

VisualAge Pacbase



Les Procédures du Développeur Windows 2000 ou NT Server

Version 3.5



VisualAge Pacbase



Les Procédures du Développeur Windows 2000 ou NT Server

Version 3.5

Note

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Notices», à la page vii.

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

<http://www.ibm.com/support/docview.wss?rs=37&context=SSEP67&uid=swg27005478>

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

Première édition (Novembre 2011)

La présente édition s'applique à :

- VisualAge Pacbase Version 3.5

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.ibm.com/software/awdtools/vapacbase/support.html> ou en nous adressant un courrier à :

IBM Paris Laboratory
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand, France.

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

© Copyright IBM Corporation 1983,2011.

Table des matières

Notices	vii	GPRC - Gestion de l' API COBOL	45
Marques	ix	GPRC - Présentation générale	45
Chapitre 1. Introduction à l'utilisation des procédures batch	1	GPRC - Entrées Utilisateur	45
Présentation du manuel	1	GPRC - Description des étapes	45
Présentation des procédures	1	GPRC - Script d'exécution	47
Identification de l'utilisateur	2	GPMC - Gestion de l'opérateur MCI	52
Autorisations d'accès	3	GPMC - Présentation générale	52
Anomalies d'exécution	3	GPMC - Entrées Utilisateur	53
Liste des "run-time errors"	5	GPMC - Description des étapes	53
Gestion des erreurs dans les procédures	6	GPMC - Script d'exécution	54
Lancement et Arrêt du serveur	6	Chapitre 3. Extractions	61
Connexion d'un émulateur 3270	6	PACX - Présentation générale	61
Chapitre 2. Editions - Générations	9	PACX - Entrées communes aux extracteurs	61
GPRT - Editions / Générations	9	EXTR/EXTA - Extraction d'entités	63
GPRT - Présentation générale	9	EXTR/EXTA - Présentation générale	63
GPRT - Entrées Utilisateur / Résultats	11	EXTR/EXTA - Entrées Utilisateur	63
GPRT - Commandes d'édition génération	12	EXUE - Extracteur de contenu d'Entités Utilisateur	66
GPRT - Lancement de la procédure	24	EXUE - Présentation générale	66
GPRT - Traitement des flots générés	24	EXUE - Entrées Utilisateur	67
GPRT - Exemple de script de génération (C4)	25	PACX - Description des étapes	67
GPRT - Traitements spécifiques	26	PACX - Script d'exécution	69
GPRT - Description des étapes	28	Chapitre 4. Extraction personnalisée/ Documentation automatisée	73
GPRT - Script d'exécution	29	Introduction	73
GPRT - Fichiers générés	34	Extractions personnalisées - PAF+	73
EMLD - Chargement des libellés d'erreur	36	XPAF - Validation d'un Plan type d'extraction	73
EMLD - Présentation générale	36	XPAF - Présentation générale	73
EMLD - Entrées Utilisateur	36	XPAF - Entrées Utilisateur	74
EMLD - Description des étapes	36	XPAF - Description des étapes	75
EMLD - Script d'exécution	37	XPAF - Script d'exécution	77
EMUP - Mise à jour des libellés d'erreur	39	XPAF - Exploitation d'un plan type d'extraction	79
EMUP - Présentation générale	39	Structuration de documentation - GDP+	80
EMUP - Entrées Utilisateur	39	XPDM - Validation d'un Plan type d'édition	80
EMUP - Description des étapes	40	XPDM - Présentation générale	80
EMUP - Script d'exécution	41	XPDM - Entrées Utilisateur	81
PPAF - Pré-processeur PAF des programmes générés	42	XPDM - Description des étapes	81
PPAF - Présentation générale	42	XPDM - Script d'exécution	83
PPAF - Entrées Utilisateur	43	Plans type d'extraction et d'édition	84
PPAF - Description des étapes	44	PRGS - Edition du fichier des plans type	84
PPAF - Script d'exécution	44		

PRGS - Présentation générale	84	Etat	114
PRGS - Entrées Utilisateur	85	Définition (ligne B)	114
PRGS - Description des étapes	85	Ligne de description des libellés (ligne 4)	115
PRGS - Script d'exécution	85	Caractéristiques édition (ligne 5, type E)	115
Chapitre 5. Mises à jour batch	87	Liste des catégories (ligne 5)	117
UPDP - Mise à jour à partir de tables PAF	87	Description de structures (ligne 6)	120
UPDP - Présentation générale	87	Ecran	121
UPDP - Entrées / Règles de mise à jour / Résultats	87	Définition (ligne H)	121
UPDP - Description des étapes	89	Complément au Dialogue (ligne H3)	124
UPDP - Script d'exécution	92	Description (ligne I)	124
UPDT - Mise à jour	94	Appel de Segments (ligne H2)	127
UPDT - Présentation générale	94	Appel de Macro-Structures (ligne M)	128
UPDT - Entrées / Règles de mise à jour / Résultats	95	Début de programme (ligne D)	128
Demande de Checkpoints	97	Zone de travail et de lien (ligne 7)	129
Entrées communes	98	Langage structuré (ligne P)	130
Lignes anciennement -G (lignes VC, VG, VE, VO)	98	Programme	131
Valorisation du Format-Guide (ligne VZ)	99	Définition (ligne 0)	131
Appels d'instances par Relations (ligne QR)	100	Appel de Structures de Données (ligne 1)	132
Blocage d'entités (ligne R)	101	Appel de Macro-Structures (ligne M)	133
Affectation de mots-clés (ligne G)	102	Début de programme (ligne D)	134
Rubrique	103	Zone de travail et de lien (ligne 7)	134
Définition (ligne C)	103	Langage structuré (ligne P)	136
Description (ligne E)	104	Source COBOL (ligne FC)	137
Entité Méthode	105	COBOL pur (ligne 9)	138
Définition (ligne K1)	105	Bloc Base de Données hiérarchique	138
Appel propriétés dans Entité ou Relat. (ligne K3)	106	Définition (ligne L1)	138
Entrées pour l'entité Relation Méthode	106	Description (ligne L2)	139
Définition (ligne K1)	106	Bloc Base de Données Codasyl	139
Appel d'Entités dans Relation ou CIF (ligne K2)	107	Définition (ligne L1)	139
Appel propriétés dans Entité ou Relation(ligne K3)	107	Description (ligne L3)	140
C.I.F Méthode	108	Bloc Base de Données (relationnel-SQL)	141
Définition (ligne K1)	108	Définition (ligne L1)	141
Appel d'Entités dans Relation ou CIF (ligne K2)	109	Description (ligne L4)	141
Structure de Données	109	Bloc Base de Données Turboimage	143
Définition (ligne A)	109	Définition (ligne L1)	143
Segment	110	Description (ligne L2)	143
Définition (ligne 2)	110	Texte	144
Description (ligne 3)	113	Définition (ligne S)	144
Sous-schémas et sous-systèmes		Description (ligne T)	144
Pactables (ligne 21)	114	Document	146
		Définition (ligne W1)	146
		Description (ligne W2)	147
		Format-guide	147

Définition (ligne V1)	147	INFP - Script d'exécution	175
Description (ligne V2)	148	ISEP - Sélection des points d'entrée	176
Méta-Entité	148	ISEP - Présentation générale	176
Définition (ligne Y1)	148	ISEP - Entrées Utilisateur	177
Libellés des lignes détail (ligne Y6)	149	ISEP - Description des étapes	179
Description (ligne Y2)	149	ISEP - Script d'exécution	180
Relation Utilisateur	150	ISOS - Sélection des chaînes et des	
Définition (ligne Y5)	150	opérateurs	181
Entité Utilisateur Cliente	151	ISOS - Présentation générale	181
Définition (ligne Y3)	151	ISOS - Entrées Utilisateur	183
Description (ligne Y4)	151	ISOS - Description des étapes	184
Entité Utilisateur Extension	152	ISOS - Script d'exécution	186
Définition (ligne YC)	152	IMFH - Fusion de fichiers FH / Création FH	
Description (ligne YD)	152	et FR	188
Thésaurus	153	IMFH - Présentation générale	188
Enrichissement du Thésaurus (ligne		IMFH - Description des étapes	188
G1)	153	IMFH - Script d'exécution	189
Bibliothèque	153	INFQ - Réinitialisation du fichier FQ	190
Définition (ligne X)	153	INFQ - Présentation générale	190
UPDT - Description des étapes	154	INFQ - Description des étapes	190
UPDT - Script d'exécution	156	INFQ - Script d'exécution	191
Chapitre 6. Pactables	159	IGRA - Décomposition zone groupe.	192
GETD-GETA - Générateurs de descriptifs	159	IGRA - Présentation générale	192
GETD-GETA - Présentation générale	159	IGRA - Description des étapes	193
GETD-GETA - Entrées Utilisateur /		IGRA - Script d'exécution	195
Résultat	160	IANA - Analyse d'impact	197
GETD-GETA - Description des étapes	161	IANA - Présentation générale	197
GETD - Script d'exécution	162	IANA - Description des étapes	199
GETA - Script d'exécution	164	IANA - Script d'exécution	201
GETI - Initialisation de descriptifs	165	IPFQ - Edition du fichier FQ	204
GETI - Présentation générale	165	IPFQ - Présentation générale	204
GETI - Entrées Utilisateur	166	IPFQ - Entrées Utilisateur	205
GETI - Description des étapes.	166	IPFQ - Description des étapes.	206
GETI - Script d'exécution	167	IPFQ - Script d'exécution	207
SMTD-RMTD - Migration des descriptifs de		IPEP - Edition des points d'entrée	208
tables	168	IPEP - Présentation générale	208
SMTD - Présentation générale.	168	IPEP - Description des étapes	209
SMTD - Description des étapes	168	IPEP - Script d'exécution	209
SMTD - Script d'exécution	168	IPIA - Edition résultats de l'analyse d'impact	210
RMTD - Présentation générale	170	IPIA - Présentation générale	210
RMTD - Description des étapes	170	IPIA - Entrées Utilisateur	212
RMTD - Script d'exécution.	170	IPIA - Description des étapes	214
		IPIA - Script d'exécution	215
Chapitre 7. Pac/Impact	173	Chapitre 8. Contrôle d'intégrité	
Préambule	173	 méthodologique.	217
INFP - Initialisation du fichier FP	174	ADM - Méthode SSADM	217
INFP - Présentation générale	174	SADM - Présentation générale	217
INFP - Entrées Utilisateur	174	SADM - Entrées Utilisateur	217
INFP - Description des étapes.	175	SADM - Description des étapes	218

SADM - Script d'exécution	218	YSMC - Entrées Utilisateur.	220
YSM - Méthode YSM	220	YSMC - Description des étapes	222
YSMC - Présentation générale.	220	YSMC - Script d'exécution	223

Notices

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 : IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk NY 10504-1785, U.S.A.

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, SMC Department, 1 place J.B.Clément, 93881 Noisy-Le-Grand Cedex, 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, VisualAge Pacbase, RACF, RS/6000, SQL/DS 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.

Chapitre 1. Introduction à l'utilisation des procédures batch

Présentation du manuel

Ce guide a pour objectif de décrire les procédures batch que tout utilisateur du produit peut être amené à utiliser.

Il s'agit d'une part des procédures standard de mise à jour, éditions-génération, et extractions.

D'autre part, sont documentées dans ce volume les procédures concernant les domaines suivants :

- l'extraction personnalisée et la documentation automatique,
- les contrôles d'intégrité sur les occurrences Méthode (liés à l'utilisation du module Pacdesign de la Station de travail pour SSADM et YSM),
- Pac/Impact.

Présentation des procédures

Les traitements batch sont regroupés en procédures. L'objectif des chapitres suivants est de présenter chacune des procédures susceptibles d'être utilisées et d'en préciser les conditions d'exécution.

Pour chaque procédure, on trouvera :

- une présentation générale incluant
 - les conditions d'exécution,
 - les actions à entreprendre en cas d'anomalie d'exécution,
- la description des entrées utilisateur, des traitements et des résultats obtenus, ainsi que les recommandations éventuelles pour l'utilisation,
- la description des étapes.

Pour utiliser une procédure sur une base donnée, un utilisateur devra en avoir l'autorisation.

Chaque utilisateur possède :

- un niveau général de droit aux procédures batch,
- un niveau de droit par base.

Les autorisations utilisateur sont définies dans la Base Administration.

NOTE

La définition et le mode d'exécution d'une procédure sont décrits dans le manuel d'installation au chapitre "Installation de la partie serveur", sous-chapitre "Installation du Système", paragraphe "Un élément du système : la Procédure".

Identification de l'utilisateur

Les procédures batch accédant aux bases requièrent, en tête des entrées utilisateur, une ligne '*', qui permet d'identifier l'utilisateur ainsi que la bibliothèque et la session où il désire travailler.

Certains des renseignements portés sur cette ligne correspondent aux informations indiquées sur l'écran de connexion en début de conversation et permettent donc de contrôler la compatibilité des commandes avec les autorisations données à l'utilisateur.

En effet, avant de lancer une procédure batch il est nécessaire de vérifier qu'on dispose bien du niveau d'autorisation requis.

Pos.	Lon.	Valeur	Signification
2	1	'*	Identifiant de la ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	bbb	Code bibliothèque
22	4	nnnn	Numéro de session (espace si session courante)
26	1		Type de session
		' '	Session courante (si numéro de session à espace)
		'T'	Session historisée de travail
27	1		Pour la procédure UPDT, en cas d'annulation multiple :
		'N'	Edition de tous les mouvements, y compris les mouvements générés (option par défaut)
		'O'	Edition des mouvements introduits par l'utilisateur et des mouvements générés erronés
		'E'	Edition des mouvements erronés uniquement
			Les 2 zones suivantes sont à renseigner pour toute procédure d'extraction génératrice de mouvements de mise à jour destinés à être appliqués à une bibliothèque/session sous contrôle DSMS (vous pouvez aussi les saisir sur la ligne '*' de l'UPDT)
40	3		Code du produit (sur 3 caractères),

Pos.	Lon.	Valeur	Signification
43	6		Numéro de l'amélioration (sur 6 caractères, les zéros non significatifs devant être inscrits)
			Ces deux données figureront dans le Journal une fois l'UPDT exécutée
49	1		Report des blocages d'occurrences :
		blanc	Remplacement du code utilisateur bloquant par celui de la ligne '*'
		'1'	Les nouvelles entités créées à partir des entités extraites ne sont pas verrouillées après l'UPDT
		'2'	Le code utilisateur bloquant reste le même
50	1		Report du mot de passe sur les procédures d'extraction sur la ligne '*' des mouvements en sortie
		blanc	Le mot de passe n'est pas reconduit sur le fichier en sortie
		'1'	Le mot de passe est reconduit (Attention : pour EXTR, la ligne '*' n'est reconduite sur le fichier en sortie que si vous avez saisi 'C' en position 1)
67	1	'N'	Cette valeur est systématiquement positionnée par les extracteurs. Elle indique que les mouvements extraits appartiennent à un environnement cohérent et implique que les mises à jour sont toujours effectuées avec un warning en cas d'anomalie par rapport à un contrôle. La conservation de la cohérence des données entraîne, entre autres, l'inhibition de la conversion minuscules/majuscules ainsi que l'acceptation de formats de rubriques supérieurs à 999 caractères.

Autorisations d'accès

La ligne * avec Code utilisateur et Mot de passe est obligatoire pour chaque procédure.

Le contrôle des autorisations d'accès est géré par l'Administrateur via l'Administrator workbench.

Anomalies d'exécution

Il arrive que des anomalies se produisent lors de l'exécution d'un programme batch. En particulier, les erreurs entrée-sortie sur les fichiers du système ou de la base provoquent une fin anormale avec un code erreur à 12, accompagné d'un message émis sur le fichier .LOG de la procédure.

Dans cette situation, vous devez avant tout rechercher la présence de ce message :

```
PROGR : pppppp INPUT-OUTPUT ERROR : FILE ff OP : oo
STATUS : ss
END OF RUN DUE TO PROVOKED ABEND
```

La présence de ce message est conditionnée par le positionnement, au préalable, de la variable "BVPTrace" à "YES" dans Init.vbs ou dans le script de lancement de la procédure.

Dans la plupart des cas, l'examen du 'status' et le type d'opération effectuée permettent de trouver la cause de la fin anormale.

Quelques valeurs courantes du code opération et du status :

Code	Opération
W	WRITE
RW	REWRITE
RU	READ UPDATE
OP	OPEN
CL	CLOSE
D	DELETE
R	READ
P	START
RN	READ NEXT

Status	Libellé
10	Fin de fichier
21	Erreur de séquence
22	Clé en double
23	Enregistrement non trouvé
24	Fichier trop petit
30	Erreur système
34	Fichier trop petit (séquentiel)
35	Fichier non trouvé
46	Pas d'enregistrement courant (lors du READ). L' erreur survient quand l'opération précédente est un START qui a échoué, laissant le pointeur indéfini.

Status	Libellé
48	Tentative d'écriture sur un fichier non ouvert ou sur un fichier séquentiel ouvert en I/O
92	Erreur logique (par exemple, ouverture fichier déjà ouvert)
93	Fichier resté ouvert en TP
95	Fichier non défini/mal défini

Si le message est absent et que le type de l'erreur concerne directement les programmes du produit, il est nécessaire de contacter l'équipe technique du produit chez IBM et de conserver tous les comptes-rendus utiles à l'analyse du problème.

En cas d'erreur autre qu'une erreur d'entrée-sortie sur un fichier de la base, le message suivant s'affiche :

```
Run Time Error nnn
```

où nnn est le numéro de l'erreur.

Le runtime Error 013 est le plus fréquent, il signale que la procédure n'a pas trouvé un fichier en entrée du programme.

Le sous-chapitre suivant contient la liste des erreurs les plus fréquentes. Chaque runtime Error est accompagné d'un message explicatif relativement succinct.

Si le runtime Error n'est pas dans la liste suivante ou si le message est insuffisant et que le type d'anomalie met en cause directement les programmes du système, il est nécessaire de contacter le Support Technique ("Hot Line") et de conserver tous les listings qui pourront servir à l'analyse du problème.

Liste des "run-time errors"

La liste suivante n'est pas exhaustive, elle décrit les erreurs les plus fréquentes.

Numéro	Signification
-----	-----
004	Nom de fichier incorrect.
005	Désignation d'unité incorrecte.
007	Plus d'espace disque disponible.
009	Répertoire inexistant ou saturé.
013	Fichier non trouvé.
026	Erreur d'entrée-sortie due à un disque corrompu.
027	Unité indisponible.
028	Plus d'espace disque disponible.
033	Erreur physique d'entrée-sortie.

105 Erreur d'allocation mémoire.
116 Impossibilité d'allouer de la mémoire.
135 Fichier non trouvé.
150 Programme interrompu par l'utilisateur.
157 Mémoire insuffisante pour charger le programme.
170 Programme système non trouvé.
173 Programme appelé non trouvé.
188 Nom de fichier trop long.
198 Mémoire insuffisante pour charger le programme.
207 Machine inexistante sur le réseau.
208 Erreur réseau.
209 Erreur réseau.
221 !
222 !> Erreur pendant un tri.
223 !

Gestion des erreurs dans les procédures

A la fin de chaque procédure, la détection d'une erreur provoque l'arrêt de la procédure avec un code retour différent de zéro. Ce code est récupérable dans la variable "Return" tout de suite après la commande de lancement de la procédure.

Ceci permet d'empêcher l'exécution d'une suite si plusieurs procédures sont enchaînées.

Lancement et Arrêt du serveur

On installe le Listener en mode Service NT.

Les Stations et les terminaux peuvent ainsi se connecter à VisualAge Pacbase.

Le raccourci 'Start [nom_base] Database Service' situé sous le Groupe de programmes [VisualAge Pacbase 3.5 Server] du Menu Démarrer, permet de lancer le listener sur la base [nom_base].

Le raccourci 'Stop [nom_base] Database Service' situé sous le Groupe de programmes [VisualAge Pacbase 3.5 Server] du Menu Démarrer, permet d'arrêter le listener sur la base [nom_base].

Les paramètres de fonctionnement du listener sont définis dans la procédure "Server.wsf"

Connexion d'un émulateur 3270

Il est possible de se connecter en mode terminal passif sur un serveur TP à l'aide d'un émulateur 3270.

Pour accéder à une base de donnée, en mode 3270 via un serveur TP, il est nécessaire de configurer l'émulateur en indiquant :

- l'adresse IP de la machine où est implémenté le serveur TP,
- le numéro de port d'écoute du serveur TP, choisi à l'installation lors de la création de la base.

Le code page de l'émulateur doit être valorisé en fonction du code langue de la base :

- code page 1147 pour une base française,
- code page 1146 pour une base anglaise.

Ces codes pages sont positionnés automatiquement, dans la procédure "Server.wsf", lors du lancement du serveur TP.

Chapitre 2. Editions - Générations

GPRT - Editions / Générations

GPRT - Présentation générale

Cette procédure a deux rôles :

- Editer de la documentation à partir de données appartenant à la base,
- Générer les programmes, écrans, composants eBusiness, descriptions de base de données, structures de données, libellés d'erreur.

Cette procédure n'a aucune influence sur la base, ce qui signifie qu'il est possible de l'utiliser pendant que celle-ci est manipulée en conversationnel.

Cependant, si l'utilisateur décide de prendre en compte les demandes d'édition introduites en conversationnel (commande +AG), il faut que les fichiers de la Base de Développement aient été fermés. En effet, cette procédure va invalider les demandes d'édition introduites en conversationnel, ce qui nécessite la disponibilité en mise à jour des fichiers.

Cette procédure fait appel à un programme unique (BVPACB), qui sert de moniteur d'enchaînement des différents programmes de la procédure.

Tous les programmes de la procédure sont donc considérés comme des sous-programmes de ce moniteur, avec lequel ils communiquent par l'intermédiaire d'une zone de communication et de codes retour particuliers. Les demandes de l'utilisateur étant très diverses, cette procédure est décomposée en 'sous-chaînes' qui traitent intégralement la préparation des éditions-générations des familles qu'elles gèrent.

A la suite de deux programmes généraux communs à toutes les chaînes (BVPACA10 et BVPACA20), les différentes sous-chaînes sont activées si nécessaire :

- Blocs base de données
- Blocs base de données SQL
- Programmes COBOL
- Ecrans Dialogue
- Ecrans Client
- Ecrans Serveur
- Libellés d'erreur eBusiness
- Libellés d'erreur et 'Windowing' Dialogue

- Gestionnaire de documentation personnalisée
- Programmes langage batch
- Dictionnaire

Les fichiers contenant le généré à compiler ou à ranger en bibliothèque source ou à assembler, etc. sont concaténés sur un seul fichier physique qui est repris dans l'étape suivante.

Le fichier des 'Libellés d'erreur de l'utilisateur' est mis à jour à partir du fichier suffixé LG pour être récupéré sur le fichier suffixé GL. Ce fichier sert à la mise à jour des libellés d'erreur du client. Il doit être en entrée des procédures EMLD ou EMUP. Par ailleurs, ces éléments sont édités sur le fichier suffixé IL.

La procédure installée ne prévoyant pas de nom pour les deux versions de ce fichier, ces noms doivent être précisés lors de la demande de génération de ces libellés.

L'impression ordinaire de la documentation personnalisée est récupérée sur le fichier suffixé IN. Cette documentation peut également être éditée sur le fichier suffixé GN (longueur d'enregistrement 265) avec le caractère de saut ASA en 1ère position de chaque enregistrement pour permettre d'éventuels traitements particuliers d'édition.

Le fichier contenant les éléments nécessaire au "Windowing" d'applications dialogue est codifié PAC7GT (longueur d'enregistrement 260). Son nom doit être précisé lors de la demande de génération.

RPPz Utilitaires :

En option C9, cette procédure permet de générer les entités avec les micro-patterns.

Elle effectue la génération des micro-patterns pour les lignes -W de type I, E, S, F qui ne proviennent pas de macros.

La génération est effectuée sans numéro de ligne.

La partie droite (colonnes 73 à 80) est générée à blanc sauf sur les lignes étiquettes où l'on a lvnn (avec nn = niveau de la fonction/sous-fonction).

Condition d'exécution

Les fichiers peuvent rester ouverts, sauf si l'utilisateur a demandé la prise en compte des demandes en conversationnel par la commande ' +AG'. Dans ce cas, les fichiers de la Base de Développement doivent avoir été fermés.

Anomalies d'exécution

Se reporter au sous-chapitre "Anomalies d'exécution" du chapitre "Introduction à l'utilisation des procédures batch".

GPRT et le module SCM

Si le module SCM est disponible sur le site, la génération peut donner lieu à des mouvements dans le fichier QJ, journal de SCM de stockage des caractéristiques du généré (CONSTANTES-PACBASE ou PACBASE-CONSTANTS).

Seules les entités définies dans un Environnement SCM et générées à partir d'une session de production ou de la session courante sont traitées pour compléter le fichier QJ.

L'application dans la(les) Base(s) de Développement des mouvements enregistrés dans QJ peut être effectuée automatiquement après la génération, par des options indiquées comme paramètres au niveau de l'étape de génération. Cette opération ne requiert pas la fermeture de cette(ces) Base(s) de Développement.

Les entités générées définies dans les Environnements SCM sont ainsi complétées par des informations concernant le dernier traitement de ces entités. Celles qui sont générées en session courante passent à l'état "en attente de mise en production".

Des erreurs éventuellement détectées sont mémorisées dans le fichier QJ. Elles sont éditées lors de l'archivage des mouvements par la procédure ARPM, et les mouvements erronés sont restaurés dans le fichier QJ pour un nouveau traitement.

GPRT - Entrées Utilisateur / Résultats

Entrées utilisateur

Les entrées nécessaires pour soumettre une commande batch d'édition-génération sont :

- Une ligne obligatoire identifiant l'utilisateur et le contexte d'édition-génération,
- Autant de lignes que de demandes d'édition ou de génération désirées,

- Une ligne (' +AG'), optionnelle, permettant la prise en compte des demandes déjà introduites en conversationnel.

Tous les autres types de mouvements sont ignorés.

Résultats

On obtient deux types de résultats :

- Un compte-rendu des demandes,
- L'ensemble des éditions demandées.

Les éditions sont triées par utilisateur/bibliothèque et sont précédées d'une bannière (page de garde).

Remarque :

Cette procédure n'incrémente pas le numéro de session.

GPRT - Commandes d'édition génération

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
2	1		Code ligne
		'Z'	Valeur par défaut
3	2		Critère majeur d'ordre d'édition
			Permet de choisir l'ordre de listage des éditions qui seront sélectionnées.
5	4		Commandes d'Edition-Génération
			Remarques préalables : à la suite de la valeur de la commande d'édition, on peut parfois trouver un code entre parenthèses ; la signification de chaque code est la suivante :
			(S) : une sélection est possible dans la zone ENTITE
			(A) : la modification des commandes avant et après est possible dans le LIBELLE d'édition
			(T) : la sélection du type s'effectue dans le LIBELLE d'édition

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			(K) : sélection par mot-clé (commande LK_) ; elle doit obligatoirement comporter un ou plusieurs mot(s)-clé(s), critère(s) de la sélection ; il existe trois types de sélection dans la ligne suite : * la saisie d'un seul mot-clé, pour lister toutes les occurrences possédant au moins ce mot-clé, * la saisie de plusieurs mots-clés espacés d'un blanc pour lister les occurrences possédant tous les mots-clés saisis, * la saisie de plusieurs mots-clés reliés par le signe = pour lister les occurrences possédant au moins un des mots-clés saisis. De plus, la zone SEL permet de spécifier le type de mot-clé (L : mot-clé implicite, M : mot-clé explicite, blanc : mot-clé implicite et explicite)
			Thesaurus
		DCK	Description des paramètres du Thésaurus (ces paramètres étant définis en inter-bibliothèque, cette commande est incompatible avec l'option U1 et il faut utiliser l'option C1 ou I1 donnant les mêmes résultats).
		LCK	Liste des mots-clés du thésaurus.
			Textes
		DCT	(S) Descriptif du texte sélectionné ; si l'on saisit '*' dans la zone ENTITE, les descriptifs de tous les textes seront imprimés, triés sur leur code instance.
		DTT	(T) Descriptif des textes par type.
		L*T	Liste des titres des textes et des paragraphes.
		LCT	Liste des textes par code.
		LKT	(K) Liste des textes par mot-clé.
		LNT	Liste des textes par nom.
		LTT	(T) Liste des textes par type.
			Rapport (PDM)
		DCV	(S) Impression du descriptif du Rapport dont le code est spécifié dans la zone entité ; lorsque le code n'est pas renseigné, les descriptifs de tous les Documents sont imprimés, triés par code.
		FLV	(A) Contrôle du flot.
		LCV	Liste des Rapports triés par code.
		LKV	(K) Liste sélective des Rapports suivant le ou les mot(s)-clé(s) renseigné(s) sur la ligne suite.
		LNV	Liste des Rapports triés par nom.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		PCV	(S) Edition du Rapport dont le code est spécifié dans la zone 'Entité'; lorsque le code n'est pas renseigné, les descriptifs de tous les Rapports sont imprimés, triés par code. L'édition locale en format RTF s'obtient en générant le Rapport en option C2 ; l'édition partielle est documentée dans le manuel "Gestionnaire de Documentation Personnalisée" chapitre "Mode d'accès", sous-chapitre "Edition-génération".
			Rubriques et Propriétés
		DCE	(S) Descriptif des Rubriques.
		DFE	(S) Descriptif des rubriques non définies dans le Dictionnaire.
		LAE	Liste des Rubriques triées par nom COBOL.
		LCE	(S) Liste des Rubriques et Propriétés triées par code.
		LKE	(K) Liste des Rubriques et Propriétés par mot-clé.
		LNE	Liste des Rubriques et Propriétés par nom.
		LXE	Liste des Rubriques et Propriétés définies dans le Dictionnaire mais qui ne sont pas utilisées.
			Structures de Données
		DCD	(S) Descriptif des Structures de Données.
		FLD	(A) Contrôle du flot.
		GCD	(S) Génération de data pour la S.D. sélectionnée. Lors de la phase de génération, un Segment pourra contenir jusqu'à 9999 Rubriques. Un message d'anomalie sera édité dans le compte-rendu de génération si cette limite est dépassée. (pour plus de détails concernant la génération, se reporter au chapitre correspondant du "Manuel de Référence Dictionnaire").
			C3 : Génération des commentaires qui seront exploités par VA Pac Connector (outil du module eBusiness).
			C4 : Possibilité de s'affranchir des formats-guides DATA et DATASQ.
		LCD	Liste des Structures de Données par code.
		LED	Liste des libellés d'erreur par Structure de Données et pour chaque Segment ; cette édition ne liste que les libellés ayant déjà été générés.
		LKD	(K) Liste des Structures de Données par mot-clé.
		LND	Liste des Structures de Données par nom.
		LOD	Liste des Structures de Données par nom externe.

POS	NON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		LPD	Liste des Structures de Données par nom externe de Programme.
		LTD	Liste des Structures de Données par type.
			Segments et Vues Logiques
		DCS	(S) Descriptif des Segments dans un format donné. Sélection du format dans le LIBELLE d'édition en conversationnel (le format R permet d'obtenir le nom relationnel des rubriques appelées au lieu de leur libellé).
			Vous devez indiquer la Structure de Données dans la zone ENTITE et les 2 derniers caractères des codes des Segments sur la ligne suite du LIBELLE d'édition (Attention : vous ne pouvez indiquer qu'une ligne suite, soit un total de 25 Segments).
		LCS	Liste des Segments par code.
		LKS	(K) Liste des Segments par mot-clé.
		LNS	Liste des Segments par nom.
			Formats Guides
		DCI	(S) Descriptif des Formats Guides.
		LCI	Liste des Formats Guides par code.
		LKI	(K) Liste des Formats Guides par mot-clé.
		LNI	Liste des Formats Guides par nom.
		LXI	Liste des clés de chaînage.
			Blocs Bases de Données
		DTB	Descriptif des Blocs Bases de Données par type.
		FLB	(A) Contrôle du flot.
		FLS	(A) Contrôle du flot pour Blocs Bases de Données Relationnelles SQL.
		GCB	(A) Génération du Bloc Base de Données.
			Lors de la phase de génération, les Segments appelés dans le Bloc peuvent contenir jusqu'à 999 Rubriques chacun. Un message d'erreur sera édité dans le compte-rendu de génération si cette limite est dépassée.
		GSQ	(A) Génération du Bloc Base de Données Relationnelles SQL.
			Lors de la phase de génération, les Segments appelés dans le Bloc peuvent contenir jusqu'à 999 Rubriques chacun. Un message d'erreur sera édité dans le compte-rendu de génération si cette limite est dépassée.

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
	LCB	Liste des Blocs Bases de Données par code.
	LEB	Liste des Blocs Bases de Données par nom externe.
	LES	Liste des objets SQL par nom externe.
	LKB	(K) Liste des Blocs Bases de Données par mot-clé.
	LNB	Liste des Blocs Bases de Données par nom.
	LTB	Liste des Blocs Bases de Données par type.
	LTS	Liste des objets SQL par code.
		Dossiers, Vues de Dossier, Composants Applicatifs, Ecrans C/S (Composants Clients TUI), Ecrans, Dialogues.
	DCO	(S) Descriptif des Ecrans sélectionnés.
	DGC	(S) Descriptif d'un Ecran Pacbench C/S
	DGS	(S) Descriptif d'un Composant Applicatif Pacbench C/S.
	DSO	(S) Descriptif de l'Ecran sélectionné.
	FGC	(A) Contrôle du flot Ecran Pacbench C/S.
	FGE	(A) Contrôle du flot libellés d'erreur Pacbench C/S.
	FGS	(A) Contrôle du flot Composant Applicatif.
	FLE	(A) Contrôle du flot libellés d'erreur Dialogue.
	FLO	(A) Contrôle du flot Ecran.
	FME	(A) Contrôle du flot libellés d'erreur eBusiness.
	FMS	(A) Contrôle du flot Serveur.
	FSO	(A) Contrôle du flot Ecran source.
	GCO	(A) Génération des Ecrans sélectionnés.
		Lors de la phase de génération, les Segments appelés dans un Ecran peuvent contenir jusqu'à 999 Rubriques chacun. Un message d'erreur sera édité dans le compte-rendu de génération si cette limite est dépassée.
		Note : Si vous avez indiqué l'option 'Formatage du COBOL généré' sur la Bibliothèque, jusqu'à 150 lignes sont possibles dans un paragraphe COBOL (entre deux points).
	GGC	(S) Génération d'un écran C/S (composant client TUI).
		Lors de la phase de génération, les Segments appelés dans un Ecran C/s peuvent contenir jusqu'à 999 Rubriques chacun. Un message d'erreur sera édité dans le compte-rendu de génération si cette limite est dépassée.
	GGs	(S) Génération applicable à un Composant Applicatif, Moniteur de Communication, Serveur d'erreur, Dossier.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Lors de la phase de génération, les Segments appelés peuvent contenir jusqu'à 999 Rubriques chacun. Un message d'erreur sera édité dans le compte-rendu de génération si cette limite est dépassée.
		GEC	Pacbench C/S :
			C1 : génération des libellés d'erreur au niveau du Dialogue Client ou Serveur et de chaque composant
			Note : La zone 'LANG' qui s'affiche après validation vous permet de saisir un code langue (EN ou FR) pour la génération des libellés. Si vous ne saisissez rien, les libellés seront générés dans votre langue de travail. Si vous saisissez un code différent de EN ou FR, les libellés seront générés en anglais.
			C2 : génération obtenue par l'option 1 plus aide en ligne
			C3 : génération des libellés d'erreur uniquement au niveau du dialogue client
		GED	C1 : libellés d'erreur générés au niveau Structure de Données et pour chaque Segment
			Note : La zone 'LANG' qui s'affiche après validation vous permet de saisir un code langue (EN ou FR) pour la génération des libellés. Si vous ne saisissez rien, les libellés seront générés dans votre langue de travail. Si vous saisissez un code différent de EN ou FR, les libellés seront générés en anglais.
			C2 : libellés d'erreur générés par l'option 1 plus aide en ligne
		GEO	Dialogue :
			C1 : génération des libellés d'erreur au niveau Dialogue et pour chaque Ecran du Dialogue
			Note : La zone 'LANG' qui s'affiche après validation vous permet de saisir un code langue (EN ou FR) pour la génération des libellés. Si vous ne saisissez rien, les libellés seront générés dans votre langue de travail. Si vous saisissez un code différent de EN ou FR, les libellés seront générés en anglais.
			Note : La bannière entête du compte-rendu de génération affiche un récapitulatif des erreurs détectées avec la liste, le cas échéant, des écrans non générés.
			C2 : génération obtenue par l'option 1 plus aide en ligne
			C3 : génération des libellés d'erreur uniquement au niveau du dialogue

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
		C4 : création du fichier contenant les informations nécessaires à l'habillage des Ecrans avec Dialog Web Revamping. Attention, des caractères spéciaux dans le code Dialogue/Ecran génèrent une erreur.
		Remarque : Si un suffixe de Segment (ou Ecran) est renseigné sur la ligne suite d'une des 3 commandes précédentes, les libellés d'erreur sont générés/imprimés pour ce Segment (ou Ecran) seulement.
	GEF	(S) Génération des libellés d'erreurs par Dossier C/S.
		Note : La zone 'LANG' qui s'affiche après validation vous permet de saisir un code langue (EN ou FR) pour la génération des libellés. Si vous ne saisissez rien, les libellés seront générés dans votre langue de travail. Si vous saisissez un code différent de EN ou FR, les libellés seront générés en anglais.
	GEI	(S) Génération des libellés d'erreurs par Composant INIT/TERM.
		Note : La zone 'LANG' qui s'affiche après validation vous permet de saisir un code langue (EN ou FR) pour la génération des libellés. Si vous ne saisissez rien, les libellés seront générés dans votre langue de travail. Si vous saisissez un code différent de EN ou FR, les libellés seront générés en anglais.
	GES	(S) Génération des libellés d'erreurs par Composant C/S.
		Note : La zone 'LANG' qui s'affiche après validation vous permet de saisir un code langue (EN ou FR) pour la génération des libellés. Si vous ne saisissez rien, les libellés seront générés dans votre langue de travail. Si vous saisissez un code différent de EN ou FR, les libellés seront générés en anglais.
	GSO	Génération de code source pour l'Ecran sélectionné.
	GVC	(S) Extraction de proxy, applicable à une Vue de Dossier, un Dossier ou à un Composant Applicatif.
	GMF	(S) Génération de Dossiers.
	GMI	(S) Génération Serveur INIT/TERM.
	GMM	(S) Génération Moniteur de Communications.
	GMS	(S) Génération Serveur.
		Lors de la phase de génération, les Segments appelés dans le Serveur peuvent contenir jusqu'à 999 Rubriques chacun. Un message d'erreur sera édité dans le compte-rendu de génération si cette limite est dépassée.

POS	NON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		GME	(S) Génération Serveur d'erreur.
		LCO	Liste des Ecrans par code.
		LEC	Liste des libellés d'erreur par Dialogue Client et pour chaque Ecran C/S ; cette édition ne liste que les libellés ayant déjà été générés.
		LEO	Liste des libellés d'erreur par Dialogue et pour chaque écran ; cette édition ne liste que les libellés ayant déjà été générés.
		LKO	(K) Liste des Ecrans par mot-clé.
		LNO	Liste des Ecrans par nom.
		LOT	Liste des Ecrans par code transaction renseigné.
		LPO	Liste des Ecrans C/S par nom externe du programme.
		LSO	Liste des Ecrans C/S par nom externe de la grille.
		LTO	Liste des Ecrans par type.
			ETATS
		DCR	(S) Descriptif des Etats.
			Note : Dans la zone ENTITE, saisissez un code Etat (sur 3 caractères) pour éditer le Descriptif de cet Etat, ou un blanc pour éditer le Descriptif de tous les Etats. (Les commandes créées avec les versions antérieures à la 3.5 V05 doivent être modifiées manuellement avant de soumettre la demande d'édition).
		LCR	Liste des Etats par code.
		LTR	Liste des Etats par type.
		LKR	(K) Liste des Etats par mot-clé.
		LNR	Liste des Etats par nom.
			Programmes
		DCP	(S) Descriptif des Programmes.
		DSP	(S) Descriptif du Programme sélectionné issu de Reverse engineering.
		FLP	(A) Contrôle du flot.
		FSP	(A) Contrôle du flot : Programmes issus de Reverse engineering.
		GCP	(A) Génération du Programme sélectionné.

