



VisualAge Pacbase 2.5

**DSMS ACCESS FACILITY
MANUEL DE REFERENCE**

DDSM000251F

Remarque

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section "Remarques" de la page suivante.

En application de votre contrat de licence, vous pouvez consulter ou télécharger la documentation de VisualAge Pacbase, régulièrement mise à jour, à partir du site Web du Support Technique :

<http://www.software.ibm.com/ad/vapacbase/support.htm>

La section Catalogue dans la page d'accueil de la Documentation vous permet d'identifier la dernière édition disponible du présent document.

Première Edition (Août 1998)

La présente édition s'applique à :

- VisualAge Pacbase Version 2.5

Vous pouvez nous adresser tout commentaire sur ce document (en indiquant sa référence) via le site Web de notre Support Technique à l'adresse suivante :

<http://www.software.ibm.com/ad/vapacbase/support.htm>

ou en nous adressant un courrier à :

IBM Paris Laboratory
Support VisualAge Pacbase
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

IBM pourra disposer comme elle l'entendra des informations contenues dans vos commentaires, sans aucune obligation de sa part.

© Copyright International Business Machines Corporation 1983, 1999. Tous droits réservés.

REMARQUES

Ce document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM. Cela ne signifie pas qu'IBM ait l'intention de les annoncer dans tous les pays où la compagnie est présente.

Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM.

Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

Intellectual Property and Licensing
International Business Machines Corporation
North Castle Drive, Armonk, New-York 10504-1785
USA

Les détenteurs de licences du présent produit souhaitant obtenir des informations sur celui-ci à des fins : (i) d'échange d'informations entre des programmes développés indépendamment et d'autres programmes (y compris celui-ci) et (ii) d'utilisation mutuelle des informations ainsi échangées doivent s'adresser à :

IBM Paris Laboratory
Département SMC
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

De telles informations peuvent être mises à la disposition du Client et seront soumises aux termes et conditions appropriés, y compris dans certains cas au paiement d'une redevance.

IBM peut modifier ce document, le produit qu'il décrit ou les deux.

MARQUES

IBM est une marque d'International Business Machines Corporation, Inc. AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, PACBASE, RACF, RS/6000, SQL/DS, TeamConnection et VisualAge sont des marques d'International Business Machines Corporation, Inc. dans certains pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.

UNIX est une marque enregistrée aux Etats-Unis et/ou dans d'autres pays et utilisée avec l'autorisation exclusive de la société X/Open Company Limited.

D'autres sociétés peuvent être propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

TABLE DES MATIERES

1. INTRODUCTION	7
2. FONCTIONNEMENT ET UTILISATION DANS LES PROGRAMMES.....	11
2.1. FONCTIONNEMENT GENERAL	12
2.2. SYNTAXE DU LANGAGE DAF-SQL.....	18
2.3. OPTIMISATION DES ACCES A LA BASE	28
2.4. RECOMMANDATION POUR LE PARAMETRE IDENT	31
2.5. MISE EN OEUVRE SOUS VISUALAGE PACBASE.....	32
2.6. LE PROGRAMME UTILISATEUR "TRANSLATE"	33
2.7. IMBRICATION DES CURSEURS DAF.....	38
2.8. EXECUTION DU PROGRAMME UTILISATEUR	40
3. EXEMPLES DE PROGRAMME UTILISANT DAF	41
3.1. INTRODUCTION	42
3.2. EXEMPLE BATCH	43
3.3. EXEMPLE TP	50
4. MISE EN OEUVRE DE DAF PAR MATERIEL	61
4.1. VERSION MVS/CICS	62
4.2. VERSION CICS/DOS/VS	63
4.3. VERSION IMS/VS/ESA	64
4.4. VERSION GCOS7/TDS.....	66
4.5. VERSION GCOS8 DMIV-TP/TP8	67
4.6. VERSION UNISYS 2200.....	68
4.7. VERSION POUR UNIX, OS2 ET WINDOWS/NT	69
5. MESSAGES D'ERREUR.....	72
5.1. LE TRANSLATEUR.....	73
5.2. L'EXTRACTEUR.....	76

VisualAge Pacbase - Manuel de Référence
DSMS ACCESS FACILITY
INTRODUCTION

PAGE 7

1

1. INTRODUCTION

PRESENTATION DU MODULE

Le module DAF (DSMS Access Facility) permet d'extraire par programme les informations contenues dans la Base DSMS à l'aide de requêtes SQL.

VISION RELATIONNELLE DE LA BASE

DAF fournit une description relationnelle du métamodèle DSMS, préalable indispensable à l'expression d'une requête SQL. Un modèle relationnel organise les données dans des tables structurées en colonnes. Les relations entre objets sont des correspondances entre des colonnes de tables.

Dans le modèle tabulaire DAF :

- Chaque entité est décrite par un ensemble de tables,
- Les informations relatives à la définition et à la description de chaque entité sont décrites par des colonnes,
- Les chaînages peuvent être décrits de deux manières :
 - . Par des relations directes entre des colonnes de tables,
 - . Par des tables dites "virtuelles" dont les colonnes sont, d'une part les colonnes identifiant les deux tables à mettre en relation, d'autre part les colonnes représentatives du chaînage.

DESCRIPTION DES TABLES

Le Manuel "Tables DAF" donne la description des tables et colonnes du métamodèle DSMS.

PRESENTATION DU LANGAGE SQL

Le langage SQL (pour Structured Query Language) est actuellement le langage standard d'interrogation des bases de données relationnelles. Il permet de formuler des requêtes dans une syntaxe proche de la langue anglaise, de récupérer l'information ainsi définie et de mettre à jour une base.

Une requête SQL est définie par une clause `SELECT` (voir le sous-chapitre "Syntaxe du Langage PAF-SQL"). Les tables d'une base de données relationnelle sont constituées de rangs (assimilables à des enregistrements pour les fichiers). Une requête SQL définit un sous-ensemble parmi les informations d'une base, sans préciser a priori quand ou comment ces informations sont obtenues et traitées. A ce titre, SQL peut être considéré comme un langage "ensembliste".

Afin d'intégrer SQL dans les langages procéduraux utilisés en informatique de gestion, ce langage permet d'associer la notion de curseur à une requête. Le curseur permet de préciser les accès à l'information définie dans la requête associée lors de la déclaration du curseur.

SQL fournit donc les ordres de manipulation de curseur permettant de lire dans la base les informations définies par la requête (`OPEN`), de retourner un par un les rangs dans une zone de travail du programme utilisant SQL (`FETCH`) et enfin de terminer l'utilisation du curseur (`CLOSE`).

Ainsi, une fois le curseur déclaré, les ordres de manipulation de curseur permettent de voir un curseur comme un fichier, l'ordre `FETCH` étant l'équivalent du `READ` en COBOL.

REQUETES SQL STANDARD

Avec le module DAF, la base DSMS devient accessible par le langage de requêtes standard dans le monde des bases de données relationnelles.

A partir du modèle tabulaire décrit, les accès à la base s'effectuent par la déclaration et l'utilisation de curseurs SQL.

Les requêtes peuvent être décrites dans tous les programmes utilisateur, batch ou TP, écrits directement en COBOL ou générés par VisualAge Pacbase. Dans tous les cas, les requêtes sont traitées par le traducteur DAF, en amont du compilateur COBOL, qui traduit les requêtes en ordres d'appel du sous-programme d'extraction DAF.

UN SOUS-PROGRAMME D'EXTRACTION

Un sous-programme DAF gère les accès à la base DSMS. Il récupère les paramètres internes construits par le traducteur lui permettant de réaliser l'extraction demandée.

Les informations extraites sont transmises au programme utilisateur dans la zone de communication générée par le traducteur (zone de communication COBOL en Working Storage Section).

VisualAge Pacbase - Manuel de Référence	PAGE	11
DSMS ACCESS FACILITY		
FONCTIONNEMENT ET UTILISATION DANS LES PROGRAMMES		2

2. FONCTIONNEMENT ET UTILISATION DANS LES PROGRAMMES

2.1. FONCTIONNEMENT GENERAL

FONCTIONNEMENT GENERAL

LE PROGRAMME UTILISATEUR

La mise en oeuvre de DAF est possible à travers un programme batch ou conversationnel, écrit directement en COBOL ou généré par VisualAge Pacbase. Dans tous les cas, c'est le sous-programme d'extraction DAF qui gère les accès à la base.

Dans son programme, l'utilisateur doit d'abord déclarer (DECLARE CURSOR) le ou les curseurs lui permettant l'accès aux tables souhaitées. Ensuite, pour chaque curseur déclaré, la séquence d'ordres est la suivante :

CONNECT connexion du curseur à un contexte DSMS (Utilisateur, Produit et Filiale - ces 2 derniers critères ne conditionnant pas les extractions)

OPEN ouverture du curseur, c'est-à-dire extraction préliminaire des informations de la base,

FETCH restitution, un par un, des enregistrements extraits,

CLOSE fermeture du curseur.

L'ordre SET permet la modification dynamique des paramètres de fonctionnement du translateur.

Enfin, les ordres INIT et QUIT permettent d'effectuer les actions d'initialisation et de terminaison techniques en fonction du mode d'extraction et du type de matériel (par exemple ouverture et fermeture de fichiers en BATCH).

LE TRANSLATEUR DAF

Les appels à DAF ne sont donc pas écrits en pur COBOL, des ordres spécifiques en langage SQL sont insérés dans le programme utilisateur. Le rôle du traducteur est donc de transformer les phrases SQL-PAF en COBOL avant la compilation. Tout comme le préprocesseur CICS par exemple, le traducteur DAF place en commentaire les phrases SQL-PAF et les traduit en COBOL à la suite de ces commentaires.

Le traducteur traduit les ordres SQL-PAF en déclarations (pour l'ordre DECLARE) et en appels au sous-programme d'extraction pour les autres ordres (sauf pour un ordre très spécifique et non SQL : l'ordre SET).

