

VisualAge Pacbase



Guide d'installation Serveur Windows & Composants Client

Version 3.5



VisualAge Pacbase



Guide d'installation Serveur Windows & Composants Client

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.

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.

Sixième édition (Décembre 2013)

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 1983,2013.

Table des matières

Notices	vii	Chapitre 4. Installation/Réinstallation des composants Client	23
Marques	ix	Informations préalables	23
Chapitre 1. Préambule	1	Répertoire racine	23
Objet du manuel	1	Lancement de l'installation	24
Architecture du produit	1	Principes et architectures de la communication	24
Contenu de la livraison	2	Administrator & Developer workbench	27
Bibliographie	3	Pour la fonction Publication : Open Jade et Tidy.	29
Chapitre 2. Prérequis	5	Outils eBusiness	30
Prérequis partie Serveur	5	Station de Travail VisualAge Pacbase.	32
Matériel et logiciel	5	Web Application Models (WAM)	37
Espace disque.	5	Middleware	37
Prérequis partie Client	6	Mise à jour des paramètres de la communication	40
Matériel.	6	Le fichier bases.ini	40
Espace disque.	6	Le fichier vaplocat.ini	42
Logiciel	6	Mise à jour, Ajout, Désinstallation de composants	46
Communications	7	Chapitre 5. Tests	49
Chapitre 3. Installation de la partie Serveur 9	9	Liste des utilitaires principaux	49
Installation	9	Tests de l'installation	49
Installation du Système	10	Tests de mise à jour TP/batch, Edition-Génération	50
Un élément du Système : La Procédure	10	Tests des procédures de la Base Administration	50
Définition.	10	Tests des procédures de la Base de Développement	50
Mode d'exécution	12	Tests des utilitaires d'extraction	51
Installation du Référentiel	14	Chapitre 6. Réinstallation Serveur	53
Base Administration	14	Réinstallation	53
Base de Développement	15	Désinstallation	53
Saisie de la clé d'accès	16	Suppression des données.	53
Sauvegarde de la Base Administration	16	Chapitre 7. Reprises et échanges entre bases 2.n et 3.n	55
Créer nouvelle Base	16	Reprises des bases des versions 2.0 et 2.5	55
Paramètres de Configuration	17	Préambule	55
Déréférencer une Base	17	Remarques	55
Liste des dates des composants	17	Opérations à effectuer.	56
INSL - Script d'exécution.	17	Reprise des paramètres utilisateur (PE25)	58
Connexion	19	PE25 - Présentation générale	58
Lancement et Arrêt d'un serveur	19		
Base sous contrôle DSMS	21		
Connexion d'un émulateur 3270	21		
Moniteur Gestion des Listeners	21		

PE25 - Entrées / Traitements / Résultats	59	MB25 - Description des étapes	106
PE25 - Description des étapes	60	MB25 - Script d'exécution	106
PE25 - Script d'exécution	62	Echange mouvements GY entre base 2.n et 3.n (GY25)	107
Reprise de la Base de Développement (PC25)	64	GY25 - Présentation générale	107
PC25 - Présentation générale	64	GY25 - Description des étapes	108
PC25 - Remarques sur les données reprises	64	GY25 - Script d'exécution	109
PC25 - Entrées / Traitements / Résultats	66	Echange mouvements MB entre base 3.n et 2.n (MB30)	110
PC25 - Description des étapes	66	MB30 - Présentation générale	110
PC25 - Script d'exécution	70	MB30 - Description des étapes	110
Reprise commandes d'édition-génération (PG20)	74	MB30 - Script d'exécution	111
PG20 - Présentation générale	74	Echange mouvements GY entre base 3.n et 2.n (GY30)	112
PG20 - Entrées / Traitements / Résultats	75	GY30 - Présentation générale	112
PG20 - Description des étapes	75	GY30 - Description des étapes	112
PG20 - Script d'exécution	79	GY30 - Script d'exécution	113
Reprise commandes d'édition-génération (PG25)	83	Procédures - Tableau récapitulatif des changements	114
PG25 - Présentation générale	83	Reprises des bases de la version 3.0	116
PG25 - Entrées / Traitements / Résultats	84	Opérations à effectuer	116
PG25 - Description des étapes	84	Chapitre 8. Utilitaires RPP 119	
PG25 - Script d'exécution	88	MIAM - Aide à la migration	119
Reprise mouvements PJ (PJ25)	91	MIAM - Présentation générale	119
PJ25 - Présentation générale	91	MIAM - Entrées Utilisateur	127
PJ25 - Description des étapes	92	MIAM - Description des étapes	129
PJ25 - Script d'exécution	93	MIAM - Script d'exécution	130
Reprise du fichier PP (PP25)	94	MIBA - Migration Batch	131
PP25 - Présentation générale	94	MIBA - Présentation générale	131
PP25 - Entrées / Traitements / Résultats	95	MIBA - Entrées Utilisateur	131
PP25 - Description des étapes	95	MIBA - Description des étapes	134
PP25 - Script d'exécution	97	MIBA - Script d'exécution	139
Reprise mots de passe (UTMP)	99	MIBR - Migration Batch contexte	142
UTMP - Présentation générale	99	MIBR - Présentation générale	142
UTMP - Description des étapes	100	MIBR - Entrées Utilisateur	143
UTMP - Script d'exécution	100	MIBR - Eclatement des bibliothèques	146
Reprise paramètres Pac/Transfer (UV25)	101	MIBR - Description des étapes	148
UV25 - Présentation générale	101	MIBR - Script d'exécution	152
UV25 - Entrées / Traitements / Résultats	101	MIBJ - Migration Journal	155
UV25 - Description des étapes	101	MIBJ - Présentation générale	155
UV25 - Script d'exécution	104	MIBJ - Entrées Utilisateur	156
Echange mouvements MB entre base 2.n et 3.n (MB25)	105	MIBJ - Description des étapes	158
MB25 - Présentation générale	105	MIBJ - Script d'exécution	163
		MIMA - Migration Macros	166
		MIMA - Présentation générale	166
		MIMA - Entrées Utilisateur	167
		MIMA - Description des étapes	168
		MIMA - Script d'exécution	170
		MITH - Migration Thesaurus	172

MITH - Présentation générale	172	Présentation générale	217
MITH - Entrées Utilisateur	172	Documentation en ligne.	217
MITH - Description des étapes	173	Squelettes de génération	217
MITH - Script d'exécution	175	Base Administration	218
MIUS - Migration Utilisateurs.	176	Fichiers de la Base Administration	218
MIUS - Présentation générale	176	Sauvegarde de la Base Administration	218
MIUS - Entrées Utilisateur	176	Base de Développement.	218
MIUS - Description des étapes	178	Fichiers de la Base de Développement	218
MIUS - Script d'exécution	179	Fichiers de sauvegarde Base de Développement	219
MILA - Migration libellés d'erreur administration	181	Modules - fichiers spécifiques.	219
MILA - Présentation générale	181	Pac/Impact	219
MILA - Entrées Utilisateur	181	DSMS.	220
MILA - Description des étapes	182	PAF	220
MILA - Script d'exécution	184	Bibliothèques et fichiers complémentaires	220
MERT - Transformation des entités Merise	186		
MERT - Présentation générale.	186		
MERT - Entrées Utilisateur.	186		
MERT - Description des étapes	187		
MERT - Script d'exécution	188		
MIAx - Cartes de contrôles optionnelles	189	Chapitre 10. Annexes	223
MIA1 - Génération programmes	190	Installation du Modèle de la Base Administration.	223
MIA1 - Présentation générale	190	VINS - Présentation générale	223
MIA1 - Entrées Utilisateur	190	VINS - Entrées / Traitements / Résultats	223
MIA1 - Description des étapes	191	VINS - Description des étapes	224
MIA1 - Script d'exécution	193	VINS - Script d'exécution	225
MIA2 - Génération écrans	195	Installation du Modèle de la Base de Développement	227
MIA2 - Présentation générale	195	VINS - Présentation générale	227
MIA2 - Entrées Utilisateur	195	VINS - Entrées / Traitements / Résultats	228
MIA2 - Description des étapes	196	VINS - Description des étapes	229
MIA2 - Script d'exécution	198	VINS - Script d'exécution	231
MIA3 - Génération clients	200	Utilitaires d'initialisation des fichiers GS/HE/ZS	233
MIA3 - Présentation générale	200	LDGS - Présentation générale	233
MIA3 - Entrées Utilisateur	200	LDGS - Description des étapes	233
MIA3 - Description des étapes	200	LDGS - Script d'exécution	233
MIA3 - Script d'exécution	203	LDHE - Présentation générale.	234
MIA4 - Génération serveurs	204	LDHE - Description des étapes	234
MIA4 - Présentation générale	204	LDHE - Script d'exécution	234
MIA4 - Entrées Utilisateur	205	LDZS - Présentation générale	235
MIA4 - Description des étapes	205	LDZS - Description des étapes	235
MIA4 - Script d'exécution	207	LDZS - Script d'exécution	236
TRDQ - Conversion des blocs DBD en blocs SQL	209	Utilitaires d'aide à la reprise	236
TRDQ - Présentation générale.	209	UTU1 UTU2 - Adaptation des opérateurs 'UNS'	236
TRDQ - Entrées Utilisateur.	211	UTU1 - Extraction ligne 'P' avec 'UNS'	236
TRDQ - Description des étapes	212	UTU1 - Entrées Utilisateur	237
TRDQ - JCL d'exécution	214	UTU1 - Description des étapes	237
		UTU1 - Script d'exécution	237
		UTU2 - Mise à jour ligne 'P' avec 'UNS'	238
Chapitre 9. Les composants	217		
Composants de la partie Serveur.	217		

UTU2 - Entrées Utilisateur	238	UTM1 - Description des étapes	243
UTU2 - Description des étapes	239	UTM1 - Script d'exécution	243
UTU2 - Script d'exécution	239	UTM2 - Entrées Utilisateur.	244
UTM1 UTM2 - Migration anciennes méta		UTM2 - Description des étapes	244
entités	240	UTM2 - Script d'exécution	246

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. Préambule

Objet du manuel

L'objet de ce manuel est de guider l'administrateur dans l'installation du produit pour :

- la partie Serveur,
- la partie Client,
- la communication.

Lorsque l'installation est complète, il est conseillé de procéder à des tests à l'aide du jeu d'essai fourni sur le support d'installation.

Vous trouverez également une description des opérations à effectuer dans le cas de l'installation d'une version de correction.

Un certain nombre d'actions préalables d'administration doivent être effectuées en TP dans Administrator workbench pour que devienne opérationnelle une installation ou réinstallation de VisualAge Pacbase. Ces actions sont documentées dans le "Guide d'utilisation AD workbench", chapitre "Actions préalables d'administration".

Architecture du produit

VisualAge Pacbase est un outil de conception, de développement et de maintenance d'applications eBusiness de type graphique (GUI) ou mode caractère (TUI), Web, conversationnel ou batch.

Cet outil est constitué :

- d'une partie Serveur (incluant l'interface mode caractère),
- d'une partie Client (interface graphique).

Ces deux parties communiquent entre elles grâce à un middleware encapsulé et fourni par IBM.

NOTE : L'utilisation des écrans en mode caractère reste possible pour certaines fonctionnalités.

Vous trouverez une description détaillée des composants Serveur dans le chapitre "Les composants".

La partie Serveur

Elle est constituée des éléments suivants :

- les éléments du système : programmes, fichiers (dont documentation en ligne) et paramètres.
- la Base Administration : contenant les paramètres utilisateur et autres paramètres.
- une (ou plusieurs) Base(s) de Développement.

La partie Client

Elle est constituée de cinq composants qui peuvent être installés séparément :

- Administrator & Developer workbench (qui peut également être installé séparément).

Developer workbench inclut les modules suivants :

- Batch,
- Dialogue,
- eBusiness (trois Outils eBusiness et Services Modeler inclus),
- Services Modeler.

Chacun de ces modules s'exécute de façon indépendante.

- Les outils eBusiness :
 - Générateur de Proxy,
 - Editeur de Localisations (Location Editor),
 - Outil de test (Services Test Facility),
 - VisualAge Pacbase Connector,
 - VisualAge Pacbase Web Connection (Dialog Web Revamping Generator),
 - Web application models (WAM).
- La Station de Travail VisualAge Pacbase (Pacdesign et Pacbench)
- Web application models (WAM)
- Middleware

Les fonctions de communication permettent de gérer les échanges entre les parties Serveur et Client du produit en utilisant, selon l'environnement, les principaux protocoles de communication du marché.

Contenu de la livraison

Le contenu d'une livraison du produit varie selon les termes de votre commande :

- Guide d'Installation,
- CD-Rom pour l'installation de la partie Serveur,
- CD-Rom pour l'installation des composants du poste Client,
- CD-Rom de documentation.

Bibliographie

- Un certain nombre d'actions préalables d'administration doivent être effectuées en TP dans Administrator workbench pour que devienne opérationnelle une installation ou réinstallation de VisualAge Pacbase. Ces actions sont documentées dans le "Guide d'utilisation AD workbench", chapitre "Actions préalables d'administration".
- Les procédures, utilisées par l'Administrateur pour la gestion des Bases et l'administration des versions, et les utilitaires fournis, sont documentés dans le Guide des Procédures de l'Administrateur.
- Pour les sites IBM, l'installation d'un Système de Sécurité, assurant le contrôle des codes utilisateur, des mots de passe et des autorisations d'accès, est documentée dans le chapitre "Installation de la partie Serveur", sous-chapitre "Installations complémentaires", et également dans l'aide en ligne de Administrator workbench.

Chapitre 2. Prérequis

Prérequis partie Serveur

Matériel et logiciel

- Architecture : Un serveur Windows/NT, Windows 2000, Windows XP ou Windows 2003.
- Processeur : Poste supportant soit Windows NT server (4.0 minimum), soit Windows 2000, soit Windows XP, soit Windows 2003.
- Mémoire : 96 Mo. La mémoire nécessaire peut augmenter en fonction du nombre de serveurs installés sur la même machine.
- Logiciels :
 - Microsoft Windows Script, version 5.1 minimum. Vous pouvez la télécharger à partir de l'url suivante :
www.microsoft.com/downloads
- Pour la version MICROFOCUS :
MICRO FOCUS Application Server 3.1, 4.0 et suivantes
- Pour la version Acucorp :
ACUCOBOL-GT 6.1 ou 7.0

ATTENTION :

Outre l'installation du runtime COBOL, il faut mettre à jour les variables d'environnement du système (PATH, COBPATH, ...).

Dans le cas d'ACUCOBOL-GT, la variable PATH doit être complétée par le chemin correspondant au runtime (en général, de la forme ..\AcuGT\bin). Cette opération doit être suivie d'un reboot de la machine.

- Support d'installation : Lecteur de CD-ROM

Espace disque

L'espace disque occupé par les fichiers dépend de l'importance des applications gérées par le système.

Le tableau ci-dessous indique approximativement (en millions d'octets) l'espace disque nécessaire à l'installation des serveurs :

Encombrement installation	Total
Total installation	80
Total Système	54
Total fichiers Utilisateur de tests d'installation	35

Encombrement installation	Total
Total fichiers Administrateur de tests d'installation	260 ko

Prérequis partie Client

Matériel

Les caractéristiques matérielles nécessaires à l'installation des composants clients VisualAge Pacbase sont les suivantes :

- Processeur : Intel Pentium III 450 Mhz minimum ou processeur compatible.
- Moniteur : Moniteur graphique (800x600) VGA ou plus grande résolution (XGA ou SVGA).
- Lecteur de CD-ROM.
- Carte : carte adaptée au réseau du site.
- Mémoire (RAM): 512 Mo (1 Go conseillé).

Espace disque

Espace disque nécessaire à l'installation :

- 100 Mo pour Administrator & Developer workbench
- 30 Mo pour eBusiness Tools
- 85 Mo pour "IBM SDK for Java 2"
- 25 Mo pour le Middleware
- 6 Mo pour WAM
- 30 Mo pour la Station de Travail VisualAge Pacbase (avec une Méthode incluse).

Si "IBM SDK for Java 2" n'est pas installé sur votre poste, son installation sera automatiquement déclenchée.

Pour une installation opérationnelle de AD workbench par exemple, il vous faudra un espace disque minimum de 215 Mo (100 Mo pour AD workbench, 85 Mo pour "IBM SDK for Java 2" et 30 Mo pour le Middleware).

Logiciel

Les composants Client de VisualAge Pacbase requièrent un poste équipé d'un Windows 32 bits.

Voir aussi le chapitre "Installation des composants Client", sous-chapitre "Informations préalables".

Communications

Pour permettre la communication entre les composants clients et les serveurs dans un environnement WINDOWS, le protocole de communication est VAP Socket.

Chapitre 3. Installation de la partie Serveur

Installation

Le produit est livré sur support CD-Rom. Celui-ci contient les éléments suivants :

- Data1.Cab (tous les fichiers du serveur au format compressé)
- setup.exe (exécutable d'installation),
- setup.ini (paramètres d'installation),
- VisualAge Pacbase Server.Msi (table d'installation pour Windows Installer).

L'installation requiert les droits Administrateur NT.

L'unité et le nom du répertoire peuvent être modifiés en cours d'installation. Par défaut, le serveur est installé dans le répertoire racine :

C:\Program Files\IBM\VisualAge Pacbase\

L'installation se fait dans un répertoire particulier, \Server, et se déroule en trois étapes :

1. installation du Système, qui crée le répertoire \SYS\, avec au préalable la saisie du Code Langue de Génération (français ou anglais (par défaut)).
2. installation de la Base d'Administration , qui crée le répertoire \CONFIG\ADMIN\ et le répertoire \DATA\ADMIN\
3. installation d'une Base de Développement, qui crée le répertoire \CONFIG\[nom_base]\ et le répertoire \DATA\[nom_base]\

Chaque étape peut être exécutée soit indépendamment, soit de manière enchaînée, mais toujours dans l'ordre indiqué.

Toutes les Bases sont créées avec des données de tests d'installation et des paramètres de configuration.

Si le code saisi ne correspond ni à la base livrée (BVAP) ni à une base prédéfinie dans la base administration, alors cette référence est mise à jour par UPGP dans la base administration. A vous de compléter son référencement par la Station Administration.

ATTENTION :

Dans tout ce manuel, la dénomination [nom_base] utilisée dans la présentation du nom complet du répertoire signifie : Nom du répertoire concernant la base pour les données du serveur et non pas le code base.

Installation du Système

L'exécution de setup.exe lance le programme d'installation qui guide l'utilisateur.

Après l'écran de BIENVENUE, certaines caractéristiques modifiables du poste s'affichent :

- Nom et Organisation du propriétaire
- Localisation d'installation (par défaut : C:\Program Files\IBM\VisualAge Pacbase\)

L'installation copie :

- le fichier de conversions, le fichier des messages produits par les procédures en cas d'erreur ou pour information,
- les squelettes dans les différentes langues supportées par le produit,
- les programmes,
- les procédures,
- l'utilitaire de création de base,
- les modèles des scripts de lancement.

Durant ce traitement le fichier des libellés d'erreur, AE, est mis à jour pour prendre en compte le code langue saisi. Un compte-rendu d'exécution, INAEMSG.log, est produit dans le répertoire temporaire de WINDOWS (\TMP ou \TEMP) ou directement sous C.

Remarque :

Les load-modules BVPDS600 et BVPDS610 d'interface entre VA Pacbase et DSMS 2.5 sont livrés en standard ; en cas d'utilisation de DSMS il faut donc adapter la procédure DEXP pour utiliser ces nouveaux load-modules à la place de PDS600 et PDS610 qui ne sont plus compatibles depuis la version VA Pacbase 3.0.

Un élément du Système : La Procédure

Définition

Une procédure est un script Windows Scripting (.wsf), incluant des scripts Visual Basic Script (.vbs).

Chaque procédure n'est exécutable qu'avec un Script de lancement (.wsf) qui comporte les informations suivantes :

- une zone <resource id> constituée des entrées (141 lignes maximum) ou la définition d'un fichier des entrées par WshEnv("BVP_Input") = "full file name" (c'est la valeur prioritaire),
- le code utilisateur,
- le nom externe de la base (répertoire sous \DATA),
- les assignations complètes des fichiers paramétrables de la procédure, exemple : WshEnv("PACA15_PAC7DC") = "DSMS full file name".

Il est obligatoire de renseigner ces informations avant le lancement de la procédure.

D'autre part, d'autres éléments peuvent être précisés pour assigner des fichiers ou paramétrer les procédures (comme le type de visibilité du message : affiché ou écrit dans un fichier).

Dans tous les cas il faut vérifier les entrées utilisateurs livrées pour les rendre conformes à votre environnement (<resource id> ou BVP_Input)

Les fichiers temporaires, les comptes-rendus d'exécution et les fichiers produits sont situés dans des répertoires créés dynamiquement.

- Les fichiers temporaires dans : \DATA\[nom_base]\TMP\[code_utilisateur]\[nom_proc]-[numéro]

Ils sont supprimés à la fin de l'exécution de la procédure. Si vous ne désirez pas cette suppression, il faut, au préalable, positionner la variable "DelDo" à la valeur "NO". Ce positionnement peut être effectué de deux manières :

- soit dans le script de lancement en utilisant la commande WshEnv(DelDo)="NO". Dans ce cas, la variable n'est valable que pour la procédure.
- soit dans le fichier Init.vbs par un DelDo = "NO". Dans ce cas, la variable est valable pour TOUTES les procédures.

Dans les deux cas, les fichiers temporaires seront alors à détruire manuellement.

- Les fichiers produits, les comptes-rendus dans : \DATA\[nom_base]\USERS\[code_utilisateur]\ [nom_proc]-[numéro]

Ce numéro est un numéro d'exécution applicatif, par défaut le numéro de process de la procédure.

ATTENTION :

Dans VisualAge Pacbase pour Acucorp, lorsqu'une procédure comportant un tri COBOL s'exécute anormalement, il arrive que les fichiers temporaires de tri ne soient pas automatiquement supprimés. Ces fichiers, au nombre

de 16, dont le nom commence par "T" , se trouvent dans le répertoire temporaire de WINDOWS (\TMP ou \TEMP) ou directement sous "C:\". Ils possèdent la date et l'heure du lancement de la procédure.

Mode d'exécution

Vous pouvez exécuter le Script de lancement (BVPproc.wsf) et le superviseur (PRBVP.vbs) :

- via une ligne de commande,
- en double-cliquant dessus,
- via le 'Menu Démarrer'. Accédez au raccourci : [[nom_base] Database Utilities] du groupe [VisualAge Pacbase Server] de [nom_base] et saisissez [Nom_Script].

NB : Le superviseur PRBVP.vbs est une procédure VBScript indépendante qui lance l'exécution des Scripts de lancement. Il nécessite 2 arguments : [nom_base] [Nom_script] , [nom_base] étant le répertoire sous \DATA.

Pour chaque procédure (on identifiera le nom de la procédure par xxxx), on définit plusieurs éléments nécessaires à son exécution.

1. Soit dans le Script de lancement ; Ce script de lancement est en Windows Scripting identifié comme suit : BVPxxxx.wsf.
 - a. Code utilisateur ,
 - b. Nom externe de la base (répertoire sous \DATA) ,
 - c. Les données en entrée sont décrites dans la zone <resource id> du script et récupérées dans une variable d'environnement BVP_Resource. Le volume des données de cette zone est limité à 141 lignes maximum, mais son utilisation peut être remplacée par celle d'un fichier désigné dans la variable d'environnement BVP_Input. La limitation du volume des données est ainsi éliminée.
 - d. Les fichiers autres que le fichier des entrées peuvent être assignés "en substitution" en définissant une variable d'environnement avec :
 - . "NomStep_NomFichier" pour une assignation substituée dans une étape (ou "step"),
 - . ou "NomProc_NomFichier" pour une assignation substituée dans toute la procédure.Exemple : Définition de la variable d'environnement
WshEnv("PTU907_PAC7IN") = Fil_In with pathname pour une assignation substituée dans l'étape PTU907 de la procédure PPAS.
 - e. Les messages d'information ou d'erreur sont soit affichés à l'écran (MsgTyp = 1), soit édités dans un fichier (MsgTyp =2) (valeur par défaut).

La variable d'environnement BVP_Msg permet de paramétrer le choix.

- f. Le mode debug peut être activé pour la procédure en positionnant 2 variables : DelDo qui permet de garder les fichiers temporaires et BVPTrace qui permet d'activer le mode Trace. Ces variables peuvent être positionnées indépendamment l'une de l'autre :

```
WshEnv("DelDo")="NO"
```

```
WshEnv("BVPTrace")="YES"
```

Dans ce cas, les variables ne sont valorisées que pour la procédure.

ATTENTION : VA Pac ne gère ni les erreurs du script de lancement, ni les informations manquantes dans le script de lancement.

2. Soit dans la partie commune à toutes les procédures ; Cette partie commune est une inclusion VBScript, dans chaque procédure, ayant pour nom "INIT.vbs".

- a. Récupération des arguments : [base] et [user]

- b. Affectation de 2 paramètres avec 2 valeurs possibles :

DelDo = "YES"/"NO" : les fichiers temporaires sont supprimés (YES) ou non (NO)

BVPTrace = "YES"/"NO" :

Par défaut, le paramètre BVPTrace est positionné à la valeur "NO" dans le fichier init.vbs. Dans ce cas, seuls les "displays" COBOL sont écrits dans le fichier DISPxxxx.log.

Si le paramètre BVPTrace est positionné à la valeur "YES", le contexte VA Pac (liste des répertoires utilisés) est écrit dans le fichier xxxxMSG.log.

Sous Acucobol, la trace est activée et les anomalies d'exécution sont écrites dans le fichier acutrace.txt.

Sous Microfocus, les anomalies d'exécution sont écrites dans le fichier xxxxMSG.log avec éventuellement plus de détails dans le fichier DISPxxxx.log, notamment pour ce qui concerne les erreurs d'entrée sortie sur les fichiers.

De plus, si vous disposez d'un runtime Microfocus supérieur à la version 4.0, vous avez la possibilité d'activer la Consolidated Trace Facility (CTF). Cette trace est activée pour les programmes indiqués dans le fichier de configuration mftrace.cfg, livré dans le répertoire \$INSTDIR\Server\Config\Ctf.

Ce fichier de configuration doit être modifié pour y indiquer notamment le code du programme (de type BVPxxxx) que vous voulez tracer dans le paramètre mftrace.application. Le résultat de la trace bvptrace.txt est automatiquement calculé lors du lancement de la procédure et se situe dans le répertoire contenant le compte-rendu d'exécution de la procédure.

Toute modification des paramètres `mfttrace.emitter.textfile` est donc à éviter.

Toutes les informations concernant le paramétrage de ce fichier de configuration peuvent être trouvées à l'adresse :

<http://supportline.microfocus.com/Documentation/books/sx40sp2/sgpubb.htm>

(Cliquez sur 'Configuration File Entries').

- c. Constitution du fichier MB, données en entrée, comportant le Code Utilisateur, le Mot de passe, parfois une commande, et des éléments d'analyse.

Si le code utilisateur `[user]` n'est pas dans le script de lancement, on va le chercher dans le fichier des entrées.

Si les éléments nécessaires à l'exécution de la procédure sont absents, il y a une erreur.

Remarques :

Pour les procédures de reprise, le fichier en entrée peut être défini dans le Script de lancement de la procédure par une variable d'environnement : `WshEnv("NStep_NFile") = "old 2.5 full file name"`, permettant la substitution dans la procédure concernée. Le paramètre "USER" est obligatoire pour les procédures de Reprise, sauf si la procédure admet un fichier des entrées Utilisateur distinct du fichier de la version 2.5.

Désormais, on définit le fichier PC (sauvegarde de la Base) par une variable d'environnement : `WshEnv("BVP_SaveName")`. Cette variable permet de créer le paramètre de définition des assignations PAC7PC, PAC7PD, PAC7PY. Ainsi est spécifié correctement le fichier PC résultat, sans risque d'écrasement des données propres à chaque traitement. Si cette variable d'environnement n'est pas présente dans le script, la valeur par défaut est : `[Rep_SAVE & "\PC]`

Remarque : la valeur de définition du fichier PC au format 2.5 en entrée de la Reprise peut être spécifiée par une valeur de substitution dans le script de lancement : `WshEnv("PTU911_PAC7MC")`. Si cette variable de substitution n'est pas présente dans le script, la valeur par défaut est : `[Rep_SAVE & "\OLDPC"]`.

Installation du Référentiel

Base Administration

Cette phase crée la Base d'Administration, qui contrôle les Bases de Développement, et ajoute des procédures 'batch' propres à l'espace administration. On ne peut installer qu'une seule Base d'Administration; elle se nomme ADMIN.

Au cours de cette phase, vous indiquerez :

- Le numéro de port de la Base (situé entre 49152 et 65535). Ce numéro est unique. La valeur proposée par défaut est 52029.
- Le code langue de la Base (par défaut, Anglais)

On peut modifier les répertoires proposés (par défaut : C:\Program Files\IBM\VisualAge Pacbase\).

Les procédures exécutées sont les suivantes :

- INGU : initialisation du fichier des utilisateurs
Cette procédure n'est exécutée que lors de la première installation.
Elle définit puis charge le fichier GU avec les codes utilisateur 'TEST' (mot de passe à blanc) et 'ADMIN' (mot de passe 'ADMIN'). Ces codes utilisateur serviront lors de l'activation des clé et lors du lancement des scripts de tests.
- REST : restauration de la Base d'Administration de test
- VINS : installation du Modèle Administration

L'installation de la Base d'Administration se fait, soit à la suite de celle du système, soit plus tard avec l'icône [Créer Nouvelle Base].

Base de Développement

Toute nouvelle Base de Développement doit être référencée dans la Base d'Administration.

Si le code saisi ne correspond ni à la base livrée (BVAP) ni à une base prédéfinie dans la Base Administration, alors cette référence est mise à jour par UPGP dans la Base d'Administration. Dans ce cas on a un libellé par défaut.

La déclaration complète d'une nouvelle base se fait avec Administrator workbench. Pour plus d'informations à ce sujet, reportez-vous au chapitre 'Installation des composants Client' et également au 'Guide d'utilisation AD workbench'.

NOTE : La base de test BVAP livrée à l'installation est déjà déclarée dans la Base d'Administration.

Les opérations décrites ci-dessous sont à exécuter pour chaque création d'une nouvelle Base de Développement.

Pour installer une Base de Développement, vous indiquerez les paramètres suivants :

- Le nom de la Base (8 car.).

- Le code de la Base (4 car., tel que déclaré dans la Base d'Administration).
- Le numéro de port de la Base (compris entre 49152 et 65535, unique pour chaque Base). La valeur proposée, pour la première Base de Développement, par défaut est 3676 (numéro réservé).

Il faut reporter manuellement ce numéro dans le fichier vaplocat.ini (voir chapitre 4 "Installation/Réinstallation des composants clients", sous-chapitre "Principes et Architectures de la communication").

- Le code langue de la Base (par défaut, Anglais).
- On peut installer plusieurs Bases de Développement, chacune ayant son propre environnement.

Les répertoires proposés sont modifiables (par défaut : C:\Program Files\IBM\VisualAge Pacbase\).

Les procédures exécutées sont les suivantes :

- UPGP : déclaration de la Base de Développement dans la Base d'Administration,
- REST : restauration de la base de test,
- VINS : installation du Modèle de la Base de Développement.

Saisie de la clé d'accès

Cette étape est effectuée à partir de Administrator workbench ; elle comporte les opérations suivantes :

- Saisie de la clé d'accès,
- Définition des cibles de génération,
- Activation de la clé.

Ces opérations sont présentées dans le 'Guide d'utilisation AD workbench', chapitre "Actions préalables d'administration". Elles sont documentées en détail dans l'aide en ligne de Administrator workbench.

Sauvegarde de la Base Administration

Cette étape est conseillée pour éviter, en cas de problème ultérieur, de recommencer la saisie de la clé.

Créer nouvelle Base

Cet icône du menu "Démarrer" installe la Base Administration ou une Base de Développement, à tout moment après l'installation du système.

L'installation demande un code utilisateur de niveau Administrateur.

Les utilisateurs sont créés avec la Station "Administrator workbench". La Base Administration livrée contient un utilisateur de niveau "ADMIN" avec le mot de passe "ADMIN".

Paramètres de Configuration

Pour chaque base, on crée un répertoire de Configuration avec

- 1- BvpServer.ini, qui contient les paramètres du serveur conversationnel.
- 2- Fhredir.cfg, qui contient les paramètres du "Rollback".

Déréférencer une Base

Le raccourci 'Delete [nom_base] Database' situé sous le Groupe de programmes [VisualAge Pacbase 3.5 Server] du Menu Démarrer, permet de déréférencer une base sur une plateforme Windows.

Cette procédure effectue les opérations suivantes :

- suppression du service windows pour une base donnée.
- suppression des informations de configuration (répertoire CONFIG).
- suppression des informations spécifiques à la base dans la base de registres Windows.
- suppression des raccourcis pour la base.
- renommage des répertoires de données de la base.

Liste des dates des composants

Procédure INSL

Cette procédure édite :

- la liste des programmes batch, programmes TP et leurs caractéristiques de génération.
- Cette liste doit être éditée, en cas de détection d'anomalies dans le fonctionnement du système, pour transmettre au support du produit les références de l'installation.

Dans tous les cas il faut vérifier les entrées utilisateurs livrées pour les rendre conformes à votre environnement.

INSL - Script d'exécution

```
| -----  
|          VISUALAGE PACBASE  
| -----  
|          - LIST OF INSTALLED PROGRAMS AND FILES -  
| -----
```

```

'
'
'
' -----
'

Dim fso, dir, subdir, file, PathFile
Dim ListExclude
Dim regEx, retVal

Set WshShell = Wscript.CreateObject("Wscript.Shell")

' Set Path of programs :
' -----
Reg_PROD = "HKLM\SOFTWARE\IBM\BVP VisualAge Pacbase 3.5\"
Rep_INST = WshShell.RegRead (Reg_PROD)

Rep_PGM = Rep_INST & "Server\Sys\pgm"
Rep_CFG = Rep_INST & "Server\Config"

' Set List of cobol programs :
' -----
ListCobol = Rep_CFG & "\" & "ListCob.txt"
Set fso = WScript.CreateObject("Scripting.FileSystemObject")
Set dir = fso.GetFolder(Rep_PGM)
Set StreamLst = fso.CreateTextFile(ListCobol, true, false)
Set regEx = New RegExp

ListExclude = " BVPSEVER.EXE BVPDIAL.EXE BVPWSH.OCX "
ListExclude = " BVPLOCK.DLL BVPPAUSE.DLL      " & ListExclude
ListExclude = " BVPSPLIT.EXE BVPTRANS.EXE     " & ListExclude
ListExclude = " CODEPAGECONV.DLL VAPUTIL.DLL  " & ListExclude
ListExclude = " BVP.ICO BVPSEVER.ICO          " & ListExclude

For Each file In dir.Files
    regEx.Pattern = file.Name
    regEx.IgnoreCase = True
    retVal = regEx.Test(ListExclude)
    'Cobol program
    If retVal = False Then
        LineLst = Rep_PGM & "\" & file.Name & vbCRLF
        StreamLst.Write(LineLst)
    End If
Next
Set StreamLst = Nothing

' Launch INSL :
' -----
' Debug Mode
'CREATE_LOG = True
CREATE_LOG = False

Set WshEnv = WshShell.Environment("PROCESS")
WshEnv("LIST") = ListCobol

```

```

INSL_LOG=chr(34) & Rep_CFG & "\BVPINSL.log" & chr(34)
INSL_OUT=chr(34) & Rep_CFG & "\BVPINSL.txt" & chr(34)

WshEnv("PATH") = Rep_PGM & ";" & WshEnv("PATH")
sCmd = "Bvpdate.exe " & INSL_OUT

If ( CREATE_LOG = True ) Then
    sCmd = "cmd.exe /c " & sCmd & " 1>" & INSL_LOG & " 2>&1"
    Return = WshShell.Run( sCmd, 7 , TRUE)
Else
    Return = WshShell.Run( sCmd, 10 , TRUE)
End If

' End of job :
'-----
If ( Return <> 0 ) Then
    Display = "Abnormal end of job " & _
    WScript.ScriptName & "."
else
    ' delete list of cobol programs
    Set FileD = fso.GetFile(ListCobol)
    FileD.Delete
    Display = "End of job " & WScript.ScriptName & _
    vbCRLF & vbCRLF & _
    "File " & INSL_OUT & " has been created."
End If

If ( CREATE_LOG = True ) Then
    Display = Display & vbCRLF & vbCRLF & _
    "File " & INSL_LOG & " has been created."
End If

WScript.Echo(Display)
Wscript.Quit (0)

```

Connexion

Lancement et Arrêt d'un serveur

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".

Lorsque le listener présente des anomalies d'exécution, des messages peuvent être reportés dans la liste des événements de l'outil d'administration "Event Viewer" sous "Log/Application".

Ces événements de catégorie "ERROR" peuvent donner une première indication. Ils correspondent en général à des problèmes d'environnement d'exécution.

Cependant en cas d'anomalie de fonctionnement le support du produit peut vous demander d'activer un "mode trace" pour détecter l'origine du problème.

- MODE TRACE

Différents niveaux de trace peuvent être mis en oeuvre :

- Niveau 1
trace minimum permettant de suivre le fonctionnement du listener avec les appels au moniteur de communication COBOL,
- Niveau 2
trace détaillée du fonctionnement du listener,
- Niveau 4
trace des messages échangés entre le listener et le poste client.

Dans la procédure de lancement du listener, "server.wsf", il existe une variable d'environnement SRV_TRACE pour activer le mode trace. Pour utiliser un autre niveau de trace, il est nécessaire de relancer le listener en ayant positionné la variable SRV_TRACE.

EXEMPLE :

SRV_TRACE=1 pour une trace de niveau 1

SRV_TRACE=3 pour une trace de niveau 1 et 2

SRV_TRACE=5 pour une trace de niveau 1 et 4

Les fichiers résultat de la trace sont de deux types :

- srv[numéro_process].txt

pour tracer le listener (BvpServer.exe).

- dial[numéro_process].txt

pour tracer le fonctionnement de chaque connexion au listener (BvpDial.exe).

Ces fichiers sont produits sous le répertoire SRV_DIR assigné dans server.wsf, par défaut, à la valeur :

.../data/[nom_de_base]/tmp

La variable d'environnement SRV_TRACE_DEL est positionnée pour permettre de conserver tout ou partie des traces produites par l'exécution du listener dans "dialnnn.txt" :

SRV_TRACE_DEL : "ON" (valeur par défaut)

on ne conserve que les traces produites par une erreur de fonctionnement.

SRV_TRACE_DEL : "OFF"

on conserve toutes les traces produites.

Base sous contrôle DSMS

Dans le cas où la base [nom base] est sous contrôle DSMS, les fichiers DSMS doivent à ce moment être déclarés lors du lancement du listener de la base [nom base].

Pour ce faire, vous devez valider les lignes prévues à cet effet dans la procédure "Server.wsf" et y renseigner le nom de la base DSMS installée sur votre poste.

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.

Moniteur Gestion des Listeners

Un outil permettant de gérer les listeners est disponible dans le répertoire SYS/PGM.

Cet utilitaire permet d'afficher l'état des clients connectés sur un serveur identifié par le numéro de socket utilisé.

Cet outil se lance de la manière suivante :

TP.exe <N°Socket_Server>

L'exécution de cette commande affiche notamment :

- Le PID,
- L'adresse IP,
- Le programme en cours d'exécution,
- L'utilisateur connecté,
- L'heure de connexion.

Chapitre 4. Installation/Réinstallation des composants Client

Informations préalables

- Pour installer les composants Client de VisualAge Pacbase sur un poste Windows, vous devez avoir le profil Administrateur sur ce poste.
- L'installation des composants Client s'exécute avec InstallShield for Windows Installer (ISWI). Si Windows Installer n'est pas installé sur le poste, il sera automatiquement installé.
- Vous avez également besoin de Microsoft Windows Script, version 5.1 minimum. Vous pouvez la télécharger à partir de l'URL suivante :
<http://www.microsoft.com/downloads>
- Les composants Administrator & Developer workbench et Outils eBusiness requièrent, pour leur fonction Aide en Ligne, l'utilisation d'un navigateur internet standard (Mozilla Firefox ou Internet Explorer conseillés).
- L'installation d'un composant Client n'exige pas l'installation préalable au niveau Serveur de la Base ou des Bases VA Pac à laquelle/auxquelles il se connectera.
Toutefois, le code de chaque Base VA Pac que vous allez fournir lors de l'installation de certains composants Client, devra être strictement réutilisé lors de l'installation de chacune de ces Bases pour la partie serveur.

