

# Chapitre 6

# Rapports Imprimés

- Introduction
- Le ReportWriter

## Introduction (1)

---

- Concept de rapport
  - Un rapport est le résultat d'un traitement
  - Il est composé à la fois **de pages** et de **blocs logiques**.
    - Description du contenu de la page
      - Entête, numéro de page, ...
    - Description de chaque bloc logique
      - Titre, détails, ...
  - Pour chacun de ces composants, il existe une entête et une finalisation.

# Introduction (2)

- Exemple: un bon de commande (blocs logiques)

- Blocs logiques
  - En-tête
    - Identification du client (numéro, nom, adresse, ...)
    - Identification de la commande (numéro, date, ...)
  - Corps de la commande
    - Ligne de détail (numéro produit, quantité, PU, PT)
  - Finalisation
    - Total à facturer, date d'échéance, ...
- Structure en page
  - En-tête
    - Nom de la société, adresse, ...
  - Finalisation
    - Numéro de page, ...

# Introduction (3)

RELEVÉ DES FABRICATIONS DU xx/xx/xx AU xx/xx/xx	Groupe de lignes (bloc)	
-----		<b>PH</b> page heading
USINE: xxxxxxxxxxxxxx		<b>CH USINE</b>
ATELIER xxx		Control. head.
xxxxxxxxxxxxxxxxxxxxxxxx	xxxxxx	<b>CH ATELIER</b>
(produit)	(quant)	<b>DE - Détail</b>
xxxxxxxxxxxxxxxxxxxxxxxx	xxxxxx	<b>DE - Détail</b>
xxxxxxxxxxxxxxxxxxxxxxxx	xxxxxx	<b>DE - Détail</b>
xxxxxxxxxxxxxxxxxxxxxxxx	xxxxxx	<b>DE - Détail</b>
	-----	<b>DE - Détail</b>
TOTAL ATELIER :	xxxxxx	<b>CF ATELIER</b>
	-----	Control foot.
TOTAL USINE :	xxxxxx	<b>CF USINE</b>
-----		
DATE : xx/xx/xx PAGE xxx		<b>PF</b> page footing

# Introduction (4)

---

- Que permet COBOL ?
  - Le ReportWriter permet de:
    - décrire le rapport imprimé sous la forme de blocs logiques
    - décrire la page physique
  - En outre:
    - Il réalise certaines totalisations (courante dans les rapports comme le total à payer, ...)
    - Gère automatiquement la mise en page (rupture sur certaines données, ...)
    - Se rapproche du paradigme **déclaratif** (comme SQL)

# Introduction (5)

---

- Mise en page
  - 3 emplacements possibles pour un groupe:
    - haut de la page
      - Contient les titres de rapport et de page
    - corps de la page
      - Contient les blocs logiques
    - bas de page
      - Contient les fin de page et de rapport
  - Réalisation
    - Le garnissage: préparation des lignes dans un buffer
    - Le choix d'une page et génération de la mise en page

# Introduction (6)

---

- Positionnement du groupe de lignes (toutes les lignes d'un groupe apparaissant sur une même page)
- Positionnement de chaque ligne (de manière absolue ou relative)
- Structure logique
  - Utilisation d'un indicatif de structure dont chaque élément correspond à une décomposition logique.
    - ex: numéro d'usine, numéro d'atelier
  - Une rupture sur un élément de cet indicatif entraîne automatiquement la clôture du niveau logique correspondant.
    - Ex: changement du numéro d'usine

# Introduction (7)

---

- Remarques:
  - Il est possible d'exécuter une procédure avant la génération d'un groupe.
  - Il est possible de ne pas générer (ne pas imprimer) une partie de rapport via SUPPRESS PRINTING
  - Il existe deux modes de rapport:
    - détaillé qui produit l'ensemble des lignes de détail et les groupes de début et fin
    - récapitulatif qui produit les titres et finalisations (le rapport contient principalement des totaux)
  - Une donnée d'un groupe détail peut être mentionnée comme *impression en mode indicatif* : elle n'apparaîtra que lors de la 1ère occurrence

# Introduction (8)

---

## – Déclaration

- A l'instar de la SCREEN SECTION, la déclaration d'un rapport imprimé a lieu également en DATA DIVISION dans la REPORT SECTION.
- Le rapport prend la forme d'une structure COBOL améliorée d'options spécifiques à la génération de rapport.
- Chaque rapport est décrit suivant 2 clauses: CONTROLS qui mentionne l'indicatif de structure et PAGES qui définit le format de la page
- Chaque groupe contient une clause TYPE qui définit le type du groupe. Les procédures à exécuter avant l'édition d'un groupe sont déclarées dans les DECLARATIVES