Ces zones permettent également à l'extracteur de retourner l'information demandée dans la requête SQL-PAF. Une description plus détaillée des zones déclarées et des différents ordres SQL-PAF est donnée plus loin.

Le traducteur édite un compte rendu des erreurs décelées.

Le traducteur est paramétré par une carte commentaire insérée en ligne 2 (après la ligne IDENTIFICATION DIVISION). Cette ligne, décrite ci-dessous, est générée automatiquement par le générateur VisualAge Pacbase à la rencontre d'un opérateur EXP dans une ligne -P du programme (voir le sous-chapitre "Mise en oeuvre sous VisualAge Pacbase").

La ligne de paramètres a le format suivant (à partir de la colonne 8, la colonne 7 contenant l'astérisque des commentaires COBOL) :

POS	LONG	CONTENU
1	6	Numéro de ligne COBOL
7	1	Commentaire
8	5	Mode d'exécution (BATCH ou TP)
14	4	Libellé fixe
19	3	Code bibliothèque
23	5	Numéro de session - état de version
29	2	Variante(s) de génération (COBOL et MAP)
32	3	Libellé fixe
36	1	Code langue de la base (F ou A)
38	3	Squelette programmes Langage batch Squelette programmes Dialogue Squelette programmes Gén. COBOL
42	1	Langage du squelette (F ou A)
44	6	Délimiteur (SINGLE ou DOUBLE)

Le mode d'exécution permet de distinguer le BATCH du TP. Le mode pris par VisualAge Pacbase est simplement fonction du générateur mis en oeuvre. Ce mode permet au traducteur de déclarer ou non des zones de travail spécifiques au TP et de générer les appels au bon sous-programme d'extraction.

La variante de génération est prise par VisualAge Pacbase au niveau de la fiche Programme. Elle permet d'adapter la syntaxe générée en fonction du compilateur.

Le délimiteur de chaîne, pris par VisualAge Pacbase dans la fiche Bibliothèque, permet au traducteur de connaître le bon délimiteur de chaîne, à la fois pour la génération et pour l'analyse du source DAF.

Notons que le traducteur est un programme bilingue. Le premier code langue spécifie la langue des messages d'erreur du traducteur et le second la langue de représentation du Dictionnaire (langue de codage des tables et des colonnes). Ces deux codes langues permettent à des sites de générer des programmes pour des sites travaillant dans une autre langue. Les valeurs prises par VisualAge Pacbase pour ces deux codes langues sont respectivement ceux des fichiers AE et SG (libellés d'erreur et squelette de génération).

La ligne 2 générée par le générateur VisualAge Pacbase est lue par le traducteur DAF mais n'est pas reproduite dans le source traduit.

L'ordre SET permet de modifier facilement certains de ces paramètres à n'importe quel moment de la traduction.

L'EXTRACTEUR DAF

Le sous-programme extracteur gère les accès à la base DSMS. Il récupère les paramètres internes construits par le translateur (zone de spécifications où la requête SQL est traduite) et réalise l'extraction demandée de la façon suivante :

- * A la rencontre d'un ordre CONNECT, il établit la connexion de l'utilisateur pour le curseur précisé (vérification des autorisations définies sur les écrans TUD, TUG, TUP et TUS)
- * A la rencontre d'un ordre OPEN, il accède à la base et stocke les enregistrements extraits dans le fichier intermédiaire de travail. Le nombre d'enregistrements lus en avance et stockés dans le fichier de travail est paramétrable pour chaque curseur (paramètre SIZE dans l'ordre CONNECT).
- * A la rencontre d'un ordre FETCH, l'extracteur restitue un par un les enregistrements stockés dans le fichier intermédiaire (fetching d'un enregistrement). Les enregistrements sont transmis au programme utilisateur dans la zone de communication générée par le translateur.

Le sous-chapitre "Optimisation des accès à la base" explique plus en détail le mécanisme utilisé par l'extracteur pour gérer les ordres OPEN et FETCH.

- * L'ordre CLOSE provoque la fermeture du curseur précisé.

L'extracteur DAF est également utilisé dans la procédure DUPD. Cette procédure effectue la mise à jour batch de la base DSMS à partir du fichier séquentiel des Tables DAF.

Pour plus d'informations, référez-vous au manuel d'exploitation "Système de Gestion, de Développement, de Suivi 2.5 - IBM - CICS / OS / VS", chapitre "Mise à jour batch à partir de Tables DAF (DUPD)".

LE FICHIER INTERMEDIAIRE DE TRAVAIL

Le fichier intermédiaire de travail sert donc à stocker les enregistrements extraits de la base, suite à la rencontre d'un ordre OPEN (ou FETCH lorsqu'ont été restitués au programme utilisateur tous les enregistrements du fichier).

Les enregistrements stockés sont ensuite restitués, un par un, à la rencontre de chaque ordre FETCH.

Le nombre maximum d'enregistrements lus en avance pour un curseur est paramétrable (paramètre SIZE de l'ordre CONNECT, se reporter aux sous-chapitres "Syntaxe du Langage PAF-SQL" et "Optimisation des Accès à la Base").

Outre les données lues en avance, le fichier intermédiaire permet de stocker des paramètres techniques pour la gestion des curseurs par l'extracteur.

Le fichier intermédiaire de travail est un fichier séquentiel indexé au format variable. Les enregistrements qui y sont stockés sont liés à un utilisateur et à un curseur SQL.

La structure de la clé est légèrement différente en batch et en TP :

Traitements batch

Le fichier intermédiaire de stockage est défini pour un job batch.

Clé d'accès

- . Code du curseur,
- . Code structure,
- . Numéro d'enregistrement.

Traitements conversationnels

Un fichier de travail unique est défini pour toutes les conversations, tous utilisateurs confondus.

Clé d'accès

- . Identifiant de la conversation,
- . Code du curseur,
- . Code structure,
- . Numéro d'enregistrement.

Cas particulier de CICS

L'environnement conversationnel CICS permettant la coexistence de plusieurs bases DSMS, un fichier de travail est défini pour chacune des bases sous CICS.

2.2. SYNTAXE DU LANGAGE DAF-SQL

SYNTAXE DU LANGAGE PAF-SQL

Cette version du module DAF étant un outil d'extraction, le sous-ensemble SQL retenu est restreint aux ordres d'interrogation d'une base de données. Ainsi, les aspects création, modification et destruction de tables ne sont pas pris en compte.

GENERALITES SUR LE LANGAGE

Les accès à la base DSMS par le module DAF se font par la déclaration et l'utilisation de curseurs SQL.

Pour chaque table à laquelle il veut accéder, l'utilisateur doit d'abord déclarer un curseur en WORKING STORAGE SECTION (DECLARE CURSOR). Ensuite, en PROCEDURE DIVISION, la séquence des ordres associée à un curseur donné est la suivante : CONNECT, OPEN, FETCH et CLOSE. OPEN, FETCH et CLOSE sont des ordres standard du langage SQL, tandis que CONNECT est spécifique au langage PAF-SQL. Ces quatre ordres sont désignés dans la suite comme les ordres de manipulation de curseurs dans la syntaxe PAF-SQL.

Il est possible d'utiliser jusqu'à 100 curseurs dans un programme applicatif.

Enfin, les ordres INIT (initialisations) et QUIT (terminaisons), indépendants des curseurs, doivent être exécutés respectivement avant et après tous les autres ordres de manipulation de curseur.

Toute phrase PAF-SQL doit commencer en colonne 12 du COBOL, débiter par EXEC PAF et se terminer par END-EXEC.

DECLARATION D'UN CURSEUR

Ecrite en WORKING STORAGE SECTION, la clause DECLARE CURSOR permet de déclarer un curseur SQL.

Dans cette déclaration, les mots-clés retenus sont SELECT, FROM, WHERE, AND et OR.

Sa syntaxe est la suivante (les parties entre parenthèses sont facultatives) :

```
EXEC PAF DECLARE <code-curseur> CURSOR FOR  
SELECT * FROM <code-table> ( WHERE <liste de conditions> )  
END-EXEC
```

avec :

- <code-curseur> est le code du curseur sur 4 caractères,
- SELECT ne permet pas la sélection de colonnes, autrement dit l'accès à une table ramène l'intégralité de ses colonnes dans le programme utilisateur. La seule syntaxe autorisée est donc "SELECT *".
- FROM n'autorise pas la demande d'une jointure de tables et n'est donc suivi que par un seul code table.
- <code-table> est le code de la table DAF:

Le Manuel "Tables DAF" fournit la description des tables du métamodèle DSMS.

- WHERE n'autorise pas les SUBSELECTs.

- <liste de conditions> est une suite de conditions, chacune portant sur une colonne de la table, reliées entre elles par des parenthèses et les connecteurs logiques AND et OR. Le nombre de conditions élémentaires est limité à 50. Chaque condition s'écrit sous la forme COLONNE OPERATEUR OPERANDE, où :

. COLONNE est le code d'une colonne de la table,

. OPERATEUR est un des opérateurs suivants :

= égal,
> supérieur,
>= supérieur ou égal,
< inférieur,
<= inférieur ou égal,
<> différent.

. OPERANDE est soit :

- une autre colonne de la table,
- une constante COBOL,
- une variable COBOL du programme.

NOTE : Les constantes alphanumériques ne doivent pas dépasser 60 caractères. Dans le cas où cette taille est insuffisante, il suffit d'utiliser une variable COBOL convenablement initialisée. Les constantes numériques sont limitées aux constantes entières non signées de 18 digits maximum. Enfin le translateur ne vérifie pas la déclaration des variables COBOL utilisées comme opérande. Attention, il n'est pas possible de créer des variables COBOL indicées.

Les contraintes sur la clause SELECT ne sont pas restrictives : d'une part, la possibilité d'ouvrir plusieurs curseurs pallie l'inconvénient des accès monotables ; d'autre part, il est souvent délicat d'exprimer une requête très complexe en SQL. Enfin, l'objectif de DAF n'est pas de fournir une interface SQL exhaustive mais de permettre d'accéder à toute l'information contenue dans la Base DSMS. C'est au niveau du programme utilisateur que sont structurés les accès à l'information.