Un certain nombre d'actions préalables d'administration doivent être effectuées en TP dans Administrator workbench pour que devienne opérationnelle une installation ou réinstallation de VisualAge Pacbase. Ces actions sont documentées dans le "Guide d'utilisation AD workbench", chapitre "Actions préalables d'administration".

Répertoire racine

Par défaut, le répertoire racine de tous les composants Client VisualAge Pacbase est :

C:\Program Files\IBM\VisualAge_Pacbase_35

Si vous devez vous connecter simultanément à AD workbench et au module Pacbench de la Station de Travail VisualAge Pacbase, les deux composants doivent impérativement être installés sous le même répertoire racine.

Les répertoires de cette racine ne sont modifiables qu'une seule fois, au début, c'est-à-dire à la première installation du premier composant.

NOTE : Ne pas utiliser de blancs dans les noms des répertoires.

Que cette racine ait été modifiée ou non, les autres composants s'installeront nécessairement sous cette même racine.

Par contre, lors de l'installation d'une version ultérieure d'un composant, le choix de la nouvelle racine est libre, sachant qu'elle doit être différente de la racine de la version courante (contrôlé à l'installation).

Par "version" il faut comprendre la version désignée par le premier identifiant, par exemple "3.5". Les versions désignées par "Vnn" constituent donc des sous-versions.

Pour plus d'informations, voir le sous-chapitre "Mise à jour, Ajout, Désinstallation de composants".

Lancement de l'installation

Insérer dans le lecteur le CD-ROM livré.

L'exécution de setup.exe lance l'interface graphique du Wizard InstallShield qui va vous guider tout au long de l'installation.

Le premier panneau affiche le texte de la licence du runtime Java. Vous acceptez les termes de la licence ; le panneau suivant vous demande alors votre identification (Nom et Organisation).

NOTE : Tous les composants Client de VisualAge Pacbase sont installés en mode utilisation partagée pour le poste.

Ensuite, la liste des composants Client de l'offre VisualAge Pacbase s'affiche sur l'écran.

Choisissez le composant Client que vous souhaitez installer.

La suite de l'installation est décrite dans les sous-chapitres spécifiques à chaque composant.

Principes et architectures de la communication

Cette section vous présente les principes de communication entre les composants Client et le serveur VisualAge Pacbase.

Les pages suivantes contiennent des informations indispensables au choix du protocole de communication et au paramétrage du middleware associé.

Ces informations vous seront également utiles lors d'installations ultérieures (autres composants Client ou nouvelles versions de composants déjà installés).

Les protocoles de communication disponibles

- Si le serveur VisualAge Pacbase est installé sous Windows ou UNIX, le protocole VAP Socket devra être utilisé.
- Si le serveur est installé sous MVS/CICS, vous avez le choix entre MVS CICS Socket, MVS CICS CPI-C ou MQ-CICS Bridge, en fonction de l'installation effectuée sur le serveur (voir le chapitre correspondant).
- Si le serveur est installé sous MVS/IMS, vous avez le choix entre IMS Connect et MQ-IMS Bridge, en fonction de l'installation effectuée sur le serveur (voir le chapitre correspondant).

LE MIDDLEWARE

Le middleware doit toujours être installé sur chaque poste de développeur. Cette installation est automatiquement déclenchée à la première installation sous une racine donnée d'un des composants Client suivants : Administrator & Developer workbench, Station de Travail VisualAge Pacbase et les Outils eBusiness.

Le middleware installé sur chaque poste Développeur assure ainsi la communication directe entre le ou les composants Client et le serveur.

Cependant, vous pouvez aussi opter pour une communication de type gateway.

Une gateway effectue une gestion centralisée et optimisée des accès serveur.

Dans ce contexte, le middleware est également à installer sur un serveur intermédiaire, en sélectionnant l'item Middleware dans la liste des composants Client installables (voir le sous-chapitre correspondant).

Les composants Client communiquent alors avec une passerelle (la Gateway VisualAge Pacbase) qui s'exécute sur ce serveur intermédiaire.

LES FICHIERS DE LA COMMUNICATION

Pour les deux composants Administrator & Developer workbench et Station de Travail VisualAge Pacbase, le paramétrage de la communication s'articule nécessairement sur deux fichiers, bases.ini et vaplocat.ini.

Le fichier vaplocat.ini est également exploité par le composant Outils eBusiness.

Ces fichiers sont créés de façon transparente et se trouvent dans un répertoire nommé 'common'.

Une réinstallation n'a pas d'impact sur les fichiers bases.ini et vaplocat.ini. Un fichier base_new.ini est créé pour information uniquement ; il contient la dernière version livrée de ce fichier.

IMPORTANT : Pour ajouter/supprimer ultérieurement des Bases VisualAge Pacbase, voire modifier certains paramètres liés à la communication, vous devrez modifier ces fichiers.

Les détails concernant la structure des données dans ces deux fichiers sont fournis à la fin de ce chapitre (Mise à jour des paramètres de la communication).

Le fichier vaplocat.ini :

- Lorsque la communication est directe, le fichier vaplocat.ini exploité se trouve sur chaque poste Développeur.
- Lorsque la communication est de type gateway, le fichier vaplocat.ini exploité se trouve sur le serveur intermédiaire.

Dans les deux cas, ce fichier est placé dans le sous-répertoire 'common' du répertoire racine de l'installation.

C'est dans ce fichier que sont décrite(s) la (ou les) localisation(s).

Une localisation :

- Identifie le protocole utilisé pour accéder au serveur VisualAge Pacbase,
- Donne les coordonnées physiques de ce serveur pour ce protocole,
- Définit les paramètres de communication nécessaires au bon fonctionnement du protocole.

Le fichier bases.ini :

Le fichier bases.ini se trouve sur le poste de chaque développeur, dans le sous-répertoire 'common' du répertoire racine de l'installation.

Ce fichier établit la liste des Bases VisualAge Pacbase accessibles. A chaque Base est associé un nom de localisation (location en anglais).

Plusieurs Bases peuvent avoir une localisation identique. Ces localisations sont définies dans l'autre fichier (vaplocat.ini).

Administrator & Developer workbench

Si "IBM SDK for Java 2" n'est pas installé sur votre poste, son installation sera automatiquement déclenchée. Pour cette installation, comme pour celle de Administrator & Developer workbench, la racine utilisée dépend du contexte courant de l'installation. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Le script d'installation vous demande de sélectionner Administrator workbench ou Developer workbench ou les deux :

- Administrator workbench
- Developer workbench

Developer workbench inclut les modules suivants, chacun étant exécutable séparément.

- le module Batch,
- le module eBusiness et trois des outils eBusiness :
 - Générateur de Proxy
 - Editeur de Localisations (Location Editor)
 - Outil de test (Services Test Facility)
- le module Services Modeler (toutes ses fonctionnalités étant également disponibles dans le module eBusiness).

Que vous installiez Administrator ou Developer workbench ou les deux, l'utilitaire CFM (Configuration File Manager) est toujours installé. CFM permet d'inhiber l'affichage de browsers non utiles pour vos équipes et/ou de permettre l'affichage des browsers spécifiques de Méta Entités définies sur le site. Consultez le Support Technique VisualAge Pacbase pour plus d'informations à ce sujet.

Dans le panneau suivant, vous indiquez le mode de communication utilisé (communication directe ou via gateway).

NOTE : Des informations IMPORTANTES concernant la communication vous sont données en début de chapitre.

Ce panneau ne s'affiche pas si vous avez déjà installé, sous la même racine, Administrator & Developer workbench ou la Station de Travail VisualAge Pacbase.

- Si vous choisissez la communication directe, le script d'installation du middleware s'enclenchera automatiquement, une fois l'installation du workbench terminée, et vous demandera de configurer les paramètres de la communication.

Pour la documentation de cette partie de l'installation, voir le sous-chapitre Middleware.

- Si vous choisissez gateway, renseignez d'abord l'adresse IP de cette gateway. L'installation du middleware sur le poste Développeur -- également nécessaire dans ce contexte -- sera alors automatiquement enclenchée après celle du workbench.

Ensuite, dans ce même contexte initial, vous indiquez la (première) Base à laquelle accèdera Administrator & Developer workbench.

Pour cela, un panneau vous propose de renseigner :

1. Le nom d'une Base VA Pac, déjà installée ou non au niveau serveur.

Les noms que vous saisissez ici seront ceux affichés dans l'assistant de connexion, indiquant ainsi à l'utilisateur la (ou les) Base(s) qu'il pourra sélectionner.

Le nom donné à chacune des Bases doit donc être suffisamment explicite pour que celles-ci soient facilement identifiables dans la liste des Bases proposées par l'assistant de connexion.

2. Le code logique de cette Base.

Longueur maximum : 4 caractères.

Si cette Base n'est pas encore installée au niveau Serveur, notez bien le code que vous saisissez ; il devra être réutilisé lors de cette installation.

NOTE : La Base dédiée à l'administration du site est automatiquement créée. Son code logique est le code réservé : "****".

3. Le nom de la Localisation.

Longueur maximum : 20 caractères. Défaut : Location-1

Plusieurs Bases peuvent pointer sur la même localisation.

4. Pour finir, vous précisez le type de contrôle que le workbench appliquera à la connexion. Référez-vous à l'Administrateur de la Base concernée, pour qu'il y ait adéquation entre le ou les contrôles imposés au niveau serveur et leur implémentation au niveau du workbench.

Cet indicateur a trois valeurs possibles, à sélectionner via deux cases à cocher :

"VisualAge Pacbase signon" :

indique que l'utilisateur aura à donner son code et son mot de passe pour se connecter à la Base VisualAge Pacbase.

"Middleware signon" :

indique que l'utilisateur aura à donner son code et son mot de passe pour se connecter au système hôte (dans les deux zones affichées sous "Références Middleware" de l'assistant de connexion).

Si seule la case "Middleware signon" est cochée, le contrôle de la connexion à la Base VA Pac est assuré par le système de sécurité.

Si vous cochez les deux cases, l'utilisateur devra donner son code et son mot de passe pour la connexion au système hôte et pour la connexion à la Base.

Vous avez ainsi défini l'accès à une première Base VA Pac. Le script d'installation vous propose alors de définir la communication et la connexion avec une autre Base, et cela, autant de fois que nécessaire.

L'installation proprement dite peut ensuite démarrer ; appuyez sur le bouton [INSTALL].

NOTE : Cette installation est suivie de celle des outils eBusiness (s'ils n'ont pas déjà été installés sous la même racine), et -- dans le contexte initial, défini ci-dessus -- de celle du middleware.

LES FICHIERS DE LANCEMENT

Les fichiers de lancement des modules sont les suivants :

```
wb_admin.bat
wb_global.bat
wb_batch.bat
wb_dialog.bat
wb_eBusiness.bat
wb_services.bat
wb_extensibility.bat
wb_cfm.bat
```

Ces fichiers se trouvent sous le répertoire racine de Administrator & Developer workbench ("ADWorkbench").

MENU DEMARRER / CHOIX PROGRAMMES

A l'issue de cette installation, vous trouverez au niveau du bureau Windows, dans le menu Démarrer/choix Programmes, la section VisualAge Pacbase 3.5 Components avec les sous-sections suivantes :

```
Administrator-Developer workbench
    Administration
    Batch
    Dialog
    eBusiness
    Services Modeler
    Extensibility
    cfm
    Global
```

Pour la fonction Publication : Open Jade et Tidy

Depuis la version 3.5 V05, la publication se fait par défaut avec la technologie XSL qui ne nécessite plus les outils OpenJade et Tidy.

Dans le cas où l'on préfère utiliser la publication DSSSL, il est nécessaire d'installer les outils préalablement cités.

Pour les versions antérieures à la 3.5 V05, l'installation des outils OpenJade et Tidy reste un prérequis obligatoire à la publication des documents.

Pour disposer des outils Open Jade et Tidy, il faut se reporter aux consignes d'installation figurant sur la page internet du support VA Pacbase <http://www.ibm.com/software/awdtools/vapacbase/download.html> .

Outils eBusiness

Les Outils eBusiness sont les suivants :

- Générateur de Proxy
- Editeur de Localisations (Location Editor)
- Outil de test (Services Test Facility)
- VisualAge Pacbase Web Connection (Dialog Web Revamping Generator)
- Web Application Models (WAM)

Cette installation permet l'utilisation des Outils eBusiness indépendamment de Developer workbench, sans communication avec un serveur VisualAge Pacbase. Les Outils eBusiness (sauf WAM) sont installés en tant que "features" et "tools" de VisualAge for Java.

Si "IBM SDK for Java 2" n'est pas installé sur votre poste, son installation sera automatiquement déclenchée.

Pour cette installation, comme pour celle des Outils eBusiness, la racine utilisée dépend du contexte courant. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Pour démarrer l'installation, cliquez sur le bouton [INSTALL].

Le composant Middleware est automatiquement installé à la suite de l'installation des Outils eBusiness s'il n'est pas déjà présent sous la racine de l'installation courante. Vous aurez alors à renseigner certains paramètres de communication.

Pour la documentation de cette partie de l'installation, voir le sous-chapitre Middleware.

Le middleware installé dans ce contexte particulier assure la communication entre les serveurs et les proxies générés.

Pour information, les paramètres de ce type de communication seront à positionner par chaque développeur en utilisant l'outil Location Editor inclus dans cette installation.

Par ailleurs, le composant des Outils eBusiness peut fonctionner en mode communication par gateway. Dans ce cas, vous devrez aussi installer le composant Middleware et configurer la communication, au niveau du serveur intermédiaire supportant la gateway VisualAge Pacbase.

NOTE : Des informations IMPORTANTES concernant la communication vous sont données en début de chapitre.

LES FICHIERS DE LANCEMENT

Les fichiers de lancement s'appellent :

- Pour le Générateur de Proxy :
vapGen.exe
- Pour l'Editeur de Localisations (Location Editor) :
vapLocationEditor.exe
- Pour l'outil de test (Services Test Facility) :
vapServicesTestFacility.exe
- Pour Pacbase Web Connection :
PacWebgen.exe

Ces fichiers se trouvent dans le sous-répertoire suivant :

ebusinesstools\bin

- Contrairement aux autres outils eBusiness, WAM est installé dans un répertoire spécifique, nommé Wam. Ce répertoire contient un fichier zippé et un fichier readme. Lisez-le et suivez les instructions qu'il contient.

MENU DEMARRER / CHOIX PROGRAMMES

A l'issue de cette installation, vous trouverez au niveau du bureau Windows, dans le menu Démarrer/choix Programmes, la section VisualAge Pacbase 3.5 Components avec les sous-sections suivantes :

eBusiness Tools

- Location Editor
- Proxy Generator
- Services Test Facility
- Dialog Web Revamping Generator

Station de Travail VisualAge Pacbase

ATTENTION : La Station de Travail fonctionne dans un environnement 16 bits; assurez-vous que le poste sur lequel fonctionne la Station supporte bien ce type d'applications.

La racine utilisée pour l'installation à effectuer dépend du contexte de cette installation. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Le premier panneau vous invite à sélectionner la langue dans laquelle l'interface de la Station de Travail VisualAge Pacbase apparaîtra.

La langue installée par défaut est l'anglais.

Le panneau suivant vous invite à sélectionner la méthode à installer.

NOTE : Si vous souhaitez installer une autre méthode sur le poste, le processus de cette installation sera à reprendre à partir du début.

Si elle est proposée, l'option "Local Install" est à sélectionner systématiquement.

NOTE : L'option "sub-features" est identique à l'option "feature" seule.

Dans le panneau suivant, vous sélectionnez :

- Le ou les modules à installer :
 - Pacdesign,
 - Pacbench.
- Le mode de connexion :
 - en mode "libre" : Choix par l'utilisateur de travailler en local ou connecté au Référentiel VisualAge Pacbase.
 - en mode "forcé" : Connexion systématique au Référentiel.

Dans le panneau suivant, vous indiquez le mode de communication utilisé (communication directe ou via gateway).

NOTE : Des informations IMPORTANTES concernant la communication vous sont données en début de chapitre.

Ce panneau ne s'affiche pas si vous avez déjà installé, sous la même racine, la Station de Travail VisualAge Pacbase ou Administrator & Developer workbench.

- Si vous choisissez la communication directe, le script d'installation du middleware s'enclenchera automatiquement, une fois l'installation de la Station de Travail terminée, et vous demandera de configurer les paramètres de la communication.
Pour la documentation de cette partie de l'installation, voir le sous-chapitre Middleware.
- Si vous choisissez gateway, renseignez ici l'adresse IP de cette gateway. L'installation du middleware sur le poste Développeur -- également nécessaire dans ce contexte -- sera automatiquement enclenchée après celle de la Station.

Ensuite, dans ce même contexte initial, vous indiquez la (première) Base à laquelle accèdera la Station de Travail VisualAge Pacbase.

Pour cela, un panneau vous propose de renseigner :

1. Le nom d'une Base VA Pac, déjà installée ou non au niveau serveur.

Les noms que vous saisissez ici seront ceux affichés dans la boîte de connexion, indiquant ainsi à l'utilisateur la (ou les) Base(s) qu'il pourra sélectionner.

Le nom donné à chacune des Bases doit donc être suffisamment explicite pour que celles-ci soient facilement identifiables dans la liste des Bases proposées par l'assistant de connexion.

NOTE : Si vous utilisez un fichier personnalisé des paramètres, indiquez, après le nom de la Base, le nom de ce fichier, encadré des signes "<" et ">".

Pour plus de détails sur ces paramètres, reportez-vous à la fin du sous-chapitre.

2. Le code logique de cette Base.

Longueur maximum : 4 caractères.

Si cette Base n'est pas encore installée au niveau Serveur, notez bien le code que vous saisissez ; il devra être réutilisé lors de cette installation.

Les codes que vous saisissez ici seront également affichés dans la boîte de connexion.

La Base dédiée à l'administration du site est automatiquement créée. Son code logique est le code réservé : "****".

NOTE : Un code logique doit être unique pour une Localisation donnée (voir le point 3. suivant).

3. Le nom de la Localisation.

Longueur maximum : 20 caractères.

Défaut : Location-1

Plusieurs Bases peuvent pointer sur la même localisation.

4. Pour finir, vous précisez le type de contrôle que la Station appliquera à la connexion. Référez-vous à l'Administrateur de la Base concernée, pour qu'il y ait adéquation entre le ou les contrôles imposés au niveau serveur et leur implémentation au niveau Station.

Cet indicateur a trois valeurs possibles, à sélectionner via deux cases à cocher :

"Pacbase is secured" :

indique que l'utilisateur aura à donner son code et son mot de passe pour se connecter à la Base VisualAge Pacbase.

"Middleware is secured" :

indique que l'utilisateur aura à donner son code et son mot de passe, dans la boîte Identification Middleware, pour se connecter au système hôte. Si seule la case Middleware est cochée, le contrôle de la connexion à la Base VA Pac est assuré par le système de sécurité.

Si vous cochez les deux cases, l'utilisateur devra donner son code et son mot de passe pour la connexion au système hôte et pour la connexion à la Base.

Vous avez ainsi défini l'accès à une première Base VA Pac. Le script d'installation vous propose alors de définir la communication et la connexion avec une autre Base, et cela, autant de fois que nécessaire.

L'installation proprement dite peut ensuite démarrer ; appuyez sur le bouton [INSTALL].

NOTE : Cette installation -- dans le contexte initial, défini ci-dessus -- est automatiquement suivie de celle du middleware.

LE FICHER DE LANCEMENT

Le fichier de lancement est le suivant :

pexec.exe

Ce fichier se trouve dans le répertoire racine de la Station de Travail VisualAge Pacbase (SPAC\nnnl où "nnn" désigne le numéro de version et où "l" identifie la langue de la version installée).

MENU DEMARRER / CHOIX PROGRAMMES

A l'issue de cette installation, vous trouverez au niveau du bureau Windows, dans le menu Démarrer/choix Programmes, la section VisualAge Pacbase 3.5 Components avec les sous-sections suivantes :

WorkStation

```
WorkStation 3.5  
WorkStation 3.5 News  
<méthode> News
```

LE FICHER DES PARAMETRES DE L'INSTALLATION

Certains paramètres de l'installation de la Station de Travail sont positionnés dans le fichier Pacbase.dat.

La procédure d'installation de la Station de Travail crée automatiquement ce fichier dans le répertoire \SPAC\NNNL où "NNN" désigne le numéro de version et "L" le code langue de la version installée.

Le fichier Pacbase.dat - nécessairement conforme à la dernière installation effectuée - est donc utilisé par défaut lors du lancement de la Station.

Cependant, vous pouvez créer un ou plusieurs fichiers de paramétrage. Ceci peut s'avérer utile dans le cas assez rare où plusieurs méthodes sont installées sur un poste. Le changement de méthode sera ainsi facilité à la reconnexion.

Les noms de ces fichiers sont libres mais doivent respecter la norme des fichiers DOS. Il est recommandé de leur donner l'extension .dat.

Vous devez constituer ces fichiers à l'image du fichier Pacbase.dat et les placer dans le même répertoire que lui.

En cas de réinstallation de la Station, les fichiers *.dat que vous avez créés ne seront pas supprimés.

DESCRIPTION DU FICHER DES PARAMETRES

Chacune des lignes qui composent ce fichier possède la structure suivante :

- un identifiant numérique sur trois caractères en positions 1 à 3 ;
- le libellé de la ligne, dont la position est libre ;
- la valeur du paramètre, entre crochets ([et]), dont la position est également libre.

Voici un exemple de fichier PACBASE.DAT :

```
001 Version Station           [350F]  
002 Serveur                   [PACBASE]  
003 Gestionnaire de communication [MWCOM]  
004 Paramètres de communication [MWCOM]  
005 Système d'exploitation     [WINDOWS]  
006 Méthode                   [MER]  
007 Disque EXE                 [C]
```

008 Disque EXE (défaut)	[C]
009 Disque Données système	[C]
010 Disque Données utilisateur	[C]
011 Mode d'exécution connexion	[E]

Le fichier Pacbase.dat ne doit pas être détruit.

Les valeurs possibles pour le paramètre Méthode sont :

VALEUR DU PARAMETRE	NOM DE LA METHODE
MER	MERISE
DON	YSM
ADM	SSADM (en anglais seulement)
OMT	OMT
FAA	IFW

ATTENTION : Les paramètres 001 à 005 et 011 ne sont pas modifiables.

LE FICHER VAPLOCAT.INI

Le fichier Vaplocat.ini indique la localisation des Bases et les paramètres de communication. Si vous modifiez ce fichier lors de l'utilisation de la Station de Travail, vous devez impérativement redémarrer votre poste.

Pour plus d'informations sur ce fichier, consultez les sous-chapitres "Informations préalables" et "Mise à jour des paramètres de la communication".

FICHIERS POUR LE SUIVI DES EDITIONS/GENERATIONS

Pour permettre l'utilisation de la fonction "Suivi des éditions et générations" dans la Station de Travail, le fichier GPMON.PRM doit contenir le chemin d'accès aux Bases auxquelles l'utilisateur doit accéder.

Ce fichier doit se trouver dans le Gestionnaire de fichiers sous le chemin suivant :

SPAC\NNNL\GPMON.PRM

Si ce fichier n'existe pas dans votre arborescence, vous devez le créer sous les répertoires indiqués ci-dessus. Il vous suffit de le nommer GPMON et de lui attribuer une extension .PRM.

Vous devez ensuite l'ouvrir, sous un Bloc-Notes par exemple, et saisir une ligne SU pour chaque Base à laquelle l'utilisateur doit accéder, comme suit :

```
SU bbbb u:/Racine/USERS,
```

bbbb pour le code Base,

u pour le lecteur racine,

Racine pour le chemin d'accès au répertoire USERS.

FICHER DES LIBELLES NON-STANDARD

Pour que la Station de Travail prenne en compte les libellés non-standard des fichiers issus d'une procédure d'édition et de génération personnalisée, vous devez créer un fichier GPMON.LBL dans lequel vous indiquez les libellés de ces fichiers.

Ce fichier doit être créé sous le répertoire BASE et doit comporter les codes et libellés de ces fichiers. Leurs extensions doivent être conformes aux extensions standards : Ix, Gx, Ox ou Dx (exemple, IP, GA...).

Le libellé commence en quatrième colonne. Les lignes du fichier ont une longueur de 60 caractères maximum.

Web Application Models (WAM)

La racine utilisée pour l'installation à effectuer dépend du contexte de cette installation. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Pour démarrer l'installation, appuyez sur le bouton [Install].

WAM est installé dans un répertoire spécifique, nommé Wam. Ce répertoire contient un fichier zippé et un fichier readme. Lisez-le et suivez les instructions qu'il contient.

Middleware

L'installation spécifique du composant Middleware, sur une machine dédiée (serveur intermédiaire), n'est nécessaire que lorsqu'une communication par gateway est mise en oeuvre.

En effet, le composant Middleware est automatiquement installé, juste après la première installation (sous une racine donnée) d'un des autres composants Client.

La racine utilisée pour l'installation à effectuer dépend du contexte de cette installation. Pour plus de détails, reportez-vous à la première page de ce chapitre.

Pour utiliser Administrator & Developer workbench ou la Station de Travail VA Pac, les paramètres de localisation de vos Bases VisualAge Pacbase doivent toujours être renseignés.

NOTE : Des informations IMPORTANTES concernant la communication vous sont données en début de chapitre.

- Si la communication est assurée via la gateway VisualAge Pacbase, l'installation du Middleware sur ce serveur intermédiaire vous demande de définir la localisation nécessaire pour une première Base VA Pac.
Pour les détails concernant la mise à jour de ce fichier, reportez-vous au sous-chapitre "Informations complémentaires", section "Mise à jour des paramètres de la communication".
- Si la communication est directe, les localisations vous sont automatiquement proposées ; en effet, elles ont été prédéfinies, juste auparavant, lors de l'installation de Administrator & Developer workbench ou de la Station de Travail.

Ensuite, et quel que soit le composant Client concerné, vous aurez à saisir un certain nombre de paramètres différents, selon le protocole utilisé.

Liste des paramètres

- VAP SOCKET
 - IP address : Adresse IP et port utilisé par le serveur VA Pac.
Ne pas saisir de "0" cadrés à gauche, source d'erreur pour certaines configurations.
- MVS CICS SOCKET
 - IP address : Adresse IP et port utilisé par le serveur VA Pac.
Ne pas saisir de "0" cadrés à gauche, source d'erreur pour certaines configurations.
 - Transaction code : Code de la transaction CICS du Moniteur de Communication VisualAge Pacbase.
 - Code Page : Valeur identifiant le codage des caractères utilisé par le serveur VisualAge Pacbase.
La valeur 1147 correspond à l'EBCDIC français
- MVS CICS CPI-C
 - Destination-id entry : BVPSCPI (valeur par défaut).

Si vous modifiez cette valeur, elle doit être identique à celle positionnée dans Symbolic destination name, paramètre inclus dans la configuration de ce protocole de communication.

- Code page : Valeur identifiant le codage des caractères utilisé par le serveur VisualAge Pacbase.

La valeur 1147 correspond à l'EBCDIC français

- MQ-CICS Bridge et MQ-IMS Bridge

- LocCcsid : Code du jeu de caractères de la machine en local (819 par défaut), utilisé pour la conversion des messages par MQSeries lors des lectures et écritures des Queues.

9 caractères maximum.

- Queue Manager name : Nom du Queue Manager local de la partie cliente.

S'il y a un MQSeries Server intermédiaire sur NT, ce paramètre doit identifier le Queue Manager NT intermédiaire et pas le Queue Manager.

4 caractères maximum.

- Request Queue name :

48 caractères maximum.

- Reply Queue name :

48 caractères maximum.

- Transaction code :

4 caractères maximum pour CICS

8 caractères maximum pour IMS

- MVS IMS Connect

- IP address : Adresse IP et port utilisé par le serveur VA Pac.

- Data Store : Nom du lien vers IMS défini dans IMS Connect (IMS Data Store ID).

- RACF group : Nom du groupe RACF pour IMS Connect.

- Code Page : Valeur identifiant le codage des caractères utilisé par le serveur VisualAge Pacbase.

La valeur 1147 correspond à l'EBCDIC français.

- Transaction code : Code de la transaction IMS du Moniteur de Communication VisualAge Pacbase.

L'installation proprement dite peut maintenant démarrer. Pour cela, appuyez sur le bouton [Install].

Mise à jour des paramètres de la communication

Le fichier bases.ini

Vous aurez besoin de mettre à jour le fichier bases.ini pour rajouter ou supprimer une Base, ou pour modifier des paramètres de communication.

Par défaut, le chemin d'accès à ce fichier est :

C:\Program Files\IBM\VisualAge_Pacbase_35\Common\

Le format de ce fichier répond aux normes standard des fichiers .ini de Windows.

Chaque section de ce fichier définit une configuration permettant d'accéder à une Base VisualAge Pacbase. Le nom de cette section est placé entre crochets [Nom de la section].

Le nom de chaque section sera présenté à l'utilisateur lors de sa connexion. Dans la liste ainsi affichée, il choisira sa base de connexion. C'est pourquoi ce nom doit être le plus explicite possible. D'autant plus que vous pouvez définir plusieurs configurations/sections pour une Base, qui doivent donc être facilement identifiables ; vous définissez ainsi plusieurs options de communication pour une même Base.

NOTE : Avec la Station de Travail VA Pac, vous pouvez utiliser un fichier personnalisé des paramètres. Dans ce cas, indiquez, après le nom de la Base, le nom de ce fichier, encadré des signes "<" et ">". Pour plus de détails sur ces paramètres, reportez-vous plus haut, à la fin du sous-chapitre "Station de Travail VisualAge Pacbase".

CONTENU DE CHAQUE SECTION

Les paramètres de chaque section sont listés ci-dessous à raison d'un paramètre par ligne.

- baseCode = code logique de la Base VisualAge Pacbase (obligatoire)
Longueur maximum : 4 caractères

NOTE : Dans le cas de la Station de Travail, ce code logique doit être unique dans le fichier bases.ini pour une localisation donnée.

- signOn = paramètre précisant le mode de contrôle de l'utilisateur à la connexion. Ce paramètre obligatoire prend une des trois valeurs suivantes :
 - VAPac : indique que l'utilisateur n'aura à donner son code et son mot de passe que pour sa connexion à la Base VisualAge Pacbase.

- Middleware : indique que l'utilisateur n'aura à donner son code et son mot de passe que pour sa connexion au système hôte. Le contrôle de la connexion à la Base étant assuré par RACF (ou équivalent).
- VAPac Middleware : indique que l'utilisateur devra donner son code et son mot de passe pour la connexion au système hôte et pour la connexion à la Base (option par défaut).
- communicationAdapter = ce paramètre indique le mode de communication retenu.
 - DIRECT : middleware local
 - GATEWAY : middleware distant (via gateway VisualAge Pacbase)
Selon l'option choisie certains paramètres sont à définir spécifiquement, ils sont détaillés ci-après.
- connectionCleaningInterval = laps de temps (exprimé en ms.) qui sépare deux lancements de la procédure de nettoyage des connexions.
Paramètre utile pour la Station de Travail VisualAge Pacbase uniquement.
Valeur par défaut pour Windows NT/2000/XP = 60000.
Pour Windows 95/98, ce paramètre prend obligatoirement la valeur -1, non modifiable, pour éviter les problèmes de multi-threading, non supporté par ces plateformes.

PARAMETRES POUR ADAPTATEUR DIRECT (MIDDLEWARE LOCAL)

- locationsFile = ce paramètre indique le chemin et le nom du fichier qui contient les définitions des localisations. défaut : ..\common\vaplocat.ini
- location = nom de la localisation pour la Base
Longueur maximum : 20 caractères.
Défaut : Location-1
Plusieurs Bases peuvent pointer sur la même localisation.
RAPPEL : Une localisation identifie le protocole de communication utilisé pour accéder au serveur VisualAge Pacbase et les coordonnées physiques de ce serveur pour ce protocole.
- traceFile = chemin et nom du fichier qui réceptionnera la trace d'exécution du middleware.
Par défaut, ce fichier est automatiquement créé (avec timestamp) dans le sous-répertoire VapTrace.
- traceLevel = niveau de détail de la trace d'exécution du middleware. Les valeurs possibles sont :
 - 0 : pas de trace
 - 1 : trace des erreurs (défaut)
 - 2 : trace standard non détaillée
 - 3 : trace pour informations

- 4 et + : trace de debug
- codePageFile = chemin et nom du fichier contenant les tables de conversion des codes page.
Défaut : ..\middleware\CharConv.txt

PARAMETRES POUR ADAPTATEUR GATEWAY (MIDDLEWARE DISTANT)

- host = nom ou adresse IP du host où la gateway VisualAge Pacbase est installée.
Ne pas saisir de "0" cadrés à gauche, source d'erreur pour certaines configurations.
Défaut : 127.0.0.1 pour host local
- port = valeur du port IP où la gateway réceptionne les requêtes clientes.
Défaut : 5647
- location = nom de la localisation pour la Base
Longueur maximum : 20 caractères.
Défaut : Location-1
Plusieurs Bases peuvent pointer sur la même localisation.
RAPPEL : Une localisation identifie le protocole de communication utilisé pour accéder au serveur VisualAge Pacbase et les coordonnées physiques de ce serveur pour ce protocole.
- Les paramètres traceFile et traceLevel, documentés ci-dessus, peuvent être utilisés lors du lancement de la GATEWAY. Pour plus de détails, voir le Guide d'utilisation du Middleware.

Le fichier vaplocat.ini

Vous aurez besoin de mettre à jour le fichier vaplocat.ini pour rajouter ou supprimer une Base, ou pour éventuellement modifier d'autres paramètres décrits ci-dessous.

Par défaut, le chemin d'accès à ce fichier est :

C:\Program Files\IBM\VisualAge_Pacbase_35\Common\

Pour ajouter une Base VisualAge Pacbase, créez une ligne où vous indiquez le nom de la localisation entre les caractères "<" et ">".

La longueur maximum de ce nom est de 20 caractères.

Les paramètres varient selon le protocole sélectionné, vous saisissez un paramètre par ligne.

- VAP SOCKET

- ```
<NomLocalisation>
COMM_TYPE=SOCKET
MONITOR=BVPSCPI
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
IXO_ADDRESS=127.0.0.1 3676
```
- MVS CICS SOCKET

```
<NomLocalisation>
COMM_TYPE=TCPMVS
MONITOR=BVPSSOC
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1147
IXO_ADDRESS=127.0.0.1 3676
IXO_TRANSID=V303
```
  - MVS CICS CPI-C

```
<NomLocalisation>
COMM_TYPE=CPIC
MONITOR=BVPSCPI
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1147
```
  - MQ-CICS Bridge

```
<NomLocalisation>
MESSAGE_LENGTH=31744
MONITOR=BVPSCPI
COMM_TYPE=MQCICS
IXO_LOCALCCSID=819
IXO_QUEUEMANAGER=EECC
IXO_REQUESTQUEUE=VAP.CICS_BRIDGE.VAPCS.REQUESTER
IXO_REPLYQUEUE=VAP.CICS_BRIDGE.VAPCS.REPLY
IXO_DYNAMICREPLYQUEUE=VAP.CICS_BRIDGE.VAPCS.DYNAMICREPLY
IXO_TRANSID=WK51
IXO_REQUESTEXPIRY=120
IXO_TIMEOUT=35
IXO_HEADERVERSION=1
```
  - IMS Connect

```
<NomLocalisation>
COMM_TYPE=TCPIMS
MONITOR=XXXXXXXXX
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1147
IXO_ADDRESS=127.0.0.1 3676
IXO_TRANSID=WK35
IXO_DATASTORE=IMSC
IXO_RACFGROUP=FR42
```
  - MQ-IMS Bridge

```
<NomLocalisation>
MESSAGE_LENGTH=31744
MONITOR=XXX
```

```

COMM_TYPE=MQIMS
IXO_LOCALCCSID=819
IXO_QUEUEMANAGER=EECC
IXO_REQUESTQUEUE=VAP. IMS_BRIDGE.VAPCS.REQUESTER
IXO_REPLYQUEUE=VAP. IMS_BRIDGE.VAPCS.REPLY
IXO_DYNAMICREPLYQUEUE=VAP. IMS_BRIDGE.VAPCS.DYNAMICREPLY
IXO_TRANSID=WK51
IXO_REQUESTEXPIRY=120
IXO_TIMEOUT=35
IXO_HEADERVERSION=1

```

### Explication des paramètres :

La liste ci-dessous est triée par ordre alphabétique des paramètres.

- **COMM\_TYPE** : Ce paramètre désigne le protocole de communication utilisé.

Les valeurs possibles sont :

- **SOCKET** : Serveur VA Pac sous Windows ou UNIX, utilisant TCP/IP.
- **TCPMVS** : Serveur VA Pac sous MVS/CICS, utilisant un listener TCP/IP.
- **CPIC** : Serveur VA Pac sous MVS/CICS, utilisant le protocole CPI-C.
- **MQCICS** : Serveur VA Pac sous MVS/CICS, utilisant le protocole MQ-CICS-BRIDGE.
- **MQMCICS**: idem MQCICS et si le middleware est installé sur la même machine que MQSeries Server et que l'on souhaite utiliser un lien direct avec MQSeries.
- **TCPIMS** : Serveur VA Pac sous MVS/IMS, utilisant le protocole IMS Connect.
- **MQIMS** : Serveur VA Pac sous MVS/IMS, utilisant le protocole MQ-IMS-BRIDGE.
- **MQMIMS** : idem MQIMS et si le middleware est installé sur la même machine que MQSeries Server et que l'on souhaite utiliser un lien direct avec MQSeries.
- **IXO\_ADDRESS** : Adresse IP et port utilisé par le serveur VA Pac.

Ne pas saisir de "0" cadrés à gauche, source d'erreur pour certaines configurations.

Le numéro de port doit correspondre au numéro de port indiqué lors de la configuration de la machine host pour OS390 CICS ou IMS/VS.

Idem pour Windows ou UNIX, sachant que pour ces plateformes, un numéro identifie spécifiquement une Base VA Pac. Pour plus de détails, voir le chapitre "Installation de la partie Serveur", sous-chapitre "Installation du Référentiel", section "Base de Développement", dans les Guides d'Installation respectifs.



- IXO\_DATASTORE : Nom du lien vers IMS défini dans IMS Connect (IMS DataStore ID).
- IXO\_DYNAMICREPLYQUEUE : Nom dynamique de la Queue.  
MQ-CICS Bridge et MQ-IMS Bridge.  
Lorsque ce paramètre est renseigné, la Queue de réponse est créée dynamiquement par MQSeries pour pouvoir transmettre les messages de réponse (cf. documentation MQSeries pour l'utilisation de ce type de Queue). Ce paramètre doit alors contenir le nom dynamique de la Queue (Dynamic Queue name) et il faut fournir le nom du "Queue Model" dans le paramètre IXO\_REPLYQUEUE. (48 caractères maximum).
- IXO\_HEADERVERSION : Version de la structure MQCIH (pour MQ-CICS Bridge) ou MQIIH (pour MQ-IMS Bridge).  
Les valeurs autorisées sont 1 (défaut) ou 2, en fonction de l'interface Bridge utilisée. (1 caractère maximum).
- IXO\_LOCALCCSID : Code du jeu de caractères de la machine ne local.  
MQ-CICS Bridge et MQ-IMS Bridge.  
Ce code (819 par défaut) est utilisé pour la conversion des messages par MQSeries lors des lectures et écritures des Queues.  
(9 caractères maximum).
- IXO\_QUEUEMANAGER : Nom du Queue Manager local de la partie cliente.  
MQ-CICS Bridge et MQ-IMS Bridge.  
(4 caractères maximum).  
S'il y a un MQSeries Server intermédiaire sur NT, ce paramètre identifie le Queue Manager NT intermédiaire et pas le Queue Manager.
- IXO\_RACFGROUP : Nom du groupe RACF pour IMS Connect.
- IXO\_REPLYQUEUE : Nom de la Queue des messages de réponse.  
MQ-CICS Bridge et MQ-IMS Bridge.  
(48 caractères maximum).
- IXO\_REQUESTEXPIRY : Délai d'expiration du message de requête, exprimé en secondes.  
MQ-CICS Bridge et MQ-IMS Bridge.  
(9 caractères maximum).
- IXO\_REQUESTQUEUE : Nom de la Queue des messages de requêtes.  
MQ-CICS Bridge et MQ-IMS Bridge.  
(48 caractères maximum).
- IXO\_TIMEOUT : Temps maximum pendant lequel une station de travail attend la réponse du serveur avant de signaler une erreur de communication.  
L'unité de ce paramètre est la seconde.

Sa valeur par défaut est 30.

- IXO\_TRANSID : Identifiant de la transaction.

4 caractères maximum pour CICS.

8 caractères maximum pour IMS.

