

Plus efficace avec l'éditeur ISPF - Les commandes de ligne

par Guillaume Ventre ([Développement Mainframe](#)) ([Blog](#))

Date de publication : 29/12/2007

Dernière mise à jour : 16/01/2008

ISPF (Interactive System Productivity Facility) est la composition de 4 composants majeurs : DM (Dialog Manager), PDF (Program Development Facility), SCLM (Software Configuration Library Manager) et C/S (Client/Server).

Il permet la communication entre les utilisateurs et le central. ISPF constitue l'environnement de travail du développeur Mainframe IBM et son éditeur l'outil le plus efficace pour créer ou modifier tout dataset.

Vous trouverez à la suite un rappel et/ou approfondissement sur l'utilisation des commandes de ligne de l'éditeur d'ISPF.

Pour approfondir ce sujet vous pouvez consulter la documentation officielle IBM *Interactive System Productivity Facility (ISPF) - Edit and Edit Macros - z/OS Version 1 Release 7.0 - Publication No. SC34-4820-04* (Chapitre 9 - Edit Line Commands) disponible au format PDF à l'adresse :



http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ISPZEM40/CCONTENTS?DT=2

- 1 - Généralités
 - 1.1 - commandes de ligne
 - 1.2 - Utilisation des commandes de ligne
- 2 - commandes de ligne basiques
 - 2.1 - I - Insertion
 - 2.2 - D - Suppression (Delete)
 - 2.3 - R - Répétition (Repeat)
- 3 - commandes de ligne de copie et de déplacement
 - 3.1 - C - Copie
 - 3.2 - M - Déplacement (Move)
 - 3.3 - Destination
 - 3.3.1 - A - Après cette ligne (After)
 - 3.3.2 - B - Avant cette ligne (Before)
 - 3.3.3 - O - Par dessus cette ligne (Overlay)
- 4 - commandes de ligne de décalage
 - 4.1 - () - Décalage de colonne
 - 4.2 - <> - Décalage de données
- 5 - commandes de ligne d'exclusion et d'affichage
 - 5.1 - X - Exclusion
 - 5.2 - F - Affichage des premières lignes exclues (First)
 - 5.3 - L - Affichage des dernières lignes exclues (Last)
 - 5.4 - S - Affichage des lignes d'indentation du plus petit niveau (Show)
- 6 - commandes de ligne de manipulation de texte
 - 6.1 - TS - Division de ligne (Text Split)
 - 6.2 - TF - Remontée et aboutement de ligne (Text Flow)
 - 6.3 - TE - Edition de texte (Text Enter)
 - 6.4 - LC - Bas de casse (LowerCase)
 - 6.5 - UC - Haut de casse (UpperCase)
- 7 - commandes de ligne spéciales
 - 7.1 - COLS - Affichage d'une règle horizontale
 - 7.2 - MASK - Affichage d'un masque d'insertion
 - 7.3 - BNDS - Limitation de la zone de travail (Bounds)
- 8 - commandes de ligne diverses
 - 8.1 - MD - Transformation en ligne de donnée (Make Dataline)
- 9 - Remerciements

1 - Généralités

1.1 - commandes de ligne

Pour les non initiés, une commande de ligne est une commande d'édition directement saisie sur la ligne à traiter. Elle est saisie en écrasant la séquence numérique en début de ligne.

Dans l'exemple ci-dessous, la commande de ligne **D** (DELETE) est saisie à la ligne 55 :

```

Command ==> _____ Scroll ==> CSR
000045  *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
000046  *                                                                 *
000047  *           ENVIRONMENT DIVISION                               *
000048  *                                                                 *
000049  *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
000050  ENVIRONMENT DIVISION.
000051  * +-----+
000052  * !           CONFIGURATION SECTION                           !
000053  * +-----+
000054  CONFIGURATION SECTION.
000055  *SOURCE-COMPUTER. IBM-3090 WITH DEBUGGING MODE.
000056  SOURCE-COMPUTER. IBM-3090.
000057  OBJECT-COMPUTER. IBM-3090.
000058  SPECIAL-NAMES.
000059  DECIMAL-POINT IS COMMA.
000060  * +-----+
000061  * !           INPUT-OUTPUT SECTION                           !
000062  * +-----+
000063  INPUT-OUTPUT SECTION.
    
```

1.2 - Utilisation des commandes de ligne

Il existe au plus 3 manières pour appliquer une ligne de commande sur une ligne ou un bloc de ligne. Certaines commandes de ligne s'appliquent à une seule ligne (**TS**), d'autres s'appliquent à un bloc de lignes (**D**).