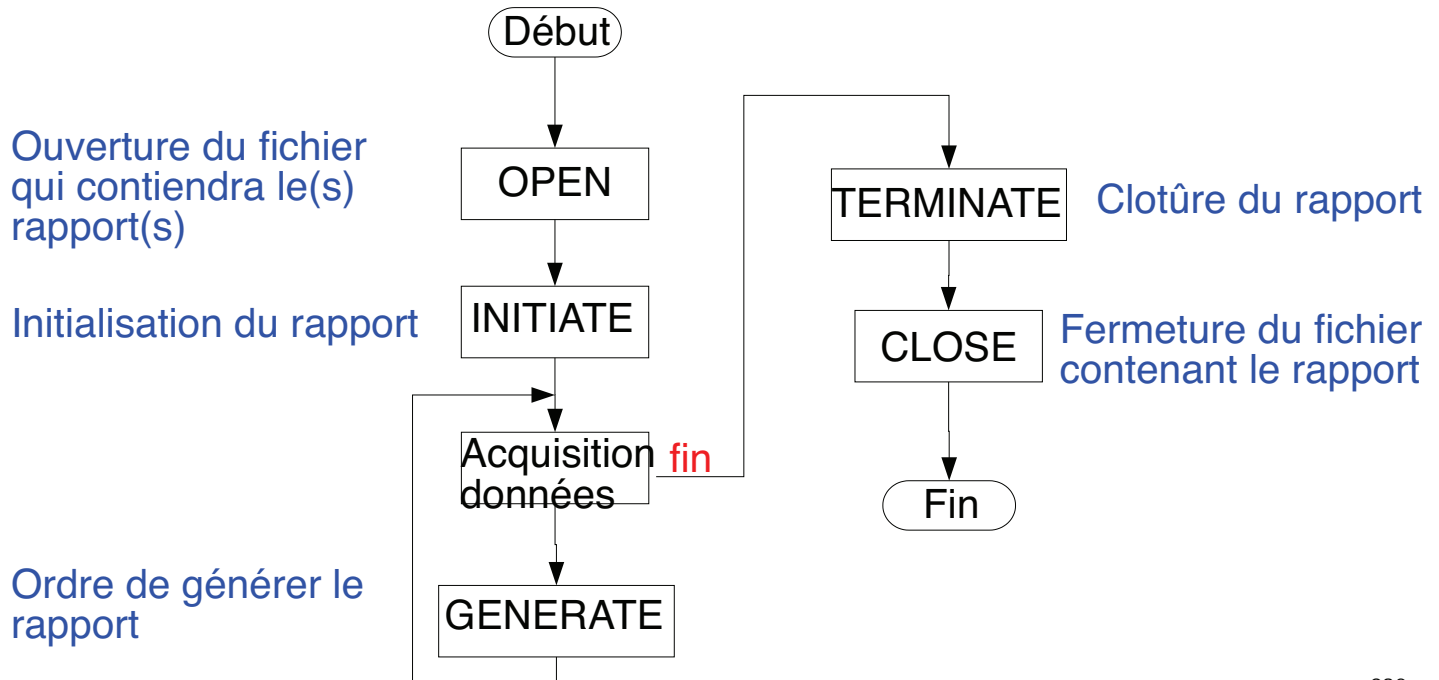
# Introduction (9)

---

- A chaque ligne est mentionnée sa position sur la page via LINE.
- Plusieurs clauses peuvent être associée à chaque donnée imprimée :
  - COLUMN pour mentionner la position
  - VALUE si la donnée est constante et cela permet d'en mentionner la valeur
  - SOURCE si la valeur de la donnée provient d'une variable déclarée autre-part dans le programme.
  - SUM si la valeur de la donnée est une totalisation réalisée par le module COBOL de génération de rapport.

# Introduction (10)

- Un programme d'édition:



COBOL, 2010

236

# ReportWriter (1)

- Détails en FILE SECTION:
  - Un fichier est imprimé est un fichier comme les autres:
    - On trouve en **ENVIRONMENT DIVISION** une clause **SELECT** qui décrit son emplacement sur disque.
    - La rubrique **FD** ne contient aucune description d'enregistrement mais une clause spécifique **REPORT**. Aucune clause SAME AREA autorisée.
    - Le programmeur doit effectuer les opérations **OPEN** / **CLOSE** mais ne doit rédiger aucune instruction **WRITE**.
      - Ecriture réalisée par le ReportWriter.