Une transaction IMS devra être déclarée sous IMS par un GEN INPUT IMS de la manière suivante :

```
APPLCTN PSB=BVPSSOC
TRANSACT CODE=WK35,SEGSIZE=32000,MODE=SNGL,SEGNO=00050,
PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
MSGTYPE=(MULTSEG,RESPONSE,$CLS)
```

- HOST\_ENCODING : Identifie le codage des caractères utilisé par le serveur VisualAge Pacbase.

La valeur 1147 correspond à l'EBCDIC français.

Non utilisé pour MQ-CICS Bridge et MQ-IMS Bridge.

- MESSAGE\_LENGTH :

La valeur de ce paramètre est obligatoirement 31744.

- MONITOR :

Code du moniteur de communication pour VisualAge Pacbase, c'est-à-dire BVPSCPI, ou BVPSSOC pour MVS CICS SOCKET.

Non utilisé pour IMS Connect et MQ-IMS Bridge.

**NOTE :** Pour MVS CICS CPI-C, vous pouvez toutefois saisir une autre valeur que BVPSCPI sachant que la valeur du paramètre MONITOR doit dans tous les cas être identique à celle positionnée dans 'Symbolic destination name', paramètre inclus dans la configuration de ce protocole de communication.

Tous les paramètres pouvant être exploités dans le fichier vaplocat.ini ne sont pas explicités ici. En effet, certains de ces paramètres permettent des réglages fins concernant le fonctionnement du middleware, réglages également possibles lors de l'exploitation des proxies (générées par les Outils eBusiness). Dans ce contexte particulier, ces paramètres sont édités via l'outil Editeur de Localisations (Location Editor) et sont donc documentés via son aide en ligne. Vous pouvez également consulter la documentation de ces paramètres dans le Guide d'utilisation du Middleware.

---

## Mise à jour, Ajout, Désinstallation de composants

Le CD des composants Client VA Pac vous propose également les options suivantes :

- "Update" pour mettre à jour un composant déjà installé.
- "Modify" pour ajouter un nouveau sous-composant à un composant déjà installé.

Cette option n'est disponible que pour :

- Administrator & Developer workbench :  
ajout de l'un ou l'autre des workbench quand l'un des deux seulement  
avait été installé,
- Station de Travail VisualAge Pacbase :  
ajout d'une Méthode.
- "Remove" pour désinstaller un composant.

Vous pouvez également utiliser la même fonction proposée par le service Windows "Ajout/suppression de Programmes" dans le Panneau de configuration.



---

## Chapitre 5. Tests

---

### Liste des utilitaires principaux

Le tableau résume l'ensemble des utilitaires de gestion de la Base d'Administration et de la Base de Développement.

Script	Description
ARCH	Archivage du journal de la Base Administration
PACS	Sauvegarde de la Base Administration (SAVE)
REOR	Réorganisation de la Base Administration
REST	Restauration de la Base Administration
PAGX	Extraction Base Administration
UPGP	Mise à jour batch format PAF Base Administration
ARCH	Archivage du journal de la Base Développement
PACS	Sauvegarde de la Base Développement (SAVE)
PACS	Gestion des bibliothèques de la Base Développement (MLIB)
REOR	Réorganisation de la Base Développement
REST	Restauration de la Base Développement
UPDT	Mise à jour batch de la Base Développement
GPRT	Edition génération
PACX	Extraction de bibliothèques (EXLI)
PACX	Extraction du journal (EXPJ)
PACX	Extraction d'entités (EXTR)
PACS	Extraction de sous-réseaux (UXSR)

---

### Tests de l'installation

Ces tests comprennent plusieurs phases :

- Tests de mises à jour TP et batch, d'édition-génération
- Tests des procédures d'administration
- Tests des procédures de développement
- Tests des utilitaires d'extraction.

## Tests de mise à jour TP/batch, Edition-Génération

Ces tests comprennent les étapes suivantes :

- Tests d'utilisation en TP :
  - Démarrer le serveur de la base de test.
  - Dérouler les enchaînements d'écrans.
  - Effectuer des mises à jour.
- Tests de mise à jour batch :
  - Exécuter la procédure UPDT.
- Tests d'édition et de génération de programmes :
  - Exécuter la procédure GPRT.

## Tests des procédures de la Base Administration

En tout premier lieu, faites quelques consultations et mises à jour avec la Station "Administrator workbench".

Procédez ensuite aux tests, sachant que les fichiers de la Base Administration doivent être fermés à l'accès TP.

Ces tests comprennent les étapes listées ci-dessous, à exécuter dans l'ordre suivant :

- Archivage du journal créé lors des tests d'utilisation : exécuter la procédure ARCH donnant un fichier PJ(1).
- Sauvegarde de la Base Administration : exécuter la procédure PACS (option SAVE) donnant un fichier PC(1) ; la clé d'évaluation n'autorisant pas cette procédure, il faudra saisir la clé fournie auparavant.
- Réorganisation de la sauvegarde séquentielle PC(1) de la Base Administration : exécuter la procédure REOR donnant un fichier PC(2).
- Restauration de la Base Administration à partir de l'archive PJ(1) et de la sauvegarde de la base PC(2) : exécuter la procédure REST.

Pour tous ces tests, le serveur de la Base d'Administration doit être arrêté.

- Extraction de données de la Base Administration : exécuter la procédure PAGX et sauvegarder le fichier GY des données extraites.
- Mise à jour PAF de la Base Administration : exécuter la procédure UPGP avec le fichier extrait par l'exécution de la procédure PAGX

## Tests des procédures de la Base de Développement

En tout premier lieu, faites quelques consultations et mises à jour avec la Station "Developer workbench".

Procédez ensuite aux tests, sachant que les fichiers de la Base de Développement doivent être fermés à l'accès TP.

Ces tests comprennent les étapes listées ci-dessous, à exécuter dans l'ordre suivant :

- Archivage du journal créé lors des tests d'utilisation : exécuter la procédure ARCH donnant un fichier PJ(1).
- Sauvegarde directe de la Base de Développement : exécuter la procédure PACS (option SAVE) donnant un fichier PC(1) ; la clé d'évaluation n'autorisant pas cette procédure, il faudra saisir la clé fournie auparavant.
- Gestionnaire de bibliothèque : ajout/suppression de bibliothèque dans la Base de Développement : exécuter la procédure PACS (MLIB option) donnant un fichier PC(2).
- Réorganisation de la sauvegarde séquentielle PC(2) de la Base de Développement : exécuter la procédure REOR donnant un fichier PC(3).
- Restauration de la Base de Développement à partir de l'archive PJ(1) et de la sauvegarde de la base PC(3) : exécuter la procédure REST.

Après restauration et réouverture des fichiers de la Base de Développement, il est recommandé de refaire quelques tests rapides en mode conversationnel afin de s'assurer du bon fonctionnement de l'application.

### **Tests des utilitaires d'extraction**

Ces tests ont pour objet d'exécuter les procédures d'extraction de la base.

Ils comprennent les étapes suivantes, à exécuter dans l'ordre :

- Extraction d'une bibliothèque du réseau sous forme de mouvements : exécuter la procédure PACX option EXLI.
- Extraction d'entités d'une bibliothèque : exécuter la procédure PACX option EXTR.
- Extraction de mouvements sélectionnés et/ou listes de mouvements sélectionnés à partir du journal archivé (PJ) : exécuter la procédure PACX option EXPJ.
- Extraction de sous-réseaux : exécuter la procédure PACS option UXSR.

Pour tous ces tests, les fichiers de la base peuvent être ouverts dans le conversationnel.

Chacun des jobs peut enchaîner sur une mise à jour UPDT ou UPDP permettant de vérifier la validité des mouvements extraits.





---

## Chapitre 6. Réinstallation Serveur

Une réinstallation de la partie système du Serveur VisualAge Pacbase est à effectuer à la suite de la réception d'une sous-version apportant des corrections d'anomalies ou des développements ponctuels sur la version installée.

Le service Windows NT -Ajout/Suppression de programmes- permet d'intervenir avec les options "Mise à jour/Suppression".

---

### Réinstallation

La mise à jour de la partie système est proposée automatiquement, si vous avez une version antérieure déjà installée.

Les scripts de lancement opérationnels, dans les répertoires \DATA\[Nom\_de\_base]\SCRIPT, ne sont pas touchés. Pour avoir la dernière version, lors de la réinstallation, il faut se référer aux modèles de scripts de lancement, dans le répertoire \SYS\Script\_Models.

**ATTENTION :** Il est nécessaire de mettre à jour les Modèles des Bases de Développement et d'Administration par l'exécution de la procédure VINS pour chaque base.

---

### Désinstallation

La désinstallation de la partie système se fait par le service "ajout/suppression de programmes" du panneau de configuration (application 'VisualAge Pacbase Server').

On propose, au même moment, de supprimer de la base de Registre les accès aux données mais de sauvegarder ces données sous un autre répertoire \DATA.

---

### Suppression des données

La suppression d'une base de données est une opération définitive. Il est conseillé de faire une sauvegarde avant.

Les différentes étapes sont des commandes manuelles :

- Supprimer le répertoire des données de la base et tous les sous-répertoires dépendants.

- Supprimer le répertoire:

Winnt\Profiles\AllUsers\Start Menu\Programs\VisualAge  
Pacbase 3.5 Server\[nom\_base]

- Supprimer du registre HK\_Local\_Machine la clé :

Software\IBM\BVP VisualAge Pacbase 3.5\Server\BVP\_DATA\[nom\_base]

Il existe une méthode automatique pour effectuer une suppression complète des informations d'une Base (données, registres, services) : procédure BVPDELDB (cf. "Chapitre 3 - Déréférencer une Base").

---

## Chapitre 7. Reprises et échanges entre bases 2.n et 3.n

---

### Reprises des bases des versions 2.0 et 2.5

#### Préambule

Il est à noter que la reprise d'une base peut être complétée par l'exécution de procédures utilitaires destinées à préparer cette reprise vers la nouvelle version.

Celles-ci sont de deux types :

#### Procédures à exécuter sur la base 2.5.

Leur documentation détaillée figure dans le manuel "Utilitaires 2.5 d'aide à la reprise".

Elles se composent de :

- UTAG : épuration du fichier AG,
- UTFG : typage des formats guide,
- UTSD : association de mot-clé à un type de structure de données.

#### Procédures à exécuter sur la nouvelle base.

Leur documentation détaillée figure dans le chapitre "Annexes" du présent manuel.

Elles se composent de :

- UTU1 : extraction des lignes 'UNS',
- UTU2 : mise à jour des lignes 'UNS'.

#### Remarques

Les fichiers issus de la reprise devront être convertis, si nécessaire, en code page IBM-923 s'ils ne sont pas déjà en code page ISO8859.

L'utilitaire "byptrans" fourni dans un environnement 3.n permet de convertir un fichier d'un code page donné vers le code page IBM-923. Si nécessaire, cet utilitaire devra être appliqué aux fichiers PC, PE, PJ, PP, PG issus de la version 2.5 pour qu'ils soient repris correctement en 3.n.

Voici un exemple :

Fichier "IN", d'une version 2.5, à convertir d'un code page PC850 vers le code page 923, du référentiel 3.n :

```
bvptrans F-IN F-OUT 850 923
```

A noter qu'avant de transférer une base 250, il faut lancer la procédure LVBL suivie d'une REOR sur la plateforme d'origine.

## Opérations à effectuer

L'installation de cette version nécessite, d'une part, la reprise des fichiers AG (commandes édition-génération), AE AP (paramètres utilisateurs) et AB AC (fichiers PEI) dans la nouvelle Base Administration, et d'autre part la reprise de l'ancienne Base de Développement.

### Enchaînement à effectuer :

Il se déroule en sept étapes :

1) Sauvegarde de tous les anciens fichiers nécessaires par exécution des procédures suivantes dans l'ancienne version.

- SAVE : sauvegarde de la Base de Développement (PC),
- PARM : sauvegarde des paramètres utilisateur (PE),
- SVAG : sauvegarde des commandes d'édition-génération (PG),
- SVPE : sauvegarde de l'environnement PEI (PP),
- ARCH : archivage du journal (PJ).

2) Mise en place de la Base Administration.

Cette étape nécessite le déroulement de l'installation de la Base Administration.

Elle permet de créer les fichiers AN, AR, AY, AJ et GU.

Elle consiste en l'exécution des scripts suivants :

- création de la Base Administration proprement dite,
  - INGU : création et initialisation du fichier GU des codes utilisateurs,
  - REST : initialisation de la Base Administration avec les données d'installation,
  - VINS : installation du Modèle administration (voir la description en annexe en fin de manuel),
- Mise en oeuvre de la clé d'accès à partir de la Station Administrator workbench ; elle comporte les opérations suivantes :
  - saisie de la clé d'accès,
  - définition des cibles,

- activation de la clé,
- réorganisation éventuelle de la Base Administration si le script précédent (VINS) le demande dans son compte-rendu d'exécution,
  - PACS (SAVE) : sauvegarde de la Base Administration,
  - REOR : réorganisation de la Base Administration,
  - ARCH : initialisation du fichier journal de la Base Administration,
  - REST : restauration de la Base Administration,
- reprise des données de l'ancienne base,
  - PE25 : reprise des paramètres utilisateurs à partir du fichier PE généré lors de l'étape 1,
  - UTMP: reprise des mots de passe 2.n (facultatif),
  - PACS (SAVE) : sauvegarde de la Base Administration.

### 3) Reprise d'une Base de Développement.

Cette étape nécessite l'installation préalable de la base VA Pacbase correspondante avec les données de test.

Puis consiste en l'exécution des procédures suivantes :

- PC25 : reprise et réorganisation de l'ancienne Base de Développement à partir de la sauvegarde de l'ancienne base créée dans l'étape 1,
- REOR : réorganisation de la nouvelle base,
- ARCH : archivage de la Base de Développement,
- REST : restauration de la nouvelle Base de Développement à partir de la sauvegarde obtenue précédemment ; vous devez penser à indiquer le code de la Base de Développement sur l'entrée utilisateur de la procédure,
- VINS : installation du Modèle développement de la nouvelle base (voir la description en annexe en fin de manuel et créer le fichier MR pour prise en compte de ses mouvements dans la réorganisation),

L'exécution des trois procédures suivantes, si elle n'est pas demandée dans le compte-rendu de VINS, peut être faite dans un souci d'optimisation.

- PACS (SAVE) : sauvegarde de la nouvelle Base de Développement,
- REOR : réorganisation de la nouvelle Base de Développement,
- REST : restauration de la Base de Développement à partir du fichier de sauvegarde obtenu en sortie de la procédure de réorganisation précédente.

Les étapes 4, 5, 6 et 7 sont facultatives.

### 4) Reprise des commandes d'édition-génération.

Elle consiste en l'exécution des procédures suivantes :

- PG20 : Reprise d'un fichier PG 2.0,
- PG25 : Reprise d'un fichier PG 2.5

5) Reprise du journal (PJ).

Elle consiste en l'exécution de la procédure suivante :

- PJ25 : reprise des données du fichier PJ.

6) Reprise des fichiers PEI.

Elle consiste en l'exécution de la procédure suivante :

- PP25 : reprise du fichier PP,

7) Reprise des paramètres de Pac/Transfer (UV).

Elle consiste en l'exécution de la procédure suivante :

- UV25 : reprise des données du fichier UV.

## **Reprise des paramètres utilisateur (PE25)**

### **PE25 - Présentation générale**

#### Principe

Cette procédure (PE25) permet de reprendre le fichier PE, issu de la sauvegarde des paramètres utilisateurs par la procédure PARM, pour les intégrer dans la Base Administration.

#### Condition d'exécution

Les fichiers de la Base Administration doivent être fermés dans le transactionnel.

#### Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

#### Résultat obtenu

La Base Administration intégrant les paramètres utilisateurs issus de la version 2.0 ou 2.5.

#### Remarques

En cas de reprise d'une base sous contrôle RACF, il faudra, lors de la connexion à la Base Administration utiliser un code administrateur de votre base et non le code ADMIN fourni à l'installation.

Il faut également noter que la reprise réinitialise les attributions de licences.

### **PE25 - Entrées / Traitements / Résultats**

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

Une ligne 'A' (facultative) permettant de préciser le code et le nom de l'administrateur. Cette ligne n'est nécessaire que dans le cas de l'utilisation d'un système de sécurité

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'A'	Code ligne
3	8	bbbbbbbb	Code de l'administrateur
11	36		Nom de l'administrateur

Une ligne 'B' par base permettant de préciser les caractéristiques des Bases de Développement qui seront gérées dans la nouvelle Base Administration soit :

- le code de la base : c'est le code logique qui sera indiqué lors de la restauration de la base.
- le nom de la base

Si le code ou le nom de la base n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'B'	Code ligne
3	4	bbbb	Code base logique
7	36		Nom de la base
43	4	cccc	Code de la base (zone interne)

Une ligne 'C' (facultative) permettant de préciser le code de la base rattaché aux lignes de commandes optionnelles. Si la commande est commune à toutes les bases, cette ligne n'est pas à saisir et le code base devient '\*\*\*\*'.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'C'	Code ligne
3	4	bbbb	Code de la base

## PE25 - Description des étapes

Traitement des paramètres utilisateur (PE) : PTU920

Code	Nom physique	Type	Libellé
PAC7EN	Rép. save : OLDPE	Entrée	Paramètres utilisateur ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
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	Rép. tmp : WGY	Sortie	Mouvements paramètres des utilisateurs (longueur=310)
PAC7ET	Rép. user : PE25ET920	Etat	Compte-rendu en cas d'anomalie

Mise en forme des mouvements : PAF900

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



Code	Nom physique	Type	Libellé
PAC7GY	Rép. tmp : WGY ou WGZ	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 Administration : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Sortie	Données de la Base Administration
PAC7AN	Base Admin - Rép. base : AN	Sortie	Index de la Base Administration
PAC7AY	Base Admin - Rép. base : AY	Sortie	Extension de la Base Administration
PAC7AJ	Base Admin - Rép. Journal : AJ	Sortie	Journal de la Base Administration
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
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	NUL	Sortie	Mouvements erronés UPDT (longueur=80)

Code	Nom physique	Type	Libellé
PAC7RY	NUL	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		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

### PE25 - Script d'exécution

```

|-----|
| VISUALAGE PACBASE |
|-----|
| RETRIEVAL OF PE FILE |
|-----|
|
<job id=PE25>

<script language="VBScript">
Dim MyProc
MyProc = "PE25"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1029"))
'-----
Call StateList (base, statusL)

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

' No Rollback
WshEnv("BVPRB") = "N"

Call Msg_Log (Array("1022" , "PTU920"))

```

```

'-----
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU920","PAC7EN",Rep_SAVE & "\OLDPE")
Call BvpEnv("PTU920","PAC7ET",Rep_USR & "\PE25ET920.txt")
Call BvpEnv("PTU920","PAC7GY",Rep_TMP & "\WGY.tmp")
Call RunCmdLog ("BVPTU920")
Call Err_Cod(Return , 0 , "PTU920")

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.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_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PE25IEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PE25IFA15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
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>

```

## Reprise de la Base de Développement (PC25)

### PC25 - Présentation générale

#### Principe

La procédure (PC25) permet de reprendre le fichier PC issu de la sauvegarde de l'ancienne Base de Développement en un fichier PC nouveau format.

#### Condition d'exécution

Aucune.

#### Edition obtenue

Cette procédure édite un compte-rendu donnant le nombre de manuels transformés en rapports, le code de la nouvelle Base de Développement et le nombre d'enregistrements en sortie du fichier PC.

#### Résultat obtenu

Le résultat de cette procédure est une image séquentielle de la Base de Développement au nouveau format. Ce nouveau fichier PC doit servir d'entrée à l'étape obligatoire suivante : la réorganisation.

### PC25 - Remarques sur les données reprises

#### Eclatement du descriptif commentaire (-G)

Le descriptif commentaire est éclaté en plusieurs descriptifs.

- Commentaires proprement dits  
Ils regroupent les commentaires et les alias COBOL (-GC).

Attention :

Si en version 2.0 ou 2.5, le type de la ligne de documentation n'est pas adapté à l'entité (ex : ligne de génération sur une rubrique) il deviendra un commentaire.

- Lignes de génération  
Elles regroupent les types G, P, V et Z (-GG).
- Paramètres de génération  
On y retrouve le type de ligne O (-GO).
- Gestion des libellés d'erreur  
Ils regroupent les types C, D, F, S, T, U (-GE).
- Appels d'entités par relations  
On y retrouve le type de ligne R (-CR).
- Cas particulier du format guide  
Le type présent sur le descriptif du format guide détermine la valeur du type de la fiche de définition, soit 'C' pour les commentaires, 'G' pour les compléments de génération ou 'O' pour les options de générations. Les appels de formats guides pourront donc être visualisables en -GC, -GG ou -GO.

**ATTENTION :** Si en version 2.0 ou 2.5 sur un même descriptif, le type a plusieurs valeurs, un message d'erreur est édité : une intervention manuelle est demandée pour corriger l'incohérence.

Là encore si l'appel de format guide est 'Généré' ou 'Option de dialogue' à tort, l'appel devient un commentaire.

Remarque importante :

Si en version 2.n , une ligne de type autre que commentaire est écrasée par une ligne de commentaire, du fait de l'éclatement des -G, cet écrasement ne pourra se retrouver dans la version actuelle, il faudra intervenir manuellement dans la nouvelle base.

### Type de table des structures de données

Les structures de données de type table (G, T, M, N) et vue logique (V) conservent leur type. Tous les autres types (fichiers...) deviennent Z. L'état n'est plus supporté par la structure de donnée, le type J disparaît.

### Transformation des manuels U

Les manuels sont transformés en rapports, leurs codes sont complétés par 'EIBM'.

Données extension : entités utilisateurs, formats guides, maquettes des états.

Les enregistrements suite de ces entités disparaissent, il n'y a plus qu'un index par enregistrement principal et un par enregistrement suite. Des données extension sont créées qui concatènent les informations présentes sur les anciens enregistrements. Ces données longues peuvent avoir une longueur de 1000 et sont réparties sur plusieurs enregistrements. Un seul index est désormais créé qui pointe sur le premier de ces enregistrements.

### **PC25 - Entrées / Traitements / Résultats**

Une ligne \* avec le code de la nouvelle Base de Développement.

Cette ligne est facultative si le code base attribué en version 2.5 convient. Ce code base doit avoir été défini dans la Base Administration.

Si aucun code base n'est renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'*'	Code ligne
3	4	bbbb	Code nouvelle base

### **PC25 - Description des étapes**

Traitements généraux : PTU911

Code	Nom physique	Type	Libellé
PAC7MC	Rép. save : OLDPC	Entrée	Image séquentielle du réseau ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
PAC7PB	Rép. tmp : WPC	Sortie	Premier enregistrement des données (longueur=153)
PAC7PE	Rép. tmp : WPE	Sortie	Fiches des O.E.U. (version 2.5), maquettes d'états et des commentaires (excepté les appels de formats guides) (longueur=193)

Code	Nom physique	Type	Libellé
PAC7PG	Rép. tmp : WPG	Sortie	Descriptions des formats guides et des commentaires avec appels de formats guides (longueur=193)
PAC7PL	Rép. tmp : WPL	Sortie	Fiches des rapports, de leurs descriptions et des manuels (longueur=193)
PAC7PZ	Rép. tmp : WPZ	Sortie	Entités utilisateurs et descriptifs de leurs occurrences (version 2.5.) (longueur=193)
PAC7PF	Rép. tmp : WPF	Sortie	Enregistrements autres (longueur=153)
PAC7PM	Rép. tmp : WPM	Sortie	Fichier compte-rendu (longueur=62)
PAC7ET	Rép. user : PC25ET911	Etat	Compte-rendu en cas d'absence du code base

#### Traitement des manuels et des rapports : PTU909

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7PB	Rép. tmp : WPC	Entrée	Premier enregistrement des données
PAC7PL	Rép. tmp : WPL	Entrée	Fiches des rapports, de leurs descriptions et des manuels
PAC7PI	Rép. tmp : WPI	Sortie	Fiches des rapports et de leurs descriptions triés et reformatés (longueur=153)
PAC7PM	Rép. tmp : WPM	Entrée/Sortie	Fichier compte-rendu

#### Traitement des commentaires PTU92A

Code	Nom physique	Type	Libellé
PAC7PG	Rép. tmp : WPG	Entrée	Descriptions des formats guides et de leurs appels dans les commentaires
PAC7PM	Rép. tmp : WPM	Entrée/Sortie	Compte-rendu

Code	Nom physique	Type	Libellé
PAC7PE	Rép. tmp : WPH	Sortie	Descriptions des formats guides et de leurs appels dans les commentaires (longueur=193)

#### Tri des Formats guides : PTU92B

Code	Nom physique	Type	Libellé
PAC7PE	Rép. tmp : WPH	Entrée	Descriptions des formats guides et de leurs appels dans les commentaires
PAC7PK	Rép. tmp : WPK	Sortie	Descriptions des formats guides et de leurs appels dans les commentaires (longueur=193)

#### Tri des Formats guides : PTU92C

Code	Nom physique	Type	Libellé
PAC7PK	Rép. tmp : WPK	Entrée	Descriptions des formats guides et de leurs appels dans les commentaires
PAC7KP	Rép. tmp : WKP	Sortie	Descriptions des formats guides et de leurs appels dans les commentaires (longueur=193)
PAC7PB	Rép. tmp : WPC	Entrée	Premier enregistrement des données
PAC7PD	Rép. tmp : WPD	Sortie	Premier enregistrement des données (longueur=153)

#### Traitement des maquettes d'états : PTU919

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7PE	Rép. tmp : WPE	Entrée	Fiches des O.E.U (version 2.5), des maquettes d'état et des commentaires (excepté les appels de formats guides)
PAC7PB	Rép. tmp : WPD	Entrée	Premier enregistrement des données
PAC7PM	Rép. tmp : WPM	Entrée/Sortie	Fichier compte-rendu



Code	Nom physique	Type	Libellé
PAC7ZP	Rép. tmp : WEP	Sortie	Fiches des O.E.U (version 2.5), des maquettes d'états, (longueur=193)
PAC7PO	Rép. tmp : WPO	Sortie	Commentaires (excepté les appels de formats guides) (longueur=153)
PAC7KP	Rép. tmp : WKP	Entrée	Commentaires (avec appels de formats guides)
PAC7PD	Rép. tmp : WPB	Sortie	Premier enregistrement des données (longueur=153)

#### Traitement des méta entités : PTU912

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7PZ	Rép. tmp : WPZ	Entrée	Entités utilisateurs (version 2.5)
PAC7PB	Rép. tmp : WPC	Entrée	Premier enregistrement des données
PAC7ZP	Rép. tmp : WZP	Sortie	Enregistrements du Modèle développement (fiches et descriptions) (longueur=193)

#### Traitement des entités utilisateurs : PTU913

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7PX	Rép. tmp : WEP	Entrée	Fiches des O.E.U (version 2.5) et des maquettes d'états
PAC7PZ	Rép. tmp : WZP	Entrée	Modèle de développement (fiches et descriptifs) et descriptions des O.E.U (version 2.5)
PAC7PB	Rép. tmp : WPB	Entrée	Premier enregistrement des données
PAC7ZP	Rép. tmp : WZX	Sortie	Données extension du modèle développement et des entités utilisateurs, des maquettes d'états et des commentaires (avec appels de formats guides) (longueur=193)

Code	Nom physique	Type	Libellé
PAC7PD	Rép. tmp : WPR	Sortie	Premier enregistrement des données (longueur=153)

#### Tri des données longues : PTU91A

Code	Nom physique	Type	Libellé
PAC7PZ	Rép. tmp : WZX	Entrée	Données extension intermédiaires
PAC7ZP	Rép. tmp : WXZ	Sortie	Données longues triées (longueur=193)

#### Fusion des fichiers : PTU914

Cette étape reconstitue l'image séquentielle finale à partir des fichiers intermédiaires issus des étapes précédentes.

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Fichier des libellés d'erreur
PAC7ZP	Rép. tmp : WXZ	Entrée	Données extension triées
PAC7PO	Rép. tmp : WPO	Entrée	Commentaires (sans appel de formats guides)
PAC7PD	Rép. tmp : WPR	Entrée	Premier enregistrement des données
PAC7PI	Rép. tmp : WPI	Entrée	Fiches des rapports et de leurs descriptions
PAC7PF	Rép. tmp : WPF	Entrée	Enregistrements autres
PAC7PM	Rép. tmp : WPM	Entrée	Fichier compte-rendu
PAC7PC	Rép. save : PC-new	Sortie	Image séquentielle du réseau (version actuelle)
PAC7ET	Rép. user : PC25ET914	Etat	Compte-rendu de la reprise

#### **PC25 - Script d'exécution**

```

|-----|
| VISUALAGE PACBASE |
|-----|
| RETRIEVAL OF PC FILE |
|-----|

```

```

<job id=PC25>

<script language="VBScript">
Dim MyProc
MyProc = "PC25"
</script>

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

<script language="VBScript">

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

If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
End If

Call Msg_Log (Array("1022" , "PTU911"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU911","PAC7MC",Rep_SAVE & "\OLDPC")
Call BvpEnv("PTU911","PAC7PB",Rep_TMP & "\WPC.tmp")
Call BvpEnv("PTU911","PAC7PE",Rep_TMP & "\WPE.tmp")
Call BvpEnv("PTU911","PAC7PF",Rep_TMP & "\WPF.tmp")
Call BvpEnv("PTU911","PAC7PG",Rep_TMP & "\WPG.tmp")
Call BvpEnv("PTU911","PAC7PL",Rep_TMP & "\WPL.tmp")
Call BvpEnv("PTU911","PAC7PZ",Rep_TMP & "\WPZ.tmp")
Call BvpEnv("PTU911","PAC7PM",Rep_TMP & "\WPM.tmp")
Call BvpEnv("PTU911","PAC7ET",Rep_USR & "\PC25ET911.txt")
Call RunCmdLog ("BVPTU911")
Call Err_Cod(Return , 0 , "PTU911")

Call Msg_Log (Array("1022" , "PTU909"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU909","PAC7PB",Rep_TMP & "\WPC.tmp")
Call BvpEnv("PTU909","PAC7PL",Rep_TMP & "\WPL.tmp")
Call BvpEnv("PTU909","PAC7PI",Rep_TMP & "\WPI.tmp")
Call BvpEnv("PTU909","PAC7PM",Rep_TMP & "\WPM.tmp")
Call RunCmdLog ("BVPTU909")
Call Err_Cod(Return , 0 , "PTU909")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WPL.tmp")

Call Msg_Log (Array("1022" , "PTU92A"))
'-----
Call BvpEnv("PTU92A","PAC7PG",Rep_TMP & "\WPG.tmp")
Call BvpEnv("PTU92A","PAC7PE",Rep_TMP & "\WPH.tmp")

```

```

Call BvpEnv("PTU92A","PAC7PM",Rep_TMP & "\\WPM.tmp")
Call RunCmdLog ("BVPTU92A")
Call Err_Cod(Return , 0 , "PTU92A")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPG.tmp")

Call Msg_Log (Array("1022" , "PTU92B"))
'-----
Call BvpEnv("PTU92B","PAC7PE",Rep_TMP & "\\WPH.tmp")
Call BvpEnv("PTU92B","PAC7PK",Rep_TMP & "\\WPK.tmp")
Call RunCmdLog ("BVPTU92B")
Call Err_Cod(Return , 0 , "PTU92B")

Call Msg_Log (Array("1022" , "PTU92C"))
'-----
Call BvpEnv("PTU92C","PAC7PB",Rep_TMP & "\\WPC.tmp")
Call BvpEnv("PTU92C","PAC7PK",Rep_TMP & "\\WPK.tmp")
Call BvpEnv("PTU92C","PAC7PD",Rep_TMP & "\\WPD.tmp")
Call BvpEnv("PTU92C","PAC7KP",Rep_TMP & "\\WKP.tmp")
Call RunCmdLog ("BVPTU92C")
Call Err_Cod(Return , 0 , "PTU92C")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPK.tmp")

Call Msg_Log (Array("1022" , "PTU919"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
Call BvpEnv("PTU919","PAC7PB",Rep_TMP & "\\WPD.tmp")
Call BvpEnv("PTU919","PAC7PE",Rep_TMP & "\\WPE.tmp")
Call BvpEnv("PTU919","PAC7PM",Rep_TMP & "\\WPM.tmp")
Call BvpEnv("PTU919","PAC7PD",Rep_TMP & "\\WPB.tmp")
Call BvpEnv("PTU919","PAC7PO",Rep_TMP & "\\WPO.tmp")
Call BvpEnv("PTU919","PAC7ZP",Rep_TMP & "\\WEP.tmp")
Call BvpEnv("PTU919","PAC7KP",Rep_TMP & "\\WKP.tmp")
Call RunCmdLog ("BVPTU919")
Call Err_Cod(Return , 0 , "PTU919")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPD.tmp")
Call DelFile (Rep_TMP & "\\WPE.tmp")
Call DelFile (Rep_TMP & "\\WPH.tmp")

Call Msg_Log (Array("1022" , "PTU912"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
Call BvpEnv("PTU912","PAC7PZ",Rep_TMP & "\\WPZ.tmp")
Call BvpEnv("PTU912","PAC7ZP",Rep_TMP & "\\WZP.tmp")
Call BvpEnv("PTU912","PAC7PB",Rep_TMP & "\\WPC.tmp")

```

```

Call RunCmdLog ("BVPTU912")
Call Err_Cod(Return , 0 , "PTU912")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPZ.tmp")
Call DelFile (Rep_TMP & "\\WPC.tmp")

Call Msg_Log (Array("1022" , "PTU913"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
Call BvpEnv("PTU913","PAC7PB",Rep_TMP & "\\WPB.tmp")
Call BvpEnv("PTU913","PAC7PD",Rep_TMP & "\\WPR.tmp")
Call BvpEnv("PTU913","PAC7PX",Rep_TMP & "\\WEP.tmp")
Call BvpEnv("PTU913","PAC7PZ",Rep_TMP & "\\WZP.tmp")
Call BvpEnv("PTU913","PAC7ZP",Rep_TMP & "\\WZX.tmp")
Call RunCmdLog ("BVPTU913")
Call Err_Cod(Return , 0 , "PTU913")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WKP.tmp")
Call DelFile (Rep_TMP & "\\WPB.tmp")
Call DelFile (Rep_TMP & "\\WEP.tmp")
Call DelFile (Rep_TMP & "\\WZP.tmp")

Call Msg_Log (Array("1022" , "PTU91A"))
'-----
Call BvpEnv("PTU91A","PAC7PZ",Rep_TMP & "\\WZX.tmp")
Call BvpEnv("PTU91A","PAC7ZP",Rep_TMP & "\\WXZ.tmp")
Call RunCmdLog ("BVPTU91A")
Call Err_Cod(Return , 0 , "PTU91A")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WZX.tmp")

Call Msg_Log (Array("1022" , "PTU914"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
WshEnv("PAC7PC") = BVP_SvName & "-new"
Call BvpEnv("PTU914","PAC7PF",Rep_TMP & "\\WPF.tmp")
Call BvpEnv("PTU914","PAC7PI",Rep_TMP & "\\WPI.tmp")
Call BvpEnv("PTU914","PAC7PD",Rep_TMP & "\\WPR.tmp")
Call BvpEnv("PTU914","PAC7PM",Rep_TMP & "\\WPM.tmp")
Call BvpEnv("PTU914","PAC7PO",Rep_TMP & "\\WPO.tmp")
Call BvpEnv("PTU914","PAC7ZP",Rep_TMP & "\\WXZ.tmp")
Call BvpEnv("PTU914","PAC7ET",Rep_USR & "\\PC25ET914.txt")
Call RunCmdLog ("BVPTU914")
Call Err_Cod(Return , 0 , "PTU914")

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

```

```

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(BVP_SvName & "")

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

</script>
</job>

```

## Reprise commandes d'édition-génération (PG20)

### PG20 - Présentation générale

#### Principe

La procédure (PG20) permet de reprendre le fichier PG image séquentielle des commandes d'édition-génération de la version 2.0 au nouveau format.

Elle met à jour la Base de Développement avec les commandes d'édition-génération, et la Base Administration avec les lignes de Script (présentes sur l'écran GP visualisé en option C4 version 2.0).

#### Condition d'exécution

Les fichiers de la Base Administration et de la Base de Développement doivent être fermés dans le transactionnel.

#### Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

#### Remarques

La mise à jour rejettera tout mouvement à insérer dans une bibliothèque ou une session inexistante dans la nouvelle base. Le fichier PG peut contenir des commandes spécifiques à une bibliothèque ou une session purgée ultérieurement.

La mise à jour d'une commande d'édition-génération pour une entité ne peut s'effectuer si l'entité n'existe pas. Exemple : pour la commande GCP PROGRA, le programme PROGRA devra être présent dans la base.

Les codes utilisateurs présents dans le fichier PG et absents dans la Base Administration sont automatiquement créés pour les utilisateurs ayant des Scripts.

## PG20 - Entrées / Traitements / Résultats

Une ligne \* avec code utilisateur, mot de passe et code de la Base de Développement concernée par les lignes de Script mises à jour dans la Base Administration.

Si le code utilisateur ou le code de la base n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	mot de passe
22	4	cccc	Code base

## PG20 - Description des étapes

Formatage des commandes d'édition génération : PTU908

Code	Nom physique	Type	Libellé
PAC7IN	Rép. save : PG20	Entrée	Commandes édition-génération ancienne version
PAC7OU	Rép. save : OLDPG	Sortie	Commandes édition-génération reformatées (longueur=150)

Traitement des commandes d'édition génération : PTU921

Code	Nom physique	Type	Libellé
PAC7PG	Rép. save : OLDPG	Entrée	Commandes édition-génération ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements des commandes d'édition-génération (longueur=310)
PAC7GZ	Rép. tmp : WGZ	Sortie	Mouvements des lignes de Script (longueur=310)

Code	Nom physique	Type	Libellé
PAC7ET	Rép. user : PG20ET921	Etat	Compte-rendu d'anomalies

#### Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PAC7AN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
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	Rép. tmp : WGY ou WGZ	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 Administration : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Sortie	Données de la Base Administration
PAC7AN	Base Admin - Rép. base : AN	Sortie	Index de la Base Administration



Code	Nom physique	Type	Libellé
PAC7AY	Base Admin - Rép. base : AY	Sortie	Extension de la Base Administration
PAC7AJ	Base Admin - Rép. Journal : AJ	Sortie	Journal de la Base Administration
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
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	NUL	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	NUL	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		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

Mise en forme des mouvements : PAF900

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

Code	Nom physique	Type	Libellé
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	Rép. tmp : WGY	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

Code	Nom physique	Type	Libellé
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
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	NUL	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	NUL	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		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

### PG20 - Script d'exécution

```

| -----
| VISUALAGE PACBASE
|
| -----
| RETRIEVAL OF PG FILE SINCE 2.0
|
| -----
|
<job id=PG20>

<script language="VBScript">
Dim MyProc
MyProc = "PG20"
</script>

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

```

```

<script language="VBScript">

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

Call Msg_Log (Array("1029"))
'-----
Call StateList (base, statusL)

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

'-PG- Retrieval for the Aministration database and
'the Development database
'THUS the same steps are executed is this procedure :
' = PTUBAS + ... PACA15 ... x 2 : concurrent access on AR
'For ADMIN <Semlock> and <Semadmin>' are set
' to Administration database
'For XXXX (database typed) <Semlock> is set
' to Development database
' and <Semadmin> is set to Administration database

' No Rollback
WshEnv("BVPRB") = "N"

Call Msg_Log (Array("1022" , "PTU908"))
'-----
Call BvpEnv("PTU908","PAC7IN",Rep_SAVE & "\PG20")
Call BvpEnv("PTU908","PAC7OU",Rep_SAVE & "\OLDPG")
Call RunCmdLog ("BVPTU908")
Call Err_Cod(Return , 0 , "PTU908")

Call Msg_Log (Array("1022" , "PTU921"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU921","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PTU921","PAC7GZ",Rep_TMP & "\WGZ.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU921","PAC7PG",Rep_SAVE & "\OLDPG")
Call BvpEnv("PTU921","PAC7ET",Rep_USR & "\PG20ET921.txt")
Call RunCmdLog ("BVPTU921")
Call Err_Cod(Return , 0 , "PTU921")

'for the ADMIN base(<Z> space) :

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGZ.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")

```

```

Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Call RunCmdLog ("BVP AF900")

If Return <> 0 then
Call Msg_Log (Array("1025", "PAF900/ADMinistration", Return))
End If

If Return = 0 then

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7IE") = Rep_AUSR & "\PG20IE1A15.txt"
WshEnv("PAC7IF") = Rep_AUSR & "\PG20IF1A15.txt"
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
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
If Return > 4 Then
Call Msg_Log (Array("1025", "PACA15/ADMinistration", Return))
End If
End If

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WGZ.tmp")
Call DelFile (Rep_TMP & "\WME.tmp")
Call DelFile (Rep_TMP & "\WMV.tmp")
Call DelFile (Rep_TMP & "\WMW.tmp")
Call DelFile (Rep_TMP & "\WMX.tmp")
Call DelFile (Rep_TMP & "\WMY.tmp")
Return = 0
' To continue

```

'For the DEV base(<Y> space) :

```
Call Msg_Log (Array("1022" , "PAF900"))
'-----
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","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.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("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PG20IE2A15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PG20IF2A15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
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>
```

## **Reprise commandes d'édition-génération (PG25)**

### **PG25 - Présentation générale**

#### Principe

La procédure PG25 permet de reprendre le fichier PG image séquentielle des commandes d'édition-génération de la version 2.5 au nouveau format.