POS	NON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Lors de la phase de génération, les Segments des Structures de Données appelées dans le Programme pourront contenir jusqu'à 9999 Rubriques. Un message d'anomalie sera édité dans le compte-rendu de génération si cette limite est dépassée.
			Note : Si vous avez indiqué l'option 'Formatage du COBOL généré' sur la Bibliothèque, jusqu'à 150 lignes sont possibles dans un paragraphe COBOL (entre deux points).
		GSP	(A) Génération du programme sélectionné issu de Reverse engineering (comportant des lignes -SC).
		LCP	Liste des Programmes par code (le nombre de bibliothèques pour un passage GPRT avec le même code utilisateur est limité à 98).
		LEP	Liste des Programmes par nom externe.
		LKP	(K) Liste des Programmes par mot-clé.
		LNP	Liste des Programmes par nom.
		LTP	Liste des Programmes par type.
			Entités Méthode
		DCM	(S) Descriptif de l'entité Méthode sélectionnée.
		DCMC	Descriptif des C.I.F. d'une Méthode.
		DCMO	Descriptif des Objets Méthode.
		DCMR	Descriptif des Relations Méthode.
		LCMC	Liste des C.I.F.
		LCMO	Liste des Objets Méthode par code.
		LCMP	Liste des Propriétés par code.
		LCMR	Liste des Relations (avec leurs C.I.F.) par code.
		LKM	Liste des entités Méthode dont les noms et/ou les mots explicites contiennent les mots-clés spécifiés.
			Méta-Entités
		DCF	(S) Descriptif des Méta-Entités.
		DCQ	(S) Descriptif des Relations Utilisateur.
		DCY	(S) Descriptif de l'Entité Utilisateur extension sélectionnée.
		DC\$	(S) Descriptif des Entités Utilisateur (attention, cette commande doit obligatoirement être de la forme DC\$xx, avec xx correspondant au type d'appel d'entité).
		LCF	Liste des Méta-Entités par code.

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
	LCQ	Liste des Relations Utilisateur par code.
	LCY	Liste des Entités Utilisateur Extension par code (code entité obligatoire).
	LC\$	Liste des Entités Utilisateur par type et code (code entité obligatoire). (attention, cette commande doit obligatoirement être de la forme LC\$xx, avec xx correspondant au type d'appel d'entité).
	LKF	Liste des Méta-Entités par mot-clé.
	LKQ	(K) Liste des Relations Utilisateur par mot-clé.
	LKY	(K) Liste des Entités Utilisateur Extension par mot-clé.
	LK\$	(K) Liste des Entités Utilisateur par mot-clé.
	LNf	Liste des Méta-Entités par nom.
	LNQ	Liste des Relations Utilisateur par nom.
	LNy	Liste des Entités Utilisateur Extension par nom (code entité obligatoire).
	LN\$	Liste des Entités Utilisateur par nom (code entité obligatoire).
		Transcodification des minuscules
	UPC	Transcodification automatique des minuscules en majuscules ; les lettres et les lettres accentuées courantes seront transformées au moment de l'édition ; cette transcodification porte sur les éditions de toute documentation utilisateur (MANUEL:), du dossier de toute entité (DOC:), des libellés d'erreur (LIB ERR:) ; la commande UPC affiche le libellé 'MISE EN MAJUSCULE MANUEL:_ DOC:_ LIB.ERREUR:_ ' ; indiquer '1' pour l'entité à transcoder.
		Génération utilisateur
	GUT	(S) Génération Commandes Utilisateur.
		Une commande GUT se compose d'une première ligne de type commentaire et de une à cinq lignes suite contenant les informations à transmettre au flot de génération. Aucun contrôle n'est effectué sur ce code commande.
		Première ligne : le code entité est facultatif ; le libellé est constitué de deux parties modifiables (la première sur 25 caractères est initialisée par 'GENERATION UTILISATEUR' et la deuxième sur 11 caractères est à la charge de l'utilisateur) ; seule la zone de 11 caractères est transmise à la procédure de génération et apparaît sur l'état PAC7ID.

POS	NON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Lignes suite : cinq lignes suite sont autorisées ; elles sont simplement transmises dans le flot de génération et se retrouvent dans le compte-rendu PAC7ID ainsi que dans le fichier intermédiaire PAC7JG ; pour traiter ce type de demande, il faut adapter la chaîne de génération.
			Remarque : une commande GUT sans ligne suite génère quand même une ligne dans le flot de génération, mais avec espace dans la commande.
			Edition Tables PAF des Entités Méthode
		PCM	(S) Descriptif des tables PAF pour les entités spécifiques d'une Méthode ; cette commande est nécessairement suivie du code de la Méthode concernée.
9	6		Code entité
			Suivant la nature de la demande d'édition de la documentation, cette zone permet de préciser le code de l'occurrence d'entité à éditer ; elle peut prendre toutes les valeurs de type d'entités contenues dans la base.
			Pour la commande PCM les valeurs possibles pour les méthodes sont :
		M	Merise
		D	YSM
		O	OMT
		F	IFW
15	1		Indicateur sélection bibliothèque
			Permet de sélectionner les Bibliothèques à partir desquelles l'édition doit s'effectuer.
		C	Bibliothèques de niveau supérieur ou égal, avec sélection de la ligne de plus bas niveau en cas de doubles.
16	1		OPTION A EDITER
			Permet d'indiquer l'option d'édition ;
			il peut exister jusqu'à cinq options numérotées de 1 à 4 ainsi que 9 (option par défaut : 1).
			Chaque option correspond généralement à des variantes de présentation des lignes à éditer, dans le cas où vous désirez obtenir des informations supplémentaires pour une édition (édition avec ou sans mot-clé, programmes avec ou sans documentation ventilée, ...).
			Le détail de chacune des options d'édition est donné pour chaque entité dans les manuels de référence correspondants.

POS	CLASSE	VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
			La valeur 9 est dédiée à RPPz.
17	2		Critère de génération
			Permet de saisir le code langue pour les commandes d'édition-génération de type GEx.
19	1		Option cartes avant programme
			Code option du jeu de lignes de contrôle à insérer avant chaque Programme généré ; au niveau de la Bibliothèque, cette option représente les valeurs par défaut à affecter aux Programmes lors de leur création.
20	1		Option cartes après grille
			Entités Ecran et Ecran/CS : code option du jeu de lignes de contrôle à insérer avant chaque grille d'Ecran ou Ecran/CS générée.
		\$	Pas de génération.
			NOTE : cette zone n'est pas utilisée dans le cas d'un développement Pacbench C/S avec Dossier.
			Composant Applicatif / mono-vue (sans Dossier) : code option du jeu de lignes de contrôle à insérer avant le Gestionnaire de Services généré ; la valeur '\$' inhibe la génération du Gestionnaire de Services et autorise la génération du Composant Applicatif.
21	1		Option cartes après programme
			Code option du jeu de lignes de contrôle à insérer après chaque Programme généré ; au niveau de la Bibliothèque, cette option représente les valeurs par défaut à affecter aux Programmes lors de leur création.
22	1		Option cartes après grille
			Entités Ecran et Ecran/CS : code option du jeu de lignes de contrôle à insérer après chaque grille d'Ecran ou Ecran/CS générée.
		\$	Pas de génération
			NOTE : cette zone n'est pas utilisée dans le cas d'un développement Pacbench C/S avec Dossier
			Composant Applicatif / mono-vue (sans Dossier) : code option du jeu de lignes de contrôle à insérer après le Gestionnaire de Services généré.
23	1		Impression partielle d'un Document
			Zone affichée avec la commande PCV uniquement.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		blanc	Edition du Document entier (valeur par défaut)
		C ou 1	Edition d'un chapitre ou d'une section de niveau 1, respectivement. Zone utilisée conjointement à la zone suivante.
		S ou 2	Edition d'un sous-chapitre ou d'une section de niveau 2 (appartenant à la section de niveau 1 indiquée dans la zone suivante), respectivement. Zone utilisée conjointement aux deux zones suivantes.
24	2		Code Chapitre/Num. Section Niveau 1
		C	La valeur 'ZZ' est interdite CH/
26	2		Code Sous-Chap./Num.Section Niveau 2
		C	SC/
30	1		Type de sélection par mot clé
		blanc	Sélection sur libellé et mots clés explicites
		L	Sélection sur libellé uniquement
		M	Sélection sur mots clés explicites uniquement
31	50		Libellé édition suite

GPRT - Lancement de la procédure

GPRT - Traitement des flots générés

Dans le cas où il y a génération de sources à compiler et où le code retour de l'étape précédente est inférieur à 8, le flot généré doit être traité pour assurer la compilation et l'édition de lien des sources produits.

Quelle que soit la solution adoptée, les différents sources générés peuvent être précédés et/ou suivis de lignes de commande, dont la saisie est optionnelle.

Les Lignes de Commande AVANT/APRES sont saisies par l'Administrateur.

Ce sont des commandes qui permettent d'ajouter des lignes au début et/ou à la fin du source COBOL généré (par exemple pour séparer les différents programmes).

Pour chaque type de lignes AV/AP, il y a un code précisé sur la Définition du Programme ou de la Bibliothèque, ou bien encore lors de la Demande de Génération.

GPRT - Exemple de script de génération (C4)

Ces instructions permettent d'exécuter toute commande d'Edition/Génération, saisie soit dans le Gestionnaire de génération de Developer workbench, soit dans l'écran GP.

L'Administrateur initialise pour chaque utilisateur les scripts de génération adaptés à une Base VisualAge Pacbase.

Chaque utilisateur peut modifier SON script de génération en utilisant soit le Gestionnaire de génération de Developer workbench (Onglet Lignes de commandes), soit l'écran GP en vue C4.

Script de Génération (C4) - Lignes Avant

```
<job id=BVPGPRT>
<script language="VBScript">
Dim WshSh
Set WshSh = WScript.CreateObject("WScript.Shell")
Set WshEnv = WshSh.Environment("PROCESS")
Set Args = Wscript.Arguments
</script>
<resource id="GPRT"><![CDATA[
```

Script de Génération (C4) - Lignes Après

```
]]></resource>
<script language="VBScript">
Rep = "HKLM\SOFTWARE\IBM\BVP VisualAge Pacbase 3.5"
Rep = Rep & "\server"
Rep_PROC = WshSh.RegRead (Rep & "\BVP_SYS\PROC\")
USER=Args(0) 'code utilisateur
BASE=Args(1) 'code externe de la base
WshEnv("BVP_Updtpm") = "NO"
WshEnv("BVP_Merge") = "NO"
WshEnv("BVP_Gpmon") = "YES"
WshEnv("BVP_Resource") = getresource("GPRT")
PROC=Rep_PROC & "\gprr.wsf"
CMD=chr(34) &PROC & chr(34)&" " &BASE & " " & USER
RetGprt = WshSh.Run(CMD , 1 , TRUE )
</script>
</job>
```

Dans ces Lignes Après, on doit positionner les variables d'environnement suivantes pour exécuter PACAGP :

BVP_Merge, permet de fusionner dans le fichier GPRTOM les fichiers générés dans l'étape GPRT.

BVPACAGP, pour permettre l'exécution du traitement spécifique PACAGP.

Pour utiliser SCM, la variable d'environnement suivante doit être positionnée dans ces Lignes Après :

BVP_Updtpm, qui déclenche l'exécution de la mise à jour UPPM, pour SCM.

Pour visualiser les fichiers générés dans la Station de Travail (module GPMON) dans un format correct, il est nécessaire de positionner le paramètre suivant dans les Lignes Après :

```
WshEnv("BVP_Gpmon") = "YES"
```

GPRT - Traitements spécifiques

INTERFACE AVEC WORKBENCH

L'objet de cette interface est de séparer dans des fichiers distincts les sources des programmes, écrans ou clauses 'COPY' générés, puis d'écrire ces fichiers sous un répertoire spécifié par l'utilisateur.

Le programme 'bvpsplit' effectue ces traitements.

La mise en oeuvre de cette option ne peut se faire qu'en activant un fichier de commandes dans la procédure GPRT (exemple BVPACAGP) et nécessite la définition de LIGNES AVANT pour les entités VA Pac à traiter, dans le Gestionnaire de Génération.

DEFINITION DES LIGNES de COMMANDE - AVANT (Gestionnaire)

Afin que le programme 'bvpsplit' puisse séparer les fichiers source en sortie de génération, il est nécessaire d'insérer des LIGNES AVANT contenant, outre des chaînes de caractères spécifiques à ces lignes, le nom du fichier à produire, son extension et le répertoire sous lequel il est attendu.

La première LIGNE de COMMANDE AVANT doit contenir :

```
*****   Délimiteur pour 'bvpsplit', obligatoirement  
          en colonnes 1 à 7  
nomfich   Nom du fichier à produire  
ext       Extension, sur 3 caractères maximum
```

Les informations sont séparées par un blanc. Par exemple :

```
***** MONPROG cb1
```

La deuxième LIGNE de COMMANDE AVANT, doit contenir :

```
*&&&&*   Délimiteur pour 'bvpsplit', obligatoirement  
          en colonnes 1 à 7
```

chemin Répertoire d'écriture des fichiers.
Le répertoire choisi doit exister et doit
être accessible par la procédure GPRT.

Les informations sont séparées par un blanc.

Par exemple :

```
*&&&&* S:\COBOL\2592T\CODUTI1
```

Soit, par exemple pour les LIGNES AVANT programme :

l'utilisateur génère à partir de sessions historisées et souhaite récupérer ses programmes générés sous la forme

'nom_externe'.cbl, sous le répertoire

S:\COBOL\'session'\code_uti' du réseau.

Les cartes de contrôle sont définies dans le Workbench Administration ('D' définit la carte AVANT et 'W' sera le code carte) :

```
***** <External name of generated program or block> cbl  
*&&&&* S:\cobol\<<Session number>\<User code>
```

Il faut ensuite appeler ces LIGNES AVANT (code 'W' dans l'exemple) sur les entités à générer.

MISE EN OEUVRE DANS LA PROCEDURE GPRT

Dans le fichier de lancement de la procédure GPRT, la variable BVPACAGP doit être positionnée et contenir le chemin du fichier des commandes qui sera exécuté à la fin de la procédure GPRT, après les éditions générations.

Ensuite, il faut appeler le programme 'bvpsplit' dans le fichier des commandes BVPACAGP, en lui indiquant le répertoire de génération de l'utilisateur.

BVPACAGP doit donc contenir l'exécution de 'bvpsplit' avec en argument le répertoire des générés.

TRAITEMENTS ET MESSAGES D'ERREUR

Le programme 'bvpsplit' traite tous les fichiers GPRTO* en sortie de la procédure GPRT, sous le répertoire désigné.

Un compte rendu d'exécution est édité sous le répertoire de l'utilisateur (Rep_USR, par défaut).

Exemple de script BVPACAGP en .vbs

```
' =====  
' VisualAge Pacbase : BVPACAGP.vbs  
' =====  
' Args(0) = User code  
' Args(1) = Job number  
' Args(2) = User directory  
  
Set WshShell = Wscript.CreateObject("Wscript.Shell")  
Set Args = Wscript.Arguments  
Set WshEnv = WshShell.Environment("PROCESS")  
Set WshVolEnv = WshShell.Environment("VOLATILE")  
MyProc = Wscript.ScriptName  
UserDir = Ucase(Args(2))  
  
' starting BVPSPPLIT  
EXESPLIT = chr(34) & "BVPSPPLIT" & chr(34) & " " _  
           & chr(34) & UserDir & chr(34)  
Return = WshShell.Run(EXESPLIT , 1 , TRUE )  
Set Args = Wscript.Arguments  
Wscript.Quit (Return)
```

Exemple de script BVPACAGP en .cmd

```
REM =====  
REM VisualAge Pacbase : BVPACAGP.cmd  
REM =====  
REM %1 = User code  
REM %2 = Job number  
REM %3 = User directory  
  
echo %0 command procedure > %3/BVPACAGP.LOG  
echo ----- >> %3/BVPACAGP.LOG  
echo  
echo User code = %1 >> %3/BVPACAGP.LOG  
echo Job number = %2 >> %3/BVPACAGP.LOG  
echo user directory = %3 >> %3/BVPACAGP.LOG  
echo  
echo execution : BVPSPPLIT %3 >> %3/BVPACAGP.LOG  
BVPSPPLIT " %3 "  
if ERRORLEVEL 1 GOTO ERR  
GOTO END  
:ERR  
echo error in executing BVPSPPLIT >> %3/BVPACAGP.LOG  
EXIT 1  
:END  
EXIT 0
```

GPRT - Description des étapes

Editions et générations : PACB

La documentation fournie dépend des commandes d'édition-génération prises en compte ; son volume ainsi que celui des fichiers intermédiaires est donc extrêmement variable. Des bannières en tête et en fin de la documentation permettent d'identifier les utilisateurs qui ont lancé des éditions.

Tous les programmes, écrans, blocs base de données, etc., éventuellement générés sont récupérés sur des fichiers dont le nom est de la forme GPRTOx.

Certains programmes appelés par le moniteur peuvent envoyer des codes retour particuliers :

- BVPACA10 (Récupération des mouvements) :
 - 0 : OK
 - 2 : OK avec présence de la commande ' +AG'
 - 8 : Aucune demande. Dans ce cas, la procédure s'arrête.
 - BVPACB31 (Génération SQL) :
 - 8 : Une erreur a été détectée lors de la génération.
 - Extracteurs ou générateurs (30 ou 40)
 - 0 : OK - Pas de génération
 - 4 : OK - Génération
 - Autre : Erreurs
 - BVPACW10 (Support gestion de configuration)
 - 0 : OK
 - 2 : Pas de traitement
 - 4 : Au moins une erreur de paramétrage est rencontrée.
 - 8 : Au moins une erreur de contexte rencontrée.
- Cette étape envoie également un code retour général :

Code	Libellé
4	OK avec génération de sources à compiler
6	OK avec génération de sources à compiler et de documentation personnalisée ou de libellés d'erreur
8	OK avec génération de documentation personnalisée ou de libellés d'erreur
10	OK sans génération
12	Erreur d'entrée-sortie
16	Erreur de tri

GPRT - Script d'exécution

```

| -----
|          VISUALAGE PACBASE
|
| -----
|          - GENERATION (IN INTERNAL READER) AND PRINTING -
|
| -----
|
| IN ADDITION TO THE GENERATED ENTITIES, THE FILE MUST

```

```
' CONTAIN THE JCL REQUIRED TO COMPILE THEM,
' USING THE BEGINNING/END OF JCL JOB STREAM OPTIONS AND
' THE BEFORE/AFTER PROGRAM OPTIONS.
'
' THE GENERATION AND PRINTING PROCEDURE, GPRT, HAS A
' TWO-FOLD PURPOSE:
' . TO PRINT DOCUMENTATION USING DATA CONTAINED IN THE
'   DATABASE, AND
' . TO GENERATE PROGRAMS, SCREENS, DATABASE
'   DESCRIPTIONS DATA STRUCTURES, AND ERROR MESSAGES.
'
' -----
'
```

```
<job id=GPRT>
```

```
<script language="VBScript">
Dim MyProc
MyProc = "GPRT"
</script>
<script language="VBScript" src="INIT.vbs"/>
```

```
<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If
```

```
Dim CodLang
If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
Else
CodLang = WshShell.RegRead (Rep_SYS & "\GENLANG")
End If
```

```
If Not FSO.FileExists( Rep_AJOURNAL & "\QJ") Then
Call Msg_Log (Array("1022", "PCMINI"))
```

```
'-----
WshEnv("PAC7QJ") = Rep_AJOURNAL & "\QJ"
Call RunCmdLog ("BVPCMINI")
```

```
Call Err_Cod(Return , 0 , "PCMINI")
End if
```

```
If Not FSO.FileExists( Rep_ABASE & "\GK") Then
Call Msg_Log (Array("1022", "PTUIGK"))
```

```
'-----
WshEnv("PACGGK") = Rep_ABASE & "\GK"
Call RunCmdLog ("BVPTUIGK")
```

```
Call Err_Cod(Return , 0 , "PTUIGK")
End if
```

```

Call Msg_Log (Array("1022" , "PACB"))
'-----
WshEnv("PAC7QJ") = Rep_AJOURNAL & "\QJ"
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7LB") = Rep_BASE & "\LB"
WshEnv("PACGGK") = Rep_ABASE & "\GK"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7GS") = Rep_BASE & "\GS"
WshEnv("PAC7SC") = Rep_SKEL & "\SC" & CodLang
WshEnv("PAC7SF") = Rep_SKEL & "\SF" & CodLang
WshEnv("PAC7SG") = Rep_SKEL & "\SG" & CodLang
WshEnv("PAC7SN") = Rep_SKEL & "\SN" & CodLang
WshEnv("PAC7SR") = Rep_SKEL & "\SR" & CodLang
WshEnv("PAC7SS") = Rep_SKEL & "\SS" & CodLang
WshEnv("PAC7SL") = Rep_SKEL & "\SL" & CodLang
WshEnv("PAC7ME") = Fic_Input
Call BvpEnv("PACB", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("PACB", "PAC7DG", Rep_TMP & "\WDG.tmp")
Call BvpEnv("PACB", "PAC7EB", Rep_TMP & "\WEB.tmp")
Call BvpEnv("PACB", "PAC7EE", Rep_TMP & "\WEE.tmp")
Call BvpEnv("PACB", "PAC7EG", Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACB", "PAC7EI", Rep_TMP & "\WEI.tmp")
Call BvpEnv("PACB", "PAC7EN", Rep_TMP & "\WEN.tmp")
Call BvpEnv("PACB", "PAC7EP", Rep_TMP & "\WEP.tmp")
Call BvpEnv("PACB", "PAC7EQ", Rep_TMP & "\WEQ.tmp")
Call BvpEnv("PACB", "PAC7ER", Rep_TMP & "\WER.tmp")
Call BvpEnv("PACB", "PAC7EV", Rep_TMP & "\WEV.tmp")
Call BvpEnv("PACB", "PAC7EW", Rep_TMP & "\WEW.tmp")
Call BvpEnv("PACB", "PAC7OB", Rep_USR & "\GPRTOB.txt")
Call BvpEnv("PACB", "PAC7GB", Rep_TMP & "\GPRTGB.tmp")
Call BvpEnv("PACB", "PAC7OD", Rep_USR & "\GPRTOD.txt")
Call BvpEnv("PACB", "PAC7GD", Rep_TMP & "\GPRTGD.tmp")
Call BvpEnv("PACB", "PAC7OE", Rep_USR & "\GPRTOE.txt")
Call BvpEnv("PACB", "PAC7GE", Rep_TMP & "\GPRTGE.tmp")
Call BvpEnv("PACB", "PAC7OF", Rep_USR & "\GPRTOF.txt")
Call BvpEnv("PACB", "PAC7GF", Rep_TMP & "\GPRTGF.tmp")
Call BvpEnv("PACB", "PAC7OG", Rep_USR & "\GPRTOG.txt")
Call BvpEnv("PACB", "PAC7GG", Rep_TMP & "\GPRTGG.tmp")
Call BvpEnv("PACB", "PAC7GI", Rep_USR & "\GPRTGI.txt")

Call BvpEnv("PACB", "PAC7GK", RepT_USR & "\ERRGK.txt")
Call BvpEnv("PACB", "PAC7GL", RepT_USR & "\ERRGL.txt")
Call BvpEnv("PACB", "PAC7GM", RepT_USR & "\ERRGM.txt")

Call BvpEnv("PACB", "PAC7GN", Rep_TMP & "\WXGN.tmp")
Call BvpEnv("PACB", "PAC7GO", Rep_TMP & "\WGO.tmp")
Call BvpEnv("PACB", "PAC7OP", Rep_USR & "\GPRTOP.txt")
Call BvpEnv("PACB", "PAC7GP", Rep_TMP & "\GPRTGP.tmp")
Call BvpEnv("PACB", "PAC7OQ", Rep_USR & "\GPRTOQ.txt")

```

```

Call BvpEnv("PACB","PAC7GQ",Rep_TMP & "\GPRTGQ.tmp")
Call BvpEnv("PACB","PAC7OR",Rep_USR & "\GPRTOR.txt")
Call BvpEnv("PACB","PAC7GR",Rep_TMP & "\GPRTGR.tmp")
Call BvpEnv("PACB","PAC7GT",Rep_USR & "\PAWGT.txt")
Call BvpEnv("PACB","PAC7OV",Rep_USR & "\GPRTOV.txt")
Call BvpEnv("PACB","PAC7GV",Rep_TMP & "\GPRTGV.tmp")
Call BvpEnv("PACB","PAC7G6",Rep_USR & "\GPRTG6.txt")
Call BvpEnv("PACB","PAC7DB",Rep_USR & "\GPRTDB.txt")
Call BvpEnv("PACB","PAC7IA",Rep_USR & "\GPRTIA.txt")
Call BvpEnv("PACB","PAC7ID",Rep_USR & "\GPRTID.txt")
Call BvpEnv("PACB","PAC7IK",Rep_USR & "\GPRTIK.txt")
Call BvpEnv("PACB","PAC7IL",Rep_USR & "\GPRTIL.txt")
Call BvpEnv("PACB","PAC7IM",Rep_USR & "\GPRTIM.txt")
Call BvpEnv("PACB","PAC7IN",Rep_USR & "\GPRTIN.txt")
Call BvpEnv("PACB","PAC7IO",Rep_USR & "\GPRTIO.txt")
Call BvpEnv("PACB","PAC7IW",Rep_USR & "\GPRTIW.txt")
Call BvpEnv("PACB","PAC7JG",Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACB","PAC7KB",Rep_TMP & "\WKB.tmp")
Call BvpEnv("PACB","PAC7KD",Rep_TMP & "\WKD.tmp")
Call BvpEnv("PACB","PAC7KE",Rep_TMP & "\WKE.tmp")
Call BvpEnv("PACB","PAC7KF",Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACB","PAC7KG",Rep_TMP & "\WKG.tmp")
Call BvpEnv("PACB","PAC7KM",Rep_TMP & "\WKM.tmp")
Call BvpEnv("PACB","PAC7KN",Rep_TMP & "\WKN.tmp")
Call BvpEnv("PACB","PAC7KP",Rep_TMP & "\WKP.tmp")
Call BvpEnv("PACB","PAC7KQ",Rep_TMP & "\WKQ.tmp")
Call BvpEnv("PACB","PAC7KR",Rep_TMP & "\WKR.tmp")
Call BvpEnv("PACB","PAC7KS",Rep_TMP & "\WKS.tmp")
Call BvpEnv("PACB","PAC7KU",Rep_TMP & "\WKU.tmp")
Call BvpEnv("PACB","PAC7KV",Rep_TMP & "\WKV.tmp")
Call BvpEnv("PACB","PAC7LG",Rep_TMP & "\NUL.tmp")
'PAC7LG not used, on default
Call BvpEnv("PACB","PAC7LI",Rep_TMP & "\WLI.tmp")
Call BvpEnv("PACB","PAC7LK",Rep_TMP & "\NUL.tmp")
'PAC7LK not used, on default
Call BvpEnv("PACB","PAC7LM",Rep_TMP & "\NUL.tmp")
'PAC7LM not used, on default
Call BvpEnv("PACB","PAC7MG",Rep_TMP & "\WVG.tmp")
Call BvpEnv("PACB","PAC7MV",Rep_TMP & "\WVW.tmp")
Call BvpEnv("PACB","PAC7SO",Rep_TMP & "\WSO.tmp")
Call BvpEnv("PACB","PAC7WA",Rep_TMP & "\WWA.tmp")
Call BvpEnv("PACB","PAC7WC",Rep_TMP & "\WWC.tmp")
Call BvpEnv("PACB","PAC7W1",Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACB","PAC7W2",Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACB","PAC7W3",Rep_TMP & "\WW3.tmp")
Call BvpEnv("PACB","PAC7W4",Rep_TMP & "\WW4.tmp")
Call BvpEnv("PACB","PAC7W6",Rep_TMP & "\WW6.tmp")
Call BvpEnv("PACB","PAC7W7",Rep_TMP & "\WW7.tmp")
Call BvpEnv("PACB","PAC7W8",Rep_TMP & "\WW8.tmp")
Call BvpEnv("PACB","PAC7W9",Rep_TMP & "\WW9.tmp")
Call BvpEnv("PACB","SYSPAF",Rep_TMP & "\WPAF.tmp")
Call RunCmdLog ("BVPACB")

```

```

If Return < 10 then
Call Msg_Log (Array("1062"))

```

```

Return = 0
End if
If Return = 10 then
Call Msg_Log (Array("1063"))
Return = 0
End if

If Return > 10 then
Call Msg_Log (Array("1064"))
End if
Call Err_Cod(Return , 10 , "PACB")

If BVP_Merge = "YES" then
Call Msg_Log (Array("1022" , "COPY in OM"))
'-----
If Not FS0.FileExists(Rep_USR & "\GPRTOM.txt") Then
Set LogGen = FS0.CreateTextFile(Rep_USR & "\GPRTOM.txt", TRUE)
LogGen.Close
End if
OM = FS0.GetFile(Rep_USR & "\GPRTOM.txt")

Call CopMFiL(OM , WshEnv("PAC70P") ,OM )
Call DelFile (WshEnv("PAC70P"))
Call CopMFiL(OM , WshEnv("PAC70Q") ,OM )
Call DelFile (WshEnv("PAC70Q"))
Call CopMFiL(OM , WshEnv("PAC70E") ,OM )
Call DelFile (WshEnv("PAC70E"))
Call CopMFiL(OM , WshEnv("PAC70R") ,OM )
Call DelFile (WshEnv("PAC70R"))
Call CopMFiL(OM , WshEnv("PAC70G") ,OM )
Call DelFile (WshEnv("PAC70G"))
Call CopMFiL(OM , WshEnv("PAC70V") ,OM )
Call DelFile (WshEnv("PAC70V"))
Call CopMFiL(OM , WshEnv("PAC70D") ,OM )
Call DelFile (WshEnv("PAC70D"))
Call CopMFiL(OM , WshEnv("PAC70F") ,OM )
Call DelFile (WshEnv("PAC70F"))
Call CopMFiL(OM , WshEnv("PAC70B") ,OM )
Call DelFile (WshEnv("PAC70B"))

Set OM = FS0.GetFile(Rep_USR & "\GPRTOM.txt")
FilFull = OM.size

If FilFull = 0 then DelFile(OM) end if

End If

If BVPACAGP <> " " then
Call Msg_Log (Array("1022" , BVPACAGP))
'-----
Return = WshShell.Run(BVPACAGP , 1 , TRUE )
Call Err_Cod(Return , 0 , BVPACAGP)
End If

If BVP_Updtpm = "YES" then

```

```

    Call Msg_Log (Array("1022" , "PCMPUF"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7IC") = Rep_TMP & "\NUL.tmp"
'PAC7IC not used, on default
WshEnv("PAC7QJ") = Rep_AJOURNAL & "\QJ"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call RunCmdLog ("BVPCMPUF")
Call Err_Cod(Return , 0 , "PCMPUF")
end if

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr(Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

GPRT - Fichiers générés

Codification des fichiers créés sur disque

Tous les fichiers résultats de la procédure d'édition génération sont créés sous le répertoire des fichiers temporaires.

Une codification particulière a été adoptée pour ces fichiers, afin que l'utilisateur retrouve aisément ses générés ou éditions.

Fichiers contenant des générés ou éditions

Ils sont tous préfixés par "GPRT.". Par exemple, on trouvera :

Générés	Editions
-----	-----
GPRT0B (Blocs)	GPRTIA (Compte rendu)
GPRT0Q (SQL)	
GPRT0D (Data)	GPRTID (Dictionnaire)
GPRT0E (Ecrans - OSD)	GPRTIH (PEI)
GPRT0P (Programmes)	GPRTIL (Libellés Err. OSD)

GPRTOR (Reverse)	GPRTIN (Rapports)
GPRTOG (Ecrans Client)	GPRTIK (Libellés Err. OCS)
GPRTOV (Ecrans Serveur)	GPRTII (Err.Génération ICS)
GPRTOF (e-Business)	

Fichiers des libellés d'erreur

Ils sont tous préfixés par "ERR.". On trouvera donc :

Fichiers en entrée : ERRLG et ERRLK
Fichiers en entrée : ERR.LG (OSD) et ERR.LK (OCS)
Fichiers en sortie : ERRGL et ERRGK

En fin de procédure, on assure la rotation de GL sur LG et de GK sur LK par une copie.

Fichier de l'habillage automatique d'applications Dialogue

Il est préfixé par "PAW.". On trouvera donc :

PAW.GT qui contient les éléments nécessaires au "windowing".

Fichiers temporaires

Ce sont des fichiers internes à la procédure.

Ils sont préfixés par "WW" et sont détruits en fin de procédure. L'utilisateur n'a, a priori, jamais besoin de les récupérer.

Ils sont préfixés par "WX" et sont détruits en fin de procédure. Si l'utilisateur souhaite les récupérer pour des traitements particuliers, il doit changer leur répertoire d'assignation (à copier dans \USERS\...).

WXGI (Interface VisualAge Pacbase-GIP)

WXGM (Libellés type PAC700)

WXGN (Rapports sur 265 caractères)

Remarque concernant la génération des libellés d'erreur

Les libellés d'erreur (commandes GEO ou GGO) doivent être générés de préférence en mode batch. Leur génération par l'écran GP est déconseillée, car le serveur batch (qui traite les demandes soumises depuis l'écran GP) ne peut pas assurer la rotation des fichiers séquentiels générés (pas de génération cumulative possible). Le risque encouru est donc la perte des libellés précédemment générés. Pour éviter cette perte, il faut SYSTEMATIQUEMENT

charger le fichier indexé des libellés d'erreur utilisateur (procédure EMUP) après chaque génération du fichier séquentiel.

Par défaut, la procédure GPRT ne fait pas de génération cumulative des libellés d'erreur, les fichiers LG et LK étant assignés comme fichiers nuls. Pour activer la génération cumulative il faut assigner les fichiers comme suit :

```
WshEnv("PAC7LG") = RepT_USR & "\ERRLG.txt"  
WshEnv("PAC7LK") = RepT_USR & "\ERRLK.txt"
```

Traitement des éditions au format RTF (fic. GPRTG6.txt)

Les fichiers générés au format RTF sur le serveur VA Pac nécessitent une conversion au jeu de caractères ASCII avant leur traitement par la Station de Travail VA Pac.

Commande de conversion au jeu de caractères ASCII :

```
bvptrans <fichier source> <fichier dest> ibm-923 ibm-850
```

Ces commandes peuvent être incluses dans la procédure BVPACAGP.

EMLD - Chargement des libellés d'erreur

EMLD - Présentation générale

Cette procédure permet de définir le fichier des libellés d'erreur de l'utilisateur, et d'effectuer son chargement initial avec les libellés d'erreur générés, sous forme séquentielle, par la procédure GPRT (fichier suffixé GL).

Condition d'exécution

Vous devez, au préalable, avoir procédé à l'exécution de la procédure GPRT avec demande de génération des libellés d'erreur.

Avant le traitement standard, on procède au tri ASCII du fichier des libellés d'erreur (PTUSGL).

EMLD - Entrées Utilisateur

Une ligne '*' avec Code utilisateur et Mot de passe.

EMLD - Description des étapes

Tri des libellés d'erreur séquentiels générés : PTUSGL

Code	Nom physique	Type	Libellé
PAC7LG	Rép. user : ERRGL	Entrée	Libellés d'erreur utilisateur générés

Code	Nom physique	Type	Libellé
PAC7GL	Rép. tmp : ERRGL	Sortie	Libellés d'erreur utilisateur triés

Chargement des libellés d'erreur en fichier indexé : PACL93

Code	Nom physique	Type	Libellé
PAC7MB	Entrée utilisateur	Entrée	Mouvements en entrée
PAC7GL	Rép. tmp : ERRGL	Entrée	Libellés d'erreur utilisateur séquentiels
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Fichier des données de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7EM	Rép. user : ERRMSG	Sortie	Fichier indexé libellés d'erreur utilisateur
PAC7IY	Rép. user : EMLDIYL93	Etat	Compte-rendu
PAC7DD	Rép. user : EMLDDDL93	Etat	Contrôle autorisation

Code retour :

- 8 : Pas d'autorisation procédure batch.

EMLD - Script d'exécution

```

|-----
|      VISUALAGE PACBASE
|-----
|      - LOADING OF USER'S ERROR MESSAGES -
|-----
|
| THE EMLD PROCEDURE PERFORMS THE INITIAL LOADING OF USER
| DEFINED ERROR MESSAGES. THESE MESSAGES ARE OBTAINED
| FROM THE SEQUENTIAL OUTPUT FILE OF THE GPRT PROCEDURE
| (FILE WITH THE GL SUFFIX).
|

```

```

'
' INPUT :
' - USER IDENTIFICATION LINE (REQUIRED)
' -----
'
<job id=EMLD>

<script language="VBScript">
Dim MyProc
MyProc = "EMLD"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call BvpEnv("PTUSGL","PAC7LG",RepT_USR & "\ERRGL.txt")
If FSO.FileExists(WshEnv("PAC7LG")) Then

Call Msg_Log (Array("1022" , "PTUSGL"))
'-----
Call BvpEnv("PTUSGL","PAC7LG",RepT_USR & "\ERRGL.txt")
Call BvpEnv("PTUSGL","PAC7GL",Rep_TMP & "\ERRGL.txt")
Call RunCmdLog ("BVPTUSGL")

Call Msg_Log (Array("1022" , "PACL93"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACL93","PAC7GL",Rep_TMP & "\ERRGL.txt")
Call BvpEnv("PACL93","PAC7EM",RepT_USR & "\ERRMSG")
Call BvpEnv("PACL93","PAC7IY",Rep_USR & "\EMLDIYL93.txt")
Call BvpEnv("PACL93","PAC7DD",Rep_USR & "\EMLDDL93.txt")
Call RunCmdLog ("BVPACL93")

If Return = 8 Then
Call Msg_Log (Array("1027"))
End If
Call Err_Cod(Return , 0 , "PACL93")

Else

Call Msg_Log (Array("1041" , RepT_USR & "\ERRGL.txt"))

End If

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr (Rep_TMP)

```

```
Wscript.Quit (Return)
```

```
</script>  
</job>
```

EMUP - Mise à jour des libellés d'erreur

EMUP - Présentation générale

Cette procédure permet de mettre à jour le fichier des libellés d'erreur de l'utilisateur par les libellés d'erreur obtenus sous forme séquentielle par la procédure GPRT (fichier suffixé GL) ou par mouvements pour l'annulation de libellés d'erreur au niveau entité.

Condition d'exécution

Le fichier des libellés d'erreur utilisateur doit exister.

Dans le cas d'une création et/ou d'une modification de libellés d'erreur, la procédure GPRT doit avoir été exécutée, avec demande de génération de libellés d'erreur.

Avant le traitement standard, on procède au tri ASCII du fichier des libellés d'erreur (PTUSGL).

EMUP - Entrées Utilisateur

Une ligne '*' par bibliothèque contenant les entités dont les libellés d'erreur doivent être annulés :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	bbb	Code bibliothèque

Une ligne commande par entité pour laquelle l'annulation des libellés d'erreur est demandée :

Pos.	Lon.	Valeur	Signification
1	1	'A'	Code mouvement (annulation)
2	2		Code entité (de la zone choix en conversationnel)
		'O '	Ecran
		'D '	Structure de Données

Pos.	Lon.	Valeur	Signification
		'S '	Segment
4	6		Code de l'entité

EMUP - Description des étapes

Tri des libellés d'erreur séquentiels générés : PTUSGL

Code	Nom physique	Type	Libellé
PAC7LG	Rép. user : ERRGL	Entrée	Libellés d'erreur utilisateur générés
PAC7GL	Rép. tmp : ERRGL	Sortie	Libellés d'erreur utilisateur triés

Mise à jour libellés d'erreur utilisateur indexés: PACL92

Code	Nom physique	Type	Libellé
PAC7GL	Rép. tmp : ERRGL	Entrée	Libellés d'erreur utilisateur séquentiels
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7MB	Entrée utilisateur	Entrée	Mouvements en entrée
PAC7EM	Rép. user : ERRMSG	Sortie	Fichier indexé libellés d'erreur utilisateur
PAC7IU	Rép. user : EMUPIUL92	Etat	Compte-rendu des mouvements
PAC7IX	Rép. user : EMUPIXL92	Etat	Récapitulatif des libellés d'erreur
PAC7DD	Rép. user : EMUPDDL92	Etat	Contrôle autorisation

Code retour :

- 8 : Pas d'autorisation procédure Batch.

EMUP - Script d'exécution

```
'-----
'          VISUALAGE PACBASE
'-----
'          - USER'S ERROR MESSAGES UPDATING -
'-----
' THE EMUP PROCEDURE UPDATES THE USER-DEFINED ERROR
' MESSAGE FILE. THESE MESSAGES ARE OBTAINED FROM THE
' SEQUENTIAL OUTPUT FILE OF THE GPRT PROCEDURE (FILE WITH
' A GL SUFFIX) OR FROM TRANSACTIONS FOR ERROR
' MESSAGE DELETIONS AT THE ENTITY LEVEL.
'
' INPUT :
' - USER IDENTIFICATION LINE (REQUIRED)
' - COMMAND LINE :
' COL 1  : "D"    TRANSACTION CODE (DELETION)
' COL 2  : ENTITY TYPE; SAME AS IN CHOICE FIELD.
'          "O "  SCREEN
'          "D "  DATA STRUCTURE
'          "S "  SEGMENT
' COL 4  : (6 CAR.) ENTITY CODE
'-----
<job id=EMUP>
<script language="VBScript">
MyProc = "EMUP"
Dim MyProc
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call BvpEnv("PTUSGL","PAC7LG",RepT_USR & "\ERRGL.txt")
If FSO.FileExists(WshEnv("PAC7LG")) Then

Call Msg_Log (Array("1022" , "PTUSGL"))
'-----
Call BvpEnv("PTUSGL","PAC7LG",RepT_USR & "\ERRGL.txt")
Call BvpEnv("PTUSGL","PAC7GL",Rep_TMP & "\ERRGL.txt")
Call RunCmdLog ("BVPTUSGL")

Call Err_Cod(Return , 0 , "PTUSGL")

Call Msg_Log (Array("1022" , "PACL92"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
```

```

WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACL92","PAC7GL",Rep_TMP & "\ERRGL.txt")
Call BvpEnv("PACL92","PAC7EM",RepT_USR & "\ERRMSG")
Call BvpEnv("PACL92","PAC7IU",Rep_USR & "\EMUPIUL92.txt")
Call BvpEnv("PACL92","PAC7IX",Rep_USR & "\EMUPIXL92.txt")
Call BvpEnv("PACL92","PAC7DD",Rep_USR & "\EMUPDDL92.txt")
Call RunCmdLog ("BVPACL92")

If Return = 8 Then
Call Msg_Log (Array("1027"))
End If
Call Err_Cod(Return , 0 , "PACL92")

Else

Call Msg_Log (Array("1041" , RepT_USR & "\ERRGL.txt"))

End If

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr (Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

PPAF - Pré-processeur PAF des programmes générés

PPAF - Présentation générale

Cette procédure traite les programmes générés utilisateur contenant des requêtes SQL d'accès à la base par opérateurs PAF.