Pour de plus amples informations, voir le Sous-Chapitre "Imbrication des Curseurs DAF" dans ce Chapitre.

MANIPULATION D'UN CURSEUR

Les ordres de manipulations de curseurs s'écrivent en PROCEDURE DIVISION.

CONNECT

Premier dans la séquence des ordres de manipulation d'un curseur, l'ordre CONNECT permet d'établir la connexion à DSMS. A tout moment, il est possible de connecter ou de re-connecter un curseur, pour modifier la size par exemple.

Syntaxe :

```
EXEC PAF CONNECT <code-curseur> TO  
    USER = <code-utilisateur>  
    PASS = <mot-de-passe>  
    PRODUCT = <code-produit>  
    SUBSIDIARY = <code-filiale>  
    SIZE = <nombre>  
    IDENT = <identifiant-TP>  
    BASE = <code-base>  
END-EXEC
```

Les paramètres désignés à la droite du symbole = et décrits ci-dessous doivent être soit des littéraux, soit des variables du langage COBOL.

USER code utilisateur DSMS,

PASS mot de passe DSMS,

PRODUCT code Produit DSMS,

SUBSIDIARY code Filiale DSMS,

SIZE nombre maximum d'enregistrements stockés dans le fichier intermédiaire de travail,

IDENT identifiant banalisé de conversation en TP,

BASE code transaction DSMS sur 4 caractères.

Tous les paramètres du CONNECT sont obligatoires lors de la première connexion de chaque curseur. En effet, l'ordre CONNECT permet de contrôler l'utilisateur et son mot de passe.

Les deux paramètres IDENT et BASE ne sont utiles qu'en TP. Par conséquent, le translateur DAF les ignore lors de la précompilation d'un programme utilisateur de type BATCH.

Le paramètre IDENT est une zone banalisée de 25 caractères permettant d'identifier la conversation TP utilisant PAF. Elle fait partie de la clé d'accès au fichier intermédiaire. La valeur prise par ce paramètre dépend naturellement du moniteur TP. Par exemple, sous CICS, il est recommandé d'alimenter ce paramètre à l'aide de la variable DFHTERMID fournie en standard par CICS. Un ensemble de recommandations pour alimenter de façon standard la variable IDENT en fonction du système d'exploitation est proposé dans le sous-chapitre "Recommandations pour le paramètre IDENT".

Le paramètre BASE contient le code transaction d'accès à la base DSMS sur quatre caractères. Il permet l'accès en TP aux fichiers de la base ainsi qu'au fichier de travail DAF. (Ce paramètre n'est utilisé que pour le matériel IBM sous CICS).

REMARQUE : PARAMETRE SIZE

Le paramètre SIZE doit être alimenté par une quantité strictement positive. Dans le cas contraire, il est repositionné à la valeur arbitraire 1 par l'extracteur DAF.

OPEN

A la rencontre d'un ordre OPEN, l'extracteur accède à la base DSMS et écrit, dans le fichier intermédiaire de stockage, les enregistrements sélectionnés en fonction de la déclaration du curseur.

Un curseur peut être ouvert s'il a été connecté auparavant et s'il n'a pas été déjà ouvert.

Syntaxe :

EXEC PAF OPEN <code-curseur> END-EXEC

FETCH

A la rencontre d'un ordre FETCH, l'extracteur restitue au programme utilisateur un enregistrement du fichier intermédiaire.

Le curseur doit être ouvert ; pour ce curseur, l'utilisateur écrit autant de FETCH que nécessaire.

Syntaxe :

EXEC PAF FETCH <code-curseur> END-EXEC

Remarque :

La syntaxe standard de l'ordre FETCH dans le langage SQL prévoit le mot-clé INTO, permettant d'associer à chaque colonne sélectionnée dans le SELECT du DECLARE CURSOR une zone COBOL. L'extracteur PAF-SQL restituant l'ensemble des colonnes dans une zone de communication, générée en WORKING STORAGE SECTION à la rencontre de l'ordre DECLARE, le mot-clé INTO, suivi de la liste des zones COBOL cibles, n'est pas conservé dans la syntaxe de l'ordre FETCH.

CLOSE

Il n'est possible de fermer qu'un curseur ouvert. L'ordre CLOSE suit donc un ordre OPEN ou un FETCH.

Syntaxe :

EXEC PAF CLOSE <code-curseur> END-EXEC

LES PARAMETRES DU TRANSLATEUR ET L'ORDRE SET

SET

Cet ordre permet la modification dynamique de certains paramètres pour le fonctionnement du translateur. Pour tout programme utilisateur traduit, ces paramètres sont initialisés par la deuxième ligne (en commentaire, après IDENTIFICATION DIVISION, voir le Sous-Chapitre "Fonctionnement général").

A la rencontre de l'ordre SET, le translateur, après avoir mis à jour les paramètres précisés, reporte dans le programme traduit l'ordre en commentaire et ne génère rien de plus. SET est donc le seul ordre DAF à ne pas être traduit par des phrases COBOL.

Cet ordre peut être écrit en WSS ou en Procedure Division. Toutefois, si ces paramètres ne doivent pas changer au cours du programme, il est conseillé d'écrire l'ordre SET en début de Working Storage Section.

Syntaxe :

```
EXEC PAF SET
    LIB = <code-bibliothèque>
    DELIM = <délimiteur>
    TYPE = <variantes-génération>
    MODE = <mode-exécution>
    LINK = <mode-d'appel>
    ROOT = <racine>
    SESSION = <session-et-version>
END-EXEC
```

Cette syntaxe est du même type que celle de l'ordre CONNECT. En particulier, il n'est pas obligatoire d'affecter une nouvelle valeur à l'ensemble des paramètres LIB, DELIM, TYPE, MODE et ROOT, ni de les écrire dans cet ordre.

Les paramètres situés à la droite du signe = et décrits ci-dessous ne sont pas, contrairement au cas du CONNECT, des constantes ou des variables COBOL. Les paramètres sont donnés par leur valeur alphanumérique pure, sans délimiteur de chaînes. Par exemple, le code Bibliothèque étant une chaîne de 3 caractères, il faut écrire pour modifier ce dernier :

```
EXEC PAF SET LIB = T00 END-EXEC
```

Description des paramètres :

LIB (3 caractères)

Permet de modifier la Bibliothèque (par défaut, c'est la Bibliothèque de génération).

DELIM (6 caractères)

Permet de modifier le délimiteur (DELIM = SINGLE pour la quote ' ou DELIM = DOUBLE pour la double quote ").

TYPE (2 caractères)

Permet de modifier le type de COBOL à générer en terme de variante / sous-variante VisualAge Pacbase. Ceci est utile par exemple pour remplacer IBM-CICS (00 ou X0) par IBM-IMS (01 ou X1) quand la distinction n'existe pas (génération BATCH), les appels de sous-programmes accédant à des fichiers étant spécifiques en IMS. Voir les valeurs de ces variantes dans les Manuels de Référence "Module Langage Batch" et "Dialogue Général".

MODE (5 caractères)

Permet de modifier le mode d'exécution (BATCH ou TP), pour ceux utilisant le générateur BATCH pour écrire des programmes TP.

LINK (6 caractères)

Permet de spécifier le mode d'appel de l'extracteur (STATIC ou DYNAM) respectivement pour des appels statiques ou dynamiques. Pour CICS, ce paramètre n'est pas pris en compte et le traducteur génère un ordre LINK pour l'appel à l'extracteur. Pour DOS, en BATCH, l'appel est toujours dynamique et s'exécute via le module PACDYNAM. Pour les matériels BULL DPS7 et DPS8, l'ordre est ignoré dans le cas du TP, où l'appel est toujours statique.

ROOT (2 caractères)

Permet de modifier les deux premiers caractères des noms externes des sous-programmes d'extraction. Le paramètre s'appelle ROOT car une façon standard de paramétrer ces sous-programmes est d'affecter aux deux premiers caractères le radical de la base DSMS d'extraction.

SESSION (5 caractères)

Permet de modifier la session (session de génération VisualAge Pacbase par défaut).

INITIALISATIONS ET TERMINAISONS TECHNOLOGIQUES

Les ordres INIT et QUIT permettent à l'extracteur d'effectuer respectivement les initialisations et terminaisons technologiques, dépendant du système d'exploitation. Les ordres INIT et QUIT doivent donc être exécutés respectivement avant et après tout autre ordre de DAF.

INIT

EXEC PAF INIT END-EXEC

QUIT

EXEC PAF QUIT END-EXEC

2.3. OPTIMISATION DES ACCES A LA BASE

OPTIMISATION DES ACCES A LA BASE

CHOIX DE LA TABLE

Il dépend de l'utilisation future des données extraites.

Pour un programme qui ne devra exploiter que des colonnes contenues dans une table du type ENTX....., il est préférable d'appeler cette table plutôt que ENTDEF WHERE En outre, la plupart des références croisées n'existant que si l'entité chaînée est présente (valeur différente de blanc ou zéro), celles-ci ne sont utilisées que pour les tables du type ENTX.....

EXEMPLE : ACCES AUX AMELIORATIONS AYANT LA GRAVITE 'X'

```
. SELECT * FROM AMEXGRA WHERE CGRA = 'X'
```

Références croisées sur la Gravité et sur les Améliorations.

```
. SELECT * FROM AMEDEF WHERE CGRA = 'X'
```

Liste des améliorations puis test sur la valeur de la gravité (CGRA).

INTERET DE LA CLAUSE WHERE

L'utilisation de la clause WHERE permet d'améliorer les performances à deux niveaux ; optimisation des accès à la Base et réduction du volume des données extraites stockées dans le fichier intermédiaire SYSDAF.

INTERET DU PARAMETRE SIZE DANS L'ORDRE CONNECT

Afin d'optimiser les accès à la base DSMS, l'extracteur permet de lire plusieurs enregistrements en avance. L'intérêt de ces lectures en avance est de ne pas être pénalisé par une reprise de lecture à chaque fois qu'on fetch un enregistrement. En effet, ces lectures provoquent un repositionnement systématique (START) sur les fichiers de la base.