COBOL, 2010

237

# ReportWriter (2)

---

## - Format:

```
FD nom-fichier
  [GLOBAL] [EXTERNAL]
  [clause RECORD CONTAINS]
  [clause BLOCK CONTAINS]
  [clause CODE-SET]
  clause REPORT.
```

- La clause report est obligatoire:  
REPORT {nom-rapport}
  - Les rapports indiqués doivent faire l'objet d'une rubrique RD en REPORT SECTION.
  - Un rapport ne peut être mentionné qu'une seule fois dans une seule clause REPORT.
- Les clauses sont celles des fichiers habituels en COBOL.

# ReportWriter (3)

---

## • Registres spéciaux

- Le module ReportWriter définit des registres spéciaux qu'il maintient. Le programmeur à toutefois la possibilité de les utiliser dans son programme.

- LINE-COUNTER – format: PIC 9(6). Indique le dernier numéro de la ligne imprimée (dans la page).  
L'instruction INITIATE initialise cette valeur à 0.
  - Une clause SOURCE ou une instruction dans la PROCEDURE DIVISION peut désigner ce registre.
  - Si plusieurs rapports sont définis:  
LINE-COUNTER [ OF nom-rapport ]

# ReportWriter (4)

---

- PAGE-COUNTER – format PIC 9(6). Ce registre donne le numéro de la page en cours d'impression.  
L'instruction INITIATE place la valeur 1.
  - Ce registre peut être désigné dans la PROCEDURE DIVISION ou via une clause SOURCE.
  - Si plusieurs rapports sont définis, il faut indiquer le nom complet comme suit:  
**PAGE-COUNTER [OF nom-rapport ]**
  - La valeur de ce registre **peut** être modifié par le programme en vue de commencer la numérotation à une autre valeur que 1.

# ReportWriter (5)

---

- Détails en DATA DIVISION
  - La REPORT SECTION permet de décrire un rapport imprimé.

```
REPORT SECTION.  
{ RD nom-rapport  
  { 01 declaration-groupe  
    { [declaration-ligne]  
      {declaration-donnée}1-N  
    }0-N  
  }1-N  
}0-N
```



# ReportWriter (6)

---

## – Remarques:

- Un groupe identifie un événement que le ReportWriter doit détecter. A chaque fois que l'événement survient:
  - Le ReportWriter effectue des totalisations automatiques
  - exécute les procédures qui doivent être déclenchées (elles sont définies dans une section DECLARATIVES).
  - effectue l'impression d'une ou plusieurs lignes.
- Un groupe peut représenter:
  - un groupe imprimable qui est composé de plusieurs lignes
  - un groupe non-imprimable qui contient des totalisations non-imprimable
  - un groupe vide qui est présent pour l'exécution d'une procédure

# ReportWriter (7)

---

- Une déclaration de ligne doit être liée à (au moins) une déclaration de donnée imprimable
  - Une totalisation automatique non-imprimable peut être liée à une déclaration de ligne.
- ## – Désignation d'un rapport
- Il est désigné par une clause REPORT de la FILE SECTION. Il peut également être mentionné dans une instruction INITIATE, TERMINATE ou GENERATE (génération récapitulative dans ce cas).
- ## – Désignation d'un groupe
- Dans la section DECLARATIVES, par une instruction USE BEFORE REPORTING

## ReportWriter (8)

---

- pour un groupe détail, via l'instruction GENERATE (PROCEDURE DIVISION) et également dans une clause SUM (UPON ...) en REPORT SECTION.
- Une ligne n'est jamais référencée.
  - Il est donc inutile de lui donner un nom
- Une variable de totalisation (déclarée en REPORT SECTION) peut être référencée par le programme ou via SOURCE / SUM

## ReportWriter (9)

---

- Option rubrique RD

```
RD nom-rapport
  [IS GLOBAL]
  [clause CODE]
  [clause CONTROL]
  [clause PAGE].
```
- Interprétation
  - La clause CODE permet d'introduire un repère spécifique des lignes des différents rapport. Etait utile sur des bandes magnétiques.
  - La clause CONTROL mentionne l'indicatif de structure (décrivant la hiérarchie du rapport).
  - La clause PAGE décrit la mise en page.

# ReportWriter (10)

---

## - Détail de la clause CONTROL

```
CONTROL IS { nom-donnee | FINAL nom-donnee }
```

- Le nom-donnee doit être déclaré en DATA DIVISION (mais **pas** en REPORT SECTION)
- Une donnée ne peut être mentionnée qu'une seule fois.
- On mentionne les indicatifs de structure dans l'ordre hiérarchique décroissant (USINE ATELIER)
  - A chaque élément cité pourra être associé un groupe *control heading* et/ou un groupe *control footing*
- FINAL indique que le rapport complet est également un groupe
  - Il pourra également être associé à un groupe *control heading* et/ou un groupe *control footing*.