Condition d'exécution

Aucune.

Mise en oeuvre

Cette procédure peut être exécutée de plusieurs manières :

- soit à la suite d'une génération de programmes par GPRT dont la sortie générée est reprise en entrée de la procédure PPAF avant envoi en compilation ou rangement dans une bibliothèque de programmes-sources,
- soit par appel de la procédure dans les lignes de contrôle avant/après du programme généré . Il faut alors avoir préalablement entré le JCL adéquat dans les options choisies (écran PC).

PPAF - Entrées Utilisateur

C'est le code source COBOL des programmes contenant des opérateurs PAF qui doit être résolu par le pré-processeur avant compilation.

Chaque programme contient, après la ligne "Identification division", une ligne de commande pour le pré-processeur :

Pos.	Lon.	Valeur	Signification
1	6	nnnnnn	Numéro de ligne COBOL
7	1	'*'	Commentaire
8	5	'TP '	Programme TP
		'BATCH'	Programme batch
14	5	'LIB: '	Libellé fixe
19	3	bbb	Code bibliothèque
22	1	blanc	non utilisé
23	5	nnnns	Numéro session - Etat de session
28	1	blanc	non utilisé
29	2	--	Variante(s) de génération
32	4	'AR:'	Libellé fixe
36	1	l	Code langue de la base
38	4	'SC:'	Squelette programmes Langage batch
		'SG:'	Squelette programmes Dialogue
		'SR:'	Squelette programmes générés COBOL
42	1	l	Langue du squelette
43	1	blanc	pas utilisé
44	6	'SINGLE'	Simple quotes ou
		'DOUBLE'	Doubles quotes

Exemples :

```
000020*TP____LIB:_APP_2345__00_AR:_F_SG:_F_SINGLE
```

```
000020*BATCH_LIB:_APP_2300T_4__AR:_F_SC:_F_DOUBLE
```

Cette ligne est générée automatiquement par la procédure GPRT.

Edition obtenue

Un compte-rendu des erreurs éventuellement rencontrées.

Résultat obtenu

Le résultat obtenu est le source COBOL dans lequel ont été résolus les opérateurs PAF et générés les appels aux sous-programmes batch ou conversationnel d'accès à la base.

PPAF - Description des étapes

Pré-processeur : PAFP10

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAF80	Rép. user : PAF80	Entrée	Programmes générés
COB80	Rép. user : COB80	Sortie	Programmes générés à compiler
PAFREP	Rép. user : PPAFREP10 ou PAFREP	Etat	Compte-rendu d'erreur

PPAF - Script d'exécution

```
| -----  
|          VISUALAGE PACBASE  
| -----  
|          - VA PAC ACCESS FACILITY PRE-PROCESSING -  
| -----  
|  
| USING PAF OPERATORS, THE PPAF PROCEDURE PROCESSES  
| GENERATED USER PROGRAMS CONTAINING SQL REQUESTS FOR  
| ACCESS TO THE DATABASE.  
|  
| USER INPUT IS THE COBOL SOURCE CODE OF PROGRAMS  
| CONTAINING PAF OPERATORS TO BE PROCESSED BY  
| BY THE PRE-PROCESSOR BEFORE COMPILATION.  
| -----  
|  
<job id=PPAF>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "PPAF"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>
```



```

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PAFP10"))
'-----
WshEnv("COBSW") = "-N"
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
Call BvpEnv("PAFP10","PAF80",Rep_USR & "\PAF80.txt")
Call BvpEnv("PAFP10","COB80",Rep_USR & "\COB80.txt")
Call BvpEnv("PAFP10","PAFREP",Rep_USR & "\PPAFREP10.txt")
Call RunCmdLog ("BVPAPF10")

Call Err_Cod(Return , 0 , "PAFP10")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr (Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

GPRC - Gestion de l' API COBOL

GPRC - Présentation générale

GPRT et l'API COBOL : GPRC

Cette procédure permet l'utilisation en batch des services Client/Serveur tels que le Dossier et les Composants élémentaires.

Dans ce cas la procédure GPRT est complétée par des traitements spécifiques pour donner la procédure GPRC, c'est donc une génération de sources pour l' API COBOL du gestionnaire de dossier. GPRC est une procédure dédiée UNIQUEMENT à ce type de génération.

Pour plus de précisions il faut se référer au manuel "Guide d'utilisation de l'API COBOL".

GPRC - Entrées Utilisateur

Se reporter à la description des entrées utilisateurs GPRT.

GPRC - Description des étapes

Editions et générations : PACB

Le source générée fourni dépend des commandes d'édition-génération prises en compte.

Les entités pouvant utiliser l'API COBOL sont :

- les programmes,
- les macros,
- les écrans,
- les composants élémentaires.

Pour plus de renseignements, se reporter à la description faite pour GPRT.

API COBOL extracteur : PAPG1S

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
GENERE	Rép. tmp : GPRCOM	Entrée	Source générée GPRT
PAC7W1	Rép. tmp : WW1	Sortie	Fichier de travail

API COBOL : PAPG5S

Code	Nom physique	Type	Libellé
PAC7W1	Rép. tmp : WW1	Entrée	Fichier en entrée
PAC7W2	Rép. tmp : WW2	Sortie	Fichier en sortie

Générateur API COBOL : PAPG7S

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7EW	Rép. tmp : WEW	Sortie	Fichier des erreurs de génération

Code	Nom physique	Type	Libellé
PAC7W2	Rép. tmp : WW2	Entrée	Fichier intermédiaire
PAC7W3	Rép. tmp : WW3	Entrée	Fichier intermédiaire
PAC7SA	Système - Rép. skel : SA	Entrée	Squelette des libellés API COBOL

API COBOL - Insertion COBOL : PAPG9S

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7EW	Rép. tmp : WEW	Entrée	Fichier des erreurs de génération
GENERE	Rép. tmp : GPRCOM	Entrée	Source généré GPRT
PAC7W3	Rép. tmp : WW3	Sortie	Fichier intermédiaire
COB80	Rép. user : GPRTOM	Entrée	Source API COBOL généré
PAC7ED	Rép. user : GPRTED	État	Compte-rendu des erreurs

GPRC - Script d'exécution

```

' -----
'     VISUALAGE PACBASE
'
' -----
'     - GENERATION AND PRINTING WITH API COBOL -
'
' -----
'
' IN ADDITION TO THE GENERATED ENTITIES, THE FILE MUST
' CONTAIN THE JCL REQUIRED TO COMPILE THEM,
' USING THE BEGINNING/END OF JCL JOB STREAM OPTIONS AND
' THE BEFORE/AFTER PROGRAM OPTIONS.
'
' -----
'
<job id=GPRC>

<script language="VBScript">
Dim MyProc
MyProc = "GPRC"
</script>
<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

```

```

Dim CodLang
If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
Else
CodLang = WshShell.RegRead (Rep_SYS & "\GENLANG")
End If

```

```

If Not FSO.FileExists( Rep_AJOURNAL & "\QJ") Then
Call Msg_Log (Array("1022" , "PCMINI"))

```

```

'-----
WshEnv("PAC7QJ") = Rep_AJOURNAL & "\QJ"
Call RunCmdLog ("BVPCMINI")

```

```

Call Err_Cod(Return , 0 , "PCMINI")
End if

```

```

If Not FSO.FileExists( Rep_ABASE & "\GK") Then
Call Msg_Log (Array("1022" , "PTUIGK"))

```

```

'-----
WshEnv("PACGGK") = Rep_ABASE & "\GK"
Call RunCmdLog ("BVPTUIGK")

```

```

Call Err_Cod(Return , 0 , "PTUIGK")
End if

```

```

Call Msg_Log (Array("1022" , "PACB"))

```

```

'-----
WshEnv("PAC7QJ") = Rep_AJOURNAL & "\QJ"
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7LB") = Rep_BASE & "\LB"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGK") = Rep_ABASE & "\GK"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7GS") = Rep_BASE & "\GS"
WshEnv("PAC7SC") = Rep_SKEL & "\SC" & CodLang
WshEnv("PAC7SF") = Rep_SKEL & "\SF" & CodLang
WshEnv("PAC7SG") = Rep_SKEL & "\SG" & CodLang
WshEnv("PAC7SN") = Rep_SKEL & "\SN" & CodLang
WshEnv("PAC7SR") = Rep_SKEL & "\SR" & CodLang
WshEnv("PAC7SS") = Rep_SKEL & "\SS" & CodLang
WshEnv("PAC7SL") = Rep_SKEL & "\SL" & CodLang
WshEnv("PAC7ME") = Fic_Input
Call BvpEnv("PACB", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("PACB", "PAC7DG", Rep_TMP & "\WDG.tmp")
Call BvpEnv("PACB", "PAC7EB", Rep_TMP & "\WEB.tmp")
Call BvpEnv("PACB", "PAC7EE", Rep_TMP & "\WEE.tmp")
Call BvpEnv("PACB", "PAC7EG", Rep_TMP & "\WEG.tmp")

```

```

Call BvpEnv("PACB", "PAC7EI", Rep_TMP & "\\WEI.tmp")
Call BvpEnv("PACB", "PAC7EN", Rep_TMP & "\\WEN.tmp")
Call BvpEnv("PACB", "PAC7EP", Rep_TMP & "\\WEP.tmp")
Call BvpEnv("PACB", "PAC7EQ", Rep_TMP & "\\WEQ.tmp")
Call BvpEnv("PACB", "PAC7ER", Rep_TMP & "\\WER.tmp")
Call BvpEnv("PACB", "PAC7EV", Rep_TMP & "\\WEV.tmp")
Call BvpEnv("PACB", "PAC7EW", Rep_TMP & "\\WEW.tmp")
Call BvpEnv("PACB", "PAC7OB", Rep_USR & "\\GPRCOB.txt")
Call BvpEnv("PACB", "PAC7GB", Rep_TMP & "\\GPRCGB.tmp")
Call BvpEnv("PACB", "PAC7OD", Rep_USR & "\\GPRCOD.txt")
Call BvpEnv("PACB", "PAC7GD", Rep_TMP & "\\GPRCGD.tmp")
Call BvpEnv("PACB", "PAC7OE", Rep_USR & "\\GPRCOE.txt")
Call BvpEnv("PACB", "PAC7GE", Rep_TMP & "\\GPRCGE.tmp")
Call BvpEnv("PACB", "PAC7OF", Rep_USR & "\\GPRCOF.txt")
Call BvpEnv("PACB", "PAC7GF", Rep_TMP & "\\GPRCGF.tmp")
Call BvpEnv("PACB", "PAC7OG", Rep_USR & "\\GPRCOG.txt")
Call BvpEnv("PACB", "PAC7GG", Rep_TMP & "\\GPRCGG.tmp")
Call BvpEnv("PACB", "PAC7GI", Rep_USR & "\\GPRCGI.txt")

Call BvpEnv("PACB", "PAC7GK", Rep_USR & "\\ERRGK.txt")
Call BvpEnv("PACB", "PAC7GL", Rep_USR & "\\ERRGL.txt")

Call BvpEnv("PACB", "PAC7GN", Rep_TMP & "\\WXGN.tmp")
Call BvpEnv("PACB", "PAC7GO", Rep_TMP & "\\WGO.tmp")
Call BvpEnv("PACB", "PAC7OP", Rep_USR & "\\GPRCOP.txt")
Call BvpEnv("PACB", "PAC7GP", Rep_TMP & "\\GPRCGP.tmp")
Call BvpEnv("PACB", "PAC7OQ", Rep_USR & "\\GPRCOQ.txt")
Call BvpEnv("PACB", "PAC7GQ", Rep_TMP & "\\GPRCQG.tmp")
Call BvpEnv("PACB", "PAC7OR", Rep_USR & "\\GPRCOR.txt")
Call BvpEnv("PACB", "PAC7GR", Rep_TMP & "\\GPRCGR.tmp")
Call BvpEnv("PACB", "PAC7GT", Rep_USR & "\\PAWGT.txt")
Call BvpEnv("PACB", "PAC7OV", Rep_USR & "\\GPRCOV.txt")
Call BvpEnv("PACB", "PAC7GV", Rep_TMP & "\\GPRCGV.tmp")
Call BvpEnv("PACB", "PAC7G6", Rep_USR & "\\GPRCG6.txt")
Call BvpEnv("PACB", "PAC7DB", Rep_USR & "\\GPRCDB.txt")
Call BvpEnv("PACB", "PAC7IA", Rep_USR & "\\GPRCIA.txt")
Call BvpEnv("PACB", "PAC7ID", Rep_USR & "\\GPRCID.txt")
Call BvpEnv("PACB", "PAC7IK", Rep_USR & "\\GPRCIK.txt")
Call BvpEnv("PACB", "PAC7IL", Rep_USR & "\\GPRCIL.txt")
Call BvpEnv("PACB", "PAC7IM", Rep_USR & "\\GPRCIM.txt")
Call BvpEnv("PACB", "PAC7IN", Rep_USR & "\\GPRCIN.txt")
Call BvpEnv("PACB", "PAC7IW", Rep_USR & "\\GPRCIW.txt")
Call BvpEnv("PACB", "PAC7JG", Rep_TMP & "\\WJG.tmp")
Call BvpEnv("PACB", "PAC7KB", Rep_TMP & "\\WKB.tmp")
Call BvpEnv("PACB", "PAC7KD", Rep_TMP & "\\WKD.tmp")
Call BvpEnv("PACB", "PAC7KE", Rep_TMP & "\\WKE.tmp")
Call BvpEnv("PACB", "PAC7KF", Rep_TMP & "\\WKF.tmp")
Call BvpEnv("PACB", "PAC7KG", Rep_TMP & "\\WKG.tmp")
Call BvpEnv("PACB", "PAC7KM", Rep_TMP & "\\WKM.tmp")
Call BvpEnv("PACB", "PAC7KN", Rep_TMP & "\\WKN.tmp")
Call BvpEnv("PACB", "PAC7KP", Rep_TMP & "\\WKP.tmp")
Call BvpEnv("PACB", "PAC7KQ", Rep_TMP & "\\WKQ.tmp")
Call BvpEnv("PACB", "PAC7KR", Rep_TMP & "\\WKR.tmp")
Call BvpEnv("PACB", "PAC7KS", Rep_TMP & "\\WKS.tmp")
Call BvpEnv("PACB", "PAC7KU", Rep_TMP & "\\WKU.tmp")

```

```

Call BvpEnv("PACB","PAC7KV",Rep_TMP & "\WKV.tmp")
Call BvpEnv("PACB","PAC7LG",Rep_TMP & "\NUL.tmp")
'PAC7LG not used, on default
Call BvpEnv("PACB","PAC7LI",Rep_TMP & "\WLI.tmp")
Call BvpEnv("PACB","PAC7LK",Rep_TMP & "\NUL.tmp")
'PAC7LK not used, on default
Call BvpEnv("PACB","PAC7MG",Rep_TMP & "\WMG.tmp")
Call BvpEnv("PACB","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACB","PAC7SO",Rep_TMP & "\WSO.tmp")
Call BvpEnv("PACB","PAC7WA",Rep_TMP & "\WWA.tmp")
Call BvpEnv("PACB","PAC7WC",Rep_TMP & "\WWC.tmp")
Call BvpEnv("PACB","PAC7W1",Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACB","PAC7W2",Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACB","PAC7W3",Rep_TMP & "\WW3.tmp")
Call BvpEnv("PACB","PAC7W4",Rep_TMP & "\WW4.tmp")
Call BvpEnv("PACB","PAC7W6",Rep_TMP & "\WW6.tmp")
Call BvpEnv("PACB","PAC7W7",Rep_TMP & "\WW7.tmp")
Call BvpEnv("PACB","PAC7W8",Rep_TMP & "\WW8.tmp")
Call BvpEnv("PACB","PAC7W9",Rep_TMP & "\WW9.tmp")
Call BvpEnv("PACB","SYSPAF",Rep_TMP & "\WPAF.tmp")
Call RunCmdLog ("BVPACB")

If Return < 10 then
Call Msg_Log (Array("1062"))
Return = 0
End if
If Return = 10 then
Call Msg_Log (Array("1063"))
Return = 0
End if

If Return > 10 then
Call Msg_Log (Array("1064"))
End if
Call Err_Cod(Return , 10 , "PACB")

Call Msg_Log (Array("1022" , "COPY in OM"))
'-----
If Not FS0.FileExists(Rep_TMP & "\GPRCOM.txt") Then
Set LogGen = FS0.CreateTextFile(Rep_TMP & "\GPRCOM.txt", TRUE)
LogGen.Close
End if
OM = FS0.GetFile(Rep_TMP & "\GPRCOM.txt")

Call CopMFile(OM , WshEnv("PAC70P") ,OM )
Call DelFile (WshEnv("PAC70P"))
Call CopMFile(OM , WshEnv("PAC70Q") ,OM )
Call DelFile (WshEnv("PAC70Q"))
Call CopMFile(OM , WshEnv("PAC70E") ,OM )
Call DelFile (WshEnv("PAC70E"))
Call CopMFile(OM , WshEnv("PAC70R") ,OM )
Call DelFile (WshEnv("PAC70R"))
Call CopMFile(OM , WshEnv("PAC70G") ,OM )
Call DelFile (WshEnv("PAC70G"))
Call CopMFile(OM , WshEnv("PAC70V") ,OM )

```

```

Call DelFile (WshEnv("PAC70V"))
Call CopMFi1(OM , WshEnv("PAC70D") ,OM )
Call DelFile (WshEnv("PAC70D"))
Call CopMFi1(OM , WshEnv("PAC70F") ,OM )
Call DelFile (WshEnv("PAC70F"))
Call CopMFi1(OM , WshEnv("PAC70B") ,OM )
Call DelFile (WshEnv("PAC70B"))

Call Msg_Log (Array("1022" , "PAPG1S"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PACB","GENERE",Rep_TMP & "\GPRCOM.txt")
Call BvpEnv("PACB","PAC7W1",Rep_TMP & "\WW1.tmp")
Call RunCmdLog ("BVPAPG1S")
Call Err_Cod(Return , 0 , "PAPG1S")

Call Msg_Log (Array("1022" , "PAPG5S"))
'-----
Call BvpEnv("PACB","PAC7W1",Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACB","PAC7W2",Rep_TMP & "\WW2.tmp")
Call RunCmdLog ("BVPAPG5S")
Call Err_Cod(Return , 0 , "PAPG5S")

Call Msg_Log (Array("1022" , "PAPG7S"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PACB","PAC7EW",Rep_TMP & "\WEW.tmp")
Call BvpEnv("PACB","PAC7W3",Rep_TMP & "\WW3.tmp")
Call BvpEnv("PACB","PAC7W2",Rep_TMP & "\WW2.tmp")
WshEnv("PAC7SA") = Rep_SKEL & "\SA" & CodLang
Call RunCmdLog ("BVPAPG7S")
Call Err_Cod(Return , 0 , "PAPG7S")

Call Msg_Log (Array("1022" , "PAPG9S"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PACB","PAC7EW",Rep_TMP & "\WEW.tmp")
Call BvpEnv("PACB","PAC7W3",Rep_TMP & "\WW3.tmp")
Call BvpEnv("PACB","GENERE",Rep_TMP & "\GPRCOM.txt")
Call BvpEnv("PACB","COB80",Rep_USR & "\GPRCOM.txt")
Call BvpEnv("PACB","PAC7ED",Rep_USR & "\GPRCED.txt")
Call RunCmdLog ("BVPAPG9S")
Call Err_Cod(Return , 0 , "PAPG9S")

If BVPACAGP <> " " then
    Call Msg_Log (Array("1022" , BVPACAGP))
'-----
Return = WshShell.Run(BVPACAGP , 1 , TRUE )
    Call Err_Cod(Return , 0 , BVPACAGP)

```

```

End If

If BVP_Updtpm = "YES" then
  Call Msg_Log (Array("1022" , "PCMPUF"))
  '-----
  WshEnv("PAC7AE") = Rep_SKEL & "\AE"
  WshEnv("SEMLOCK") = Rep_BASE & "\LO"
  WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
  WshEnv("PAC7IC") = Rep_TMP & "\NUL.tmp"
  'PAC7IC not used, on default
  WshEnv("PAC7QJ") = Rep_AJOURNAL & "\QJ"
  WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
  WshEnv("PAC7AN") = Rep_BASE & "\AN"
  WshEnv("PAC7AR") = Rep_BASE & "\AR"
  WshEnv("PAC7AY") = Rep_BASE & "\AY"
  WshEnv("PACGGN") = Rep_ABASE & "\AN"
  WshEnv("PACGGR") = Rep_ABASE & "\AR"
  WshEnv("PACGGU") = Rep_ABASE & "\GU"
  Call RunCmdLog ("BVPMPUF")
  Call Err_Cod(Return , 0 , "PCMPUF")
end if

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr(Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

GPMC - Gestion de l'opérateur MCI

GPMC - Présentation générale

Génération et Move Corresponding

L'opérateur MCI saisi sous VA Pacbase n'est pas interprété par le générateur PACB, mais plus tard par les deux programmes UTIMCR et UTIMCI qui travaillent sur le COBOL en sortie du générateur.

Règles de base :

L'ordre MOVE CORRESPONDING concerne deux zones groupe : le premier opérande doit être saisi sur la même ligne que l'opérateur, le second à la suite ou sur la ligne suivante.

Chaque zone dépendante du premier groupe doit avoir son équivalent dans le second pour apparaître dans le MOVE.

La comparaison entre les zones COBOL est faite uniquement sur leur code "rubrique", c'est à dire la chaîne de caractères suivant le premier tiret (qui lui, est précédé d'un préfixe).

Par exemple, dans "PREFIXE-ZONE-NUMERO-UN" toute la chaîne "ZONE-NUMERO-UN" est recherchée dans la décomposition de l'autre groupe pour la faire éventuellement intervenir dans le MOVE CORRESPONDING.

Lorsqu'un groupe est suivi d'un indice, toutes ses zones générées le sont de la même façon.

Aucun contrôle n'étant effectué sur la pertinence des "MOVE" ainsi générés, c'est le compilateur COBOL qui détectera les éventuelles erreurs.

Le fichier WKMCI, écrit par UTIMCR, répertorie les lignes des MCI détectés dans le COBOL analysé (une ou deux lignes par ordre selon la saisie de l'utilisateur) et est lu par le programme UTIMCI. Le fichier COB80 contenant le COBOL issu du générateur est lu par les deux programmes. Le fichier final (MCI80) est à l'image du fichier COB80 et ajoute, dans le COBOL d'origine, les lignes MCI recopiées en commentaires suivis des ordres MOVE en découlant.

GPMC - Entrées Utilisateur

Se reporter à la description des entrées utilisateurs GPRT.

GPMC - Description des étapes

Editions et générations : PACB

Le source généré fourni dépend des commandes d'édition-génération prises en compte.

Pour plus de renseignements, se reporter à la description faite pour GPRT.

Générateur MCI : UTIMCR

Code	Nom physique	Type	Libellé
MCI80	Rép. user : GPMCOM	Entrée	Source généré GPRT
WKMCI	Rép. tmp : WWK	Sortie	Fichier de travail

Codes retour :

- 4 : Aucun ordre MCI n'a été détecté et le traitement s'arrête

Générateur MCI : UTIMCI

Code	Nom physique	Type	Libellé
MCI80	Rép. user : GPMCOM	Entrée	Source généré GPRT
WKMCI	Rép. tmp : WWK	Entrée	Fichier de travail
COB80	Rép. user : COB80	Sortie	Source résultat

Code retour :

- 8 : Source ou Cible d'un opérateur MCI inexistante

GPMC - Script d'exécution

```
'-----  
'          VISUALAGE PACBASE  
'-----  
'          - MOVE CORRESPONDING GENERATION -  
'-----  
'  
' IN ADDITION TO THE GENERATED ENTITIES, THE FILE MUST  
' CONTAIN THE JCL REQUIRED TO COMPILE THEM,  
' USING THE BEGINNING/END OF JCL JOB STREAM OPTIONS AND  
' THE BEFORE/AFTER PROGRAM OPTIONS.  
'-----  
'  
<job id=GPMC>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "GPMC"  
</script>  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Dim RetPacb  
Dim CodLang  
If base = "ADMIN" Then  
Call Msg_Log (Array("1028",base))  
Wscript.Quit (0)  
Else  
  CodLang = WshShell.RegRead (Rep_SYS & "\GENLANG")  
End If  
  
If Not FSO.FileExists( Rep_AJOURNAL & "\QJ") Then  
Call Msg_Log (Array("1022", "PCMINI"))  
'-----  
WshEnv("PAC7QJ") = Rep_AJOURNAL & "\QJ"
```

```

Call RunCmdLog ("BVPCMINI")

Call Err_Cod(Return , 0 , "PCMINI")
End if

If Not FSO.FileExists( Rep_ABASE & "\GK") Then
Call Msg_Log (Array("1022" , "PTUIGK"))
'-----
WshEnv("PACGGK") = Rep_ABASE & "\GK"
Call RunCmdLog ("BVPTUIGK")

Call Err_Cod(Return , 0 , "PTUIGK")
End if

Call Msg_Log (Array("1022" , "PACB"))
'-----
WshEnv("PAC7QJ") = Rep_AJOURNAL & "\QJ"
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7LB") = Rep_BASE & "\LB"
WshEnv("PACGGK") = Rep_ABASE & "\GK"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7GS") = Rep_BASE & "\GS"
WshEnv("PAC7SC") = Rep_SKEL & "\SC" & CodLang
WshEnv("PAC7SF") = Rep_SKEL & "\SF" & CodLang
WshEnv("PAC7SG") = Rep_SKEL & "\SG" & CodLang
WshEnv("PAC7SN") = Rep_SKEL & "\SN" & CodLang
WshEnv("PAC7SR") = Rep_SKEL & "\SR" & CodLang
WshEnv("PAC7SS") = Rep_SKEL & "\SS" & CodLang
WshEnv("PAC7SL") = Rep_SKEL & "\SL" & CodLang
WshEnv("PAC7ME") = Fic_Input
Call BvpEnv("PACB", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("PACB", "PAC7DG", Rep_TMP & "\WDG.tmp")
Call BvpEnv("PACB", "PAC7EB", Rep_TMP & "\WEB.tmp")
Call BvpEnv("PACB", "PAC7EE", Rep_TMP & "\WEE.tmp")
Call BvpEnv("PACB", "PAC7EG", Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACB", "PAC7EI", Rep_TMP & "\WEI.tmp")
Call BvpEnv("PACB", "PAC7EN", Rep_TMP & "\WEN.tmp")
Call BvpEnv("PACB", "PAC7EP", Rep_TMP & "\WEP.tmp")
Call BvpEnv("PACB", "PAC7EQ", Rep_TMP & "\WEQ.tmp")
Call BvpEnv("PACB", "PAC7ER", Rep_TMP & "\WER.tmp")
Call BvpEnv("PACB", "PAC7EV", Rep_TMP & "\WEV.tmp")
Call BvpEnv("PACB", "PAC7EW", Rep_TMP & "\WEW.tmp")
Call BvpEnv("PACB", "PAC7OB", Rep_USR & "\GPRTOB.txt")
Call BvpEnv("PACB", "PAC7GB", Rep_TMP & "\GPRTGB.tmp")
Call BvpEnv("PACB", "PAC7OD", Rep_USR & "\GPRTOD.txt")
Call BvpEnv("PACB", "PAC7GD", Rep_TMP & "\GPRTGD.tmp")
Call BvpEnv("PACB", "PAC7OE", Rep_USR & "\GPRTOE.txt")
Call BvpEnv("PACB", "PAC7GE", Rep_TMP & "\GPRTGE.tmp")
Call BvpEnv("PACB", "PAC7OF", Rep_USR & "\GPRTOF.txt")

```

```

Call BvpEnv("PACB","PAC7GF",Rep_TMP & "\GPRTGF.tmp")
Call BvpEnv("PACB","PAC7OG",Rep_USR & "\GPRTOG.txt")
Call BvpEnv("PACB","PAC7GG",Rep_TMP & "\GPRTGG.tmp")
Call BvpEnv("PACB","PAC7GI",Rep_USR & "\GPRTGI.txt")
Call BvpEnv("PACB","PAC7GK",Rep_USR & "\ERRGK.txt")
Call BvpEnv("PACB","PAC7GL",Rep_USR & "\ERRGL.txt")
Call BvpEnv("PACB","PAC7GM",Rep_USR & "\ERRGM.txt")
Call BvpEnv("PACB","PAC7GN",Rep_TMP & "\WXGN.tmp")
Call BvpEnv("PACB","PAC7GO",Rep_TMP & "\WGO.tmp")
Call BvpEnv("PACB","PAC7OP",Rep_USR & "\GPRTOP.txt")
Call BvpEnv("PACB","PAC7GP",Rep_TMP & "\GPRTGP.tmp")
Call BvpEnv("PACB","PAC7OQ",Rep_USR & "\GPRTOQ.txt")
Call BvpEnv("PACB","PAC7GQ",Rep_TMP & "\GPRTGQ.tmp")
Call BvpEnv("PACB","PAC7OR",Rep_USR & "\GPRTOR.txt")
Call BvpEnv("PACB","PAC7GR",Rep_TMP & "\GPRTGR.tmp")
Call BvpEnv("PACB","PAC7GT",Rep_USR & "\PAWGT.txt")
Call BvpEnv("PACB","PAC7OV",Rep_USR & "\GPRTOV.txt")
Call BvpEnv("PACB","PAC7GV",Rep_TMP & "\GPRTGV.tmp")
Call BvpEnv("PACB","PAC7G6",Rep_USR & "\GPRTG6.txt")
Call BvpEnv("PACB","PAC7DB",Rep_USR & "\GPRTDB.txt")
Call BvpEnv("PACB","PAC7IA",Rep_USR & "\GPRTIA.txt")
Call BvpEnv("PACB","PAC7ID",Rep_USR & "\GPRTID.txt")
Call BvpEnv("PACB","PAC7IK",Rep_USR & "\GPRTIK.txt")
Call BvpEnv("PACB","PAC7IL",Rep_USR & "\GPRTIL.txt")
Call BvpEnv("PACB","PAC7IM",Rep_USR & "\GPRTIM.txt")
Call BvpEnv("PACB","PAC7IN",Rep_USR & "\GPRTIN.txt")
Call BvpEnv("PACB","PAC7IO",Rep_USR & "\GPRTIO.txt")
Call BvpEnv("PACB","PAC7IW",Rep_USR & "\GPRTIW.txt")
Call BvpEnv("PACB","PAC7JG",Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACB","PAC7KB",Rep_TMP & "\WKB.tmp")
Call BvpEnv("PACB","PAC7KD",Rep_TMP & "\WKD.tmp")
Call BvpEnv("PACB","PAC7KE",Rep_TMP & "\WKE.tmp")
Call BvpEnv("PACB","PAC7KF",Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACB","PAC7KG",Rep_TMP & "\WKG.tmp")
Call BvpEnv("PACB","PAC7KM",Rep_TMP & "\WKM.tmp")
Call BvpEnv("PACB","PAC7KN",Rep_TMP & "\WKN.tmp")
Call BvpEnv("PACB","PAC7KP",Rep_TMP & "\WKP.tmp")
Call BvpEnv("PACB","PAC7KQ",Rep_TMP & "\WKQ.tmp")
Call BvpEnv("PACB","PAC7KR",Rep_TMP & "\WKR.tmp")
Call BvpEnv("PACB","PAC7KS",Rep_TMP & "\WKS.tmp")
Call BvpEnv("PACB","PAC7KU",Rep_TMP & "\WКУ.tmp")
Call BvpEnv("PACB","PAC7KV",Rep_TMP & "\WKV.tmp")
Call BvpEnv("PACB","PAC7LG",Rep_TMP & "\NUL.tmp")
'PAC7LG not used, on default
Call BvpEnv("PACB","PAC7LI",Rep_TMP & "\WLI.tmp")
Call BvpEnv("PACB","PAC7LK",Rep_TMP & "\NUL.tmp")
'PAC7LK not used, on default
Call BvpEnv("PACB","PAC7LM",Rep_TMP & "\NUL.tmp")
'PAC7LM not used, on default
Call BvpEnv("PACB","PAC7MG",Rep_TMP & "\WМG.tmp")
Call BvpEnv("PACB","PAC7MV",Rep_TMP & "\WМV.tmp")
Call BvpEnv("PACB","PAC7SO",Rep_TMP & "\WSO.tmp")
Call BvpEnv("PACB","PAC7WA",Rep_TMP & "\WWA.tmp")
Call BvpEnv("PACB","PAC7WC",Rep_TMP & "\WWC.tmp")
Call BvpEnv("PACB","PAC7W1",Rep_TMP & "\WW1.tmp")

```

```

Call BvpEnv("PACB","PAC7W2",Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACB","PAC7W3",Rep_TMP & "\WW3.tmp")
Call BvpEnv("PACB","PAC7W4",Rep_TMP & "\WW4.tmp")
Call BvpEnv("PACB","PAC7W6",Rep_TMP & "\WW6.tmp")
Call BvpEnv("PACB","PAC7W7",Rep_TMP & "\WW7.tmp")
Call BvpEnv("PACB","PAC7W8",Rep_TMP & "\WW8.tmp")
Call BvpEnv("PACB","PAC7W9",Rep_TMP & "\WW9.tmp")
Call BvpEnv("PACB","SYSPAF",Rep_TMP & "\WPAF.tmp")
Call RunCmdLog ("BVPACB")
RetPacb = Return

If Return < 10 then
Call Msg_Log (Array("1062"))
Return = 0
End if

If Return = 10 then
Call Msg_Log (Array("1063"))
Return = 0
End if

If Return > 10 then
Call Msg_Log (Array("1064"))
End if

Call Err_Cod(Return , 10 , "PACB")

If RetPacb < 10 Then
  Call Msg_Log (Array("1022" , "COPY in OM"))
  '-----
  If Not FS0.FileExists(Rep_USR & "\GPRTOM.txt") Then
    Set LogGen = FS0.CreateTextFile(Rep_USR & "\GPRTOM.txt", TRUE)
    LogGen.Close
  End if
  OM = FS0.GetFile(Rep_USR & "\GPRTOM.txt")

  Call CopMFiL(OM , WshEnv("PAC70P") ,OM )
  Call DelFile (WshEnv("PAC70P"))
  Call CopMFiL(OM , WshEnv("PAC70Q") ,OM )
  Call DelFile (WshEnv("PAC70Q"))
  Call CopMFiL(OM , WshEnv("PAC70E") ,OM )
  Call DelFile (WshEnv("PAC70E"))
  Call CopMFiL(OM , WshEnv("PAC70R") ,OM )
  Call DelFile (WshEnv("PAC70R"))
  Call CopMFiL(OM , WshEnv("PAC70G") ,OM )
  Call DelFile (WshEnv("PAC70G"))
  Call CopMFiL(OM , WshEnv("PAC70V") ,OM )
  Call DelFile (WshEnv("PAC70V"))
  Call CopMFiL(OM , WshEnv("PAC70D") ,OM )
  Call DelFile (WshEnv("PAC70D"))
  Call CopMFiL(OM , WshEnv("PAC70F") ,OM )
  Call DelFile (WshEnv("PAC70F"))
  Call CopMFiL(OM , WshEnv("PAC70B") ,OM )
  Call DelFile (WshEnv("PAC70B"))

```

```

Set OM = FSO.GetFile(Rep_USR & "\GPRTOM.txt")
FilFull = OM.size

If FilFull = 0 then DelFile(OM) end if

Call Msg_Log (Array("1022" , "UTIMCR"))
'-----
Call BvpEnv("UTIMCR","MCI80",Rep_USR & "\GPRTOM.txt")
Call BvpEnv("UTIMCR","WKMC I",Rep_TMP & "\WWK.tmp")
Call RunCmdLog ("BVPUTMCR")

Call Err_Cod(Return , 8 , "UTIMCR")

If Return = 8 Then
'No MCI found
'-----
Call Msg_Log (Array("1067"))
end if

If Return = 4 Then

    Ret = fso.MoveFile (Rep_USR & "\GPRTOM.txt" , _
    Rep_TMP & "\GPRTOM.txt")
    If Ret <> 0 Then
        Call Msg_Log (Array("1025", "MOVE-OM", Ret))
        Wscript.Quit (0)
    End If

Call Msg_Log (Array("1022" , "UTIMCI"))
'-----
Call BvpEnv("UTIMCI","COB80",Rep_USR & "\GPRTOM.txt")
Call BvpEnv("UTIMCI","MCI80",Rep_TMP & "\GPRTOM.txt")
Call BvpEnv("UTIMCI","WKMC I",Rep_TMP & "\WWK.tmp")
Call RunCmdLog ("BVPUTMCI")

Call Err_Cod(Return , 0 , "UTIMCI")
End If
End If

If BVPACAGP <> " " then
    Call Msg_Log (Array("1022" , BVPACAGP))
'-----
Return = WshShell.Run(BVPACAGP , 1 , TRUE )
    Call Err_Cod(Return , 0 , BVPACAGP)
End If

If BVP_Updtpm = "YES" then
    Call Msg_Log (Array("1022" , "PCMPUF"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7IC") = Rep_TMP & "\NUL.tmp"
'PAC7IC not used, on default
WshEnv("PAC7QJ") = Rep_AJOURNAL & "\QJ"

