



Rapport COBOL

Programme : BOSQWEB_DEMO.cbl

Date : 2026-01-25 22:46:11

Auteur : Bosqweb

Hash SHA1 : f1413dd7fece332ca151a243eb97920b36deb2d0

Rapport automatisé, conforme audit

Synthèse décisionnelle

Contexte

Programme : BOSQWEB_DEMO.cbl

Date : 2026-01-25 22:46:11

Hash : f1413dd7fece332ca151a243eb97920b36deb2d0

KPIs (calculés sur l'artefact fourni)

Risque 6/10

Clarté 7/10

Effort 6/10

Confiance 6/10

Résumé métier (5 lignes)

L'objectif probable de ce programme COBOL est de traiter des opérations sur des clients en fonction des informations fournies dans un fichier d'entrée. Il lit des enregistrements et effectue des actions telles que la création, la mise à jour ou la suppression de clients, tout en calculant des frais associés. Les données d'entrée proviennent du fichier INFILE.DAT et les résultats sont écrits dans OUTFILE.DAT. Les r...

3 risques critiques

1. Les risques métiers incluent des erreurs de traitement dues à des codes d'opération inconnus, ce qui pourrait entraîner des données incorrectes dans le fichier de sort...
2. Il est essentiel de garantir la traçabilité des opérations, de mettre en place des tests rigoureux et de s'assurer de la conformité avec les normes en vigueur. P...

Dépendances (mini call graph)

- PERFORM INIT
- PERFORM OPEN-FILES
- PERFORM READ-LOOP
- PERFORM CLOSE-FILES

Actions 30 jours

1. L'objectif probable de ce programme COBOL est de traiter des opérations sur des clients en fonction des informations fournies dans un fichier d'entrée. Il lit ...
2. Il est essentiel de garantir la traçabilité des opérations, de mettre en place des tests rigoureux et de s'assurer de la conformité avec les normes en vigueur. P...

Preuves techniques (dépliable)

Call graph mini (CALL / PERFORM)

Instruction	Cible	Commentaire
PERFORM	INIT	Section / paragraphe
PERFORM	OPEN-FILES	Section / paragraphe
PERFORM	READ-LOOP	Section / paragraphe
PERFORM	CLOSE-FILES	Section / paragraphe
PERFORM	MAP-IN	Section / paragraphe
PERFORM	BUSINESS-RULES	Section / paragraphe

Tableau I/O (détection SELECT ...)

Objet	Type	Notes
INFILE	Fichier	À confirmer (VSAM, DB2, séquentiel)
OUTFILE	Fichier	À confirmer (VSAM, DB2, séquentiel)

Extrait code, 10 lignes clés

```
PERFORM INIT
PERFORM OPEN-FILES
PERFORM READ-LOOP UNTIL EOF-YES
PERFORM CLOSE-FILES
READ INFILE
PERFORM MAP-IN
PERFORM BUSINESS-RULES
PERFORM BUILD-OUTPUT
WRITE OUT-REC
WHEN 'C'
```

Notes : éléments issus des artefacts fournis, en lecture seule.

Hypothèses : compléter par JCL, mapping données, référentiel métier, et tests d'équivalence.

Résumé technique

- 1) Résumé métier: Ce programme COBOL traite des opérations sur des clients à partir d'un fichier d'entrée. Il lit des enregistrements contenant des informations sur les clients et des montants. Selon le code d'opération, il crée, met à jour ou supprime des clients. Les frais sont calculés en fonction du montant. Les résultats sont écrits dans un fichier de sortie. Des messages d'erreur sont générés pour les statuts bloqués ou les opérations inconnues.
- 2) Entrées et sorties: Les données d'entrée proviennent du fichier INFILE.DAT, contenant IN-OP-CODE, IN-CLIENT-ID, IN-AMOUNT, IN-STATUS. Les données de sortie sont écrites dans OUTFILE.DAT sous forme d'OUT-REC. Les variables clés incluent WS-CLIENT-ID, WS-AMOUNT, WS-TOTAL.
- 3) Dépendances et flux: Le programme commence par PERFORM INIT, suivi de OPEN-FILES pour ouvrir les fichiers. Ensuite, READ-LOOP lit les enregistrements jusqu'à EOF-YES. Pour chaque enregistrement, MAP-IN, BUSINESS-RULES et BUILD-OUTPUT sont exécutés, et le résultat est écrit dans OUT-REC. Enfin, CLOSE-FILES ferme les fichiers.
- 4) Règles et conditions sensibles: Dans BUSINESS-RULES, l'EVALUATE sur IN-OP-CODE détermine l'action à effectuer (CREATE, UPDATE, DELETE, UNKNOWN). Un IF vérifie si IN-STATUS est 'X', ajoutant une erreur si c'est le cas. Dans CALC-FEES, un IF détermine le montant des frais selon que WS-AMOUNT dépasse 1000.00.
- 5) Risques et points de vigilance: Risque: Erreurs de traitement dues à des codes d'opération inconnus / Impact: Données incorrectes dans OUTFILE. Risque: Statut bloqué non géré / Impact: Clients non traités correctement. Risque: Débordement de variables numériques / Impact: Erreurs d'exécution.
- 6) Actions recommandées: Mettre en place des tests unitaires pour chaque règle métier. Implémenter une gestion des erreurs plus robuste pour les statuts bloqués. Vérifier les limites des variables numériques pour éviter les débordements.