Elle met à jour la Base de Développement avec les commandes d'édition-génération, et la Base Administration avec les lignes de Script (présentes sur l'écran GP visualisé en option C4 version 2.5).

#### Condition d'exécution

Les fichiers de la Base Administration et de la Base de Développement doivent être fermés dans le transactionnel.

#### Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

#### Remarques

La mise à jour rejettera tout mouvement à insérer dans une bibliothèque ou une session inexistante dans la nouvelle base. Le fichier PG peut contenir des commandes spécifiques à une bibliothèque ou une session purgée ultérieurement.

La mise à jour d'une commande d'édition-génération pour une entité ne peut s'effectuer si l'entité n'existe pas. Exemple : pour la commande GCP PROGRA, le programme PROGRA devra être présent dans la base.

Tout utilisateur ayant des lignes de Scripts pour ces générations en mode TP (écran GP visualisé en option C4) est automatiquement créé dans la Base Administration.

## PG25 - Entrées / Traitements / Résultats

Une ligne \* avec code utilisateur, mot de passe et code de la Base de Développement concernée par les lignes de Script mises à jour dans la Base Administration.

Si le code utilisateur ou le code de la base n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	mot de passe
22	4	cccc	Code base

## PG25 - Description des étapes

Traitement des commandes d'édition génération : PTU921

Code	Nom physique	Type	Libellé
PAC7PG	Rép. save : OLDPG	Entrée	Commandes édition-génération ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements des commandes d'édition-génération (longueur=310)
PAC7GZ	Rép. tmp : WGZ	Sortie	Mouvements des lignes de Script (longueur=310)
PAC7ET	Rép. user : PG25ET921	Etat	Compte-rendu d'anomalies

Mise en forme des mouvements : PAF900

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



Code	Nom physique	Type	Libellé
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	Rép. tmp : WGY ou WGZ	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 Administration : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Sortie	Données de la Base Administration
PAC7AN	Base Admin - Rép. base : AN	Sortie	Index de la Base Administration
PAC7AY	Base Admin - Rép. base : AY	Sortie	Extension de la Base Administration
PAC7AJ	Base Admin - Rép. Journal : AJ	Sortie	Journal de la Base Administration
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

Code	Nom physique	Type	Libellé
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	NUL	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	NUL	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		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

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	Rép. tmp : WGY	Entrée	Mouvements de mise à jour

Code	Nom physique	Type	Libellé
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
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	NUL	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	NUL	Sortie	Mouvements erronés UPDP (longueur=310)

Code	Nom physique	Type	Libellé
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		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

### PG25 - Script d'exécution

```

| -----
| VISUALAGE PACBASE
| -----
| RETRIEVAL OF PG FILE SINCE 2.5
| -----
|
<job id=PG25>

<script language="VBScript">
Dim MyProc
MyProc = "PG25"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1029"))
'-----
Call StateList (base, statusL)

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

'-PG- Retrieval for the Aministration database
'and the Development database
'THUS the same steps are EXECUTED is this procedure :
' = PTUBAS + ... PACA15 ... x 2 : concurrent access on AR
'For ADMIN <Semlock> and <Semadmin>' are set to
' Administration database
'For XXXX (database typed) <Semlock> is set to

```

```

'Development database
' and <Semadmin> is set to Administration database

' No Rollback
WshEnv("BVPRB") = "N"

Call Msg_Log (Array("1022" , "PTU921"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU921","PAC7GY",Rep_TMP & "\WGZ.tmp")
Call BvpEnv("PTU921","PAC7GZ",Rep_TMP & "\WGZ.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU921","PAC7PG",Rep_SAVE & "\OLDPG")
Call BvpEnv("PTU921","PAC7ET",Rep_USR & "\PG25ET921.txt")
Call RunCmdLog ("BVPTU921")
Call Err_Cod(Return , 0 , "PTU921")

'for the ADMIN base(<Z> space) :

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGZ.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Call RunCmdLog ("BVPAF900")

If Return <> 0 then
Call Msg_Log (Array("1025", "PAF900/ADMinistration", Return))
End If

If Return = 0 then
Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PG25IEADM.txt")

```

```

Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PG25IFADM.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
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
If Return > 4 Then
Call Msg_Log (Array("1025", "PACA15/ADMinistration", Return))
End If
End If

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WGZ.tmp")
Call DelFile (Rep_TMP & "\WME.tmp")
Call DelFile (Rep_TMP & "\WMV.tmp")
Call DelFile (Rep_TMP & "\WMW.tmp")
Call DelFile (Rep_TMP & "\WMX.tmp")
Call DelFile (Rep_TMP & "\WMY.tmp")
Return = 0
' To continue

'For the DEV base(<Y> space) :

Call Msg_Log (Array("1022" , "PAF900"))
'-----
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","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.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("PACGGU") = Rep_ABASE & "\\GU"
WshEnv("PACGGY") = Rep_ABASE & "\\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\\PG25IEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\\PG25IFA15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\\WMV.tmp")
WshEnv("PAC7RB") = Rep_TMP & "\\WRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\\WRY.tmp"
'PAC7RY not used, on default
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>

```

## Reprise mouvements PJ (PJ25)

### PJ25 - Présentation générale

#### Principe

Cette procédure (PJ25) permet de reprendre le fichier PJ, fichier journal (fichier séquentiel des mouvements), en un nouveau fichier d'archive au format de la nouvelle version.

#### Condition d'exécution

Aucune.

#### Edition obtenue

Cette procédure édite un compte-rendu indiquant le nombre de mouvements lus, en précisant d'une part le nombre de mouvements repris tels quels, c'est à dire dans leur format 2.n, et le nombre de mouvements transformés en mouvements de la nouvelle version. Le nombre d'enregistrements écrits est également édité.

**Attention :** : Le nombre d'enregistrements écrits peut être très supérieur au nombre d'enregistrements lus. A partir d'un ancien mouvement 2.n, plusieurs mouvements de la nouvelle version peuvent être créés, notamment pour les méta entités et les entités utilisateurs.

Certains mouvements d'appels de format guide peuvent être incomplets si leur contenu est réparti sur plusieurs mouvements et qu'un seul de ces mouvements apparait dans le journal. Un display est alors édité.

### Résultat obtenu

Un fichier journal PJ au format de la nouvelle version.

### Remarques

Cette opération de transformation du journal est facultative. Elle doit être exécutée si des procédures batch le nécessitent (Utilisation de Pac/Transfer).

Cette reprise s'effectue exclusivement des versions 2.0 ou 2.5 vers la nouvelle version.

La reprise de certains mouvements nécessite la recherche d'informations dans la base de la nouvelle version. Il est fort possible que les données correspondantes n'existent plus dans la nouvelle base (session ou bibliothèque supprimée par exemple). Dans ce cas, l'ancien mouvement sera repris dans son format 2.n.

## **PJ25 - Description des étapes**

Traitement du fichier séquentiel des mouvements : PTU918

Code	Nom physique	Type	Libellé
PAC7PJ	Rép. save : OLDPJ	Entrée	Fichier journal ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
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 extensions de la Base de Développement
PAC7JP	Rép. save : PJ	Sortie	Fichier journal au nouveau format (longueur=170)
PAC7ET	Rép. user : PJ25ET918	Etat	Compte-rendu

## PJ25 - Script d'exécution

```

| -----
| VISUALAGE PACBASE
|
| -----
| RETRIEVAL OF PJ FILE
|
| -----
|
<job id=PJ25>

<script language="VBScript">
Dim MyProc
MyProc = "PJ25"
</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" , "PTU918"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
Call BvpEnv("PTU918","PAC7PJ",Rep_SAVE & "\OLDPJ")
Call BvpEnv("PTU918","PAC7ET",Rep_USR & "\PJ25ET918.txt")
Call BvpEnv("PTU918","PAC7JP",Rep_SAVE & "\PJ")
Call RunCmdLog ("BVPTU918")
Call Err_Cod(Return , 0 , "PTU918")

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

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

```

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

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

## Reprise du fichier PP (PP25)

### PP25 - Présentation générale

#### Principe

Cette procédure permet de reprendre le fichier PP image séquentielle de l'environnement de production (PEI) de la version 2.0 ou 2.5 et met à jour la Base de Développement de la version installée.

#### Condition d'exécution

Les fichiers de la Base de Développement doivent être fermés dans le transactionnel.

#### Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

#### Résultat obtenu

Cette procédure génère un fichier mouvement contenant les environnements de production existants, la liste des entités générées, les environnements par défaut (-GO de la Bibliothèque), la liste des sessions de production au nouveau format, et met à jour la Base de Développement de la nouvelle version installée.

#### Remarques

Tout mouvement de mise à jour dans une Bibliothèque ou une Session inexistante dans la nouvelle Base sera rejeté.

Le fichier PP peut contenir des environnements avec des codes bibliothèques ou des sessions non encore créées ou purgées ultérieurement dans la Base 2.0 ou 2.5.

Lors de la création des environnements dans la session la plus ancienne (0001 ou session d'initialisation de la bibliothèque), une application par défaut sera créée automatiquement dans le même contexte.

## PP25 - Entrées / Traitements / Résultats

Une ligne \* avec Code utilisateur et Mot de passe

Si le code utilisateur n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	mot de passe
19	3	'***'	option de mise à jour en inter-
			bibliothèque des environnements

## PP25 - Description des étapes

Traitement de l'environnement de production : PTU923

Code	Nom physique	Type	Libellé
PAC7PP	Rép. save : OLDPP	Entrée	Sauvegarde de l'environnement de production ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
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 extensions de la Base de Développement
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements des environnements de production (longueur=310)
PAC7ET	Rép. user : PP25ET923	Etat	Compte-rendu en cas d'anomalie

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	Rép. tmp : WGY	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

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
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	NUL	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	NUL	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		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

### PP25 - Script d'exécution

```

| -----
| VISUALAGE PACBASE
| -----
| RETRIEVAL OF PP FILE
| -----
|
<job id=PP25>

<script language="VBScript">
Dim MyProc
MyProc = "PP25"
</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" , "PTU923"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
Call BvpEnv("PTU923","PAC7GY",Rep_TMP & "\WGY.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU923","PAC7PP",Rep_SAVE & "\OLDPP")
Call BvpEnv("PTU923","PAC7ET",Rep_USR & "\PP25ET923.txt")
Call RunCmdLog ("BVPTU923")
Call Err_Cod(Return , 0 , "PTU923")

Call Msg_Log (Array("1022" , "PAF900"))
'-----
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","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.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("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PP25IE1A15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PP25IF1A15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"

```

```

'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
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>

```

## Reprise mots de passe (UTMP)

### UTMP - Présentation générale

#### Principe

Cette procédure est un utilitaire qui permet de reprendre les mots de passe des utilisateurs 2.N à partir du fichier PE issu de la sauvegarde des paramètres utilisateurs par la procédure PARM, pour les intégrer dans le fichier des utilisateurs de la Base Administration.

Le rafraîchissement des mots de passe est fonction de l' option choisie sur la carte utilisateur.

#### Condition d'exécution

Le fichier des utilisateurs de la Base Administration doit être fermé dans le transactionnel.

#### Edition obtenue

En cas de création à tort, la procédure édite un message d' anomalie sous forme de displays.

#### Résultat obtenu





## Reprise paramètres Pac/Transfer (UV25)

### UV25 - Présentation générale

#### Principe

La procédure (UV25) permet de reprendre le fichier UV des paramètres de Pac/Transfer de la version 2.0 ou 2.5 au nouveau format.

Elle met à jour la Base Administration.

#### Condition d'exécution

Les fichiers de la Base Administration doivent être fermés dans le transactionnel.

#### Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

### UV25 - Entrées / Traitements / Résultats

Une ligne \* avec code utilisateur, mot de passe et code de la Base de Développement concernée par les transferts.

Si le code utilisateur ou le code de la base n'est pas renseigné, un message d'anomalie est émis et la procédure ne peut s'effectuer.

La structure de la ligne est la suivante :

Position	Longueur	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	mot de passe
22	4	cccc	Code base

### UV25 - Description des étapes

#### Traitement des paramètres de transfert : PTU922

Code	Nom physique	Type	Libellé
PAC7UV	Rép. base : OLDUV	Entrée	Paramètres de transfert de l'ancienne version
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur

Code	Nom physique	Type	Libellé
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements des paramètres de transfert (longueur=310)
PAC7ET	Rép. user : UV25ET922	Etat	Compte-rendu en cas d'anomalie

Mise en forme des mouvements : PAF900

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PAC7AN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
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	Rép. tmp : WGY ou WGZ	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 Administration : PACA15

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Sortie	Données de la Base Administration
PAC7AN	Base Admin - Rép. base : AN	Sortie	Index de la Base Administration
PAC7AY	Base Admin - Rép. base : AY	Sortie	Extension de la Base Administration
PAC7AJ	Base Admin - Rép. Journal : AJ	Sortie	Journal de la Base Administration
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	NUL	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	NUL	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	NUL	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		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

## UV25 - Script d'exécution

```
|-----|
| VISUALAGE PACBASE |
|-----|
| RETRIEVAL OF UV FILE |
|-----|
|
|<job id=UV25>
|
|<script language="VBScript">
|Dim MyProc
|MyProc = "UV25"
|</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" , "PTU922"))
|'-----|
|WshEnv("PAC7AE") = Rep_SKEL & "\AE"
|Call BvpEnv("PTU922", "PAC7ET", Rep_USR & "\UV25ET922.txt")
|Call BvpEnv("PTU922", "PAC7GY", Rep_TMP & "\WGY.tmp")
|WshEnv("PAC7MB") = Fic_Input
|Call BvpEnv("PTU922", "PAC7UV", Rep_BASE & "\OLDUV")
|Call RunCmdLog ("BVPTU922")
|Call Err_Cod(Return , 0 , "PTU922")
|
|Call Msg_Log (Array("1022" , "PAF900"))
|'-----|
|WshEnv("PAC7AE") = Rep_SKEL & "\AE"
|WshEnv("PAC7AN") = Rep_ABASE & "\AN"
|WshEnv("PAC7AR") = Rep_ABASE & "\AR"
|WshEnv("PACGGN") = Rep_ABASE & "\AN"
|WshEnv("PACGGR") = Rep_ABASE & "\AR"
|WshEnv("PACGGU") = Rep_ABASE & "\GU"
|Call BvpEnv("PAF900", "PAC7GY", Rep_TMP & "\WGY.tmp")
|Call BvpEnv("PAF900", "PAC7ME", Rep_TMP & "\WME.tmp")
|Call BvpEnv("PAF900", "PAC7MV", Rep_TMP & "\WMV.tmp")
|Call BvpEnv("PAF900", "PAC7MW", Rep_TMP & "\WMW.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_AJOURNAL & "\AJ"
|WshEnv("PAC7AN") = Rep_ABASE & "\AN"
|WshEnv("PAC7AR") = Rep_ABASE & "\AR"
```

```

WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15", "PAC7IE", Rep_USR & "\UV25IEA15.txt")
Call BvpEnv("PACA15", "PAC7IF", Rep_USR & "\UV25IFA15.txt")
Call BvpEnv("PACA15", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15", "PAC7MV", Rep_TMP & "\WMV.tmp")
WshEnv("PAC7DC") = Rep_TMP & "\NULDC.tmp"
'PAC7DC not used, on default
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
Call Err_Cod(Return , 0 , "PACA15")

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

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

</script>
</job>

```

## Echange mouvements MB entre base 2.n et 3.n (MB25)

### MB25 - Présentation générale

#### Principe

Cette procédure permet de reprendre des mouvements de format UPDT de la version 2.0 ou 2.5 vers la nouvelle version.

#### Limites

Les Entités utilisateur, Occurrences d'Entités utilisateur, Relations, Manuels de type 'U' ne sont pas traités.

Les appels de Format-guides sont traités comme des commentaires.

Les types de Format-guides changent entre la 2.n et la 3.n : pour permettre l'échange de mouvements entre ces versions, il faut exécuter en 2.n la procédure UTFG qui attribue les types de la 3.n.

## Condition d'exécution

Aucune.

## Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

## Résultat obtenu

Cette procédure génère un fichier mouvement à destination de la procédure UPDT de la nouvelle version ainsi qu'un fichier des mouvements rejetés.

## **MB25 - Description des étapes**

Reprise du fichier MB : PTU926

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
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
PAC7MB	Rép. user : OLDMB	Entrée	Mouvements MB 2.5
PAC7MV	Rép. user : MB25MV	Sortie	Mouvements de reprise pour UPDT
PAC7ME	Rép. user : MB25ME	Sortie	Mouvements rejetés
PAC7EF	Rép. user : MB25EF926	Etat	Compte-rendu de reprise
PAC7DD	Rép. user : MB25DD926	Etat	Contrôle autorisation

## **MB25 - Script d'exécution**

```
|-----
| VISUALAGE PACBASE
|-----
| RETRIEVAL OF MB FILE
|-----
|
<job id=MB25>
```

```

<script language="VBScript">
Dim MyProc
MyProc = "MB25"
</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" , "PTU926"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU926","PAC7MB",RepT_USR & "\OLDMB")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU926","PAC7EF",Rep_USR & "\MB25EF926.txt")
Call BvpEnv("PTU926","PAC7DD",Rep_USR & "\MB25DD926.txt")
Call BvpEnv("PTU926","PAC7ME",Rep_USR & "\MB25ME.txt")
Call BvpEnv("PTU926","PAC7MV",Rep_USR & "\MB25MV.txt")
Call RunCmdLog ("BVPTU926")
Call Err_Cod(Return , 0 , "PTU926")

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

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

</script>
</job>

```

## Echange mouvements GY entre base 2.n et 3.n (GY25)

### GY25 - Présentation générale

#### Principe

Cette procédure permet de reprendre des mouvements de format UPDP de la version 2.0 ou 2.5 vers la nouvelle version.

#### Limites

Les Entités utilisateur, Relations, Manuels de type 'U' ne sont pas traités.

Les appels de Format-guides sont traités comme des commentaires.

Les types de Format-guides changent entre la 2.n et la 3.n : pour permettre l'échange de mouvements entre ces versions, il faut exécuter en 2.n la procédure UTFG qui attribue les types de la 3.n.

#### Condition d'exécution

Aucune.

#### Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

#### Résultat obtenu

Cette procédure génère un fichier mouvement à destination de la procédure UPDP de la nouvelle version ainsi qu'un fichier des mouvements rejetés.

### **GY25 - Description des étapes**

Reprise du fichier GY : PTU927

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
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	Rép. user : OLDGY	Entrée	Mouvements GY 2.5
PAC7MV	Rép. user : GY25MV	Sortie	Mouvements de reprise pour UPDP
PAC7ME	Rép. user : GY25ME	Sortie	Mouvements rejetés
PAC7EF	Rép. user : GY25EF927	Etat	Compte-rendu de reprise
PAC7DD	Rép. user : GY25DD927	Etat	Contrôle autorisation



## GY25 - Script d'exécution

```
| -----
| VISUALAGE PACBASE
| -----
| RETRIEVAL OF GY FILE
| -----
|
<job id=GY25>

<script language="VBScript">
Dim MyProc
MyProc = "GY25"
</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" , "PTU927"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU927","PAC7GY",RepT_USR & "\OLDGY")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU927","PAC7EF",Rep_USR & "\GY25EF927.txt")
Call BvpEnv("PTU927","PAC7DD",Rep_USR & "\GY25DD927.txt")
Call BvpEnv("PTU927","PAC7ME",Rep_USR & "\GY25ME.txt")
Call BvpEnv("PTU927","PAC7MV",Rep_USR & "\GY25MV.txt")
Call RunCmdLog ("BVPTU927")

Call Err_Cod(Return , 0 , "PTU927")

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

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

</script>
</job>
```

## Echange mouvements MB entre base 3.n et 2.n (MB30)

### MB30 - Présentation générale

#### Principe

Cette procédure permet de reprendre des mouvements de format UPDT de la nouvelle version vers la version 2.0 ou 2.5.

#### Limites

Les Méta entités, Entités utilisateur, Relations, Descriptions, Mots-clés, Documentation associée et Appels de Relation ne sont pas traités.

#### Condition d'exécution

Aucune.

#### Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

#### Résultat obtenu

Cette procédure génère un fichier mouvement à destination de la procédure UPDT de la version 2.0 ou 2.5 ainsi qu'un fichier des mouvements rejetés.

### MB30 - Description des étapes

#### Reprise du fichier MB : PTU928

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
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
PAC7MB	Rép. user : MB30	Entrée	Mouvements MB de la nouvelle version

Code	Nom physique	Type	Libellé
PAC7MV	Rép. user : MB30MV	Sortie	Mouvements de reprise pour UPDT
PAC7ME	Rép. user : MB30ME	Sortie	Mouvements rejetés
PAC7EF	Rép. user : MB30EF928	Etat	Compte-rendu de reprise
PAC7DD	Rép. user : MB30DD928	Etat	Contrôle autorisation

### MB30 - Script d'exécution

```

| -----
| VISUALAGE PACBASE
|
| -----
| RETRIEVAL OF MB FILE
|
| -----
|
<job id=MB30>

<script language="VBScript">
Dim MyProc
MyProc = "MB30"
</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" , "PTU928"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU928","PAC7MB",RepT_USR & "\MB30.txt")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU928","PAC7EF",Rep_USR & "\MB30EF928.txt")
Call BvpEnv("PTU928","PAC7DD",Rep_USR & "\MB30DD928.txt")
Call BvpEnv("PTU928","PAC7ME",Rep_USR & "\MB30ME.txt")
Call BvpEnv("PTU928","PAC7MV",Rep_USR & "\MB30MV.txt")
Call RunCmdLog ("BVPTU928")
Call Err_Cod(Return , 0 , "PTU928")

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

Call Msg_Log (Array("1023"))

```

```
'-----
Wscript.Quit (Return)

</script>
</job>
```

## Echange mouvements GY entre base 3.n et 2.n (GY30)

### GY30 - Présentation générale

#### Principe

Cette procédure permet de reprendre des mouvements de format UPDP de la nouvelle version vers la version 2.0 ou 2.5.

#### Limites

Les Méta entités, Relations, Descriptions, Mots-clés, Documentation associée et Appels de Relation ne sont pas traités. En revanche, les Entités utilisateur sont reprises sous forme d'Occurrences d'Entités utilisateur 2.n, à charge pour l'utilisateur de créer l'Entité utilisateur 2.n correspondante.

#### Condition d'exécution

Aucune.

#### Edition obtenue

Cette procédure édite un compte-rendu signalant les anomalies rencontrées.

#### Résultat obtenu

Cette procédure génère un fichier mouvement à destination de la procédure UPDP de la version 2.0 ou 2.5 ainsi qu'un fichier des mouvements rejetés.

### GY30 - Description des étapes

#### Reprise du fichier GY : PTU929

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
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement

Code	Nom physique	Type	Libellé
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	Rép. user : GY30	Entrée	Mouvements GY de la nouvelle version
PAC7MV	Rép. user : GY30MV	Sortie	Mouvements de reprise pour UPDP
PAC7ME	Rép. user : GY30ME	Sortie	Mouvements rejetés
PAC7EF	Rép. user : GY30EF929	Etat	Compte-rendu de reprise
PAC7DD	Rép. user : GY30DD929	Etat	Contrôle autorisation

### GY30 - Script d'exécution

```

| -----
| VISUALAGE PACBASE
|
| -----
| RETRIEVAL OF GY FILE
|
| -----
|
<job id=GY30>

<script language="VBScript">
Dim MyProc
MyProc = "GY30"
</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" , "PTU929"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU929","PAC7GY",RepT_USR & "\GY30.txt")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU929","PAC7EF",Rep_USR & "\GY30EF929.txt")

```

```

Call BvpEnv("PTU929","PAC7DD",Rep_USR & "\GY30DD929.txt")
Call BvpEnv("PTU929","PAC7ME",Rep_USR & "\GY30ME.txt")
Call BvpEnv("PTU929","PAC7MV",Rep_USR & "\GY30MV.txt")
Call RunCmdLog ("BVPTU929")

Call Err_Cod(Return , 0 , "PTU929")

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

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

</script>
</job>

```

## Procédures - Tableau récapitulatif des changements

Liste des nouvelles procédures, depuis la 2.5

Procédure	Remarques
ARCH	Archivage de la Base Administration
ARPM	Archivage du journal QJ du module SCM
INAE	Initialisation du fichier des libellés d'erreur (AE)
INGU	Initialisation du fichier des codes utilisateur (GU)
GPMC	Génération (MOVE CORRESPONDING)
GPRC	Génération (API COBOL)
REOR	Réorganisation de la Base Administration
REST	Restauration de la Base Administration
PACS	Sauvegarde de la Base Administration (option SAVE)
PAGX	Extraction de la Base Administration
UKD1	Maj Batch des clés d'accès et des droits
VINS	Mise à jour du Modèle Administration
PACS	Gestion de la Base de Développement
GY25	Reprise d'un fichier GY pour UPDP vers 3.n
MB25	Reprise d'un fichier MB pour UPDT vers 3.n
GY30	Reprise d'un fichier GY pour UPDP vers 2.n
MB30	Reprise d'un fichier MB pour UPDT vers 2.n
PC25	Reprise de la Base de Développement

Procédure	Remarques
PE25	Reprise des paramètres utilisateur (PE)
PG20	Reprise des commandes édition-génération 2.0 (PG)
PG25	Reprise des commandes édition-génération 2.5 (PG)
PJ25	Reprise du journal 2.n
PP25	Reprise du fichier PP
UTMP	Reprise des mots de passe
UV25	Reprise des paramètres Pac/Transfer (UV)
UTM1	Migration des méta entités (étape 1)
UTM2	Migration des méta entités (étape 2)
UTU1	Extraction lignes 'P' avec 'UNS'
UTU2	Mise à jour lignes 'P' avec 'UNS'
STAT	Statistiques sur la base
VINS	Mise à jour du modèle Développement
CHPM	Compte-rendu contrôle environnement et éléments de la base
CPPM	Comparaison base avec configuration utilisateur
EXPM	Extraction des environnements
GPPM	Ecriture compte-rendu génération dans QJ
HIPM	Mise en production des éléments
PUPM	Purge entités SCM
SIPM	Simulation de génération
UPPM	Mise à jour des éléments
TRED	Pac/Transfer : édition paramètres
UPGP	Mise à jour PAF Base Administration

#### Liste des procédures supprimées depuis la release 2.5

Procédure	Programmes	Remarques
CPSN	PTU850 PTU855	Intégrée dans procédure PACX
EMSN	PTU810	
MESN	PTU815	
SASN	PTU130 PTU140	Intégrée dans procédure PACS
MLIB	PTU100 PTU120	Intégrée dans procédure PACS
SAVE	PTU500	Intégrée dans procédure PACS
CRYP	PACU99	

Procédure	Programmes	Remarques
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	
SVAG	PTU550	
GET0	PACTI1	
GET1	PACT41	
GET2	PACT41 PACT51	
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	
SVPE	PACR60	
RVDE	PREI00 PRE986	
RVKE	PREI40 PREI50	
STOP	PTUR00	
TRRT	REUV802 PTUG90	
VDWN	PVA100 PVA110	
VPUR	PVA400	
VPU1	PVA300 PVA305 PVA310	
VPU2	PVA320	
LVBL	PTULVB	
QREO	PTUN00 PTUN10 PTUN40	
RPPG	PTU908	
RPTD	PTAR20	

---

## Reprises des bases de la version 3.0

### Opérations à effectuer

La release 3.5 peut être installée dans le même environnement que la release 3.0. Ce cas est considéré comme une réinstallation.



Avant toute opération, il est conseillé de sauvegarder les Bases de Développement et d'Administration (SAVE, ARCH), en version 3.0.

- Si cette version est installée dans le même environnement que la 3.0, se reporter au chapitre 'Réinstallation Serveur' du présent manuel.  
Après avoir exécuté la mise à jour du Modèle administration (VINS), il faut se connecter à la station administrateur pour renseigner la nouvelle clé d'accès au système.
- Si cette version est installée dans un nouvel environnement, après avoir suivi toutes les étapes de l'installation de la partie serveur, il faut restaurer les bases dans l'ordre suivant :
  - REST : restauration de la Base d'Administration,
  - VINS : mise à jour du Modèle d'administration,
  - Connexion par la station Administration workbench pour y saisir et activer la nouvelle clé d'accès au système,
  - REST : restauration de chaque Base de Développement,
  - VINS : mise à jour du Modèle développement pour chaque Base de Développement.
- Dans tous les cas, une réorganisation des bases est obligatoire (ARCH, SAVE, REOR, REST).



---

## Chapitre 8. Utilitaires RPP

---

### MIAM - Aide à la migration

#### MIAM - Présentation générale

##### Principe

Cette procédure propose les fonctionnalités suivantes :

- détection des caractères spéciaux : alimentation du fichier des caractères spéciaux CS.
- détection des entités dont le code correspond à une chaîne de caractères interdite (ex: non compatible avec Windows) : alimentation du fichier des caractères spéciaux CS.
- détection des Segments renommés en 00 dans les -CD lorsque le Segment 00 existe au sein du réseau.
- détection des Etats avec des libellés en double.
- détection des structures appelées dans des catégories d'Etat et non définies dans la description de l'Etat.
- détection des lignes de structure d'Etat appelant plusieurs Rubriques avec le même positionnement.
- détection des libellés appelés dans des catégories d'Etat et non définis dans la description de l'Etat.
- détection dans les -GE : références à des Segments ou Rubriques inexistantes dans les -CE ou -CS des Ecrans, ou références à un Texte inexistant.
- détection, dans les -CS des Ecrans, des utilisations multiples d'un même segment en affichage ou en réception au sein d'une même catégorie.
- détection des lignes de working de Macros déclarant une table dont le nom contient le caractère \$.
- détection des lignes de working à ne pas reprendre (de type M, C, L ou A).
- détection des variantes de génération obsolètes.
- détection des Ecrans et Programmes générés avec un nom externe en double.
- détection des utilisations de Méthode dans des entités standards.
- détection des lignes de fonction de type EL dont le niveau est incompatible avec une fonction de type CO DU DW les précédant.
- détection des conditionnements orphelins : lignes de conditionnement de type AN OR DU DW DC DI DV provenant de Macro et précédées par une ligne de conditionnement de type IF n'appartenant pas à la Macro.

- détection des lignes P de type N en double.
- détection des lignes P appartenant à une Macro-Structure avec un niveau de fonction inférieur à 99 associé à un code opérateur différent de 'N' et '\*'.
- détection des entités P à typer en Macro : alimentation du fichier PM.
- détection des Programmes non encore pris en compte (Variante C ou Nature D, F, S).
- détection des Programmes issus de Reverse.
- détection, dans les Ecrans, des lignes P de type \*C, \*A, \*P et \*R qui sont dans des fonctions autres que la fonction F80.
- détection, dans les Ecrans ou dans les Macro-structures, des lignes P de type \*C présentes en fonction F80.
- détection, dans les Ecrans ou les Programmes, des lignes P de types DI, DV et DC.
- détection des entités non reprises : alimentation du fichier NR à destination de la procédure MIBR.

Ceci concerne :

- les Rubriques dont les usages ne sont pas reconnus dans RPP (différents de 0 1 2 3 5 6 7 8 9 C D F G H I J N O P Q R T U W X Y Z) ; les Rubriques d'usage 4 de type Large Object (type L) font exception et sont reconnues,
- les Blocs Bases de Données de type AR, D1, D3, QA, QD, QG, QI, QO, QV, SE, TI, TR,
- les Etats pour maquettes,
- les Méta-Entités de type vrac et les Entités Utilisateur associées,
- les Méta-Entités Pacdesign et les Entités Utilisateur associées ainsi que les rubriques définissant les Méta-Entités 7M 7N 5Q si l'on n'a pas forcé la reprise de ces dernières avec le paramètre MT dans le fichier BVPAMIAM,
- les Dialogues ou Ecrans de type F (Serveur de Dossier) ou FV (Vue de Dossier),
- les Dialogues Ecrans ou Programmes contenant des ordres PAF (opérateur EXP) si le paramètre PF avec la valeur NO a été saisi dans le fichier paramètre BVPAMIAM,
- les Programmes de nature obsolète C.

Pour chaque cas rencontré il sera écrit un enregistrement dans le fichier RQ des anomalies.

Description du fichier CS des caractères ou chaînes de caractères spéciaux détectés :

Pos.	Lon.	Valeur	Signification
1	2		Type de caractère
		WN	Caractères interdits (ex: incompatibles avec Windows)
		CS	Caractère spécial
3	4		Caractère ou chaîne de caractères détecté
9	6		Caractère ou chaîne de caractères de remplacement : ce champ devra être saisi avant le passage des procédures MIBA ou MIBR
13	66		FILLER

Description du fichier NR des entités non reprises :

Pos.	Lon.	Signification
1	3	Code Bibliothèque
4	3	Type PAF
7	30	Code entité
37	2	Type de Bloc
39	1	Usage Rubrique
40	6	Méta-Entité
46	35	FILLER

Description du fichier PM des macros non typées :

Pos.	Lon.	Signification
1	6	Code entité
7	44	FILLER

Description du fichier RQ des anomalies :

Pos.	Lon.	Valeur	Signification
1	3		Bibliothèque
4	3		Type PAF
7	6		Code entité
13	24		FILLER
37	3		Type de l'anomalie
		BL	Bloc à ne pas reprendre

Pos.	Lon.	Valeur	Signification
		CS	Caractère spécial
		DI1	Dialogue de type F (Serveur de Dossier) ou FV (Vue de Dossier)
		EC2	Ecran de type F (Serveur de Dossier) ou FV (Vue de Dossier)
		EC3	Ecran contenant dans les -CS des utilisations multiples d'un même Segment en affichage ou en réception au sein d'une même Catégorie
		ET1	Etat avec libellé en double
		ET3	Etat pour maquette GDP
		ET4	Libellé inexistant appelé dans une catégorie de l'Etat
		ET5	Structure inexistante appelée dans une catégorie de l'Etat
		ET6	Structure appelant plusieurs rubriques avec le même positionnement
		EU1	EU Pacdesign non reprise
		EU2	EU de type vrac non reprise
		GE1	Référence inexistante (-GE)
		MCN	Macro non typée
		MP1	Ligne P de type *C *A *P *R pas en F80
		MP2	Ligne P de type *C présente en F80
		MP3	Ligne P de type DI DV DC
		MP4	Fonction de type EL précédée d'une fonction de type CO ou DU ou DW de niveau incompatible
		MT1	Méta-Entité Pacdesign non reprise
		MT2	Méta-Entité de type vrac non reprise
		MT3	Rubrique de Méta-Entité non reprise
		NEX	Doublons sur nom d'entité générée
		NV1	Niveau de fonction incompatible avec l'opérateur
		OR1	Conditionnements orphelins
		PA1	Présence opérateur PAF
		RU1	Utilisation Méthode dans entité standard
		PG1	Ligne P de Type N en double
		PG2	Programme de Nature obsolète
		PG3	Programme avec Variante ou Nature non encore prise en compte

Pos.	Lon.	Valeur	Signification
		PG4	Programme Reverse non pris en compte
		SG1	Renommage Segment dans -CD
		UR	Usage de Rubrique à ne pas reprendre
		VAR	Variante de génération obsolète
		WK	Ligne de working de type M C L ou A
		WK1	Ligne de working Macro déclarant une table dont le nom contient un \$
		WN	Caractères interdits (ex: incompatibles avec Windows)
40	40		Libellé anomalie
80	2		Section
82	2		Paragraphe
84	3		Numéro de ligne
87	2		Code fichier
89	4		Code Segment
93	2		Numéro de libellé
95	6		Code manuel
101	6		Code Rubrique
107	3		Type PAF
110	6		Code entité
116	2		Code d'appel d'Entité Utilisateur
118	33		FILLER

Conséquences et actions à mener pour les différents types d'anomalies :

Code	Signification	Conséquence	Action possible	Conséquence si aucune action
CS	Caractères spéciaux	Alimentation du fichier CS	(1)	Rejet import TA2. Problème de génération si entité utilisée
WN	Caractères interdits (ex incompatibles avec Windows)	Alimentation du fichier CS	(1)	Rejet import TA2. Problème de génération si entité utilisée
MCN	Macro non typée	Alimentation du fichier PM	(2)	

<b>Code</b>	<b>Signification</b>	<b>Conséquence</b>	<b>Action possible</b>	<b>Conséquence si aucune action</b>
VAR	Variante de génération obsolète	Anomalie dans le fichier RQ	Alimenter le fichier NR	Rejet de génération
EC3	Utilisation multiple d'un même segment en affichage ou réception au sein d'une même catégorie dans les -CS d'un écran	Anomalie dans le fichier RQ	Corriger dans VA Pacbase	Erreur à la génération
ET1	Etat avec libellé en double	Anomalie dans le fichier RQ	Corriger dans VA Pacbase	Warning dans RPP
ET4	Libellé inexistant appelé dans une catégorie de l'Etat	Anomalie dans le fichier RQ	Corriger dans VA Pacbase : supprimer l'appel du Libellé	Différence de génération dans l'aide à la migration
ET5	Structure inexistante appelée dans une catégorie de l'Etat	Anomalie dans le fichier RQ	Corriger dans VA Pacbase : supprimer l'appel de la Structure	Différence de génération dans l'aide à la migration
ET6	Structure appelant plusieurs Rubriques avec le même positionnement	Anomalie dans le fichier RQ	Corriger dans VA Pacbase : corriger la Structure	Différence de génération dans l'aide à la migration
GE1	Référence inexistante (-GE)	Anomalie dans le fichier RQ	Corriger dans VA Pacbase : modifier le -CE ou le -CS ou l'appel de référence du Texte	Erreur signalée dans l'import TA2 (transformation en commentaire)
MP4	Fonction de type EL précédée d'une fonction de type CO ou DU ou DW de niveau incompatible	Anomalie dans le fichier RQ	Corriger dans VA Pacbase	Différences à la génération



Code	Signification	Conséquence	Action possible	Conséquence si aucune action
NEX	Doublon sur le nom d'entité générée dans une même Bibliothèque	Anomalie dans le fichier RQ	Corriger dans VA Pacbase : changer les noms externes ou positionner la variante N sur les programmes pour lesquels on veut conserver le modèle ou alimenter le fichier NR	Si volonté de conserver plusieurs entités générant le même nom externe, il est nécessaire de migrer manuellement
NV1	Niveau de fonction incompatible avec l'opérateur	Anomalie dans le fichier RQ	Corriger dans VA Pacbase : modifier le niveau et/ou l'opérateur	Possible problème de génération
OR1	Conditionnement orphelin	Anomalie dans le fichier RQ	Corriger dans VA Pacbase	Possible problème de génération dans les cas complexes
RU1	Utilisation de Méthode dans Entité standard	Anomalie dans le fichier RQ		
PG1	Ligne P de Type N en double	Anomalie dans le fichier RQ	Corriger dans VA Pacbase	Possible problème de génération
PG3	Programme avec variante ou nature non encore prise en compte	Anomalie dans le fichier RQ	Attendre évolutions	Ne peuvent être migrés aujourd'hui
PG4	Programme Reverse non pris en compte	Anomalie dans le fichier RQ	Attendre évolutions	Ne peuvent être migrés aujourd'hui
SG1	Renommage Segment en 00 dans un -CD	Anomalie dans le fichier RQ		Différences de génération dans l'aide à la migration
WK	Ligne working de Type M C L ou A	Anomalie dans le fichier RQ		Non prise en compte
WK1	Ligne working avec un nom de table contenant un \$	Anomalie dans le fichier RQ	Corriger manuellement dans RPP soit dans le fichier MIMA soit après le dispatch	Indice de la table erroné

Code	Signification	Conséquence	Action possible	Conséquence si aucune action
BL	Bloc base de Données non repris	Alimentation du fichier NR	(3)	
DI1	Dialogue de Type F (Serveur de Dossier ou FV (Vue Dossier)	Alimentation du fichier NR	(3)	
EC2	Ecran de Type F (Serveur de Dossier ou FV (Vue Dossier)	Alimentation du fichier NR	(3)	
ET3	Etat pour maquette GDP	Alimentation du fichier NR	(3)	
EU1	EU Pacdesign non repris	Alimentation du fichier NR	(3) Attention il s'agit d'une option ; il faut modifier les entrées de la procédure si reprise souhaitée	
EU2	EU de Type vrac non reprise	Alimentation du fichier NR	(3)	
MT1	Méta-entité Pacdesign non reprise	Alimentation du fichier NR	(3) Attention il s'agit d'une option ; il faut modifier les entrées de la procédure si reprise souhaitée	
MT2	Méta-entité de Type vrac non reprise	Alimentation du fichier NR	(3)	
MT3	Rubrique de Méta-entité non reprise	Alimentation du fichier NR sauf si rubrique des Méta-entités 7N 7M 5Q si forçage avec le paramètre MT dans le fichier BVPAMIAM	(3)	
PA1	Présence d'opérateur PAF	Alimentation du fichier NR	(3)	

Code	Signification	Conséquence	Action possible	Conséquence si aucune action
PG2	Programme de nature obsolète	Alimentation du fichier NR	(3)	
UR	Usage de Rubrique à ne pas reprendre	Alimentation du fichier NR	(3)	

(1) Le fichier CS des caractères spéciaux permet de remplacer des caractères ou chaînes de caractères par des valeurs considérées valides.