```

```

WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call RunCmdLog ("BVPCMPUF")
Call Err_Cod(Return , 0 , "PCMPUF")
end if
Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr(Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Chapitre 3. Extractions

PACX - Présentation générale

Cette procédure permet d'effectuer différents types d'extractions de données de la Base de Développement via l'utilisation d'un extracteur PAF (sélection de critères).

Voir le chapitre "UPDP - Mise à jour à partir de Tables PAF" dans le manuel "Les Procédures du Développeur".

Ces données sont extraites sous forme de mouvements qui pourront être utilisés en entrée des procédures suivantes :

- UPDT
- UPDP
- CPSN (si l'utilitaire optionnel "Gestionnaire de Bases d'information réparties" a été acquis).

Condition d'exécution

Aucune puisque le réseau n'est pas directement mis à jour.

PACX - Entrées communes aux extracteurs

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	bbb	Code bibliothèque d'extraction ou code bibliothèque destinataire si RMEN avec remontée
22	4	nnnn	Numéro de session (blanc=courante)
26	1	'T'	Etat de la session si session historisée
29	4	cccc	Code de l'extracteur (1)
33	1	'1'	Formatage pour UPDT
		'2'	CPSN : formatage pour UPDT avec codes mouvements explicites
		' '	Pas de formatage pour UPDT
34	1	'1'	Formatage pour UPDP (PAF)

Pos.	Lon.	Valeur	Signification
		'2'	CPSN : formatage pour UPDP (PAF) avec codes mouvements explicites
		' '	Pas de formatage pour UPDP (PAF)
35	1	'1'	Formatage pour CPSN
		' '	Pas de formatage pour CPSN
40	3	ppp	Code produit DSMS
43	6	nnnnnn	Numéro d'amélioration DSMS (module DSMS seulement)
49	1		Gestion des verrous
		' '	Extraction des verrous avec code utilisateur = code utilisateur de la ligne '*'
		'1'	Pas d'extraction de verrous
		'2'	Extraction des verrous avec code utilisateur = code utilisateur d'origine
		'N'	Pour RMEN seulement : non extraction des entités verrouillées par un autre utilisateur
50	1	' '	Pas de report du mot de passe
		'1'	Report du mot de passe
69	3	bbb	Code bibliothèque pour la ligne '*' du ou des fichiers en sortie (uniquement pour EXTR, EXLI, EXUE)
76	5	nnnnT	Session pour la ligne '*' du ou des fichiers en sortie (uniquement pour EXTR, EXLI, EXUE)

Les différentes valeurs du code extracteur sont :

- EXTR : extraction d'entités avec mouvements extraits triés.
- EXTA : extraction d'entités avec mouvements extraits triés, mais respectant l'ordre des lignes d'identification de l'entrée. Ainsi, si chaque demande est précédée d'une ligne '*', les mouvements extraits seront triés dans l'ordre des demandes. Le formatage est forcé à UPDT.
- EXUE : extraction des Entités utilisateur.
- EXPJ : extraction du journal (le formatage pour CPSN n'est pas possible).
- EXLI : extraction de bibliothèques ou de sous-réseaux de bibliothèques (formatage pour UPDP, UPDT ou CPSN).
- EXPU : extraction pour purge (le formatage pour CPSN n'est pas possible).
- RMEN : extraction d'entités pour remontée/remplacement/recodification (le formatage pour CPSN n'est pas possible). RMEN est soumis à un contrat d'acquisition.
- CPSN : comparaison de sous-réseaux ou d'entités.

Attention

- Un seul type d'extracteur par lancement : si la procédure détecte plusieurs types d'extracteur, elle ne prendra en compte que le premier rencontré.
- Seul le type de formatage de la première carte est pris en compte.
- Formatage pour CPSN : cette procédure fait partie de l'utilitaire optionnel "Gestionnaire de bases d'informations réparties". A ce titre, son utilisation est soumise à un contrat d'acquisition.
- Nombre maximum de lignes '*' en entrée : 1 pour RMEN et EXPJ, 1000 pour EXTR, EXTA, EXUE et EXPU.
- Les verrous techniques de Pacdesign/Pacbench ne sont pas extraits par PACX.

Edition obtenue

Cette procédure édite :

- Un compte-rendu avec la liste des programmes exécutés et le nombre de mouvements générés.
- La liste des demandes avec indication des erreurs éventuelles.
- Un ou plusieurs comptes-rendus selon l'extracteur.

EXTR/EXTA - Extraction d'entités

EXTR/EXTA - Présentation générale

Ces types d'extracteur permettent la sélection de tout ou partie d'entité.

La demande de type 'ALL' extrait l'entité dans sa totalité, soit l'entité proprement dite, plus les entités utilisées, plus les entités utilisées par les entités utilisées etc. Les entités utilisées non chaînées ne sont pas extraites.

En fonction du formatage demandé, le fichier obtenu pourra être mis en entrée des procédures UPDT, UPDP ou CPSN (si la demande est de type 'ALL', 'ONLY' ou 'EXPT' le formatage pour CPSN est interdit).

Pour EXTA le formatage est forcé à UPDT.

EXTR/EXTA - Entrées Utilisateur

Une ou deux lignes de commande par entité à extraire :

Première ligne :

Pos.	Lon.	Valeur	Signification
2	1	'W'	Code ligne
3	1	'1'	Numéro de ligne

Pos.	Lon.	Valeur	Signification
4	2	'EX'	
6	1		Code de sélection de bibliothèque :
		'U'	Bibliothèque seule
		'C'	Bibliothèque et ses 'centrales'
		'+'	Bibliothèque et ses 'centrales' avec génération des lignes d'identification (lignes '*')
7	33	Choix	Entité à extraire codifiée de la même manière que la zone 'Choix' en conversationnel
40	4		Type d'extraction (1) :
		' '	Entité seulement (obligatoire pour EXTA)
		'ALL '	Entité et entités utilisées
		'ONLY'	Entité et seulement les entités utilisées dont le type est mentionné dans la suite de la ligne
		'EXPT'	Entité et entités utilisées sauf celles dont le type est mentionné dans la suite de la ligne
44			Table de 12 postes de 3 caractères contenant les exceptions ou les sélections :
			'RUB' : Rubrique
			'DBD' : Bloc Base de Données
			'SDO' : Structure de Données
			'SEG' : Segment
			'ETA' : Etat
			'TXT' : Texte
			'RAP' : Document
			'PGM' : Programme
			'DIA' : Dialogue
			'ECR' : Ecran
			'FOG' : Format Guide
			'MET' : Méthode
			'MEC' : Méta-Entité Cliente
			'RLC' : Relation Utilisateur Cliente
			'\$tt' : Entité Utilisateur Cliente (tt = code d'appel)
			'MEE' : Méta-Entité Extension
			'RLE' : Relation Utilisat. Extension
			'Ytt' : Entité Utilisat. Extension (tt = code d'appel (1))

(1) Les valeurs ALL ONLY EXPT ne sont utilisables que si le module optionnel 'Extracteur profond' a été acquis.

Deuxième ligne (ligne suite pour sélections et exceptions) :

Pos.	Lon.	Valeur	Signification
2	1	'W'	Code ligne
3	1	'2'	Numéro de ligne
44			Table de 12 postes de 3 caractères contenant les exceptions ou les sélections.

(1) Les valeurs des codes d'appels sont :

CE	Composant élémentaire
CS	Application eBusiness
C1	Interface SCM
DO	Dossier
D1	Document Publishing
D2	Type de document
F1	Fichier externe
G1	Commande édition génération
MC	Moniteur de communication
MS	Message
OP	Opération
PT	Part
SB	SOAP Binding
SI	Initialisation serveur
SV	Service
UM	Interface UML
VL	Vue logique
5Q	Définition règle qualité
7E	Plan type d'extraction

La procédure EXTR fonctionne également pour les choix propres à la Base de développement ; il suffit de les entrer à partir de la 7ème position de la façon suivante :

//A_CCCXXXXXX

où A représente le code méthode et CCC le code local de l'entité.

Type d'extraction

- L'utilisation de l'option 'extracteur profond' (type d'extraction 'ALL', 'EXPT' ou 'ONLY') ne fonctionne pas pour l'extracteur EXTA pour lequel le type d'extraction doit être à blanc.
- L'extraction d'une structure de données extrait par défaut ses segments. Si l'on ne souhaite pas l'extraction de ceux-ci, il faut renseigner le type d'extraction à la valeur 'EXPTSEG'. Ceci est possible même si l'option 'extracteur profond' n'a pas été acquise.
- L'extraction d'un dialogue n'extrait par défaut que le dialogue. Pour extraire également les écrans du dialogue, il faut renseigner le type d'extraction à la valeur 'ALL'.
- Idem pour une méta-entité et ses entités utilisateur.
- L'extraction s'arrête au premier niveau de sélection ou d'exclusion. Par exemple si l'on extrait un programme avec le type d'extraction 'EXPTSEG', les rubriques utilisées par les segments utilisés par le programme ne seront pas extraites puisque l'extracteur n'aura pas considéré ces segments.

Edition obtenue

Cette procédure édite la liste des entités extraites :

- triées pour EXTR,
- dans l'ordre des demandes pour EXTA.

EXUE - Extracteur de contenu d'Entités Utilisateur

EXUE - Présentation générale

La procédure EXUE extrait les contenus des entités utilisateur sélectionnées par le type de la méta-entité sous la forme d'enregistrements simples sur un fichier séquentiel.

Cette procédure fait partie de l'utilitaire optionnel "Module de personnalisation". A ce titre, son utilisation est soumise à un contrat d'acquisition.

Voir le manuel "Personnalisation".

EXUE - Entrées Utilisateur

Une ligne commande par entité utilisateur :

Pos.	Lon.	Valeur	Signification
2	4	'W1EX'	Code ligne
6	1	'\$'	Identifiant extraction d'EU cliente
		'Y'	Identifiant extraction d'EU extension
7	1		Code de sélection de bibliothèque
		'U'	Bibliothèque seule
		'C'	Bibliothèque et ses 'centrales'
8	2	cc	Code d'appel de la méta-entité

Edition obtenue

Cette procédure édite la liste des E.U. extraites.

Résultat obtenu

Le résultat obtenu est un fichier séquentiel de format fixe où sont enregistrés les contenus des entités sélectionnées.

Chaque enregistrement a une longueur de 230 caractères. Il comprend :

- une partie commune contenant toutes les caractéristiques nécessaires à l'identification de chaque ligne extraite,
- une partie spécifique dont le formatage dépend de la description de la méta-entité.

PACX - Description des étapes

Extraction : PACX

Cette étape extrait les mouvements en fonction des entrées utilisateur.

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AY	Rép. base : AY	Entrée	Données extension de la Base de Développement

Code	Nom physique	Type	Libellé
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7PJ	Rép. save : PJ	Entrée	Mouvements archivés
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7MA	NUL	Entrée	Fichier maître CPSN
PAC7ES	NUL	Entrée	Fichier esclave CPSN
PAC7BM	Rép. tmp : WBM	Entrée/Sortie	Entrées utilisateur
PAC7MM	Rép. tmp : WMM	Entrée/Sortie	Fichier travail EXPU
PAC7MJ	Rép. tmp : WMJ	Entrée/Sortie	Fichier travail EXPJ
PAC7TE	Rép. tmp : WTE	Entrée/Sortie	Fichier travail RMEN
PAC7RE	Rép. tmp : WRE	Entrée/Sortie	Fichier travail RMEN
PAC7RM	Rép. tmp : WRM	Entrée/Sortie	Fichier travail RMEN
PAC7WD	Rép. tmp : WWD	Entrée/Sortie	Mouvements extraits
SYSEXT	Rép. tmp : WSY	Entrée/Sortie	Fichier de travail (indexé)
PAC7RQ	NUL	Sortie	Fichier travail
PAC7MV	Rép. user : PACXMV	Sortie	Mouvements extraits pour UPDT
PAC7MR	Rép. user : PACXMR	Sortie	Mouvements extraits pour REOR (EXPU)
PAC7MX	Rép. user : PACXMX	Sortie	Entités non extraites (PACX)
PAC7GY	Rép. user : PACXGY	Sortie	Mouvements extraits pour UPDP
PAC7TD	Rép. user : PACXTD	Sortie	Mouvements extraits pour CPSN
PAC7UE	Rép. user : PACXUE	Sortie	Mouvements extraits pour EXUE
PAC7IA	Rép. user : PACXIA	Etat	Edition générale de l'enchaînement des programmes
PAC7DD	Rép. user : PACXDD	Etat	Edition des anomalies sur mouvements en entrée

Code	Nom physique	Type	Libellé
PAC7ED	Rép. user : PACXED	Etat	Compte-rendu d'extractions
PAC7EE	Rép. user : PACXEE	Etat	Compte-rendu d'extractions
PAC7EG	Rép. user : PACXEG	Etat	Compte-rendu d'extractions
PAC7EM	Rép. user : PACXEM	Etat	Compte-rendu d'extractions
PAC7EP	Rép. user : PACXEP	Etat	Compte-rendu d'extractions
PAC7EQ	Rép. user : PACXEQ	Etat	Compte-rendu d'extractions
PAC7EU	Rép. user : PACXEU	Etat	Compte-rendu d'extractions
PAC7EZ	Rép. user : PACXEZ	Etat	Compte-rendu d'extractions

Codes retour :

- 0 : Pas d'erreur
- 4 : Erreur dans les entrées utilisateur (précisée dans PAC7EE) ou dans les extractions EXTR/EXUE (précisée dans PAC7EZ)
- 8 : Erreur dans la ligne '*' (précisée dans PAC7DD) ou dans EXLI (base indisponible)

PACX - Script d'exécution

```

'-----
'          VISUALAGE PACBASE
'-----
'          - EXTRACTIONS FROM DATABASE -
'          - EXTRACTIONS COMPARATOR   -
'-----
' THE PACX PROCEDURE ALLOWS TO PERFORM VARIOUS TYPES
' OF DATA EXTRACTIONS FROM THE DEVELOPMENT DATABASE
' VIA PAF EXTRACTOR.
'
' POSSIBLE VALUES FOR THE EXTRACTOR CODE INCLUDE:
' - EXTR:  EXTRACTION OF ENTITIES
' - EXTA:  EXTRACTION OF ENTITIES (EXTRACTED TRANSACTIONS
'          ARE SORTED, ACCORDING TO THE INPUT
'          IDENTIFICATION LINES ORDER.
'          EACH REQUEST IS THUS PRECEDED BY A "*" LINE,

```

```

'          EXTRACTED TRANSACTIONS WILL BE SORTED IN THE
'          REQUEST ORDER).
' - EXUE: EXTRACTION OF USER ENTITIES
' FOLLOWING VALUES ARE RESERVED FOR THE ADMINISTRATOR:
' - EXLI:EXTRACTION OF LIBRARIES OR LIBRARY SUB-NETWORKS
' - EXPJ:EXTRACTION OF JOURNAL (FORMATTING FOR CPSN IS
'       NOT POSSIBLE)
' - EXPU:EXTRACTION OF ENTITIES TO BE PURGED
'       (FORMATTING FOR CPSN IS NOT POSSIBLE)
' - RMEN:EXTRACTION OF ENTITIES FOR UPLOAD/REPLACEMENT/
'       RECODING (FORMATTING FOR CPSN IS NOT POSSIBLE).
'       RMEN IS SUBJECT TO A SEPARATE PURCHASE AGREEMENT
' - CPSN:COMPARISON OF SUB-NETWORKS.
'
'-----
'

```

```
<job id=PACX>
```

```

<script language="VBScript">
Dim MyProc
MyProc = "PACX"
</script>

```

```
<script language="VBScript" src="INIT.vbs"/>
```

```
<script language="VBScript">
```

```
If c_error = 1 then Wscript.Quit (1) End If
```

```
Call Msg_Log (Array("1022" , "PACX"))
```

```

'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7PJ") = Rep_SAVE & "\PJ"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACX","PAC7BM",Rep_TMP & "\WBM.tmp")
Call BvpEnv("PACX","PAC7WD",Rep_TMP & "\WWD.tmp")
Call BvpEnv("PACX","PAC7MM",Rep_TMP & "\WMM.tmp")
Call BvpEnv("PACX","PAC7MJ",Rep_TMP & "\WMJ.tmp")
Call BvpEnv("PACX","PAC7TE",Rep_TMP & "\WTE.tmp")
Call BvpEnv("PACX","PAC7RE",Rep_TMP & "\WRE.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX","PAC7RQ",Rep_TMP & "\NUL.tmp")

```

```

Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\NUL.tmp")
'PAC7MA not used, on default
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\NUL.tmp")
'PAC7ES not used, on default

```

```
'Example of Output File reuse in next procedure :
```

```

' Call BvpEnv("PACX","PAC7xx",RepT_USR & "\PACXxx.txt")
'With RepT_USR is Global User Directory.

'One for each procedure : Rep_USR & "\PACXxx.txt"
'One for all the procedure : RepT_USR & "\PACXxx.txt"

'Call BvpEnv("PACX","PAC7UE",Rep_USR & "\PACXUE.txt")
Call BvpEnv("PACX","PAC7UE",RepT_USR & "\PACXUE.txt")

'Call BvpEnv("PACX","PAC7GY",Rep_USR & "\PACXGY.txt")
Call BvpEnv("PACX","PAC7GY",RepT_USR & "\PACXGY.txt")

'Call BvpEnv("PACX","PAC7TD",Rep_USR & "\PACXTD.txt")
Call BvpEnv("PACX","PAC7TD",RepT_USR & "\PACXTD.txt")

'Call BvpEnv("PACX","PAC7MV",Rep_USR & "\PACXMV.txt")
Call BvpEnv("PACX","PAC7MV",RepT_USR & "\PACXMV.txt")

'Call BvpEnv("PACX","PAC7MR",Rep_USR & "\PACXMR.txt")
Call BvpEnv("PACX","PAC7MR",RepT_USR & "\PACXMR.txt")

'Call BvpEnv("PACX","PAC7MX",Rep_USR & "\PACXMX.txt")
Call BvpEnv("PACX","PAC7MX",RepT_USR & "\PACXMX.txt")

Call BvpEnv("PACX","PAC7IA",Rep_USR & "\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_USR & "\PACXED.txt")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_USR & "\PACXEG.txt")
Call BvpEnv("PACX","PAC7EM",Rep_USR & "\PACXEM.txt")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_USR & "\PACXEU.txt")
Call BvpEnv("PACX","PAC7EZ",Rep_USR & "\PACXEZ.txt")

Call BvpEnv("PACX","SYSEXT",Rep_TMP & "\WSY.tmp")
Call RunCmdLog ("BVPACX")
If Return = 4 Then
Call Msg_Log (Array("1030"))
End If
If Return = 8 Then
Call Msg_Log (Array("1057"))
End If
Call Err_Cod(Return , 0 , "PACX")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Chapitre 4. Extraction personnalisée/Documentation automatisée

Introduction

Les fonctions PAF+/Extraction et GDP+/Edition peuvent être utilisées séparément l'une de l'autre ou conjointement (PAF-GDP).

PAF+ permet l'écriture du Plan type d'extraction et son exploitation quand le PTEEx est un Extracteur utilisateur.

GDP+ permet l'écriture et l'exploitation du Plan type d'édition.

Les fonctionnalités PAF-GDP sont mises en oeuvre quand le Plan type d'édition fait appel à un Plan type d'extraction de type macro-commande.

- Lorsque la fonction PAF+/Extraction est utilisée seule, elle permet de générer un Programme extracteur utilisateur avec possibilité de mise en forme des données extraites.
- Lorsque la fonction GDP+/Edition est utilisée seule, elle permet de créer des squelettes de normalisation pour l'édition des rapports (options d'édition standard, textes systématiquement appelés, codification des appels normalisée).
- Lorsque les deux fonctions sont utilisées conjointement, PAF+ extrait des données de la base qui seront ensuite exploitées par GDP+, pour être finalement édités dans un Volume.

Pour plus d'informations sur ces fonctionnalités, se reporter aux manuels "Pacbase Access Facility (PAF)" et "Gestionnaire de Documentation Personnalisée (GDP)".

Extractions personnalisées - PAF+

XPAF - Validation d'un Plan type d'extraction

XPAF - Présentation générale

La procédure de validation d'un Plan type d'extraction permet à l'utilisateur de réaliser des extractions spécifiques que les procédures standard ne peuvent réaliser. Se reporter au manuel "Pacbase Access Facility (PAF)".

Résultat obtenu

Il est de deux types suivant que le domaine extrait est destiné ou non à s'intégrer dans un document : macro-commande ou extracteur utilisateur.

Macro-commande : c'est un sous-programme à activer dans une demande d'édition par GPRT (PCV).

Extracteur utilisateur : c'est un programme source à compiler et à exécuter.

Condition préalable

Pour pouvoir utiliser ce module, le gestionnaire de la base aura effectué une mise à jour de la base avec le fichier mouvements fourni à l'installation et contenant la méta-entité extension .PPTEX, de type 7E (procédure VINS).

Le fichier GS, initialisé par la procédure LDGS, doit exister au préalable.

Mise en oeuvre

Dans un premier temps, il s'agit pour l'utilisateur de créer une instance Y7E de cette méta-entité extension. Sa définition et sa description détermineront les caractéristiques et la mise en forme souhaitée de l'extraction.

Anomalies d'exécution

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle après suppression du problème.

Edition obtenue

Cette procédure édite un compte-rendu de contrôle décrivant le 'Plan type d'extraction' ainsi que sa simulation.

XPAF - Entrées Utilisateur

Une ligne '*' pour la bibliothèque et la session à consulter.

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	bbb	Code bibliothèque
22	4	nnnn	Numéro de session
26	1	'T'	Etat de la session
68	1	' '	Edition normale

Pos.	Lon.	Valeur	Signification
		'1'	Edition en majuscules

Une ligne 'EX' pour renseigner les éléments suivants :

Pos.	Lon.	Valeur	Signification
2	2	'EX'	Code ligne
4	2		Type méta-entité extension (Y7E par défaut)
6	6	eeeeee	Code Entité utilisateur
			Bibliothèque et session à préciser si les méta-entités dont on extrait les instances sont dans un sous-réseau parallèle (extractions d'Entités utilisateur gérées dans la Station de travail par exemple)
12	3	bbb	Code bibliothèque le cas échéant
15	4	nnnn	Numéro de session le cas échéant
19	1	'T'	Etat de la session
20	6	'UPDATE'	Pour une mise à jour de GS
		SPACE	Vérification de l'existence du plan type dans GS (sans mise à jour). Vérification de l'utilisation de l'entité dans le réseau sélectionné.

Exemples :

*user____passwordBIB

EX7EEXT002_____UPDATE

*user____passwordBIB

EX7EEXT002

XPAF - Description des étapes

Accès et contrôle : PTEX30

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur VisualAge Pacbase
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration

Code	Nom physique	Type	Libellé
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AY	Rép. base : AY	Entrée	Données extension de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7SP	Système - Rép. skel : SP	Entrée	Squelette variable
PAC7GS	Rép. base : GS	Entrée/ Sortie	Schémas d'extraction
PAC7ED	Rép. tmp : WED	Sortie	Compte rendu transmis au programme d'édition
PAC7GP	Rép. tmp : WGP	Sortie	Source généré provisoire
PAC7DD	Rép. user : XPAFDDX30	Etat	Compte-rendu

Génération extracteur : PTEX80

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur VisualAge Pacbase
PAC7SF	Système - Rép. skel : SF	Entrée	Squelette fixe
PAC7GP	Rép. tmp : WGP	Entrée	Source généré par le PTEX30
PAC7ST	Rép. user : PAF80	Sortie	Source généré, à traduire

Pré-processeur : PAFP10

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAF80	Rép. user : PAF80	Entrée	Programmes générés
COB80	Rép. user : COB80	Sortie	Programmes générés à compiler
PAFREP	Rép. user : PPAFREP10 ou PAFREP	Etat	Compte-rendu d'erreur

Edition du PTEX : PTEXD0

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PAC7ED	Rép. tmp : WED	Entrée	Compte-rendu du PTEX30
PAC7GS	Rép. base : GS	Entrée/ Sortie	Schémas d'extraction
PAC7RD	Rép. user : XPAFRDXD0	Etat	Compte-rendu de contrôle

XPAF - Script d'exécution

```

| -----
|          VISUALAGE PACBASE
|
| -----
|          - XPAF (PAF EXTENSION) -
|
| -----
|
| THE EXTRACTION MASTER PATH VALIDATION PROCEDURE,
| XPAF, ALLOWS FOR THE SIMULATION OF SPECIFIC EXTRACTIONS
| THAT THE STANDARD PROCEDURES ARE NOT ABLE TO PERFORM.
|
| INPUT :
|   - USER IDENTIFICATION LINE (REQUIRED)

```

```

'      COL 2 : "*"
'      COL 3 : USERIDXX
'      COL 11 : PASSWORD
'      COL 19 : (BBB) LIBRARY CODE
'      COL 22 : (4 N) SESSION NUMBER
'      COL 26 : (1 CAR.) SESSION VERSION
'      COL 68 : " " STANDARD PRINT
'              "1" UPPERCASE PRINT
'
' - COMMAND LINE :
' COL 2 : "EX" LINE CODE
' COL 4 : (2 CAR.) METAENTITY TYPE (7E BY DEFAULT)
' COL 6 : (6 CAR.) USER ENTITY CODE
' COL 12 : (BBB) LIBRARY CODE (IF THE U.E.O.
' COL 15 : (4 N) SESSION NUMBER ARE IN PARALLEL
' COL 19 : (1 CAR.) SESSION VERSION SUB-NETWORK)
' COL 20 : "UPDATE" UPDATE OF GS
'          " " CHECK OF THE PRESENCE OF THE
'          MASTER PATH IN GS.
'
' -----
'

```

```
<job id=XPAF>
```

```

<script language="VBScript">
Dim MyProc
MyProc = "XPAF"
</script>

```

```
<script language="VBScript" src="INIT.vbs"/>
```

```
<script language="VBScript">
```

```
If c_error = 1 then Wscript.Quit (1) End If
```

```
Call Msg_Log (Array("1022" , "PTEX30"))
```

```

'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTEX30","PAC7ED",Rep_TMP & "\WED.tmp")
Call BvpEnv("PTEX30","PAC7DD",Rep_USR & "\XPAFDDX30.txt")
Call BvpEnv("PTEX30","PAC7GP",Rep_TMP & "\WGP.tmp")
WshEnv("PAC7GS") = Rep_BASE & "\GS"
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7SP") = Rep_SKEL & "\SP"
Call RunCmdLog ("BVPTEX30")

```

```

If Return <> 8 Then
Call Err_Cod(Return , 0 , "PTEX30")
End If

```

```

If Return = 0 Then

Call Msg_Log (Array("1022" , "PTEX80"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTEX80","PAC7GP",Rep_TMP & "\WGP.tmp")
WshEnv("PAC7SF") = Rep_SKEL & "\SF"
Call BvpEnv("PTEX80","PAC7ST",Rep_USR & "\PAF80.txt")
Call RunCmdLog ("BVPTEX80")
Call Err_Cod(Return , 0 , "PTEX80")

Call Msg_Log (Array("1022" , "PAFP10"))
'-----
WshEnv("COBSW") = "-N"
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PAFP10","PAF80",Rep_USR & "\PAF80.txt")
Call BvpEnv("PAFP10","COB80",Rep_USR & "\COB80.txt")
Call BvpEnv("PAFP10","PAFREP",Rep_USR & "\PAFREP.txt")
Call RunCmdLog ("BVPAFP10")
Call Err_Cod(Return , 0 , "PAFP10")

End If

Call Msg_Log (Array("1022" , "PTEXD0"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("PTEXD0","PAC7ED",Rep_TMP & "\WED.tmp")
WshEnv("PAC7GS") = Rep_BASE & "\GS"
Call BvpEnv("PTEXD0","PAC7RD",Rep_USR & "\XPAFRDXD0.txt")
Call RunCmdLog ("BVPTEXD0")

Call Err_Cod(Return , 0 , "PTEXD0")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

XPAF - Exploitation d'un plan type d'extraction

EXECUTION D'UN EXTRACTEUR UTILISATEUR (PTE_x DE TYPE E)

Une fois validé, compilé et linké, un Programme Extracteur Utilisateur est prêt à être exécuté.

EXECUTION D'UNE MACRO-COMMANDE (PTEx DE TYPE M)

Une fois validée, compilée et linkée, une macro-commande n'est pas prête à être exécutée. Elle doit être appelée dans un Plan type d'édition.

NOTE

Un PTEx est indépendant de la base où il est défini et décrit.

Structuration de documentation - GDP+

XPDM - Validation d'un Plan type d'édition

XPDM - Présentation générale

Un Plan type d'édition est un rapport (entité 'V') de type P destiné à être appelé dans un autre rapport. Il peut remplir plusieurs fonctions :

- Mémoriser des descriptions générales (options d'édition par exemple) qui ne seront plus à redéfinir pour chaque rapport.
- Editer des informations extraites à l'aide d'un Plan type d'extraction avec des possibilités de récursivité.

La procédure XPDM met à jour le fichier GS des Plans type d'extraction si aucune erreur grave n'est détectée. La procédure peut aussi être utilisée sans mise à jour de GS.

Pour plus d'informations, se reporter au manuel "Gestionnaire de Documentation Personnalisée".

Anomalies d'exécution

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle après suppression du problème.

Edition obtenue

Cette procédure édite la description du plan type d'édition ainsi que des commentaires et, éventuellement, un relevé des anomalies détectées.

XPDM - Entrées Utilisateur

Une ligne '*' pour définir le contexte.

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	bbb	Code bibliothèque
22	4	nnnn	Numéro de session
26	1	'T'	Etat de la session
68	1	' '	Edition normale
		'1'	Edition en majuscules

Une ligne 'EP' pour renseigner les éléments suivants :

Pos.	Lon.	Valeur	Signification
2	2	'EP'	Code ligne
4	6	rrrrrr	Code du rapport
10	6	'UPDATE'	Pour une mise à jour du fichier GS
		SPACE	Vérification de l'existence du plan type dans GS. Vérification de l'utilisation du rapport dans le sous-réseau sélectionné. Si vérification, pas de mise à jour de GS.

Exemples

_ *user ____passwordBIB

_EPMANUELUPDATE

_ *user ____passwordBIB

_EPMANUEL

XPDM - Description des étapes

Extraction du plan type d'édition : PTED30

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur

Code	Nom physique	Type	Libellé
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AY	Rép. base : AY	Entrée	Données extension de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7GS	Rép. base : GS	Entrée/Sortie	Schémas d'extraction
PAC7ED	Rép. tmp : WED	Sortie	Fichier pour compte rendu
PAC7SG	Rép. tmp : WSG	Sortie	Préparation à la mise à jour de GS
PAC7DD	Rép. user : XPDMDDD30	Etat	Compte rendu

Maj de GS et édition du plan type d'édition : PTED60

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PAC7ED	Rép. tmp : WED	Entrée	Fichier édition
PAC7SG	Rép. tmp : WSG	Entrée	Préparation à la mise à jour de GS
PAC7GS	Rép. base : GS	Sortie	Schémas d'extraction
ETATGP	Rép. user : XPDMGPD60	Etat	Compte-rendu

XPDM - Script d'exécution

```
'-----  
'          VISUALAGE PACBASE  
'-----  
'          - XPDM   (PDM EXTENSION) -  
'-----  
'  
' A MASTER OUTLINE IS A P-TYPE VOLUME ("V" ENTITY)  
' DESIGNED TO BE CALLED IN ANOTHER PDM VOLUME.  
'  
' INPUT :  
' - USER IDENTIFICATION LINE (REQUIRED)  
'   COL 2 : "*"   
'   COL 3 : USERIDXX  
'   COL 11 : PASSWORD  
'   COL 19 : (BBB)   LIBRARY CODE  
'   COL 22 : (4 N)   SESSION NUMBER  
'   COL 26 : (1 CAR.) SESSION VERSION  
'   COL 68 : " "     STANDARD PRINT  
'             "1"     UPPERCASE PRINT  
' - COMMAND LINE :  
'   COL 2 : "EP"     LINE CODE  
'   COL 4 : (6 CAR.) REPORT CODE  
'   COL 10 : "UPDATE" UPDATE OF GS  
'             " "     CHECK OF THE PRESENCE OF VOLUME  
'                   IN GS.  
'-----  
'  
<job id=XPDM>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "XPDM"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Call Msg_Log (Array("1022" , "PTED30"))  
'-----  
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"  
WshEnv("PAC7AN") = Rep_BASE & "\\AN"  
WshEnv("PAC7AR") = Rep_BASE & "\\AR"  
WshEnv("PAC7AY") = Rep_BASE & "\\AY"  
WshEnv("PACGGN") = Rep_ABASE & "\\AN"  
WshEnv("PACGGR") = Rep_ABASE & "\\AR"  
WshEnv("PACGGU") = Rep_ABASE & "\\GU"  
Call BvpEnv("PTED30","PAC7DD",Rep_USR & "\\XPDMDDD30.txt")  
Call BvpEnv("PTED30","PAC7ED",Rep_TMP & "\\WED.tmp")
```

```

WshEnv("PAC7GS") = Rep_BASE & "\GS"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTED30","PAC7SG",Rep_TMP & "\WSG.tmp")
Call RunCmdLog ("BVPTED30")
Call Err_Cod(Return , 0 , "PTED30")

Call Msg_Log (Array("1022" , "PTED60"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("PTED60","PAC7ED",Rep_TMP & "\WED.tmp")
WshEnv("PAC7GS") = Rep_BASE & "\GS"
Call BvpEnv("PTED60","PAC7SG",Rep_TMP & "\WSG.tmp")
Call BvpEnv("PTED60","ETATGP",Rep_USR & "\XPDMGPD60.txt")
Call RunCmdLog ("BVPTED60")
Call Err_Cod(Return , 0 , "PTED60")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Plans type d'extraction et d'édition

PRGS - Edition du fichier des plans type

PRGS - Présentation générale

Cette procédure permet l'édition du contenu du fichier PAC7GS où sont stockés les Plans types d'édition et les Plans types d'extraction.

Résultat obtenu

On obtient une édition faisant apparaître le Plan type d'extraction avec les Plans types d'édition qui s'y rattachent.

PRGS - Entrées Utilisateur

Une ligne '*' pour identifier l'utilisateur.

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe

PRGS - Description des étapes

Edition du fichier des plans types : PTEP90

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7GS	Rép. base : GS	Entrée	Schémas d'extraction
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7DD	Rép. user : PRGSDDP90	Etat	Compte rendu
ETATGS	Rép. user : PRGSGSP90	Etat	Etat du fichier des plans types

PRGS - Script d'exécution

```
|-----  
|          VISUALAGE PACBASE  
|-----  
|          - PRINT OF MASTER PATH FILE -  
|-----  
|  
| THE PRGS PROCEDURE PRINTS THE CONTENTS OF THE  
| PAC7GS FILE, WHERE MASTER OUTLINES AND EXTRACTION  
| MASTER PATHS ARE STORED.  
|-----  
|
```

```

<job id=PRGS>

<script language="VBScript">
Dim MyProc
MyProc = "PRGS"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTEP90"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTEP90","PAC7DD",Rep_USR & "\PRGSDDP90.txt")
WshEnv("PAC7GS") = Rep_BASE & "\GS"
Call BvpEnv("PTEP90","ETATGS",Rep_USR & "\PRGSGSP90.txt")
WshEnv("PAC7MB") = Fic_Input
Call RunCmdLog ("BVPTEP90")
If Return = 8 Then
Call Msg_Log (Array("1027"))
End If
Call Err_Cod(Return , 0 , "PTEP90")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr (Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

Chapitre 5. Mises à jour batch

UPDP - Mise à jour à partir de tables PAF

UPDP - Présentation générale

Cette procédure effectue la mise à jour batch du réseau à partir d'un fichier séquentiel à l'image des tables PAF.

Son principe de fonctionnement est tout à fait similaire à celui de la procédure UPDT, excepté le format des mouvements en entrée.

Anomalies d'exécution

Se reporter au chapitre UPDT.

UPDP - Entrées / Règles de mise à jour / Résultats

Le fichier séquentiel des mouvements en entrée provient d'un programme extracteur PAF ou de la procédure PACX. Ses enregistrements sont à l'image des tables PAF. Pour la description de ces tables, voir la documentation des Tables Pacbase Access Facility.

Pos.	Lon.	Contenu
1	1	Code mouvement (C, M, X, A ou D, B, S)
2	10	Code de la table PAF
12	299	Contenu de la table PAF, telle qu'elle est décrite dans le manuel "Les Tables Pacbase Access Facility".

Restriction pour les Tables Description et Définition des Entités Utilisateur Clientes et Extensions.

La taille du fichier en entrée d'UPDP est de 310 caractères. Or, ces tables ont une longueur supérieure à 310 caractères. Il faut donc reformater ces enregistrements de la façon suivante :

Pour les fiches de définition des EUs Clientes et Extensions - \$TTDEF ou YTTDEF.

Pos.	Lon.	Contenu
1	1	Code mouvement (C, M, X, A ou D, B, S)
2	10	Code de la table PAF

Pos.	Lon.	Contenu
12	1	Code suite du mouvement : à blanc pour le premier enregistrement, caractère quelconque pour les enregistrements suite.
13	1	Caractère séparateur (1)
14	55	Zone des mots-clés explicites
69	305	Contenu des colonnes concaténées

Pour les autres descriptions des EUs Clientes et Extensions - \$TTDxx ou YTTDxx.

Pos.	Lon.	Contenu
1	1	Code mouvement (C, M, X, A ou D, B)
2	10	Code de la table PAF
12	1	Code suite du mouvement : à blanc pour le premier enregistrement, caractère quelconque pour les enregistrements suite.
13	1	Caractère séparateur (1)
14	30	Code de l'Entité Utilisateur
44	305	Contenu des colonnes concaténées

(1) Si le caractère séparateur est à blanc, les données sont concaténées les unes derrière les autres en respectant la longueur des données.

Si le caractère séparateur est présent, les données sont séparées par ce caractère. Les blancs non significatifs se trouvant à la fin de la donnée sont facultatifs.

Le caractère séparateur ne doit pas être présent dans la donnée naturellement.

Règles de mise à jour

Les mouvements de mise à jour ne sont pas triés.

Chaque ensemble de mouvements affectant une bibliothèque ou une session doit être précédé d'une ligne de code table ASSIGN :

Pos.	Lon.	Valeur	Signification
2	10	'ASSIGN'	Code de la table
12	8	uuuuuuuu	Code utilisateur
20	8	pppppppp	Mot de passe

Pos.	Lon.	Valeur	Signification
28	3	bbb	Code bibliothèque
31	4	nnnn	Numéro de session (blanc=courante)
35	1	'T'	Etat de la session si session test
40	3	ppp	Code produit (si contrôle de la base par DSMS)
43	6	nnnnnn	Numéro de produit (si contrôle de la base par DSMS)
49	1		Top généré par extracteur
		' '	Mouvements manuels
		'N'	Mouvements issus d'extracteur
		'G'	Mouvements issus des reprises PG25 ou PP25

Lorsque la mise à jour s'effectue pendant que le conversationnel est actif, le flot des mouvements en entrée doit être précédé d'une ligne de code de la table CHECKP :

(Se reporter au chapitre UPDT).

Pos.	Lon.	Valeur	Signification
2	10	'CHECKP'	Code de la table
12	4	nnnn	Nombre de mouvements traités entre deux pauses ou checkpoints
16	4	'UPDT'	Procédure de mise à jour
20	2	nn	Plates-formes LAN : temps de pause, en secondes, entre deux séries de mises à jour

Editions obtenues

Se reporter au chapitre UPDT.

Résultat obtenu

Se reporter au chapitre UPDT.

UPDP - Description des étapes

Vérification de l'intégrité de la base : PTUBAS

Code		Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement

Code		Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7DS		Etat	Compte-rendu validité

. Code retour :

Cet utilitaire envoie un code retour 4 et provoque un ABEND en cas d'invalidité de la base.

Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7GY	Entrée utilisateur	Entrée	Mouvements de mise à jour
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements mis en forme (doit pouvoir contenir tous les mouvements en entrée plus les mouvements d'annulation élémentaire générés par les mouvements d'annulation multiple) (longueur=170)
PAC7ME	Rép. tmp : WME	Sortie	Fichier de travail (longueur=372)
PAC7MW	Rép. tmp : WMW	Sortie	Fichier de travail (longueur=170)
PAC7MX	Rép. tmp : WMX	Sortie	Fichier de travail (longueur=743)
PAC7MY	Rép. tmp : WMY	Sortie	Fichier de travail (longueur=743)

Mise à jour de la Base de Développement : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Sortie	Données de la Base de Développement
PAC7AN	Rép. base : AN	Sortie	Index de la Base de Développement
PAC7AY	Rép. base : AY	Sortie	Extension de la Base de Développement
PAC7AJ	Rép. journal : AJ	Sortie	Journal de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	Rép. user : UPDPRBA15	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. user : UPDPRYA15	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE	Rép. user : UPDPIEA15	Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF	Rép. user : UPDPIFA15	Etat	Récapitulatif des mouvements erronés (longueur=132)

La liste des mouvements propres à un utilisateur est précédée d'une bannière portant son code.

Codes retour :

- 0 : OK sans erreur
- 2 : erreur warning
- 4 : erreur grave

UPDP - Script d'exécution

```
'-----  
' VISUALAGE PACBASE  
'-----  
' - BATCH UPDATE FROM PAF TABLES -  
'-----  
'  
' THE UPDP PROCEDURE PERFORMS AN UPDATE OF THE DATABASE  
' FROM A SEQUENTIAL FILE REFLECTING PAF TABLES.  
'  
' THE SEQUENTIAL FILE OF INPUT TRANSACTIONS IS PRODUCED  
' BY A PAF EXTRACTOR PROGRAM. ITS RECORDS MIRROR  
' THE PAF TABLES.  
' EACH SET OF TRANSACTIONS IMPACTING A LIBRARY OR SESSION  
' MUST BE PRECEDED BY AN ASSIGN TABLE CODE LINE.  
' WHEN THE UPDATE IS PERFORMED WHILE THE TP IS ACTIVE  
' (ON PLATFORMS THAT SUPPORT THIS FUNCTIONALITY),  
' THE INPUT TRANSACTION FLOW MUST BE PRECEDED BY A CHECKP  
' TABLE CODE LINE.  
'-----  
'  
<job id=UPDP>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "UPDP"  
  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Call Msg_Log (Array("1022" , "PTUBAS"))  
'-----  
WshEnv("PAC7AE") = Rep_SKEL & "\AE"  
WshEnv("PAC7AR") = Rep_BASE & "\AR"  
Call BvpEnv("PTUBAS", "PAC7DS", Rep_USR & "\UPDTSBAS.txt")  
Call RunCmdLog ("BVPTUBAS")  
WshVolEnv("RC") = Return  
If Return = 4 Then  
Call Msg_Log (Array("1051"))  
End If  
Call Err_Cod(Return , 0 , "PTUBAS")  
  
'Input File extracted from PACX  
'in RepT_USR is Global User Directory.  
  
