Table des matières

1
GÉNÉRALITÉS SUR LE LANGAGE COBOL

1.1 ORIGINE DU LANGAGE

COBOL, ***CO****mmon* ***B****usiness* ***O****riented* ***L****anguage*, est un langage de gestion appliqué au traitement informatisé des services de paye, de facturation, de comptabilité, etc.

Surtout utilisé sur des ordinateurs centraux, *mainframes*, COBOL est de plus en plus développé sur des micro-ordinateurs compte tenu des coûts élevés d'utilisation des ordinateurs centraux.

Historique du COBOL

Dans les années 1950, chaque compagnie d'ordinateurs développait son propre langage de programmation. Il devint donc nécessaire de créer un langage utilisable d'un ordinateur à l'autre quelque soit le constructeur, d'où le mot anglais *common.*

Le langage COBOL fut créé en 1959, et, dès les années 1960, plusieurs versions firent leur apparition.

En 1974, des standards furent établis afin de créer une version appelée COBOL ANSI (*American National Standards Institute*) qui, avec de légères modifications, est adaptable à toute marque d'ordinateur.

Avec l'avènement des micro-ordinateurs, la version ANSI 74 fut remaniée afin de doter COBOL *d'instructions interactives*. IBM fut le concepteur de cette nouvelle version (1982) appelée PC-COBOL (*Personal COBOL*).

En 1985, la version ANSI 74 fut révisée pour devenir la version COBOL ANSI 85 qui est encore aujourd'hui un standard.

À la suite du PC-COBOL, la compagnie Microsoft développa quelques versions du langage appelées Microsoft COBOL utilisables sur des systèmes d'exploitation tels que DOS et OS/2. Par la suite, Microsoft vendit ses droits d'exploitation à la compagnie Micro Focus. Depuis, cette compagnie n'a pas cessé de l'améliorer afin qu'il soit utilisable sur n'importe quel système d'exploitation, que ce soit DOS, Windows 3.1, Windows 95 ou Windows NT, Unix ou OS/2, et qu'il soit d'autre part, compatible avec tous les « dialectes » COBOL utilisés sur ordinateurs centraux tels que IBM OS/VS ou IBM VS COBOL II.

Noter que ces dernières versions du langage utilisées sur ordinateurs centraux *ne sont pas interactives*. De plus, Micro Focus a développé des outils permettant d'appliquer des méthodes de programmation *orientée objet*.

Avantages et désavantages du COBOL

C'est un langage très facile à lire. Ainsi, un programme en COBOL peut aisément être modifié par un autre programmeur.

Il est structuré et performant car ses instructions comportent une importante variété d'options. Enfin, le programmeur a une grande liberté dans le choix des mots, jusqu'à 30 caractères pour un nom de variable.

Le principal *désavantage* du COBOL est la *longueur* de sa codification.

Notes préliminaires

Dans ce manuel, les logiciels utilisés comme référence sont la version Micro Focus COBOL Workbench, V3.4 et V4.0 pour Windows. La mention **(MF)** signifie que l'item qui précède est spécifique à Micro Focus COBOL.

Étant donné que le langage COBOL est surtout utilisé sur les ordinateurs centraux, les principales différences de programmation sur un ordinateur central seront annotées **(VSC2)**, faisant référence au « dialecte » IBM VS COBOL II.

Dans ce manuel, l'utilisation de plusieurs termes anglais a été conservée, car tous les compilateurs COBOL affichent les messages d'erreurs dans cette langue et il est de première importance d'en saisir le sens.

Certaines clauses du langage COBOL ont été volontairement omises, celles-ci étant peu utilisées.

1.2 OBSERVATION D'UN PROGRAMME COBOL

La structure générale d'un programme COBOL comprend quatre divisions :

1. IDENTIFICATION DIVISION ;
2. ENVIRONMENT DIVISION ;
3. DATA DIVISION ;
4. PROCEDURE DIVISION.

Voir le programme en Annexe A.

IDENTIFICATION

Cette division présente principalement des généralités sur le programme et le programmeur.

ENVIRONMENT

Cette partie du programme est formée de deux sections :

- CONFIGURATION : désigne l'équipement utilisé ;
- INPUT-OUTPUT : définit les fichiers d'entrée et de sortie utilisés.

