

# Mise au point

## 1. GENERALITES

Le module de mise au point (**Debugging**) de COBOL offre au programmeur des moyens de contrôler l'exécution de son algorithme. Ils consistent en :

- ◆ des lignes de mise au point (**D** en colonne 7)
- ◆ des sections déclaratives (USE FOR DEBUGGING)
- ◆ un registre spécial (DEBUG-ITEM) géré par le système et accessible au pg

Ces outils seront mis en (ou hors) fonction grâce à des inverseurs (commutateurs, "switchs") logiques, soit lors de la compilation, soit lors de l'exécution.

Ce module est classé obsolète en ANS85.

## 2. INVERSEURS

L'**inverseur de compilation** est constitué par la clause WITH DEBUGGING MODE du paragraphe SOURCE-COMPUTER. Sa présence dans le pg entraîne la compilation de toutes les instructions de mise au point (D en col. 7 et sections de mise au point) y figurant. Son absence implique que les instructions précédentes seront ignorées du compilateur (i.e. considérées comme des commentaires).

L'**inverseur d'exécution** est un switch logique mis en ou hors fonction lors de l'appel de l'exécutable, grâce à un ordre ou un paramètre particulier.

Si il est ON, les phrases USE FOR DEBUGGING sont validées; sinon (OFF), elles sont inhibées.

## 3. SECTIONS DECLARATIVES DE MISE AU POINT

Ce sont des sections spécifiques placées dans la partie déclaratives, au début de celle-ci (donc **avant** toute éventuelle section de traitement des erreurs d'entrée-sortie), identifiées par une phrase USE FOR DEBUGGING ... et constituant des **blocs autonomes** (pas de référence d'une procédure déclarative à une procédure non-déclarative, dans un sens ou dans l'autre). Cf chap.5 "Gestion des Fichiers".

USE FOR DEBUGGING ON { [ALL REFERENCES OF] nd } ...  
 { {nom-fichier} ...  
 {nom-procédure} ...  
ALL PROCEDURES } ...

- ◆ Toute variable apparaissant après ON peut
  - n'apparaître que **dans une seule phrase** USE
  - n'y figurer qu'**une seule fois**.
- ◆ Si ALL PROCEDURES est cité dans une phrase USE, aucun nom de procédure ne peut figurer dans aucune phrase USE.

- ◆ Si nom-donnée contient une clause OCCURS ou est subordonné à une rubrique la contenant, on ne spécifie pas l'indice ou l'index normalement nécessaire.

### Règles d'application

- ◆ Lorsqu'un nom-fichier est spécifié, la section déclarative est exécutée **après** exécution valide d'une instruction OPEN, CLOSE, READ, START ou DELETE.
- ◆ Lorsqu'un nom-procédure est spécifié, la section déclarative est exécutée immédiatement **avant** chaque exécution de la procédure citée; s'il s'agit de ALL PROCEDURES, cette règle s'applique pour chaque procédure du pg.
- ◆ Lorsqu'un nom-donnée est spécifié **seul**, la section déclarative est exécutée:
  - a) **avant** toute instruction WRITE ou REWRITE qui cite explicitement le nom-donnée (cas de l'article), ou après la locution FROM;
  - b) **après** chaque initialisation, modification, évaluation de nom-donnée quand cette rubrique apparaît après VARYING, AFTER, UNTIL dans une instruction PERFORM;
  - c) **après** toute autre instruction qui cite explicitement nom-donnée ET entraîne la **modification** de son contenu.
- ◆ Lorsqu'un nom-donnée est spécifié précédé de la locution ALL REFERENCES OF, la section déclarative est exécutée :
  - a) idem a) ci-dessus
  - b) idem b) ci-dessus
  - c) **avant** transfert du contrôle quand nom-donnée est cité dans la locution DEPENDING ON d'une instruction GO TO;
  - d) **après** execution de toute autre instruction qui cite explicitement nom-donnée (même si elle ne modifie pas son contenu).

#### Remarques :

- Si nom-donnée est spécifié dans une phrase qui n'est ni exécutée ni évaluée, la section déclarative n'est PAS exécutée.
- Pour une instruction donnée, la déclarative n'est exécutée qu'une seule fois, quel que soit le nombre de fois où nom-donnée est cité dans l'instruction; pour une itérative (PERFORM), elle est exécutée à chaque iteration.