Ces lectures en avance peuvent donc être vues comme l'équivalent logique des BUFFERS d'entrée-sortie utilisés par les méthodes d'accès aux fichiers dans tout système d'exploitation.

Pour chaque curseur, DAF autorise le paramétrage de ce nombre de lectures en avance par le paramètre SIZE de l'ordre CONNECT.

GESTION DES ENREGISTREMENTS LUS PAR L'EXTRACTEUR

Pour mieux comprendre la façon d'optimiser les accès à la base, il est nécessaire de détailler le comportement de l'extracteur en ce qui concerne les lectures en avance. Deux ordres peuvent provoquer des lectures de la base : OPEN et FETCH.

La fonction de lecture de l'extracteur consiste à lire la base, à vérifier la validité de l'enregistrement lu et à le stocker dans le fichier de travail s'il est valide. Ce cycle est répété tant que le nombre d'enregistrements stockés est inférieur au nombre donné par le paramètre SIZE. Le traitement peut cependant s'arrêter si la fonction de lecture détecte la fin des enregistrements pour le curseur. Dans tous les cas, la fonction de lecture retourne le nombre d'enregistrements effectivement lus et un indicateur de fin de curseur.

Cette fonction de lecture est systématiquement exécutée à l'ouverture du curseur. Elle peut également l'être lors du traitement d'un ordre FETCH. Le fetching consiste essentiellement à lire le fichier de travail pour restituer un nouvel enregistrement. Dans le cas où l'enregistrement restitué est le dernier lu par la fonction de lecture, celle-ci est à nouveau exécutée si elle n'a pas précédemment détecté la fin du curseur.

OPTIMISATION PAR LE PARAMETRE SIZE

Le paramètre SIZE est donc essentiel dans l'efficacité de l'extracteur DAF en terme de temps de réponse.

Ainsi, en TP, il est pénalisant de prendre un SIZE inférieur au nombre d'enregistrements fetchés par écran (nombre d'enregistrements affichés sur l'écran par exemple). Dans l'hypothèse simplificatrice d'un curseur non sélectif (tous les enregistrements lus sont valides), le SIZE optimal est un multiple du nombre d'enregistrements fetchés par écran. Dès lors que la conversation peut être interrompue à chaque réception d'écran, le SIZE optimal est effectivement égal au nombre d'enregistrements fetchés par écran.

En BATCH, le problème doit être pris en sens inverse. Dans le cas d'un curseur qu'on doit fetcher jusqu'à son terme, il semble intéressant, à première vue, de choisir un paramètre SIZE très grand pour minimiser le nombre de reprises de lecture. Cependant, des valeurs de SIZE trop importantes provoquent la création dynamique d'une quantité d'enregistrements pénalisante pour la plupart des méthodes d'accès aux fichiers séquentiels indexés. L'idéal en BATCH est de définir une taille de BUFFER d'entrée-sortie pour le fichier de travail suffisante pour qu'une exécution de la fonction de lecture ne provoque que des entrées-sorties logiques.

Ces quelques remarques devraient permettre aux programmes utilisant DAF de fonctionner avec des temps d'exécution ou des temps de réponse satisfaisants.

2.4. RECOMMANDATION POUR LE PARAMETRE IDENT

RECOMMANDATIONS POUR LE PARAMETRE IDENT

PRINCIPE GENERAL

Le principe général pour l'alimentation du paramètre IDENT est d'identifier de façon unique une conversation en environnement multi-utilisateurs. Cette identification est donc naturellement dépendante en grande partie du moniteur TP sous lequel s'exécute le programme translaté.

Lorsque cela est possible, les recommandations ci-dessous tiennent compte des variables standard fournies par le moniteur TP (identifiant du terminal, etc...).

CICS-MVS ou DOS

La variable CICS EIBTRMID identifie chaque terminal.

IMS

Le nom du terminal logique se trouve dans la zone IO/PCB (variable S-IPCB-XNMTE dans DIALOGUE).

DPS7 et DPS8

La variable associée au SYMBOLIC SOURCE dans la COMMUNICATION SECTION (7-CD01-XTERM en DIALOGUE).

ICL

La variable système n'a pas de nom standard. En DIALOGUE, il s'agit de la zone TERMINAL-NAME dans l'INPUT-MESSAGE.

UNISYS SERIES A

La variable système s'appelle COMS-IN-STATION.

Environnement MS-DOS/OS2

Le fichier de travail est géré localement et le mode TP est identique au BATCH (pas de zone IDENT).

2.5. MISE EN OEUVRE SOUS VISUALAGE PACBASE

MISE EN OEUVRE SOUS VISUALAGE PACBASE

La clause DECLARE CURSOR doit figurer en WORKING STORAGE SECTION. Elle sera donc insérée complètement sur des lignes "-W" du Programme ou de l'Ecran, à un emplacement choisi par l'utilisateur. Dans les lignes -W, il convient d'écrire la chaîne 'EXEC PAF' en colonne 5 et la chaîne 'END-EXEC' après la phrase de déclaration du curseur.

Exemple

NLG S NIVEAU	DESCRIPTION
100	EXEC PAF DECLARE CU01 CURSOR FOR
110	SELECT * FROM AMEDEF
120	WHERE NDOC = 5 END-EXEC

Exceptée la clause DECLARE CURSOR, tous les ordres SQL s'insèrent dans le programme utilisateur par le langage structuré. Un opérateur spécifique (EXP) permet, sur ces lignes, de générer les appels EXEC PAF et END-EXEC, qui encadrent les ordres SQL-PAF.

Exemple

OPE	OPERANDE
EXP	OPEN CU01

génère :

```
EXEC PAF
OPEN CU01
END-EXEC
```

2.6. LE PROGRAMME UTILISATEUR "TRANSLATE"

LE PROGRAMME UTILISATEUR "TRANSLATE"

En amont du compilateur COBOL, le traducteur DAF traduit les requêtes SQL-PAF en déclarations ou instructions COBOL. (Attention : le traducteur DAF est incompatible avec une demande de formatage COBOL.) Les séquences EXEC PAF ... END-EXEC sont reportées dans le programme COBOL sous forme de commentaires.

WORKING STORAGE SECTION

La phrase 'DECLARE <code-curseur> CURSOR FOR SELECT * FROM <code-table>' génère, en WORKING STORAGE SECTION du programme utilisateur, sous le niveau 01 <code-curseur>-CURSOR, les déclarations de données suivantes :

- la zone de gestion du curseur,
- la zone de spécifications, où la requête est traduite,
- la zone de communication, description de la table DAF sélectionnée.

Dans cette génération, les zones accessibles dans le programme utilisateur (gestion du curseur et communication) sont préfixées par le code du curseur. La zone de spécifications est générée sous forme d'un FILLER.

Exemple

Extraction de descriptions d'Amélioration (table AMEDST), pour l'Amélioration numéro 023456 et de code langue différent de F. Soit le curseur de code AM04 :

```
EXEC PAF  DECLARE AM04 CURSOR FOR SELECT * FROM  
          AMEDST WHERE CAME = 23456 AND CLAN <> 'F'  
END EXEC
```

Les zones suivantes sont générées sous le niveau :

01 AM04-CURSOR.

ZONE DE GESTION DU CURSEUR

En batch :

```
05 AM04-SAVE .
10 FILLER PIC X(06) VALUE 'AM0401' .
10 AM04-CODTAB PIC X(10) VALUE 'AMEDST ' .
10 AM04-CODRET PIC 9(00002) .
10 AM04-ORDRE PIC X(00001) .
10 AM04-FI PIC X(00001) .
10 AM04-FT PIC X(00001) .
10 AM04-CUTICU PIC X(00008) .
10 AM04-CPASCU PIC X(00008) .
10 AM04-CPROCU PIC X(00003) .
10 AM04-CFILCU PIC X(00005) .
10 AM04-NENRCU PIC 9(00006) .
10 AM04-INTERN PIC X(00034) .
```

CODTAB Code de la table.

CODRET Code retour de l'extracteur.

Si aucune erreur n'a été décelée, le code retour est à zéro.

Se reporter au chapitre "Messages d'erreurs", sous-chapitre "Extracteur", pour une description exhaustive de ces codes retour.

ORDRE Chaque ordre DAF est repéré par un numéro :

```
1 INIT
2 CONNECT
4 OPEN
6 FETCH
8 CLOSE
9 QUIT
```

FI Fin de lecture du curseur :
1 lecture du dernier enregistrement du curseur
0 sinon

FT Fin de traitement du curseur :
1 FETCH d'un curseur au-delà de son dernier enregistrement
0 sinon

NOTE : FI est positionné à '1' dans deux cas :
. L'ordre FETCH retourne le dernier enregistrement correspondant à la
sélection du curseur,
. Aucune donnée n'est sélectionnée par l'ordre OPEN.

Si on exécute un ordre FETCH quand FI = '1',
l'indicateur FT est retourné à '1' sans nouvel
enregistrement exploitable.
L'ordre OPEN peut donc lui aussi renseigner
l'indicateur FI pour éviter un FETCH inutile.

Les zones suivantes sont les paramètres utilisateur de l'ordre
CONNECT ; leur alimentation est donc générée par le traducteur, en
PROCEDURE DIVISION, à la rencontre de cet ordre.

CUTICU Code utilisateur DSMS.

CPASCU Mot de passe DSMS.

CPROCU Code du Produit pour contrôle à la connexion.

CFILCU Code de la Filiale pour contrôle à la connexion.

NENRCU Nombre maximum d'enregistrements du fichier intermédiaire.

INTERN Zone technologique interne.

NOTE: La zone de gestion du curseur est intitulée <code-curseur>-SAVE car, en TP,
c'est la zone qu'il convient de sauvegarder lorsque le programme appelant rend
la main au moniteur. L'extracteur TP sauvegarde en effet une représentation de
la requête (zone <code-curseur>-TECH, voir ci-dessous) dans le fichier de
travail.