En prenant comme exemple la commande de ligne **D** (DELETE), voici les 3 manières de sélectionner les lignes à traiter :

- Un **D** unique supprime une seule ligne,
- Un **D** suivi d'un nombre supprime autant de lignes que le nombre indiqué, **D3** supprime ainsi 3 lignes consécutives,
- Un double **DD** supprime le bloc de lignes compris entre le premier et le dernier **DD**.

Ci-dessous les lignes 400, 700 à 900 et 1200 à 1800 seront supprimées :

```

000300
d 0400      IF DATA = 1 THEN
000500      DO;
000600      XDATA(1) = 1
    
```

```
d3 700          XDATA(2) = 2 ;
000800          END ;
000900          ELSE
001000          IF XDATA(1) = 0 ;
001100            DO ;
dd 200          XDATA(3) = 1
001300          XDATA(4) = 0
001400          END ;
001500          ELSE
001600            ;
001700
dd 800
```

2 - commandes de ligne basiques

Les commandes de lignes les plus utilisées sont **I** (insert), **D** (delete) et **R** (repeat)

2.1 - I - Insertion

I permet d'insérer une ligne "vierge"

In permet d'insérer n lignes "vierges"

Les lignes insérées sont identifiées par "****" dans la séquence numérique en début de ligne :

```
000300
000400      Deux lignes ont été insérées après cette ligne.
****
****
000500
```

2.2 - D - Suppression (Delete)

D permet de supprimer une ligne.

Dn permet de supprimer n lignes.

Un double **DD** permet de supprimer un bloc de lignes compris entre le premier **DD** et le dernier **DD**.

2.3 - R - Répétition (Repeat)

R permet de répéter une ligne.

Rn permet de répéter une ligne n fois.

Un double **RR** permet de répéter un bloc de lignes compris entre le premier **RR** et le dernier **RR**.

Il est possible de répéter n fois un bloc de lignes en le délimitant par un **RR** simple et un **RRn**.

3 - commandes de ligne de copie et de déplacement

3.1 - C - Copie

C permet de copier une ligne.

Cn permet de copier n lignes.

Un double **CC** permet de copier un bloc de lignes compris entre le premier **CC** et le dernier **CC**.

3.2 - M - Déplacement (Move)


M permet de déplacer une ligne.

Mn permet de déplacer n lignes.

Un double **MM** permet de déplacer un bloc de lignes compris entre le premier **MM** et le dernier **MM**.

3.3 - Destination

Après avoir sélectionné le bloc de lignes à copier ou à déplacer il faut indiquer à l'éditeur où le positionner. Pour cela on utilise les commandes de ligne **A**, **B** et **O**.

 *Il est possible de faire l'inverse en positionnant tout d'abord la destination par **A**, **B** ou **O** puis en indiquant le bloc de ligne qui y sera copié ou déplacé.*

3.3.1 - A - Après cette ligne (After)

A permet d'indiquer que le bloc doit être copié ou déplacé après la ligne sélectionnée :

```
a 0400      les données sont déplacées après cette ligne.  
000500
```

3.3.2 - B - Avant cette ligne (Before)

B permet d'indiquer que le bloc doit être copié ou déplacé avant la ligne sélectionnée :

```
000300  
b 0400      les données sont déplacées avant cette ligne.
```

3.3.3 - O - Par dessus cette ligne (Overlay)

O permet d'indiquer que le bloc doit être fusionné à (aux) ligne(s) sélectionnée(s), cette fusion ne s'effectue que sur le caractère blanc (*blank*).

```
C 0200                                     /* Evol GVE */  
000300      select  
000400      when sys = 'A' then ext='CTA'
```

```
000500      when sys = 'B' then ext='CTB'  
03 600      when sys = 'C' then ext='CTC'  
000700      when sys = 'D' then ext='CTD'  
000800      when sys = 'Z' then ext='CTZ'  
000900      end
```

Résultat

```
000200                                          /* Evol GVE */  
000300      select  
000400          when sys = 'A' then ext='CTA'  
000500          when sys = 'B' then ext='CTB'  
000600          when sys = 'C' then ext='CTC' /* Evol GVE */  
000700          when sys = 'D' then ext='CTD' /* Evol GVE */  
000800          when sys = 'Z' then ext='CTZ' /* Evol GVE */  
000900      end
```