Call Msg_Log (Array("1022" , "PAF900"))  
'-----
```



```

WshEnv("PAC7GY") = Fic_Input
If Not FSO.FileExists(WshEnv("PAC7GY")) Then
  Call Msg_Log (Array("1004" , "PAC7GY"))
  Msg = Nls_Lib
  EndJob (1)
  Wscript.Quit (1)
End If

WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Call RunCmdLog ("BVPAF900")
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PACA15","PAC7DC",Rep_BASE & "\DC")
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\UPDPIEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\UPDPIFA15.txt")
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA15","PAC7RB",Rep_USR & "\UPDPRBA15.txt")
Call BvpEnv("PACA15","PAC7RY",Rep_USR & "\UPDPRYA15.txt")
Call RunCmdLog ("BVPACA15")
If Return = 2 Then
Call Msg_Log (Array("1061"))
End If
If Return = 4 Then
Call Msg_Log (Array("1060"))
End If
Call Err_Cod(Return , 4 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

```

```
Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>
```

UPDT - Mise à jour

UPDT - Présentation générale

Cette procédure effectue la mise à jour batch du réseau. Elle permet l'accès à toutes les bibliothèques en fonction des autorisations des différents utilisateurs.

Pour les utilisateurs du module DSMS (DSM), elle consulte le fichier des éléments du produit (DC).

Condition d'exécution

Cette procédure mettant à jour le réseau, il est nécessaire de fermer les fichiers AR, AN, AJ et AY dans le conversationnel (sauf pour les matériels permettant la concurrence batch/conversationnel).

Remarques importantes

Dans le cas de mises à jour comportant de très nombreux mouvements (un ordre de grandeur est de 5000 mouvements), il peut être nécessaire, avant l'exécution de la procédure :

- d'effectuer une sauvegarde, un archivage et un rechargement, pour éventuellement agrandir les fichiers, ou pour les réorganiser physiquement de façon à rendre disponible tout l'espace prévu initialement,
- d'inhiber temporairement la journalisation (voir le chapitre "Gestion des Bases de Développement", sous-chapitre "Restauration", dans le manuel "Les procédures de l'Administrateur").

Cette procédure incrémente le numéro de session courante dans deux cas :

- elle constitue la première connexion de la journée à la base,
- elle contient un ordre d'historisation de la base, à lancer par l'administrateur (voir le manuel "Les procédures de l'administrateur").

Anomalies d'exécution

Se reporter au sous-chapitre "Anomalies d'exécution", chapitre "Introduction à l'utilisation des procédures batch" de ce manuel.

Il convient de distinguer deux sortes d'anomalies :

- Anomalies apparaissant avant l'exécution du programme BVPACA15 ou sur l'ouverture des fichiers dans celui-ci : il suffit de relancer la procédure après avoir remédié au problème.
- Anomalies survenant lors de l'exécution du programme BVPACA15 : la base est laissée dans un état incohérent s'il n'y a pas de rollback. Si le problème est apparu lors d'une entrée-sortie sur un fichier de la base, l'examen du message édité dicte la solution.

Si la procédure est stoppée suite à l'anomalie "Libellé court déjà utilisé", la base demeure cohérente et la procédure peut être relancée immédiatement après correction du libellé.

Dans tous les cas, la reprise ne peut se faire que par rechargement d'une sauvegarde avec application des mouvements archivés postérieurs à cette sauvegarde (procédure REST).

UPDT - Entrées / Règles de mise à jour / Résultats

Règles de mise à jour

Chaque ensemble de mouvements affectant une bibliothèque doit être précédé d'une ligne '*', précisant le contexte.

Les mouvements de mise à jour ne sont pas triés.

Editions obtenues

Deux éditions sont fournies par cette procédure :

- un compte rendu global de la mise à jour,
- une liste des mouvements rejetés par la mise à jour.

Elles sont éditées par utilisateur, chaque ensemble de mouvements étant séparé par une 'flamme'.

Résultat obtenu

Une fois la mise à jour effectuée, le résultat obtenu est :

- Un réseau prêt à être manipulé en conversationnel ou en mode batch.
- Une journalisation des mouvements ayant modifié le réseau, si son inhibition n'a pas été demandée lors de la dernière restauration.

Ces mouvements sont composés d'une partie commune contenant le code action, un identifiant de ligne et une partie spécifique dont le détail est donné dans les sections suivantes pour chaque description d'entité.

Codes action

Code action	Libellé
C	Création de la ligne dans la bibliothèque
M	Modification de la ligne
blanc	Création ou modification suivant l'état de la bibliothèque
X	Création ou modification avec possibilité d'insertion du caractère '&'
A	Annulation de la ligne
B	Début d'annulation multiple
R	Fin d'annulation multiple au-delà de cette ligne
S	Annulation totale d'une entité

Remarques concernant l'annulation

Si une entité est utilisée dans plusieurs bibliothèques les annulations faites dans une bibliothèque inférieure sont rejetées.

Par contre, il est possible en batch d'annuler globalement (code annulation multiple 'B') une entité et ses utilisations dans les écrans, états ou segments. Cependant, ces annulations ne seront effectives que dans les bibliothèques de mise à jour.

Le code B entraîne la génération de mouvements d'annulations élémentaires.

Le code S ne peut être mis que sur une fiche entité, il n'y aura qu'un seul mouvement journalisé. Des contrôles préalables seront effectués avant la mise à jour.

Note

Une zone non renseignée n'est pas modifiée. La remise à blanc se fait par la saisie du caractère '&'.

Codes action spécifiques : "F" et "P"

Les codes action "F" et "P" sont utilisés dans le cadre des extractions pour mise à jour.

Le code "F" permet de forcer une mise à jour, i.e. à la suite d'une extraction (via EXLI ou tout autre extracteur), permet de créer une Définition incomplète uniquement pour que les utilisations de ces entités (en principe des Entités Utilisateur) soient satisfaites, leur tri étant impossible.

Cette valeur du code action entraîne la mise à jour de la base.

Le code action "P" permet de donner une ligne d'identification à toutes les lignes de Description qui suivent sans pour autant mettre à jour la Définition de cette entité.

Ce peut être le cas de lignes P d'un programme dans une bibliothèque où la Définition est uniquement dans une bibliothèque supérieure.

Demande de Checkpoints

Cette spécification permet à l'utilisateur de demander des points de synchronisation lors de la mise à jour batch 'UPDT'.

Les checkpoints sont effectués suivant une fréquence déterminée par l'utilisateur. (ex : une fréquence de 0100, signifie qu'un checkpoint sera fait tous les 100 mouvements traités).

Fréquence des checkpoints dans une mise à jour batch

Pour la mise à jour (UPDT), la fréquence des checkpoints est demandée à l'aide d'une ligne d'accès de code Y placée avant la première ligne '*' du flot de mise à jour. Cette ligne doit être définie comme suit :

Pos.	Lon.	Valeur	Signification
2	1	'Y'	code ligne
4	4	nnnn	fréquence des checkpoints (valeur par défaut : 0000)
8	2	nn	temps de pause à chaque prise de checkpoint

Pour la restauration (REST ou RESY), la fréquence des checkpoints est demandée à l'aide de l'entrée utilisateur définie pour ces procédures.

Mise à jour concurrente batch-TP

La mise en place de checkpoints dans le programme BVPACA15 de la procédure UPDT permet de faire fonctionner celle-ci en concurrence avec le TP. Cette concurrence UPDT-TP doit être réservée à de petits lots de mouvements exceptionnels.

En effet, le passage de la procédure UPDT pendant la session TP peut entraîner des blocages entre 2 points de reprise successifs, impliquant une augmentation des temps de réponses TP.

Dans le cas d'un abort non bloquant (journal plein ou problème sur l'appel d'un checkpoint), la procédure pourra être relancée après avoir éliminé les mouvements déjà traités dans l'entrée utilisateur.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
4	4		fréquence des checkpoints

Entrées communes

Lignes anciennement -G (lignes VC, VG, VE, VO) :

La ligne d'accès de code VC, si elle permet l'appel de Format guide, permet également l'affectation de commentaires sur une entité ou certaines lignes de description.

Insertion de commentaires (ligne VC)

- une ligne VC avec le type et le code de l'entité concernée et le numéro de ligne.
- une ligne VC contenant le commentaire en colonne 4 et le caractère suite ("*") en colonne 80.

Appel d'un Format Guide (lignes VC et VZ)

- une seule ligne VC suffit. Sur la première ligne, on trouve le type et le code de l'entité concernée, accompagnés d'un numéro de ligne s'il s'agit d'une Description, le type de ligne valorisé à "I" et le code du Format-Guide.
- une ligne VZ par zone variable du Format-Guide appelé, (voir section suivante, valorisation du Format-Guide), le sous-numéro de ligne et la valeur de la description.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'VC'	Code ligne pour un écran GC
		'VG'	Code ligne pour un écran GG
		'VE'	Code ligne pour un écran GE (l'appel de Format Guide n'est pas possible dans cet écran)
		'VO'	Code ligne pour un écran GO

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
4	2		Type entité support des Commentaires
6	30		Code entité
36	3		Numéro de ligne
39	3		Numéro de la ligne commentée
42	1		Type de ligne
		'I'	Valeur pour l'insertion d'un appel de Format-Guide.
43	6		Code du format guide appelé

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'VC'	Code ligne pour un écran GC
		'VG'	Code ligne pour un écran GG
		'VE'	Code ligne pour un écran GE (l'appel de Format Guide n'est pas possible dans cet écran)
		'VO'	Code ligne pour un écran GO
4	60		Texte de Documentation
80	1		Ligne suite
		'*'	Valeur à saisir pour indiquer une ligne suite à la précédente.

Valorisation du Format-Guide (ligne VZ) :

La ligne d'accès de code VZ permet de saisir le contenu de la partie variable.

La structure de la ligne VZ doit se calquer sur celle de la Description du format guide. Les parties variables sont les unes derrière les autres, en comptant la longueur maximale de chaque partie variable.

Note

Une ligne d'accès de code VZ est obligatoirement derrière une ligne d'accès de code VC d'appel de format guide.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
2	2		Code ligne sur 2 caractères
		'VZ'	
4	2		Sous-numéro de la ligne
6	20		Libellé
			Il s'agit de la partie fixe de la ligne qui apparaît lors de l'appel du Format Guide.
			Son contenu dépend du "Type de ligne" de Format Guide.
			le libellé n'est pas recadré (pour être contigu à la valeur, il doit donc être cadré à droite).
			Sur les lignes à générer (dont le "Type à affecter lors de l'appel" est valorisé à 'G' sur la Description du Format Guide), chaque instruction doit être cadrée à gauche. Si l'instruction comprend plusieurs lignes, la première est cadrée à gauche, les autres doivent débiter par au moins un blanc.
26	40		DESCRIPTION / 2EME PARTIE
			Cette zone est spécifique à un appel de Format Guide.
			L'option C2 dans la zone CODE OPERATION permet la tabulation automatique dans cette zone.
			Cette zone est initialisée par défaut avec des blancs ou par la valeur indiquée dans la zone VALEUR INITIALE d'une ligne de description standard d'un Format Guide (Type = "blanc"). Si des paramètres symboliques ont été définis dans la Description du Format Guide (-D), vous pouvez les indiquer dans cette zone. Ils seront remplacés par leur valeur correspondante et demeureront affichés sur la droite de l'écran.

Appels d'instances par Relations (ligne QR) :

La ligne d'accès de code QR permet l'appel d'instances d'entités par Relations.

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'QR'	
4	2		Type entité support des Commentaires
6	2		Type de la Méta Entité

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			C'est un code alphanumérique saisi à la création et caractérisant de façon unique la Méta Entité dans tous ses appels (deux Méta Entités différentes ne peuvent avoir le même type).
			Il ne peut être modifié si des Entités Utilisateur ont déjà été définies pour cette Méta Entité.
			Lorsqu'il est utilisé pour définir ou pour décrire une Entité Utilisateur, il est précédé du caractère '\$' (exemple : si le type de la ME 'JOB' est 'JO', les EU sont référencées par '\$JO.....')
8	30		Code entité (30 caractères)
38	3		Numéro de ligne
			Numérique. Il est recommandé de commencer par la ligne 100 et de numéroter les lignes de 20 en 20 pour permettre des insertions ultérieures.
			Zone alphanumérique dans le cas de la génération d'un accès SQL personnalisé. Il est possible de créer des lignes au-delà de la limite qui était fixée initialement à "1000".
41	6		Code Relation Utilisateur
47	30		Code entité (30 caractères)

Blocage d'entités (ligne R) :

La ligne d'accès de code R permet de bloquer la mise à jour des entités.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'R'	
3	2		Type d'entité
			Détermine le type d'entité sur laquelle un ou plusieurs mots-clés seront affectés :
		'L1'	Blocs Base de Données
		'H'	Ecran
		'K1'	Entité Méthode
		'Y1'	Méta-entité cliente
		'Y3'	Entité Utilisateur

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'B'	Etat
		'V1'	Format Guide
		'tt'	Instances d'Entité Utilisateur tt (à utiliser pour la mise à jour des mots-clés des instances d'une Entité Utilisateur tt)
		'0'	Programme
		'W1'	Rapport (G.D.P.)
		'Y5'	Relation Utilisateur
		'C'	Rubrique
		'2'	Segment
		'A'	Structure de Données
		'S'	Texte
5	2		Type de la Méta Entité
			C'est un code alphanumérique saisi à la création et caractérisant de façon unique la Méta Entité dans tous ses appels (deux Méta Entités différentes ne peuvent avoir le même type).
			Il ne peut être modifié si des Entités Utilisateur ont déjà été définies pour cette Méta Entité.
			Lorsqu'il est utilisé pour définir ou pour décrire une Entité Utilisateur, il est précédé du caractère '\$' (exemple : si le type de la ME 'JOB' est 'JO', les EU sont référencées par '\$JO.....')
7	30		Code entité
37	36		Libellé du blocage
73	8		Code utilisateur

Affectation de mots-clés (ligne G) :

La ligne d'accès de code G permet de définir et d'affecter des mots-clés.

Sur une première ligne, on trouve le type et le code de l'entité concernée.

Les mots-clés (55 caractères) sont saisis sur une seconde ligne, une ligne suite, caractérisée par le caractère '*' en fin de ligne.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
2	2	Code ligne sur 2 caractères
	'G'	
4	2	Type entité support des Commentaires
6	30	Code entité
36	1	Type d'appel
	'\$'	A utiliser pour la mise à jour des mots clés des Entités Utilisateur.

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
1	1	Code mouvement
2	2	Code ligne sur 2 caractères
	'G'	
4	55	Mots clés explicites de l'entité
		Zone banalisée contenant l'ensemble des mots-clés explicites de l'entité.
80	1	Ligne suite
	'*'	Valeur à saisir pour indiquer une ligne suite à la précédente.

Rubrique

Définition (ligne C) :

La ligne d'accès de code C permet de définir une Rubrique.

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
1	1	Code mouvement
2	1	Code ligne sur 1 caractère
	'C'	
3	6	Code Rubrique
9	1	Type (propriété, rubrique ou alias)
10	36	Nom de la rubrique
46	1	Type de format
	'T'	Format interne

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
47	10	Format interne de la rubrique
74	1	Usage interne de la rubrique
75	6	Code de la rubrique mère

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
1	1	Code mouvement
2	1	Code ligne sur 1 caractère
	'C'	
3	6	Code Rubrique
9	1	Type (propriété, rubrique ou alias)
10	36	Nom de la rubrique
46	1	Type de format
	'E'	Format d'entrée.
47	10	Format d'entrée de la rubrique
74	1	Usage interne de la rubrique
75	6	Code de la rubrique mère

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
1	1	Code mouvement
2	1	Code ligne sur 1 caractère
	'C'	
3	6	Code Rubrique
9	1	Type (propriété, rubrique ou alias)
10	36	Nom de la rubrique
46	1	Type de format
	'S'	Format d'édition
47	27	Format d'édition de la rubrique
74	1	Usage interne de la rubrique
75	6	Code de la rubrique mère

Description (ligne E) :

La ligne d'accès de code E permet de décrire une Rubrique.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'E'	
3	6		Code Rubrique
9	3		Numéro de ligne
12	1		Type de ligne
		'T'	Valeur pour l'insertion d'un appel de Format-Guide.
13	1		Type d'action (saut, suite)
			Numérique
14	13		Valeur de la rubrique
27	54		Signification de la valeur rubrique

Entité Méthode

Définition (ligne K1) :

La ligne d'accès de code K1 permet de définir une entité, une relation ou une C.I.F méthode.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'K1'	
4	6		Code de l'entité méthode
10	36		Nom de l'entité méthode
46	1		Type de l'entité méthode
		'O'	Objet
		'R'	Relation
		'C'	Contrainte d'intégrité fonctionnelle
47	9		Nombre d'occurrences entité méthode
			Numérique
56	6		Code de la relation impliquée

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Pour une C.I.F. uniquement : c'est le code de la relation à laquelle la C.I.F. s'applique
62	6		Code de l'entité mère
68	10		Commentaire de l'entité

Appel propriétés dans Entité ou Relat. (ligne K3) :

La ligne d'accès de code K3 permet l'appel de propriétés dans une entité ou une relation méthode.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'K3'	
4	6		Code de l'entité méthode
10	3		Numéro de ligne
13	6		Code Rubrique
19	1		Clé d'accès ou indicatif de tri
20	3		Nombre de répétitions de la rubrique
			Numérique
23	2		Nombre de rubriques dans un groupe

Entrées pour l'entité Relation Méthode

Définition (ligne K1) :

La ligne d'accès de code K1 permet de définir une entité, une relation ou une C.I.F méthode.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'K1'	
4	6		Code de l'entité méthode
10	36		Nom de l'entité méthode

POS	LN	CLASSE ONVALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
46	1		Type de l'entité méthode
		'O'	Objet
		'R'	Relation
		'C'	Contrainte d'intégrité fonctionnelle
47	9		Nombre d'occurrences entité méthode
			Numérique
56	6		Code de la relation impliquée
			Pour une C.I.F. uniquement : c'est le code de la relation à laquelle la C.I.F. s'applique
62	6		Code de l'entité mère
68	10		Commentaire de l'entité

Appel d'Entités dans Relation ou CIF (ligne K2) :

La ligne d'accès de code K2 permet l'appel d'entités dans une relation ou dans une C.I.F.

POS	LN	CLASSE ONVALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'K2'	
4	6		Code de la relation
10	3		Numéro de ligne
13	6		Code de l'entité méthode
19	7		Cardinalité minimale de la relation
26	7		Cardinalité maximale de la relation
33	7		Cardinalité moyenne de la relation

Appel propriétés dans Entité ou Relation(ligne K3) :

La ligne d'accès de code K3 permet l'appel de propriétés dans une entité ou une relation méthode.

POS	CLASSE	VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'K3'	
4	6		Code de l'entité méthode
10	3		Numéro de ligne
13	6		Code Rubrique
19	1		Clé d'accès ou indicatif de tri
20	3		Nombre de répétitions de la rubrique
			Numérique
23	2		Nombre de rubriques dans un groupe

C.I.F Méthode

Définition (ligne K1) :

La ligne d'accès de code K1 permet de définir une entité, une relation ou une C.I.F méthode.

POS	CLASSE	VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'K1'	
4	6		Code de l'entité méthode
10	36		Nom de l'entité méthode
46	1		Type de l'entité méthode
		'O'	Objet
		'R'	Relation
		'C'	Contrainte d'intégrité fonctionnelle
47	9		Nombre d'occurrences entité méthode
			Numérique
56	6		Code de la relation impliquée
			Pour une C.I.F. uniquement : c'est le code de la relation à laquelle la C.I.F. s'applique
62	6		Code de l'entité mère

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
68	10		Commentaire de l'entité

Appel d'Entités dans Relation ou CIF (ligne K2) :

La ligne d'accès de code K2 permet l'appel d'entités dans une relation ou dans une C.I.F.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'K2'	
4	6		Code de la relation
10	3		Numéro de ligne
13	6		Code de l'entité méthode
19	7		Cardinalité minimale de la relation
26	7		Cardinalité maximale de la relation
33	7		Cardinalité moyenne de la relation

Structure de Données

Définition (ligne A) :

La ligne d'accès de code A permet de définir une structure de données.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'A'	
3	2		Code de la structure de données
5	30		Nom de la structure de données
35	44		Commentaire de la structure données
79	1		Type de structure de données
80	1		Option d'édition des fichiers

POS	LONGUEUR	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		'O'	Les descriptifs de fichiers comporteront l'indication des contrôles et mises à jour ; cette option n'est à prendre que pour les fichiers comportant des contrôles, des mises à jour, des rubriques à répétitions variables ou à valeurs initiales ; elle est obligatoire pour générer des libellés d'erreur.
		'N'	Les descriptifs de fichiers ne comporteront pas l'indication des contrôles et mises à jour ; dans ce cas, les longueurs et adresses des rubriques dans l'enregistrement seront indiquées (option par défaut)
		'E'	Les descriptifs de fichier seront présentés en format d'entrée avec adresses longueurs et valeurs initiales des rubriques dans l'enregistrement
		'I'	Les descriptifs de fichier seront présentés en format interne avec adresses longueurs et valeurs initiales des rubriques dans l'enregistrement

Segment

Définition (ligne 2) :

La ligne d'accès de code 2 permet de définir un Segment.

POS	LONGUEUR	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'2'	
4	4		Code complet du segment
8	1		Type de ligne définition de segment
		'L'	
12	10		Code structure (valeur ou code)
22	6		Code mouvement - sens du transfert
			Langage batch :
			Code de la Rubrique indiquant la nature du mouvement. A ce code sont associées six valeurs pour lesquelles les contrôles et les mises à jour sont automatiques :
			. mouvement 1 création,
			. mouvement 2 modification,

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
		. mouvement 3 suppression,
		. mouvement 4 modification,
		. mouvement 5 modification,
		. mouvement 6 modification.
		Si ce code est absent, toutes les mises à jour sont considérées comme des modifications (les contrôles de présence des Segments doivent être indiqués dans la zone PRESENCE MOUVEMENT 4).
		Le code mouvement et ses valeurs doivent figurer sur un des Segments de la Structure de Données, de préférence sur la partie commune 00.
28	5	Code mvt / présence: création
		(Module Langage Batch)
		VALEUR DU CODE ACTION (Segment 00)
		Contient la valeur du code mouvement 1 en création
		CONTROLE DE PRESENCE (Segments 01-99)
	O	Segment obligatoire en création
	I	Segment interdit en création
	F	Pas de contrôle (valeur par défaut)
33	5	Code mvt / présence: modification
		(Module Langage Batch)
		VALEUR DU CODE ACTION (Segment 00)
		Contient la valeur du code mouvement 2 en modification
		CONTROLE DE PRESENCE (Segments 01-99)
	O	Segment obligatoire en modification
	I	Segment interdit en modification
	F	Pas de contrôle (valeur par défaut)
38	5	Code mvt / présence: suppression
		(Module Langage Batch)
		VALEUR DU CODE ACTION (Segment 00)
		Contient la valeur du code mouvement 3 en suppression.
		CONTROLE DE PRESENCE (Segments 01-99)
	O	Segment obligatoire en suppression
	I	Segment interdit en suppression

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		F	Pas de contrôle (valeur par défaut)
43	5		Code mvt / présence: modification 4
			(Module Langage Batch)
			VALEUR DU CODE ACTION (Segment 00)
			Contient la valeur du code mouvement 4 en modification 4.
			CONTROLE DE PRESENCE (Segments 01-99)
		O	Segment obligatoire en modification 4
		I	Segment interdit en modification 4
		F	Pas de contrôle (valeur par défaut)
48	5		Code mvt / présence: modification 5
			(Module Langage Batch)
			VALEUR DU CODE ACTION (Segment 00)
			Contient la valeur du code mouvement 5 en modification 5.
			CONTROLE DE PRESENCE (Segments 01-99)
		O	Segment obligatoire en modification 5
		I	Segment interdit en modification 5
		F	Pas de contrôle (valeur par défaut)
53	5		Code mvt / présence: modification 6
			(Module Langage Batch)
			VALEUR DU CODE ACTION (Segment 00)
			Contient la valeur du code mouvement 6 en modification 6.
			CONTROLE DE PRESENCE (Segments 01-99)
		O	Segment obligatoire en modification 6
		I	Segment interdit en modification 6
		F	Pas de contrôle (valeur par défaut)
58	1		Présence mouvement 1 création
59	1		Présence mouvement 2 modification
60	1		Présence mouvement 3 suppression
61	1		Présence mouvement 4 modification
62	1		Présence mouvement 5 modification
63	1		Présence mouvement 6 modification
64	4		Nombre de postes

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
68	9		Nombre d'occurrences du segment
			Numérique
77	1		Indicateur de suite

Description (ligne 3) :

La ligne d'accès de code 3 permet de décrire un segment.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'3'	
3	4		Code complet du segment
7	3		Numéro de ligne
10	6		Code Rubrique
16	18		Libellé court de la rubrique
34	10		Format interne de la rubrique
44	1		Usage interne de la rubrique
45	3		Nombre de répétitions de la rubrique
			Numérique
48	2		Nombre de rubriques dans un groupe
50	1		Clé d'accès ou indicatif de tri
51	1		Présence rubrique en création 1
52	1		Présence rubrique en modification 2
53	1		Présence rubrique en suppression 3
54	1		Présence rubrique en modification 4
55	1		Présence rubrique en modification 5
56	1		Présence rubrique en modification 6
57	1		Classe à contrôler
58	1		Opérateur logique de contrôle
59	1		Négation
		'N'	Négation ('NOT' généré)
		blanc	Pas de négation

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
60	1		Type contrôle, mise a jour, valeur
61	10		Valeur de la rubrique à contrôler
71	2		Code fichier correspondant
73	2		Code enregistrement correspondant
75	6		Code rubrique correspondante

Sous-schémas et sous-systèmes Pactables (ligne 21) :

La ligne d'accès de code 21 permet de définir l'ensemble des sous-schémas et des sous-systèmes d'une Table.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'21'	
4	4		Code complet du segment
8	1		Type de ligne définition de segment
		'S'	Définition d'un sous-schéma
		'Y'	Définition d'un sous-système
9	1		Numéro sous-schéma ou sous-système
10	30		Nom du sous-schéma ou sous-système
40	4		Nombre de postes

Etat

Définition (ligne B) :

La ligne d'accès de code B permet de définir un état.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'B'	
3	3		Code de l'état

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
6	30		Nom de l'état imprimé
36	36		Commentaire libre de l'état
72	1		Code nature de l'état
73	1		Type de l'état
74	3		Longueur du libellé
			Numérique
77	2		Nombre de chiffres partie entière
			Numérique
79	2		Nombre de chiffres partie décimale
			Numérique

Ligne de description des libellés (ligne 4) :

La ligne d'accès de code 4 permet de décrire des libellés.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'4'	
3	3		Code de l'état
6	2		Numéro de ligne
8	2		Numéro de libellé
			Numérique
10	1		Numéro de la partie du libellé
11	1		Saut
			Numérique
12	1		Numéro de police de caractères
15	66		Libellé d'édition

Caractéristiques édition (ligne 5, type E) :

La ligne d'accès de code 5, type E permet de décrire la composition de l'édition pour les caractéristiques générales.

POS	LONGUEUR	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'5'	
3	3		Code de l'état
14	1		Type de ligne
		'E'	Valeur par défaut
15	3		Longueur de la ligne
24	1		Option
		'BLANC'	Les fonctions d'édition sont générées en fonction de la variante de matériel indiquée sur l'écran de définition de la Bibliothèque.
			Par exemple :
			Dans le cas de Bibliothèques de conversion, les fonctions d'édition sont automatiquement reformulées en fonction de la variante de la Bibliothèque.
		'N'	Interdit toute reformulation automatique de la fonction d'édition, dans une Bibliothèque de conversion.
		'*'	Option 'BEFORE' (et non pas 'AFTER') dans les ordres d'écriture de l'état.
25	2		Valeur initiale compteur de lignes
			ZONE NUMERIQUE PURE (Valeur par défaut 60).
27	4		Nb postes de la table des catégories
			ZONE NUMERIQUE PURE (Valeur par défaut 100) :
			Représente le nombre de postes de la table des catégories à réserver pour cet Etat au niveau de la génération.
			Si cette valeur est '0000', chaque ligne de chaque catégorie est éditée directement, sinon l'édition intervient après stockage des codes des catégories.
			Remarque : si cette valeur est supérieure à '1000', la table n'est pas générée.
31	2		Priorité de section
			Provoque une segmentation entre les fonctions d'édition dans un Programme. N'est à employer que si les fichiers en entrée des Programmes d'édition sont classés en majeur par code Etat, et les Programmes générés en COBOL ANS.

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
		Les priorités inférieures à 50 ne provoquent un recouvrement qu'en association avec la clause 'SEGMENT LIMIT' à insérer dans l'ENVIRONNEMENT DIVISION.
33	13	Commentaires
		Les commentaires indiqués dans l'en-tête s'appliquent à l'état. Ceux indiqués dans la partie répétitive s'appliquent à chaque ligne.
46	35	Condition d'exécution
		Les conditions indiquées dans l'en-tête s'appliquent à l'exécution de l'Etat.
		Les conditions indiquées dans la partie répétitive s'appliquent à l'exécution de la catégorie d'un Etat.
		Format de saisie :
		Les conditions s'expriment en COBOL sans 'IF', sans point, sans GO TO.

Liste des catégories (ligne 5) :

La ligne d'accès de code 5 permet de décrire la composition de l'édition pour les catégories.

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
1	1	Code mouvement
2	1	Code ligne sur 1 caractère
	'5'	
3	3	Code de l'état
6	2	Catégorie
		Ensemble de lignes éditées sous une même condition (39 lignes maximum par catégories).
		La valeur ZZ est interdite. Il est recommandé de laisser des trous dans la séquence des catégories et de ne pas commencer par AA, pour pouvoir insérer des catégories nouvelles par la suite.
8	3	Numéro de ligne
14	1	Type de ligne

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
	'A'	Cette valeur s'applique uniquement aux catégories itératives. Elle repère la première ligne d'un début de page d'une catégorie itérative (en-tête). Les en-têtes sont automatiquement édités en début de page dans un Etat. Ils sont également édités lorsque le nombre de lignes répétées (ou itératives) dépasse le nombre de lignes par page autorisé pour l'Etat, entraînant ainsi l'édition d'une nouvelle page.
	'E'	Dans les bordereaux batch, ligne décrivant l'Etat (caractéristiques générales et conditionnement).
	'T'	Repère la première ligne d'une catégorie éditée plusieurs fois , le nombre de répétitions étant un nombre indiqué à la suite (mis au préalable dans un indice indiqué à la suite ou dans l'indice standard pour la catégorie. Le nombre de catégories itératives est limité à 5 au sein d'un même écran).
	'Z'	Cette valeur s'applique exclusivement aux catégories itératives. Elle repère la première ligne d'une catégorie de fin de page, éditée automatiquement lorsque la catégorie répétée (ou itérative) dépasse le nombre de lignes par page autorisé pour l'état.
	'*'	Repère une ligne détail dans un état avec totalisation. Dans un état de ce type, la structure des lignes détail et de totalisation est commune. Les rubriques à totaliser sont repérées par le mode de remplissage 'T', les autres devant être conditionnées en fonction de la catégorie éditée (voir ligne de description des structures).
		Une catégorie contenant une ligne détail ne peut en contenir qu'une, ne peut contenir de ligne de totalisation, ne peut être itérative, peut comporter d'autres lignes ordinaires.
		La totalisation dans les compteurs générés pour chaque rubrique à totaliser est déclenchée par le conditionnement de la catégorie qui contient la ligne détail.
		Les lignes détail sont limitées à 12 par programme. 6 lignes de totalisation maximum peuvent être reprises de l'état où elles sont définies dans un autre état.
	'T'	Repère une ligne détail dans un état avec totalisation , mais à la différence de '*' la catégorie contenant cette ligne ne sera jamais éditée.
	0 à 9	Repère une ligne de totalisation générale (valeur 0), ou à un niveau de rupture donné.
		Une catégorie contenant une ligne de totalisation peut en contenir plusieurs, ne peut contenir de ligne détail, ne peut être itérative, peut comporter d'autres lignes ordinaires.

POS	CLASSE	VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
			Une totalisation au niveau général (0) ne peut être demandée sans une totalisation à un niveau de rupture (il faut qu'un niveau de rupture ait été spécifié sur un fichier sur l'écran -CD).
			La ligne détail peut être définie dans un autre état (dans ce cas son numéro de structure est réservé, mais non décrit) ; le code état et le numéro de structure de la ligne détail sont indiqués à la suite du type de ligne ; seules les rubriques totalisées seront éditées en fonction du niveau de rupture.
15	3		Caractéristiques
			Sur une ligne de totalisation (type de ligne 0 à 9), appartenant à un Etat qui ne contient pas la ligne détail correspondante, indique l'Etat et la structure de la ligne détail servant de base pour totaliser :
			. Etat de base : 1 colonne,
			. Structure de base : 2 colonnes.
			Sur la première ligne d'une catégorie répétée ou itérative (type de ligne I), définit le nombre de répétitions, sous forme d'un nombre ou d'un indice à trois positions qui doit être rempli avant l'édition.
18	2		Structure de ligne à éditer
			Numérique
20	2		Numéro de libellé
			Numérique
22	2		Saut
			Numérique
24	1		Type de saut
27	2		Code de la fonction
29	2		Code de la sous-fonction
33	13		Commentaires
			Les commentaires indiqués dans l'en-tête s'appliquent à l'état. Ceux indiqués dans la partie répétitive s'appliquent à chaque ligne.
46	35		Condition d'exécution
			Les conditions indiquées dans l'en-tête s'appliquent à l'exécution de l'Etat.
			Les conditions indiquées dans la partie répétitive s'appliquent à l'exécution de la catégorie d'un Etat.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Format de saisie :
			Les conditions s'expriment en COBOL sans 'IF', sans point, sans GO TO.

Description de structures (ligne 6) :

La ligne d'accès de code 6 permet de décrire des structures.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'6'	
3	3		Code de l'état
6	2		Structure de ligne à éditer
			Numérique
8	3		Début de la rubrique à éditer
			Numérique
11	1		Numéro de ligne pour une rubrique
12	6		Code Rubrique
18	2		Suite
		'**'	Coder '**' pour indiquer une ligne suite, sur laquelle vous pourrez indiquer la suite d'un conditionnement.
20	14		Format d'édition
			Valeur par défaut : format interne étendu ; c'est le format que prend la rubrique lorsqu'elle appartient à un état ou un écran (en affichage) ; il est automatiquement reporté dans la description des états, des écrans et des segments ; il doit être codé comme une picture COBOL ; l'usage est toujours Display ; la clause 'Blank when zero' est indiquée dans cette zone pour les reprises d'anciennes versions ; en création ou mise à jour, utiliser la zone 'Generation clause blank when zero'
			Formats symboliques pour les dates :
		'D'	Format type Display (entrée)
		'I'	Format type interne
		'E'	Format d'édition (étendu)
		'S'	Format type interne avec prise en compte du siècle

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'C'	Format type display (entrée) avec prise en compte du siècle
		'M'	Format d'édition (étendu) avec prise en compte du siècle
		'G'	Format grégorien
		'T'	Format 'Time'
		'TS'	Format 'Timestamp'
			Module méthode : ce format peut être omis pour une propriété
			Pour l'utilisation des formats avec les différents types de blocs bases de données, se reporter aux tableaux récapitulatifs du chapitre "Colonnes : rubriques" du Manuel de Référence DBD Relationnelles SQL
34	1		Mode de remplissage de la rubrique
35	1		Code type de zone source
36	2		Code structure de données source
38	2		Code enregistrement de la source
40	6		Code de la rubrique source
46	3		Indice associé a la source
49	32		Condition d'exécution

Ecran

Définition (ligne H) :

La ligne d'accès de code H permet de définir un dialogue ou un écran.

Elle a deux présentations, comportant une partie commune qui contient des renseignements généraux (nom, nombre de lignes et de colonnes, etc.) et deux parties spécifiques (la zone suite détermine la partie sélectionnée) :

- Zone suite à blanc : attributs, zones d'appel de documentation (touches fonction ou caractère particulier), caractère d'initialisation des zones de saisie ;
- Zone suite à '*' : code programme généré, nom de la map, nom de la transaction.

En général, une ligne de type blanc suffit à définir un dialogue et une ligne de type '*' comportant les noms externes suffit à définir un écran (Un écran reprenant le plus souvent les options définies au niveau du dialogue).

Cependant, les deux présentations de la ligne de code H peuvent être saisies pour définir un dialogue ou un écran.

POS	CLASSE LON VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1	Code mouvement
2	2	Code ligne sur 2 caractères
	'H '	
4	2	Code du dialogue
6	4	Code de l'écran dans le dialogue
10	30	Nom en clair de l'écran ou dialogue
40	2	Nombre de lignes dans l'écran
		Numérique
42	3	Nombre de colonnes dans l'écran
		Numérique
45	1	Présentation de la rubrique
46	2	Nombre de points de tabulation
		Numérique
48	2	Variante langage transactionnel
50	1	Option cartes avant programme
51	1	Option cartes avant grille
52	1	Option cartes après programme
53	1	Option cartes après grille
54	1	Attribut d'intensité libellé
55	1	Attribut d'intensité affichage
56	1	Attribut d'intensité saisie
57	1	Attribut d'intensité message erreur
58	1	Attribut d'intensité zone erronée
59	1	Attribut de couleur libellé
60	1	Attribut de couleur zone d'affichage
61	1	Attribut de couleur zone de saisie
62	1	Attribut de couleur message d'erreur
63	1	Attribut de couleur zone erronée
64	1	Attribut de présentation libellé
65	1	Attribut de présentation affichage
66	1	Attribut de présentation saisie

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
67	1	Attribut présentation message erreur
68	1	Attribut présentation zone erronée
70	2	Demande de documentation écran
72	2	Demande de documentation rubrique
74	1	Caractère d'initialisation
78	2	TYPE D'ECRAN
80	1	Top ligne suite

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
1	1	Code mouvement
2	2	Code ligne sur 2 caractères
	'H '	
4	2	Code du dialogue
6	4	Code de l'écran dans le dialogue
10	30	Nom en clair de l'écran ou dialogue
40	2	Nombre de lignes dans l'écran
		Numérique
42	3	Nombre de colonnes dans l'écran
		Numérique
45	1	Présentation de la rubrique
46	2	Nombre de points de tabulation
		Numérique
48	2	Variante langage transactionnel
50	1	Option cartes avant programme
51	1	Option cartes avant grille
52	1	Option cartes après programme
53	1	Option cartes après grille
54	8	Code du programme généré (nom ext.)
62	8	Code de la grille générée (nom ext.)
70	8	Code transaction (nom externe)
78	2	TYPE D'ECRAN
80	1	Top ligne suite

Complément au Dialogue (ligne H3) :

La ligne d'accès de code H3 permet la saisie des compléments d'un dialogue.

Il doit être précédé dans le flot des mouvements par une ligne d'accès de code H précisant le code du dialogue.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'H3'	
4	2		Zone commune de conversation
6	1		Organisation
7	8		Nom externe fichier libellés erreur
15	4		Code premier écran du dialogue
19	6		Code du bloc base de données
25	4	NUMER.	Complément longueur conversation
			Numérique
29	47		Zone options du dialogue

Description (ligne I) :

La ligne d'accès de code I permet de décrire un écran en batch.

Remarques :

Il doit être précédé dans le flot des mouvements par une ligne de code H précisant le code de l'écran.

Sur les codes lignes de description des écrans (codes lignes de type I), pour obtenir la remise à blanc de la zone "type de libellé", il faut indiquer le caractère ? dans la colonne 31.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'I'	

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
3	3	Numéro de ligne
6	6	Code Rubrique
12	1	Type de positionnement de rubrique
13	2	Positionnement : numéro de ligne
		Numérique
15	3	Positionnement : numéro de colonne
		Numérique
18	1	Nature de la rubrique dans l'écran
19	1	Présentation de la rubrique
20	1	Attribut d'intensité libellé
21	1	Attribut d'intensité contenu
22	1	Attribut de présentation libellé
23	1	Attribut de présentation contenu
24	1	Attribut de couleur libellé
25	1	Attribut de couleur contenu
26	1	Positionnement du curseur
27	2	Nombre de répétitions horizontales
29	2	Nombre de répétitions verticales
31	1	Présence de la rubrique
32	1	Option de traitement de la rubrique
33	1	Option de mise à jour
34	4	Code segment correspondant
38	6	Code rubrique correspondante
44	1	Code type de zone source
45	4	Code du segment source
49	6	Code de la rubrique source
60	2	Niveau des traitements générés

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
1	1	Code mouvement
2	1	Code ligne sur 1 caractère
	'I'	

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
3	3	Numéro de ligne
6	6	Code Rubrique
12	1	Type de positionnement de rubrique
13	2	Positionnement : numéro de ligne
		Numérique
15	3	Positionnement : numéro de colonne
		Numérique
18	1	Nature de la rubrique dans l'écran
19	1	Présentation de la rubrique
20	1	Attribut d'intensité libellé
21	1	Attribut d'intensité contenu
22	1	Attribut de présentation libellé
23	1	Attribut de présentation contenu
24	1	Attribut de couleur libellé
25	1	Attribut de couleur contenu
26	1	Positionnement du curseur
27	2	Nombre de répétitions horizontales
29	2	Nombre de répétitions verticales
31	1	Type d'alimentation
		Cette zone ainsi que la zone 'Libellé' sont obtenues avec le code opération C2 ; la zone 'Type d'information' indique à quoi correspond le contenu de la zone 'Libellé'
	blanc	Libellé apparaissant à l'écran
	'I'	Valeur initiale
	'P'	Valeur de présentation, utilisée uniquement pour la simulation
	'A'	Cette option n'est applicable qu'aux libellés ; elle permet d'initialiser une zone de plus de 30 caractères par la répétition d'un caractère ; la structure de la zone libellé est alors la suivante : nnnX où nnn est le nombre de répétitions désiré, et X le caractère d'initialisation
		IBM 36, IBM 38/AS 400 :
	'Y'	La zone contient le numéro d'indicateur pour le positionnement des attributs
32	30	Libellé apparaissant sur écran

Appel de Segments (ligne H2) :

La ligne d'accès de code H2 permet d'appeler des segments dans un écran.

Elle doit être précédée dans le flot des mouvements par une ligne d'accès de code H précisant le code de l'écran.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'H2'	
4	1		Indicateur de catégorie (écran)
5	4		Code segment dans l'écran
			Regroupe les deux colonnes CSDOPG (Code Structure de Données Programme) et CENRPG (Code Enregistrement dans Programme).
9	2		Numéro de ligne
11	1		Mode d'accès
		'S'	Séquentiel (option par défaut).
		'R'	Direct.
		'D'	Dynamique (Réservé à l'organisation V).
12	1		Utilisation en réception
13	1		Utilisation en affichage
14	4		Code segment précédent
18	14		Zone d'alimentation de la clé
			Regroupe les deux colonnes CSEGSR (Code Segment Source) et CRUBSR (Code Rubrique Source).
32	6		Code Rubrique
38	1		Rupture d'affichage sur la rubrique
39	1		Organisation
40	1		Type de description générée
41	8		Nom externe du fichier
49	2		Code de la structure de données
51	2		Code enregistrement
53	1		Numéro sous-schéma ou sous-système
54	2		Niveau des traitements générés

Appel de Macro-Structures (ligne M) :

Les macro-structures sont appelées à l'aide de la ligne d'accès de code M.

Cette ligne doit toujours être précédée d'une ligne de définition de programme ou d'écran car elle ne comporte pas le code programme ou écran.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'M'	
3	2		Numéro de ligne
5	1		Option ventilation des macros
		blanc	Les lignes de macros sont ventilées dans les programmes appelants lors de la mise à jour
		'N'	Pas de ventilation des lignes de macros lors de la mise à jour
6	1		Délimiteur des valeurs de paramètres
7	6		Code de la macrostructure
13	50		Commentaires, valeur des paramètres
80	1		Ligne suite

Début de programme (ligne D) :

Le début de programme est modifié à l'aide de la ligne d'accès de code D.

Cette ligne doit toujours être précédée d'une ligne de définition de programme ou d'écran car elle ne comporte pas le code programme ou écran.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'D'	
4	2		Code de la section générée
7	2		Code du paragraphe généré
9	3		Numéro de ligne
13	66		Instruction à insérer

Zone de travail et de lien (ligne 7) :

Les zones de travail et de lien sont décrites à l'aide de la ligne d'accès de code 7.

Cette ligne doit toujours être précédée d'une ligne de définition de programme ou d'écran car elle ne comporte pas le code programme ou écran.

POS	CLASSE	VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'7'	
5	2		Début numéro de ligne (emplacement)
7	3		Numéro de ligne
10	1		Type de déclaration
		blanc	Les données saisies dans les zones NIVEAU DE DECLARATION et DECLARATION DE DONNEES DE TRAVAIL sont générées telles qu'elles sont saisies
		'.'	Continuation d'un littéral
		'*'	Commentaire inséré dans le Programme généré. La dimension de la table n'est pas considérée comme un commentaire, voir la dernière zone de la ligne.
		'\$'	Cette valeur figurera dans le généré COBOL en colonne 7 et le reste de la ligne working sera repris tel quel
		'A'	Déclaration d'une Application eBusiness. Pour une documentation complète, consultez le "Guide d'utilisation de l'API COBOL".
		'F'	Appel d'une Structure de Données
			Si la ligne est créée à blanc, elle sera affichée avec un guide de saisie, les informations nécessaires correspondant exactement aux informations équivalentes de la ligne d'appel de Structures de Données pour les organisations 'W' et 'L' (code S.D. dans le Programme paramétrable dans une Macro-Structure, code de la S.D. en bibliothèque paramétrable dans une Macro-Structure, sélection d'enregistrements (non précédés d'une '*') paramétrable dans une Macro-Structure sachant que dans un Ecran il n'est pas possible de renommer un Segment, choix du type de format (I, E ou S), type de description généré (1 à 8), nombre niveau de l'enregistrement (1 à 5), organisation, (sous-schéma).

POS	LN	VALEUR	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
				Pour le batch uniquement, les lignes '-W' de type 'F' sont traitées comme des lignes d'appel de S.D. (-CD) ; si deux lignes '-W' de type 'F' concernant la même S.D. (même code S.D. dans le Programme) sont séparées, elles seront quand même générées l'une derrière l'autre
			'T'	Appel de Rubrique avec prise en compte de son format interne
			'E'	Appel de Rubrique avec prise en compte de son format d'entrée
			'S'	Appel de Rubrique avec prise en compte de son format d'édition
				Pour ces trois types d'appels, un contrôle sur l'existence dans le Dictionnaire est effectué
				Un chaînage particulier est réalisé interdisant l'annulation de cette Rubrique si les lignes où elle est appelée n'ont pas été préalablement annulées
				Si la Rubrique n'existe pas dans le Dictionnaire, un "warning" est envoyé par le système
				Dans le cas d'une substitution globale (.C2), le contrôle de la Rubrique ne sera pas fait, mais en revanche les chaînages seront bien créés
				Pour ces trois derniers types de déclaration, le nom-donnée déclaré doit être du type W-FFEE-RRRRRR avec :
				W = un préfixe quelconque (par exemple I, E ou S),
				FFEE = un code Segment quelconque,
				RRRRRR = un code Rubrique existant dans le Dictionnaire
				Ce nom-donnée doit être cadré à gauche de la zone DECLARATION
				Module Dialogue (IMS)
			'M'	Déclaration d'un Ecran comme Moniteur
			'C'	Affectation d'un Ecran à un autre Ecran déclaré comme Moniteur (type 'M')
11	17			Niveau de la déclaration
28	48			Déclaration des données de travail
76	5			Dimension table

Langage structuré (ligne P) :

Les traitements sont décrits à l'aide de la ligne d'accès de code P.

Cette ligne doit toujours être précédée d'une ligne de définition de programme ou d'écran car elle ne comporte pas le code programme ou écran.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'P'	
3	2		Code fonction
		'AA - 99'	
		'\$n'	Pour les macro-structures, le code fonction peut être paramétré
5	2		Code sous-fonction
		'AA - 99'	
		'\$n'	Pour les macro-structures, le code sous-fonction peut être paramétré
7	3		Numéro de ligne
10	3		Opérateur
13	32		Opérandes
45	2		Niveau des traitements générés
47	2		Type de structure ou de condition
49	32		Condition d'exécution

Programme

Définition (ligne 0) :

La ligne d'accès de code 0 (zéro) permet de définir un programme.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'0'	Zéro
7	6		Code programme
13	6		Code classement du programme généré
19	30		Nom en clair du programme

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
49	1		Variante du langage à générer
50	1		Ordre de mise en bibliothèque COBOL
51	1		Option numérotation et cadrage COBOL
54	1		Indicateurs SQL formatés avec '-'
55	1		Option cartes avant programme
56	1		Option cartes après programme
57	8		Code programme généré (nom externe)
65	1		Type de programmation
66	1		Nature du programme
67	1		Contrôle de présence zone numérique
68	1		Type de l'entité programme

Appel de Structures de Données (ligne 1) :

La ligne d'accès de code 1 permet l'appel de structures de données.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'1'	
3	2		Code structure de données programme
5	2		Code de la structure de données
7	6		Nom externe structure de données
13	1		Organisation
14	1		Mode d'accès
15	1		Mode de blocage
16	1		Ouverture
17	1		Type d'unité
18	5		Facteur de blocage
			Numérique
23	1		Type de blocage
24	10		Zone complément structure de données
34	6		Code de la clé dans l'enregistrement

POS	CLASSE	VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
			Clé dans l'enregistrement, obligatoire pour les fichiers en organisation indexée
40	1	NUMER.	Nombre de niveaux de rupture
			Numérique
41	1		Nombre de niveaux de synchronisation
			Numérique
42	1		Utilisation structure de données
43	6		Code Rubrique
49	2		Correspondance principal - résultat
51	2		Correspondance source - sélectionné
53	1		Niveau de rupture sur mouvement
59	4		Type d'unité physique
63	1		Complément au type d'unité physique
64	9		Argument, sélection états enregistr.
73	1		Choix du type de format
74	1		Description sélectionnée
75	1		Type de description générée
76	1		Niveau d'enregistrement COBOL
77	2		Début numéro de ligne (emplacement)
79	2		Suite de la description

Appel de Macro-Structures (ligne M) :

Les macro-structures sont appelées à l'aide de la ligne d'accès de code M.

Cette ligne doit toujours être précédée d'une ligne de définition de programme ou écran car elle ne comporte pas le code programme ou écran.

POS	CLASSE	VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'M'	
3	2		Numéro de ligne
5	1		Option ventilation des macros

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		blanc	Les lignes de macros sont ventilées dans les programmes appelants lors de la mise à jour
		'N'	Pas de ventilation des lignes de macros lors de la mise à jour
6	1		Délimiteur des valeurs de paramètres
7	6		Code de la macrostructure
13	50		Commentaires, valeur des paramètres
80	1		Ligne suite

Début de programme (ligne D) :

Le début de programme est modifié à l'aide de la ligne d'accès de code D.

Cette ligne doit toujours être précédée d'une ligne de définition de programme ou d'écran car elle ne comporte pas le code programme ou écran.

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'D'	
4	2		Code de la section générée
7	2		Code du paragraphe généré
9	3		Numéro de ligne
13	66		Instruction à insérer

Zone de travail et de lien (ligne 7) :

Les zones de travail et de lien sont décrites à l'aide de la ligne d'accès de code 7.

Cette ligne doit toujours être précédée d'une ligne de définition de programme ou d'écran car elle ne comporte pas le code programme ou écran.

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'7'	
5	2		Début numéro de ligne (emplacement)
7	3		Numéro de ligne
10	1		Type de déclaration
		blanc	Les données saisies dans les zones NIVEAU DE DECLARATION et DECLARATION DE DONNEES DE TRAVAIL sont générées telles qu'elles sont saisies
		'.'	Continuation d'un littéral
		'*'	Commentaire inséré dans le Programme généré. La dimension de la table n'est pas considérée comme un commentaire, voir la dernière zone de la ligne.
		'\$'	Cette valeur figurera dans le généré COBOL en colonne 7 et le reste de la ligne working sera repris tel quel
		'A'	Déclaration d'une Application eBusiness. Pour une documentation complète, consultez le "Guide d'utilisation de l'API COBOL".
		'F'	Appel d'une Structure de Données
			Si la ligne est créée à blanc, elle sera affichée avec un guide de saisie, les informations nécessaires correspondant exactement aux informations équivalentes de la ligne d'appel de Structures de Données pour les organisations 'W' et 'L' (code S.D. dans le Programme paramétrable dans une Macro-Structure, code de la S.D. en bibliothèque paramétrable dans une Macro-Structure, sélection d'enregistrements (non précédés d'une '*') paramétrable dans une Macro-Structure sachant que dans un Ecran il n'est pas possible de renommer un Segment, choix du type de format (I, E ou S), type de description généré (1 à 8), nombre niveau de l'enregistrement (1 à 5), organisation, (sous-schéma).
			Pour le batch uniquement, les lignes '-W' de type 'F' sont traitées comme des lignes d'appel de S.D. (-CD) ; si deux lignes '-W' de type 'F' concernant la même S.D. (même code S.D. dans le Programme) sont séparées, elles seront quand même générées l'une derrière l'autre
		'I'	Appel de Rubrique avec prise en compte de son format interne
		'E'	Appel de Rubrique avec prise en compte de son format d'entrée
		'S'	Appel de Rubrique avec prise en compte de son format d'édition

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Pour ces trois types d'appels, un contrôle sur l'existence dans le Dictionnaire est effectué
			Un chaînage particulier est réalisé interdisant l'annulation de cette Rubrique si les lignes où elle est appelée n'ont pas été préalablement annulées
			Si la Rubrique n'existe pas dans le Dictionnaire, un "warning" est envoyé par le système
			Dans le cas d'une substitution globale (.C2), le contrôle de la Rubrique ne sera pas fait, mais en revanche les chaînages seront bien créés
			Pour ces trois derniers types de déclaration, le nom-donnée déclaré doit être du type W-FFEE-RRRRRR avec :
			W = un préfixe quelconque (par exemple I, E ou S),
			FFEE = un code Segment quelconque,
			RRRRRR = un code Rubrique existant dans le Dictionnaire
			Ce nom-donnée doit être cadré à gauche de la zone DECLARATION
			Module Dialogue (IMS)
		'M'	Déclaration d'un Ecran comme Moniteur
		'C'	Affectation d'un Ecran à un autre Ecran déclaré comme Moniteur (type 'M')
11	17		Niveau de la déclaration
28	48		Déclaration des données de travail
76	5		Dimension table

Langage structuré (ligne P) :

Les traitements sont décrits à l'aide de la ligne d'accès de code P.

Cette ligne doit toujours être précédée d'une ligne de définition de programme ou d'écran car elle ne comporte pas le code programme ou écran.

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'P'	

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
3	2		Code fonction
		'AA - 99'	
		'\$n'	Pour les macro-structures, le code fonction peut être paramétré
5	2		Code sous-fonction
		'AA - 99'	
		'\$n'	Pour les macro-structures, le code sous-fonction peut être paramétré
7	3		Numéro de ligne
10	3		Opérateur
13	32		Opérandes
45	2		Niveau des traitements générés
47	2		Type de structure ou de condition
49	32		Condition d'exécution

Source COBOL (ligne FC) :

Le langage source est décrit à l'aide du code ligne 'FC'.

Cette ligne doit toujours être précédée d'une ligne de définition de programme car elle ne comporte pas le code programme.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'FC'	
4	2		Code fonction
		'AA - 99'	
		'\$n'	Pour les macro-structures, le code fonction peut être paramétré
6	2		Code sous-fonction
		'AA - 99'	
		'\$n'	Pour les macro-structures, le code sous-fonction peut être paramétré
8	3		Numéro de ligne

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
11	3	Opérateur
14	67	Ligne source reverse

COBOL pur (ligne 9) :

Le source COBOL est décrit à l'aide de la ligne d'accès de code 9.

Cette ligne doit toujours être précédée d'une ligne de définition de programme car elle ne comporte pas le code programme.

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
1	1	Code mouvement
2	1	Code ligne sur 1 caractère
	'9'	
3	6	Numéro de ligne COBOL
9	1	Ligne suite
10	65	Instruction COBOL
75	6	Fin de la ligne COBOL

Bloc Base de Données hiérarchique

Définition (ligne L1) :

La ligne d'accès de code L1 permet de définir un bloc base de données.

POS	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
NON	VALEUR	REPLISSAGE
1	1	Code mouvement
2	2	Code ligne sur 2 caractères
	'L1'	
4	6	Code du bloc base de données
10	36	Nom du bloc base de données
46	8	Nom externe du bloc
54	2	Type de bloc base de données
56	8	Nom externe du schéma
64	1	Option cartes avant bloc

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
65	1		Option cartes après bloc
66	4		Numéro de version

Description (ligne L2) :

La ligne d'accès de code L2 permet de décrire un bloc hiérarchique.

Sous-structures SOCRATE/CLIO

La même ligne d'accès est utilisée, mais seuls sont renseignés le code du bloc, le code action, le numéro de ligne et, dans la colonne réservée au code relation méthode, le code de la structure à laquelle appartient la sous-structure.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'L2'	
4	6		Code du bloc base de données
10	3		Numéro de ligne
13	4		Code du segment fils
17	4		Code du segment père
21	6		Code de la relation
27	1		Clé d'accès ou indicatif de tri
28	5		Nombre d'occurrences du lien
			Numérique
33	36		Commentaire de la ligne du bloc BD
69	6		Identifiant chemin/Turboimage
75	6		Critère de tri du chemin/Turboimage

Bloc Base de Données CodasyI

Définition (ligne L1) :

La ligne d'accès de code L1 permet de définir un bloc base de données.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'L1'	
4	6		Code du bloc base de données
10	36		Nom du bloc base de données
46	8		Nom externe du bloc
54	2		Type de bloc base de données
56	8		Nom externe du schéma
64	1		Option cartes avant bloc
65	1		Option cartes après bloc
66	4		Numéro de version

Description (ligne L3) :

La ligne d'accès de code L3 permet de décrire un bloc CODASYL, DB2, TANDEM.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'L3'	
4	6		Code du bloc base de données
10	3		Numéro de ligne
13	1		Type de ligne
		'S'	Set.
		'*'	Suite d'un set ; pour un set multi-membres, le premier segment 'Member' est indiqué sur la ligne de type 'S', les autres sont indiqués sur des lignes de type '*'
		'R'	Record
		'A'	Area
14	6		Code du set ou de l'area
20	4		Code du segment père
24	4		Code du segment fils
28	6		Code de la relation

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
34	5		Nombre d'occurrences du lien
			Numérique
39	36		Commentaire de la ligne du bloc BD

Bloc Base de Données (relationnel-SQL)

Définition (ligne L1) :

La ligne d'accès de code L1 permet de définir un bloc base de données.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'L1'	
4	6		Code du bloc base de données
10	36		Nom du bloc base de données
46	8		Nom externe du bloc
54	2		Type de bloc base de données
56	8		Nom externe du schéma
64	1		Option cartes avant bloc
65	1		Option cartes après bloc
66	4		Numéro de version

Description (ligne L4) :

La ligne d'accès de code L4 permet de décrire un bloc relationnel.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'L4'	
4	6		Code du bloc base de données
10	3		Numéro de ligne
13	1		Code structure mouvement batch SQL

POS	CLASSE	VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
14	1		Type de record SQL
		'P'	Espace (sauf pour Interel RDBC, Interel RFM, Nonstop SQL, Sybase et SQL Server)
		'T'	Table
		'V'	Vue
		'I'	Index
		'A'	Alter table : mise à jour des colonnes
		'K'	Primary Key.
			RDMS 1100 : prise en compte lors de la génération de la table qui la précède.
			DB2, Datacom/DB, SQL/DS, Oracle V6 V7, DB2/2, DB2/6000, Sybase et SQL Server : prise en compte lors de la génération par un ordre Alter table.
		'J'	Foreign key
			DB2, Datacom/DB, SQL/DS, Oracle V6 et V7, Sybase et SQL Server : prise en compte lors de la génération par un ordre Alter table
		'C'	Package (uniquement Oracle V7)
		'E'	Fonction (uniquement Oracle V7)
		'Q'	Procédure (Oracle V7, Sybase, SQL Server).
		'R'	Oracle V7, Sybase et SQL Server : Trigger
15	18		Nom externe de l'objet
33	4		Code complet du segment
37	1		Rang
41	1		Type de clé
43	1		Type de mouvement généré
44	6		Rubrique clé 1
50	1		Ordre (asc. ou des.) rubrique clé 1
51	6		Rubrique clé 2
57	1		Ordre (asc. ou des.) rubrique clé 2
58	6		Rubrique clé 3
64	1		Ordre (asc. ou des.) rubrique clé 3
65	6		Rubrique clé 4
71	1		Ordre (asc. ou des.) rubrique clé 4

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
72	6		Rubrique clé 5
78	1		Ordre (asc. ou des.) rubrique clé 5

Bloc Base de Données Turboimage

Définition (ligne L1) :

La ligne d'accès de code L1 permet de définir un bloc base de données.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'L1'	
4	6		Code du bloc base de données
10	36		Nom du bloc base de données
46	8		Nom externe du bloc
54	2		Type de bloc base de données
56	8		Nom externe du schéma
64	1		Option cartes avant bloc
65	1		Option cartes après bloc
66	4		Numéro de version

Description (ligne L2) :

La ligne d'accès de code L2 permet de décrire un bloc hiérarchique.

Sous-structures SOCRATE/CLIO

La même ligne d'accès est utilisée, mais seuls sont renseignés le code du bloc, le code action, le numéro de ligne et, dans la colonne réservée au code relation méthode, le code de la structure à laquelle appartient la sous-structure.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'L2'	
4	6		Code du bloc base de données
10	3		Numéro de ligne
13	4		Code du segment fils
17	4		Code du segment père
21	6		Code de la relation
27	1		Clé d'accès ou indicatif de tri
28	5		Nombre d'occurrences du lien
			Numérique
33	36		Commentaire de la ligne du bloc BD
69	6		Identifiant chemin/Turboimage
75	6		Critère de tri du chemin/Turboimage

Texte

Définition (ligne S) :

La ligne d'accès de code S permet de définir un texte.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	1		Code ligne sur 1 caractère
		'S'	
3	6		Code texte
9	36		Nom du texte
45	2		Type de texte
47	2		Type de division

Description (ligne T) :

La ligne d'accès de code T permet de décrire des lignes de texte.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement

POS	LONGUEUR	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
2	1		Code ligne sur 1 caractère
		'T'	
3	6		Code texte
9	2		Paragraphe du texte
11	3		Numéro de ligne
14	1		Type de ligne de texte
			Titre de division : chaque division doit contenir au moins une ligne titre ; le choix TtttttLT permet de consulter la liste des titres
		'L'	Titre : cette ligne ne sera pas éditée dans un Volume
		'K'	Titre : cette ligne sera éditée dans un Volume
		'.'	Titre : cette ligne sera soulignée par des '.' lors de l'édition dans un Volume
		'_'	Titre : cette ligne sera soulignée par des '_' lors de l'édition dans un Volume
		'+'	Titre : cette ligne sera soulignée par des '+' lors de l'édition dans un Volume
		'='	Titre : cette ligne sera soulignée par des '=' lors de l'édition dans un Volume
			Ligne de texte :
		blanc	Ligne de texte imprimée sans saut supplémentaire (valeur par défaut)
		('1' '9')	Nombre de lignes à sauter avant l'impression de la ligne de texte : la valeur '1' provoque un retour à la ligne, la valeur '2' crée une ligne blanche, la valeur '3' deux lignes blanches, etc.
		'*'	Saut de page
			Ventilation : Elle permet d'affecter un texte à des instances d'autres entités appelées dans la zone 'Ligne de texte'. Elle commence à la première ligne de la division contenant la ligne de type 'T' et s'arrête à la fin du texte ou après une ligne de type 'J'. On peut arrêter la ventilation pour une instance, toutes les instances d'une entité donnée ou toutes les entités. Le choix '-AT' permet de visualiser les textes ventilés sur l'instance d'entité. Il est possible de ventiler les textes sur les entités suivantes : 'B' (Bloc base de données), 'D' (Structure de données), 'E' (Rubrique), 'F' (Méta-entité), 'I' (Format guide), 'M' (Méthode), 'O' (Ecran), 'P' (Programme), 'Q' (Relation utilisateur), 'R' (Etat), 'S' (Segment), 'T' (Texte), 'V' (Rapport), '\$' (Entité utilisateur)

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'T'	Début de ventilation : celle-ci commence au début de la division où la ligne est insérée
		'J'	Arrêt de ventilation
		'B'	Idem 'T' avec en plus la possibilité de saisir des codes d'instances d'Entité Utilisateur dépassant 6 caractères.
		'E'	Idem 'J' avec en plus la possibilité de saisir des codes d'instances d'Entité Utilisateur dépassant 6 caractères.
		'Y'	Référence à un autre texte ou à une division de texte. Le code du texte référencé et de la division sont indiqués dans la zone 'Ligne de texte'
			Pour le texte référencé, on dispose des informations suivantes : Le choix 'TtttttXT' donne la liste des textes faisant référence au Texte 'ttttt' complet. Sur le sommaire du texte référencé, les titres de ses divisions sont suivis par les titres des divisions y faisant référence (-LT)
			Les lignes de type 'L', 'T', 'J', 'B', 'E' et 'Y' ne sont pas imprimées dans les Volumes
15	60		Ligne de texte
75	6		Code Rubrique

Document

Définition (ligne W1) :

La ligne de code W1 permet de définir un document.

POS	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'W1'	
4	6		Code du rapport
10	36		Nom en clair du rapport
46	1		Type de rapport
47	1		Page de garde du rapport
48	1		Niveau de la table des matières
49	1		Emplacement de la table des matières
50	6		Code texte

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
56	3		Code de l'état
59	3		Etat des polices de caractères
62	3		Code maquette spécifique
65	1		Gestion des niveaux du rapport

Description (ligne W2) :

La ligne de code W2 permet de décrire un document.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'W2'	
4	6		Code du rapport
10	2		Code niveau 1
12	2		Code niveau 2
14	3		Numéro de ligne
17	1		Type de ligne de description
18	1		Niveau de la section
19	1		Saut
			Numérique
20	1		Souligné du titre
21	1		Numéro de la fenêtre
22	1		Mode de cadrage
23	50		Titre, option d'édition ou sélection
73	4		Curseur de référence

Format-guide

Définition (ligne V1) :

La ligne d'accès de code V1 permet de définir un format guide.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'V1'	
4	6		Code du format guide
10	36		Nom du format guide
46	1		Type de format guide
		'C'	Commentaire
		'G'	Génération
		'O'	Option

Description (ligne V2) :

La ligne d'accès de code V2 permet de décrire un format guide.

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'V2'	
4	6		Code du format guide
10	3		Numéro de ligne
13	1		Type de ligne
		'T'	Valeur pour l'insertion d'un appel de Format-Guide.
14	20		Libellé de la ligne du format guide
34	29		Valeur initiale (ligne format guide)
63	3		Longueur du paramètre (ligne fog)
			Numérique.
66	6		Clé de chaînage
72	1		Type à affecter lors de l'appel

Méta-Entité

Définition (ligne Y1) :

La ligne d'accès de code Y1 permet de définir une Méta-Entité.

POS	LN	CLASSE ON VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'Y1'	
4	6		Code de la Méta-Entité Cliente
10	36		Libellé de la Méta-Entité Cliente
46	2		Code d'appel de la Méta-Entité

Libellés des lignes détail (ligne Y6) :

La ligne d'accès de code Y6 permet de définir les libellés des lignes détail des entités utilisateur dépendant de la méta-entité.

POS	LN	CLASSE ON VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'Y6'	
4	6		Code de la Méta-Entité Cliente
10	2		Type de Description
12	1		Type Description de la Méta-Entité
13	30		libellé de la Description de la M.E.
43	8		Code sous-programme
51	1		Mode de stockage des données
54	2		Code Description père de la M.E.

Description (ligne Y2) :

Le code ligne permettant de définir les lignes détail des entités utilisateur dépendant d'une Méta-Entité est 'Y2'.

POS	LN	CLASSE ON VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'Y2'	

POS	CLASSE	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
LONG	VALEUR	VALEUR	REPLISSAGE
4	6		Code de la Méta-Entité Cliente
10	2		Type de Description
12	3		Numéro de ligne
15	6		Code Rubrique
21	2		Rang identifiant de la Rubrique
23	1		Top nature de la Rubrique
24	1		Top transfo en majuscules
25	1		Top contrôle format de la Rubrique
26	1		Top contrôle présence de la Rubrique
27	1		Top contrôle valeur de la Rubrique
28	6		Code Relation Utilisateur Cliente
73	1		Code identifiant du père
74	1		Code identifiant appelé par Relation

Relation Utilisateur

Définition (ligne Y5) :

La ligne d'accès de code Y5 permet de définir une relation utilisateur.

POS	CLASSE	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
LONG	VALEUR	VALEUR	REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'Y5'	
4	6		Code Relation Utilisateur Cliente
10	36		Nom Relation Utilisateur Cliente
46	14		Nom réduit Rel. Utilisateur Cliente
60	1		Type Relation Utilisateur Cliente
61	3		Type d'Entité (3 caractères)
			Les valeurs autorisées sont les codes des types d'entité tels qu'ils sont définis dans le chapitre "Codification des entités DAF", sous-chapitre "Codification des tables".
64	1		Top d'annulation logique

Entité Utilisateur Cliente

Définition (ligne Y3) :

La ligne d'accès de code Y3 permet de définir une Entité Utilisateur Cliente.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'Y3'	
4	2		Code d'appel de la Méta-Entité
6	6		Code court de l'Entité Utilisateur
12	2		Rang identifiant de la Rubrique
14	1		N° mouvement pour Entité Utilisateur
		blanc	Première ligne
		'*'	Ligne suivante
15	66		Mvt Définition Entité Utilisateur

Description (ligne Y4) :

La ligne d'accès de code Y4 permet de décrire les lignes détail d'une Entité Utilisateur Cliente.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'Y4'	
4	2		Numéro de Description
6	6		Identifiant court description E.U.
12	2		Rang identifiant de la Rubrique
14	1		N° mouvement pour Entité Utilisateur
		blanc	Première ligne
		'*'	Ligne suivante
15	66		Mvt Description Entité Utilisateur

Entité Utilisateur Extension

Définition (ligne YC) :

La ligne d'accès de code YC permet de définir une Entité Utilisateur Extension.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'YC'	
4	2		Code d'appel de la Méta-Entité
6	6		Code court de l'Entité Utilisateur
12	2		Rang identifiant de la Rubrique
14	1		N° mouvement pour Entité Utilisateur
		blanc	Première ligne
		'*'	Ligne suivante
15	66		Mvt Définition Entité Utilisateur

Description (ligne YD) :

La ligne d'accès de code YD permet de décrire les lignes détail d'une Entité Utilisateur Extension.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'YD'	
4	2		Numéro de Description
6	6		Identifiant court description E.U.
12	2		Rang identifiant de la Rubrique
14	1		N° mouvement pour Entité Utilisateur
		blanc	Première ligne
		'*'	Ligne suivante
15	66		Mvt Description Entité Utilisateur

Thésaurus

Enrichissement du Thésaurus (ligne G1) :

La ligne d'accès de code G1 permet de documenter les mots-clés.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	1		Code mouvement
2	2		Code ligne sur 2 caractères
		'G1'	
4	13		Mot-clé
17	1		Top ligne suite
18	1		Type de description du mot clé
		'D'	Commentaire
		'S'	Synonyme(s)
19	55		Descriptif du mot clé

Bibliothèque

Définition (ligne X) :

La ligne d'accès de code X permet de définir une Bibliothèque.

POS	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	36		Nom en clair de la bibliothèque
37	1		Inversion jour et mois dans la date
38	1		Option cadre vertical de la doc.
39	1		Option cartes avant flot
40	1		Option cartes après flot
41	2	NUMER.	Nombre de lignes par page
			Numérique.
43	1		Option suppression des sauts de page
44	1		Option insertion des commentaires
45	1		Modification des entités extraites
46	1		Option cartes avant programme
47	1		Option cartes après programme

POS	LONG	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
48	1		Langage généré
49	1		Variante du langage à générer
50	1		Type de programmation
51	1		Protection des entités extraites
57	1		Format de la date
58	1		Caractère de séparation décimal
59	1		Variante du moniteur transactionnel
60	1		Formatage du COBOL généré
61	1		Délimiteur littéral alphanumérique
62	1		Option cadre horizontal de la doc.
63	1		option date machine avec siècle
64	2	NUMER.	année de fenêtrage
			Numérique

UPDT - Description des étapes

Vérification de l'intégrité de la base : PTUBAS

Code		Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7DS		Etat	Compte-rendu validité

. Code retour :

Cet utilitaire envoie un code retour 4 et provoque un ABEND en cas d'invalidité de la base.

Mise en forme des mouvements : PACA05

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement

Code	Nom physique	Type	Libellé
PAC7AY	Rép. base : AY	Entrée	Données extension de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7MB	Entrée utilisateur	Entrée	Mouvements de mise à jour
PAC7ME	Rép. tmp : WME	Sortie	Fichier de travail (longueur=372)
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements mis en forme (longueur=170, doit pouvoir contenir tous les mouvements en entrée plus les mouvements d'annulation élémentaire générés par les mouvements d'annulation multiple)
PAC7MW	Rép. tmp : WMW	Sortie	Fichier de travail

Mise à jour de la Base de Développement : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Sortie	Données de la Base de Développement
PAC7AN	Rép. base : AN	Sortie	Index de la Base de Développement
PAC7AY	Rép. base : AY	Sortie	Extension de la Base de Développement
PAC7AJ	Rép. journal : AJ	Sortie	Journal de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration

Code	Nom physique	Type	Libellé
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	Rép. user :RBA15	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. user :RYA15	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE	Rép. user :IEA15	Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF	Rép. user :IFA15	Etat	Récapitulatif des mouvements erronés (longueur=132)

La liste des mouvements propres à un utilisateur est précédée d'une bannière portant son code.

Codes retour :

- 0 : OK sans erreur
- 2 : erreur warning
- 4 : erreur grave

UPDT - Script d'exécution