En TP :

La zone de gestion est la même qu'en batch, mais avec deux zones supplémentaires positionnées devant.

Elles permettent d'identifier la base et le terminal (voir la description de l'ordre CONNECT) :

10	AM04-IDENT	PIC X(25).
10	AM04-BASE	PIC X(4).

Dans tous les cas, il est impératif de ne pas modifier le contenu des zones -SAVE et -TECH (voir ci-dessous).

ZONE DE SPECIFICATIONS

Déclarée sous le niveau : 05 AM04-TECH, cette zone est un FILLER de longueur variable, où la requête est traduite pour l'extracteur. L'utilisateur n'y a pas accès.

ZONE DE COMMUNICATION

05	AM04.	
10	AM04-CPRO	PIC X(00003).
10	AM04-CAME	PIC 9(00006).
10	AM04-CLAN	PIC X(00001).
10	AM04-CLIX	PIC X(00003).
10	AM04-DTEX	PIC X(00030).

PROCEDURE DIVISION

En PROCEDURE DIVISION, chaque ordre SQL est remplacé par :

- L'alimentation de la zone AM04-ORDRE,
- L'appel au sous-programme d'extraction, auquel est transmis toute la zone 01 AM04-CURSOR.

A la rencontre d'un ordre CONNECT, le traducteur génère de plus l'alimentation des zones de gestion du curseur, qui décrivent les paramètres utilisateur.

A la rencontre d'un ordre OPEN, le traducteur génère également l'alimentation des zones techniques du curseur (-TECH) lorsque ce dernier dépend d'une ou de plusieurs zones COBOL. Ces alimentations sont dans certains cas conditionnées.

2.7. IMBRICATION DES CURSEURS DAF

IMBRICATION DES CURSEURS DAF

LIMITATION DE LA SYNTAXE PAF-SQL

La syntaxe PAF-SQL ne retient qu'un sous-ensemble restreint du langage SQL. En particulier, il n'est pas possible de définir de curseurs avec des clauses SELECT imbriquées. L'imbrication des clauses SELECT est cependant utile pour obtenir des informations conditionnées par des séquences de chaînages par exemple (liste des Phases appelées dans les Améliorations reliées aux événements ...).

Cependant, cette limitation n'est pas réellement contraignante dès lors que DAF autorise la déclaration de plusieurs curseurs (jusqu'à 100), ces curseurs pouvant dépendre de zones COBOL. En effet, si un curseur dépend d'une zone COBOL faisant partie de la zone de communication d'un autre curseur, les deux curseurs équivalent à un curseur utilisant des clauses SELECT imbriquées.

IMBRICATION DES CURSEURS

Considérons les tables TVEDEF (table des versions) et AMEQUA (versions qualifiées pour les améliorations) Considérons d'autre part les colonnes CLTC (Code lot technique) et CAME (Code Amélioration). Pour obtenir la liste des Améliorations qualifiées pour tous les lots techniques de la base, la clause SELECT de la requête SQL standard s'écrit :

```
SELECT * FROM AMEQUA WHERE  
CLTC = (SELECT CLTC FROM TVEDEF)
```

Il est possible de simuler cette imbrication en SQL-PAF par le jeu de 2 curseurs :

```
DECLARE TVE CURSOR FOR SELECT * FROM TVEDEF WHERE CRLE  
= SPACE (CRLE représentant la zone "RELEASE" de la version)
```

```
DECLARE AMLT CURSOR FOR SELECT * FROM AMEQUA WHERE  
CLTC = TVE-CLTC
```

On constate que le premier curseur permet d'obtenir la liste de tous les lots techniques. Le second permet d'obtenir toutes les Améliorations appelant le LTC dont le code est TVE-CLTC, c'est-à-dire le code du LTC actuellement fetchée dans le premier curseur.

En PROCEDURE DIVISION, après connexion des deux curseurs, on ouvre le curseur TVE. Puis, à chaque FETCH sur TVE et tant que TVE-FT n'est pas à 1 (curseur non terminé), on ouvre le curseur AMLT. Cette ouverture permet la

prise en compte du nouveau code LTC dans la zone TVE-CLTC (voir le sous-chapitre "Le programme utilisateur traduit"). Le curseur AMLT est alors fetché jusqu'à son terme puis fermé ; ensuite, on retourne au fetching du curseur TVE.

Ainsi, on simule par imbrication des traitements de curseurs un curseur unique défini par des clauses SELECT imbriquées. Notons que ce mécanisme peut se généraliser à une profondeur quelconque d'imbrications dans la limite de 100 curseurs au maximum pour un programme utilisateur.

2.8. EXECUTION DU PROGRAMME UTILISATEUR

EXECUTION DU PROGRAMME UTILISATEUR

BATCH

En batch, le fichier intermédiaire de stockage est défini pour un job. Il est donc nécessaire, en début de chaîne, de créer le fichier (et éventuellement de détruire le précédent). Ce fichier est séquentiel indexé avec une clé de longueur 12 et une taille maximum d'enregistrement égale à 416, la taille moyenne étant 116.

Ensuite, pour exécuter son programme batch l'utilisateur doit, dans son JCL, déclarer les fichiers nécessaires à l'exécution de l'extracteur (fichiers DSMS et fichier intermédiaire de stockage). Pour plus d'informations sur les fichiers utilisés par l'extracteur, se référer aux exemples de JCL fournis dans les manuels d'exploitation.

TP

En TP, le fichier intermédiaire de stockage est un séquentiel indexé avec une clé de longueur 12 commençant en position 2 et une taille maximum d'enregistrement égale à 450, la taille moyenne étant 150.

VisualAge Pacbase - Manuel de Référence	PAGE	41
DSMS ACCESS FACILITY		
EXEMPLES DE PROGRAMME UTILISANT DAF		3

3. EXEMPLES DE PROGRAMME UTILISANT DAF

3.1. INTRODUCTION

DEUX EXEMPLES D'UTILISATION DE DAF

Ce chapitre propose deux exemples d'utilisation de DAF. permettant de présenter la mise en oeuvre de DAF en BATCH et en TP et de proposer des utilisations types de DAF (contrôle de qualité de normes, administration de la base, ...).

Le premier programme, DAFEX1, fonctionne en mode BATCH. Il dresse la liste de toutes les Améliorations de type TY qui n'ont pas d'événement relié déjà fermé. Il nécessite la déclaration de deux curseurs : un pour la liste des Améliorations de type TY et un pour les événements reliés. Ce programme constitue donc un bon exemple d'utilisation de curseurs imbriqués.

Le second programme, DAFEX2, est un dialogue dressant la liste des Evénements ne se conformant pas à un standard local (phases obligatoires, Référence externe sur la base XXOF et charge de travail précisée). Ce programme utilise un seul curseur et gère les défilements d'écrans. Il illustre le mécanisme de transmission du contexte DAF entre chaque itération du dialogue.

3.2. EXEMPLE BATCH

EXEMPLE BATCH

Objectif :

- Dresser la liste des Améliorations de type 'TY' dont les événements associés sont fermés.

Déclarations DAF :

- CU01 sélectionne les Améliorations de type 'TY'
- CU02, ouvert pour chaque Amélioration trouvée par CU01, la liste des événements associés, fermés.

Logique procédurale :

- F02BA : Initialisation de DAF.
- F02CA : Connexion du curseur CU01. Le code utilisateur et le mot de passe sont ici "en dur" mais pourraient être obtenus par lecture d'un fichier en entrée.
- F02DA : Connexion du curseur CU02.
- F21BA : Ouverture du curseur CU01. Cet ordre se traduit par la lecture des références croisées sur le type 'ty' et des Améliorations du produit 'PAC' et par l'écriture de ces chaînages dans le fichier de travail DAF.
- F21CA : Ces fiches de définitions sont appelées, c'est-à-dire relues une à une à partir du fichier de travail DAF (ordre FETCH).
- F21DA : Tant que la fin du curseur CU01 n'est pas atteinte (CU01-FI = '0'), le curseur CU02 est ouvert pour chaque Amélioration lue. Ce curseur sélectionne uniquement les événements fermés liés à cette amélioration. Par conséquent, une fin de curseur immédiate (CU02-FI = '1') dès le premier ordre FETCH signifie que cette amélioration n'a pas d'événement fermé relié. Lorsque cette condition se produit, une ligne est formatée et imprimée sur un Etat. Le curseur CU02 est finalement fermé pour permettre sa réouverture avec la Rubrique suivante du curseur CU01.
- F79 : Lorsque toutes les améliorations ont été lues par l'ordre FETCH, le curseur CU01 est fermé et un ordre QUIT est exécuté afin de fermer les fichiers de la base et le fichier de travail DAF.