4 - commandes de ligne de décalage

4.1 - () - Décalage de colonne

La commande de ligne `)` permet de décaler d'une à n colonnes vers la droite la ligne ou le bloc de ligne sélectionné. La commande de ligne `(` permet la même chose vers la gauche :

```

-----1-----2-----3-----4-----5---
)      2007-12-01 265899513456 CERGY
)10    2007-12-02 513599556151 PARIS
000500 2007-12-02 326565651218 MASSY
))     2007-12-02 132464131515 ASNIERE
000700 2007-12-04 654512316312 NOISY
000800 2007-12-07 313214894125 EVRY
))5    2007-12-08 165126546484 CHATOU
001000 2007-12-10 545412465766 SAINT-OUEN

```

Résultat

```

-----1-----2-----3-----4-----5---
000300 2007-12-01 265899513456 CERGY
000400      2007-12-02 513599556151 PARIS
000500 2007-12-02 326565651218 MASSY
000600      2007-12-02 132464131515 ASNIERE
000700      2007-12-04 654512316311 NOISY
000800      2007-12-07 313214894121 EVRY
000900      2007-12-08 165126546484 CHATOU
001000 2007-12-10 545412465766 SAINT-OUEN

```

Une ou deux parenthèse(s) sans donnée numérique décale(nt) les lignes sélectionnées de 2 colonnes. Pour décaler d'une seule colonne il faut spécifier la valeur 1 : `)1`.

Si le décalage demandé est supérieure au LRECL du fichier, les données sont perdues, idem si le décalage demandé " rabat " les données avant la colonne 1.

4.2 - <> - Décalage de données

Le fonctionnement est identique aux commandes de ligne `(` et `)` mais diffère par le mode de décalage des données.

Ce n'est pas le bloc entier de donnée qui est décalé mais le premier " mot " de la ligne sélectionnée.

Un " mot " est représenté par une suite de caractères alphanumériques consécutifs séparées par deux espaces. Ci-dessous la ligne comporte trois mots (**2007-12-01** n'est pas un mot car il n'est pas précédé d'un espace)

```

-----1-----2-----3-----4-----5---
000300 2007-12-01 265899513456 CERGY      002
           Mot1           Mot2           Mot3

```

Un mot est décalé aussi loin qu'il peut l'être :

- sans perdre de donnée,
- sans être en deçà de la colonne 1,
- sans être au delà du LRECL du fichier,
- sans supprimer de blanc (*blank*) seul,

- sans supprimer de blanc entre apostrophe.

```
-----1-----2-----3-----4-----5---
> 2007-12-01 265899513456 CERGY 002
>5 2007-12-01 265899513456 CERGY 002
>12 2007-12-01 265899513456 CERGY 002
```

Résultat

```
-----1-----2-----3-----4-----5---
000300 2007-12-01 265899513456 CERGY 002
000400 2007-12-01 265899513456 CERGY 002
000500 2007-12-01 265899513456 CERGY 002
```

5 - commandes de ligne d'exclusion et d'affichage

5.1 - X - Exclusion

Il est possible de masquer de une à n ligne(s) dans l'éditeur ISPF. Les lignes ainsi traitées ne sont pas supprimées, elles sont juste écartées de l'affichage.

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5---
X 0300 2007-12-01 265899513456 CERGY
000400 2007-12-02 513599556151 PARIS
000500 2007-12-02 326565651218 MASSY
XX 600 2007-12-02 132464131515 ASNIERE
000700 2007-12-04 654512316312 NOISY
000800 2007-12-07 313214894125 EVRY
XX 900 2007-12-08 165126546484 CHATOU
001000 2007-12-10 545412465766 SAINT-OUEN
    
```

Résultat

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5---
- - - - - 1 Line(s) not Displayed
000400 2007-12-02 513599556151 PARIS
000500 2007-12-02 326565651218 MASSY
- - - - - 4 Line(s) not Displayed
001000 2007-12-10 545412465766 SAINT-OUEN
    
```

5.2 - F - Affichage des premières lignes exclues (First)

La commande de ligne **F** permet de faire apparaître d'une à n première(s) ligne(s) du bloc de lignes exclues par la ligne de command **X**.