# ReportWriter (11)

---

- Exemple: soit la description d'un fichier

```
...  
02 INS.  
  03 Province PIC 9.  
  03 Arrondissement PIC 9.  
  03 Commune PIC 999.
```

*Chaque enregistrement décrit une commune de Belgique. On désire éditer ces enregistrements ainsi que des totaux récapitulatifs par commune, par arrondissement administratif, par province et pour le pays. Une commune est identifiée par son numéro INS pacc (p = province, a=arrondissement dans la province, ccc=commune dans l'arrondissement).*

*Pour ce faire, on programmera la clause suivante:*

```
CONTROL FINAL Province Arrondissement Commune.
```

*Ou encore:*

```
CONTROL FINAL Province Arrondissement INS.
```

# ReportWriter (12)

## - Détail de la clause PAGE

- Elle est responsable de la mise en page. Elle doit être présente si:

- Le rapport fait plus d'une page
- Un groupe *page heading* et/ou *page footing* doit être défini
- L'espacement entre les lignes n'est pas relatif.

```
PAGE LIMIT entier-1 LINE
[HEADING entier-2]
[FIRST DETAIL entier-3]
[LAST DETAIL entier-4]
[FOOTING entier-5]
```

- Il faut que:

- $1 \leq \text{entier-2} \leq \text{entier-3} \leq \text{entier-4} \leq \text{entier-5} \leq \text{entier-1}$
- Si le placement est complètement absolu, seul LIMIT est requis

248

COBOL, 2010

# ReportWriter (13)

- Explications:

- entier-1 définit la hauteur de la page en nombre de lignes
- Les zones dans la pages peuvent être occupée par les groupes comme suit:

		Si page de garde séparée				
	1	—				
HEADING	HEADING	RH PH		RH RF		Début de la page
	—	RH PH		RH RF		
	—	RH PH		RH RF		
1 <sup>st</sup> DETAIL	1 <sup>st</sup> DETAIL	CH DE	CF	RH RF	Corps de la page	
	—	CH DE	CF	RH RF		
	—	CH DE	CF	RH RF		
	—	CH DE	CF	RH RF		
LAST DETAIL	LAST DETAIL	CH DE	CF	RH RF		
	—		CF	RH RF		
FOOTING	FOOTING		CF	RH RF		
	—		CF	RH RF		
	—		CF	RH RF		
LIMIT	LIMIT		PF RF	RH RF	Fin de la page	
	—		PF RF	RH RF		
	—		PF RF	RH RF		

249

COBOL, 2010

# ReportWriter (14)

---

- Règles à respecter (calculs en nombre de lignes):
  - $PH + RH \leq \text{FIRST-DETAIL} - \text{HEADING}$ 
    - RH uniquement si pas de page de garde séparée
  - $PF + RF \leq \text{PAGE-LIMIT} - \text{FOOTING}$ 
    - RF uniquement si pas de page de garde séparée
  - Tous les groupes  $CF \leq \text{FOOTING} - \text{LAST-DETAIL}$ 
    - Tous ces groupes doivent se trouver sur une même page
  - $DE + CH \leq \text{LAST-DETAIL} - (\text{FIRST-DETAIL} - 1)$ 
    - On prend en compte le plus long détail possible

# ReportWriter (15)

---

## - Description d'un groupe

```
01 nom-groupe  
  clause TYPE  
  [clause NEXT-GROUP].
```

- Règle:
  - NEXT-GROUP ne peut être utilisé que pour des données imprimable
- Interprétation:
  - La clause TYPE est requise et elle indique l'événement qui produira une nouvelle occurrence du groupe
  - La clause NEXT-GROUP indique que des lignes vides suivent le groupe défini.

# ReportWriter (16)

---

- La clause TYPE

```
TYPE IS ( RH | PH |  
         CH (nom-donnee | FINAL) |  
         DE |  
         CF (nom-donnee | FINAL) |  
         PF | RF )
```

- Interprétation

- Cette clause obligatoire détermine le type du groupe (et donc l'événement associé)
- Si les données sont imprimables, elle détermine également l'emplacement.

- Règles

- FINAL ou nom-donnee désigne, dans CH/CF, le bloc logique auquel se rapporte le groupe
- une option CH/CF/DE est obligatoire.