EXEMPLES DE PROGRAMME UTILISANT DAF
EXEMPLE BATCH

3
2

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! FICHE DU PROGRAMME           DAFEX1                                     !
!                               !                                       !
! NOM DU PROGRAMME.....: AM."TY" LIEES AUCUN EV. OUVERT           !
!                               !                                       !
! CODE CLASSEMENT DU PROGRAMME.....: DAFEX1                         !
!                               !                                       !
! VARIANTE DU LANGAGE A GENERER.....: 0                             !
! OPTION NUMEROTATION CADRAGE COBOL..:                               !
! OPTION CARTES AVANT PROGRAMME.....:                               !
! OPTION CARTES APRES PROGRAMME.....:                               !
! CODE DU PROGRAMME GENERE.....: DAFEX1                             !
! TYPE DE PROGRAMMATION.....: P                                     !
! NATURE DU PROGRAMME.....: B                                       !
! TYPE DE L'ENTITE.....: P   PROGRAMME                             !
! CONTROLE DE PRESENCE ZONE NUMERIQUE:                               !
!                               !                                       !
!                               !                                       !
! MOTS CLES ASSOCIES.:                                             !
!                               !                                       !
! NO DE SESSION.....: 2013           BIBLIOTHEQUE : CIV   BLOCAGE :   !
!                               !                                       !
! O: C1 CH: PDAFEX1           ACTION:                               !
-----

```

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! STRUCTURES DE DONNEES DU PROGRAMME  DAFEX1 AM."TY" LIEES AUCUN EV. OUVERT !
!                                     30                                     !
! A PR SU : SD EXTERN OAMOU BLOC T R S U RE SE M UNIT P SELECTION F O D N EM!
!   DP   : DP DAFETA SSFOU   0 R   I   D   I   1   !
!       :   Z.COMPL.:           CLE ACC.:           IDENT. :           !
!       :           :           :           :           :           !
!       :   Z.COMPL.:           CLE ACC.:           IDENT. :           !
!       :           :           :           :           :           !
!       :   Z.COMPL.:           CLE ACC.:           IDENT. :           !
!       :           :           :           :           :           !
!       :   Z.COMPL.:           CLE ACC.:           IDENT. :           !
!       :           :           :           :           :           !
!       :   Z.COMPL.:           CLE ACC.:           IDENT. :           !
!       :           :           :           :           :           !
!       :   Z.COMPL.:           CLE ACC.:           IDENT. :           !
!       :           :           :           :           :           !
!       :   Z.COMPL.:           CLE ACC.:           IDENT. :           !
!       :           :           :           :           :           !
! O: C1 CH: -CD
-----

```

EXEMPLES DE PROGRAMME UTILISANT DAF
EXEMPLE BATCH

3
2

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! ZONES DE TRAVAIL DU PROGRAMME   P DAFEX1 AM."TY" LIEES AUCUN EV. OUVERT
!
! DEBUT DU NUMERO DE LIGNE : BA
! A NLG S NIVEAU                DESCRIPTION                TABLE
! 100 * LISTE DES AMELIORATIONS DE TYPE "TY"
! 110   EXEC PAF DECLARE CU01 CURSOR FOR
! 120   SELECT * FROM AMEXTYP WHERE
! 130   CTYP = 'TY'
! 140   END-EXEC
! 200 * LISTE EVENEMENTS FERMES LIES A L'AMELIORATION
! 210   EXEC PAF DECLARE CU02 CURSOR FOR
! 220   SELECT * FROM EVTNAME WHERE
! 230   CAME = CU01-CAME AND
! 240   OFER = 'F'
! 250   END-EXEC
!
!
!
!
! O: C1 CH: -W
!
-----

```

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! TRAITEMENTS PROGRAMME P DAFEX1 AM."TY" LIEES AUCUN EV. OUVERT FONCTION: 02
!
! A SS NLG OPE OPERANDE                NVTY CONDITION
!   N   INITIALISATION ET CONNEXIONS   05BL
! -----
! BA   N   INITIALISATION                10BL
! BA 100 EXP INIT
! -----
! CA   N   CONNEXION DE CU01            10BL
! CA 100 EXP CONNECT CU01 TO
! CA 110   USER   = 'USER'
! CA 120   PASS   = 'PASS'
! CA 130   PRODUIT = 'PAC'
! CA 140   FILIALE = 'FI'
! CA 160   SIZE   = 40
! CA 220 MES 'ERREUR CONNECT CU01 '      99IT CU01-CODRET NOT = ZERO
! CA 230                                CU01-CODRET
! CA 240 GFT
! -----
! DA   N   CONNEXION DE CU02            10BL
! DA 100 EXP CONNECT CU02 TO
!
! O: C1 CH: -P02
!
-----

```


EXEMPLES DE PROGRAMME UTILISANT DAF
EXEMPLE BATCH

3
2

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! FICHE DE L'ETAT          DPD                                     !
!                                     !
! NOM.....: AM."TY" LIEES AUCUN EV. OUVERT                       !
!                                     !
! COMMENTAIRE.....:                                             !
!                                     !
! NATURE.....: E ETAT                                           !
! TYPE DE L'ETAT.....: L                                         !
!                                     !
! LONGUEUR DU LIBELLE...: 132                                     !
! FORMAT TOTALISATION   : PARTIE ENTIERE : 11                   !
!                       : PARTIE DECIMALE: 07                   !
!                                     !
!                                     !
!                                     !
! MOTS CLES ASSOCIES.:                                         !
!                                     !
!                                     !
! NO DE SESSION.....: 2013          BIBLIOTHEQUE : CIV          BLOCAGE : !
!                                     !
! O: C1 CH: Rdpd                ACTION:                          !
-----

```

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! DESCRIPTION DE L'ETAT   PRF PROPRIETES SS NOM RELATIONNEL     LONGUEUR= 132!
!                                     !
! A NL LI S              1   1   2   2   3   3   4   4   5   5   6   6 !
! 1...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5... !
! 00 1                                                           !
! 03 2   LISTE DES AMELIORATIONS SANS EVENEMENTS FERMES RELIES !
! 06 3   -----                                               !
! 09 4   -----                                               !
! 12 5   I  PRODUIT I  NUMERO I  F/O I                           !
! 15 4   -----                                               !
! 18 6   I           I           I           I                   !
! 21 4   -----                                               !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
! O: C1 CH: -L                                                 !
-----

```


3.3. EXEMPLE TP

EXEMPLE TP

Objectif :

- Dresser la liste des Améliorations ne se conformant pas à un standard local.

Déclarations DAF :

- CU01 sélectionne les fiches de définition des Améliorations dont la première phase, la base de la référence externe ou la charge de travail diffèrent de ceux entrés par l'utilisateur du programme.

Logique procédurale :

- F02 : Initialisation de DAF uniquement au premier passage (EIBCALEN = 0).
- F06CA : Connexion du curseur CU01. Le code utilisateur et le mot de passe sont ici "en dur" mais pourraient être entrés dans un menu et passés via la COMMAREA. Il faut noter que le code de la base DSMS doit être spécifié (YY0F). De plus, le code du terminal (EIBTRMID) est affecté à l'identifiant DAF pour garantir l'unicité des clés entre conversations dans le fichier DAF. Enfin notons que le paramètre SIZE est fixé à 10, ce qui correspond au nombre de lignes répétitives de l'Ecran. Ainsi on ne lit que le nombre d'Améliorations nécessaires à un affichage afin d'éviter des lectures trop longues pour le TP et inutiles si l'utilisateur quitte le dialogue sans consulter toute la liste.

Dans le cas de ce programme, la fonction F06 est exécutée uniquement au premier passage. Ainsi l'ordre CONNECT et l'ordre OPEN ne sont exécutés qu'une fois. Ce programme ne prend donc pas en compte des modifications interactives des critères de sélection.

- F06DA : Ouverture du curseur CU01. Cet ordre provoque la lecture des fiches de définition des Améliorations non conformes au standard spécifié dans l'en-tête et le stockage de ces définitions dans le fichier de travail DAF.

- F06EA : La zone de gestion à sauvegarder du curseur CU01 (CU01-SAVE) est transférée dans la COMMAREA par la zone de sauvegarde.
- F4031 : Fermeture du curseur CU01 et terminaison de DAF (CLOSE et QUIT).
- F52BA : Restauration de la zone CU01-SAVE à partir de la COMMAREA. Si tous les enregistrements du curseur ont été lus (ordre FETCH), fermeture du curseur.
- F60PF : Si le dernier enregistrement du curseur n'a pas encore été lu et tant qu'on traite la catégorie répétitive de l'Ecran, on lit l'enregistrement suivant.
- F60PQ : Si le dernier enregistrement a été lu on écrit le message "FIN" et on sort de l'itération.
- F75 : S'il n'y a pas eu d'erreur, on sauvegarde le curseur en COMMAREA.

EXEMPLES DE PROGRAMME UTILISANT DAF
EXEMPLE TP

3
3

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! DEFINITION DU DIALOGUE .....: DF                                     !
!                                     !                                     !
! NOM DU DIALOGUE .....: REQUETES DAF                                 !
!                                     !                                     !
! TYPE DE L'ECRAN.....:          ECRAN STANDARD                       !
! TAILLE DE L'ECRAN (LIGNES,COLONNES): 24    080                     !
! PRESENTATION, TABULATION, INITIAL. : L     01                       !
! APPEL DE DOC. ECRAN, RUBRIQUE .....: 04    05                       !
!                                     !                                     !
!                                     LIBELLE AFFICH. SAISIE L.ERREUR Z.ERR!
! ATTRIBUT D'INTENSITE .....:      N     N     N     N     B !
! ATTRIBUT DE PRESENTATION .....:      N     N     N     R     B !
! ATTRIBUT DE COULEUR .....:      W     W     W     P     W !
!                                     !                                     !
! VARIANTES .....:      0  0     IBM OS CICS (PROG. ET MAP BMS!
! CARTES AVANT, CARTES APRES .....: CC     (PROGRAMME)  KK     (MAP) !
! NOMS EXTERNES .....:          (PROGRAMME)          (MAP) !
! TRANSACTION .....:                                     !
!                                     !
! MOTS CLES ASSOCIES.: PAF                                           !
! NO DE SESSION.....: 6080 T      BIBLIOTHEQUE : T01      BLOCAGE :   !
!                                     !
! O: C1 CH: Odf0010          ACTION:                                   !
-----

```

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! DEFINITION DE L'ECRAN .....: DF0010                                 !
!                                     !                                     !
! NOM DE L'ECRAN .....: Améliorations non standard                   !
!                                     !                                     !
! TYPE DE L'ECRAN.....:          ECRAN STANDARD                       !
! TAILLE DE L'ECRAN (LIGNES,COLONNES): 24    080                     !
! PRESENTATION, TABULATION, INITIAL. : L     01                       !
! APPEL DE DOC. ECRAN, RUBRIQUE .....: 04    05                       !
!                                     !                                     !
!                                     LIBELLE AFFICH. SAISIE L.ERREUR Z.ERR!
! ATTRIBUT D'INTENSITE .....:      N     N     N     N     B !
! ATTRIBUT DE PRESENTATION .....:      N     N     N     R     B !
! ATTRIBUT DE COULEUR .....:      W     W     W     P     W !
!                                     !                                     !
! VARIANTES .....:      * X  0     IBM VS2 CICS (PROG.ET MAP.BMS!
! CARTES AVANT, CARTES APRES .....: * C9     (PROGRAMME)  KK     (MAP) !
! NOMS EXTERNES .....:      D1PDAF01(PROGRAMME)  D1MDAF1  (MAP) !
! TRANSACTION .....:      * DAF1                                     !
!                                     !
! MOTS CLES ASSOCIES.:                                               !
! NO DE SESSION.....: 6080 T      BIBLIOTHEQUE : T01      BLOCAGE :   !
!                                     !
! O: C1 CH: Odf0010          ACTION:                                   !
-----