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5---
F - - - - - 1 Line(s) not Displayed
000400 2007-12-02 513599556151 PARIS
000500 2007-12-02 326565651218 MASSY
F2 - - - - - 4 Line(s) not Displayed
001000 2007-12-10 545412465766 SAINT-OUEN
    
```

Résultat

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5---
000300 2007-12-01 265899513456 CERGY
000400 2007-12-02 513599556151 PARIS
000500 2007-12-02 326565651218 MASSY
000600 2007-12-02 132464131515 ASNIERE
000700 2007-12-04 654512316312 NOISY
- - - - - 2 Line(s) not Displayed
001000 2007-12-10 545412465766 SAINT-OUEN
    
```

5.3 - L - Affichage des dernières lignes exclues (Last)

La commande de ligne **L** permet de fait apparaître d'une à n dernière(s) ligne(s) du bloc de lignes exclues par la ligne de command **X**.

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5---
L - - - - - 1 Line(s) not Displayed
000400 2007-12-02 513599556151 PARIS
000500 2007-12-02 326565651218 MASSY
    
```


6 - commandes de ligne de manipulation de texte

Ces commandes de lignes sont très utiles pour la saisie ou la manipulation de texte.

6.1 - TS - Division de ligne (Text Split)

La commande de ligne TS permet de diviser une ligne en deux au point d'insertion du curseur et d'insérer une à n ligne(s) entre les deux.

Fonctionnement : le texte avant le curseur est inchangé, le texte après le curseur est positionné sur une nouvelle ligne et s'aligne au texte de la ligne suivante. Entre les 2 lignes, une ligne est insérée :

```
ts 300      Cette ligne sera divisée_en deux et une ligne va
000400     être insérée.
```

Résultat

```
000300     Cette ligne sera divisée
000400     en deux et une ligne va
000500     être insérée.
```

Si on indique une valeur numérique à la commande de ligne il y aura autant de lignes insérées que cette valeur :

```
Ts3 00     Cette ligne sera divisée_en deux et 3 ligne vont
000400     être insérées.
```

Résultat

```
000300     Cette ligne sera divisée
000400     en deux et 3 lignes vont
000500     être insérées.
```

6.2 - TF - Remontée et aboutement de ligne (Text Flow)

La commande de ligne **TF** permet de remonter et d'abouter des fragments de phrases d'un " paragraphe ".

Chaque ligne du " paragraphe " est complétée à sa suite par la ligne suivante afin de remplir toutes les colonnes. Chaque ligne aboutée est séparée par un blanc.

La fin d'un " paragraphe " est déterminée par une ligne blanche ou un changement d'indentation.

```
-----1-----2-----3-----4-----5---
tf 001 ce texte est un exemple
000002 d utilisation de la ligne
000003 de commande TF (Text Flow)
000004 qui permet de remonter des
000005 fragments de phrases en un
000006 seul paragraphe.
```

Résultat

```
-----1-----2-----3-----4-----5---
000001 ce texte est un exemple d utilisation de la ligne de
```

Résultat


```
000002 commande TF (Text Flow) qui permet de remonter des
000003 fragments de phrases en un seul paragraphe.
```

Une valeur numérique ajoutée à cette commande de ligne indique la colonne à laquelle doit être tronqué le paragraphe, par exemple **tf40** :

```
-----1-----2-----3-----4-----5---
000001 ce texte est un exemple d utilisation de
000002 la commande de ligne TF (Text Flow) qui
000003 permet de remonter des fragments de
000004 phrases en un seul paragraphe.
```

Si une ligne aboutée finit par les caractères suivants, la séparation entre les 2 lignes aboutées est constituée de 2 blancs au lieu d'un :

: , . , ? , \$, . " , ? " , \$ " , .) , ?) , \$)

 *Il est possible d'utiliser cette commande de ligne d'une manière détournée pour obtenir une mise à la ligne de chaque " mot " d'un paragraphe, pour cela il suffit d'indiquer une valeur numérique inférieure à la taille d'un " mot " :*

```
-----1-----2-----3-----4-----5---
tf2 01 0001 0002 0003 0004 0005
```

Résultat

```
-----1-----2-----3-----4-----5---
000001 0001
000002 0002
000003 0003
000004 0004
000005 0005
```

6.3 - TE - Edition de texte (Text Enter)

La commande de ligne **TE** permet la saisie de texte sans se soucier des fins et des retours de ligne.

6.4 - LC - Bas de casse (LowerCase)

La commande de ligne **LC** permet de transformer en minuscule le texte d'une à n ligne(s) :