# ReportWriter (17)

---

- Si un rapport détaillé doit être produit, au moins un groupe DE doit être défini.
- Pour chaque élément indiqué en CONTROL, un groupe CF et CH peut être produit
- Pour la mise en page, un groupe RH, RF, PH, PF peut être défini
- L'ordre de présentation doit être respecté:

```
  RH  
  PH  
  CH FINAL  
  CH indicatif-majeur  
  CH indicatif-mineur  
  DE  
  CF indicatif-mineur  
  CF indicatif-majeur  
  CF FINAL  
  PF  
  RF
```

# ReportWriter (18)

---

- La clause NEXT-GROUP

```
NEXT-GROUP IS (NEXT-PAGE | entier-1 |  
              PLUS entier-2)
```

- Interprétation:

- Permet d'indiquer, pour un groupe imprimable, le nombre de ligne vide après la dernière ligne.
  - L'option NEXT-PAGE ajoute suffisamment de ligne pour passer à la page suivante
  - L'option entier-1 ajoute des lignes jusqu'à arriver à la ligne entier-1.
  - PLUS entier-2 ajoute entier-2 lignes à la fin du groupe imprimable.
- Si cette option est spécifiée dans un groupe CF, seul une rupture sur l'événement associé produira les lignes vides demandées.

# ReportWriter (19)

---

- Description d'une ligne [clause LINE NUMBER]

```
numéro-niveau [nom-donnee]  
              clause LINE NUMBER.
```

- Règles

- Ne peut se trouver que dans un groupe imprimable (chaque ligne doit contenir au moins une donnée imprimable).
- La rubrique doit être liée à une déclaration de groupe (on utilisera le numéro 02).
- Les lignes doivent être ordonnées suivant l'ordre d'apparition (haut->bas).
- Si le groupe ne contient qu'une seule ligne, la clause LINE NUMBER peut être spécifiée au niveau 01
- Si la ligne ne contient qu'une seule donnée, les clauses décrivant cette donnée peuvent être reportées à la description de la ligne.

# ReportWriter (20)

---

- Format:

```
LINE NUMBER is (entier-1 [ON NEXT PAGE] |  
                PLUS entier-2)
```

- Règles:

- l'option de positionnement absolu (*entier-1 [ON NEXT PAGE]*) n'est permise que si une clause PAGE est définie pour le rapport
- La séquence des clauses LINE NUMBER **pour un groupe** doit être conforme au schéma
  - 0-1 LINE NUMBER entier-1 NEXT PAGE
  - 0-m LINE NUMBER entier-1
  - 0-n LINE NUMBER PLUS entier-2
- Il faut veiller à ne pas dépasser la zone définie car les données d'un groupe doivent se trouver sur une même page.

# ReportWriter (21)

---

- Placement absolu [*entier-1 [ON NEXT PAGE]* )

- La ligne est imprimée à la ligne entier-1
- Si l'option NEXT PAGE est précisée, les données sont placées sur une nouvelle page
- Si la ligne mentionnée est antérieure à la ligne courante du rapport, une nouvelle page est générée.

- Placement relatif PLUS entier-2.

- Première ligne d'un groupe ou bloc logique
  - Le groupe (CH, DE ou CF) entier est placé suivant la contrainte mentionnée.
  - Tout le groupe est placé sur la page en cours ou, à défaut, sur une nouvelle page
  - La première ligne du premier groupe dans une page est placée à la 1ère position libre de la zone (FIRST-DETAIL).



# ReportWriter (22)

---

- Règles générales
  - La ligne est placée à la position LC (Ligne-courante) + entier-2
  - La valeur de LC est adaptée si la zone autorisée pour le groupe est < à la valeur de LC. Dans ce cas LC est fixé à la zone autorisée - 1.
  - entier-2 > 0
- Changement de page
  - Lors d'un changement de page, les groupes PF et PH sont produit (s'ils sont présent).

# ReportWriter (23)

---

## - Donnée imprimable

- Format

```
niveau [nom-donnee]
  clause PICTURE
    [clause JUSTIFIED RIGHT]
    [clause SIGN SEPARATE]
    [clause BLANK WHEN ZERO]
    [clause USAGE IS DISPLAY]
  clause COLUMN NUMBER
  clause (VALUE | SOURCE | SUM)
  [clause GROUP INDICATE].
```

- Remarques:

- Pour les données imprimable, la clause PICTURE est obligatoire; son format peut être édité.
- La rubrique doit être liée à une déclaration de ligne (on utilisera le niveau 03)