Ce fichier est utilisé par les procédures MIBA, MIBR, MIBJ et MIMA.

En l'absence de valeur de remplacement les valeurs détectées sont remises à blanc.

(2) Le fichier PM des programmes utilisés en tant que macro-structure, mais non typés macro-structure permet aux procédures MIBA, MIBR et MIBJ de les traiter en tant que macro-structure.

(3) Le fichier NR contient les entités qui ne seront pas reconduites à l'issue des traitements.

Ceci concerne les procédures MIBR et MIBJ.

Ce fichier peut être édité pour compléter avec des entités que l'utilisateur sait obsolètes.

## MIAM - Entrées Utilisateur

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3		Code bibliothèque d'extraction
		***	Toutes les bibliothèques
		bbb	Bibliothèque bbb
22	4	nnnn	Numéro de session blanc = session courante
26	1		Type de session
		'T'	Si sélection session historisée
		' '	Si session courante ou H

Ce traitement s'appuie sur le fichier paramètre BVPAMIAM, fourni à l'installation, qui a la structure suivante :

Pos.	Lon.	Valeur	Signification
1	2	'BL'	Type de bloc non repris
		'CS'	Caractère spécial
		'MT'	Forçage reprise de Méta-Entité (1)
		'PF'	Indique si l'on souhaite reprendre ou non les programmes ou écrans contenant des opérateurs PAF ; par défaut il y a reprise ; si l'on ne souhaite pas de reprise ajouter une ligne de type PF avec la valeur NO en position 3
		'UR'	Usage de rubrique reconnu
		'VO'	Variante de génération Ecran autorisées (2)
		'VP'	Variante de génération Programme autorisées (2)
		'VS'	Variante de génération Pac C/S autorisées (2)
		'WK'	Type de ligne de working à ne pas reprendre
		'WN'	Chaîne de caractères interdite (ex: incompatible avec windows)
3	78		Valeur du paramètre

Ce fichier contient les paramètres standard fournis à l'installation afin de permettre à la procédure d'effectuer les contrôles nécessaires.

Il pourra être enrichi avec les caractères spéciaux souhaités : pour cela il suffit d'ajouter une ligne de type CS avec le caractère voulu.

(1) Le paramètre MT permet de forcer la reprise des Méta-Entités de code 7N (Endevor), 7M (Environnement) ou 5Q (Qualité) ainsi que les Méta-Entités Pacdesign (dans ce cas indiquer la valeur DS). Seules ces quatre valeurs sur deux caractères seront prises en compte.

(2) Par défaut, les valeurs des variantes de génération autorisées sont :

VO : 'NX013O456M7U8FIKQZ'

VP : 'N013458FIKOUXQZC'

VS : 'NX03457OU8FIRQ'.

Elles pourront être remplacées par les valeurs saisies au niveau de chaque paramètre dans lequel il faudra indiquer la totalité des variantes que l'on souhaite prendre en compte.

En cas de saisie de plusieurs lignes d'un même type seule la dernière sera prise en compte.

Remarques :

Le fichier paramètre ne nécessite pas d'être trié et toute valeur non reconnue sera ignorée.

Pour les plateformes Windows, ce fichier est livré dans le répertoire SYS/SKEL.

Il est possible de retirer des paramètres.

## **MIAM - Description des étapes**

### Aide à la migration : BVPLTPAM

Code	Nom physique	Type	Libellé
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
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
PAC7CK	Rép. skel : BVPAMIAM	Entrée	Paramètres d'analyse
PAC7CS	Rép. user : MIAMCSPE	Sortie	Caractères spéciaux
PAC7NR	Rép. user : MIAMNR	Sortie	Entités non reprises
PAC7PM	Rép. user : MIAMPAC	Sortie	Programmes à typer en Macro
PAC7RQ	Rép. user : MIAMRQER	Sortie	Anomalies détectées
PAC7DD	Rép. user : MIAMDDPAM	Sortie	Etat d'anomalies

## MIAM - Script d'exécution

```
'-----
' VISUALAGE PACBASE
'-----
' MIGRATION HELP
'-----
'
<job id=MIAM>

<script language="VBScript">
Dim MyProc
MyProc = "MIAM"
</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" , "PLTPAM"))
'-----
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("PAC7MB") = Fic_Input
WshEnv("PAC7CK") = BVP_MiamCk
Call BvpEnv("PLTPAM", "PAC7CS", Rep_USR & "\MIAMCSPE.txt")
Call BvpEnv("PLTPAM", "PAC7NR", Rep_USR & "\MIAMNREP.txt")
Call BvpEnv("PLTPAM", "PAC7PM", Rep_USR & "\MIAMPMAC.txt")
Call BvpEnv("PLTPAM", "PAC7RQ", Rep_USR & "\MIAMRQER.txt")
Call BvpEnv("PLTPAM", "PAC7DD", Rep_USR & "\MIAMDDPM.txt")
Call RunCmdLog ("BVPLTPAM")
Call Err_Cod(Return , 0 , "PLTPAM")

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

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

</script>
</job>
```

---

## MIBA - Migration Batch

### MIBA - Présentation générale

#### Principe

Cette procédure est un utilitaire qui permet de préparer la migration batch suite à l'extraction des entités dans le référentiel, par la mise en forme de fichiers qui serviront de base aux outils de migration.

### MIBA - Entrées Utilisateur

Ce sont les entrées de la procédure PACX avec les valeurs spécifiques suivantes :

Une ligne '\*' telle que :

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
22	4	nnnn	Numéro de session (blanc=courante)
26	1	'T'	Etat de la session si session historisée
29	4	EXTR	Code de l'extracteur
34	1	'1'	Formatage pour UPDP (PAF)
55	1	'1'	Top demande d'horodatage

Ligne 'O': Options cartes avant / après des commandes de génération

Pos.	Lon.	Valeur	Signification
2	1	'O'	Code ligne
3	3	bbb	Commande de génération
			'GCP' : Programme
			'GCO' : Dialogue
			'GGC' : Client
			'GGS' : Serveur
7	2	av	option carte avant
9	2	ap	option carte après

Ajouter la ligne de commande suivante :

Pos.	Lon.	Valeur	Signification
2	1	'W'	Code ligne
3	1	'1'	Numéro de ligne
4	2	'EX'	
6	1	'+'	Code de sélection de bibliothèque :
			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
		'ALL '	Entité et entités utilisées

### Principe de la migration

Pour une entité donnée, l'extraction va s'effectuer à partir de chacune des bibliothèques d'appartenance de chaque élément de son dossier, au sein d'un même réseau.

#### Première étape : extraction PACX en vision '+'

A chaque extraction d'une fiche d'entité, on recherche la bibliothèque d'appartenance de chaque élément de son dossier.

La bibliothèque la plus haute est la bibliothèque de référence. Dans chaque bibliothèque plus basse trouvée, le dossier de l'entité sera ultérieurement créé (étape suivante).

L'ensemble des bibliothèques d'un même réseau qui contient un élément du dossier est détecté.

Pour chaque bibliothèque, une carte de commande en vision 'C' de l'entité extraite, accompagnée d'une carte assign appropriée, est formatée et triée. L'ensemble de ces cartes va constituer le fichier d'entrée de la seconde étape.

#### Seconde étape : extraction PACX en vision 'C'

L'extraction des entités est réalisée dans la bibliothèque d'extraction et les bibliothèques de niveau supérieur. Dans chaque bibliothèque d'extraction, un dossier d'entité est créé, hérité de ses bibliothèques supérieures ou égales.

L'extraction PACX a toutefois été enrichie:



- L'extraction du Segment s'accompagne de l'extraction de sa Structure de Données.
- L'extraction d'un Ecran s'accompagne de l'extraction de sa fiche Dialogue.

Seules les demandes erronées sont éditées dans l'état PAC7EZ.

Cette procédure nécessite la présence du fichier PAC7PM en entrée du programme BVPACS96.

Ce fichier, défini avec des enregistrements de 50 caractères, contient, en position 1, les codes VA Pacbase des Programmes utilisés en tant que Macrostructure, mais non définis comme tel dans le référentiel.

Ce fichier est obtenu via la procédure MIAM.

Cette procédure doit également permettre de transformer des caractères spéciaux non reconnus dans un code entité par un autre caractère ou des mots interdits (ex: mots réservés windows) par un autre code. La table de correspondance est représentée par le fichier PAC7CS en entrée des programmes BVPACS94 et BVPACS96 (fichier optionnel, fourni par la procédure MIAM)

Ce fichier, s'il existe, doit être renseigné dans le script BVPMIBA par valorisation de la ligne WshEnv("BVP\_MibaSpecChar").

Si la ligne n'est pas valorisée, un fichier MibaSpecChar.txt sera créé vide sous le répertoire [NOMBASE].

Cette procédure génère également les mouvements d'entrée des procédures :

- MIMA (migration des Macrostructures) : commandes GCM et GMO.
- GPRT (EDITION - GENERATION): commandes GCP, GCO, GGC et GGS
- MIA1 (Aide à la migration des Programmes): commandes GCP
- MIA2 (Aide à la migration des Ecrans): commandes GCO
- MIA3 (Aide à la migration des Clients): commandes GGC
- MIA4 (Aide à la migration des Serveurs): commandes GGS
- Les options cartes avant/après de ces mouvements proviennent des lignes 'O' des entrées utilisateurs.

**.Remarques :** commandes GMO : création des Macrostructures prioritaires.  
Toutes les lignes de spécifique d'un écran qui ont pour effet un déplacement d'une fonction ou sous fonction d'une Macrostructure sont remontées dans la Macrostructure prioritaire.

Lors de la migration du référentiel, les écrans ayant des lignes de positionnements relatifs spécifiques (\*A, \*P, \*C, \*R) sont détectés. Une ligne de commande GMO est créée pour chacun de ces écrans qui seront transformés en Macrostructures prioritaires.

Ces Macrostructures seront générées par la procédure MIMA sous le code : code écran suffixé de 'SP'.

## MIBA - 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
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	Entrées utilisateur
PAC7MA	NUL	Entrée	Fichier travail
PAC7ES	NUL	Entrée	Fichier travail
PAC7BM	Rép. tmp : WBM	Entrée/Sortie	Entrées utilisateur
PAC7MM	Rép. tmp : WMM	Entrée/Sortie	Fichier travail
PAC7MJ	Rép. tmp : WMJ	Entrée/Sortie	Fichier travail
PAC7TE	Rép. tmp : WTE	Entrée/Sortie	Fichier travail
PAC7RE	Rép. tmp : WRE	Entrée/Sortie	Fichier travail
PAC7RM	Rép. tmp : WRM	Entrée/Sortie	Fichier travail
PAC7WD	Rép. tmp : WWD	Entrée/Sortie	Mouvements extraits
SYSEXT	Rép. tmp : WSY	Entrée/Sortie	Fichier de travail
PAC7MV	Rép. tmp : WMV	Sortie	Fichier travail
PAC7CP	NULCP	Sortie	Fichier travail

Code	Nom physique	Type	Libellé
PAC7MR	Rép. tmp : WMR	Sortie	Fichier travail
PAC7MX	Rép. tmp : WMX	Sortie	Entités non extraites (PACX)
PAC7GY	NUL	Sortie	Mouvements extraits pour UPDP
PAC7RQ	Rép. tmp : WQR	Sortie	Correspondance Entités/Bibliothèque origine
PAC7TD	Rép. tmp : WTD	Sortie	Fichier travail
PAC7UE	Rép. tmp : WUE	Sortie	Fichier travail
PAC7IA	Rép. user : PACXIA	Etat	Edition générale de l'enchaînement des programmes
PAC7DD	Rép. user : PACXDD	Etat	Edition des anomalies sur mouvements en entrée
PAC7ED	Rép. tmp : WED	Etat	Compte-rendu d'extractions
PAC7EE	Rép. user : PACXEE	Etat	Compte-rendu d'extractions
PAC7EG	Rép. tmp : WEG	Etat	Compte-rendu d'extractions
PAC7EM	Rép. tmp : WEM	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. tmp : WEU	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 (précisée dans PAC7EZ)
- 8 : Erreur dans la ligne '\*' (précisée dans PAC7DD)

Tri et mise en forme cartes commande : BVPACS92

Code	Nom physique	Type	Libellé
PAC7QR	Rép. tmp : WQR	Entrée	Correspondance Entités/Bibliothèque
PAC7RQ	Rép. tmp : WPACXRQ	Sortie	Correspondance Entités/Bibliothèque triée
PAC7MZ	Rép. tmp : WMB	Sortie	Cartes de commandes pour entrée PACX

### Extraction : PACX2

Cette étape extrait les mouvements en fonction des entrées issues du programme BVPACS92.

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
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	Rép. tmp : WMB	Entrée	Entrées utilisateur
PAC7MA	NUL	Entrée	Fichier travail
PAC7ES	NUL	Entrée	Fichier travail
PAC7BM	Rép. tmp : WBM	Entrée/Sortie	Entrées utilisateur
PAC7MM	Rép. tmp : WMM	Entrée/Sortie	Fichier travail
PAC7MJ	Rép. tmp : WMJ	Entrée/Sortie	Fichier travail
PAC7TE	Rép. tmp : WTE	Entrée/Sortie	Fichier travail
PAC7RE	Rép. tmp : WRE	Entrée/Sortie	Fichier travail
PAC7RM	Rép. tmp : WRM	Entrée/Sortie	Fichier travail
PAC7WD	Rép. tmp : WWD	Entrée/Sortie	Mouvements extraits
SYSEXT	Rép. tmp : WSY2	Entrée/Sortie	Fichier de travail

Code	Nom physique	Type	Libellé
PAC7MV	Rép. tmp : WMV	Sortie	Fichier travail
PAC7MR	Rép. tmp : WMR	Sortie	Fichier travail
PAC7MX	Rép. tmp : WMX	Sortie	Entités non extraites (PACX)
PAC7CP	Rép. tmp : WCP	Sortie	Fichier travail
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements extraits pour UPDP
PAC7RQ	NUL	Sortie	Correspondance Entités/Bibliothèque origine
PAC7TD	Rép. tmp : WTD	Sortie	Fichier travail
PAC7UE	Rép. tmp : WUE	Sortie	Fichier travail
PAC7IA	Rép. user : PACX2IA	Etat	Edition générale de l'enchaînement des programmes
PAC7DD	Rép. user : PACX2DD	Etat	Edition des anomalies sur mouvements en entrée
PAC7ED	Rép. tmp : WED	Etat	Compte-rendu d'extractions
PAC7EE	Rép. user : PACX2EE	Etat	Compte-rendu d'extractions
PAC7EG	Rép. tmp : WEG	Etat	Compte-rendu d'extractions
PAC7EM	Rép. tmp : WEM	Etat	Compte-rendu d'extractions
PAC7EP	Rép. user : PACX2EP	Etat	Compte-rendu d'extractions
PAC7EQ	Rép. user : PACX2EQ	Etat	Compte-rendu d'extractions
PAC7EU	Rép. tmp : WEU	Etat	Compte-rendu d'extractions
PAC7EZ	Rép. user : PACX2EZ	Etat	Compte-rendu d'extractions

Codes retour :

- 0 : Pas d'erreur
- 4 : Erreur dans les entrées utilisateur (précisée dans PAC7EE) ou dans les extractions (précisée dans PAC7EZ)

- 8 : Erreur dans la ligne '\*' (précisée dans PAC7DD)

#### Migration batch : BVPACS94

Code	Nom physique	Type	Libellé
PAC7RQ	Rép. tmp : WPACXRQ	Entrée	Correspondance Entités/Bibliothèque la plus basse
PAC7CS	Rép. user : MibaSpecChar	Entrée	Fichier de correspondance des caractères spéciaux
PAC7QR	Rép. tmp : WPAC7QR	Sortie	Correspondance Entités/Bibliothèque la plus basse

#### Migration batch : BVPACS96

Code	Nom physique	Type	Libellé
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7GY	Rép. tmp : WGY	Entrée	Fichier issu de l'extraction PACX
PAC7PM	Rép. user : MvMiba	Entrée	Fichier de la liste des programmes à transformer en MSP
PAC7CS	Rép. user : MibaSpecChar	Entrée	Fichier de correspondance des caractères spéciaux
PAC7TA	Rép. tmp : WTA	Sortie	Fichier des regroupements
PAC7MM	Rép. user : INPUTMIMA	Sortie	Fichier des commandes MIMA
PAC7MI	Rép. user : MIA1	Sortie	Fichier des commandes GCP
PAC7MO	Rép. user : MIA2	Sortie	Fichier des commandes GCO
PAC7MC	Rép. user : MIA3	Sortie	Fichier des commandes GGC
PAC7MS	Rép. user : MIA4	Sortie	Fichier des commandes GGS

#### Réorganisation des mouvements : BVPACS97

Code	Nom physique	Type	Libellé
PAC7TA	Rép. tmp : WTA	Entrée	Fichier des regroupements
PAC7AT	Rép. tmp : WAT	Sortie	Fichier des regroupements retrié

#### Tri et mise en forme fichier import : BVPACS91

Code	Nom physique	Type	Libellé
PAC7AT	Rép. tmp : WAT	Entrée	Fichier des regroupements
PAC7TI	Rép. tmp : WTI	Sortie	Fichier des regroupements retrié

## Tri global : BVPACS98

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Sortie	Données de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	libellés d'erreur
PAC7CP	Rép. tmp : WCP	Entrée	Compteur des entités extraites
PAC7RQ	Rép. tmp : WPAC7QR	Entrée	Correspondance Entités/Bibliothèque la plus basse
PAC7TI	Rép. tmp : WTI	Entrée	Fichier des regroupements retrié
PAC7TG	Rép. user : MIBA	Sortie	Fichier résultat tri global

### MIBA - Script d'exécution

```
| -----
| VISUALAGE PACBASE
| -----
| BATCH MIGRATION
| -----
|
<job id=MIBA>

<script language="VBScript">
Dim MyProc
MyProc = "MIBA"
</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("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","PAC7RQ",Rep_TMP & "\\WQR.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\\NULCP.tmp")
Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\\WUE.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\\WTD.tmp")
Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\\WEU.tmp")
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 , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACS92"))
'-----
Call BvpEnv("PACS92","PAC7QR",Rep_TMP & "\\WQR.tmp")
Call BvpEnv("PACS92","PAC7RQ",Rep_TMP & "\\WPACXRQ.tmp")
Call BvpEnv("PACS92","PAC7MZ",Rep_TMP & "\\WMB.tmp")
Call RunCmdLog ("BVPACS92")
Call Err_Cod(Return , 0 , "PACS92")

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"
Call BvpEnv("PACX","PAC7MB",Rep_TMP & "\\WMB.tmp")
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", "PAC7RQ", Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX", "PAC7RM", Rep_TMP & "\\WRM.tmp")
Call BvpEnv("PACX", "PAC7MA", Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX", "PAC7ES", Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX", "PAC7UE", Rep_TMP & "\\WUE.tmp")
Call BvpEnv("PACX", "PAC7CP", Rep_TMP & "\\WCP.tmp")
Call BvpEnv("PACX", "PAC7GY", Rep_TMP & "\\WGY.tmp")
Call BvpEnv("PACX", "PAC7TD", Rep_TMP & "\\WTD.tmp")
Call BvpEnv("PACX", "PAC7MV", Rep_TMP & "\\WMV.tmp")
Call BvpEnv("PACX", "PAC7MR", Rep_TMP & "\\WMR.tmp")
Call BvpEnv("PACX", "PAC7MX", Rep_TMP & "\\WMX.tmp")
Call BvpEnv("PACX", "PAC7IA", Rep_USR & "\\PACX2IA.txt")
Call BvpEnv("PACX", "PAC7DD", Rep_USR & "\\PACX2DD.txt")
Call BvpEnv("PACX", "PAC7ED", Rep_TMP & "\\WED.tmp")
Call BvpEnv("PACX", "PAC7EE", Rep_USR & "\\PACX2EE.txt")
Call BvpEnv("PACX", "PAC7EG", Rep_TMP & "\\WEG.tmp")
Call BvpEnv("PACX", "PAC7EM", Rep_TMP & "\\WEM.tmp")
Call BvpEnv("PACX", "PAC7EP", Rep_USR & "\\PACX2EP.txt")
Call BvpEnv("PACX", "PAC7EQ", Rep_USR & "\\PACX2EQ.txt")
Call BvpEnv("PACX", "PAC7EU", Rep_TMP & "\\WEU.tmp")
Call BvpEnv("PACX", "PAC7EZ", Rep_USR & "\\PACX2EZ.txt")
Call BvpEnv("PACX", "SYSEXT", Rep_TMP & "\\WSY2.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 , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACS94"))
'-----
WshEnv("PAC7CS") = BVP_MibaCs
Call BvpEnv("PACS94", "PAC7RQ", Rep_TMP & "\\WPACXRQ.tmp")
Call BvpEnv("PACS94", "PAC7QR", Rep_TMP & "\\WPAC7QR.tmp")
Call RunCmdLog ("BVPACS94")
Call Err_Cod(Return , 0 , "PACS94")

Call Msg_Log (Array("1022" , "PACS96"))
'-----
Call BvpEnv("PACS96", "PAC7GY", Rep_TMP & "\\WGY.tmp")
WshEnv("PAC7CS") = BVP_MibaCs
WshEnv("PAC7PM") = BVP_MibaFi
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACS96", "PAC7TA", Rep_TMP & "\\WTA.tmp")
Call BvpEnv("PACS96", "PAC7MM", Rep_USR & "\\INPUTMIMA.txt")
Call BvpEnv("PACS96", "PAC7MI", Rep_USR & "\\MIA1.txt")
Call BvpEnv("PACS96", "PAC7MO", Rep_USR & "\\MIA2.txt")
Call BvpEnv("PACS96", "PAC7MC", Rep_USR & "\\MIA3.txt")
Call BvpEnv("PACS96", "PAC7MS", Rep_USR & "\\MIA4.txt")

```

```

Call RunCmdLog ("BVPACS96")
Call Err_Cod(Return , 0 , "PACS96")

Call Msg_Log (Array("1022" , "PACS97"))
'-----
Call BvpEnv("PACS97","PAC7AT",Rep_TMP & "\\WAT.tmp")
Call BvpEnv("PACS97","PAC7TA",Rep_TMP & "\\WTA.tmp")
Call RunCmdLog ("BVPACS97")
Call Err_Cod(Return , 0 , "PACS97")

Call Msg_Log (Array("1022" , "PACS91"))
'-----
Call BvpEnv("PACS91","PAC7AT",Rep_TMP & "\\WAT.tmp")
Call BvpEnv("PACS91","PAC7TI",Rep_TMP & "\\WTI.tmp")
Call RunCmdLog ("BVPACS91")
Call Err_Cod(Return , 0 , "PACS91")

Call Msg_Log (Array("1022" , "PACS98"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
WshEnv("PAC7AR") = Rep_BASE & "\\AR"
Call BvpEnv("PACS98","PAC7CP",Rep_TMP & "\\WCP.tmp")
Call BvpEnv("PACS98","PAC7TI",Rep_TMP & "\\WTI.tmp")
Call BvpEnv("PACS98","PAC7RQ",Rep_TMP & "\\WPAC7QR.tmp")
Call BvpEnv("PACS98","PAC7TG",Rep_USR & "\\MIBA.ta2")
Call RunCmdLog ("BVPACS98")
Call Err_Cod(Return , 0 , "PACS98")

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

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

</script>
</job>

```

---

## MIBR - Migration Batch contexte

### MIBR - Présentation générale

#### Principe

Cette procédure est un utilitaire qui permet de préparer la migration batch suite à l'extraction des entités du référentiel dans un contexte donné (session et réseau ascendant ou descendant d'une bibliothèque) par la mise en forme de fichiers qui serviront de base aux outils de migration.

## MIBR - Entrées Utilisateur

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
22	4	nnnn	Numéro de session blanc = session courante
26	1		Type de session
		'T'	Si sélection session historisée
		' '	Si session courante ou H
33	1		Code extraction de bibliothèque
		'A'	Extraction d'une bibliothèque de plus bas niveau et de ses bibliothèques supérieures
		'D'	Extraction d'une bibliothèque et de ses bibliothèques dépendantes
		'U'	Extraction d'une bibliothèque unique

Autant de lignes que de bibliothèques à extraire.

Pos.	Lon.	Valeur	Signification
3	3	bbb	Code bibliothèque à extraire

Ligne 'O': Options cartes avant / après des commandes de génération

Pos.	Lon.	Valeur	Signification
2	1	'O'	Code ligne
3	3	bbb	Commande de génération
			'GCP' : Programme
			'GCO' : Dialogue
			'GGC' : Client
			'GGS' : Serveur
7	2	av	option carte avant
9	2	ap	option carte après

Ligne 'P': projet issu de l'éclatement de bibliothèques

Pos.	Lon.	Valeur	Signification
2	1	'P'	Code ligne
3	3	bbb	Code bibliothèque à éclater
6	3	ccc	Type entité Pacbase à déplacer
			'E ' : Rubrique
			'S ' : Structure de donnée et Segment
			'R ' : Etat
			'B ' : Bloc base de donnée
			'P ' : Programme
			'O ' : Dialogue et écran
			'F ' : Méta-Entité Cliente
			'\$tt' : Entité Utilisateur Cliente (tt = code d'appel)
			'T ' : Texte
9	15		Code du nouveau projet

### Principe de la migration

Un contexte d'extraction est sélectionné. La session et la bibliothèque dont sera extrait le réseau ascendant ou descendant sont précisées.

Première étape : extraction des entités présentes dans le contexte.

Tous les index de la base sont lus en séquentiel.

La sélection s'effectue sur les index primaires de toutes les familles d'entités. La bibliothèque la plus haute de la fiche de l'entité est la bibliothèque de référence. La détection d'un élément du dossier dans une bibliothèque plus basse va provoquer la création ultérieure du dossier de l'entité dans cette bibliothèque (étape suivante).

Les overrides des fiches des entités et de leurs dossiers complets, au sein d'un même réseau de bibliothèques sont ainsi détectés (c'est à dire la présence multiple de la fiche ou l'existence d'une ou plusieurs parties de son dossier dans une bibliothèque différente de celle de la fiche).

L'ensemble des bibliothèques d'un même réseau qui contient un élément du dossier est détecté.

Pour chaque bibliothèque, une carte de commande en vision 'C' de l'entité extraite, accompagnée d'une carte assign appropriée, est formatée et triée. L'ensemble de ces cartes va constituer le fichier d'entrée de la troisième étape.

#### Seconde étape : prise en compte des entités non reconduites

Cette étape permet de ne pas reconduire en sortie toutes les entités présentes dans le fichier PAC7NR.

Elle nécessite donc la présence du fichier PAC7NR des entités non reconduites, fourni par la procédure MIAM.

Ce fichier, s'il existe, doit être renseigné dans le script d'exécution par valorisation de la ligne WshEnv("BVP\_MibrNotRet").

Si la ligne n'est pas valorisée, un fichier Mvmibrnr.txt sera créé vide sous le répertoire [NOMBASE].

#### Troisième étape : extraction PACX en vision 'C'

L'extraction des entités est réalisée dans la bibliothèque d'extraction et les bibliothèques de niveau supérieur. Dans chaque bibliothèque d'extraction, un dossier d'entité est créé, hérité de ses bibliothèques supérieures ou égales.

L'extraction PACX a toutefois été enrichie:

- L'extraction du Segment s'accompagne de l'extraction de sa Structure de Données.
- L'extraction d'un Ecran s'accompagne de l'extraction de sa fiche Dialogue.

Seules les demandes erronées apparaissent dans l'état PAC7EZ.

Cette procédure nécessite la présence du fichier PAC7PM en entrée du programme BVPACS96.

Ce fichier, défini avec des enregistrements de 50 caractères, contient, en position 1, les codes VA Pacbase des Programmes utilisés en tant que Macrostructure, mais non définis comme tel dans le référentiel.

Ce fichier est obtenu via la procédure MIAM.

Cette procédure doit également permettre de transformer des caractères spéciaux non reconnus dans un code entité par un autre caractère ou des mots interdits (ex: mots réservés windows) par un autre code..

La table de correspondance est représentée par le fichier PAC7CS en entrée des programmes BVPACS94 et BVPACS96 (fichier optionnel, fourni par la procédure MIAM).

Cette procédure génère également les mouvements d'entrée des procédures suivantes :

- MIMA (migration des Macrostructures): commandes GCM et GMO
- MIA1 (Aide à la migration des Programmes): commandes GCP
- MIA2 (Aide à la migration des Ecrans): commandes GCO
- MIA3 (Aide à la migration des Clients): commandes GGC
- MIA4 (Aide à la migration des Serveurs): commandes GGS
- Les options cartes avant/après de ces mouvements proviennent des lignes 'O' des entrées utilisateurs.(cf sous-chapitre "MIAx - Cartes de contrôles optionnelles")

**.Remarques :** commandes GMO : création des Macrostructures prioritaires.

Toutes les lignes de spécifique d'un écran qui ont pour effet un déplacement d'une fonction ou sous fonction d'une Macrostructure sont remontées dans la Macrostructure prioritaire.

Lors de la migration du référentiel, les écrans ayant des lignes de positionnements relatifs spécifiques (\*A, \*P, \*C, \*R) sont détectés. Une ligne de commande GMO est créée pour chacun de ces écrans qui seront transformés en Macrostructures prioritaires.

Ces Macrostructures seront générées par la procédure MIMA sous le code : code écran suffixé de 'SP'.

## **MIBR - Eclatement des bibliothèques**

Une volumétrie trop importante de certaines bibliothèques peut être résolue par la procédure MIBR qui propose leur éclatement.

La bibliothèque à alléger est divisée en N nouveaux projets.

A l'aide des lignes P, il faut définir pour chaque type d'entité Pacbase concernée vers quel nouveau projet elles seront déplacées. Les projets sont tous dépendants et leur hiérarchie dépend du type de l'entité Pacbase associé.

L'ordre est décrit dans la table des entrées de la ligne P.

Cependant les projets des entités utilisateurs peuvent être placés au bon vouloir de l'utilisateur.

On peut associer le même projet à plusieurs types d'entité Pacbase à condition que les types soient directement dépendants (hiérarchie immédiate).

Pour s'assurer que la liste des nouveaux projets issus d'un éclatement de bibliothèque soit cohérente, il est contrôlé que tous les types d'entité Pacbase (à l'exception des méta-entités) sont présents dans les lignes P.

Limites des entrées P:

- 5 bibliothèques à éclater
- 40 projets par bibliothèque

Soit 30 projets pour les entités utilisateurs et un projet par type pour les autres entités Pacbase. On peut utiliser le code générique \$\*\* pour rassembler toutes les entités utilisateurs dans un même projet. On peut demander un projet pour une E.U. donnée et un autre projet pour l'ensemble des autres E.U.(\$\*\*).

-Le code projet ne peut contenir les caractères ' \ /:%?<>"| ' ou être équivalent au mot réservé 'INTER'.

Règles d'utilisations :

Pour utiliser cette option, la demande de migration se fera en mode descendant 'D' .

Ordre de saisie des cartes en entrée de MIBR:

- Ensemble des cartes d'extraction des réseaux de bibliothèques.
- Ensemble des cartes options cartes avant / après
- Ensemble des cartes projets.

Respecter la hiérarchie des bibliothèques à éclater en ordonnant les lignes P de la bibliothèque la plus haute à la plus basse.

La procédure MIBR, suite à la saisie des demandes utilisateurs en terme de dispatching, sauvegarde ces informations. Ainsi, elles pourront être prises en compte dans la procédure MIBJ.

Ce fichier SPLITLIB issu de MIBR va constituer une entrée de MIBJ.

Il contient l'ensemble des projets issu de l'éclatement des bibliothèques ainsi que leur hiérarchie.

Ce fichier, s'il existe, doit être renseigné dans le script d'exécution par valorisation de la ligne WshEnv("BVP\_MibrSplitLib").

Si la ligne n'est pas valorisée, un fichier SPLITLIB.txt sera créé vide sous le répertoire [NOMBASE].

## MIBR - Description des étapes

### Chargement fichier NR des entités non reprises: BVPACS9B

Code	Nom physique	Type	Libellé
PAC7RN	Rép. user : MibrNotRet	Entrée	Entités non reprises issues de MIAM
PAC7NR	Rép. tmp : WNR	Sortie	Entités non reprises issues de MIAM formatées

### Extraction du réseau : BVPACS21

Code	Nom physique	Type	Libellé
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
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
PAC7MB	Entrée utilisateur	Entrée	Mouvements utilisateur
PAC7RQ	Rép. tmp : WQR	Sortie	Correspondance Entités/Bibliothèque la plus basse
PAC7DD	Rép. user : MIBRDDDS21	Sortie	Etat d'anomalies

### Prise en compte des entités non reconduites BVPACS9A

Code	Nom physique	Type	Libellé
PAC7NR	Rép. tmp : WNR	Entrée	Entités non reprises issues de MIAM
PAC7QR	Rép. tmp : WQR	Entrée	Correspondance Entité/Bibliothèque la plus basse



Code	Nom physique	Type	Libellé
PAC7RQ	Rép. tmp : WRQ	Sortie	Correspondance Entité/Bibliothèque épurée

Tri et mise en forme cartes commande : BVPACS92

Code	Nom physique	Type	Libellé
PAC7QR	Rép. tmp : WRQ	Entrée	Correspondance Entités/Bibliothèque
PAC7RQ	Rép. tmp : WPACXRQ	Sortie	Correspondance Entités/Bibliothèque triée
PAC7MZ	Rép. tmp : WMB	Sortie	Cartes de commandes pour entrée PACX

Extraction : PACX

Cette étape extrait les mouvements en fonction des entrées issues du programme BVPACS92.

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
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	Entrées utilisateur
PAC7MA	NUL	Entrée	Fichier travail
PAC7ES	NUL	Entrée	Fichier travail
PAC7BM	Rép. tmp : WBM	Entrée/Sortie	Entrées utilisateur
PAC7MM	Rép. tmp : WMM	Entrée/Sortie	Fichier travail
PAC7MJ	Rép. tmp : WMJ	Entrée/Sortie	Fichier travail

Code	Nom physique	Type	Libellé
PAC7TE	Rép. tmp : WTE	Entrée/Sortie	Fichier travail
PAC7RE	Rép. tmp : WRE	Entrée/Sortie	Fichier travail
PAC7RM	Rép. tmp : WRM	Entrée/Sortie	Fichier travail
PAC7WD	Rép. tmp : WWD	Entrée/Sortie	Mouvements extraits
SYSEXT	Rép. tmp : WSY	Entrée/Sortie	Fichier de travail
PAC7MV	Rép. tmp : WMV	Sortie	Fichier travail
PAC7MR	Rép. tmp : WMR	Sortie	Fichier travail
PAC7MX	Rép. tmp : WMX	Sortie	Entités non extraites (PACX)
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements extraits pour UPDP
PAC7RQ	NUL	Sortie	Correspondance Entités/Bibliothèque origine
PAC7TD	Rép. tmp : WTD	Sortie	Fichier travail
PAC7UE	Rép. tmp : WUE	Sortie	Fichier travail
PAC7CP	Rép. tmp : WCP	Sortie	Fichier travail
PAC7IA	Rép. user : PACXIA	Etat	Edition générale de l'enchaînement des programmes
PAC7DD	Rép. user : PACXDD	Etat	Edition des anomalies sur mouvements en entrée
PAC7ED	Rép. tmp : WED	Etat	Compte-rendu d'extractions
PAC7EE	Rép. user : PACXEE	Etat	Compte-rendu d'extractions
PAC7EG	Rép. tmp : WEG	Etat	Compte-rendu d'extractions
PAC7EM	Rép. tmp : WEM	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. tmp : WEU	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 (précisée dans PAC7EZ)
- 8 : Erreur dans la ligne '\*' (précisée dans PAC7DD)

Migration batch : BVPACS94

Code	Nom physique	Type	Libellé
PAC7RQ	Rép. tmp : WPACXRQ	Entrée	Correspondance Entités/Bibliothèque la plus basse
PAC7CS	Rép. user : MibaSpecChar	Entrée	Fichier de correspondance des caractères spéciaux
PAC7QR	Rép. user : SPLITLIB	Sortie	Correspondance Entités/Bibliothèque la plus basse

Migration batch : BVPACS96

Code	Nom physique	Type	Libellé
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7GY	Rép. tmp : WGY	Entrée	Fichier issu de l'extraction PACX
PAC7PM	Rép. user : MvMiba	Entrée	Fichier de la liste des programmes à transformer en MSP
PAC7CS	Rép. user : MibaSpecChar	Entrée	Fichier de correspondance des caractères spéciaux
PAC7TA	Rép. tmp : WTA	Sortie	Fichier des regroupements
PAC7MM	Rép. user : INPUTMIMA	Sortie	Fichier des commandes MIMA
PAC7MI	Rép. user : MIA1	Sortie	Fichier des commandes GCP
PAC7MO	Rép. user : MIA2	Sortie	Fichier des commandes GCO
PAC7MC	Rép. user : MIA3	Sortie	Fichier des commandes GGC
PAC7MS	Rép. user : MIA4	Sortie	Fichier des commandes GGS

Réorganisation des mouvements : BVPACS97

Code	Nom physique	Type	Libellé
PAC7TA	Rép. tmp : WTA	Entrée	Fichier des regroupements
PAC7AT	Rép. tmp : WAT	Sortie	Fichier des regroupements retrié

### Tri et mise en forme fichier import : BVPACS91

Code	Nom physique	Type	Libellé
PAC7AT	Rép. tmp : WAT	Entrée	Fichier des regroupements
PAC7TI	Rép. tmp : WTI	Sortie	Fichier des regroupements retrié

### Tri global : BVPACS98

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Sortie	Données de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	libellés d'erreur
PAC7CP	Rép. tmp : WCP	Entrée	Compteur des entités extraites
PAC7RQ	Rép. user : SPLITLIB	Entrée	Correspondance Entités/Bibliothèque la plus basse
PAC7TI	Rép. tmp : WAT	Entrée	Fichier des regroupements retrié
PAC7TG	Rép. user : MIBR	Sortie	Fichier résultat tri global

## **MIBR - Script d'exécution**

```
|-----|
| VISUALAGE PACBASE |
|-----|
| BATCH MIGRATION |
|-----|
|
|<job id=MIBR>
|
|<script language="VBScript">
| Dim MyProc
| MyProc = "MIBR"
|</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" , "PACS9B"))
| '-----'
| WshEnv("PAC7RN") = BVP_MibrNr
| Call BvpEnv("PACS9B", "PAC7NR", Rep_TMP & "\WNR.tmp")
| Call RunCmdLog ("BVPACS9B")
| Call Err_Cod(Return , 0 , "PACS9B")
|</script>
```

```

Call Msg_Log (Array("1022" , "PACS21"))
'-----
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("PAC7MB") = Fic_Input
Call BvpEnv("PACS21","PAC7DD",Rep_USR & "\MIBRDDS21.txt")
Call BvpEnv("PACS21","PAC7RQ",Rep_TMP & "\WQR.tmp")
Call RunCmdLog ("BVPACS21")
Call Err_Cod(Return , 0 , "PACS21")

Call Msg_Log (Array("1022" , "PACS9A"))
'-----
Call BvpEnv("PACS9A","PAC7NR",Rep_TMP & "\WNR.tmp")
Call BvpEnv("PACS9A","PAC7QR",Rep_TMP & "\WQR.tmp")
Call BvpEnv("PACS9A","PAC7RQ",Rep_TMP & "\WRQ.tmp")
Call RunCmdLog ("BVPACS9A")
Call Err_Cod(Return , 0 , "PACS9A")

Call Msg_Log (Array("1022" , "PACS92"))
'-----
Call BvpEnv("PACS92","PAC7QR",Rep_TMP & "\WRQ.tmp")
Call BvpEnv("PACS92","PAC7RQ",Rep_TMP & "\WPACXRQ.tmp")
Call BvpEnv("PACS92","PAC7MZ",Rep_TMP & "\WMB.tmp")
Call RunCmdLog ("BVPACS92")
Call Err_Cod(Return , 0 , "PACS92")

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"
Call BvpEnv("PACX","PAC7MB",Rep_TMP & "\WMB.tmp")
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","PAC7RQ",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\WCP.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\WTD.tmp")

```

```

Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\\WEU.tmp")
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 , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACS94"))
'-----
WshEnv("PAC7CS") = BVP_MibaCs
Call BvpEnv("PACS94","PAC7RQ",Rep_TMP & "\\WPACXRQ.tmp")
Call BvpEnv("PACS94","PAC7QR",Rep_USR & "\\SPLITLIB.txt")
Call RunCmdLog ("BVPACS94")
Call Err_Cod(Return , 0 , "PACS94")

Call Msg_Log (Array("1022" , "PACS96"))
'-----
Call BvpEnv("PACS96","PAC7GY",Rep_TMP & "\\WGY.tmp")
WshEnv("PAC7CS") = BVP_MibaCs
WshEnv("PAC7PM") = BVP_MibaFi
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACS96","PAC7TA",Rep_TMP & "\\WTA.tmp")
Call BvpEnv("PACS96","PAC7MM",Rep_USR & "\\INPUTMIMA.txt")
Call BvpEnv("PACS96","PAC7MI",Rep_USR & "\\MIA1.txt")
Call BvpEnv("PACS96","PAC7M0",Rep_USR & "\\MIA2.txt")
Call BvpEnv("PACS96","PAC7MC",Rep_USR & "\\MIA3.txt")
Call BvpEnv("PACS96","PAC7MS",Rep_USR & "\\MIA4.txt")
Call RunCmdLog ("BVPACS96")
Call Err_Cod(Return , 0 , "PACS96")

Call Msg_Log (Array("1022" , "PACS97"))
'-----
Call BvpEnv("PACS97","PAC7AT",Rep_TMP & "\\WAT.tmp")
Call BvpEnv("PACS97","PAC7TA",Rep_TMP & "\\WTA.tmp")
Call RunCmdLog ("BVPACS97")
Call Err_Cod(Return , 0 , "PACS97")

Call Msg_Log (Array("1022" , "PACS91"))
'-----

```

```

Call BvpEnv("PACS91","PAC7AT",Rep_TMP & "\WAT.tmp")
Call BvpEnv("PACS91","PAC7TI",Rep_TMP & "\WTI.tmp")
Call RunCmdLog ("BVPACS91")
Call Err_Cod(Return , 0 , "PACS91")

Call Msg_Log (Array("1022" , "PACS98"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PACS98","PAC7CP",Rep_TMP & "\WCP.tmp")
Call BvpEnv("PACS98","PAC7TI",Rep_TMP & "\WTI.tmp")
Call BvpEnv("PACS98","PAC7RQ",Rep_USR & "\SPLITLIB.txt")
Call BvpEnv("PACS98","PAC7TG",Rep_USR & "\MIBR.ta2")
Call RunCmdLog ("BVPACS98")
Call Err_Cod(Return , 0 , "PACS98")

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

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

</script>
</job>

```

---

## MIBJ - Migration Journal

### MIBJ - Présentation générale

#### Principe

La procédure de migration des mouvements du journal (MIBJ) s'applique au fichier journal archivé du référentiel. Elle met en forme des fichiers qui serviront de base aux outils de migration

- Elle permet de sélectionner le journal dans une plage de dates, sessions, bibliothèques, etc.
- Elle permet d'extraire du référentiel les entités des mouvements sélectionnés.