```

|-----|
|          VISUALAGE PACBASE          |
|-----|
|          - BATCH UPDATE -          |
|-----|
|
| REFER TO THE BATCH FORMS AND TO THE DESCRIPTION OF THE
| INPUT CORRESPONDING TO EACH ENTITY.
|
| INPUT :
| - USER IDENTIFICATION LINE (REQUIRED)
|   COL 2 : "*"
|   COL 3 : USERIDXX
|   COL 11 : PASSWORD
|   COL 28 : LANGUAGE CODE, USEFUL WHEN TRANSACTION ARE
|             NOT IN THE SAME LANGUAGE AS THE DATABASE.
|   COL 67 : "N"  DEFAULT VALUE WITH EXTRACTORS
| - COMMAND LINE

```



```

'      THE LIST OF ALL AVAILABLE VALUES FOR THE ENTITY
'      TO BE UPDATED IS FOUND IN REFERENCE MANUAL.
'
' -----
'
<job id=UPDT>

<script language="VBScript">
Dim MyProc
MyProc = "UPDT"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTUBAS"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTUBAS","PAC7DS",Rep_USR & "\UPDTSBAS.txt")
Call RunCmdLog ("BVPTUBAS")
WshVolEnv("RC") = Return
If Return = 4 Then
Call Msg_Log (Array("1051"))
End If
Call Err_Cod(Return , 0 , "PTUBAS")

Call Msg_Log (Array("1022" , "PACA05"))
'-----
WshEnv("PAC7MB") = Fic_Input

'Example of Input File extracted from PACX :
' Call BvpEnv("PACA05","PAC7xx",RepT_USR & "\PAC7xx.txt")
'With RepT_USR is Global User Directory.

WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PACA05","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA05","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA05","PAC7MW",Rep_TMP & "\WMW.tmp")
Call RunCmdLog ("BVPACA05")

Call Err_Cod(Return , 0 , "PACA05")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"

```

```

WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("PACA15","PAC7DC",Rep_BASE & "\DC")
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMAADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\UPDTIEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\UPDTIFA15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA15","PAC7RB",Rep_USR & "\UPDTRBA15.txt")
Call BvpEnv("PACA15","PAC7RY",Rep_USR & "\UPDTRYA15.txt")
Call RunCmdLog ("BVPACA15")
If Return = 2 Then
Call Msg_Log (Array("1061"))
End If
If Return = 4 Then
Call Msg_Log (Array("1060"))
End If
Call Err_Cod(Return , 4 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Chapitre 6. Pactables

GETD-GETA - Générateurs de descriptifs

GETD-GETA - Présentation générale

Le Générateur de descriptifs de tables est une interface entre le dictionnaire et le module Pactables.

(Se reporter au schéma de mise en oeuvre du sous-chapitre "Le module Pactables" dans l'introduction du manuel "Pactables").

Son utilisation est soumise à un contrat d'acquisition du module Pactables.

Cette interface extrait de la base de spécifications les descriptions de tables nécessaires au fonctionnement du module Pactables.

L'extraction est effectuée par deux procédures, GETA et GETD. Le choix de la procédure utilisée dépend de l'environnement d'installation de Pactables :

- GETA lorsque le dictionnaire et Pactables sont dans le même environnement,
- GETD si le dictionnaire et Pactables sont dans des environnements disjoints. Dans ce dernier cas, la procédure fonctionne avec un fichier des descriptifs de tables, image du fichier de descriptifs utilisés par le module Pactables. Par conséquent, à la première utilisation de cette procédure, ce fichier doit avoir été initialisé :
 - soit par recopie du fichier des descriptifs du module Pactables s'il existe,
 - soit par la procédure d'initialisation décrite dans le présent chapitre (GETI).

La procédure (GETA ou GETD) produit un fichier interface qui doit être repris en entrée de la procédure GETT du module Pactables (se reporter au manuel d'exploitation Pactables).

Condition d'exécution

Aucune par rapport à la base de spécifications, celle-ci étant seulement consultée par la procédure.

Anomalies d'exécution

Si la génération se termine anormalement avant mise à jour du fichier des descriptifs, la procédure peut être relancée telle quelle après suppression de la cause de l'anomalie.

Si la génération se termine anormalement en cours de mise à jour du fichier des descriptifs, ce fichier doit être restauré avant de relancer la procédure.