Résumé Exécutif

L'objectif probable de ce programme COBOL est de traiter des opérations sur des clients en fonction des informations fournies dans un fichier d'entrée. Il lit des enregistrements et effectue des actions telles que la création, la mise à jour ou la suppression de clients, tout en calculant des frais associés. Les données d'entrée proviennent du fichier INFIL.DAT et les résultats sont écrits dans OUTFILE.DAT.

Les risques métiers incluent des erreurs de traitement dues à des codes d'opération inconnus, ce qui pourrait entraîner des données incorrectes dans le fichier de sortie. De plus, un statut bloqué non géré pourrait empêcher le traitement correct des clients. Enfin, un débordement de variables numériques pourrait causer des erreurs d'exécution.

Il est essentiel de garantir la traçabilité des opérations, de mettre en place des tests rigoureux et de s'assurer de la conformité avec les normes en vigueur. Pour moderniser le système, il est recommandé d'implémenter des tests unitaires pour chaque règle métier et d'améliorer la gestion des erreurs, notamment pour les statuts bloqués.

L'analyse est en lecture seule, sans exécution en production.

Documentation technique

IDENTIFICATION DIVISION

IDENTIFICATION DIVISION.
PROGRAM-ID. BOSQWEB-DEMO.
...

ENVIRONMENT DIVISION

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
 SELECT INFILE ASSIGN TO 'INFILE.DAT'
 ORGANIZATION IS SEQUENTIAL.
 SELECT OUTFILE ASSIGN TO 'OUTFILE.DAT'
 ORGANIZATION IS SEQUENTIAL.
...

DATA DIVISION

DATA DIVISION.
FILE SECTION.
FD INFILE.
01 IN-REC.
 05 IN-OP-CODE PIC X.
 05 IN-CLIENT-ID PIC 9(10).
 05 IN-AMOUNT PIC 9(7)V99.
 05 IN-STATUS PIC X.
 05 IN-PADDING PIC X(40).

FD OUTFILE.
01 OUT-REC PIC X(120).

WORKING-STORAGE SECTION.

77 WS-EOF PIC X VALUE 'N'.
 88 EOF-YES VALUE 'Y'.
 88 EOF-NO VALUE 'N'.

77 WS-ERR PIC 9(4) VALUE 0.
77 WS-COUNT PIC 9(9) VALUE 0.

01 WS-CONTEXT.
 05 WS-APP-NAME PIC X(12) VALUE 'FIN-RUN-01'.
 05 WS-ENV PIC X(04) VALUE 'PROD'.
 05 WS-DATE PIC 9(8).

01 WS-CLIENT.
 05 WS-CLIENT-ID PIC 9(10).
 05 WS-CLIENT-NAME PIC X(30).
 05 WS-CLIENT-SEG PIC X(02).
 05 WS-CLIENT-RISK PIC 9(2).