# ReportWriter (24)

---

- Si la ligne ne contient qu'une seule donnée, les clauses décrivant cette donnée peuvent être reportées à la description de la ligne à moins que celle-ci définisse un groupe.
- Signification
  - L'interprétation des clauses contenues dans la clause PICTURE est identique à celle vue précédemment
    - Il faut savoir qu'il faut mentionner l'option SEPERATE CHARACTER dans le SIGN.
- Clause COLUMN NUMBER
  - Format:  
COLUMN NUMBER IS entier

# ReportWriter (25)

---

- Mentionne La position (à partir de 1) où la donnée décrite sera produite.
- Les positions doivent être croissante (de gauche à droite) suivant l'ordre d'apparition des informations.
  - Aucun chevauchement n'est permis.
- Les positions non occupées sont produites comme des caractères blanc.
- La clause VALUE
  - Format  
VALUE constante
  - La valeur de la donnée est mentionnée dans la clause. Cela permet notamment de produire les titres d'un rapport.

# ReportWriter (26)

---

- La clause SOURCE

- Format:

- SOURCE (identificateur | PAGE-COUNTER |  
LINE-COUNTER)

- Interprétation:

- Indique la source pour la zone de donnée décrite (zone où la valeur sera lue).

- Remarques:

- L'identificateur peut être une totalisation automatique (voir clause SUM) ou une autre donnée décrite hors de la REPORT SECTION.
    - Les valeurs PAGE-COUNTER ou LINE-COUNTER sont qualifiées par le rapport en cours et désignent respectivement les compteurs de pages ou de lignes.

# ReportWriter (27)

---

- La clause SUM

- Format:

- ```
{ SUM {identificateur}
  [UPON {groupe-detail} ] } Liste
  [ RESET (nom-donnee | FINAL ) ] }
```

Clause

- Signification:

- La **clause** SUM peut comprendre plusieurs **listes** SUM. Une clause définit une totalisation.
    - Chaque donnée citée dans la clause sera totalisée dans la variable décrite.

- Règle:

- **Une clause SUM ne peut apparaître que dans un CF.**

# ReportWriter (28)

---

– Totalisations:

- Une **liste** SUM peut comporter une donnée décrite hors de la REPORT SECTION ou une totalisation.
- L'option UPON ne peut être mentionnée que si la liste ne contient aucune totalisation.

- Le sous-total

```
SUM {identificateur}
  [UPON {groupe-detail}]
```

- La donnée est déclarée hors de la REPORT SECTION.
- Chaque groupe-detail mentionne un groupe DETAIL du rapport
- Sauf indiqué via l'option UPON, l'accumulation s'effectue pour chaque groupe détail du rapport.

# ReportWriter (29)

---

- Pour les éditions détaillées, chaque fois que le groupe détail mentionné (ou tous s'il n'est pas mentionné) est produit, les identificateurs mentionnés dans la liste SUM sont accumulés dans la donnée de totalisation.
- Pour les éditions récapitulatives, la totalisation s'opère comme si le rapport comportait un groupe détail cité dans l'option UPON.
- Le total croisé
  - On appelle **total croisé** tout identificateur cité par une clause SUM qui désigne une totalisation déclarée à l'intérieur du groupe en cours de description
  - Le système COBOL va effectuer les calculs dans l'ordre des totalisations définies.

# ReportWriter (30)

- Exemples:

```
RD Facture (...) CONTROL No-Facture.
01 En-Tete TYPE DETAIL NEXT GROUP IS 13.
  02 LINE 4 ON NEXT PAGE.
  ... (date, nom et adresse du client, ...)

01 Ligne TYPE DETAIL.
  02 LINE PLUS 1.
  ... (produit, quantité, prix, taxe, ...)

01 Finale TYPE CONTROL FOOTING No-Facture.
  02 LINE 44.
    03 Total-HT COLUMN 42 PIC Z(6) SUM Prix UPON Ligne.
    03 Total-Taxe COLUMN 45 PIC Z(5) SUM Taxe UPON Ligne.
  02 LINE 48.
    03 A-Payer COLUMN 55 PIC Z(6) SUM Prix Taxe UPON Ligne.
```

## Exemple sous-total 1

# ReportWriter (31)

- Exemples (suite) :

```
RD Synthese (...) CONTROL FINAL Usine Atelier.
01 Liste-releve TYPE DETAIL.
...

01 Recap-Atelier TYPE CONTROL FOOTING Atelier.
  02 LINE PLUS 2.
    03 (...) SUM Quantite-Fabriquee. —> UPON Liste-releve

01 Recap-Usine TYPE CONTROL FOOTING Usine.
  02 LINE PLUS 2.
    03 (...) SUM Quantite-Fabriquee. —> UPON Liste-releve

01 Recap-General TYPE CONTROL FOOTING FINAL.
  02 LINE PLUS 4.
    03 (...) SUM Quantite-Fabriquee. —> UPON Liste-releve
```

