



Exemple de code pour Boîte de dialogue de fichier

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Le code suivant soumet une boîte de dialogue vous permettant de choisir un fichier à supprimer.

```
V1=FICHIER/DIALOG,Choisissez un fichier à supprimer.
V2=FICHIER/EXISTE,V1
SI/V2<>0
COMMENTAIRE/OPER,Le fichier existe. Prêt pour la suppression.
FICHIER/SUPPRIMER,V1
V3=FICHIER/EXISTE,
SI/V3==0
COMMENTAIRE/OPER,Suppression du fichier réussie.
ROUTINE/END
END_IF/
ELSE/
COMMENTAIRE/OPER,Le fichier existe toujours.
ROUTINE/END
END_ELSE/
END_IF/
COMMENTAIRE/OPER,Le fichier à supprimer n'existe pas. Choisissez un fichier qui existe.
```

Une partie importante de ce code est semblable à celui décrit dans « Exemple de code pour Lire le caractère » ou dans « Exemple de code pour Lire la ligne ».

Seules les explications spécifiques à cet exemple figurent ici.

`V1=FILE/Dialog,Choisissez un fichier à supprimer`
Cette ligne affiche une boîte de dialogue portant le titre "Choisissez un fichier à supprimer". Vous pouvez naviguer jusqu'au fichier voulu et lorsque vous cliquez sur **Ouvrir**, PC-DMIS transmet à V1 le chemin complet du fichier sélectionné. Le reste de la routine supprime le fichier sélectionné.

Vérification de la fin d'un fichier ou d'une ligne

PC-DMIS vous permet de vérifier la fin d'un fichier via les fonctions`FDF` ou `FDL` dans un test conditionnel.

`FDF` signifie FIN DE FICHIER. Cette fonction prend un pointeur de fichier de type chaîne. Lorsqu'elle est correctement placée dans une instruction conditionnelle, elle vérifie si le pointeur de fichier a atteint la fin d'un fichier déterminé. Dans ce cas, la fonction renvoie la valeur vraie.

`EOL` signifie END OF LINE (fin de ligne). Cette fonction prend un pointeur de fichier de type chaîne. Lorsqu'elle est correctement placée dans une instruction conditionnelle, elle vérifie si

le pointeur de fichier a atteint la fin d'une ligne dans le fichier déterminé. Dans ce cas, la fonction renvoie la valeur vraie. Le fonctionnement optimal se produit dans une boucle.

La syntaxe de cette commande dans la fenêtre Édition est la suivante :

`EOF(<pointeurfichier>)` ou `EOL(<pointeurfichier>)`

Descriptions de certains composants de cette commande :

<filepointer>
Nom du pointeur de fichier que vous vérifiez.

Exemple de code pour EOF et EOL

Remarque : l'exemple de code qui suit doit être entré en mode commande dans la fenêtre de modification, et non dans la boîte de dialogue **Fichier E/S**.

Le code qui suit ouvre le fichier test.txt et le parcourt. Tant que la fin du fichier n'est pas atteinte (marquée par le code `WHILE/!EOF`), PC-DMIS parcourt le fichier caractère par caractère en attribuant un caractère à V1.

Si PC-DMIS atteint la fin d'une ligne dans le fichier, il montre le dernier caractère de cette ligne.

L'opération se répète jusqu'à ce que PC-DMIS atteigne la fin du fichier. PC-DMIS affiche ensuite un texte « Fin de fichier atteinte… ».

```
FPTR=FICHIER/OUVERT,D:\temp\test.txt,LIRE
WHILE/!EOF("FPTR")
V1=FICHIER/LIRE_CARACTÈRE,FPTR
IF/EOL("FPTR")
COMMENTAIRE/OPER,NON,« Fin de ligne atteinte. Le dernier caractère est: »
,V1
END_IF/
END_WHILE/
COMMENTAIRE/OPER,NON,« Fin de fichier atteinte...»
```