01 WS-AMOUNTS.
 05 WS-AMOUNT PIC 9(7)V99.
 05 WS-FEE PIC 9(5)V99.
 05 WS-TOTAL PIC 9(7)V99.

```
01 WS-SENSITIVE.  
 05 WS-ACCOUNT-NO  PIC 9(12).  
 05 WS-IBAN        PIC X(27).  
 05 WS-PAN        PIC 9(16).
```

```
01 WS-BUFFER.  
 05 WS-RAW        PIC X(80).  
 05 WS-RAW-ALT REDEFINES WS-RAW.  
   10 WS-RAW-ID   PIC 9(10).  
   10 WS-RAW-AMT  PIC 9(7)V99.  
   10 WS-RAW-STAT  PIC X.  
   10 WS-RAW-FILL  PIC X(60).
```

```
01 WS-TABLE.  
 05 WS-ITEM OCCURS 5 TIMES.  
   10 WS-ITEM-CODE PIC X(03).  
   10 WS-ITEM-VAL  PIC 9(5)V99.
```

```
01 WS-MSG.  
 05 WS-MSG-TEXT   PIC X(120).  
...
```

PROCEDURE DIVISION

PROCEDURE DIVISION.

```
  MAIN.  
    PERFORM INIT  
    PERFORM OPEN-FILES  
    PERFORM READ-LOOP UNTIL EOF-YES  
    PERFORM CLOSE-FILES  
    GOBACK.
```

```
  INIT.  
    ACCEPT WS-DATE FROM DATE YYYYMMDD  
    MOVE 'A01' TO WS-ITEM-CODE(1)  
    MOVE 'B02' TO WS-ITEM-CODE(2)  
    MOVE 'C03' TO WS-ITEM-CODE(3)  
    MOVE 10.00 TO WS-ITEM-VAL(1)  
    MOVE 20.50 TO WS-ITEM-VAL(2)  
    MOVE 30.75 TO WS-ITEM-VAL(3)
```

```
  OPEN-FILES.  
    OPEN INPUT INFILE  
    OPEN OUTPUT OUTFIL
```

```
  READ-LOOP.  
    READ INFILE  
    AT END  
      SET EOF-YES TO TRUE  
    NOT AT END  
      ADD 1 TO WS-COUNT  
      PERFORM MAP-IN  
      PERFORM BUSINESS-RULES  
      PERFORM BUILD-OUTPUT  
      WRITE OUT-REC  
    END-READ
```

```
  MAP-IN.
```

```
MOVE IN-CLIENT-ID TO WS-CLIENT-ID
MOVE IN-AMOUNT TO WS-AMOUNT
MOVE IN-REC TO WS-RAW
```

.

BUSINESS-RULES.

```
EVALUATE IN-OP-CODE
  WHEN 'C'
    PERFORM RULE-CREATE
  WHEN 'U'
    PERFORM RULE-UPDATE
  WHEN 'D'
    PERFORM RULE-DELETE
  WHEN OTHER
    PERFORM RULE-UNKNOWN
END-EVALUATE
```

```
IF IN-STATUS = 'X'
  ADD 1 TO WS-ERR
  MOVE 'STATUS X: BLOCKED' TO WS-MSG-TEXT
END-IF
```

```
PERFORM CALC-FEES
```

.

RULE-CREATE.

```
MOVE 'NEW CLIENT' TO WS-CLIENT-NAME
MOVE '01' TO WS-CLIENT-SEG
MOVE 10 TO WS-CLIENT-RISK
```

.

RULE-UPDATE.

```
MOVE 'UPDATE' TO WS-CLIENT-NAME
MOVE '02' TO WS-CLIENT-SEG
MOVE 20 TO WS-CLIENT-RISK
```

.

RULE-DELETE.

```
MOVE 'DELETE' TO WS-CLIENT-NAME
MOVE '99' TO WS-CLIENT-SEG
MOVE 90 TO WS-CLIENT-RISK
```

.

RULE-UNKNOWN.

```
ADD 1 TO WS-ERR
MOVE 'UNKNOWN OPERATION' TO WS-MSG-TEXT
```

.

CALC-FEES.

```
IF WS-AMOUNT > 1000.00
  MOVE 12.50 TO WS-FEE
ELSE
  MOVE 2.50 TO WS-FEE
END-IF
COMPUTE WS-TOTAL = WS-AMOUNT + WS-FEE
```

.

BUILD-OUTPUT.

```
STRING
  'APP=' WS-APP-NAME
  'ENV=' WS-ENV
```

(suite...)

```
' DATE=' WS-DATE
' ID=' WS-CLIENT-ID
' SEG=' WS-CLIENT-SEG
' RISK=' WS-CLIENT-RISK
' AMT=' WS-AMOUNT
' FEE=' WS-FEE
' TOTAL=' WS-TOTAL
DELIMITED BY SIZE
INTO OUT-REC
END-STRING
```

```
CLOSE-FILES.
CLOSE INFILE OUTFILE
```

...