DATA

Dans cet exemple, le programme comporte deux sections :

- FILE : décrit les fichiers utilisés ;
- WORKING-STORAGE : contient la définition des champs de travail (77) et la description de chaque ligne à imprimer.

PROCEDURE

Cette division regroupe les instructions et elle est subdivisée en paragraphes identifiés par un nom de paragraphe.

1.3 CARACTÈRES UTILISÉS EN COBOL

Caractères de ponctuation : `( ) , . ; = ⊬`

Caractères de relation : `> < = <= >=` et `<>` **(MF)**

Opérateurs arithmétiques : `+ - * / **`

Caractères d'édition : `. + $ * -`

Caractères alphabétiques : `A` à `Z` et `a` à `z`

Caractères numériques : `0` à `9`

Délimiteurs de chaîne de caractères : " ou ' (apostrophe), selon le compilateur.

Formation des mots en COBOL

En COBOL, il y a deux types de mots : les **mots réservés**, spécifiques au langage (la liste se trouve dans le menu d'aide du logiciel Micro Focus) et les **mots choisis par le programmeur**.

Règles de formation des mots choisis par le programmeur

1. La longueur d'un mot est de 1 à 30 caractères ;
2. Un mot peut utiliser :

   Les nombres de 0 à 9 ;
   Les lettres majuscules, de A à Z ;
   Les lettres minuscules, de a à z ;
   Le tiret -.
3. Un mot ne doit ni commencer ni se terminer par un tiret ;
4. Un mot ne doit pas contenir d'espace ;
5. Le mot choisi ne doit pas être un mot réservé :

   Par exemple : `DIVISION`, `READ`, `MOVE`, `TO`, `DATE`, `NOTE`, `TABLE` ;
6. Il doit contenir au moins une lettre, sauf les noms de paragraphes et de sections ;
7. Un mot doit être unique sauf s'il est un item qualifié ;
8. Les lettres minuscules sont équivalentes aux majuscules sauf dans un littéral non numérique (chaîne de caractères).

Sortes de mots

FILE-NAME :	nom de fichier.
RECORD-NAME :	nom d'un enregistrement.
DATA-NAME :	nom d'une variable.
IDENTIFIER :	nom d'un champ occupé par une variable qui peut être simple, indicée, indexée ou qualifiée.

CONDITION-NAME : nom donné à une variable qui contient une valeur spécifique. Il permet d'identifier le contenu. Il se définit au niveau 88 et sera décrit avec l'instruction IF.

PARAGRAPH-NAME : nom de paragraphe.

QUALIFIER-NAME : item qualifié qui est utilisé pour différencier deux variables et plus qui portent le même nom. Il doit être utilisé avec OF ou IN dans une instruction.

Exemple

```
01  DATE-JOUR.
    05  JOUR   PIC 99.
    05  MOIS   PIC 99.
    05  ANNEE  PIC 99.
01  DATE-HIER.
    05  JOUR   PIC 99.
    05  MOIS   PIC 99.
    05  ANNEE  PIC 99.

ADD JOUR OF DATE-JOUR TO TOTAL.
ADD JOUR IN DATE-JOUR TO TOTAL.
```

■

SECTION-NAME : nom de section.

PROGRAM-NAME : nom du programme.

SCREEN-NAME (MF) : nom d'un item saisi ou affiché à l'écran.

INDEX-NAME : nom d'un index.

SUBSCRIPT : nom d'un indice.

1.4 CONSTANTES

Un champ dont la valeur ne peut changer lors de l'exécution du programme s'appelle une constante.

Types de constantes

Il y a trois sortes de constantes : les littéraux numériques, les littéraux non numériques et les constantes figuratives.

Littéraux numériques

Ils sont surtout utilisés pour les opérations arithmétiques.

Ils sont formés des signes + (plus), - (moins), du point décimal et des caractères 0 à 9.

Ils comportent un maximum de 18 caractères (excluant le signe et le point).

Le signe + ou - est placé en avant du nombre, le signe + étant facultatif. Un seul point décimal est permis.

Littéraux non numériques

Ils servent pour toute opération autre qu'arithmétique. Ils sont surtout utilisés pour imprimer des titres.

On les place toujours entre guillemets ou entre apostrophes. MF accepte les deux, tandis que VSC2 n'accepte que l'apostrophe.

Ils peuvent être formés de 1 à 160 caractères mis à part les guillemets ou les apostrophes.