L'extraction s'effectue dans la bibliothèque et la session du mouvement d'une part et dans toutes ses bibliothèques inférieures où des overrides de la fiche ou d'un élément du dossier de l'entité ont été détectées.

Pour chaque entité est indiquée la bibliothèque la plus haute du référentiel où elle a été détectée (bibliothèque source).

Les entités créées ou modifiées sont extraites, les entités annulées sont répertoriées.

## MIBJ - Entrées Utilisateur

Une ligne '\*' avec code utilisateur et mot de passe.

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

Une ligne 'J' donnant les caractéristiques de l'extraction :

Pos.	Lon.	Valeur	Signification
2	1	'J'	Code ligne
24	4	ssss	Numéro de session de début
28	4	ssss	Numéro de session de fin
32	8	ssaammjj	Date de début
40	8	ssaammjj	Date de fin
48	1		Version des mouvements sélectionnés
		' '	Toutes sessions
		'Z'	Version courante uniquement
		'T'	Version historique seule
49	3	bbb	Code de la bibliothèque sélectionnée

Ligne 'O': Options cartes avant / après des commandes de génération

Pos.	Lon.	Valeur	Signification
2	1	'O'	Code ligne
3	3	bbb	Commande de génération
			'GCP' : Programme
			'GCO' : Dialogue
			'GGC' : Client
			'GGS' : Serveur
7	2	av	option carte avant
9	2	ap	option carte après

## Principe de la migration



Pour chaque mouvement du journal, fiche ou dossier d'une entité, l'extraction va s'effectuer à partir de la bibliothèque et la session du mouvement et à partir de chacune des bibliothèques d'appartenance de chaque élément de son dossier, au sein du réseau inférieur.

#### Première étape : analyse du journal

Pour chaque mouvement, fiche ou dossier, on formate un enregistrement RQ avec le type de l'entité, le code de l'entité, la bibliothèque et la session du mouvement.

A chaque traitement d'une entité utilisateur on force l'extraction de sa méta-entité.

Une entité annulée est signalée dans le fichier RQ.

La liste des sessions historisées est consignée dans le fichier RQ.

Pour chaque entité, on recherche dans le référentiel sa bibliothèque d'appartenance la plus haute ainsi que l'ensemble des bibliothèques plus hautes et plus basses que celle du mouvement du journal où elle est overridee (c'est à dire la présence multiple de la fiche ou l'existence d'une ou plusieurs parties de son dossier dans une bibliothèque différente de celle de la fiche).

La bibliothèque la plus haute est la bibliothèque de référence. Dans chaque bibliothèque plus basse trouvée, le dossier de l'entité sera ultérieurement créé (étape suivante).

Pour chaque bibliothèque, une carte de commande en vision 'C' de l'entité extraite, accompagnée d'une carte assign appropriée, est formatée et triée.

L'ensemble de ces cartes va constituer le fichier d'entrée de la troisième étape.

#### Seconde étape : prise en compte des entités non reconduites

Cette étape permet de ne pas reconduire en sortie toutes les entités présentes dans le fichier PAC7NR.

Elle nécessite donc la présence du fichier PAC7NR des entités non reconduites, fourni par la procédure MIAM.

#### Troisième étape : extraction PACX en vision 'C'

L'extraction des entités est réalisée dans la bibliothèque d'extraction et les bibliothèques de niveau supérieur.

Dans chaque bibliothèque d'extraction, un dossier d'entité est créé, hérité de ses bibliothèques supérieures ou égales.

Seules les demandes erronées sont editées dans l'état PAC7EZ.

### Eclatement des bibliothèques

La procédure MIBR, suite à la saisie des demandes utilisateurs d'éclatement de bibliothèques, sauvegarde ces informations. Elles sont prises en compte dans la procédure MIBJ. Ce fichier SPLITLIB issu de MIBR va constituer une entrée de MIBJ. Il contient l'ensemble des projets issu de l'éclatement des bibliothèques ainsi que leur hiérarchie. Pour chaque entité extraite par MIBJ, présente dans le fichier PAC7RQ , on recherche si elle doit être déplacée vers un nouveau projet , c'est à dire si sa bibliothèque cible est une bibliothèque à éclater. Si oui, le nouveau projet spécifique à son type d'entité lui est attribué .

### **MIBJ - Description des étapes**

Chargement fichier NR des entités non reprises: BVPACS9B

Code	Nom physique	Type	Libellé
PAC7RN	Rép. user : MibrNotRet	Entrée	Entités non reprises issues de MIAM
PAC7NR	Rép. tmp : WNR	Sortie	Entités non reprises issues de MIAM formatées

Extraction du journal : BVPACSJR

Code	Nom physique	Type	Libellé
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
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
PAC7MB	Entrée utilisateur	Entrée	Mouvements utilisateur

Code	Nom physique	Type	Libellé
PAC7PJ	Rép. save : PJ	Entrée	Mouvements archivés
PAC7RQ	Rép. tmp : WRQJR	Sortie	Correspondance Entités/Bibliothèque la plus basse
PAC7DD	Rép. user : MIBJDDSJR	Sortie	Etat d'anomalies
PAC7EE	Rép. user : MIBJEESJR	Sortie	Compte-rendu d'extractions

#### Détection des overrides : BVPACS93

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
PAC7AN	Rép. base : AN	Entrée	Index de la Base de Développement
PAC7SL	Rép. usr : SPLITLIB	Entrée	Eclatement des Bibliothèques
PAC7RQ	Rép. tmp : WRQJR	Entrée	Correspondance Entités/Bibliothèque
PAC7QR	Rép. tmp : WQR	Sortie	Correspondance Entités/Bibliothèque triée

#### Prise en compte des entités non reconduites BVPACS9A

Code	Nom physique	Type	Libellé
PAC7NR	Rép. tmp : WNR	Entrée	Entités non reprises issues de MIAM
PAC7QR	Rép. tmp : WQR	Entrée	Correspondance Entité/Bibliothèque la plus basse
PAC7RQ	Rép. tmp : WRQ	Sortie	Correspondance Entité/Bibliothèque épurée

#### Tri et mise en forme cartes commande : BVPACS92

Code	Nom physique	Type	Libellé
PAC7QR	Rép. tmp : WRQ	Entrée	Correspondance Entités/Bibliothèque
PAC7RQ	Rép. tmp : WPACXRQ	Sortie	Correspondance Entités/Bibliothèque triée

Code	Nom physique	Type	Libellé
PAC7MZ	Rép. tmp : WMB	Sortie	Cartes de commandes pour entrée PACX

### Extraction : PACX

Cette étape extrait les mouvements en fonction des entrées issues du programme BVPACS92.

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
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	Entrées utilisateur
PAC7MA	NUL	Entrée	Fichier travail
PAC7ES	NUL	Entrée	Fichier travail
PAC7BM	Rép. tmp : WBM	Entrée/Sortie	Entrées utilisateur
PAC7MM	Rép. tmp : WMM	Entrée/Sortie	Fichier travail
PAC7MJ	Rép. tmp : WMJ	Entrée/Sortie	Fichier travail
PAC7TE	Rép. tmp : WTE	Entrée/Sortie	Fichier travail
PAC7RE	Rép. tmp : WRE	Entrée/Sortie	Fichier travail
PAC7RM	Rép. tmp : WRM	Entrée/Sortie	Fichier travail
PAC7WD	Rép. tmp : WWD	Entrée/Sortie	Mouvements extraits
SYSEXT	Rép. tmp : WSY	Entrée/Sortie	Fichier de travail
PAC7MV	Rép. tmp : WMV	Sortie	Fichier travail
PAC7MR	Rép. tmp : WMR	Sortie	Fichier travail
PAC7MX	Rép. tmp : WMX	Sortie	Entités non extraites (PACX)

Code	Nom physique	Type	Libellé
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements extraits pour UPDP
PAC7RQ	NUL	Sortie	Correspondance Entités/Bibliothèque origine
PAC7TD	Rép. tmp : WTD	Sortie	Fichier travail
PAC7UE	Rép. tmp : WUE	Sortie	Fichier travail
PAC7CP	Rép. tmp : WCP	Sortie	Fichier travail
PAC7IA	Rép. user : PACXIA	Etat	Edition générale de l'enchaînement des programmes
PAC7DD	Rép. user : PACXDD	Etat	Edition des anomalies sur mouvements en entrée
PAC7ED	Rép. tmp : WED	Etat	Compte-rendu d'extractions
PAC7EE	Rép. user : PACXEE	Etat	Compte-rendu d'extractions
PAC7EG	Rép. tmp : WEG	Etat	Compte-rendu d'extractions
PAC7EM	Rép. tmp : WEM	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. tmp : WEU	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 (précisée dans PAC7EZ)
- 8 : Erreur dans la ligne '\*' (précisée dans PAC7DD)

Migration batch : BVPACS94

Code	Nom physique	Type	Libellé
PAC7RQ	Rép. tmp : WPACXRQ	Entrée	Correspondance Entités/Bibliothèque la plus basse
PAC7CS	Rép. user : MibaSpecChar	Entrée	Fichier de correspondance des caractères spéciaux
PAC7QR	Rép. tmp : WPAC7QR	Sortie	Correspondance Entités/Bibliothèque la plus basse

#### Migration batch : BVPACS96

Code	Nom physique	Type	Libellé
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7GY	Rép. tmp : WGY	Entrée	Fichier issu de l'extraction PACX
PAC7PM	Rép. user : MvMiba	Entrée	Fichier de la liste des programmes à transformer en MSP
PAC7CS	Rép. user : MibaSpecChar	Entrée	Fichier de correspondance des caractères spéciaux
PAC7TA	Rép. tmp : WTA	Sortie	Fichier des regroupements
PAC7MM	Rép. user : INPUTMIMA	Sortie	Fichier des commandes MIMA
PAC7MI	Rép. user : MIA1	Sortie	Fichier des commandes GCP
PAC7MO	Rép. user : MIA2	Sortie	Fichier des commandes GCO
PAC7MC	Rép. user : MIA3	Sortie	Fichier des commandes GGC
PAC7MS	Rép. user : MIA4	Sortie	Fichier des commandes GGS

#### Réorganisation des mouvements : BVPACS97

Code	Nom physique	Type	Libellé
PAC7TA	Rép. tmp : WTA	Entrée	Fichier des regroupements
PAC7AT	Rép. tmp : WAT	Sortie	Fichier des regroupements retérié

#### Tri et mise en forme fichier import : BVPACS91

Code	Nom physique	Type	Libellé
PAC7AT	Rép. tmp : WAT	Entrée	Fichier des regroupements
PAC7TI	Rép. tmp : WTI	Sortie	Fichier des regroupements retérié

#### Tri global : BVPACS98

Code	Nom physique	Type	Libellé
PAC7AR	Rép. base : AR	Sortie	Données de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	libellés d'erreur
PAC7CP	Rép. tmp : WCP	Entrée	Compteur des entités extraites
PAC7RQ	Rép. tmp : WPAC7QR	Entrée	Correspondance Entités/Bibliothèque la plus basse
PAC7TI	Rép. tmp : WAT	Entrée	Fichier des regroupements retrié
PAC7TG	Rép. user : MIBJ	Sortie	Fichier résultat tri global

## MIBJ - Script d'exécution

```

|-----|
| VISUALAGE PACBASE |
|-----|
| ARCHIVED JOURNAL MIGRATION |
|-----|
|
<job id=MIBJ>

<script language="VBScript">
Dim MyProc
MyProc = "MIBJ"
</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" , "PACS9B"))
'-----|
WshEnv("PAC7RN") = BVP_MibrNr
Call BvpEnv("PACS9B", "PAC7NR", Rep_TMP & "\WNR.tmp")
Call RunCmdLog ("BVPACS9B")
Call Err_Cod(Return , 0 , "PACS9B")

Call Msg_Log (Array("1022" , "PACSJR"))
'-----|
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("PAC7MB") = Fic_Input
Call BvpEnv("PACSJR", "PAC7DD", Rep_USR & "\MIBJDDSJR.txt")

```

```

Call BvpEnv("PACSJR","PAC7EE",Rep_USR & "MIBJEESJR.txt")
Call BvpEnv("PACSJR","PAC7PJ",Rep_SAVE & "\PJ")
Call BvpEnv("PACSJR","PAC7RQ",Rep_TMP & "\WRQJR.tmp")
Call RunCmdLog ("BVPACSJR")
Call Err_Cod(Return , 0 , "PACSJR")

Call Msg_Log (Array("1022" , "PACS93"))
'-----
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7SL") = BVP_MibrSl
Call BvpEnv("PACS93","PAC7RQ",Rep_TMP & "\WRQJR.tmp")
Call BvpEnv("PACS93","PAC7QR",Rep_TMP & "\WQR.tmp")
Call RunCmdLog ("BVPACS93")
Call Err_Cod(Return , 0 , "PACS93")

Call Msg_Log (Array("1022" , "PACS9A"))
'-----
Call BvpEnv("PACS9A","PAC7NR",Rep_TMP & "\WNR.tmp")
Call BvpEnv("PACS9A","PAC7QR",Rep_TMP & "\WQR.tmp")
Call BvpEnv("PACS9A","PAC7RQ",Rep_TMP & "\WRQ.tmp")
Call RunCmdLog ("BVPACS9A")
Call Err_Cod(Return , 0 , "PACS9A")

Call Msg_Log (Array("1022" , "PACS92"))
'-----
Call BvpEnv("PACS92","PAC7QR",Rep_TMP & "\WQR.tmp")
Call BvpEnv("PACS92","PAC7RQ",Rep_TMP & "\WPACXRQ.tmp")
Call BvpEnv("PACS92","PAC7MZ",Rep_TMP & "\WMB.tmp")
Call RunCmdLog ("BVPACS92")
Call Err_Cod(Return , 0 , "PACS92")

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"
Call BvpEnv("PACX","PAC7MB",Rep_TMP & "\WMB.tmp")
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","PAC7RQ",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\WGY.tmp")

```



```

Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\\WCP.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\\WTD.tmp")
Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\\WEU.tmp")
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 , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACS94"))
'-----
WshEnv("PAC7CS") = BVP_MibaCs
Call BvpEnv("PACS94","PAC7RQ",Rep_TMP & "\\WPACXRQ.tmp")
Call BvpEnv("PACS94","PAC7QR",Rep_TMP & "\\WPAC7QR.tmp")
Call RunCmdLog ("BVPACS94")
Call Err_Cod(Return , 0 , "PACS94")

Call Msg_Log (Array("1022" , "PACS96"))
'-----
Call BvpEnv("PACS96","PAC7GY",Rep_TMP & "\\WGY.tmp")
WshEnv("PAC7CS") = BVP_MibaCs
WshEnv("PAC7PM") = BVP_MibaFi
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACS96","PAC7TA",Rep_TMP & "\\WTA.tmp")
Call BvpEnv("PACS96","PAC7MM",Rep_USR & "\\INPUTMIMA.txt")
Call BvpEnv("PACS96","PAC7MI",Rep_USR & "\\MIA1.txt")
Call BvpEnv("PACS96","PAC7MO",Rep_USR & "\\MIA2.txt")
Call BvpEnv("PACS96","PAC7MC",Rep_USR & "\\MIA3.txt")
Call BvpEnv("PACS96","PAC7MS",Rep_USR & "\\MIA4.txt")
Call RunCmdLog ("BVPACS96")
Call Err_Cod(Return , 0 , "PACS96")

Call Msg_Log (Array("1022" , "PACS97"))
'-----
Call BvpEnv("PACS97","PAC7AT",Rep_TMP & "\\WAT.tmp")
Call BvpEnv("PACS97","PAC7TA",Rep_TMP & "\\WTA.tmp")
Call RunCmdLog ("BVPACS97")
Call Err_Cod(Return , 0 , "PACS97")

```

```

Call Msg_Log (Array("1022" , "PACS91"))
'-----
Call BvpEnv("PACS91","PAC7AT",Rep_TMP & "\WAT.tmp")
Call BvpEnv("PACS91","PAC7TI",Rep_TMP & "\WTI.tmp")
Call RunCmdLog ("BVPACS91")
Call Err_Cod(Return , 0 , "PACS91")

Call Msg_Log (Array("1022" , "PACS98"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
CALL BVPENV("PACS98","PAC7CP",REP_TMP & "\WCP.TMP")
Call BvpEnv("PACS98","PAC7TI",Rep_TMP & "\WTI.tmp")
Call BvpEnv("PACS98","PAC7RQ",Rep_TMP & "\WPAC7QR.tmp")
Call BvpEnv("PACS98","PAC7TG",Rep_USR & "\MIBJ.ta2")
Call RunCmdLog ("BVPACS98")
Call Err_Cod(Return , 0 , "PACS98")

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

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

</script>
</job>

```

---

## MIMA - Migration Macros

### MIMA - Présentation générale

Cette procédure permet, simultanément, de générer et de taguer une ou plusieurs Macrostructures à la fois.

En entrée il faut :

- soit saisir une ligne '\*' (code utilisateur, mot de passe et code bibliothèque) ainsi qu'une ou plusieurs lignes de commande, pour chaque Macrostructure à générer,
- soit utiliser le fichier INPUTMIMA issu des procédures MIBA ou MIBR.

Cette commande s'appelle GCM et permet la saisie de plusieurs paramètres :

- VARIAN=x, représente la variante de la fiche de la Macrostructure. Si ce paramètre n'est pas renseigné, la Macrostructure sera générée avec la variante de sa fiche (cette valeur étant à 'N' pour la plupart des Macrostructures, il est préférable de toujours renseigner ce paramètre).
- LANGEN=x, représente le langage de génération des programmes sur la fiche de la bibliothèque. Par défaut, la Macrostructure sera générée avec la valeur de cette option sur la fiche de la bibliothèque.

Attention : toutes les demandes de génération de Macrostructures pour une même ligne '\*' doivent avoir la même valeur pour le dernier paramètre.

### Les Macrostructures prioritaires : commandes GMO

Le fichier OUTMIMA issu des procédures MIBR et MIBA contient également les commandes GMO. Elles concernent les macrostructures prioritaires créées à partir des écrans qui ont des lignes de positionnements relatifs spécifiques (\*A, \*P, \*R, \*C).

Ces Macrostructures sur 6 caractères (code écran) vont être générées sur 8 caractères, suffixées par 'SP' (COECRASP) et ne vont contenir que les lignes de type N avec positionnement relatif.

Attention : il est interdit de modifier les lignes de commandes GMO.

### **MIMA - Entrées Utilisateur**

Il faut une ligne '\*' avec code utilisateur, mot de passe et code 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 historisée

Il faut ensuite une ligne pour la commande GCM avec des paramètres à renseigner (facultatif).

Pos.	Lon.	Valeur	Signification
2	1	'Z'	Code ligne
3	2	'90'	Critère ordre édition (1)
5	4	'GCM '	Code commande (1)
9	6	cccccc	Code Macrostructure (1)
15	1	'C'	Indicateur sélection (1)
16	1	'1'	Option à éditer (1)
28	1		Présence d'une ligne suite
		' '	Pas de ligne suite
		'*'	Présence d'une ligne suite

Pos.	Lon.	Valeur	Signification
31	50		Paramètre de la commande
		VARIAN=x	Pour affecter la valeur x à la variante
		LANGEN=x	Pour affecter la valeur x au langage de génération de la bibliothèque

(1) Rubriques à ne renseigner que sur la première ligne, pas sur les lignes suite.

Cette procédure doit également permettre de transformer des caractères spéciaux non reconnus dans un code de Macrostructure par un autre caractère ou des mots interdits (ex: mots réservés windows) par un autre code. La table de correspondance est représentée par le fichier PAC7CS en entrée du programme BVPACP87 (fichier optionnel, fourni par le client).

Ce fichier contient un enregistrement d'une longueur de 20 caractères, constitué de 10 postes de 2 caractères : le premier représente le caractère à remplacer, le deuxième est le caractère remplaçant.

## MIMA - Description des étapes

Commandes Edition/génération : BVPACA10

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	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
PAC7ME	Entrée utilisateur	Entrée	Mouvements utilisateur
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements de mise à jour
PAC7MG	Rép. tmp : WMG	Sortie	Commandes Edition/Génération

Préparation spécifique GCM : BVPACA30

Code	Nom physique	Type	Libellé
PAC7MG	Rép. tmp : WMG	Entrée	Commandes Edition/Génération
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
PAC7JG	Rép. tmp : WKC	Sortie	Commande de l'extracteur
PAC7KU	Rép. tmp : WKU	Sortie	Fichier état sortie

### Génération des Macrostructures : BVPACMM

Code	Nom physique	Type	Libellé
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGR	Base Admin - Rép. base : AN	Entrée	Données de la Base Administration
PACGGN	Base Admin - Rép. base : AR	Entrée	Index 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
PAC7SC	Système - Rép. skel : SC	Entrée	Squelette de génération
PAC7CS	Rép. user : MibaSpecChar	Entrée	Fichier de correspondance des caractères spéciaux
PAC7EP	Rép. tmp : WEP	Sortie	Fichier de travail
PAC7GI	Rép. tmp : WGI	Sortie	Fichier de travail
PAC7GP	Rép. tmp : WGP	Sortie	Fichier de travail
PAC7JG	Rép. tmp : WJG	Sortie	Fichier de travail
PAC7KC	Rép. tmp : WKC	Sortie	Fichier de travail
PAC7KP	Rép. tmp : WKP	Sortie	Fichier de travail
PAC7RQ	Rép. tmp : WRQ	Sortie	Fichier de travail
PAC7AT	Rép. tmp : WAT	Sortie	Fichier de travail

Code	Nom physique	Type	Libellé
PAC7TA	Rép. tmp : WTA	Sortie	Fichier de travail
PAC7TT	Rép. user : MIMA	Sortie	Fichier généré des Macrostructures
PAC7WA	Rép. tmp : WWA	Sortie	Fichier de travail
PAC7WC	Rép. tmp : WWC	Sortie	Fichier de travail
PAC7W1	Rép. tmp : WW1	Sortie	Fichier de travail
PAC7W2	Rép. tmp : WW2	Sortie	Fichier de travail
PAC7W3	Rép. tmp : WW3	Sortie	Fichier de travail
PAC7W4	Rép. tmp : WW4	Sortie	Fichier de travail
PAC7W5	Rép. tmp : WW5	Sortie	Fichier de travail
PAC7IA	Rép. user : MIMAIID	Etat	Edition générale de l'enchaînement des programmes

## MIMA - Script d'exécution

```

|-----|
| VISUALAGE PACBASE |
|-----|
| MACRO-STRUCTURES MIGRATION |
|-----|
|
|<job id=MIMA>
|
|<script language="VBScript">
| Dim MyProc
| MyProc = "MIMA"
|</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
|
| Call Msg_Log (Array("1022" , "PACA10"))
| '-----'
| 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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA30"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PACA30","PAC7KU",Rep_TMP & "\WKU.tmp")
CALL BVPENV("PACA30","PAC7JG",REP_TMP & "\WKC.TMP")
Call BvpEnv("PACA30","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA30")
Call Err_Cod(Return , 0 , "PACA30")

Call Msg_Log (Array("1022" , "PACMM"))
'-----
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("PAC7SC") = Rep_SKEL & "\SC" & CodLang
Call BvpEnv("PACMM","PAC7EP",Rep_TMP & "\WEP.tmp")
Call BvpEnv("PACMM","PAC7GI",Rep_TMP & "\WGI.tmp")
Call BvpEnv("PACMM","PAC7GP",Rep_TMP & "\WGP.tmp")
Call BvpEnv("PACMM","PAC7JG",Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACMM","PAC7KC",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACMM","PAC7KP",Rep_TMP & "\WKP.tmp")
Call BvpEnv("PACMM","PAC7RQ",Rep_TMP & "\WRQ.tmp")
Call BvpEnv("PACMM","PAC7AT",Rep_TMP & "\WAT.tmp")
Call BvpEnv("PACMM","PAC7TA",Rep_TMP & "\WTA.tmp")
Call BvpEnv("PACMM","PAC7TT",Rep_USR & "\MIMA.cb1gen")
Call BvpEnv("PACMM","PAC7WA",Rep_TMP & "\WWA.tmp")
Call BvpEnv("PACMM","PAC7WC",Rep_TMP & "\WWC.tmp")
Call BvpEnv("PACMM","PAC7W1",Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACMM","PAC7W2",Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACMM","PAC7W3",Rep_TMP & "\WW3.tmp")
Call BvpEnv("PACMM","PAC7W4",Rep_TMP & "\WW4.tmp")
Call BvpEnv("PACMM","PAC7W5",Rep_TMP & "\WW5.tmp")
WshEnv("PAC7CS") = BVP_MibaCs
Call BvpEnv("PACMM","PAC7IA",Rep_USR & "\MIMAI.txt")
Call RunCmdLog ("BVPACMM")
Call Err_Cod(Return , 0 , "PACMM")

```

```

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

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

</script>
</job>

```

---

## MITH - Migration Thesaurus

### MITH - Présentation générale

#### Principe

Cette procédure est l'utilitaire de migration batch du thesaurus. Suite à l'extraction des entités en inter-bibliothèque dans le référentiel, un fichier mis en forme au format XML et contenant le thesaurus, sera exporté dans la SDP.

### MITH - Entrées Utilisateur

Ce sont les entrées de la procédure PACX avec les valeurs spécifiques suivantes :

Une ligne '\*' telle que :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	***	extraction en inter-bibliothèque
22	4	nnnn	Numéro de session (blanc=courante)
26	1	'T'	Etat de la session si session historisée
29	4	EXLI	Code de l'extracteur
34	1	'1'	Formatage pour UPDP (PAF)
55	1	'1'	Top plan de convergence

L'extraction EXLI s'exécute en inter-bibliothèque, puisque les entités du thesaurus sont créées en inter-bibliothèque dans le référentiel.

Le fichier GY issu du PACX est ensuite filtré, les enregistrements du thesaurus sont sélectionnés et un nouveau fichier au format XML est créé .



## MITH - 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
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	Entrées utilisateur
PAC7MA	NUL	Entrée	Fichier travail
PAC7ES	NUL	Entrée	Fichier travail
PAC7BM	Rép. tmp : WBM	Entrée/Sortie	Entrées utilisateur
PAC7MM	Rép. tmp : WMM	Entrée/Sortie	Fichier travail
PAC7MJ	Rép. tmp : WMJ	Entrée/Sortie	Fichier travail
PAC7TE	Rép. tmp : WTE	Entrée/Sortie	Fichier travail
PAC7RE	Rép. tmp : WRE	Entrée/Sortie	Fichier travail
PAC7RM	Rép. tmp : WRM	Entrée/Sortie	Fichier travail
PAC7WD	Rép. tmp : WWD	Entrée/Sortie	Mouvements extraits
SYSEXT	Rép. tmp : WSY	Entrée/Sortie	Fichier de travail
PAC7MV	Rép. tmp : WMV	Sortie	Fichier travail
PAC7MR	Rép. tmp : WMR	Sortie	Fichier travail
PAC7MX	Rép. tmp : WMX	Sortie	Entités non extraites (PACX)
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements extraits pour UPDP
PAC7RQ	NUL	Sortie	Correspondance Entités/Bibliothèque origine

Code	Nom physique	Type	Libellé
PAC7TD	Rép. tmp : WTD	Sortie	Fichier travail
PAC7UE	Rép. tmp : WUE	Sortie	Fichier travail
PAC7CP	NULCP	Sortie	Fichier travail
PAC7IA	Rép. user : PACXIA	Etat	Edition générale de l'enchaînement des programmes
PAC7DD	Rép. user : PACXDD	Etat	Edition des anomalies sur mouvements en entrée
PAC7ED	Rép. tmp : WED	Etat	Compte-rendu d'extractions
PAC7EE	Rép. user : PACXEE	Etat	Compte-rendu d'extractions
PAC7EG	Rép. tmp : WEG	Etat	Compte-rendu d'extractions
PAC7EM	Rép. tmp : WEM	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. tmp : WEU	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 (précisée dans PAC7EZ)
- 8 : Erreur dans la ligne '\*' (précisée dans PAC7DD)

Sélection des mouvements du thesaurus et formatage d'un fichier XML :  
BVPACTH1

Code	Nom physique	Type	Libellé
PAC7GY	Rép. tmp : WGY	Entrée	Fichier issu de l'extraction PACX
PAC7RQ	Rép. user : MITHFILE	Sortie	Fichier du thesaurus au format XML

## MITH - Script d'exécution

```
'-----
' VISUALAGE PACBASE
'

'-----
' THESAURUS MIGRATION
'-----
'
<job id=MITH>

<script language="VBScript">
Dim MyProc
MyProc = "MITH"
</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("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", "PAC7RQ", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7RM", Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX", "PAC7MA", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7ES", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7CP", Rep_TMP & "\NULCP.tmp")
Call BvpEnv("PACX", "PAC7UE", Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACX", "PAC7TD", Rep_TMP & "\WTD.tmp")
Call BvpEnv("PACX", "PAC7MV", Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACX", "PAC7MR", Rep_TMP & "\WMR.tmp")
Call BvpEnv("PACX", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("PACX", "PAC7IA", Rep_USR & "\PACXIA.txt")
Call BvpEnv("PACX", "PAC7DD", Rep_USR & "\PACXDD.txt")
Call BvpEnv("PACX", "PAC7ED", Rep_TMP & "\WED.tmp")
Call BvpEnv("PACX", "PAC7EE", Rep_USR & "\PACXEE.txt")
Call BvpEnv("PACX", "PAC7EG", Rep_TMP & "\WEG.tmp")
```

```

Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\WEU.tmp")
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 , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACTH1"))
'-----
Call BvpEnv("PACTH1","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACTH1","PAC7RQ",Rep_USR & "\MITHFILE.txt")
Call RunCmdLog ("BVPACTH1")
Call Err_Cod(Return , 0 , "PACTH1")

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

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

</script>
</job>

```

---

## MIUS - Migration Utilisateurs

### MIUS - Présentation générale

#### Principe

Transformation des users en create user.

Cette procédure est l'utilitaire qui permet de transformer des Users en Create users. Suite à l'extraction des utilisateurs déclarés dans l'administration, un fichier contenant les caractéristiques de chaque utilisateur est constitué. Ce fichier correspond à une liste d'ID utilisateur, d'adresses électroniques, de noms, d'ID licence et de niveau administration, séparés par des virgules. Ce fichier est une entrée à l'outil repotools de RTC.

### MIUS - Entrées Utilisateur

Ce sont les entrées de la procédure PACX avec les valeurs spécifiques suivantes :

Une ligne '\*' telle que :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	***	Extraction en inter-bibliothèque
22	4		Numéro de session (blanc=courante)
26	1		Etat de la session
29	4	EXTR	Code de l'extracteur
34	1	'1'	Formatage pour UPDP (PAF)
55	1	'1'	Top plan de convergence

Une ligne de commande :

pos.	Lon.	Valeur	Signification
2	1	'W'	Code ligne
3	1	'1'	Numéro de ligne
4	2	'EX'	
6	1		Code de sélection de bibliothèque :
		'C'	Bibliothèque et ses 'centrales'
7	33	1USE	Méta-entité des utilisateurs
40	4		Type d'extraction (1) :
		'ALL '	Entité et entités utilisées

L'extraction EXTR s'exécute en inter-bibliothèque, dans la base d'administration.

Le fichier GY issu du PACX est ensuite filtré, les enregistrements des utilisateurs sont sélectionnés et un nouveau fichier, entrée à l'outil repotools de RTC, est créé.

Remarque :

Les virgules sont des délimiteurs de champs. Quand elles sont au sein d'un champ textuel, elles sont remplacées par des points-virgules afin d'être compatibles avec RTC.

## MIUS - 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	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PAC7AY	Base Admin - Rép. base : AY	Entrée	Données extension de la Base Administration
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	Fichier des utilisateurs de la Base Administration
PAC7PJ	Base Admin - Rép. save : PJ	Entrée	Mouvements sélectionnés sur le journal
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7BM	Rép. tmp : WBM	Entrée/Sortie	Entrées utilisateur
PAC7MM	Rép. tmp : WMM	Entrée/Sortie	
SYSEXT	Rép. tmp : SY	Entrée/Sortie	Fichier de travail
PAC7MJ	Rép. tmp : WMJ	Entrée/Sortie	
PAC7TE	Rép. tmp : WTE	Entrée/Sortie	
PAC7RE	Rép. tmp : WRE	Entrée/Sortie	
PAC7RM	Rép. tmp : WRM	Entrée/Sortie	
PAC7RQ	Rép. tmp : WRQ	Entrée/Sortie	Fichier de travail
PAC7WD	Rép. tmp : WWD	Entrée/Sortie	Mouvements extraits
PAC7MV	Rép. tmp : WMV	Entrée/Sortie	
PAC7MR	Rép. tmp : WMR	Entrée/Sortie	
PAC7TD	Rép. tmp : WTD	Entrée/Sortie	
PAC7MX	Rép. tmp : WMX	Entrée/Sortie	
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements extraits pour UPGP
PAC7UE	Rép. tmp : WUE	Entrée/Sortie	

Code	Nom physique	Type	Libellé
PAC7IA	Rép. user : PACXIA	Etat	Edition générale de l'enchaînement des programmes
PAC7DD	Rép. user : PACXDD	Etat	Edition des anomalies sur mouvements en entrée
PAC7ED	Rép. tmp : WED	Etat	Compte-rendu d'extraction
PAC7EE	Rép. user : PACXEE	Etat	Compte-rendu d'extraction
PAC7EG	Rép. tmp : WEG	Etat	Compte-rendu d'extraction
PAC7EM	Rép. tmp : WEM	Etat	Compte-rendu d'extraction
PAC7EP	Rép. user : PACXEP	Etat	Compte-rendu d'extraction
PAC7EQ	Rép. user : PACXEQ	Etat	Compte-rendu d'extraction
PAC7ES	NUL		
PAC7EU	Rép. tmp : WEU	Etat	Compte-rendu d'extraction
PAC7EZ	Rép. user : PACXEZ	Etat	Compte-rendu d'extraction
PAC7MA	NUL		
PAC7CP	NULCP		

Codes retour :

- 0 : Pas d'erreur
- 4 : Erreur dans les entrées utilisateur (précisée dans PAC7EE) ou dans l'extraction
- 8 : Erreur dans la ligne '\*' (précisée dans PAC7DD)

Transformation des users en create user : BVPLTYAU

Code	Nom physique	Type	Libellé
PAC7GY	Rép. tmp : WGY	Entrée	Fichier issu de l'extraction PACX
PAC7US	Rép. user : MIUSFILE	Sortie	Fichier des utilisateurs : entrée outil repotools

## MIUS - Script d'exécution

```

| -----
| VISUALAGE PACBASE
| -----
| USERS MIGRATION
| -----
|

```

```

<job id=MIUS>

<script language="VBScript">
Dim MyProc
MyProc = "MIUS"
</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_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7PJ") = Rep_ASAVE & "\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","PAC7RQ",Rep_TMP & "\WRQ.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\NULCP.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\WTD.tmp")
Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\WEU.tmp")
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 , 4 , "PACX")

Call Msg_Log (Array("1022" , "PLTYAU"))
'-----
Call BvpEnv("PLTYAU","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PLTYAU","PAC7US",Rep_USR & "\MIUSFILE.txt")
Call RunCmdLog ("BVPLTYAU")
Call Err_Cod(Return , 0 , "PLTYAU")

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

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

</script>
</job>

```

---

## MILA - Migration libellés d'erreur administration

### MILA - Présentation générale

#### Principe

Extraction des libellés d'erreur personnalisés de l'administration.

Cette procédure est l'utilitaire qui permet de formater un fichier des libellés d'erreur. Ce fichier contient tous les libellés d'erreur standard. Une extraction des libellés d'erreur personnalisés dans la base administration permet d'overrider les libellés standard correspondant. Le fichier en sortie contient tous les libellés d'erreur pour tous langages, chaque série étant séparée par une ligne de commentaire indiquant la langue.

### MILA - Entrées Utilisateur

Ce sont les entrées de la procédure PACX avec les valeurs spécifiques suivantes :

Une ligne '\*' telle que :

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	***	Extraction en inter-bibliothèque
22	4		Numéro de session (blanc=courante)
26	1		Etat de la session
29	4	EXTR	Code de l'extracteur
34	1	'1'	Formatage pour UPDP (PAF)
55	1	'1'	Top plan de convergence

Une ligne de commande :

Pos.	Lon.	Valeur	Signification
2	1	'W'	Code ligne
3	1	'1'	Numéro de ligne
4	2	'EX'	
6	1		Code de sélection de bibliothèque :
		'C'	Bibliothèque et ses 'centrales'
7	33	YATTABLES	Entité utilisateur des tables de l'administration

L'extraction EXTR s'exécute en inter-bibliothèque, dans la base d'administration.