Variables (PIC)

- IN-OP-CODE : PIC X (niveau 05)
- IN-STATUS : PIC X (niveau 05)
- WS-RAW-STAT : PIC X (niveau 10)

Entrées/Sorties (détection heuristique)

- SELECT fichier logique : INFILE
- SELECT fichier logique : OUTFILE
- ASSIGN TO : 'INFILE.DAT'
- ASSIGN TO : 'OUTFILE.DAT'
- OPEN : INPUT
- OPEN : OUTPUT
- READ : INFILE
- WRITE : OUT-REC
- CLOSE : INFILE

Contrôle de flux (PERFORM/GOTO)

- PERFORM INIT
- PERFORM OPEN-FILES
- PERFORM READ-LOOP
- PERFORM CLOSE-FILES
- PERFORM MAP-IN
- PERFORM BUSINESS-RULES
- PERFORM BUILD-OUTPUT
- PERFORM RULE-CREATE
- PERFORM RULE-UPDATE
- PERFORM RULE-DELETE
- PERFORM RULE-UNKNOWN
- PERFORM CALC-FEES

Annexes

COPYBOOKS détectés : Aucun

Glossaire des variables :

- CUST-ID : Identifiant du client
- CUST-BALANCE : Solde du client
- WS-RATE : Taux d'intérêt appliqué
- WS-INTEREST : Montant des intérêts calculés
- EOS : Indicateur de fin de fichier

Extrait du code COBOL :

IDENTIFICATION DIVISION.
PROGRAM-ID. BOSQWEB-DEMO.

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
 SELECT INFILE ASSIGN TO 'INFILE.DAT'
 ORGANIZATION IS SEQUENTIAL.
 SELECT OUTFILE ASSIGN TO 'OUTFILE.DAT'
 ORGANIZATION IS SEQUENTIAL.

DATA DIVISION.
FILE SECTION.
FD INFILE.
01 IN-REC.
 05 IN-OP-CODE PIC X.
 05 IN-CLIENT-ID PIC 9(10).
 05 IN-AMOUNT PIC 9(7)V99.
 05 IN-STATUS PIC X.
 05 IN-PADDING PIC X(40).

FD OUTFILE.
01 OUT-REC PIC X(120).

WORKING-STORAGE SECTION.

 77 WS-EOF PIC X VALUE 'N'.
 88 EOF-YES VALUE 'Y'.
 88 EOF-NO VALUE 'N'.
 77 WS-ERR PIC 9(4) VALUE 0.
 77 WS-COUNT PIC 9(9) VALUE 0.

01 WS-CONTEXT.
 05 WS-APP-NAME PIC X(12) VALUE 'FIN-RUN-01'.
 05 WS-ENV PIC X(04) VALUE 'PROD'.
 05 WS-DATE PIC 9(8).

01 WS-CLIENT.
 05 WS-CLIENT-ID PIC 9(10).
 05 WS-CLIENT-NAME PIC X(30).
 05 WS-CLIENT-SEG PIC X(02).
 05 WS-CLIENT-RISK PIC 9(2).

01 WS-AMOUNTS.
 05 WS-AMOUNT PIC 9(7)V99.
 05 WS-FEE PIC 9(5)V99.

```
05 WS-TOTAL      PIC 9(7)V99.  
01 WS-SENSITIVE.  
  05 WS-ACCOUNT-NO  PIC 9(12).  
  05 WS-IBAN        PIC X(27).  
  05 WS-PAN         PIC 9(16).  
01 WS-BUFFER.  
  05 WS-RAW         PIC X(80).  
  05 WS-RAW-ALT REDEFINES WS-RAW.  
    10 WS-RAW-ID    PIC 9(10).  
    10 WS-RAW-AMT   PIC 9(7)V99.  
    10 WS-RAW-STAT  PIC X.  
    10 WS-RAW-FILL  PIC X(60).  
01 WS-TABLE.  
  05 WS-ITEM OCCURS 5 TIMES.  
    10 WS-ITEM-CODE PIC X(03).  
    10 WS-ITEM-VAL  PIC 9(5)V99.  
01 WS-MSG.  
  05 WS-MSG-TEXT   PIC X(120).
```

PROCEDURE DIVISION.