Tous les caractères ASCII sont permis. Noter qu'un ordinateur central utilise généralement une table de caractères différente appelée EBCDIC.

Pour insérer le caractère " ou ', il suffit de coder deux fois le caractère de façon consécutive.

Exemple

coder "" pour obtenir "
coder 'L''ECOLE' pour obtenir L'ECOLE

■

Constantes figuratives

Ce sont des mots réservés représentant une valeur spécifique. Le singulier et le pluriel sont équivalents.

Les plus utilisées sont :

- ZERO, ZEROS, ZEROES ;
- LOW-VALUE, LOW-VALUES (00 est la plus petite valeur de la table de caractères utilisée) ;
- HIGH-VALUE, HIGH-VALUES (FF est la plus grande valeur de la table de caractères utilisée) ;
- QUOTE, QUOTES (" sur MF et ' sur VSC2) ;
- SPACE, SPACES ;
- ALL (s'utilise avec un littéral non numérique ou une autre constante figurative).

Exemple

```
MOVE ALL "*" TO CHAMP.
MOVE ALL QUOTE TO CHAMP.
```

■

1.5 RÈGLES DE PONCTUATION

1. Le point (.) doit être placé à la fin de chaque phrase. Il peut changer le sens de la logique et il est souvent la source de plusieurs *bogues*.

> **Note**
>
> On tente de plus en plus à éliminer l'emploi du point dans les instructions et cette omission est permise selon certaines conditions qui seront vues au chapitre 4.

2. Le point (.), la virgule (,) et le point-virgule (;) servent à séparer les divers éléments du texte. Il n'y a pas d'espace devant, mais il y a en un après.
3. La virgule (,) et le point-virgule (;) sont utilisés dans le seul but de clarifier le texte. Ils ne changent pas le sens de la logique.
4. Les parenthèses doivent apparaître par paire, une à gauche et l'autre à droite.
5. En COBOL, il faut au moins un espace pour séparer deux mots. Dans le texte, le symbole b/ sera utilisé pour représenter cet espace.

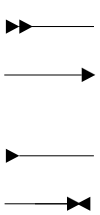
   Deux espaces ou plus sont traités comme un seul espace sauf dans les littéraux non numériques. Ils permettent de dégager le texte.
6. Les littéraux non numériques doivent être précédés d'un espace et suivis d'un espace ou des caractères point, virgule ou point-virgule.

Exemple

```
b/"COBOL"b/
b/"COBOL".
```

■

1.6 FORMAT GÉNÉRAL DES ÉNONCÉS COBOL



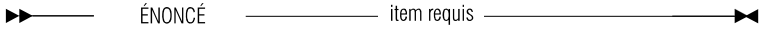
désigne le début de l'énoncé

indique que l'énoncé continue sur la ligne suivante

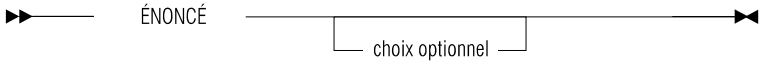
signifie la continuation de la ligne précédente

désigne la fin de l'énoncé

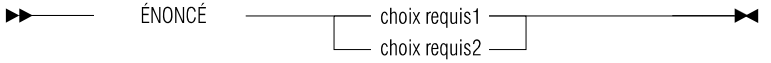
Les items obligatoires apparaissent *sur* la ligne principale.



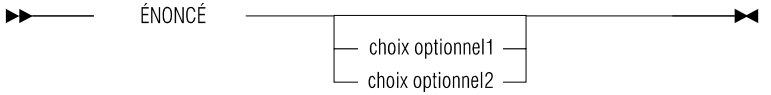
Les items optionnels apparaissent *sous* la ligne principale.



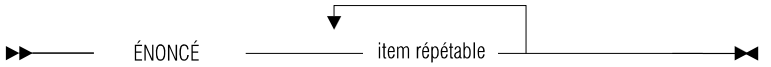
Un seul choix doit être fait lorsque deux ou plusieurs items apparaissent verticalement.



Si les choix sont optionnels, ils apparaissent *en dessous* de la ligne principale.



Une flèche gauche au-dessus de la ligne principale indique que cet item peut être répété.



Les mots en majuscules sont des mots réservés au langage COBOL, et les mots en minuscules doivent être choisis par le programmeur.