## Exemple sous-total 2

# ReportWriter (32)

- Exemples (suite):

```
RD Facture (...) CONTROL No-Facture.
01 En-Tete TYPE DETAIL NEXT GROUP IS 13.
  02 LINE 4 ON NEXT PAGE.
  ... (date, nom et adresse du client, ...)

01 Ligne TYPE DETAIL.
  02 LINE PLUS 1.
  ... (produit, quantité, prix, taxe, ...)

01 Finale TYPE CONTROL FOOTING No-Facture.
  02 LINE 44.
    03 Total-HT COLUMN 42 PIC Z(6) SUM Prix UPON Ligne.
    03 Total-Taxe COLUMN 45 PIC Z(5) SUM Taxe UPON Ligne.
  02 LINE 48.
    03 A-Payer COLUMN 55 PIC Z(6) SUM Total-HT Total-Taxe.
```

## Exemple total croisé

# ReportWriter (33)

- Le total récapitulatif
  - On appelle **total récapitulatif** tout identificateur cité par une clause SUM qui désigne une totalisation déclarée dans un autre groupe CONTROL FOOTING déjà défini dans le rapport en cours de description, groupe de niveau lié dans la hiérarchie logique du rapport.
- Remise à zéro:
  - Une donnée de totalisation est remise à 0 automatiquement lorsque le groupe CONTROL FOOTING auquel elle se rapporte est imprimé.
  - Dans certains cas (totaux intermédiaires), il est préférable de différer cette réinitialisation (à la rupture de niveau supérieur).

# ReportWriter (34)

---

- Exemples (fin):

Copyright (C) André CLARINVAL/PUN 1991

```
RD Synthese (...) CONTROL FINAL Usine Atelier.  
01 Liste-releve TYPE DETAIL.  
...  
  
01 Recap-Atelier TYPE CONTROL FOOTING Atelier.  
02 LINE PLUS 2.  
03 Total (...) SUM Quantite-Fabriquee.  
  
01 Recap-Usine TYPE CONTROL FOOTING Usine.  
02 LINE PLUS 2.  
03 Total (...) SUM Total OF Recap-Atelier  
  
01 Recap-General TYPE CONTROL FOOTING FINAL.  
02 LINE PLUS 4.  
03 Total (...) SUM Total OF Recap-Usine.
```

Exemple total récapitulatif

# ReportWriter (35)

---

- Remise à zéro (suite):  
RESET ON (nom-donnee | FINAL)
  - Le nom-donnee est un élément de la liste CONTROL de la rubrique RD
  - La mention de FINAL ou nom-donnée désigne l'élément auquel il n'est pas nécessaire d'associer un groupe CONTROL FOOTING
- Valeur initiale d'une donnée de totalisation
  - Il est possible de modifier la valeur initiale en utilisant une clause USE BEFORE REPORTING associée au groupe CONTROL HEADING de niveau correspondant.

# ReportWriter (36)

---

- La clause GROUP INDICATE
  - Format:  
GROUP INDICATE
  - Signification:
    - Ne peut se trouver que pour une donnée imprimable dans un groupe DETAIL.
    - Elle indique que la donnée ne sera imprimée que
      - lors de la production du 1er groupe lors d'un changement de page
      - lors de la production du 1er groupe lors d'une rupture sur un indicatif.
    - Dans les autres cas, la donnée n'est pas imprimée est est remplacée par des blancs.

# ReportWriter (37)

---

- Total automatique non-imprimable
  - Format:  
niveau nom-donnee  
clause PICTURE  
clause SUM.
  - Signification:
    - Cette rubrique définit une donnée de totalisation non-imprimable qui pourra être référencée par le nom.
      - Dans la REPORT SECTION: SOURCE, SUM
      - Dans la PROCEDURE DIVISION
    - La syntaxe de la clause SUM est inchangée.
    - Le format de la donnée doit être non-édité
    - Cette rubrique doit se trouver dans un groupe CONTROL FOOTING.



# ReportWriter (38)

---

- Exemple: *Calcul d'une moyenne: moyenne journalière des ventes pour chaque département d'un grand magasin, sur base de relevés quotidiens.*

**WORKING-STORAGE SECTION.**

77 Moyenne PIC 9(6).

77 Un PIC 9 VALUE 1.

**REPORT SECTION.**

RD Rapport (...) CONTROL Departement.

...