Le fichier GY issu du PACX est ensuite filtré, les enregistrements des messages d'erreur sont sélectionnés (descriptif -D02) et remis en forme dans un fichier PAC7LI.

## MILA - 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	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PAC7AY	Base Admin - Rép. base : AY	Entrée	Données extension de la Base Administration

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	Fichier des utilisateurs de la Base Administration
PAC7PJ	Base Admin - Rép. save : PJ	Entrée	Mouvements sélectionnés sur le journal
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7BM	Rép. tmp : WBM	Entrée/Sortie	Entrées utilisateur
PAC7MM	Rép. tmp : WMM	Entrée/Sortie	
SYSEXT	Rép. tmp : SY	Entrée/Sortie	Fichier de travail
PAC7MJ	Rép. tmp : WMJ	Entrée/Sortie	
PAC7TE	Rép. tmp : WTE	Entrée/Sortie	
PAC7RE	Rép. tmp : WRE	Entrée/Sortie	
PAC7RM	Rép. tmp : WRM	Entrée/Sortie	
PAC7RQ	Rép. tmp : WRQ	Entrée/Sortie	Fichier de travail
PAC7WD	Rép. tmp : WWD	Entrée/Sortie	Mouvements extraits
PAC7MV	Rép. tmp : WMV	Entrée/Sortie	
PAC7MR	Rép. tmp : WMR	Entrée/Sortie	
PAC7TD	Rép. tmp : WTD	Entrée/Sortie	
PAC7MX	Rép. tmp : WMX	Entrée/Sortie	
PAC7GY	Rép. tmp : WGY	Sortie	Mouvements extraits pour UPGP
PAC7UE	Rép. tmp : WUE	Entrée/Sortie	
PAC7IA	Rép. user : PACXIA	Etat	Edition générale de l'enchaînement des programmes
PAC7DD	Rép. user : PACXDD	Etat	Edition des anomalies sur mouvements en entrée
PAC7ED	Rép. tmp : WED	Etat	Compte-rendu d'extraction
PAC7EE	Rép. user : PACXEE	Etat	Compte-rendu d'extraction
PAC7EG	Rép. tmp : WEG	Etat	Compte-rendu d'extraction
PAC7EM	Rép. tmp : WEM	Etat	Compte-rendu d'extraction
PAC7EP	Rép. user : PACXEP	Etat	Compte-rendu d'extraction
PAC7EQ	Rép. user : PACXEQ	Etat	Compte-rendu d'extraction

Code	Nom physique	Type	Libellé
PAC7ES	NUL		
PAC7EU	Rép. tmp : WEU	Etat	Compte-rendu d'extraction
PAC7EZ	Rép. user : PACXEZ	Etat	Compte-rendu d'extraction
PAC7MA	NUL		
PAC7CP	NULCP		

Codes retour :

- 0 : Pas d'erreur
- 4 : Erreur dans les entrées utilisateur (précisée dans PAC7EE) ou dans l'extraction
- 8 : Erreur dans la ligne '\*' (précisée dans PAC7DD)

Mise en forme des libellés d'erreur : BVPLTYAT

Code	Nom physique	Type	Libellé
PAC7GY	Rép. tmp : WGY	Entrée	Fichier issu de l'extraction PACX
PAC7LI	Rép. user : MILAFILE	Sortie	Fichier des utilisateurs : entrée outil repotools

## MILA - Script d'exécution

```

| -----
| VISUALAGE PACBASE
| -----
| ERROR LABEL MIGRATION
| -----
|
<job id=MILA>

<script language="VBScript">
Dim MyProc
MyProc = "MILA"
</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_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7PJ") = Rep_ASAVE & "\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", "PAC7RQ", Rep_TMP & "\WRQ.tmp")
Call BvpEnv("PACX", "PAC7RM", Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX", "PAC7MA", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7CP", Rep_TMP & "\NULCP.tmp")
Call BvpEnv("PACX", "PAC7ES", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7UE", Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACX", "PAC7TD", Rep_TMP & "\WTD.tmp")
Call BvpEnv("PACX", "PAC7MV", Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACX", "PAC7MR", Rep_TMP & "\WMR.tmp")
Call BvpEnv("PACX", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("PACX", "PAC7IA", Rep_USR & "\PACXIA.txt")
Call BvpEnv("PACX", "PAC7DD", Rep_USR & "\PACXDD.txt")
Call BvpEnv("PACX", "PAC7ED", Rep_TMP & "\WED.tmp")
Call BvpEnv("PACX", "PAC7EE", Rep_USR & "\PACXEE.txt")
Call BvpEnv("PACX", "PAC7EG", Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACX", "PAC7EM", Rep_TMP & "\WEM.tmp")
Call BvpEnv("PACX", "PAC7EP", Rep_USR & "\PACXEP.txt")
Call BvpEnv("PACX", "PAC7EQ", Rep_USR & "\PACXEQ.txt")
Call BvpEnv("PACX", "PAC7EU", Rep_TMP & "\WEU.tmp")
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 , 4 , "PACX")

Call Msg_Log (Array("1022" , "PLTYAT"))
'-----
Call BvpEnv("PLTYAT", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PLTYAT", "PAC7LI", Rep_USR & "\MILAFIle.txt")
Call RunCmdLog ("BVPLTYAT")
Call Err_Cod(Return , 0 , "PLTYAT")

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

```

```

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

</script>
</job>

```

---

## MERT - Transformation des entités Merise

### MERT - Présentation générale

#### Principe

Cette procédure permet d'extraire les entités de la méthode Merise ainsi que les entités utilisateurs PacDesign associées à la méthode Merise et de les transformer en Entités Utilisateur qui pourront être exportées dans la SDP. La procédure fournit en sortie un fichier de mouvements de mise à jour à utiliser en entrée de la procédure UPDP.

#### Prérequis

La base de travail Pacbase doit comporter les entités de la méthode Merise.

### MERT - Entrées Utilisateur

Une ligne '\*' telle que :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code ligne
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	***	extraction en inter-bibliothèque
22	4	nnnn	Numéro de session (blanc=courante)
26	1	'T'	Etat de la session si session historisée

Dans cette procédure, seules sont traitées les lignes suivantes :

- Fiche des entités Merise (Objet, Relation et CIF).
- Lignes de commentaires -GC.
- Lignes d'appel de propriétés -CE.
- Lignes de commentaires -GC sous -CE.
- Lignes d'appel d'objets Merise -CM.
- Lignes de commentaires -GC sous -CM.

Ne sont pas traitées les lignes suivantes :

- Les appels de Formats Guide dans les -GC.
- Les utilisations des entités Merise.

#### Transformation des entités Merise en Entités Utilisateur

- Objet Merise: transformé en Entité Utilisateur de code d'appel \$9O.
- Relation Merise: transformée en Entité Utilisateur de code d'appel \$9R.
- CIF Merise : transformée en Entité Utilisateur de code d'appel \$9C.

### **MERT - Description des étapes**

#### Définition des nouvelles Méta-Entités.

- Création des nouvelles Méta-Entités en exécutant la procédure VINS avec le fichier BVPMERF1.
- Mise à niveau des Entités utilisateur Pacdesign associées à la méthode Merise au nouveau format en exécutant la procédure VINS avec le fichier BVPMERF2.
- Exécution de la procédure MERT.
- Exécution de la procédure UPDP pour mettre à jour la base Pacbase avec les nouvelles Entités Utilisateur créés lors de la procédure MERT.

Les fichiers BVPMERF1 et BVPMERF2 sont disponibles auprès du support VisualAge Pacbase.

#### Extraction des entités : PAFMER

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 PAF
PAC7UR	Rép. tmp : WUR	Entrée/Sortie	Fichier de travail
PAC7EJ	Rép. user : MERT1EJ	Etat	Listes contrôle
PAC7ET	Rép. user : MERT1ET	Etat	Compte-rendu accès PAF

Code	Nom physique	Type	Libellé
PAC7DD	Rép. user : MERT1DD	Etat	Listes anomalies

### Ecriture des Entités : BVPTUMER

Code	Nom physique	Type	Libellé
PAC7UR	Rép. tmp : WUR	Entrée	Fichier de travail
PAC7GY	Rép. user: MERTGY	Sortie	Fichier des entités Merise
PAC7EJ	Rép. user : MERT2EJ	Etat	Listes contrôle intégrité
PAC7ET	Rép. user : MERT2ET	Etat	Compte-rendu accès PAF
PAC7DD	Rép. user : MERT2DD	Etat	Listes anomalies

## MERT - Script d'exécution

```

|-----|
| VISUALAGE PACBASE |
|-----|
| - MERISE ENTITY TRANSFORMATION - |
|-----|
|
<job id=MERT>

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

Call Msg_Log (Array("1022" , "PAFMER"))
'-----|
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("PAC7MB") = Fic_Input
Call BvpEnv("PAFMER", "PAC7UR", Rep_TMP & "\WUR.tmp")
Call BvpEnv("PAFMER", "SYSPAF", Rep_TMP & "\SYSPAF.tmp")
Call BvpEnv("PAFMER", "PAC7EJ", Rep_USR & "\MERT1EJ.txt")
Call BvpEnv("PAFMER", "PAC7ET", Rep_USR & "\MERT1ET.txt")
Call BvpEnv("PAFMER", "PAC7DD", Rep_USR & "\MERT1DD.txt")
Call RunCmdLog ("BVPAFMER")
Call Err_Cod(Return , 0 , "PAFMER")

Call Msg_Log (Array("1022" , "PTUMER"))
'-----
Call BvpEnv("PTUMER", "PAC7UR", Rep_TMP & "\WUR.tmp")
Call BvpEnv("PTUMER", "PAC7GY", Rep_USR & "\MERTGY.txt")
Call BvpEnv("PTUMER", "PAC7EJ", Rep_USR & "\MERT2EJ.txt")
Call BvpEnv("PTUMER", "PAC7ET", Rep_USR & "\MERT2ET.txt")
Call BvpEnv("PTUMER", "PAC7DD", Rep_USR & "\MERT2DD.txt")
Call RunCmdLog ("BVPTUMER")
Call Err_Cod(Return , 0 , "PTUMER")

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

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

</script>
</job>

```

---

## MIAX - Cartes de contrôles optionnelles

Afin de pouvoir éclater les sources des programmes, écrans, serveurs ou clauses "COPY" générés, dans des fichiers distincts et dans des répertoires spécifiés par l'utilisateur, il est nécessaire de définir des commandes dans le Gestionnaire de Génération.

### Définition des lignes de commande AVANT.

Ces lignes contiennent, 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:

Valeur	Signification
*++++*	Délimiteur, obligatoirement en colonnes 1 à 7
PGMCOD	Nom du fichier à produire

Valeur	Signification
ext	Extension, sur 3 caractères maximum

Les informations sont séparées par un blanc.

Exemple :

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

La deuxième ligne de commande AVANT doit contenir :

Valeur	Signification
*&&&&&*	Délimiteur, obligatoirement en colonnes 1 à 7
chemin	Répertoire d'écriture des fichiers

Les informations sont séparées par un blanc.

Exemple :

```
&&&&& /vapak/cobol
```

## MIA1 - Génération programmes

### MIA1 - Présentation générale

Cette procédure permet de générer les programmes dans deux fichiers contenant respectivement le généré Cobol et le généré Cobol de contrôle.

Elle effectue la génération des micro-patterns 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).

### MIA1 - Entrées Utilisateur

Il faut une ligne '\*' avec code utilisateur, mot de passe et code 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

Il faut ensuite une ligne de commande de type GCP telle que décrite dans les entrées de la procédure GPRT pour chaque programme à générer.

## MIA1 - Description des étapes

Commandes Edition/génération : BVPACA10

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	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
PAC7ME	Entrée utilisateur	Entrée	Mouvements utilisateur
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements de mise à jour
PAC7MG	Rép. tmp : WMG	Sortie	Commandes Edition/Génération

Ventilation des commandes : BVPACA20

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
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
PACGGK	Base Admin - Rép. base : GK	Entrée/ Sortie	Droits de génération

Code	Nom physique	Type	Libellé
PAC7JG	Rép. tmp : WJG	Sortie	Commande de l'extracteur
PAC7KF	Rép. tmp : WKF	Entrée/ Sortie	Fichier de travail
PAC7MG	Rép. tmp : WMG	Entrée	Commandes Edition/Génération
PAC7W1	NUL	Entrée/ Sortie	Fichier de travail
PAC7KU	Rép. tmp : WKU	Sortie	Fichier état sortie

### Génération des programmes : BVPACMM1

Code	Nom physique	Type	Libellé
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
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
PAC7SC	Système - Rép. skel : SC	Entrée	Squelette de génération
PAC7EP	Rép. tmp : WEP	Sortie	Fichier de travail
PAC7GI	Rép. tmp : NUL	Sortie	Fichier de travail
PAC7GP	Rép. tmp : WGP	Sortie	Fichier de travail
PAC7JG	Rép. tmp : WJG	Sortie	Fichier de travail
PAC7KC	Rép. tmp : WKC	Sortie	Fichier de travail
PAC7KP	Rép. tmp : WKP	Sortie	Fichier de travail
PAC7G1	Rép. tmp : WG1	Sortie	Fichier de travail
PAC7G2	Rép. user: COBOL_MIA1	Sortie	Fichier Cobol C9
PAC7G3	Rép. user: CONTROL_COBOL_MIA1	Sortie	Fichier Cobol de contrôle
PAC7WB	Rép. tmp : WWB	Sortie	Fichier de travail

Code	Nom physique	Type	Libellé
PAC7WC	Rép. tmp : WWC	Sortie	Fichier de travail
PAC7WM	Rép. tmp : WWM	Sortie	Fichier de travail
PAC7WP	Rép. tmp : WWP	Sortie	Fichier de travail
PAC7W1	Rép. tmp : WW1	Sortie	Fichier de travail
PAC7W2	Rép. tmp : WW2	Sortie	Fichier de travail
PAC7W3	Rép. tmp : WW3	Sortie	Fichier de travail
PAC7W4	Rép. tmp : WW4	Sortie	Fichier de travail
PAC7W5	Rép. tmp : WW5	Sortie	Fichier de travail
PAC7W6	Rép. tmp : WW6	Sortie	Fichier de travail
PAC7CA	Rép. tmp : WCA	Sortie	Fichier de travail
PAC7CM	Rép. tmp : WCM	Sortie	Fichier de travail
PAC7IA	Rép. user: MIA1IA	Etat	Edition générale de l'enchaînement des programmes

## MIA1 - Script d'exécution

```

| -----
| VISUALAGE PACBASE
|
| -----
| PROGRAM GENERATION
|
| -----
|
<job id=MIA1>

<script language="VBScript">
Dim MyProc
MyProc = "MIA1"
</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

Call Msg_Log (Array("1022" , "PACA10"))

```

```

'-----
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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA20"))
'-----
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("PACGGK") = Rep_ABASE & "\GK"
Call BvpEnv("PACA20","PAC7JG",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACA20","PAC7KF",Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACA20","PAC7KU",Rep_TMP & "\WКУ.tmp")

Call BvpEnv("PACA20","PAC7MG",Rep_TMP & "\WMG.tmp")
Call BvpEnv("PACA20","PAC7W1",Rep_TMP & "\NUL.tmp")
Call RunCmdLog ("BVPACA20")
Call Err_Cod(Return , 0 , "PACA20")

Call Msg_Log (Array("1022" , "PACMM1"))
'-----
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("PAC7SC") = Rep_SKEL & "\SC" & CodLang
Call BvpEnv("PACMM1","PAC7CA",Rep_TMP & "\WCA.tmp")
Call BvpEnv("PACMM1","PAC7CM",Rep_TMP & "\WCM.tmp")
Call BvpEnv("PACMM1","PAC7EP",Rep_TMP & "\WEP.tmp")
Call BvpEnv("PACMM1","PAC7GI",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACMM1","PAC7GP",Rep_TMP & "\WGP.tmp")
Call BvpEnv("PACMM1","PAC7JG",Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACMM1","PAC7KC",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACMM1","PAC7KP",Rep_TMP & "\WKP.tmp")
Call BvpEnv("PACMM1","PAC7G1",Rep_TMP & "\WG1.tmp")
Call BvpEnv("PACMM1","PAC7G2",Rep_USR & "\COBOL_MIA1.txt")
Call BvpEnv("PACMM1","PAC7G3",Rep_USR & _
"\CONTROL_COBOL_MIA1.txt")
Call BvpEnv("PACMM1","PAC7WB",Rep_TMP & "\WВB.tmp")

```

```

Call BvpEnv("PACMM1","PAC7WC",Rep_TMP & "\WWC.tmp")
Call BvpEnv("PACMM1","PAC7WM",Rep_TMP & "\WWM.tmp")
Call BvpEnv("PACMM1","PAC7WP",Rep_TMP & "\WWP.tmp")
Call BvpEnv("PACMM1","PAC7W1",Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACMM1","PAC7W2",Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACMM1","PAC7W3",Rep_TMP & "\WW3.tmp")
Call BvpEnv("PACMM1","PAC7W4",Rep_TMP & "\WW4.tmp")
Call BvpEnv("PACMM1","PAC7W5",Rep_TMP & "\WW5.tmp")
Call BvpEnv("PACMM1","PAC7W6",Rep_TMP & "\WW6.tmp")
Call BvpEnv("PACMM1","PAC7IA",Rep_USR & "\MIA1IA.txt")
Call RunCmdLog ("BVPACMM1")
Call Err_Cod(Return , 0 , "PACMM1")

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

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

</script>
</job>

```

---

## MIA2 - Génération écrans

### MIA2 - Présentation générale

Cette procédure permet de générer les écrans dans deux fichiers contenant respectivement le généré Cobol et le généré Cobol de contrôle.

Elle effectue la génération des micro-patterns 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).

### MIA2 - Entrées Utilisateur

Il faut une ligne '\*' avec code utilisateur, mot de passe et code 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

Il faut ensuite une ligne de commande de type GCO telle que décrite dans les entrées de la procédure GPRT pour chaque écran à générer.

## MIA2 - Description des étapes

Commandes Edition/génération : BVPACA10

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	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
PAC7ME	Entrée utilisateur	Entrée	Mouvements utilisateur
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements de mise à jour
PAC7MG	Rép. tmp : WMG	Sortie	Commandes Edition/Génération

Ventilation des commandes : BVPACA20

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
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
PACGGK	Base Admin - Rép. base : GK	Entrée/ Sortie	Droits de génération



Code	Nom physique	Type	Libellé
PAC7JG	Rép. tmp : WJG	Sortie	Commande de l'extracteur
PAC7KF	Rép. tmp : WKF	Entrée/ Sortie	Fichier de travail
PAC7MG	Rép. tmp : WMG	Entrée	Commandes Edition/Génération
PAC7W1	NUL	Entrée/ Sortie	Fichier de travail
PAC7KU	Rép. tmp : WKU	Sortie	Fichier état sortie

### Génération des écrans : BVPACMM2

Code	Nom physique	Type	Libellé
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
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
PAC7SG	Système - Rép. skel : SG	Entrée	Squelette de génération
PAC7EE	Rép. tmp : WEE	Entrée	Fichier de travail
PAC7GE	Rép. tmp : WGE	Sortie	Fichier de travail
PAC7JG	Rép. tmp : WJG	Sortie	Fichier de travail
PAC7KC	Rép. tmp : WKC	Sortie	Fichier de travail
PAC7KE	Rép. tmp : WKE	Sortie	Fichier de travail
PAC7G1	Rép. tmp : WG1	Sortie	Fichier de travail
PAC7G2	Rép. user: COBOL_MIA2	Sortie	Fichier Cobol C9
PAC7G3	Rép. user: CÓNTROL_COBOL_MIA2	Sortie	Fichier Cobol de contrôle
PAC7W1	Rép. tmp : WW1	Sortie	Fichier de travail
PAC7W2	Rép. tmp : WW2	Sortie	Fichier de travail

Code	Nom physique	Type	Libellé
PAC7EZ	Rép. tmp : WEZ	Sortie	Fichier de travail
PACM2W	Rép. tmp : W2W	Sortie	Fichier de travail
PACM2P	Rép. tmp : W2P	Sortie	Fichier de travail
PAC2GW	Rép. tmp : WGW	Sortie	Fichier de travail
PAC2GP	Rép. tmp : WGP	Sortie	Fichier de travail
PAC780	Rép. tmp : W80	Sortie	Fichier de travail
PAC7IA	Rép. user: MIA2IA	Etat	Edition générale de l'enchaînement des programmes

## MIA2 - Script d'exécution

```

|-----
| VISUALAGE PACBASE
|-----
| SCREEN GENERATION
|-----
|
<job id=MIA2>

<script language="VBScript">
Dim MyProc
MyProc = "MIA2"
</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

Call Msg_Log (Array("1022" , "PACA10"))
'-----
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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10", "PAC7MV", Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10", "PAC7MG", Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA20"))
'-----
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("PACGGK") = Rep_ABASE & "\GK"
Call BvpEnv("PACA20", "PAC7JG", Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACA20", "PAC7KF", Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACA20", "PAC7KU", Rep_TMP & "\WKU.tmp")
Call BvpEnv("PACA20", "PAC7MG", Rep_TMP & "\WMG.tmp")
Call BvpEnv("PACA20", "PAC7W1", Rep_TMP & "\NUL.tmp")
Call RunCmdLog ("BVPACA20")
Call Err_Cod(Return , 0 , "PACA20")

Call Msg_Log (Array("1022" , "PACMM2"))
'-----
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("PAC7SG") = Rep_SKEL & "\SG" & CodLang
Call BvpEnv("PACMM2", "PAC7EE", Rep_TMP & "\WEE.tmp")
Call BvpEnv("PACMM2", "PAC7EZ", Rep_TMP & "\WEZ.tmp")
Call BvpEnv("PACMM2", "PAC7GE", Rep_TMP & "\WGE.tmp")
Call BvpEnv("PACMM2", "PAC7JG", Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACMM2", "PAC7KC", Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACMM2", "PAC7KE", Rep_TMP & "\WKE.tmp")
Call BvpEnv("PACMM2", "PAC7G1", Rep_TMP & "\WG1.tmp")
Call BvpEnv("PACMM2", "PAC7G2", Rep_USR & "\COBOL_MIA2.txt")
Call BvpEnv("PACMM2", "PAC7G3", Rep_USR & _
"\CONTROL_COBOL_MIA2.txt")
Call BvpEnv("PACMM2", "PAC7W1", Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACMM2", "PAC7W2", Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACMM2", "PACM2P", Rep_TMP & "\W2P.tmp")
Call BvpEnv("PACMM2", "PACM2W", Rep_TMP & "\W2W.tmp")
Call BvpEnv("PACMM2", "PAC2GP", Rep_TMP & "\WGP.tmp")
Call BvpEnv("PACMM2", "PAC2GW", Rep_TMP & "\WGW.tmp")
Call BvpEnv("PACMM2", "PAC780", Rep_TMP & "\W80.tmp")
Call BvpEnv("PACMM2", "PAC7IA", Rep_USR & "\MIA2IA.txt")
Call RunCmdLog ("BVPACMM2")
Call Err_Cod(Return , 0 , "PACMM2")

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

```

```

Call DeleteFldr (Rep_TMP)

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

</script>
</job>

```

---

## MIA3 - Génération clients

### MIA3 - Présentation générale

Cette procédure permet de générer les clients dans deux fichiers contenant respectivement le généré Cobol et le généré Cobol de contrôle.

Elle effectue la génération des micro-patterns 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).

### MIA3 - Entrées Utilisateur

Il faut une ligne '\*' avec code utilisateur, mot de passe et code 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

Il faut ensuite une ligne de commande de type GGC telle que décrite dans les entrées de la procédure GPRT pour chaque client à générer.

### MIA3 - Description des étapes

Commandes Edition/génération : BVPACA10

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	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
PAC7ME	Entrée utilisateur	Entrée	Mouvements utilisateur
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements de mise à jour
PAC7MG	Rép. tmp : WMG	Sortie	Commandes Edition/Génération

#### Ventilation des commandes : BVPACA20

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
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
PACGGK	Base Admin - Rép. base : GK	Entrée/ Sortie	Droits de génération
PAC7JG	Rép. tmp : WJG	Sortie	Commande de l'extracteur
PAC7KF	Rép. tmp : WKF	Entrée/ Sortie	Fichier de travail
PAC7MG	Rép. tmp : WMG	Entrée	Commandes Edition/Génération
PAC7W1	NUL	Entrée/ Sortie	Fichier de travail
PAC7KU	Rép. tmp : WKU	Sortie	Fichier état sortie

## Génération des clients : BVPACMM3

Code	Nom physique	Type	Libellé
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
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
PAC7SS	Système - Rép. skel : SS	Entrée	Squelette de génération
PAC7EG	Rép. tmp : WEG	Entrée	Fichier de travail
PAC7GG	Rép. tmp : WGG	Sortie	Fichier de travail
PAC7JG	Rép. tmp : WJG	Sortie	Fichier de travail
PAC7KC	Rép. tmp : WKC	Sortie	Fichier de travail
PAC7KG	Rép. tmp : WKG	Sortie	Fichier de travail
PAC7G1	Rép. tmp : WG1	Sortie	Fichier de travail
PAC7G2	Rép. user: COBOL_MIA3	Sortie	Fichier Cobol C9
PAC7G3	Rép. user: CONTROL_COBOL_MIA3	Sortie	Fichier Cobol de contrôle
PAC7W1	Rép. tmp : WW1	Sortie	Fichier de travail
PAC7W2	Rép. tmp : WW2	Sortie	Fichier de travail
PAC7EZ	Rép. tmp : WEZ	Sortie	Fichier de travail
PACM2W	Rép. tmp : W2W	Sortie	Fichier de travail
PACM2P	Rép. tmp : W2P	Sortie	Fichier de travail
PAC2GW	Rép. tmp : WGW	Sortie	Fichier de travail
PAC2GP	Rép. tmp : WGP	Sortie	Fichier de travail
PAC780	Rép. tmp : W80	Sortie	Fichier de travail
PAC7IA	Rép. user: MIA3IA	Etat	Edition générale de l'enchaînement des programmes

## MIA3 - Script d'exécution

```
'-----
' VISUALAGE PACBASE
'-----
' CLIENT GENERATION
'-----
'
<job id=MIA3>

<script language="VBScript">
Dim MyProc
MyProc = "MIA3"
</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

Call Msg_Log (Array("1022" , "PACA10"))
'-----
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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA20"))
'-----
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("PACGGK") = Rep_ABASE & "\GK"
Call BvpEnv("PACA20","PAC7JG",Rep_TMP & "\WKC.tmp")
```

```

Call BvpEnv("PACA20","PAC7KU",Rep_TMP & "\WКУ.tmp")
Call BvpEnv("PACA20","PAC7KF",Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACA20","PAC7MG",Rep_TMP & "\WМG.tmp")
Call BvpEnv("PACA20","PAC7W1",Rep_TMP & "\NUL.tmp")
Call RunCmdLog ("BVPACA20")
Call Err_Cod(Return , 0 , "PACA20")

Call Msg_Log (Array("1022" , "PACMM3"))
'-----
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("PAC7SS") = Rep_SKEL & "\SS" & CodLang
Call BvpEnv("PACMM3","PAC7EG",Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACMM3","PAC7EZ",Rep_TMP & "\WEZ.tmp")
Call BvpEnv("PACMM3","PAC7GG",Rep_TMP & "\WGG.tmp")
Call BvpEnv("PACMM3","PAC7JG",Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACMM3","PAC7KC",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACMM3","PAC7KG",Rep_TMP & "\WKG.tmp")
Call BvpEnv("PACMM3","PAC7G1",Rep_TMP & "\WG1.tmp")
Call BvpEnv("PACMM3","PAC7G2",Rep_USR & "\COBOL_MIA3.txt")
Call BvpEnv("PACMM3","PAC7G3",Rep_USR &
"\CONTROL_COBOL_MIA3.txt")
Call BvpEnv("PACMM3","PAC7W1",Rep_TMP & "\Ww1.tmp")
Call BvpEnv("PACMM3","PAC7W2",Rep_TMP & "\Ww2.tmp")
Call BvpEnv("PACMM3","PACM2P",Rep_TMP & "\W2P.tmp")
Call BvpEnv("PACMM3","PACM2W",Rep_TMP & "\W2W.tmp")
Call BvpEnv("PACMM3","PAC2GP",Rep_TMP & "\WGP.tmp")
Call BvpEnv("PACMM3","PAC2GW",Rep_TMP & "\WGW.tmp")
Call BvpEnv("PACMM3","PAC780",Rep_TMP & "\W80.tmp")
Call BvpEnv("PACMM3","PAC7IA",Rep_USR & "\MIA3IA.txt")
Call RunCmdLog ("BVPACMM3")
Call Err_Cod(Return , 0 , "PACMM3")

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

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

</script>
</job>

```

---

## MIA4 - Génération serveurs

### MIA4 - Présentation générale

Cette procédure permet de générer les serveurs dans deux fichiers contenant respectivement le généré Cobol et le généré Cobol de contrôle.



Elle effectue la génération des micro-patterns 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).

#### MIA4 - Entrées Utilisateur

Il faut une ligne '\*' avec code utilisateur, mot de passe et code 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

Il faut ensuite une ligne de commande de type GGS telle que décrite dans les entrées de la procédure GPRT pour chaque serveur à générer.

#### MIA4 - Description des étapes

Commandes Edition/génération : BVPACA10

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	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
PAC7ME	Entrée utilisateur	Entrée	Mouvements utilisateur
PAC7MV	Rép. tmp : WMV	Sortie	Mouvements de mise à jour
PAC7MG	Rép. tmp : WMG	Sortie	Commandes Edition/Génération

### Ventilation des commandes : BVPACA20

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
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
PACGGK	Base Admin - Rép. base : GK	Entrée/ Sortie	Droits de génération
PAC7JG	Rép. tmp : WJG	Sortie	Commande de l'extracteur
PAC7KF	Rép. tmp : WKF	Entrée/ Sortie	Fichier de travail
PAC7MG	Rép. tmp : WMG	Entrée	Commandes Edition/Génération
PAC7W1	NUL	Entrée/ Sortie	Fichier de travail
PAC7KU	Rép. tmp : WKU	Sortie	Fichier état sortie

### Génération des serveurs : BVPACMM4

Code	Nom physique	Type	Libellé
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
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



```

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

Call Msg_Log (Array("1022" , "PACA10"))
'-----
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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA20"))
'-----
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("PACGGK") = Rep_ABASE & "\GK"
Call BvpEnv("PACA20","PAC7JG",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACA20","PAC7KU",Rep_TMP & "\WКУ.tmp")
Call BvpEnv("PACA20","PAC7KF",Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACA20","PAC7MG",Rep_TMP & "\WMG.tmp")
Call BvpEnv("PACA20","PAC7W1",Rep_TMP & "\NUL.tmp")
Call RunCmdLog ("BVPACA20")
Call Err_Cod(Return , 0 , "PACA20")

Call Msg_Log (Array("1022" , "PACMM4"))
'-----
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("PAC7SS") = Rep_SKEL & "\SS" & CodLang
Call BvpEnv("PACMM4","PAC7EV",Rep_TMP & "\WEV.tmp")
Call BvpEnv("PACMM4","PAC7EZ",Rep_TMP & "\WEZ.tmp")
Call BvpEnv("PACMM4","PAC7GV",Rep_TMP & "\WGV.tmp")
Call BvpEnv("PACMM4","PAC7JG",Rep_TMP & "\WJG.tmp")

```

```

Call BvpEnv("PACMM4","PAC7KC",Rep_TMP & "\\WKC.tmp")
Call BvpEnv("PACMM4","PAC7KV",Rep_TMP & "\\WKV.tmp")
Call BvpEnv("PACMM4","PAC7G1",Rep_TMP & "\\WG1.tmp")
Call BvpEnv("PACMM4","PAC7G2",Rep_USR & "\\COBOL_MIA4.txt")
Call BvpEnv("PACMM4","PAC7G3",Rep_USR & _
 "\\CONTROL_COBOL_MIA4.txt")
Call BvpEnv("PACMM4","PAC7W1",Rep_TMP & "\\WW1.tmp")
Call BvpEnv("PACMM4","PAC7W2",Rep_TMP & "\\WW2.tmp")
Call BvpEnv("PACMM4","PACM2P",Rep_TMP & "\\W2P.tmp")
Call BvpEnv("PACMM4","PACM2W",Rep_TMP & "\\W2W.tmp")
Call BvpEnv("PACMM4","PAC2GP",Rep_TMP & "\\WGP.tmp")
Call BvpEnv("PACMM4","PAC2GW",Rep_TMP & "\\WGW.tmp")
Call BvpEnv("PACMM4","PAC780",Rep_TMP & "\\W80.tmp")
Call BvpEnv("PACMM'", "PAC7IA",Rep_USR & "\\MIA4IA.txt")
Call RunCmdLog ("BVPACMM4")
Call Err_Cod(Return , 0 , "PACMM4")

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

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

</script>
</job>

```

---

## TRDQ - Conversion des blocs DBD en blocs SQL

### TRDQ - Présentation générale

#### Principe

Cette procédure est un utilitaire qui permet de convertir des Blocs Bases de Données DB2 (type DB) en blocs DBD SQL (type Q2).

#### Première étape

A partir des cartes de commandes, des cartes d'entrée de RMEN sont formatées avec l'option RENAME (RN)

#### Seconde étape

RMEN renomme les codes blocs de type DB en nouveaux codes blocs précisés sur la carte de commande.

RMEN créé un fichier séquentiel contenant des :

- Mouvements de création et de modification avec le nouveau code bloc triés par :

- niveau hiérarchique croissant de bibliothèque,
- bibliothèque,
- type d'enregistrement (définitions, descriptions, utilisations).
- Mouvements d'annulation avec l'ancien code bloc triés par :
  - niveau hiérarchique décroissant de bibliothèque,
  - bibliothèque,
  - type d'enregistrement (utilisations, descriptions, définitions).

### Troisième étape

Transformation des blocs types DB en blocs type Q2.

A partir du fichier issu de RMEN et des lignes de commande, transformation des enregistrements :

- Fiches des nouveaux blocs : le type DB est transformé en type Q2. Le nom externe et le libellé peuvent être modifiés.
- Descriptifs des blocs de données Codasyl-DB2 (-DC) : ils sont transformés en descriptif de Bloc relationnel (-DR).
- Documentation des descriptifs des blocs de données Codasyl-DB2 (-DC nlg GG et -DC nlg GC) : ils sont transformés en documentation des descriptifs des blocs relationnels (-DR nlg GG et -DR nlg GC).

Les autres enregistrements issus de RMEN sont reconduits tels quels dans le fichier mouvement en sortie.

### Condition d'exécution

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

Autorisation d'accès aux procédures batch : l'utilisation de la procédure RMEN est soumise à un contrat d'acquisition.

### Edition obtenue

Cette procédure éditée dans l'étape PACX (RMEN) un compte- rendu de sélection avec indication des erreurs éventuelles.

### Résultat obtenu

Le résultat obtenu est un fichier séquentiel contenant les mouvements transposés du bloc Q2, mouvements destinés à la mise à jour batch (procédure UPDT).

### Remarques

Si l'utilisateur souhaite conserver le code du bloc d'origine, il faut poursuivre en exécutant RMEN pour renommer le nouveau code de type Q2.

### TRDQ - Entrées Utilisateur

Une seule ligne '\*' contenant le(s) bloc(s) DBD de type DB à transposer.

Cette contrainte est imposée par RMEN.

Il faut exécuter la procédure pour chaque nouveau contexte d'extraction :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code carte
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 historisée
29	4	RMEN	Code de l'extracteur
33	1	'1'	Formatage pour UPDT
50	1	' '	Pas de report du mot de passe
		'1'	Report du mot de passe

Une ligne commande par entité bloc DBD de type DB à transposer en bloc SQL de type Q2 :

Pos.	Lon.	Valeur	Signification
2	2	'W1'	Code carte
4	6	bbbbbb	Code Pacbase du bloc DB à transposer
10	6	qqqqqq	Code Pacbase du bloc Q2 obtenu : ce nouveau code doit être différent de tout code bloc existant déjà
			Modifications facultatives pour le bloc Q2 obtenu :
16	8	eeeeeeee	Nom externe du bloc Q2
24	1	x	Code carte avant bloc Q2
25	1	y	Code carte après bloc Q2
26	36	nn....	Libellé du bloc Q2
62	2	Q2	Type du bloc Q2

## TRDQ - Description des étapes

### Formatage des entrées de RMEN : BVPTUDQ1

Code	Nom physique	Type	Libellé
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7ME	Rép. tmp : WMB	Sortie	Fichier des entrées pour RMEN

### 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
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	Rép. tmp : WMB	Entrée	Entrées utilisateur
PAC7MA	NUL	Entrée	Fichier travail
PAC7ES	NUL	Entrée	Fichier travail
PAC7BM	Rép. tmp : WBM	Entrée/Sortie	Entrées utilisateur
PAC7MM	Rép. tmp : WMM	Entrée/Sortie	Fichier travail
PAC7MJ	Rép. tmp : WMJ	Entrée/Sortie	Fichier travail
PAC7TE	Rép. tmp : WTE	Entrée/Sortie	Fichier travail
PAC7RE	Rép. tmp : WRE	Entrée/Sortie	Fichier travail
PAC7RM	Rép. tmp : WRM	Entrée/Sortie	Fichier travail
PAC7WD	Rép. tmp : WWD	Entrée/Sortie	Mouvements extraits
SYSEXT	Rép. tmp : WSY	Entrée/Sortie	Fichier de travail
PAC7RQ	NUL	Sortie	Fichier travail
PAC7MV	Rép. tmp : WMV	Sortie	Fichier travail



Code	Nom physique	Type	Libellé
PAC7MR	Rép. tmp : WMR	Sortie	Fichier travail
PAC7MX	Rép. tmp : WMX	Sortie	Entités non extraites (PACX)
PAC7CP	NUL	Sortie	Fichier travail
PAC7GY	NUL	Sortie	Mouvements extraits pour UPDP
PAC7RQ	NUL	Sortie	Correspondance Entités/Bibliothèque origine
PAC7TD	Rép. tmp : WTD	Sortie	Fichier travail
PAC7UE	Rép. tmp : WUE	Sortie	Fichier travail
PAC7IA	Rép. user : PACXIA	Etat	Edition générale de l'enchaînement des programmes
PAC7DD	Rép. user : PACXDD	Etat	Edition des anomalies sur mouvements en entrée
PAC7ED	Rép. tmp : WED	Etat	Compte-rendu d'extractions
PAC7EE	Rép. user : PACXEE	Etat	Compte-rendu d'extractions
PAC7EG	Rép. tmp : WEG	Etat	Compte-rendu d'extractions
PAC7EM	Rép. tmp : WEM	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. tmp : WEU	Etat	Compte-rendu d'extractions
PAC7EZ	Rép. user : PACXEZ	Etat	Compte-rendu d'extractions

### Transposition des blocs DB en blocs SQL : BVPTUDQ3

Code	Nom physique	Type	Libellé
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7MX	Rép. tmp : WMV	Entrée	Fichier issu de RMEN

Code	Nom physique	Type	Libellé
PAC7MV	Rép. user: UPDATESQL	Sortie	Fichier MV pour mise à jour

## TRDQ - JCL d'exécution

```

 VISUALAGE PACBASE

 CONVERSION OF DBD BLOCKS INTO SQL BLOCKS

<job id=TRDQ>

<script language="VBScript">
Dim MyProc
MyProc = "TRDQ"
</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" , "PTUDQ1"))
'-----
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTUDQ1","PAC7ME",Rep_TMP & "\WMB.tmp")
Call RunCmdLog ("BVPTUDQ1")
Call Err_Cod(Return , 0 , "PTUDQ1")

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"
Call BvpEnv("PACX","PAC7MB",Rep_TMP & "\WMB.tmp")
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","PAC7RQ",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\NUL.tmp")

```

```

Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\WTD.tmp")
Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\WEU.tmp")
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 , 4 , "PACX")

Call Msg_Log (Array("1022" , "PTUDQ3"))
'-----
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTUDQ3","PAC7MV",Rep_USR & "\UPDATESQL.txt")
Call BvpEnv("PTUDQ3","PAC7MX",Rep_TMP & "\WMV.tmp")
Call RunCmdLog ("BVPTUDQ3")
Call Err_Cod(Return , 0 , "PTUDQ3")

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

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

</script>
</job>

```



---

## Chapitre 9. Les composants

---

### Composants de la partie Serveur

#### Présentation générale

D'un point de vue technique, le produit gère des données permanentes en "mode conversationnel" ou en "mode batch" grâce à l'utilisation de deux types de ressources :

- Des répertoires dans lesquelles sont stockés les programmes constitutifs du système, ainsi que les paramètres nécessaires à son fonctionnement.
- Des fichiers permanents qui matérialisent les données manipulées par les programmes définis précédemment. Ces fichiers peuvent être classés en deux catégories :
  - Les fichiers système, qui ne sont pas liés à une Base de Développement particulière et restent relativement stables.
  - Les fichiers évolutifs qui correspondent à une Base de Développement et dont le volume varie en fonction des mises à jour effectuées.