```

EXEMPLES DE PROGRAMME UTILISANT DAF
EXEMPLE TP

3
3

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! LISTE DES ECRANS UTILISANT L'ECRAN          DF0010          !
!                                     !
! ECRAN   : NLG  RUBRIQ  T LG COL N P C RH RV  C T O SEG RUB.  W SEG RUB.  NV!
!-----
!----- APPEL DE TYPE T (TITRE) -----
! DF0010   020 DF0010  A 01 025 T      00 00          !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
! O: C1 CH: -XO
-----

```

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! DESCRIPTION DE L'ECRAN DF0010 AMELIORATIONS NON STANDARD  !
!                                     !
! A NLG : RUBRIQ . ATTRIBUTS PHYSIQUES . CONTROLE MAJ . AFFICHAGE !
!       :       . T LG COL N P C RH RV . P T U SEG RUB. . W SEG RUB. NV !
!-----
! 010 : PFKEY . V . O V A . EN !
! 011 : . . . V S . 01 !
! 012 : . . . G . 02 !
! 020 : DF0010 . A 01 025 T . . !
! 025 : PAFVUE . 02 025 V . . !
! 040 : . . 04 015 L . . !
! 045 : . . 001 L . . !
! 060 : CPH1 . 02 001 V . R !
! 100 : CBASX . 01 001 V . P !
! 130 : . . 01 001 L . . !
! 135 : . . 01 001 L . . !
! 140 : RGROUP . 01 001 R 3 10 . . !
! 160 : CAME . 003 P . CU01 !
! 180 : CPH1 . 003 P . CU01 !
! 200 : CPH2 . 003 P . CU01 !
! 210 : CBASX . 003 P . CU01 !
! O: C1 CH: -CE
-----

```


EXEMPLES DE PROGRAMME UTILISANT DAF
EXEMPLE TP

3
3

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! TRAITEMENTS ECRAN      O DF0010 AMELIORATIONS NON STANDARD      FONCTION: 02  !
!                                     !
! A SS NLG OPE OPERANDE                                     NVTY CONDITION  !
!           N   INITIALISATIONS PAF                       05IT EIBCALEN = 0  !
!           100 EXP INIT                                     !
!           110 M   '0'      DF00-PFFRST                   !
! -----
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
! O: C1 CH: -P02
! -----

```

```

-----
!                                     *PTJE.PDEV.T01.6080T!
! TRAITEMENTS ECRAN      O DF0010 AMELIORATIONS NON STANDARD      FONCTION: 06  !
!                                     !
! A SS NLG OPE OPERANDE                                     NVTY CONDITION  !
!           N   CONNEXION PAF                             05IT DF00-PFFRST = '0'  !
! -----
!   BA      N   MODIF. INDICATEUR PREM. PASSAGE 10BL      !
!   BA 100 M   '1'      DF00-PFFRST                   !
! -----
!   CA      N   CONNEXION DU CURSEUR                10BL      !
!   CA 100 EXP CONNECT CU01 TO                       !
!   CA 110      USER      = 'USER'                   !
!   CA 120      PASS      = 'PASS'                   !
!   CA 130      PRODUIT   = 'PAC'                    !
!   CA 140      FILIALE   = 'FI'                     !
!   CA 160      SIZE     = 10                         !
!   CA 170      BASE     = 'D474'                   !
!   CA 180      IDENT    = EIBTRMID                  !
! -----
!   DA      N   OUVERTURE DU CURSEUR                10BL      !
!   DA 100 EXP OPEN CU01                             !
! -----
!                                     !
! O: C1 CH: -P06
! -----

```


VisualAge Pacbase - Manuel de Référence
DSMS ACCESS FACILITY
MISE EN OEUVRE DE DAF PAR MATERIEL

PAGE 61

4

4. MISE EN OEUVRE DE DAF PAR MATERIEL

4.1. VERSION MVS/CICS

MISE EN OEUVRE DU MODULE DAF POUR LA VERSION MVS/CICS

Ce système ne convient que pour la variante COBOL II (A-mode 31).

En TP, il est recommandé d'utiliser la zone EIBTRMID pour identifier les utilisateurs dans les ordres CONNECT (paramètre IDENT).

De plus, et toujours en TP, il est possible d'accéder à plusieurs bases depuis le même programme DAF, dans le cas où plusieurs bases DSMS coexistent dans le même CICS. Le code d'appel de la base (4 caractères) alimentant le paramètre BASE de l'ordre CONNECT permet de sélectionner la base que l'on désire interroger.

Ainsi, un ordre CONNECT typique sous CICS sera :

```
EXEC PAF CONNECT C001 TO
  USER      = ZC00-CUSR
  PASS      = ZC00-CPSW
  PRODUCT   = ZC00-CPRO
  SUBSIDIARY= ZC00-CFIL
  SIZE      = ZC00-LSCR
  IDENT     = EIBTRMID
  BASE      = ZC00-CBAS
```

4.2. VERSION CICS/DOS/VS

MISE EN OEUVRE DU MODULE DAF POUR LA VERSION CICS/VSE

La mise en oeuvre du module DAF sur CICS/VSE est en cours de conversion et sera disponible dans une version ultérieure.

4.3. VERSION IMS/VS/ESA

MISE EN OEUVRE DU MODULE DAF POUR LA VERSION IMS/VS/ESA :

Tout programme DAF IMS (batch ou TP) doit toujours appeler l'extracteur TP.
(La version IMS ne comporte pas d'extracteur batch; c'est l'extracteur TP qui est utilisé en batch et en TP.)

- 'PDTPDF' --> extracteur toutes entités.

L'utilisateur doit coder systématiquement l'ordre 'SET' dans la WORKING de ses programmes batch (et TP si ceux-ci sont développés à l'aide du Langage Batch).

Cet ordre doit-être défini de la manière suivante :

```
EXEC PAF SET TYPE = 01  
END-EXEC
```

D'autre part, l'utilisateur doit définir un PSB pour son programme d'application qui doit contenir :

- éventuellement, les PCB des bases utilisateur,
- OBLIGATOIREMENT, les PCB des bases du système DSMS : DA DC DE DX et DF (fichier intermédiaire)

Ces PCB doivent être définis comme suit :

```
.  
. .  
PCB TYPE=DB, DBDNAME=PACDDA$SUF, PROCOPT=GOT, KEYLEN=40  
SENSEG NAME=PAC7DA  
PCB TYPE=DB, DBDNAME=PACDDC$SUF, PROCOPT=GOT, KEYLEN=31  
SENSEG NAME=PAC7DC  
PCB TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=GOT, KEYLEN=17  
SENSEG NAME=PAC7DX  
PCB TYPE=DB, DBDNAME=PACDDX$SUF, PROCOPT=GOT, KEYLEN=50  
SENSEG NAME=PAC7DE  
PCB TYPE=DB, DBDNAME=PACDDF$SUF, PROCOPT=A, KEYLEN=37  
SENSEG NAME=PAC7DF  
.
```

Avec \$SUF = Suffixe des DBDs choisi lors de l'implantation du système DSMS.

Important : Les ordres d'appel à l'extracteur (générés par le préprocesseur) sont de la forme :

```
CALL 'extracteur' USING S-PCB-DA S-PCB-DC S-PCB-DE S-PCB-DX  
                      S-PCB-DF      nom-du-curseur .
```

En conséquence, il faut donner les mêmes noms aux PCBs dans la 'LINKAGE SECTION' et la 'PROCEDURE DIVISION USING' des programmes batch (S-PCB-xx). Pour les écrans dialogue, les fiches des PCBs appelés dans le PSB du dialogue doivent être nommées PCB-xx (le générateur dialogue ajoutera le préfixe 'S-' dans le source généré au niveau de la 'LINKAGE SECTION' et de la 'PROCEDURE DIVISION USING').

Extraction sous le contrôle d'un système de sécurité :

Les extractions d'entités peuvent s'effectuer sous le contrôle d'un système de sécurité (ex : RACF). Dans ce cas, l'extracteur doit pouvoir différencier le type (batch ou TP) du programme demandant les extractions. En effet, la méthode de contrôle du code utilisateur est différente en batch et en TP.

En batch, le code utilisateur (donné dans l'ordre CONNECT) est contrôlé directement par rapport au système de sécurité par l'intermédiaire d'un sous-programme assembleur (PACSECB) interne à l'extracteur (transparent pour l'utilisateur).

En TP, le code utilisateur est contrôlé par rapport à celui donné dans l'IO/PCB par le système de sécurité.

Pour réaliser ce contrôle, l'extracteur doit nécessairement connaître le type du programme qui l'appelle (batch ou TP). Pour ce faire, le paramètre MODE doit être codé dans l'ordre SET :

- MODE = 'TP ' <- le pgm utilisateur est un pgm TP.
- MODE = 'BATCH' <- le pgm utilisateur est un pgm Batch.

Important : Les ordres d'appel à l'extracteur (générés par préprocesseur) dans le cas d'un programme TP sont alors générés de la façon suivante :