01 Fin-Departement TYPE CONTROL FOOTING Departement.

02 Total PIC 9(7) SUM Montant-Releve.

02 Nombre-Jours PIC 99 SUM Un.

02 LINE PLUS 2.

...

03 COLUMN 45 PIC ZZZ,ZZZ SOURCE Moyenne.

**PROCEDURE DIVISION.**

DECLARATIVES.

Calcul-Moyenne SECTION.

USE BEFORE REPORTING Fin-Departement.

Calculer.

COMPUTE Moyenne ROUNDED = Total / Nombre-Jours.

END DECLARATIVES

274

# ReportWriter (39)

---

- Détails en PROCEDURE DIVISION

- DECLARATIVES

- La section DECLARATIVES est utilisées pour associer une procédure avant la production d'un groupe (après les totalisations et avant l'impression).

- Format:

nom-section SECTION.

USE BEFORE REPORTING nom-groupe.

- Remarques:

- Un groupe donné ne peut faire l'objet que d'une seule clause USE.
- Il est recommandé de ne pas modifier la valeur de l'indicatif de contrôle du rapport.

# ReportWriter (40)

---

## – SUPPRESS PRINTING

- Format:

SUPPRESS PRINTING

- Signification:

- La clause USE peut comporter une instruction SUPPRESS PRINTING qui a pour conséquence **de ne pas** produire cette occurrence du groupe.

- Remarque:

- Cette instruction ne peut se trouver qu'à l'intérieur d'une section USE BEFORE REPORTING.

# ReportWriter (41)

---

## – Initialisation du rapport: INITIATE

- Format

INITIATE {nom-rapport}

- Signification

- Cette instruction s'occupe des initialisations pour les rapports mentionnés.
  - Mise à zéro des données de totalisation et du registre LINE-COUNTER
  - Mise à un du registre PAGE-COUNTER
- Les fichiers utilisés pour l'impression du rapport doivent être ouverts.

- Remarques:

- Les nom-rapport doivent être distincts. Un rapport peut être re-initialisé s'il a été finalisé (par une instruction TERMINATE).

# ReportWriter (42)

## - Production du rapport: GENERATE

- Format:

GENERATE (groupe-detail | nom-rapport)

- Signification:

- L'impression d'un rapport **détaillé** (en mentionnant un groupe-detail) produit toutes les opérations d'édition (y compris l'ensemble du détail) pour le groupe déterminé.
- L'impression d'un rapport **récapitulatif** (en mentionnant un nom-rapport) produit toutes les opérations d'édition (sauf l'impression des lignes de détail) du rapport.
  - Elle ne peut porter que sur un rapport décrit avec la clause CONTROL et incluant 0 ou 1 groupe détail.
  - Pour un même rapport, il est possible de réaliser une impression détaillée ou récapitulative.

# ReportWriter (43)

## - L'instruction GENERATE:

- Analyse les ruptures et détermine quels groupes logiques (CONTROL FOOTING/HEADING) doivent être produits.
  - A chaque changement d'un indicatif de contrôle, une rupture survient

| 1er GENERATE |    | RH | PH | CH | DE | CF | PF | RF | ← Production du groupe                                   |
|--------------|----|----|----|----|----|----|----|----|----------------------------------------------------------|
| Ordre        |    |    |    |    | xx |    | xx |    | <b>TOTALISATIONS</b>                                     |
|              |    |    |    |    |    |    | xx |    | - sous-totaux                                            |
|              |    |    |    |    |    |    | xx |    | - totaux croisés                                         |
|              | xx | xx | xx | xx | xx | xx | xx | xx | - totaux récupératifs                                    |
|              |    |    |    |    |    |    |    |    | <b>EXECUTION DE LA SECTION USE</b>                       |
|              |    |    |    |    |    |    |    |    | <b>IMPRESSION</b>                                        |
|              |    |    |    | xx | xx | xx |    |    | - Choix et impression des groupes physique (RH PH PF RF) |
| ▼            | xx | xx | xx | xx | xx | xx | xx | xx | - impression des lignes                                  |
|              |    |    |    |    | xx |    |    |    | <b>REMISE A ZERO DES TOTALISATIONS</b>                   |

# ReportWriter (44)

---

## - Clôture du rapport : TERMINATE

- Format:

```
TERMINATE {nom-rapport }
```

- Signification:

- Cette instruction finalise les rapports mentionnés en produisant les groupes CF, PF et RF (s'ils sont définis).
- Le fichier associé à l'impression du rapport doit être ouvert au moment du TERMINATE.

- Remarque:

- Tous les noms de rapports doivent être distincts.