#### Documentation en ligne

En plus des bibliothèques décrites dans les sous-chapitres précédents, le produit comprend le fichier contenant les libellés d'erreur et la documentation en ligne : AE.

#### Squelettes de génération

Le produit comprend par ailleurs les fichiers suivants :

- Un fichier squelette de génération SA, utilisé par le générateur Batch pour les API COBOL.
- Un fichier squelette de génération SC, utilisé par le générateur Batch.
- Un fichier squelette de génération SG, utilisé par les générateurs Dialogue et Base de données.
- Un fichier squelette de génération SN, utilisé par le générateur eBusiness.
- Un fichier squelette de génération SR, utilisé par le générateur Reverse.
- Un fichier squelette de génération SP, utilisé par le générateur d'extracteurs PAF.
- Un fichier squelette SF, utilisé par le générateur d'extracteurs PAF.
- Un fichier squelette de génération SS, utilisé par le générateur eBusiness.

## **Base Administration**

### **Fichiers de la Base Administration**

- Le fichier des données AR.
- Le fichier des données extension AY.
- Le fichier des index AN.
- Le fichier journal AJ.
- Le fichier utilisateur GU.

Ce fichier étant sensible, sa sauvegarde est à prévoir dans un environnement sécurisé du site d'installation.

- Le fichier journal QJ du module SCM

### **Sauvegarde de la Base Administration**

La sauvegarde des fichiers de la Base Administration est constituée de deux fichiers séquentiels à génération.

- La sauvegarde de la base (PC).

C'est un fichier de sauvegarde séquentielle des constituants de la Base Administration : index (AN), données (AR) et extension (AY).

- La sauvegarde du journal (PJ).

Son objectif est d'accumuler tous les mouvements de mise à jour de la Base Administration depuis son installation et ayant transité par le fichier des mouvements (AJ).

Au cas où sa taille deviendrait incompatible avec les contraintes d'exploitation, la procédure ARCH de la Base Administration permet de le diviser en plusieurs fichiers dont seul le plus récent est manipulé régulièrement.

- La sauvegarde du journal QJ du module SCM.

La sauvegarde de QJ est le fichier JQ.

Ce fichier permet d'archiver les mouvements valides déjà traités par les différentes mises à jour et stockés dans le fichier journal QJ.

## **Base de Développement**

### **Fichiers de la Base de Développement**

Ces fichiers contiennent toutes les données relatives au développement des applications.

- Le fichier des données AR.
- Le fichier des données extension AY.
- Le fichier des index AN.
- Le fichier journal AJ.

Tous les mouvements passés sur la base, que ce soit en batch ou en TP, sont conservés afin de permettre une restauration de la base en cas de défaillance des sécurités standard du système, et d'autre part de fournir la source d'informations à l'outil d'analyse de l'évolution de la base.

Ces mouvements sont normalement stockés dans le fichier de sauvegarde des mouvements (PJ), le fichier des mouvements ne servant que de transit entre le moment où le système les traite et le moment où la procédure ARCH les conduit sur leur support définitif.

### **Fichiers de sauvegarde Base de Développement**

Suivant l'organisation prise lors de la restauration, la sauvegarde du réseau est constituée de deux fichiers séquentiels à "rotation" (PC et PJ) ou de quatre fichiers séquentiels à "rotation" (PC PD PY et PJ).

- La sauvegarde de la Base ou des données (PC).

C'est un fichier de sauvegarde séquentielle des constituants de la Base de Développement (données (AR), index (AN) et extension (AY) parfois ).

- La sauvegarde des index de la Base de Développement PD.
- La sauvegarde des données en vrac de la Base de Développement PY.
- La sauvegarde du journal (PJ).

Son objectif est d'accumuler tous les mouvements de mise à jour de la Base de Développement depuis son installation et ayant transité par le fichier des mouvements (AJ).

Au cas où sa taille deviendrait incompatible avec les contraintes d'exploitation, la procédure ARCH permet de le diviser en plusieurs fichiers dont seul le plus récent est manipulé régulièrement.

### **Modules - fichiers spécifiques**

#### **Pac/Impact :**

- Fichier des critères déjà impactés (FQ).
- Fichier critères de recherche ou points d'entrée (FH).
- Fichier réduit des critères pour épuration (FR).
- Fichier résultats d'impact (FO).

- Fichier des entités en production (FP).

### **DSMS :**

Pour les utilisateurs disposant du module DSMS (voir également le manuel d'exploitation de ce module), un fichier de ce Module est consulté en batch et en TP par Base de Développement.

Ce fichier contient la liste des entités concernées par chaque amélioration. Le numéro d'amélioration est renseigné par l'utilisateur lors de sa connexion à la base.

- Le fichier DSMS des éléments de la Base de Développement (DC).

L'allocation et l'initialisation de ce fichier sont effectuées lors de l'installation du module DSMS.

La définition fournie à l'installation du produit doit être utilisée si le module DSMS n'a pas encore été installé sur le site.

### **PAF :**

- Un fichier de travail PA pour PAF-TP et PUF-TP.

Un fichier indexé de travail est nécessaire à l'utilisation des modules PAF et PUF pour l'ensemble des programmes TP utilisateur accédant aux bases de même radical.

- Un fichier de travail pour PAF batch.

Un fichier indexé de travail est nécessaire à l'utilisation du module PAF par les programmes batch utilisateur. Ce fichier est alloué pour l'exécution et détruit en fin de job.

### Extension module PAF

Un fichier Schémas d'Extraction GS contenant les extracteurs de l'utilisateur et ses macro-commandes.

### **Bibliothèques et fichiers complémentaires**

Les fichiers compléments du dictionnaire sont livrés automatiquement dans le répertoire \SYS\SKEL.

### Module PQC :

- les fichiers BVPQCRA et BVPQCRF contiennent les règles de qualité standard.



- Membre spécifique du module Pacbench Quality Control

Fichier	Contenu ou format	Remarques
BVPQCRA	Fichier séquentiel anglais	Règles standard
BVPQCRF	Fichier séquentiel français	Règles standard

#### Procédure MIAM.

Le fichier des paramètres pour la procédure MIAM est livré automatiquement dans le répertoire \SYS\SKEL.

Membre	Contenu ou format	Remarques
BVPAMIAM	Fichier séquentiel	Paramètres standard pour la procédure MIAM



---

## Chapitre 10. Annexes

---

### Installation du Modèle de la Base Administration

#### VINS - Présentation générale

La procédure VINS effectue la mise à jour batch de la Base Administration à partir de mouvements livrés par IBM. Parfois on peut utiliser un fichier autre. Par exemple en cas de changement de méthode Pacdesign/Pacbench.

#### Condition d'exécution

L'accès au conversationnel doit être fermé.

#### Anomalies d'exécution

Se reporter au manuel "Les procédures batch de l'administrateur" sous-chapitre "Anomalies d'exécution".

Lorsque l'anomalie survient pendant l'exécution des programmes BVPACI30 ou BVPACI40, la base est laissée dans un état incohérent. Il convient alors, après avoir remédié au problème, de recharger la base avec récupération des mouvements archivés, puis d'exécuter à nouveau la procédure VINS.

#### VINS - Entrées / Traitements / Résultats

Cette procédure nécessite deux entrées utilisateur :

- une ligne contenant l'identification de l'utilisateur ainsi que le traitement à effectuer,
- les mouvements permettant de créer les méta entités IBM et de reprendre les entités utilisateur clientes au format 'extension' : en aucun cas, l'utilisateur ne doit modifier le contenu de ces mouvements.

La structure de la ligne est la suivante :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code carte
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	'***'	Code bibliothèque
29	4	'VINS'	
33	1	'I'	Installation des méta entités IBM

## Editions obtenues

La procédure édite :

- un compte-rendu avec la liste des programmes exécutés,
- la liste des demandes avec indication des erreurs éventuelles,
- un compte-rendu des mises à jour effectuées par l'installation.

## Résultat obtenu

Une fois la mise à jour effectuée, le réseau est prêt à être manipulé en conversationnel ou en mode batch.

## Remarque importante

Il faut savoir que les mouvements extraits pour la procédure REOR sont dans un fichier, assigné par PAC7MR, à rendre permanent par substitution.

## **VINS - Description des étapes**

### Mise à jour de la Base Administration : VINS

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
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AJ	Rép. journal : AJ	Sortie	Journal de la Base Administration
PAC7AN	Base Admin - Rép. base : AN	Sortie	Index de la Base Administration
PAC7AR	Base Admin - Rép. base : AR	Sortie	Données de la Base Administration
PAC7AY	Base Admin - Rép. base : AY	Sortie	Extension de la Base Administration
PAC7MA	Système - Rép. skel : BVPMETAD	Entrée	Mouvements méta entités IBM
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur

Code	Nom physique	Type	Libellé
PAC7BM	Rép. tmp : WBM	Entrée/sortie	Entrées utilisateur
PAC7WD	Rép. tmp : WWD	Entrée/sortie	Mouvements extraits
PAC7ES	Rép. tmp : WES	Entrée/sortie	Mouvements extraits
PAC7TD	Rép. user : TD	Entrée/sortie	Mouvements extraits
PAC7MR	Rép. tmp : WMR	Sortie	Mouvements extraits pour REOR sur Base Administration
PAC7MX	NUL	Sortie	Entités non extraites
PAC7RQ	NUL	Sortie	Fichier de travail
PAC7IA	Rép. user : VINSIA	Etat	Edition générale de l'enchaînement des programmes
PAC7CP	NULCP		
PAC7EE	NUL	Etat	Compte-rendu
PAC7EQ	Rép. user : VINSEQ	Etat	Compte-rendu
PAC7EU	Rép. user : VINSEU	Etat	Compte-rendu
PAC7ER	Rép. user : VINSER	Etat	Compte-rendu
PAC7EZ	Rép. user : VINSEZ	Etat	Compte-rendu
PAC7DD	Rép. user : VINSDD	Etat	Compte-rendu
PAC7IE	Rép. user : VINSIE	Etat	Compte-rendu
PAC7IF	Rép. user : VINSIF	Etat	Compte-rendu
PAC7IG	Rép. user : VINSIG	Etat	Compte-rendu
PAC7IH	Rép. user : VINSIH	Etat	Compte-rendu

Codes retour :

- 0 : Pas d'erreur détectée sur les fichiers
- 4 : Corriger les problèmes et relancer la procédure
- 8 : Pas d'autorisation d'accès procédure batch
- 12 : Erreur d'entrée-sortie sur un fichier

### VINS - Script d'exécution

```

| -----
| VISUALAGE PACBASE
|
| -----
| - DICTIONARY UPDATING WITH IBM MODEL DEVPT -
|
| -----
| THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE

```

```

' DATABASE, BASED ON TRANSACTIONS PROVIDED.
'
' INPUT :
' - USER IDENTIFICATION LINE (REQUIRED)
' COL 2 : "*"
' COL 3 : USERIDXX
' COL 11 : PASSWORD
' COL 29 : "VINS"
' COL 33 : "I" - INSTALLATION OF IBM META-ENTITIES
' "R" - RETRIEVAL OF USER ENTITIES WITH THE
' "EXTENSION" FORMAT
' " " "I" + "R"
' -----

```

```
<job id=VINS>
```

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

```
Dim MyProc
```

```
MyProc = "VINS"
```

```
</script>
```

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

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

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

```
Call Msg_Log (Array("1029"))
```

```
'-----
```

```
Call StateList (base, statusL)
```

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

```
' No Rollback
```

```
WshEnv("BVPRB") = "N"
```

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

```
'-----
```

```
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
```

```
WshEnv("PAC7AN") = Rep_BASE & "\AN"
```

```
WshEnv("PAC7AR") = Rep_BASE & "\AR"
```

```
WshEnv("PAC7AY") = Rep_BASE & "\AY"
```

```
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
```

```
WshEnv("PACGGN") = Rep_ABASE & "\AN"
```

```
WshEnv("PACGGR") = Rep_ABASE & "\AR"
```

```
WshEnv("PACGGU") = Rep_ABASE & "\GU"
```

```
WshEnv("PACGGY") = Rep_ABASE & "\AY"
```

```
Call BvpEnv("VINS", "PAC7BM", Rep_TMP & "\WBM.tmp")
```

```
Call BvpEnv("VINS", "PAC7ES", Rep_TMP & "\WES.tmp")
```

```
Call BvpEnv("VINS", "PAC7DD", Rep_USR & "\VINSDD.txt")
```

```
WshEnv("PAC7EE") = Rep_TMP & "\NULEE.tmp"
```

```
'PAC7EE not used, on default
```

```
Call BvpEnv("VINS", "PAC7EQ", Rep_USR & "\VINSEQ.txt")
```

```
Call BvpEnv("VINS", "PAC7ER", Rep_USR & "\VINSER.txt")
```

```
Call BvpEnv("VINS", "PAC7EU", Rep_USR & "\VINSEU.txt")
```

```

Call BvpEnv("VINS", "PAC7EZ", Rep_USR & "\VINSEZ.txt")
Call BvpEnv("VINS", "PAC7IA", Rep_USR & "\VINSIA.txt")
Call BvpEnv("VINS", "PAC7IE", Rep_USR & "\VINSIE.txt")
Call BvpEnv("VINS", "PAC7IF", Rep_USR & "\VINSIF.txt")
Call BvpEnv("VINS", "PAC7IG", Rep_USR & "\VINSIG.txt")
Call BvpEnv("VINS", "PAC7IH", Rep_USR & "\VINSIH.txt")

If base = "ADMIN" Then
Call BvpEnv("VINS", "PAC7MA", Rep_SKEL & "\BVPMETAD")
Else
Call BvpEnv("VINS", "PAC7MA", Rep_SKEL & "\BVPMETBA")
End If

'Example of Input File extracted from PACX/CPSN :
' Call BvpEnv("VINS", "PAC7xx", RepT_USR & "\PACxxx.txt")
'With RepT_USR is Global User Directory.
Call BvpEnv("VINS", "PAC7TD", Rep_USR & "\TD.txt")

WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("VINS", "PAC7MR", Rep_TMP & "\WMR.tmp")
Call BvpEnv("VINS", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("VINS", "PAC7RQ", Rep_TMP & "\NUL.tmp")
Call BvpEnv("VINS", "PAC7WD", Rep_TMP & "\WWD.tmp")
Call BvpEnv("VINS", "PAC7CP", Rep_TMP & "\NULCP.tmp")
Call BvpEnv("VINS", "SYSEXT", Rep_TMP & "\VINSYS.tmp")
Call RunCmdLog ("BVPVINS")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "VINS")

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

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

</script>
</job>

```

---

## Installation du Modèle de la Base de Développement

### VINS - Présentation générale

La procédure VINS effectue la mise à jour batch de la Base de Développement à partir de mouvements livrés par IBM. Parfois on peut utiliser un fichier autre. Par exemple en cas de changement de méthode Pacdesign/Pacbench.

Les entités sont créées en inter-bibliothèque et en session 0001Z afin d'en permettre l'accès depuis n'importe quelle bibliothèque et à partir de n'importe quelle session de la Base de Développement.

#### Condition d'exécution

L'accès au conversationnel doit être fermé.

#### Anomalies d'exécution

Se reporter au manuel "Les procédures batch de l'administrateur" sous-chapitre "Anomalies d'exécution".

Lorsque l'anomalie survient pendant l'exécution des programmes BVPACI30 ou BVPACI40, la base est laissée dans un état incohérent. Il convient alors, après avoir remédié au problème, de recharger la base avec récupération des mouvements archivés, puis d'exécuter à nouveau la procédure VINS.

### **VINS - Entrées / Traitements / Résultats**

Cette procédure nécessite deux entrées utilisateur :

- une ligne contenant l'identification de l'utilisateur ainsi que le traitement à effectuer,
- les mouvements permettant de créer les méta entités IBM et de reprendre les entités utilisateur clientes au format 'extension' : en aucun cas, l'utilisateur ne doit modifier le contenu de ces mouvements.

La structure de la ligne est la suivante :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code carte
3	8	uuuuuuuu	Code utilisateur
11	8	pppppppp	Mot de passe
19	3	'***'	Code bibliothèque
29	4	'VINS'	
33	1	'I'	Installation des méta entités IBM
		'R'	Reprise des entités utilisateur au format 'extension'
		' '	'I' + 'R'

#### Editions obtenues

La procédure édite :

- un compte-rendu avec la liste des programmes exécutés,



- la liste des demandes avec indication des erreurs éventuelles,
- un compte-rendu des mises à jour effectuées par l'installation,
- un compte-rendu des mises à jour effectuées par la reprise.

### Résultat obtenu

Une fois la mise à jour effectuée, le réseau est prêt à être manipulé en conversationnel ou en mode batch.

Pour la reprise des entités utilisateur au format extension, un fichier séquentiel de mouvements d'épuration peut être généré (si 'R' en colonne 33 de l'entrée utilisateur). Il convient alors d'effectuer une réorganisation de la base avec ce fichier en entrée. La réorganisation de la base avec le fichier PC sauvegardé après la reprise est également obligatoire dans le cas où les entités utilisateurs clientes des méta entités .PPTX et .QPAQC existent dans plusieurs sessions.

### Remarque importante

Il faut savoir que les mouvements extraits pour la procédure REOR sont dans un fichier, assigné par PAC7MR, à rendre permanent par substitution.

## **VINS - Description des étapes**

### Mise à jour de la Base de Développement : VINS

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
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7AJ	Rép. journal : AJ	Sortie	Journal de la Base de Développement
PAC7AN	Rép. base : AN	Sortie	Index de la Base de Développement
PAC7AR	Rép. base : AR	Sortie	Données de la Base de Développement

Code	Nom physique	Type	Libellé
PAC7AY	Rép. base : AY	Sortie	Extension de la Base de Développement
PAC7MA	Système - Rép. skel : BVPMETBA	Entrée	Mouvements méta entités IBM
PAC7MB	Entrée utilisateur	Entrée	Entrées utilisateur
PAC7BM	Rép. tmp : WBM	Entrée/ sortie	Entrées utilisateur
PAC7WD	Rép. tmp : WWD	Entrée/ sortie	Mouvements extraits
PAC7ES	Rép. tmp : WES	Entrée/ sortie	Mouvements extraits
PAC7TD	Rép. user : TD	Entrée/ sortie	Mouvements extraits
PAC7MR	Rép. tmp : WMR	Sortie	Mouvements extraits pour REOR
PAC7MX	NUL	Sortie	Entités non extraites
PAC7RQ	NUL	Sortie	Fichier de travail
PAC7IA	Rép. user : VINSIA	Etat	Edition générale de l'enchaînement des programmes
PAC7CP	NULCP		
PAC7EE	NUL	Etat	Compte-rendu
PAC7EQ	Rép. user : VINSEQ	Etat	Compte-rendu
PAC7EU	Rép. user : VINSEU	Etat	Compte-rendu
PAC7ER	Rép. user : VINSER	Etat	Compte-rendu
PAC7EZ	Rép. user : VINSEZ	Etat	Compte-rendu
PAC7DD	Rép. user : VINSDD	Etat	Compte-rendu
PAC7IE	Rép. user : VINSIE	Etat	Compte-rendu
PAC7IF	Rép. user : VINSIF	Etat	Compte-rendu
PAC7IG	Rép. user : VINSIG	Etat	Compte-rendu
PAC7IH	Rép. user : VINSIH	Etat	Compte-rendu

Codes retour :

- 0 : Pas d'erreur détectée sur les fichiers
- 4 : Corriger les problèmes et relancer la procédure
- 8 : Pas d'autorisation d'accès procédure batch
- 12 : Erreur d'entrée-sortie sur un fichier

## VINS - Script d'exécution

```
'-----
' VISUALAGE PACBASE
'-----
' - DICTIONARY UPDATING WITH IBM MODEL DEVPT -
'-----
'
' THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE
' DATABASE, BASED ON TRANSACTIONS PROVIDED.
'
' INPUT :
' - USER IDENTIFICATION LINE (REQUIRED)
' COL 2 : "*"
' COL 3 : USERIDXX
' COL 11 : PASSWORD
' COL 29 : "VINS"
' COL 33 : "I" - INSTALLATION OF IBM META-ENTITIES
' "R" - RETRIEVAL OF USER ENTITIES WITH THE
' "EXTENSION" FORMAT
' " " "I" + "R"
'-----
'
<job id=VINS>

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

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

<script language="VBScript">

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

Call Msg_Log (Array("1029"))
'-----
Call StateList (base, statusL)

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

' No Rollback
WshEnv("BVPRB") = "N"

Call Msg_Log (Array("1022" , "VINS"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
```

```

WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("VINS", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("VINS", "PAC7ES", Rep_TMP & "\WES.tmp")
Call BvpEnv("VINS", "PAC7DD", Rep_USR & "\VINSDD.txt")
WshEnv("PAC7EE") = Rep_TMP & "\NULEE.tmp"
'PAC7EE not used, on default
Call BvpEnv("VINS", "PAC7EQ", Rep_USR & "\VINSEQ.txt")
Call BvpEnv("VINS", "PAC7ER", Rep_USR & "\VINSER.txt")
Call BvpEnv("VINS", "PAC7EU", Rep_USR & "\VINSEU.txt")
Call BvpEnv("VINS", "PAC7EZ", Rep_USR & "\VINSEZ.txt")
Call BvpEnv("VINS", "PAC7IA", Rep_USR & "\VINSIA.txt")
Call BvpEnv("VINS", "PAC7IE", Rep_USR & "\VINSIE.txt")
Call BvpEnv("VINS", "PAC7IF", Rep_USR & "\VINSIF.txt")
Call BvpEnv("VINS", "PAC7IG", Rep_USR & "\VINSIG.txt")
Call BvpEnv("VINS", "PAC7IH", Rep_USR & "\VINSIH.txt")

If base = "ADMIN" Then
Call BvpEnv("VINS", "PAC7MA", Rep_SKEL & "\BVPMETAD")
Else
Call BvpEnv("VINS", "PAC7MA", Rep_SKEL & "\BVPMETBA")
End If

'Example of Input File extracted from PACX/CPSN :
' Call BvpEnv("VINS", "PAC7xx", RepT_USR & "\PAC7xx.txt")
'With RepT_USR is Global User Directory.
Call BvpEnv("VINS", "PAC7TD", Rep_USR & "\TD.txt")

WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("VINS", "PAC7MR", Rep_TMP & "\WMR.tmp")
Call BvpEnv("VINS", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("VINS", "PAC7RQ", Rep_TMP & "\NUL.tmp")
Call BvpEnv("VINS", "PAC7WD", Rep_TMP & "\WWD.tmp")
Call BvpEnv("VINS", "PAC7CP", Rep_TMP & "\NULCP.tmp")
Call BvpEnv("VINS", "SYSEXT", Rep_TMP & "\VINSSYS.tmp")
Call RunCmdLog ("BVPVINS")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "VINS")

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

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

</script>
</job>

```

---

## Utilitaires d'initialisation des fichiers GS/HE/ZS

### LDGS - Présentation générale

Cette procédure permet la création physique du fichier indexé GS ainsi que son initialisation.

Cette procédure est à exécuter lorsque le fichier GS n'est pas physiquement créé (en particulier lors de la première installation) ou qu'il est logiquement désorganisé.

#### Condition d'exécution

Les serveurs TP doivent être arrêtés.

#### Entrée Utilisateur

Aucune.

### LDGS - Description des étapes

#### Initialisation du fichier GS : PTLDGS

Cette étape initialise le fichier GS avec un enregistrement.

Code	Nom physique	Type	Libellé
PAC7GS	Rép. base : GS	Sortie	Fichier de travail Bases de Données

### LDGS - Script d'exécution

```
'-----
' VISUALAGE PACBASE
'-----
' INITIALIZATION OF THE GS FILE
'-----
'
<job id=LDGS>

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

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

<script language="VBScript">

Call Msg_Log (Array("1029"))
'-----
```

```

Call StateList (base, statusL)

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

Call Msg_Log (Array("1022" , "PTLDGS"))
'-----
WshEnv("PAC7GS") = Rep_BASE & "\GS"
Call RunCmdLog ("BVPTLDGS")

Call Err_Cod(Return , 0 , "PTLDGS")

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

</script>
</job>

```

## LDHE - Présentation générale

Cette procédure permet la création physique du fichier indexé HE ainsi que son initialisation.

Cette procédure est à exécuter lorsque le fichier HE n'est pas physiquement créé (en particulier lors de la première installation) ou qu'il est logiquement désorganisé.

### Condition d'exécution

Les serveurs TP doivent être arrêtés.

### Entrée Utilisateur

Aucune.

## LDHE - Description des étapes

### Initialisation du fichier HE : PTLDHE

Cette étape initialise le fichier HE avec un enregistrement.

Code	Nom physique	Type	Libellé
PAC7HE	Rép. base : HE	Sortie	Fichier sauvegarde de maquetage

## LDHE - Script d'exécution

```

'-----
' VISUALAGE PACBASE
'-----
<job id=LDHE>

```

```

<script language="VBScript">
MyProc = "LDHE"
</script>

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

<script language="VBScript">

Call Msg_Log (Array("1029"))
'-----
Call Statelist (base, statusL)

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

Call Msg_Log (Array("1022" , "PTLDHE"))
'-----
WshEnv("PAC7HE") = Rep_BASE & "\HE"
Call RunCmdLog ("BVPTLDHE")

Call Err_Cod(Return , 0 , "PTLDHE")

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

</script>
</job>

```

## **LDZS - Présentation générale**

Cette procédure permet la création physique du fichier indexé ZS ainsi que son initialisation.

Cette procédure est à exécuter lorsque le fichier ZS n'est pas physiquement créé (en particulier lors de la première installation) ou qu'il est logiquement désorganisé.

### Condition d'exécution

Les serveurs TP doivent être arrêtés.

### Entrée Utilisateur

Aucune.

## **LDZS - Description des étapes**

### Initialisation du fichier ZS : PTLDTS

Cette étape initialise le fichier ZS avec un enregistrement.

Code	Nom physique	Type	Libellé
PAC7ZS	Rép. base : ZS	Sortie	Fichier de travail Bases de Données

## LDZS - Script d'exécution

```

' -----
' VISUALAGE PACBASE
' -----
<job id=LDZS>

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

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

<script language="VBScript">

Call Msg_Log (Array("1029"))
' -----
Call StateList (base, statusL)

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

Call Msg_Log (Array("1022" , "PTLDTS"))
' -----
WshEnv("PAC7ZS") = Rep_BASE & "\ZS"
Call RunCmdLog ("BVPTLDTS")

Call Err_Cod(Return , 0 , "PTLDTS")

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

</script>
</job>

```

---

## Utilitaires d'aide à la reprise

### UTU1 UTU2 - Adaptation des opérateurs 'UNS'

#### UTU1 - Extraction ligne 'P' avec 'UNS'

##### Principe



Cette procédure est un utilitaire qui permet d'extraire les lignes 'P' des programmes contenant l'opérateur 'UNS' et n'ayant aucune information dans la zone 'Niveau-conditionnement'. L'utilisateur vérifiera le fichier en sortie. Il devra supprimer de ce fichier toutes les lignes à garder en l'état. Pour toutes les lignes conservées dans le fichier, la zone 'Niveau-conditionnement' sera forcée à '99BL' si on exécute la procédure UTU2.

### Condition d'exécution

Mettre le nom du fichier résultat des lignes à pointer dans le Script de lancement : NOMUT='... '

### Résultat obtenu

Un fichier des lignes 'P' UNS à pointer 'NOMUT'.

### **UTU1 - Entrées Utilisateur**

Pas d'entrées utilisateur.

### **UTU1 - Description des étapes**

Extraction lignes 'P' avec opérateur 'UNS' : UTIUN1

Code	Nom physique	Type	Libellé
PAC7PC	Rép. Save : PC	Entrée	Image séquentielle du réseau
UTUTO	Rép. user : UTUTI	Sortie	Lignes 'P' UNS à mettre à jour

### **UTU1 - Script d'exécution**

```

| -----
| VISUALAGE PACBASE
|
| -----
| EXTRACTION OF LINES "P"
| WITH OPERATOR "UNS"
| WITHOUT LEVEL-CONDITION TYPE
| -----
|
<job id=UTU1>

<script language="VBScript">
Dim MyProc
MyProc = "UTU1"
</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" , "UTU1"))
'-----
WshEnv("PAC7PC") = BVP_SvName & ""
Call BvpEnv("UTU1","UTÛT0",RepT_USR & "\UTUTI.txt")

Call RunCmdLog ("BVPTUNS1")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "UTU1")

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

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

</script>
</job>

```

## **UTU2 - Mise à jour ligne 'P' avec 'UNS'**

### Principe

Cette procédure est un utilitaire qui permet de mettre à jour les enregistrements extrait par la procédure UTU1. Pour toutes les lignes du fichier en entrée, on force '99BL' dans la zone 'Niveau-conditionnement'.

### Condition d'exécution

Mettre le nom du fichier résultat de la procédure UTU1 dans le Script de lancement : NOMUT='... '

### Résultat obtenu

Nouvelle image séquentielle du réseau.

## **UTU2 - Entrées Utilisateur**

Pas d'entrées utilisateur.

## UTU2 - Description des étapes

Maj lignes 'P' avec opérateur 'UNS' : UTIUN2

Code	Nom physique	Type	Libellé
PAC7PC	Rép. save : PC	Entrée	Image séquentielle du réseau
UTUTI	Rép. user : UTUTI	Entrée	Fichier des lignes 'P' UNS à mettre à jour
PAC7CP	Rép. save : PC-new	Sortie	Nouvelle Image séquentielle du réseau

## UTU2 - Script d'exécution

```
| -----
| VISUALAGE PACBASE
|
| -----
| UPDATE LINES "P"
| WITH OPERATOR "UNS"
| -----
|
<job id=UTU2>

<script language="VBScript">
Dim MyProc
MyProc = "UTU2"
</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" , "UTU2"))
| -----
WshEnv("PAC7PC") = BVP_SvName & ""
Call BvpEnv("UTU2", "UTUTI", RepT_USR & "\UTUTI.txt")
WshEnv("PAC7CP") = Rep_SAVE & "\PC-new"
Call RunCmdLog ("BVPTUNS2")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "UTU2")

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

Call Msg_Log (Array("1022" , "BACKUP"))
| -----
Call Turnover(Rep_SAVE & "\PC")
```

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

</script>
</job>
```

## **UTM1 UTM2 - Migration anciennes méta entités**

### Principe

Cette procédure est un utilitaire d'aide à la migration des méta entités de type old vers des méta entités de type formaté.

La reprise PC25 transforme les entités utilisateurs 2.n en méta entités 3.n de type old. Ces méta entités ne sont pas modifiables.

Leur transformation en méta entités de type formaté doit permettre leurs modification et enrichissement.

Cette migration nécessite l'intervention du client et va se dérouler en plusieurs étapes.

### Déroulement de la migration

#### Etape 1 : Analyse des relations de la base (UTM1)

En version 3.n, une relation pointe sur un seul type d'entité et non plus à l'instar de la version 2.n, sur une multitude de types.

Toute relation de type old doit être transformée en autant de relations que de nombre d'entités auxquelles elle est liée.

Cette première étape consiste donc à rechercher dans la base VA Pacbase les appels des relations de type old dans les lignes de description des méta entités ainsi que dans toutes les entités.

La liste des appels est éditée et alimente le fichier en sortie.

Dans ce fichier, les caractères '&&' séparent les informations éditées des informations sous forme de fichier mouvement.

L'utilisateur attribuera le nouveau code relation de son choix.

L'ancien code pourra être conservé si la relation pointe sur un seul type d'entité.

Exemple de transformation :

```
RELAT1 P 220 P PGMXCR CR VAO 2243 && ...
```

modifié en :

```
RELATE P 220 P PGMXCR CR VAO 2243 && ...
```

La relation RELATE sera ainsi spécifiquement dédiée aux entités de type 'P' (programme), alors que RELAT1 était dédiée à plusieurs types.

Remarque : la transformation n'est à effectuer que sur le nom de la Relation situé AVANT les caractères &&. La suite de la ligne doit rester en l'état.

Si un même code relation est lié à des EU, mais dans des bibliothèques de réseaux différents, il faut renommer la relation pour chaque nouvelle bibliothèque.

#### Etape 2 : Transformation des méta entités, des relations et des entités impactées et mise à jour (UTM2)

- Préalablement à la transformation ,il est contrôlé que toute relation du fichier PAC7ME pointe sur un seul type d'entité. Sinon un display est édité qui demande une uniformisation du type et une ré-exécution de UTM2. La procédure est alors stoppée.

Quand le fichier ME est cohérent , les méta entités de type old sont transformées en méta entités de type formaté.

- Les relations choisies par l'utilisateur sont modifiées ou créées.  
Si la relation existe déjà dans la base, elle pourra convenir si elle est définie dans un contexte satisfaisant pour l'entité appelante, si elle est de type contraint et si son type d'entité reliée est identique au type choisi.

Si la relation est de type old, elle sera modifiée en type contraint et le type d'entité choisi lui sera relié.

Si la relation existante ne convient pas, un message d'erreur est édité sous forme de displays et l'utilisateur devra choisir un nouveau code relation.

Toute création de relation se fera dans la bibliothèque et la session H la plus proche de son utilisation.

- Les appels des relations sont mis en forme dans les entités concernées (lignes de description des méta entités et autres).
- Si des erreurs sont détectées (code retour à 8), l'utilisateur devra modifier le fichier en entrée PAC7ME et relancer cette procédure UTM2 avant que la mise à jour ne s'exécute. Attention, ne pas oublier de supprimer ou renommer le fichier permanent &UTM2MV auparavant.

#### Mise à jour

La mise à jour est effectuée par le programme BVPACA15.

Une sauvegarde de la base est recommandée avant son exécution.

Seuls les mouvements en erreur sont édités. Il peut y avoir de nombreux rejets car ces entités créées en version 2.n sont soumises à un plus grand nombre de contrôles en version 3.n. Des interventions manuelles dans la base sont alors indispensables.

Les mouvements ne sont pas journalisés.

Le fichier mouvement &UTM2MV est déclaré en fichier permanent pour permettre à l'utilisateur de visualiser l'ensemble des mouvements soumis à la mise à jour.

### Etape 3 : Etat des lieux

Il est recommandé de réexécuter la première étape afin de s'assurer de l'absence d'appels de relations old dans la base.

Sinon, une intervention de l'utilisateur est à nouveau nécessaire ainsi que le lancement des étapes suivantes.

### Etape 4 : Réorganisation

Quand la migration est jugée satisfaisante, une réorganisation de la base est nécessaire.

#### Condition d'exécution

Aucune pendant l'étape 1 (UTM1).

Pour l'étape 2 de mise à jour, il est nécessaire de fermer les fichiers AR, AN, AJ et AY dans le conversationnel (sauf pour les matériels permettant la concurrence batch/conversationnel).

#### Edition obtenue

A l'issue de l'étape 1, un compte-rendu édite la liste des appels de relations de type old.

A l'issue de l'étape 2 avant la mise à jour, des messages d'erreur sont édités sous forme de displays.

A l'issue de la mise à jour, un compte-rendu signale les anomalies rencontrées.

#### Résultat obtenu

Une fois la réorganisation effectuée, le résultat obtenu est un réseau exempt de méta entités de type old et d'appels de relations de type old.

## UTM1 - Description des étapes

Analyse des relations de la base : PTUME1

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Base Admin - Rép. base : AN	Entrée	Index de la Base de Développement
PAC7ME	Rép. user : UTM1ME	Sortie	Liste des appels des relations dans toutes les entités (longueur = 221)
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7EQ	Rép. user : UTM1EQ	Etat	Compte-rendu

## UTM1 - Script d'exécution

```
' -----

' VISUALAGE PACBASE
'
' -----
' CONVERSION OF OLD META-ENTITIES
' ANALYSIS OF THE DATABASE RELATIONS
' -----
'
<job id=UTM1>

<script language="VBScript">
Dim MyProc
MyProc = "UTM1"
</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" , "PTUME1"))
' -----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTUME1","PAC7EQ",Rep_USR & "\UTM1EQ.txt")
Call BvpEnv("PTUME1","PAC7ME",Rep_USR & "\UTM1ME.txt")
Call RunCmdLog ("BVPTUME1")
Call Err_Cod(Return , 0 , "PTUME1")
```

```

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

</script>
</job>

```

## UTM2 - Entrées Utilisateur

Une carte \* en entrée de la seconde étape de la migration (UTM2) avec un code utilisateur et son mot de passe.

Si la carte \* n'est pas renseignée, un message d'anomalie est émis et la procédure ne peut s'effectuer.

## UTM2 - Description des étapes

Contrôle de la cohérence du fichier des relations : PTUME3

Code	Nom physique	Type	Libellé
PAC7ME	Rép. user : UTM1ME	Entrée	Liste des appels des relations dans toutes les entités

Transformation des méta entités, des relations et de leurs appels : PTUME2

Code	Nom physique	Type	Libellé
PAC7AR	Base Admin - Rép. base : AR	Entrée	Données de la Base de Développement
PAC7AN	Base Admin - Rép. base : AN	Entrée	Index de la Base de Développement
PAC7ME	Rép. user : UTM1ME	Entrée	Liste des appels des relations dans toutes les entités
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PAC7MV	Rép. user : UTM2MV	Sortie	Mouvements mis en forme (longueur = 170)
PAC7ET	Rép. user : UTM2ET	Sortie	Compte-rendu
PAC7MB	Entrée utilisateur	Entrée	Entrée utilisateur

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	NUL	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	NUL	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	NUL	Entrée	Fichier de travail
PAC7MV	Rép. user : UTM2MV	Entrée	Mouvements de mise à jour
PAC7RB	NUL	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	NUL	Sortie	Mouvements erronés UPDP (longueur=310)
PAC7IE		Etat	Compte-rendu de mise à jour (longueur=132)
PAC7IF		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

## UTM2 - Script d'exécution

```
|-----|
| VISUALAGE PACBASE |
|-----|
| CONVERSION OF OLD META-ENTITIES |
| THE RELATIONS AND THEIR CALLS |
|-----|
|
|<job id=UTM2>
|
|<script language="VBScript">
|Dim MyProc
|MyProc = "UTM2"
|</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" , "PTUME3"))
|-----|
|Call BvpEnv("PTUME3","PAC7ME",RepT_USR & "\UTM1ME.TXT")
|Call RunCmdLog ("BVPTUME3")
|Call Err_Cod(Return , 0 , "PTUME3")
|
|Call Msg_Log (Array("1022" , "PTUME2"))
|-----|
|WshEnv("PAC7AE") = Rep_SKEL & "\AE"
|WshEnv("PAC7AN") = Rep_BASE & "\AN"
|WshEnv("PAC7AR") = Rep_BASE & "\AR"
|WshEnv("PAC7MB") = Fic_Input
|Call BvpEnv("PTUME2","PAC7ET",Rep_USR & "\UTM2ET.txt")
|Call BvpEnv("PTUME2","PAC7ME",RepT_USR & "\UTM1ME.txt")
|Call BvpEnv("PTUME2","PAC7MV",Rep_USR & "\UTM2MV.txt")
|Call RunCmdLog ("BVPTUME2")
|Call Err_Cod(Return , 0 , "PTUME2")
|
|Call Msg_Log (Array("1022" , "PACA15"))
|-----|
|WshEnv("PAC7AE") = Rep_SKEL & "\AE"
|WshEnv("PAC7AJ") = Rep_TMP & "\NULAJ.tmp"
|'PAC7AJ not used, on default
|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("PACGGY") = Rep_ABASE & "\AY"
|WshEnv("SEMLOCK") = Rep_BASE & "\LO"
```

```

WshEnv("SEMADMIN") = Rep_ABASE & "\\LO"
WshEnv("PAC7IE") = Rep_TMP & "\\NULIE.tmp"
'PAC7IE not used, on default
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\\UTM2IFA15.txt")
Call BvpEnv("PACA15","PAC7MV",Rep_USR & "\\UTM2MV.txt")
WshEnv("PAC7ME") = Rep_TMP & "\\NUL.tmp"
'PAC7ME not used, on default
WshEnv("PAC7DC") = Rep_TMP & "\\NULDC.tmp"
'PAC7DC not used, on default
WshEnv("PAC7RB") = Rep_TMP & "\\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\\NULRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
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>

```







Référence : DELNT001366F - 9305

Imprimé en France