## 4. DEBUG-ITEM

Ce registre spécial, généré automatiquement, est associé à chaque exécution d'une section déclarative de mise au point; il fournit des informations sur les conditions qui ont provoqué l'exécution de cette dernière; sa description **implicite** est la suivante :

```

01  DEBUG-ITEM.
    02  DEBUG-LINE          PIC X(6) .
    02  FILLER              PIC X    VALUE SPACE.
    02  DEBUG-NAME         PIC  X(30) .
    02  FILLER              PIC X    VALUE SPACE.
    02  DEBUG-SUB-1        PIC S9999 SIGN LEADING SEPARATE.
    02  FILLER              PIC X    VALUE SPACE.
    02  DEBUG-SUB-2        PIC S9999 SIGN LEADING SEPARATE..
    02  FILLER              PIC X    VALUE SPACE.
    02  DEBUG-SUB-3        PIC S9999 SIGN LEADING SEPARATE.
    02  FILLER              PIC X    VALUE SPACE.
    02  DEBUG-CONTENTS     PIC X(n) .
  
```

Tous les noms cités sont des **mots réservés**.

- `DEBUG-CONTENTS` a une taille suffisante pour contenir la donnée définie par les règles ci-après.
- Le contenu de `DEBUG-ITEM` est **mis à blanc avant** chaque exécution d'une section déclarative; sa mise à jour s'exécute suivant les règles de `MOVE` (sans aucune conversion pour `DEBUG-CONTENTS`).
- `DEBUG-LINE` contient un numéro de ligne-source (listing).
- `DEBUG-NAME` contient les 30 premiers caractères du nom qui a provoqué l'exécution de la section déclarative.
- Si la donnée qui provoque l'exécution de la déclarative est indiquée ou indexée, le rang de chaque niveau est placé respectivement dans `DEBUG-SUB-1`, `DEBUG-SUB-2`, `DEBUG-SUB-3`.

### Règles d'application

#### A - Exécution résultant d'un nom-donnée

`DEBUG-LINE` contient le n° de ligne de l'instruction origine,

`DEBUG-NAME` contient le nom-donnée,

`DEBUG-CONTENTS` contient le contenu de cette rubrique.

#### B - Exécution résultant d'un nom-fichier

`DEBUG-LINE` contient le n° de ligne de l'instruction origine,

`DEBUG-NAME` contient le nom-fichier,

`DEBUG-CONTENTS` contient l'article lu en entier.

#### C - Exécution résultant d'une procédure citée dans `PERFORM`

`DEBUG-LINE` contient le n° de ligne de l'instruction `PERFORM`,

`DEBUG-NAME` contient le nom-procédure appelé,

`DEBUG-CONTENTS` contient "`PERFORM LOOP`".

#### D - Exécution résultant d'un nom-procédure rencontré séquentiellement

`DEBUG-LINE` contient le n° de ligne de l'instruction précédente,

`DEBUG-NAME` contient nom-procédure,

`DEBUG-CONTENTS` contient "`FALL THROUGH`".

#### E - Exécution résultant d'un nom-procédure cité dans une locution `INPUT` ou `OUTPUT` d'une instruction `SORT` ou `MERGE`

`DEBUG-LINE` contient le no de ligne de l'instruction `SORT` ou `MERGE`,

`DEBUG-NAME` contient le nom-procédure d'entrée (ou de sortie),

`DEBUG-CONTENTS` contient suivant le cas "`SORT INPUT`", "`SORT OUTPUT`"  
ou "`MERGE OUTPUT`".

#### F - Exécution résultant d'un nom-procédure cité dans `GO TO`

`DEBUG-LINE` contient le n° de ligne de l'instruction `GO TO`,

`DEBUG-NAME` contient le nom-procédure,

`DEBUG-CONTENTS` est à blanc.

#### G - Exécution résultant du premier nom-procédure cité dans le pg

`DEBUG-LINE` contient le n° de la première instruction;

`DEBUG-NAME` contient nom-procédure,

`DEBUG-CONTENTS` contient "`START PROGRAM`".

#### H - Exécution résultant d'un nom-procédure cité dans `ALTER`

`DEBUG-LINE` contient le n° de ligne de l'instruction `ALTER`,

`DEBUG-NAME` contient nom-procédure,

`DEBUG-CONTENTS` contient le nom de la procédure à exécuter (associée à `TO`).

obtenues par une recherche et de les écrire dans un fichier `PASTE.TXT`.