MAIN.

```
  PERFORM INIT  
  PERFORM OPEN-FILES  
  PERFORM READ-LOOP UNTIL EOF-YES  
  PERFORM CLOSE-FILES  
  GOBACK.
```

INIT.

```
  ACCEPT WS-DATE FROM DATE YYYYMMDD  
  MOVE 'A01' TO WS-ITEM-CODE(1)  
  MOVE 'B02' TO WS-ITEM-CODE(2)  
  MOVE 'C03' TO WS-ITEM-CODE(3)  
  MOVE 10.00 TO WS-ITEM-VAL(1)  
  MOVE 20.50 TO WS-ITEM-VAL(2)  
  MOVE 30.75 TO WS-ITEM-VAL(3)
```

OPEN-FILES.

```
  OPEN INPUT INFILE  
  OPEN OUTPUT OUTFIL
```

READ-LOOP.

```
  READ INFILE  
  AT END  
    SET EOF-YES TO TRUE  
  NOT AT END  
    ADD 1 TO WS-COUNT  
    PERFORM MAP-IN  
    PERFORM BUSINESS-RULES  
    PERFORM BUILD-OUTPUT  
    WRITE OUT-REC  
  END-READ
```

MAP-IN.

```
  MOVE IN-CLIENT-ID TO WS-CLIENT-ID
```

```
MOVE IN-AMOUNT TO WS-AMOUNT
MOVE IN-REC TO WS-RAW
```

.

BUSINESS-RULES.

```
EVALUATE IN-OP-CODE
  WHEN 'C'
    PERFORM RULE-CREATE
  WHEN 'U'
    PERFORM RULE-UPDATE
  WHEN 'D'
    PERFORM RULE-DELETE
  WHEN OTHER
    PERFORM RULE-UNKNOWN
END-EVALUATE
```

```
IF IN-STATUS = 'X'
  ADD 1 TO WS-ERR
  MOVE 'STATUS X: BLOCKED' TO WS-MSG-TEXT
END-IF
```

```
PERFORM CALC-FEES
```

.

RULE-CREATE.

```
MOVE 'NEW CLIENT' TO WS-CLIENT-NAME
MOVE '01'      TO WS-CLIENT-SEG
MOVE 10       TO WS-CLIENT-RISK
```

.

RULE-UPDATE.

```
MOVE 'UPDATE' TO WS-CLIENT-NAME
MOVE '02'      TO WS-CLIENT-SEG
MOVE 20       TO WS-CLIENT-RISK
```

.

RULE-DELETE.

```
MOVE 'DELETE' TO WS-CLIENT-NAME
MOVE '99'      TO WS-CLIENT-SEG
MOVE 90       TO WS-CLIENT-RISK
```

.

RULE-UNKNOWN.

```
ADD 1 TO WS-ERR
MOVE 'UNKNOWN OPERATION' TO WS-MSG-TEXT
```

.

CALC-FEES.

```
IF WS-AMOUNT > 1000.00
  MOVE 12.50 TO WS-FEE
ELSE
  MOVE 2.50 TO WS-FEE
END-IF
COMPUTE WS-TOTAL = WS-AMOUNT + WS-FEE
```

.

BUILD-OUTPUT.

```
STRING
  'APP=' WS-APP-NAME
  'ENV=' WS-ENV
  'DATE=' WS-DATE
```

(suite...)

```
' ID=' WS-CLIENT-ID
' SEG=' WS-CLIENT-SEG
' RISK=' WS-CLIENT-RISK
' AMT=' WS-AMOUNT
' FEE=' WS-FEE
' TOTAL=' WS-TOTAL
DELIMITED BY SIZE
INTO OUT-REC
END-STRING
```

```
CLOSE-FILES.
CLOSE INFILE OUTFILE
```

Journal de génération

Date de création : 2026-01-25 22:46:11

Fichier analysé : BOSQWEB_DEMO.cbl

Hash SHA1 : f1413dd7fece332ca151a243eb97920b36deb2d0

Auteur : Bosqweb Docobol

Table des matières

- Synthèse décisionnelle page 2
- Preuves techniques page 3
- Résumé technique page 4
- Résumé exécutif page 5
- Documentation technique page 6
- Annexes page 10
- Journal de génération page 14

Contact : infos@bosqweb.net