GETD-GETA - Entrées Utilisateur / Résultat

Entrées utilisateur

Une ligne '*' indiquant la bibliothèque contenant les descriptions de tables :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	bbb	Code bibliothèque
22	4	nnnn	Numéro de session
26	1	t	Etat de la session

Une ligne 'Z' par demande de génération ou édition :

Pos.	Lon.	Valeur	Signification
2	1	'Z'	Code ligne
5	4		Code de la demande :
		'TGS '	Demande de génération de descriptif
		'TDS '	Demande d'édition de descriptif
		'TLS '	Demande de liste de descriptifs
		'TAS '	Demande d'annulation de table
		'TMS '	Demande de modification des caractéristiques d'historique d'une table
		'TGC '	Demande de génération de commentaires
9	6	ssss	Code segment de la description de la table à extraire ('TGS ', 'TGC ')
		ttttt	Code de la table (autres demandes)
15	2	' '	sans signification
17	8	jmmssaa	Date à laquelle il sera possible de modifier le descriptif de la table (facultatif)

Pos.	Lon.	Valeur	Signification
25	8	jimmssaa	Date de l'historique du descriptif pour une table de type G. Le dernier historique est pris par défaut
		*****	Pour la génération d'une table sans historique
33	1		Type de format des rubriques :
		' '	format interne
		'E'	format d'entrée
75	6	ttttt	Numéro table (si génération pour une autre Table que celle de la fiche segment dans la base)

(Se reporter au manuel "Pactables" pour plus de détails sur ces entrées).

Remarque : Il n'est pas possible de modifier la clé d'une table. Les demandes de génération de tables déjà définies qui impliquent cette modification sont rejetées.

Résultat obtenu

Le résultat de cette procédure est un fichier séquentiel des descriptifs de tables qui doit être repris en entrée de la procédure GETT du module Pactables.

GETD-GETA - Description des étapes

Extraction et préparation de mise à jour : PACT40

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7AY	Rép. base : AY	Entrée	Données extension de la Base de Développement
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7TD	Rép. base : TD	Entrée	Descriptifs de tables
PAC7MB	Entrée utilisateur	Entrée	Demandes de descriptifs

Code	Nom physique	Type	Libellé
PAC7MD	Rép. user : MVGETD ou MVGETA	Sortie	Mouvements de mise à jour descriptifs version supérieure ou égale à 2.0
PAC7ET	Rép. user : GETDETT40 ou GETAETT40	Etat	Compte-rendu des mouvements
PAC7DD	Rép. user : GETDDDT40 ou GETADDT40	Etat	Contrôle autorisation

Code retour :

- 8 : pas d'autorisation procédure batch.

Mise en forme des descriptifs < 2.0 : PACT45

Code	Nom physique	Type	Libellé
PAC7MD	Rép. user : MVGETD ou MVGETA	Entrée	Mouvements de mise à jour descriptifs version supérieure ou égale à 2.0
PAC7ND	Rép. user : NDGETD ou NDGETA	Sortie	Mouvements de mise à jour descriptifs version inférieure à 2.0

Mise à jour des descriptifs de tables : PACT50

Cette étape est réservée à la procédure GETD seulement.

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7TD	Rép. base : TD	Entrée	Descriptifs de tables
PAC7MD	Rép. user : MVGETD	Entrée	Mouvements de mise à jour
PAC7ET	Rép. user : GETDETT50	Etat	Compte-rendu de mise à jour

GETD - Script d'exécution

```

|-----|
| VISUALAGE PACBASE |
|-----|
|           - TABLE DESCRIPTIONS GENERATION -           |
|-----|

```

```

'
' TABLE DESCRIPTION GENERATOR IS THE INTERFACE BETWEEN
' THE SPECIFICATIONS DICTIONARY AND VA PACTABLES.
' FOR FURTHER INFORMATION, REFER TO CHAPTER GENERAL
' INTRODUCTION SUBCHAPTER INTRODUCTION TO THE
' VA PACTABLES FACILITY IN THE VA PACTABLES MANUAL.
'
' GETD IF THE DICTIONARY AND VA PACTABLES ARE RUNNING
' UNDER DIFFERENT ENVIRONMENTS.
'
' -----
'
<job id=GETD>

<script language="VBScript">
Dim MyProc
MyProc = "GETD"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

If FSO.FileExists(Rep_BASE & "\TD") Then

Call Msg_Log (Array("1022" , "PACT40"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACT40","PAC7MD",Rep_USR & "\Mvgetd.txt")
Call BvpEnv("PACT40","PAC7ET",Rep_USR & "\GETDETT40.txt")
Call BvpEnv("PACT40","PAC7DD",Rep_USR & "\GETDDDT40.txt")
Call RunCmdLog ("BVPACT40")
If Return = 8 Then
Call Msg_Log (Array("1027"))
End If
Call Err_Cod(Return , 0 , "PACT40")

Call Msg_Log (Array("1022" , "PACT45"))
'-----
Call BvpEnv("PACT45","PAC7MD",Rep_USR & "\Mvgetd.txt")
Call BvpEnv("PACT45","PAC7ND",Rep_USR & "\Ndgetd.txt")
Call RunCmdLog ("BVPACT45")
Call Err_Cod(Return , 0 , "PACT45")

Call Msg_Log (Array("1022" , "PACT50"))

```

```

'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PACT40","PAC7MD",Rep_USR & "\Mvgetd.txt")
Call BvpEnv("PACT50","PAC7ET",Rep_USR & "\GETDETT50.txt")
Call RunCmdLog ("BVPACT50")
Call Err_Cod(Return , 0 , "PACT50")

Else

Call Msg_Log (Array("1044" , Rep_BASE & "\TD"))

End If

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

GETA - Script d'exécution

```

'-----
'          VISUALAGE PACBASE
'-----
'          - TABLES DESCRIPTION GENERATION -
'-----
'
' TABLE DESCRIPTION GENERATOR IS THE INTERFACE BETWEEN
' THE SPECIFICATIONS DICTIONARY AND VA PACTABLES.
' FOR FURTHER INFORMATION, REFER TO CHAPTER GENERAL
' INTRODUCTION SUBCHAPTER INTRODUCTION TO THE
' VA PACTABLES FACILITY IN THE VA PACTABLES MANUAL.
'
' GETA IF THE DICTIONARY AND VA PACTABLES ARE RUNNING
' UNDER THE SAME ENVIRONMENTS.
'-----
<job id=GETA>

<script language="VBScript">
Dim MyProc
MyProc = "GETA"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

```



```

If FSO.FileExists(Rep_BASE & "\TD") Then

Call Msg_Log (Array("1022" , "PACT40"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACT40","PAC7MD",Rep_USR & "\Mvgeta.txt")
Call BvpEnv("PACT40","PAC7ET",Rep_USR & "\GETAETT40.txt")
Call BvpEnv("PACT40","PAC7DD",Rep_USR & "\GETADDT40.txt")
Call RunCmdLog ("BVPACT40")
If Return = 8 Then
Call Msg_Log (Array("1027"))
End If
Call Err_Cod(Return , 0 , "PACT40")

Call Msg_Log (Array("1022" , "PACT45"))
'-----
Call BvpEnv("PACT45","PAC7MD",Rep_USR & "\Mvgeta.txt")
Call BvpEnv("PACT45","PAC7ND",Rep_USR & "\Ndgeta.txt")
Call RunCmdLog ("BVPACT45")
Call Err_Cod(Return , 0 , "PACT45")

Else

Call Msg_Log (Array("1044" , Rep_BASE & "\TD"))

End If

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

GETI - Initialisation de descriptifs

GETI - Présentation générale

Cette procédure permet, lors de la première utilisation des fichiers tables, lorsque ceux-ci sont disjoints de la Base de développement, d'initialiser le fichier des descriptifs de manière analogue à la procédure INTA du module Pactables.

GETI - Entrées Utilisateur

Une ligne '*' avec Code utilisateur et Mot de passe.

Une ligne de prise en compte de paramètres d'initialisation

Pos.	Lon.	Valeur	Signification
2	1	'I'	Code ligne
3	36		Libellé de l'installation
39	1		Code langue
		'F'	Français (option par défaut)
		'E'	Anglais
53	4	cccc	Classe pour système de sécurité
57	1		Type de système de sécurité
		'R'	RACF
		'S'	Top secret
58	2	nn	Nombre de lignes par page d'édition
60	1		Type de contrôle de ressources
		' '	Définition ressources tables du système de sécurité
		'P'	Définition des ressources dans la Base de développement
61	1		Blocage du code utilisateur
		' '	Possibilité autre code utilisateur
		'N'	Interdiction autre code utilisateur

GETI - Description des étapes

Initialisation du fichier des descriptifs : PACTIN

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Fichier des données de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Ligne paramètre

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Sortie	Descriptifs de tables
PAC7ED	Rép. user : GETIEDTIN	Etat	Compte-rendu d'initialisation
PAC7DD	Rép. user : GETIDDTIN	Etat	Contrôle autorisation

GETI - Script d'exécution

```

'-----
'          VISUALAGE PACBASE
'-----
' - INITIALIZATION OF TABLES MANAGEMENT FILE -
'-----
'
' THE GETI PROCEDURE MUST BE EXECUTED WHEN FIRST USING
' VA PACTABLES FILES THAT ARE STORED IN ANOTHER
' ENVIRONMENT FROM THE PRODUCT ENVIRONMENT.
' IT INITIALIZES THE DESCRIPTION FILE IN A SIMILAR WAY
' AS THE VA PACTABLES INTA PROCEDURE DOES.
'-----
'
<job id=GETI>

<script language="VBScript">
Dim MyProc
MyProc = "GETI"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PACTIN"))
'-----
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACTIN","PAC7ED",Rep_USR & "\GETIEDTIN.txt")
Call BvpEnv("PACTIN","PAC7DD",Rep_USR & "\GETIDDTIN.txt")
Call RunCmdLog ("BVPACTIN")
Call Err_Cod(Return , 0 , "PACTIN")

```

```

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

SMTD-RMTD - Migration des descriptifs de tables

SMTD - Présentation générale

La procédure SMTD effectue une sauvegarde du fichier TD des descriptifs de tables en transformant les caractères binaires en caractères étendus.

L'objectif est de pouvoir transférer le fichier TD sur différentes plate-formes et d'éviter les problèmes liés à l'interprétation de ces caractères lors des transferts.

Condition d'exécution

Aucune.

SMTD - Description des étapes

Sauvegarde de TD (fichier de description des tables) : PTASVD

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7MB	Entrée utilisateur	Entrée	Mouvements utilisateur
PAC7TC	Rép. save : PD-new	Sortie	Sauvegarde descriptifs de tables pour migration
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7DD		Etat	Contrôle autorisation

SMTD - Script d'exécution

```

'-----
' VISUALAGE PACBASE
'

```

```

' -----
'   - BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION -
' -----
'
' THE SMTD PROCEDURE BACKS UP THE TD TABLE-DESCRIPTION
' FILE BY TRANSFORMING BINARY CHARACTERS INTO THEIR
' DISPLAY FORMAT.
' -----
'
<job id=SMTD>

<script language="VBScript">
Dim MyProc
MyProc = "SMTD"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTASVD"))
'-----
WshEnv("PAC7TC") = Rep_SAVE & "\PD-new"
Call BvpEnv("PTASVD","PAC7TD",Rep_BASE & "\TD")

WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTASVD","PAC7DD",Rep_USR & "\SMTDDDSVD.txt")

Call RunCmdLog ("BVPTASVD")
Call Err_Cod(Return , 0 , "PTASVD")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\PD")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr (Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

RMTD - Présentation générale

La procédure RMTD effectue une restauration du fichier TD des descriptifs de tables à partir de sa sauvegarde séquentielle TC obtenue par la procédure SMTD.

Cette procédure ne requiert aucune condition d'exécution particulière.

RMTD - Description des étapes

Restauration de TD : PTARSD

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7MB	Entrée utilisateur	Entrée	Mouvements utilisateur
PAC7TD	Rép. save : PD	Sortie	Fichier des descriptifs de Tables
PAC7TC	Rép. base : TD	Entrée	Fichier séquentiel des descriptifs de tables
PAC7DD		Etat	Contrôle autorisation

RMTD - Script d'exécution

```
|-----|
| VISUALAGE PACBASE |
|-----|
| - RESTORATION OF TABLE DESCRIPTIONS - |
|-----|
| THE RESTORATION OF TABLE DESCRIPTIONS PROCEDURE |
| (RMTD) RESTORES THE TD FILE OF |
| TABLE DESCRIPTIONS FROM ITS TC SEQUENTIAL BACKUP |
| PRODUCED BY THE SMTD PROCEDURE. |
|-----|
|
|<job id=RMTD>
|
|<script language="VBScript">
| Dim MyProc
```

```

MyProc = "RMTD"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTARSD"))
'-----
WshEnv("PAC7TC") = Rep_SAVE & "\PD"
If Not FSO.FileExists(WshEnv("PAC7TC")) Then
  Call Msg_Log (Array("1004" , "PAC7TC"))
  Msg = Nls_Lib
  EndJob (1)
  Wscript.Quit (1)
End If
WshEnv("PAC7TD") = Rep_BASE & "\TD"

WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTARSD","PAC7DD",Rep_USR & "\RMTDDDRSD.txt")

Call RunCmdLog ("BVPTARSD")
Call Err_Cod(Return , 0 , "PTARSD")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\PD")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr (Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

Chapitre 7. Pac/Impact

Préambule

Remarque

Les utilisateurs de Pac/Impact peuvent se reporter au manuel de référence correspondant pour plus de détails.

Avertissement

L'analyse d'impact est très consommatrice de temps machine : il est donc recommandé de limiter l'étendue de l'analyse.

Il est possible de limiter l'analyse à deux niveaux différents ou de cumuler les deux afin d'obtenir un domaine d'analyse encore plus précis.

La procédure UXSR, documentée dans le manuel "Les procédures de l'Administrateur", vous permet de créer une nouvelle image de la Base de développement en 'zoomant' sur un sous-réseau (l'extraction de session est aussi disponible).

Il en résulte la création d'une nouvelle base qui est un sous-ensemble (restructuré et/ou renommé) de la base initiale. L'analyse est effectuée sur ce sous-ensemble.

En outre, avec la procédure REOR (toujours obligatoire après UXSR), vous pourrez supprimer les occurrences inutiles à l'analyse.

Vous pouvez aussi choisir de limiter votre analyse aux occurrences des entités programme, écran ou bloc base de données. Des options de sélection complémentaires sont disponibles.

Cette limitation de l'analyse est effectuée par l'utilitaire INFP documenté dans ce manuel.

Les procédures de ce module n'ont aucune influence sur les fichiers de la base. Il est cependant conseillé de fermer les fichiers conversationnels pour de meilleures performances.

INFP - Initialisation du fichier FP

INFP - Présentation générale

Cette procédure permet de déclarer les entités sur lesquelles portera l'analyse d'impact.

Pour que le fichier FP soit mis à jour par la procédure, il convient de ré-indiquer en entrée de celle-ci toutes les lignes déjà introduites auparavant, car on part toujours d'un fichier vide.

Résultat obtenu

On obtient un fichier des entités sélectionnées pour l'analyse (FP).

INFP - Entrées Utilisateur

Une ligne '*' avec Code utilisateur et Mot de passe.

Les autres entrées sont facultatives, sachant que si aucune n'est introduite, toutes les entités de tous les types seront recherchées dans l'analyse d'impact.

Si la totalité des entités d'un type d'entité est demandée (code entité égal à *****), les entités spécifiques pour ce même type seront refusées.

La valorisation d'un type sur une entrée (que l'on précise ou non une entité pour ce type) implique la valorisation des autres types que l'on souhaite analyser sur des entrées supplémentaires.

Pos.	Lon.	Valeur	Signification
1	3		Type d'entité
		'B '	Blocs bases de données
		'F '	Méta-entités
		'O '	Ecrans
		'P '	Programmes
		'T '	Textes
		'V '	Documents
		\$nn	seulement EU de type nn
		'\$**'	toutes les EU
4	6		Code de l'entité (généricité possible avec le code *****)
			ce code peut ne pas exister dans la base

INFP - Description des étapes

Contrôle des mouvements et mise à jour FP : PAN205

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7FP	Rép. base : FP	Sortie	Entités en production
PAC7IP	Rép. user : INFPIP205	Etat	Compte-rendu de contrôle

Codes retour :

- 0 : OK.
- 12 : Erreur système

INFP - Script d'exécution

```
|-----|
|          VISUALAGE PACBASE          |
|-----|
|          - IMPACT ANALYSIS:  INITIALIZATION OF "FP" FILE -          |
|-----|
|
| THE INFP PROCEDURE INITIALIZES THE FP FILE. IT ALLOWS
| TO SPECIFY THE ENTITIES WHICH ARE TO BE ANALYZED AND
| THUS TO NARROW THE SCOPE OF THE IMPACT ANALYSIS TO SOME
| (OR ALL) OCCURRENCES OF THE ENTITIES.
|
|-----|
|
|<job id=INFP>
|
|<script language="VBScript">
| Dim MyProc
| MyProc = "INFP"
|</script>
|
|<script language="VBScript" src="INIT.vbs"/>
```

```

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PAN205"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7FP") = Rep_BASE & "\FP"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PAN205","PAC7IP",Rep_USR & "\INFPIP205.txt")
Call RunCmdLog ("BVPAN205")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN205"))
End If
Call Err_Cod(Return , 0 , "PAN205")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr (Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

ISEP - Sélection des points d'entrée

ISEP - Présentation générale

Cette procédure permet de sélectionner tous les points d'entrée (Rubriques et/ou chaînes de caractères) qui seront utilisés pour effectuer l'analyse d'impact (procédure IANA).

La ligne d'identification du contexte de sélection (ligne *) est obligatoire et permet de préciser, outre la session, la bibliothèque à partir de laquelle on veut construire le sous-réseau retenu pour l'analyse (vue Z1).

Les Rubriques ou les chaînes de caractères sont sélectionnées en tant que points d'entrée lorsqu'elles correspondent aux critères de sélection spécifiés dans les lignes d'entrées utilisateur de ISEP.

Trois types de critères peuvent être utilisés (voir ci-dessous) et au moins un critère de sélection est obligatoire (aucun type particulier de critère n'est exigé).

La sélection peut comprendre plusieurs types de critères et plusieurs lignes de commande pour chaque type.

- La ligne de type 'E' permet d'extraire des Rubriques en sélectionnant un code (code générique autorisé) et/ou un ou plusieurs formats.
- La ligne de type 'S' permet d'extraire des chaînes de caractères en sélectionnant un code (code générique autorisé) et/ou un ou plusieurs formats.
- La ligne de type 'W' permet de sélectionner des Rubriques à l'aide d'un mot-clé. Le type de mot clé, les formats de Rubrique et le code peuvent être aussi précisés.

Condition d'exécution

Aucune.

Anomalies d'exécution

Quelle que soit la cause d'une fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

Résultat obtenu

On obtient un fichier des points d'entrée sélectionnés (FH) et un fichier des points d'entrée à épurer (FR) destinés à la procédure IANA.

ISEP - Entrées Utilisateur

Une seule ligne '*' (obligatoire et en tête du flot) :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code de la ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	bbb	Code de la bibliothèque supérieure du sous-réseau
22	4	ssss	Numéro de la session utile (blanc si session courante)
26	1		Etat de la session (' ' ou 'T')
69	3	iii	Code de la bibliothèque inférieure du sous-réseau (facultatif)

Une ligne 'E': sélection de Rubriques (facultative) :

Pos.	Lon.	Valeur	Signification
2	1	'E'	Code ligne

Pos.	Lon.	Valeur	Signification
3	6		Code de la Rubrique (généricité possible avec le caractère '*' en début ou en fin du code (***) ou xxx) ou avec le caractère ? suivi de la chaîne que l'on veut voir figurer dans le code (?xxx)
9	10		Format d'entrée de la Rubrique
19	10		Format interne de la Rubrique
29	1		Usage interne (D par défaut)
30	27		Format édition de la Rubrique
57	1	'N'	Pour ne pas impacter les Rubriques filles, blanc sinon

Une ligne 'S': sélection de chaînes de caractères (facultative) :

Pos.	Lon.	Valeur	Signification
2	1	'S'	Code ligne
3	30		Code de la chaîne (généricité possible avec le caractère '*' n'importe où dans le code), ou ?xx où xx est une chaîne pouvant se trouver n'importe où
33	10		Format interne de la chaîne
43	1		Usage interne (D par défaut)

Une ligne 'W' : sélection sur mots-clés (facultative) :

Pos.	Lon.	Valeur	Signification
2	1	'W'	Code ligne
3	1		Type de mot-clé (implicite 'L', explicite 'M', les deux '')
4	13		Code du mot-clé (pas de généralité)
17	10		Format entrée de la Rubrique
27	10		Format interne de la Rubrique
37	1		Usage interne (D par défaut)
38	27		Format édition de la Rubrique
65	6		Code de la rubrique (généricité possible avec le caractère '*', n'importe où dans le code)
71	1	'N'	Pour ne pas impacter les rubriques filles, blanc sinon

ISEP - Description des étapes

Sélection des points d'entrée : PAN210

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7FP	Rép. base : FP	Entrée	Entités en production
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7FH	Rép. tmp : WFH	Sortie	Points d'entrée sélectionnés
PAC7IE	Rép. user : ISEPIE210	Etat	Compte-rendu de contrôle

Codes retour :

- 0 : OK.
- 12 : Erreur système

Elimination des points d'entrée en double : PAN215

Code	Nom physique	Type	Libellé
PAC7FH	Rép. tmp : WFH	Entrée	Points d'entrée sélectionnés
PAC7HF	Rép. base : FH-new	Sortie	Points d'entrée sélectionnés triés
PAC7FR	Rép. base : FR-new	Sortie	Points d'entrée réduits à épurer

- Codes retour :
 - 0 : OK.
 - 12 : Erreur système

Rotation de fichier

Le fichier NEW est créé. Pour continuer l'analyse d'impact, il faut opérer une rotation de fichier pour obtenir le fichier courant.

FH-NEW --(rotation)--> FH , l'ancien FH devient FH-1.

FR-NEW --(rotation)--> FR , l'ancien FR devient FR-1.

ISEP - Script d'exécution

```
-----
|          VISUALAGE PACBASE
|-----
|          - IMPACT ANALYSIS : SELECTION OF ENTRY POINTS -
|-----
|
| THE ISEP PROCEDURE IS DESIGNED TO SELECT THE ENTRY
| POINTS -- DATA ELEMENTS AND/OR CHARACTER STRINGS --
| WHICH WILL BE USED AS CRITERIA BY THE IMPACT
| ANALYSIS (IANA PROCEDURE).
|-----
|
<job id=ISEP>

<script language="VBScript">
Dim MyProc
MyProc = "ISEP"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PAN210"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7FP") = Rep_BASE & "\FP"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PAN210","PAC7FH",Rep_TMP & "\WFH.tmp")
Call BvpEnv("PAN210","PAC7IE",Rep_USR & "\ISEPIE210.txt")
Call RunCmdLog ("BVPAN210")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN210"))
End If
Call Err_Cod(Return , 0 , "PAN210")
```



```

Call Msg_Log (Array("1022" , "PAN215"))
'-----
Call BvpEnv("PAN215","PAC7FH",Rep_TMP & "\WFH.tmp")
WshEnv("PAC7HF") = Rep_BASE & "\FH-new"
WshEnv("PAC7FR") = Rep_BASE & "\FR-new"
Call RunCmdLog ("BVPAN215")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN215"))
End If
Call Err_Cod(Return , 0 , "PAN215")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_BASE & "\FH")
Call Turnover(Rep_BASE & "\FR")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr(Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

ISOS - Sélection des chaînes et des opérateurs

ISOS - Présentation générale

Cette procédure est un complément à la procédure ISEP.

Elle permet de sélectionner :

- Les dates gérées par le produit (DATOR, DAT8 par exemple) qui seront utilisées comme points d'entrée pour effectuer l'analyse d'impact dès la première itération (procédure IANA).
- Les chaînes de caractères sans les considérer comme des points d'entrée (ORDER BY par exemple). Pour les chaînes donnant lieu à des points d'entrée voir ISEP (ligne S).
- Les opérateurs utilisés dans les lignes de traitement -P (ADT par exemple). Certains d'entre eux déclenchent la génération de points d'entrée de type date (DATOR pour ADT).
- Les lignes utilisant une valeur constante, soit en définition (Value), soit en transfert ou en condition.

La restitution des entités utilisant les opérateurs et les chaînes de caractères sera exécutée sur demande (procédure IPIA).

Limitation du domaine de sélection

Pour de meilleurs résultats, il est recommandé de limiter le domaine de sélection. Ceci peut être fait à deux niveaux différents, dans tous les cas avant l'exécution de la procédure ISOS :

- Via la procédure UXSR (documentée dans le manuel "Les procédures de l'administrateur"), vous pouvez créer une nouvelle Base de développement. La nouvelle base sera un sous-ensemble (restructuré et/ou renommé) de la base initiale. L'analyse est effectuée sur ce sous-ensemble.
- Via l'utilitaire INFP (documenté dans ce manuel), vous pouvez décider de limiter le domaine de sélection aux occurrences souhaitées pour les types d'entité de votre choix. Des options de sélection complémentaires sont disponibles.

La ligne d'identification du contexte de sélection (ligne *) est obligatoire et permet de préciser, outre la session, la bibliothèque à partir de laquelle on veut construire le sous-réseau retenu pour l'analyse (vue Z1).

Trois types de sélection peuvent être utilisés (voir ci-dessous) et au moins un type de sélection est obligatoire (aucun type particulier de sélection n'est exigé).

La sélection peut comprendre plusieurs types de sélection et plusieurs lignes de commande pour chaque type.

- La ligne de type 'D' permet d'extraire des rubriques de type date gérées par le produit. Le nombre de lignes 'D' est limité à 40.
- La ligne de type 'C' permet d'extraire des chaînes de caractères pouvant comporter un ou plusieurs blancs. Le délimiteur doit être précisé dans ce cas et le nombre de blancs a son importance. Ces chaînes ne sont aucunement des points d'entrée. Le nombre de lignes 'C' est limité à 50 par localisation.
- La ligne de type 'O' permet de sélectionner des opérateurs gérés dans les lignes de traitement (-P). Le nombre de lignes 'O' est limité à 50.

Condition d'exécution

Aucune.

Anomalies d'exécution

Quelle que soit la cause d'une fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

Résultat obtenu

On obtient un fichier des points d'entrée sélectionnés (FH) et un fichier des points d'entrée à épurer (FR) destinés à la procédure IANA, ainsi qu'un fichier des résultats d'analyse (FO) à destination de la procédure IANA ou IPIA.

ISOS - Entrées Utilisateur

Une seule ligne '*' (obligatoire et en tête du flot) :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code de la ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	bbb	Code de la bibliothèque supérieure du sous-réseau
22	4	ssss	Numéro de la session utile (blanc si session courante)
26	1		Etat de la session (' ' ou 'T')
28	1	'F' ou 'E'	Code langue si différent de celle du site (uniquement pour site bilingue)
69	3	iii	Code de la bibliothèque inférieure du sous-réseau (facultatif)

Une ligne 'D': sélection de dates générées (facultative) :

Pos.	Lon.	Valeur	Signification
2	1	'D'	Code ligne
3	9		Code de la rubrique date générée à extraire (elle doit être connue du système)

Une ligne 'O' : sélection d'opérateurs (facultative) :

Pos.	Lon.	Valeur	Signification
2	1	'O'	Code ligne
3	3		Code de l'opérateur recherché (il doit être connu du système)

Une ligne 'C': sélection de chaînes de caractères (facultative) :

Pos.	Lon.	Valeur	Signification
2	1	'C'	Code ligne
3	1		Délimiteur de fin de chaîne (obligatoire dès lors que la chaîne contient au moins un blanc)

Pos.	Lon.	Valeur	Signification
4	31		Code de la chaîne recherchée (doit être terminé par le délimiteur dès lors que celui-ci est spécifié)
35	1		Lieu de recherche de la chaîne
		'D'	Recherche dans la partie définition (-W des programmes et/ou écrans et -9 des programmes)
		'T'	Recherche dans la partie traitement (-P des programmes et/ou écrans, -8, -9, -SC des programmes, -CE et -CS des écrans)
		'R'	Recherche dans la partie traitement spécifique aux états : condition catégorie et structure ainsi que code rubrique source (structure)
		' '	Recherche dans les trois parties précitées

Une ligne 'V' : sélection de valeurs constantes (facultative) :

Pos.	Lon.	Valeur	Signification
2	1	'V'	Code ligne
3	1	' ou "	Délimiteur de début de valeur, obligatoire
4	31		Code de la valeur recherchée, obligatoire et terminé par le délimiteur (soit ', soit ")
35	1		Lieu de recherche de la constante
		'D'	Recherche dans la partie définition (-W des programmes et/ou écrans et -9 des programmes)
		'T'	Recherche dans la partie traitement (-P des programmes et/ou écrans, -8, -9, -SC des programmes, -CE et -CS des écrans)
		'R'	Recherche dans la partie traitement spécifique aux états : condition catégorie et structure ainsi que code rubrique source (structure)
		' '	Recherche dans les trois parties précitées

ISOS - Description des étapes

Sélection des chaînes et des opérateurs : PAN212

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration

Code	Nom physique	Type	Libellé
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7FP	Rép. base : FP	Entrée	Entités à analyser
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7FH	Rép. tmp : WFH	Sortie	Points d'entrée sélectionnés (longueur=160)
PAC7MF	Rép. tmp : WFO	Sortie	Résultats d'analyse d'impact (longueur=266)
PAC7IE	Rép. user : ISOSIE212	Etat	Compte-rendu de contrôle

Codes retour :

- 0 : OK.
- 12 : Erreur système

Elimination des points d'entrée en double : PAN215

Code	Nom physique	Type	Libellé
PAC7FH	Rép. tmp : WFH	Entrée	Points d'entrée sélectionnés
PAC7HF	Rép. base : FH-new	Sortie	Points d'entrée sélectionnés triés
PAC7FR	Rép. base : FR-new	Sortie	Points d'entrée réduits à épurer

Codes retour :

- 0 : OK.
- 12 : Erreur système

Mise à jour des résultats d'analyse d'impact : PAN260

Code	Nom physique	Type	Libellé
PAC7MF	Rép. tmp : WFO	Entrée	Résultat analyse d'impact (de l'itération)

Code	Nom physique	Type	Libellé
PAC7OF	Rép. base : FO	Entrée	Résultats de l'analyse précédente
PAC7FO	Rép. base : FO-new	Sortie	Résultats d'analyse d'impact triés

Codes retour :

- 0 : OK.
- 12 : Erreur système

Rotation de fichier

Le fichier NEW est créé. Pour continuer l'analyse d'impact, il faut opérer une rotation de fichier pour obtenir le fichier courant.

FH-NEW --(rotation)--> FH , l'ancien FH devient FH-1.

FR-NEW --(rotation)--> FR , l'ancien FR devient FR-1.

FO-NEW --(rotation)--> FO , l'ancien FO devient FO-1.

ISOS - Script d'exécution

```

| -----
|          VISUALAGE PACBASE
| -----
| - IMPACT ANALYSIS: SELECTION OF STRINGS AND OPERATORS
| -----
|
| ISOS IS A COMPLEMENT TO THE ISEP PROCEDURE.
| FOR BETTER PERFORMANCE, IT IS ADVISABLE TO NARROW THE
| SCOPE OF THE SELECTION. THIS CAN BE DONE
| AT TWO DIFFERENT LEVELS, AND SHOULD ALWAYS
| BE DONE BEFORE RUNNING THE PROCEDURE.
| -----
|
|<job id=ISOS>
|
|<script language="VBScript">
|Dim MyProc
|MyProc = "ISOS"
|</script>
|
|<script language="VBScript" src="INIT.vbs"/>
|
|<script language="VBScript">

```

```

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PAN212"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7FP") = Rep_BASE & "\FP"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PAN212","PAC7FH",Rep_TMP & "\WFH.tmp")
Call BvpEnv("PAN212","PAC7MF",Rep_TMP & "\WFO.tmp")
Call BvpEnv("PAN212","PAC7IE",Rep_USR & "\ISOSIE212.txt")
Call RunCmdLog ("BVPAN212")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN212"))
End If
Call Err_Cod(Return , 0 , "PAN212")

Call Msg_Log (Array("1022" , "PAN215"))
'-----
Call BvpEnv("PAN215","PAC7FH",Rep_TMP & "\WFH.tmp")
WshEnv("PAC7HF") = Rep_BASE & "\FH-new"
WshEnv("PAC7FR") = Rep_BASE & "\FR-new"
Call RunCmdLog ("BVPAN215")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN215"))
End If
Call Err_Cod(Return , 0 , "PAN215")

Call Msg_Log (Array("1022" , "PAN260"))
'-----
Call BvpEnv("PAN260","PAC7MF",Rep_TMP & "\WFO.tmp")
WshEnv("PAC70F") = Rep_BASE & "\FO"
WshEnv("PAC7FO") = Rep_BASE & "\FO-new"
Call RunCmdLog ("BVPAN260")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN260"))
End If
Call Err_Cod(Return , 0 , "PAN260")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_BASE & "\FH")
Call Turnover(Rep_BASE & "\FO")
Call Turnover(Rep_BASE & "\FR")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

```

```

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

IMFH - Fusion de fichiers FH / Création FH et FR

IMFH - Présentation générale

Cette procédure permet de fusionner deux, ou plusieurs fichiers des points d'entrée sélectionnés (FH), pour :

- n'avoir qu'un seul fichier FH après élimination des doubles éventuels,
- obtenir le fichier des points d'entrée à épurer (FR) en phase avec le fichier FH créé.

Cette procédure est à utiliser pour fusionner le fichier FH issu de la procédure ISEP et celui issu de la procédure ISOS.

L'utilisation annexe de cette procédure est la recréation du fichier FR à partir d'un fichier FH.

Résultat obtenu

On obtient un fichier des points d'entrée sélectionnés (FH) et un fichier des points d'entrée à épurer (FR) destinés à la procédure IANA.

IMFH - Description des étapes

Elimination des points d'entrée en double : PAN215

Code	Nom physique	Type	Libellé
PAC7FH	Rép. base : FH	Entrée	Points d'entrée sélectionnés (fichiers à fusionner)
PAC7HF	Rép. base : FH-new	Sortie	Points d'entrée sélectionnés triés
PAC7FR	Rép. base : FR-new	Sortie	Points d'entrée réduits à épurer

Codes retour :

- 0 : OK.
- 12 : Erreur système

IMFH - Script d'exécution

```
' -----
'          VISUALAGE PACBASE
' -----
' - IMPACT ANALYSIS: MERGE FH FILES AND CREATION FR FILE
' -----
' THIS PROCEDURE SHOULD BE USED WHEN YOU WANT TO MERGE
' THE FH FILE PRODUCED BY THE ISEP PROCEDURE WITH THAT
' ISSUED BY THE ISOS PROCEDURE.
' -----
' -----
<job id=IMFH>

<script language="VBScript">
Dim MyProc
MyProc = "IMFH"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PAN215"))
'-----

If FSO.FileExists(Rep_BASE & "\\FH-1") Then
  Set MyFile = fso.GetFile(Rep_BASE & "\\FH-1")
  MyFile.Copy (Rep_TMP & "\\WFH.tmp")
else
  Set MyFile = FSO.CreateTextFile(Rep_TMP & "\\WFH.tmp")
  MyFile.Close
end if
If FSO.FileExists(Rep_BASE & "\\FH") Then
  Call CopMFiL (Rep_TMP & "\\WFH.tmp" ,Rep_BASE & "\\FH" , Rep_TMP & "\\WFH.tmp")
end if

Call BvpEnv("PAN215","PAC7FH",Rep_TMP & "\\WFH.tmp")

WshEnv("PAC7HF") = Rep_BASE & "\\FH-new"
WshEnv("PAC7FR") = Rep_BASE & "\\FR-new"
Call RunCmdLog ("BVPAN215")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN215"))
End If
Call Err_Cod(Return , 0 , "PAN215")

Call Msg_Log (Array("1022" , "BACKUP"))
```

```

'-----
Call Turnover(Rep_BASE & "\FH")
Call Turnover(Rep_BASE & "\FR")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

INFQ - Réinitialisation du fichier FQ

INFQ - Présentation générale

Cette procédure permet la réinitialisation du fichier FQ, contenant les critères de recherche déjà impactés par l'analyse et que l'on souhaite abandonner.

Cette opération doit être effectuée pour toute nouvelle analyse d'impact, soit parce que les points d'entrée changent, soit parce que le contexte d'analyse change.

En revanche, elle ne doit pas l'être entre deux itérations d'une même analyse d'impact.

Entrées utilisateur

Une ligne '*' avec Code utilisateur et Mot de passe.

Résultat obtenu

On obtient un fichier des critères de recherche (FQ) réinitialisé.

INFQ - Description des étapes

Réinitialisation du fichier FQ : PAN200

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7FQ	Rép. base : FQ-new	Sortie	Fichier séquentiel des critères impactés réinitialisé
PAC7DD	Rép. user : INFQDD200	Etat	Compte rendu d'anomalies

INFQ - Script d'exécution

```

' -----
'          VISUALAGE PACBASE
' -----
'          - IMPACT ANALYSIS:  INITIALIZATION OF "FQ" FILE -
' -----
'
' THIS ACTION SHOULD BE PERFORMED BEFORE A NEW IMPACT
' ANALYSIS EITHER BECAUSE THE ENTRY POINTS HAVE CHANGED
' OR BECAUSE THE ANALYSIS CONTEXT HAS CHANGED.
' -----
'
<job id=INFQ>

<script language="VBScript">
Dim MyProc
MyProc = "INFQ"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PAN200"))
' -----
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PAN200","PAC7DD",Rep_USR & "\INFQDD200.txt")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7FQ") = Rep_BASE & "\FQ-new"
Call RunCmdLog ("BVPAN200")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN200"))
End If
Call Err_Cod(Return , 0 , "PAN200")

```

```

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_BASE & "\FQ")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

IGRA - Décomposition zone groupe

IGRA - Présentation générale

Cette procédure décompose en zones élémentaires :

- des points d'entrée, détectés par la procédure ISEP, s'ils sont de type groupe,
- des critères de recherche d'impact, obtenus lors de l'exécution de la procédure IANA, s'ils sont de type groupe.

C'est une procédure optionnelle qui ne génère aucun critère de recherche d'impact.

Avant son exécution, vous avez la possibilité d'épurer :

- des points d'entrée, après l'exécution de la procédure ISEP,
- des critères de recherche d'impact, après l'exécution de la procédure IANA qui précède.

Dans les deux cas, les éliminations sont effectuées dans le fichier FR, (sous éditeur) par inhibition (valeur "E" dans le code action des lignes correspondantes) pour les conserver toutes pour un futur IANA.

Il n'est pas nécessaire d'éliminer les zones non groupe (l'analyseur les ignorera de toute façon).

La notion de niveau ou d'itération n'a pas de sens pour cette procédure.

Les points d'entrée (première itération) ou les critères de recherche d'impact (autres itérations) sont édités à l'issue de la prise en compte des critères épurés. Cette édition classe les critères en critères acceptés et critères éliminés.

Le fichier des résultats d'impact peut être vide ou contenir les résultats d'autres IANA, ISOS voire IGRA, dans des contextes d'exécution identiques

ou différents. Ceci permet de cumuler les résultats de toutes les itérations de l'analyse d'impact pour un ou plusieurs contextes.

La restitution de toutes les informations pour un contexte donné est paramétrable lors de l'édition par la procédure IPIA.

Le fichier FP figurant en entrée de cette procédure contient la liste des entités ou types d'entités à analyser. Si aucune entité ou type d'entité n'est inséré dans ce fichier avant son initialisation par la procédure INFP, toutes les entités analysables seront analysées.

Les entités à analyser sont définies de la façon suivante : type sur 3 caractères et code sur 6 caractères, la codification générique ***** étant possible.

Condition d'exécution

Aucune, si ce n'est que le fichier FH (points d'entrée ou critères de recherche d'impact) doit exister et ne doit pas être vide.

Anomalies d'exécution

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème. Vérifiez toutefois l'état des fichiers à génération (FH, FR et FO).

Entrées utilisateur

Une ligne '*' avec Code utilisateur et Mot de passe.

Résultat obtenu

On obtient un fichier des résultats d'analyse (FO) à destination de la procédure IPIA.