```
-----1-----2-----3-----4-----5---
lc 001 CE TEXTE EST EN MAJUSCULE IL VA PASSER EN MINUSCULE
000002 CE TEXTE RESTE EN MAJUSCULE
lcc 03 CE TEXTE EST EN MAJUSCULE IL VA PASSER EN MINUSCULE
000004 CE TEXTE EST EN MAJUSCULE IL VA PASSER EN MINUSCULE
lcc 05 CE TEXTE EST EN MAJUSCULE IL VA PASSER EN MINUSCULE
```

Résultat

```
-----1-----2-----3-----4-----5---
000001 ce texte est en majuscule il va passer en minuscule
000002 CE TEXTE RESTE EN MAJUSCULE
000003 ce texte est en majuscule il va passer en minuscule
000004 ce texte est en majuscule il va passer en minuscule
000005 ce texte est en majuscule il va passer en minuscule
```

6.5 - UC - Haut de casse (UpperCase)

La commande de ligne **UC** permet de transformer en majuscule le texte d'une à n ligne(s) :

```
-----1-----2-----3-----4-----5---  
uc 001 ce texte est en minuscule il va passer en majuscule  
00002 ce texte reste en minuscule  
u3 003 ce texte est en minuscule il va passer en majuscule  
00004 ce texte est en minuscule il va passer en majuscule  
00005 ce texte est en minuscule il va passer en majuscule
```

Résultat

```
-----1-----2-----3-----4-----5---  
00001 CE TEXTE EST EN MINUSCULE IL VA PASSER EN MAJUSCULE  
00002 ce texte reste en minuscule  
00003 CE TEXTE EST EN MINUSCULE IL VA PASSER EN MAJUSCULE  
00004 CE TEXTE EST EN MINUSCULE IL VA PASSER EN MAJUSCULE  
00005 CE TEXTE EST EN MINUSCULE IL VA PASSER EN MAJUSCULE
```

7 - commandes de ligne spéciales

Les commandes de lignes abordées dans ce paragraphe insèrent une ligne temporaire à l'endroit où la commande de ligne est saisie. Ces lignes ne font pas partie des données et ne sont jamais sauvegardées.

7.1 - COLS - Affichage d'une règle horizontale

Cette commande de ligne permet d'afficher une règle horizontale (très pratique pour trouver la position des données) :

```
000400 2007-12-02 513599556151 PARIS
cols 0 2007-12-02 326565651218 MASSY
000600 2007-12-04 654512316312 NOISY
000700 2007-12-08 165126546484 CHATOU
000800 2007-12-10 545412465766 SAINT-OUEN
```

Résultat

```
000400 2007-12-02 513599556151 PARIS
=COLS> ----+----1----+----2----+----3----+----4----+----5---
000500 2007-12-02 326565651218 MASSY
000600 2007-12-04 654512316312 NOISY
000700 2007-12-08 165126546484 CHATOU
000800 2007-12-10 545412465766 SAINT-OUEN
```

Dans ce fichier l'identifiant est en colonne 12 et la ville en position 25.

7.2 - MASK - Affichage d'un masque d'insertion

Lors de l'insertion d'une ligne, le contenu du masque d'insertion est repris dans la nouvelle ligne. S'il ne contient rien la ligne insérée est vierge. Pour changer le contenu du masque d'insertion il suffit de saisir le texte souhaité.

Exemple d'utilisation, commentaire dans le cadre de la maintenance d'un programme cobol :

Saisie de la commande de ligne MASK

```
----+----1----+----2----+----3----+----4----+----5---
mask 1
000002          IF FIN-ANORMALE
```

Affichage du masque d'insertion

```
----+----1----+----2----+----3----+----4----+----5---
=MASK>
000002          IF FIN-ANORMALE
```

Mise à jour du masque d'insertion

```
----+----1----+----2----+----3----+----4----+----5---
=MASK> GV0712
000002          IF FIN-ANORMALE
```

Insertion d'une ligne

```
----+----1----+----2----+----3----+----4----+----5---
=MASK> GV0712
000002          IF FIN-ANORMALE
000003          PERFORM SYSOUT-GESTIO-FIN-ANORMALE
000004          CALL WS-SY00066 USING WS-ABEND
000005          ELSE
000006          PERFORM SYSOUT-GESTIO-FIN-NORMALE
```

Insertion d'une ligne