```
CALL 'extracteur' USING S-PCB-DA S-PCB-DC S-PCB-DE S-PCB-DX  
                      S-PCB-DF      nom-du-curseur S-IPCB .
```

Pour le batch, les ordres d'appel sont générés de la même façon avec ou sans le contrôle d'un système de sécurité (pas de paramètre S-IPCB dans le CALL).

4.4. VERSION GCOS7/TDS

MISE EN OEUVRE DU MODULE DAF POUR LA VERSION GCOS7

La mise en oeuvre du module DAF sur GCOS7 est en cours de conversion et sera disponible dans une version ultérieure.

4.5. VERSION GCOS8 DMIV-TP/TP8

MISE EN OEUVRE DU MODULE DAF SUR GCOS8 DMIV-TP/TP8

La mise en oeuvre du module DAF sur GCOS8 DMIV-TP/TP8 est en cours de conversion et sera disponible dans une version ultérieure.

4.6. VERSION UNISYS 2200

MISE EN OEUVRE DU MODULE DAF SUR UNISYS 2200

La mise en oeuvre du module DAF sur UNISYS 2200 est en cours de conversion et sera disponible dans une version ultérieure.

4.7. VERSION POUR UNIX, OS2 ET WINDOWS/NT

MISE EN OEUVRE DU MODULE DAF POUR LES VERSIONS

UNIX, WINDOWS/NT ET OS2

Les composants de la mise en oeuvre sont :

- . le pré-processeur DAF ; les extracteurs DAF ; le dictionnaire DAF ; le script de compilation et d'exécution d'un programme.

Il existe un programme qui joue le role de pré-processeur :

- . DAFD10 (voir procédure DPDF dans le manuel d'exploitation DSMS).

Il existe deux sous-programmes d'extraction :

- . PDSBDF (pour les extractions BATCH) et PDSTDF (pour les extractions TP).

Les fichiers compilés des extracteurs sont livrés dans le sous répertoire des exécutable COBOL batch de DSMS.

Les fichiers source COBOL des extracteurs sont livrés dans le sous répertoire de DSMS "dafcgi". Ces sous-programmes doivent être compilés sur le site lorsque la version du compilateur Micro Focus du site est différente de celle utilisée pour DSMS.

LE DICTIONNAIRE DAF

Les entités Rubriques, Structures de Données et Segments pouvant être utilisées pour l'écriture de programmes faisant appel au module DAF, sont fournies sous forme de mouvements batch (fichier "DAFDIC" du sous répertoire "dafcbl" de DSMS).

L'introduction de ce "dictionnaire DAF" dans VisualAge Pacbase par la procédure de mise à jour batch UPDT est sous la responsabilité du gestionnaire de la base.

COMPILATION ET EXECUTION DES PROGRAMMES DAF

Le répertoire dafcgi sous la racine d'installation de DSMS contient en exemple un script de compilation et un script d'exécution d'un programme DAF ("dafcomp" et "dafrun"). Il est conseillé de dupliquer ce répertoire dans un répertoire de travail.

Puis de copier les programmes DAF générés dans le répertoire de travail.
Se positionner ensuite sous ce répertoire pour modifier et lancer les scripts de compilation et d'exécution.

On peut utiliser le script de compilation pour compiler les sous-programmes système DAF.

EXECUTION D'UN EXTRACTEUR DAF

L'exécution de l'extracteur DAF doit être précédée des assignations de fichiers suivantes :

- . Fichiers permanents en entrée :
- Fichier des données DSMS : PAC7DA
- Fichier des éléments VisualAge Pacbase : PAC7DC
- Fichier des références croisées : PAC7DX
- Fichier des libellés d'erreurs : PAC7DE
- Fichier intermédiaire (SYSDAF) : PAC7DF

VisualAge Pacbase - Manuel de Référence
DSMS ACCESS FACILITY
MESSAGES D'ERREUR

PAGE 72

5

5. MESSAGES D'ERREUR

5.1. LE TRANSLATEUR

LE TRANSLATEUR

Le translateur peut détecter un certain nombre d'erreurs dans l'écriture des ordres SQL-PAF. Chaque erreur provoque l'édition d'un message d'erreur dans un état de compte-rendu accompagné du numéro de ligne du début de la séquence SQL-PAF dans le programme traduit.

La liste de ces messages d'erreur est présentée ci-dessous avec, dans certains cas, une explication complémentaire.

Code colonne inconnu : <code-colonne>

Le code <code-colonne> n'est pas une colonne de la table désignée après le FROM (dans la langue choisie).

Trop de conditions élémentaires dans le SELECT.

Le nombre maximum (50) de conditions élémentaires a été dépassé dans la requête.

Code curseur trop long : <code-curseur>

Le code du curseur doit être une chaîne de 4 caractères.

Code curseur déjà déclaré : <code-curseur>

Trop de curseurs déclarés.

Le nombre maximum (100) de curseurs déclarés a été dépassé dans le programme.

Code curseur inconnu : <code-curseur>

Un ordre de manipulation de curseur a été rencontré pour un curseur non déclaré dans le programme.

Aucun ordre CONNECT pour le curseur : <code-curseur>

Aucun ordre OPEN pour le curseur : <code-curseur>

Aucun ordre FETCH pour le curseur : <code-curseur>

Aucun ordre CLOSE pour le curseur : <code-curseur>

Aucun ordre INIT dans le programme : <code-curseur>

Séquence trop longue

Une séquence PAF est un groupe de lignes compris entre les chaînes EXEC PAF (en colonne 12) et END-EXEC. Ce message est envoyé par le translateur lorsque le nombre de lignes dépasse 50.

Fin de programme pendant une séquence PAF.

Opérande numérique à tort : <opérande>

Opérande alphanumérique à tort : <opérande>

Opérande trop long : <opérande>

La longueur maximale d'un opérande est de 120 caractères (pour une constante alphanumérique).

Opérande COBOL incorrect : <opérande>

Types des colonnes différents : <code-col1> <code-col2>

Une condition élémentaire porte sur la comparaison de deux colonnes dont l'une est numérique et l'autre alphanumérique.

Parenthèse ouvrante absente à tort.

La liste des conditions élémentaires succédant au mot-clé WHERE doit comporter un nombre égal de parenthèses ouvrantes et fermantes. Le nombre total de parenthèses ouvrantes doit être égal au nombre de parenthèses fermantes.

Parenthèse fermante absente à tort.

Idem message précédent.

Aucun ordre QUIT dans le programme

Syntaxe invalide : <syntaxe>

La syntaxe précisée dans le message n'appartient pas à la grammaire SQL-PAF.

Littéral trop grand : <littéral>

La taille maximum d'un littéral est de 120 caractères.

Trop de littéraux sur une seule ligne

Le nombre de littéraux sur une ligne d'une séquence PAF ne doit pas dépasser 40.

Fin de littéral incorrect : <littéral>

Code table inconnu : <code-table>

Le code <code-table> n'est pas le code d'une table PAF (dans la langue choisie).

Délimiteur incorrect : <délimiteur>

Le délimiteur précisé dans l'ordre SET doit prendre l'une des deux valeurs : SINGLE ou DOUBLE.

Mode exécution incorrect : <mode-exécution>

Le mode d'exécution précisé dans l'ordre SET doit prendre l'une des deux valeurs : BATCH ou TP.

Type de génération incorrect : <type-génération>

Le type de génération (code alphanumérique sur 2 caractères) précisé dans l'ordre SET doit être une variante VA Pac.

Utilisation de DAF interdite
Vérifier la clé d'accès.

5.2. L'EXTRACTEUR

L'EXTRACTEUR

Le code retour du sous-programme extracteur est une zone, générée par le traducteur, de code <code-curseur>-CODRET.

La valeur 00 du code retour signale qu'aucune erreur n'a été détectée.

Les erreurs possibles sont de trois types :

- . Erreurs dans la séquence des ordres PAF,
- . Erreurs lors des accès fichiers,
- . Erreurs sur les données extraites.

ERREURS DANS LES SEQUENCES DES ORDRES

Codes retour : 01 à 10

Le tableau suivant résume les erreurs possibles dans la séquence des ordres PAF : lorsqu'une séquence est illégale, la case correspondante contient la valeur du code retour. Les ordres en LIGNES (1) précèdent les ordres en COLONNES (2). "NULL" en ligne signifie qu'aucun ordre n'a été rencontré précédemment.

Par exemple, la première ligne illustre qu'aucun ordre ne peut précéder l'ordre INIT.

(2)-----

(1)	!	INIT	!	CONNECT	!	OPEN	!	FETCH	!	CLOSE	!	QUIT	!	
!	NULL	!		!	01	!	01	!	01	!	01	!	01	!
!	INIT	!	02	!		!	03	!	04	!	05	!		!
!	CONNECT	!	02	!		!		!	06	!	07	!		!
!	OPEN	!	02	!		!	08	!		!		!		!
!	FETCH	!	02	!		!	08	!		!		!		!
!	CLOSE	!	02	!		!		!	09	!	10	!		!
!	QUIT	!		!	01	!	01	!	01	!	01	!	01	!

- 01 Initialisations non effectuées,
- 02 Initialisations déjà effectuées,

- 03 OPEN d'un curseur non connecté,
- 04 FETCH d'un curseur non connecté,
- 05 CLOSE d'un curseur non connecté,
- 06 FETCH d'un curseur non ouvert,
- 07 CLOSE d'un curseur non ouvert,
- 08 OPEN d'un curseur non fermé,
- 09 FETCH d'un curseur fermé,
- 10 CLOSE d'un curseur fermé.

ERREURS LORS DES ACCES FICHIERS

Les erreurs détectées lors des accès concernent les fichiers de la base DSMS ainsi que le fichier intermédiaire de stockage.

Codes retour :

- 21 erreur ouverture du fichier des données DA,
- 22 erreur ouverture du fichier des données DC,
- 23 erreur ouverture du fichier des libellés d'erreurs,
- 24 erreur ouverture du fichier intermédiaire,
- 26 erreur ouverture du fichier des références croisées,
- 31 erreur lecture ou écriture du fichier intermédiaire,
- 32 erreur lecture d'un fichier DSMS,
- 40 erreur lors de la connexion à la base DSMS,
- 41 utilisation de DAF non autorisée (clé d'accès).

ERREURS SUR LES DONNEES EXTRAITES

Code retour : 50

Les erreurs sur les données détectées lors de l'extraction sont les erreurs de numéricité des colonnes (donc définies avec un format numérique).