IGRA - Description des étapes

Prise en compte des critères épurés : PAN230

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7FH	Rép. base : FH	Entrée	Fichier des critères de recherche
PAC7FR	Rép. base : FR	Entrée	Fichier réduit des critères épurés
PAC7HF	Rép. tmp : WHF	Sortie	Fichier des critères de recherche (longueur=160)
PAC7DD	Rép. user : IGRADD230	Etat	Compte rendu d'anomalies

Codes retour :

- 0 : OK
- 12 : Erreur système

Edition des points d'entrée : PAN220

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7HF	Rép. tmp : WHF	Entrée	Fichier des critères triés
PAC7IL	Rép. user : IGRAIL220	Etat	Liste critères acceptés / éliminés

Codes retour :

- 0 : OK
- 12 : Erreur système

Décomposition des zones groupe : PAN255

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7FP	Rép. base : FP	Entrée	Entités à analyser
PAC7FH	Rép. tmp : WHF	Entrée	Critères impactés
PAC7MF	Rép. tmp : WFO	Sortie	Résultats d'analyse d'impact (longueur=266)

Codes retour :

- 0 : OK.
- 12 : Erreur système

Mise à jour des résultats d'analyse d'impact : PAN260

Code	Nom physique	Type	Libellé
PAC7MF	Rép. tmp : WFO	Entrée	Résultat analyse d'impact (niveau)
PAC7OF	Rép. base : FO	Entrée	Résultats de l'analyse précédente
PAC7FO	Rép. base : FO-new	Sortie	Résultats d'analyse d'impact triés

Codes retour :

- 0 : OK.
- 12 : Erreur système

IGRA - Script d'exécution

```

| -----
|          VISUALAGE PACBASE
|
| -----
|          - IMPACT ANALYSIS: PRINT OF "FQ" FILE -
|
| -----
|
| THE IGRA PROCEDURE BREAKS DOWN GROUP FIELDS
| INTO ELEMENTARY FIELDS:
| 1. ENTRY POINTS DETECTED BY THE ISEP PROCEDURE, IF
| THEY ARE OF THE GROUP TYPE.
| 2. IMPACT SEARCH CRITERIA OBTAINED BY RUNNING THE IANA
| PROCEDURE, IF THEY ARE OF THE GROUP TYPE.
|

```

```

' -----
'
<job id=IGRA>

<script language="VBScript">
Dim MyProc
MyProc = "IGRA"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PAN230"))
'-----
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PAN230","PAC7DD",Rep_USR & "\IGRADD230.txt")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7FH") = Rep_BASE & "\FH"
WshEnv("PAC7FR") = Rep_BASE & "\FR"
Call BvpEnv("PAN230","PAC7HF",Rep_TMP & "\WHF.tmp")
Call RunCmdLog ("BVPAN230")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN230"))
End If
Call Err_Cod(Return , 0 , "PAN230")

Call Msg_Log (Array("1022" , "PAN220"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PAN220","PAC7HF",Rep_TMP & "\WHF.tmp")
Call BvpEnv("PAN220","PAC7IL",Rep_USR & "\IGRAIL220.txt")
Call RunCmdLog ("BVPAN220")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN220"))
End If
Call Err_Cod(Return , 0 , "PAN220")

Call Msg_Log (Array("1022" , "PAN255"))
'-----
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7FP") = Rep_BASE & "\FP"
Call BvpEnv("PAN255","PAC7MF",Rep_TMP & "\WFO.tmp")
Call BvpEnv("PAN255","PAC7FH",Rep_TMP & "\WHF.tmp")
Call RunCmdLog ("BVPAN255")

```



```

If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN255"))
End If
Call Err_Cod(Return , 0 , "PAN255")

Call Msg_Log (Array("1022" , "PAN260"))
'-----
Call BvpEnv("PAN260","PAC7MF",Rep_TMP & "\WF0.tmp")
WshEnv("PAC70F") = Rep_BASE & "\F0"
WshEnv("PAC7F0") = Rep_BASE & "\F0-new"
Call RunCmdLog ("BVPAN260")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN260"))
End If
Call Err_Cod(Return , 0 , "PAN260")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_BASE & "\F0")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

IANA - Analyse d'impact

IANA - Présentation générale

Cette procédure recherche les rubriques et les chaînes de caractères impactées à partir :

- des points d'entrée, détectés par la procédure ISEP lorsque IANA est lancée pour la première fois,
- des critères de recherche d'impact, obtenus lors de l'exécution précédente de la procédure IANA.

La procédure IANA est donc itérative. Les itérations se succèdent jusqu'à ce que la procédure ne trouve plus aucun critère de recherche d'impact.

Avant l'exécution de la procédure IANA, vous avez la possibilité d'épurer :

- des points d'entrée, après l'exécution de la procédure ISEP,
- des critères de recherche d'impact, après l'exécution de la procédure IANA qui précède.

Dans les deux cas, les éliminations sont effectuées dans le fichier FR, (sous éditeur), soit par suppression physique soit par inhibition (valeur "E" dans le code action des lignes correspondantes).

Les points d'entrée (première itération) ou les critères de recherche d'impact (autres itérations) sont édités à l'issue de la prise en compte des critères épurés. Cette édition classe les critères en critères acceptés et critères éliminés.

Le fichier des critères déjà impactés (FQ) peut être réinitialisé si on ne veut pas garder trace des critères de recherche déjà traités.

Toutefois, il est recommandé de réinitialiser ce fichier avant la première exécution de IANA qui suit une nouvelle exécution de ISEP. Pour ce faire, utiliser la procédure INFQ.

Le fichier résultats d'impact peut être vide, ou contenir l'analyse d'impact d'autres contextes d'exécution. Il permet de cumuler les résultats de toutes les itérations de l'analyse d'impact pour un contexte donné.

Le fichier des entités à analyser (FP) est un fichier utilisé en entrée par les procédures d'analyse. Il contient la liste des entités ou types d'entités qui devront être analysés. Si aucune entrée n'est insérée dans ce fichier avant son initialisation par la procédure INFP, toutes les entités analysables seront analysées.

Les entités à analyser sont spécifiées de la façon suivante : type sur 3 caractères et code sur 6 caractères, la codification générique ***** étant possible.

Condition d'exécution

Aucune, si ce n'est que le fichier FH (points d'entrée ou critères de recherche d'impact) doit exister et ne doit pas être vide.

Anomalies d'exécution

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème. Vérifiez toutefois l'état des fichiers à génération (FH, FR et FO).

Entrées utilisateur

Une ligne '*' avec Code utilisateur et Mot de passe.

Cette procédure est itérative tant que le fichier FH (critères de recherche d'impact) n'est pas vide (code retour fixé à la valeur 4 s'il est vide, et à 0 dans le cas contraire).

Résultat obtenu

On obtient un fichier des résultats d'analyse (FO) à destination de la procédure IPIA.

IANA - Description des étapes

Prise en compte des critères épurés : PAN230

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7FH	Rép. base : FH	Entrée	Fichier des critères de recherche
PAC7FR	Rép. base : FR	Entrée	Fichier réduit des critères épurés
PAC7HF	Rép. tmp : WHF	Sortie	Fichier des critères de recherche (longueur=160)
PAC7DD	Rép. user : IANADD230	Etat	Compte rendu d'anomalies

Codes retour :

- 0 : OK
- 12 : Erreur système

Edition des points d'entrée : PAN220

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7HF	Rép. tmp : WHF	Entrée	Fichier des critères triés

Code	Nom physique	Type	Libellé
PAC7IL	Rép. user : IANAIL220	Etat	Liste critères acceptés / éliminés

Codes retour :

- 0 : OK
- 12 : Erreur système

Analyse d'impact : PAN250

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7AY	Rép. base : AY	Entrée	Données extension de la Base de Développement
PAC7FP	Rép. base : FP	Entrée	Entités à analyser
PAC7FH	Rép. tmp : WHF	Entrée	Critères impactés
PAC7FQ	Rép. tmp : WFQ	Entrée / Sortie	Critères impactés déjà traités
PAC7HF	Rép. tmp : WFH	Sortie	Nouveaux critères impactés (longueur=160)
PAC7MF	Rép. tmp : WFO	Sortie	Résultats d'analyse d'impact (longueur=266)

Codes retour :

- 0 : OK
- 4 : OK. Itération terminée.
- 12 : Erreur système

Mise à jour des résultats d'analyse d'impact : PAN260

Code	Nom physique	Type	Libellé
PAC7MF	Rép. tmp : WFO	Entrée	Résultat analyse d'impact (niveau)
PAC7OF	Rép. base : FO	Entrée	Résultats de l'analyse précédente
PAC7FO	Rép. base : FO-new	Sortie	Résultats d'analyse d'impact triés

Codes retour :

- 0 : OK
- 12 : Erreur système

Elimination des points d'entrée en double : PAN215

Code	Nom physique	Type	Libellé
PAC7FH	Rép. tmp : WFH	Entrée	Points d'entrée sélectionnés
PAC7HF	Rép. base : FH-new	Sortie	Points d'entrée sélectionnés triés
PAC7FR	Rép. base : FR-new	Sortie	Points d'entrée réduits à épurer

Codes retour :

- 0 : OK
- 12 : Erreur système

IANA - Script d'exécution

```

|-----|
|          VISUALAGE PACBASE          |
|-----|
|          - IMPACT ANALYSIS -        |
|-----|
|
| THE IANA PROCEDURE IS USED TO SEARCH DATA ELEMENTS AND
| CHARACTER-STRINGS ACCORDING TO:
| 1.THE ENTRY POINTS PROVIDED BY THE ISEP PROCEDURE WHEN
|    IANA IS RUN FOR THE FIRST TIME,
| 2.THE IMPACT SEARCH CRITERIA PRODUCED
|    BY A PRECEDING EXECUTION OF IANA.
| IANA IS THEREFORE AN ITERATIVE PROCESS, WHICH RUNS
| UNTIL NO MORE IMPACT SEARCH CRITERIA ARE FOUND.
|-----|
|

```

```

<job id=IANA>

<script language="VBScript">
Dim MyProc
MyProc = "IANA"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

If Not FSO.FileExists(Rep_BASE & "\F0") Then
    Call Msg_Log (Array("1054"))
    Call Msg_Log (Array("1023"))
    Wscript.Quit (0)
End if

If Not FSO.FileExists(Rep_BASE & "\FQ") Then
    Call Msg_Log (Array("1053" , "FQ"))
    Call DisplayInfo (Msg)
    Wscript.Quit (0)
End if

Call Msg_Log (Array("1022" , "COPY"))
'-----
Set MyFile = fso.GetFile(Rep_BASE & "\FQ")
MyFile.Copy (Rep_TMP & "\WFQ")
If Cobol = "Microfocus" then
    Set MyFile = fso.GetFile(Rep_BASE & "\FQ.idx")
    MyFile.Copy (Rep_TMP & "\WFQ.idx")
else 'acu
    Set MyFile = fso.GetFile(Rep_BASE & "\FQ.vix")
    MyFile.Copy (Rep_TMP & "\WFQ.vix")
end if

Call Msg_Log (Array("1022" , "PAN230"))
'-----
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PAN230","PAC7DD",Rep_USR & "\IANADD230.txt")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7FH") = Rep_BASE & "\FH"
WshEnv("PAC7FR") = Rep_BASE & "\FR"
Call BvpEnv("PAN230","PAC7HF",Rep_TMP & "\WHF.tmp")
Call RunCmdLog ("BVPAN230")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN230"))
End If
Call Err_Cod(Return , 0 , "PAN230")

```

```

Call Msg_Log (Array("1022" , "PAN220"))
'-----
Call BvpEnv("PAN220","PAC7HF",Rep_TMP & "\WHF.tmp")
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PAN220","PAC7IL",Rep_USR & "\IANAIL220.txt")
Call RunCmdLog ("BVPAN220")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN220"))
End If
Call Err_Cod(Return , 0 , "PAN220")

```

```

Call Msg_Log (Array("1022" , "PAN250"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7FP") = Rep_BASE & "\FP"
WshEnv("PAC7FQ") = Rep_TMP & "\WFQ"
Call BvpEnv("PAN250","PAC7HF",Rep_TMP & "\WFH.tmp")
Call BvpEnv("PAN250","PAC7MF",Rep_TMP & "\WFO.tmp")
Call BvpEnv("PAN250","PAC7FH",Rep_TMP & "\WHF.tmp")
Call RunCmdLog ("BVPAN250")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN250"))
End If
If Return = 4 Then
Call Msg_Log (Array("1055" , "PAN250"))
End If
Call Err_Cod(Return , 4 , "PAN250")

```

```

Call Msg_Log (Array("1022" , "PAN260"))
'-----
Call BvpEnv("PAN260","PAC7MF",Rep_TMP & "\WFO.tmp")
WshEnv("PAC70F") = Rep_BASE & "\FO"
WshEnv("PAC7F0") = Rep_BASE & "\FO-new"
Call RunCmdLog ("BVPAN260")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN260"))
End If
Call Err_Cod(Return , 0 , "PAN260")

```

```

Call Msg_Log (Array("1022" , "PAN215"))
'-----
Call BvpEnv("PAN215","PAC7FH",Rep_TMP & "\WFH.tmp")
WshEnv("PAC7HF") = Rep_BASE & "\FH-new"
WshEnv("PAC7FR") = Rep_BASE & "\FR-new"
Call BvpEnv("PAN215","PAC7FH",Rep_TMP & "\WFH.tmp")
Call RunCmdLog ("BVPAN215")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN215"))

```

```

End If
Call Err_Cod(Return , 0 , "PAN215")

Call Msg_Log (Array("1022" , "COPY"))
'-----
Set MyFile = fso.GetFile(Rep_TMP & "\WFQ")
MyFile.Copy (Rep_BASE & "\FQ-new")
If Cobol = "Microfocus" then
    Set MyFile = fso.GetFile(Rep_TMP & "\WFQ.idx")
    MyFile.Copy (Rep_BASE & "\FQ-new.idx")
else 'acu
    Set MyFile = fso.GetFile(Rep_TMP & "\WFQ.vix")
    MyFile.Copy (Rep_BASE & "\FQ-new.vix")
end if
Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_BASE & "\FH")
Call Turnover(Rep_BASE & "\FR")
Call Turnover(Rep_BASE & "\FO")
Call Turnover(Rep_BASE & "\FQ")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr(Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

IPFQ - Edition du fichier FQ

IPFQ - Présentation générale

Cette procédure permet l'édition de tous les points d'entrée et de tous les critères de recherche d'impact utilisés ou non lors d'une analyse d'impact complète.

Tous ces critères et points d'entrée sont mémorisés sur le fichier FQ.

La procédure IPFQ propose quatre types d'édition :

- liste des points d'entrée acceptés,
- liste des points d'entrée éliminés,
- liste des critères de recherche d'impact acceptés,
- liste des critères de recherche d'impact éliminés.

L'édition présente les critères ou les points d'entrée classés par ordre alphabétique à l'intérieur de chaque catégorie, et ce par bibliothèque de définition de ces critères.

Dans l'ordre d'édition, les catégories sont :

- chaîne de caractères,
- rubrique définie au dictionnaire,
- rubrique définie dans la description du segment,
- rubrique définie dans la structure de l'état,
- rubrique définie en working de l'écran ou du programme.

La procédure IPFQ permet également une sélection de certains points d'entrée ou critères de recherche d'impact.

Dans le cas d'une sélection, seuls les critères sélectionnés sont édités.

Condition d'exécution

Aucune, si ce n'est que le fichier FQ doit exister.

Anomalies d'exécution

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

Résultat obtenu

On obtient l'édition des points d'entrée et des critères de recherche.

IPFQ - Entrées Utilisateur

Une ligne '*' avec Code utilisateur et Mot de passe.

Une ligne 'S' par sélection de critères (facultative) :

Pos.	Lon.	Valeur	Signification
2	1	'S'	Code ligne
3	1		Type de critère
		'E'	Rubrique définie au dictionnaire
		'C'	Chaîne de caractères
		'X'	Rubrique groupe ou non définie
		'*'	Tous les types de critères
4	1		Code de la provenance

Pos.	Lon.	Valeur	Signification
		'3'	Ligne -CE de segment
		'6'	Ligne -CE d'état
		'7'	Ligne -W d'écran ou de programme
		'*'	Toutes provenances
6	1		Pour le type de zone
		'G'	Pour zone groupe
		' '	Pour zone élémentaire
		'*'	Pour tout type de zone

IPFQ - Description des étapes

Extraction des critères : PAN240

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7FQ	Rép. base : FQ	Entrée	Critères impactés lors de l'analyse
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7FH	Rép. tmp : WFH	Sortie	Fichier des critères de recherche (longueur=160)
PAC7IX	Rép. user : IPFQIX240	Etat	Compte-rendu de contrôle

Edition des critères impactés : PAN220

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7HF	Rép. tmp : WFH	Entrée	Points d'entrée ou critères triés

Code	Nom physique	Type	Libellé
PAC7IL	Rép. user : IPFQIL220	Etat	Liste des points d'entrée et des critères

Codes retour

- 0 : OK.
- 12 : Erreur système

IPFQ - Script d'exécution

```

'-----
'          VISUALAGE PACBASE
'-----
'          - IMPACT ANALYSIS:  GROUP FIELDS ANALYSIS -
'-----
'
' THE IPFQ PROCEDURE PRINTS ALL THE ENTRY POINTS AND
' IMPACT SEARCH CRITERIA USED (ACCEPTED OR REJECTED)
' DURING A THOROUGH IMPACT ANALYSIS.
' ALL THE CRITERIA AND ENTRY POINTS ARE STORED IN THE FQ
' FILE.
' PROCEDURE, IF THEY ARE OF THE GROUP TYPE.
'-----
'
<job id=IPFQ>

<script language="VBScript">
Dim MyProc
MyProc = "IPFQ"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

If Not FS0.FileExists( Rep_BASE & "\FQ") Then
    Call Msg_Log (Array("1053", "FQ"))
    Call Msg_Log (Array("1023"))
    Wscript.Quit (0)
End if

Call Msg_Log (Array("1022" , "PAN240"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"

```

```

WshEnv("PACGGR") = Rep_ABASE & "\\AR"
WshEnv("PACGGU") = Rep_ABASE & "\\GU"
Call BvpEnv("PAN240","PAC7FH",Rep_TMP & "\\WFH.tmp")
WshEnv("PAC7FQ") = Rep_BASE & "\\FQ"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PAN240","PAC7IX",Rep_USR & "\\IPFQIX240.txt")
Call RunCmdLog ("BVPAN240")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN240"))
End If
Call Err_Cod(Return , 0 , "PAN240")

Call Msg_Log (Array("1022" , "PAN220"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
Call BvpEnv("PAN220","PAC7HF",Rep_TMP & "\\WFH.tmp")
Call BvpEnv("PAN220","PAC7IL",Rep_USR & "\\IPFQIL220.txt")
Call RunCmdLog ("BVPAN220")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN220"))
End If
Call Err_Cod(Return , 0 , "PAN220")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_BASE & "\\FQ")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

IPEP - Edition des points d'entrée

IPEP - Présentation générale

Cette procédure propose deux types d'édition :

- La liste des points d'entrée, produite après la procédure ISEP qui génère les points d'entrée.
- La liste des critères de recherche d'impact, produite après la procédure IANA qui produit les critères de recherche d'impact.

L'édition présente les critères ou les points d'entrée classés par ordre alphabétique à l'intérieur de chaque catégorie, et ce par bibliothèque de définition de ces critères.

Dans l'ordre d'édition, les catégories sont :

- chaîne de caractères,

- rubrique définie au dictionnaire,
- rubrique définie dans la description du segment,
- rubrique définie dans la structure de l'état,
- rubrique définie en Working de l'écran ou du programme.

Condition d'exécution

Aucune, si ce n'est que le fichier FH doit exister.

Anomalies d'exécution

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

Entrées utilisateur

Aucune entrée n'est nécessaire à l'exécution de la procédure IPEP.

Résultat obtenu

On obtient une édition des points d'entrée.

IPEP - Description des étapes

Edition des points d'entrée : PAN220

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7HF	Rép. base : FH	Entrée	Fichier des points d'entrée triés
PAC7IL	Rép. user : IPEPIL220	Etat	Liste des points d'entrée

Codes retour :

- 0 : OK.
- 12 : Erreur système

IPEP - Script d'exécution

```

| -----
|          VISUALAGE PACBASE
| -----
|          - IMPACT ANALYSIS:  PRINTING OF ENTRY POINTS -
| -----
|
| THE IPEP PROCEDURE PRODUCES TWO TYPES OF PRINTOUTS.

```

```

' 1. LIST OF ENTRY POINTS:
' THIS LIST IS OBTAINED AFTER THE ISEP PROCEDURE, SINCE
' THIS PROCEDURE SELECTS THE ENTRY POINTS.
' 2. LIST OF IMPACT SEARCH CRITERIA:
' THIS LIST IS OBTAINED AFTER THE IANA PROCEDURE, SINCE
' THIS PROCEDURE SELECTS THE IMPACT SEARCH CRITERIA.
' -----
'
<job id=IPEP>

<script language="VBScript">
Dim MyProc
MyProc = "IPEP"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PAN220"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7HF") = Rep_BASE & "\FH"
Call BvpEnv("PAN220","PAC7IL",Rep_USR & "\IPEPIL220.txt")
Call RunCmdLog ("BVPAN220")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN220"))
End If
Call Err_Cod(Return , 0 , "PAN220")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)
Wscript.Quit (Return)

</script>
</job>

```

IPIA - Edition résultats de l'analyse d'impact

IPIA - Présentation générale

Cette procédure permet d'imprimer plusieurs états et de formater les résultats de l'analyse en mouvements de mise à jour batch.

Les éditions produites par cette procédure sont :

- Résultats d'analyse par point d'entrée (analyse après chaque itération) : cet état est demandé par la valeur '1' en position 7 de la ligne 'P' des entrées utilisateur.

- Liste des critères de recherche d'impact par point d'entrée (exploitable à la fin des itérations) : cet état est demandé par la valeur '1' en position 8 de la ligne 'P' des entrées utilisateur.
- Résultat d'analyse d'impact par bibliothèque (les résultats sont formatés en mouvements de mise à jour batch sous forme d'un état ou d'un fichier) : l'état est demandé par la valeur '1' en position 9 de la ligne 'P' des entrées utilisateur (option supplémentaire sauts de page et sauts de ligne demandée par la valeur '2' en position 9), le fichier est demandé par la valeur '1' en position 12 de la ligne 'P' des entrées utilisateur.
- Résumé des occurrences impactées (liste de toutes les occurrences impactées avec le nombre de lignes impactées, pour chaque type de ligne, non classées par points d'entrée) : cet état est demandé par la valeur '1' en position 10 de la ligne 'P' des entrées utilisateur.
- Liste des points d'entrée par critère de recherche impactée pour chaque zone impactée (liste des points d'entrée et des critères de recherche d'impact à l'origine de l'impact pour chaque itération) : cet état est demandé par la valeur '1' en position 14 de la ligne 'P' des entrées utilisateur.
- Etat récapitulatif (nombre de lignes impactées par type d'entité et par type de ligne, toutes lignes confondues) : cet état est demandé par la valeur '1' en position 11 de la ligne 'P' des entrées utilisateur.
- Analyse des chaînes de caractères (liste des utilisations de chacune des chaînes de caractères demandées par la procédure ISOS) : cet état est demandé par la valeur '1' en position 19 de la ligne 'P' des entrées utilisateur.
- Analyse des opérateurs (liste des utilisations de chacun des opérateurs demandés par la procédure ISOS) : cet état est demandé par la valeur '1' en position 20 de la ligne 'P' des entrées utilisateur.
- Liste des entités impactées par point d'entrée (liste des entités impactées par les points d'entrée de type rubrique, tous critères de recherche confondus) : cet état est demandé par la valeur '1' en position 21 de la ligne 'P' des entrées utilisateur.
- Nombre de lignes modifiées ventilées par description pour chaque entité : cet état très synthétique permet d'affiner les estimations par types de lignes cumulés par bibliothèque et il est demandé par la valeur '1' en position 22 de la ligne 'P' des entrées utilisateur.
- Analyse des constantes (liste des utilisations de chacune des constantes demandées par la procédure ISOS) : cet état est demandé par la valeur '1' en position 23 de la ligne 'P' des entrées utilisateur.

Condition d'exécution

Aucune, si ce n'est que le fichier FO doit exister et ne doit pas être vide.

Anomalies d'exécution

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

Résultat obtenu

On obtient une édition des résultats d'analyse et une édition de la liste des mouvements triés par bibliothèque.

IPIA - Entrées Utilisateur

Une ligne d'identification du contexte (ligne *) est obligatoire. Elle doit être en tête de flot.

Si une bibliothèque inférieure a été précisée pour ISEP sur la ligne d'identification, elle doit être rappelée ici.

La ligne * doit être suivie d'une seule ligne de type P, formatée comme suit :

Pos.	Lon.	Valeur	Signification
2	1	'P'	Code de la ligne
3	1		Ne rien renseigner sauf DOS/VSE
		'T'	Option par défaut tous matériels
		'N'	Si CURRENT-DATE = JJ/MM/AA
4	3	bbb	Code de la bibliothèque (cette sélection est uniquement disponible avec les requêtes entrées en positions 9 et 10)
7	1	' '	Pas de résultat d'analyse d'impact par point d'entrée
		'1'	Résultats de l'analyse d'impact par point d'entrée
8	1	' '	Pas de liste des critères impactés par point d'entrée
		'1'	Liste des critères impactés par point d'entrée
9	1	' '	Pas d'impression de résultat formaté
		'1'	Impression des résultats formatés en mouvements de mise à jour batch, classés par bibliothèque
		'2'	Même liste avec les sauts de page et de ligne
10	1	' '	Pas de liste d'occurrences impactées
		'1'	Liste des occurrences impactées
11	1	' '	Pas d'état récapitulatif par bibliothèque
		'1'	Etat récapitulatif par bibliothèque
12	1		Identique aux valeurs de la position 9, mais mise en forme d'un fichier et non d'une impression
13	1	' '	Pas d'inhibition des lignes indirectement impactées

Pos.	Lon.	Valeur	Signification
		'1'	Option générale : inhibe les lignes indirectement impactées (ex: -CD)
14	1	' '	Pas de liste des points d'entrée par critère de recherche d'impact
		'1'	Liste des points d'entrée par critère de recherche d'impact
15	2	nn	Numéro du niveau souhaité (itération de IANA)
17	2	pp	Nombre de lignes imprimées par page
19	1	' '	Pas de résultat d'analyse des chaînes de caractères
		'1'	Résultat d'analyse des chaînes de caractères
20	1	' '	Pas de résultat d'analyse des opérateurs
		'1'	Résultat d'analyse des opérateurs
21	1	' '	Pas d'entités impactées par point d'entrée
		'1'	Entités impactées par point d'entrée
22	1	' '	Pas de nombre de lignes par description
		'1'	Nombre de lignes par description
23	1	' '	Pas de résultat d'analyse des constantes
		'1'	Résultat d'analyse des constantes
24	1	' '	Pas de résultat d'analyse des zones groupe
		'1'	Résultat d'analyse des zones groupe
25	10		Sélection des mouvements générés
		blanc	Sélection de toutes les entités
		autres	Sélection demandée où les valeurs possibles et cumulables, sont :
		'B'	Blocs base de données
		'E'	Rubriques
		'F'	Méta-entités
		'O'	Ecrans, écrans C/S,...
		'P'	Programmes
		'R'	Etats
		'S'	Segments et structures de données
		'T'	Textes
		'V'	Documents
		'\$'	Entités utilisateur
35	1	' '	Pas de prise en compte des mouvements de ISOS

Pos.	Lon.	Valeur	Signification
		'1'	Prise en compte des mouvements de ISOS

IPIA - Description des étapes

Edition des résultats d'impact : PAN270

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7FO	Rép. base : FO	Entrée	Fichier des résultats d'impact
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7BM	Rép. tmp : WMB	Sortie	Entrées utilisateur converties
PAC7GY	Rép. user : IPIAGY	Sortie	Mouvements PAF pour UPDP (longueur=310)
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements batch pour édition (longueur=80)
PAC7IF	Rép. user : IPIAIF270	Etat	Résultats de l'analyse

Codes retour :

- 0 : OK
- 12 : Erreur système

Edition des mouvements générés : PAN280

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7BM	Rép. tmp : WMB	Entrée	Entrées utilisateur
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements batch générés
PAC7VM	Rép. user : MVIPIA	Sortie	Mouvements batch sélectionnés (longueur=80)
PAC7IT	Rép. user : IPIAIT280	Etat	Liste des mouvements par bibliothèque

Codes retour :

- 0 : OK
- 12 : Erreur système

IPIA - Script d'exécution

```
'-----  
'          VISUALAGE PACBASE  
'-----  
'          - IMPACT ANALYSIS : PRINTING OF RESULTS -  
'-----  
'  
' THE IPIA PROCEDURE IS USED TO PRINT  
' REPORTS ON THE ANALYSIS RESULTS  
' AND TO FORMAT THESE RESULTS IN  
' BATCH UPDATE TRANSACTIONS.  
'-----  
'  
<job id=IPIA>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "IPIA"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Call Msg_Log (Array("1022" , "PAN270"))  
'-----  
WshEnv("PAC7MB") = Fic_Input  
WshEnv("PAC7AE") = Rep_SKEL & "\AE"  
WshEnv("PAC7AR") = Rep_BASE & "\AR"  
WshEnv("PACGGN") = Rep_ABASE & "\AN"  
WshEnv("PACGGR") = Rep_ABASE & "\AR"  
WshEnv("PACGGU") = Rep_ABASE & "\GU"  
WshEnv("PAC7FO") = Rep_BASE & "\FO"  
Call BvpEnv("PAN270", "PAC7BM", Rep_TMP & "\WMB.tmp")  
Call BvpEnv("PAN270", "PAC7GY", Rep_USR & "\IPIAGY.txt")  
Call BvpEnv("PAN270", "PAC7MV", Rep_TMP & "\WMV.tmp")  
Call BvpEnv("PAN270", "PAC7IF", Rep_USR & "\IPIAIF270.txt")  
Call RunCmdLog ("BVPAN270")  
If Return = 12 Then  
Call Msg_Log (Array("1026" , "PAN270"))  
End If  
Call Err_Cod(Return , 0 , "PAN270")
```

```

Call Msg_Log (Array("1022" , "PAN280"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PAN280","PAC7BM",Rep_TMP & "\WMB.tmp")
Call BvpEnv("PAN280","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAN280","PAC7VM",Rep_USR & "\MVIPIA.txt")
Call BvpEnv("PAN280","PAC7IT",Rep_USR & "\IPIAIT280.txt")
Call RunCmdLog ("BVPAN280")
If Return = 12 Then
Call Msg_Log (Array("1026" , "PAN280"))
End If
Call Err_Cod(Return , 0 , "PAN280")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr(Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

Chapitre 8. Contrôle d'intégrité méthodologique

ADM - Méthode SSADM

SADM - Présentation générale

Cette procédure est fournie pour les utilisateurs ayant acquis le module Pacdesign Méthode SSADM.

Elle contrôle la validité et l'intégrité des occurrences remontées (par utilisateur) de la Station de Travail dans le Référentiel VisualAge Pacbase.

NOTE : La Méthode SSADM ainsi que les fonctionnalités de la procédure n'existent qu'en anglais. Pour plus d'informations, consultez l'aide en ligne.

Condition d'exécution

Aucune.

SADM - Entrées Utilisateur

Une ligne '*' pour l'accès à la bibliothèque :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	bbb	Code bibliothèque
22	4	nnnn	Numéro de session (blanc=courante)
26	1	'T'	Etat de la session si session Test
37	25	Réservé IMS : identifiant de la requête (cf. PAF batch IMS)

Les lignes de requête d'édition :

Pos.	Lon.	Valeur	Signification
2	1	'T'	Code ligne
3	1		Code état à éditer
		'V'	"Validation of SSADM entities"
		'1'	"Cross-boundaries dataflows within a DFD"

Pos.	Lon.	Valeur	Signification
		'2'	"Operational masters within a DSD"
		'3'	"All entities with their attributes"
4	6	eeeeee	Code entité (nécessaire pour '1' ou '2')

Edition obtenue

Cette procédure édite, en fonction des requêtes d'édition :

- Un compte-rendu "Validation of SSADM entities".
- "List of cross-boundaries dataflows within a DFD".
- "List of operational masters within a DSD".
- "List of all entities with their attributes".

SADM - Description des étapes

Contrôle d'intégrité des entités SSADM : PADM10

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
SYSPAF	Rép. tmp : SYSPAF	Entrée/Sortie	Fichier indexé standard PAF
PAC7EJ	Rép. user : SADMEJM10	Etat	Listes contrôle intégrité SSADM
PAC7ET	Rép. user : SADMETM10	Etat	Compte-rendu accès PAF
PAC7DD	Rép. user : SADMDDM10	Etat	Listes anomalies

SADM - Script d'exécution

```

|-----|
| VISUALAGE PACBASE |
|-----|

```

```

'          - PACDESIGN SSADM INTEGRITY CHECKING -
'
' -----
'
' THIS PROCEDURE IS SUPPLIED FOR USERS OF THE WORKSTATION
' AND THE SSADM PACDESIGN APPLICATION DESIGN METHODOLOGY.
'
' INPUT :
' - USER IDENTIFICATION LINE (REQUIRED)
'   COL 2 : "*"
'   COL 3 : USERIDXX
'   COL 11 : PASSWORD
'   COL 19 : (BBB) LIBRARY CODE
'   COL 22 : (4 N) SESSION NUMBER
'   COL 26 : (1 CAR.) SESSION VERSION
'   COL 37 (25 CAR.) RESERVED IMS
' - COMMAND LINE :
'   COL 2 : "T" LINE CODE
'   COL 3 : CODE FOR REPORT TO BE PRINTED
'           "V" : VALIDATION OF SSADM ENTITIES
'           "1" : CROSS-BOUNDARIES DATA FLOWS
'                 WITHIN A DFD
'           "2" : OPERATIONAL MASTERS WITHIN A DSD
'           "3" : ALL ENTITIES WITH THEIR ATTRIBUTES
'   COL 4 : (6 CAR.) ENTITY CODE
'                 (REQUIRED FOR "1" OR "2")
' -----
'
<job id=SADM>

<script language="VBScript">
Dim MyProc
MyProc = "SADM"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PADM10"))
' -----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PADM10","PAC7DD",Rep_USR & "\SADMDDM10.txt")
Call BvpEnv("PADM10","PAC7EJ",Rep_USR & "\SADMEJM10.txt")
Call BvpEnv("PADM10","PAC7ET",Rep_USR & "\SADMETM10.txt")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PADM10","SYSPAF",Rep_TMP & "\SYSPAF.tmp")

```

```

Call RunCmdLog ("BVPADM10")

Call Err_Cod(Return , 0 , "PADM10")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

YSM - Méthode YSM

YSMC - Présentation générale

Cette procédure est fournie pour les utilisateurs ayant acquis le module Pacdesign Méthode YSM.

Elle effectue les traitements suivants :

- Contrôle de la validité et de l'intégrité des entités remontées de la Station de travail dans la base de spécifications par l'utilisateur.
- Contrôle de la cohérence d'un Dataflow diagram par rapport à son diagramme parent.
- Edition des différentes listes hiérarchiques de certaines entités de la base de spécifications.

Note

La méthode YSM ainsi que les fonctionnalités de la procédure n'existent qu'en anglais. Pour plus d'informations, se reporter au manuel "Pacdesign".

Condition d'exécution

Aucune.

YSMC - Entrées Utilisateur

Ligne '*' pour l'accès à la bibliothèque (obligatoire) :

Pos.	Lon.	Valeur	Signification
2	1	'*	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe

Pos.	Lon.	Valeur	Signification
19	3	bbb	Code bibliothèque
22	4	nnnn	Numéro de session (blanc=courante)
26	1	'T'	Etat de la session si session Test
37	25	Réservé IMS : identifiant de la requête (cf. PAF batch IMS)

Ligne requête de validation des entités (facultative) :

Pos.	Lon.	Valeur	Signification
2	1	'T'	Code ligne
3	1		Code état à éditer
		'W'	"Validation of YSM entities"

Lignes requêtes de contrôle des entités PRC (facultatives) :

Pos.	Lon.	Valeur	Signification
2	1	'T'	Code ligne
3	1		Code état à éditer
		'Y'	"Inter process consistency checking"
4	6	eeeeee	Code entité (PRC)

Lignes requêtes d'édition (facultatives) :

Pos.	Lon.	Valeur	Signification
2	1	'T'	Code ligne
3	1		Code état à éditer
		'0'	"List of Relationships"
		'4'	"Process Decomposition list (CTX)"
		'5'	"Process Decomposition list (DFD)"
		'6'	"Datastore Decomposition list"
		'7'	"Eventflow Decomposition list"
		'8'	"Group Dataflow Decomposition list"
		'9'	"Multiple Dataflow Decomposition list"
4	6	eeeeee	Code entité (REL/CTX/PRC/DST/EFL/DFL)

Edition obtenue

Cette procédure édite :

- Un compte-rendu "Validation of YSM entities".
- Un compte-rendu "Inter process consistency checking".
- Les états :
 - "List of relationships".
 - "Process decomposition list (CTX)".
 - "Process decomposition list (DFD)".
 - "Datastore Decomposition list".
 - "Eventflow Decomposition list".
 - "Group Dataflow Decomposition list".
 - "Multiple Dataflow Decomposition list".

YSMC - Description des étapes

Validation des entités YSM : PYSMCC

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
SYSPAF	Rép. tmp : SYSPAF	Entrée/Sortie	Fichier indexé standard PAF
PAC7EJ	Rép. user : YSMCEJMCC	Etat	Listes contrôle intégrité
PAC7EI	Rép. user : YSMCEIMCC	Etat	Etat de validation
PAC7DD	Rép. user : YSMCDDMCC	Etat	Listes anomalies

Validation des entités : PYSMC3

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur

Code	Nom physique	Type	Libellé
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
SYSPAF	Rép. tmp : SYSPAF	Entrée/Sortie	Fichier indexé standard PAF
PAC7EJ	Rép. user : YSMCEJMC3	Etat	Listes contrôle intégrité

Contrôle des entités 2 : PYSMC2

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AR	Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
SYSPAF	Rép. tmp : SYSPAF	Entrée/Sortie	Fichier indexé standard PAF
PAC7EJ	Rép. user : YSMCEJMC2	Etat	Listes contrôle intégrité

YSMC - Script d'exécution

```

-----
| VISUALAGE PACBASE
|
| -----
| - PACDESIGN YSM INTEGRITY CHECKING -
|
| -----

```

```

'
' THIS PROCEDURE IS SUPPLIED FOR USERS OF THE WORKSTATION
' AND THE YSM PACDESIGN APPLICATION METHODOLOGY.
'
'
' IT CHECKS THE VALIDITY AND THE INTEGRITY OF THE
' ENTITIES UPLOADED FROM THE WORKSTATION TO THE HOST
' SPECIFICATIONS DICTIONARY BY THE USER.
' IT CHECKS THE CONSISTENCY BETWEEN A DATA FLOW
' DIAGRAM AND ITS PARENT DIAGRAM.(PRC)
'
' IT ESTABLISHES DIFFERENT HIERARCHICAL LISTS OF
' CERTAIN ENTITIES OF THE DATABASE.
' -----
'
<job id=YSMC>

<script language="VBScript">
Dim MyProc
MyProc = "YSMC"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PYSMCC"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PYSMCC","PAC7DD",Rep_USR & "\YSMCCDDMCC.txt")
Call BvpEnv("PYSMCC","PAC7EI",Rep_USR & "\YSMCEIMCC.txt")
Call BvpEnv("PYSMCC","PAC7EJ",Rep_USR & "\YSMCEJMCC.txt")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PYSMCC","SYSPAF",Rep_TMP & "\SYSPAF.tmp")
Call RunCmdLog ("BVPYSMCC")

Call Err_Cod(Return , 0 , "PYSMCC")

Call Msg_Log (Array("1022" , "PYSMC3"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PYSMC3","PAC7EJ",Rep_USR & "\YSMCEJMC3.txt")
WshEnv("PAC7MB") = Fic_Input

```

```

Call BvpEnv("PYSMC3","SYSPAF",Rep_TMP & "\SYSPAF.tmp")
Call RunCmdLog ("BVPYSMC3")
Call Err_Cod(Return , 0 , "PYSMC3")

Call Msg_Log (Array("1022" , "PYSMC2"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PYSMC2","PAC7EJ",Rep_USR & "\YMCEJMC2.txt")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PYSMC2","SYSPAF",Rep_TMP & "\SYSPAF.tmp")
Call RunCmdLog ("BVPYSMC2")
Call Err_Cod(Return , 0 , "PYSMC2")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```




Référence : DELNT003361F - 8794

Imprimé en France