```
i 0007          MOVE WS-CODE-RETOUR TO RETURN-CODE
00008          END-IF .
```

Résultat

```
-----1-----2-----3-----4-----5---
=MASK> GV0712
00002          IF FIN-ANORMALE
00003          PERFORM SYSOUT-GESTIO-FIN-ANORMALE
00004          CALL WS-SY00066 USING WS-ABEND
00005          ELSE
00006          PERFORM SYSOUT-GESTIO-FIN-NORMALE
00007          MOVE WS-CODE-RETOUR TO RETURN-CODE
' ' ' ' ' GV0712
00008          END-IF .
```

Maintenance

```
-----1-----2-----3-----4-----5---
=MASK> GV0712
00002          IF FIN-ANORMALE
00003          PERFORM SYSOUT-GESTIO-FIN-ANORMALE
00004          CALL WS-SY00066 USING WS-ABEND
00005          ELSE
00006          PERFORM SYSOUT-GESTIO-FIN-NORMALE
00007          MOVE WS-CODE-RETOUR TO RETURN-CODE
' ' ' ' ' GV0712
00008          MOVE 'FIN NORMAL' TO DISP-STRING
00008          END-IF .
```

<Enter>

```
-----1-----2-----3-----4-----5---
=MASK> GV0712
00002          IF FIN-ANORMALE
00003          PERFORM SYSOUT-GESTIO-FIN-ANORMALE
00004          CALL WS-SY00066 USING WS-ABEND
00005          ELSE
00006          PERFORM SYSOUT-GESTIO-FIN-NORMALE
00007          MOVE WS-CODE-RETOUR TO RETURN-CODE
00008 GV0712    MOVE 'FIN NORMAL' TO DISP-STRING
00009          END-IF .
```

7.3 - BNDS - Limitation de la zone de travail (Bounds)

La commande de ligne **BNDS** permet de limiter la zone de travail. Par défaut elle commence à la colonne 1 et se termine au LRECL du fichier édité.

Les limites sont représentées par les signes < et > respectivement pour la limite inférieure (gauche) et la limite supérieure (droite).

Pour modifier ces limites il suffit de ressaisir les limites.

```
-----1-----2-----3-----4-----5---
bnds 1
00002          IF FIN-ANORMALE
```

Résultat

```
-----1-----2-----3-----4-----5---
=BNDS> <          >
00002          IF FIN-ANORMALE
```


Cette commande de ligne fonctionne avec les commandes de ligne de décalage et les commandes de ligne de manipulation de texte **TS**, **TF** et **TE**. Elle fonctionne aussi avec les commandes principales **FIND**, **CHANGE**, **EXCLUDE** et **SORT** que nous verrons ultérieurement et peut être aussi utilisée avec un déplacement **M** ou une copie **C** si utilisé conjointement avec la destination **O**.

Exemple, décalage de 5 colonnes uniquement à partir de la ville :

```

-----1-----2-----3-----4-----5---
000400 2007-12-02 513599556151 PARIS
=BND> < >
000500 2007-12-02 326565651218 MASSY
)) 600 2007-12-04 654512316312 NOISY
000700 2007-12-08 165126546484 CHATOU
)) 800 2007-12-10 545412465766 SAINT-OUEN

```

Résultat

```

-----1-----2-----3-----4-----5---
000400 2007-12-02 513599556151 PARIS
=BND> < >
000500 2007-12-02 326565651218 MASSY
000600 2007-12-04 654512316312 NOISY
000700 2007-12-08 165126546484 CHATOU
000800 2007-12-10 545412465766 SAINT-OUEN

```

8 - commandes de ligne diverses

8.1 - MD - Transformation en ligne de donnée (Make Dataline)

La commande de ligne **MD** permet de transformer une ligne temporaire en ligne de donnée.

Par exemple une ligne temporaire **NOTE**, **COLS** ou **MSG** peuvent devenir des données du fichier.

 *Les lignes temporaires **PROF**, **BNDS**, **TABS** et **MASK** ne peuvent être transformées.*

9 - Remerciements

Je tiens à remercier tout particulièrement **olsimare**, **RideKick**, **Millie**, **Gorgonite** et **al1_24** pour leur relecture attentive.

