

//EXCLUSION PAR CHAINE DE CARACTERE

La simple utilisation de la commande primaire EXCLUDE (ou X) suivie d'une chaîne de caractère peut se révéler très utile.

Supposons par exemple, que vous vouliez passer en revue toutes les lignes de commentaire d'un source COBOL afin de supprimer les commentaires obsolètes, l'enchaînement des commandes suivantes affichera les seules lignes de commentaire :

```
COMMAND => X ALL '*' 7;FLIP
```

Que l'on peut traduire par : exclure (masquer temporairement) toutes les lignes du fichier source qui ont un astérisque (*) en colonne 7, puis, inverser l'exclusion.

//EXCLUSION GLOBALE SUIVIE D'UNE RECHERCHE GLOBALE

Au lieu d'utiliser une chaîne de caractère avec la commande primaire EXCLUDE, c'est l'ensemble du fichier en édition que l'on masque. Puis, en utilisant une chaîne de caractère avec la commande FIND (ou F) ALL, les lignes cibles sont révélées.

On obtient le même résultat que l'exemple précédent avec l'enchaînement de commandes :

```
COMMAND => X ALL;F ALL '*' 7
```

Que l'on peut traduire par : exclure (masquer temporairement) toutes les lignes du fichier en édition. Puis, rechercher et rendre visible toutes les lignes qui ont un astérisque (*) en colonne 7.

//SUPPRESSION DE LIGNE A PARTIR DE LEUR STATUT EXCLU (X) OU NON-EXCLU (NX)

C'est une méthode très utile pour « nettoyer » certains fichiers. Elle nécessite l'utilisation de la commande primaire DELETE (ou DEL) en conjonction avec le paramètre X (exclu) ou NX (non-exclu).

En reprenant l'exemple du début, si l'on souhaite finalement supprimer tous les commentaires, il suffira de supprimer toutes les lignes non-exclues, pour cela on saisira la commande suivante :

```
COMMAND => DEL ALL NX
```

Que l'on peut traduire par : supprimer toutes les lignes visibles (les commentaires)

On peut évidemment faire plus rapide, si le but final est de supprimer toutes les lignes de commentaire d'un source COBOL, on saisira directement :

```
COMMAND => X ALL '*' 7;DEL ALL X
```

Que l'on peut traduire par : exclure (masquer temporairement) toutes les lignes du fichier source qui ont un astérisque (*) en colonne 7, puis supprimer les lignes masquées.

//LIMITER LA PORTEE DE LA COMMANDE PRIMAIRE "CHANGE" PAR EXCLUSION DE LIGNES AU LIEU D'ETIQUETTES

Les étiquettes (ou labels), qui seront traitées dans un article spécifique, permettent de restreindre l'étendu d'une commande à un bloc de ligne qu'elles déterminent, par exemple :

```
COMMAND => C ALL ' ' 1 '---' .A .B
```

Que l'on peut traduire par : changer tous les doubles espaces en colonne 1 par '---' dans le bloc de ligne .A - .B.

Une condition préalable à cette commande est bien entendu le placement des étiquettes .A sur la première ligne et .B sur la dernière ligne du bloc de ligne souhaité.

L'exclusion peut se substituer à l'utilisation des étiquettes dans le cas de la commande primaire CHANGE.

Pour limiter la portée d'une commande primaire CHANGE aux seules lignes exclues, il suffit d'utiliser la syntaxe suivante :

```
COMMAND => C ALL ' ' 1 '---' X
```

Que l'on peut traduire par : changer tous les doubles espaces en colonne 1 des lignes exclues par '---'.

L'intérêt principal de cette substitution est que le CHANGE peut être réalisé sur plusieurs zones non consécutives, il suffit pour cela d'utiliser n'importe quelle méthode d'exclusion vue précédemment et/ou l'utilisation des commandes de ligne X ou XX/XX.

//RETOUR A LA NORMALE

Pour réafficher les lignes masquées, il suffit de faire un RESET (ou RES) en ligne de commande :

```
COMMAND => RES
```