

VisualAge Pacbase



# Guide d'installation Serveur UNIX & Composants Client

*Version 3.5*





VisualAge Pacbase



# Guide d'installation Serveur UNIX & 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 section Catalogue dans la page d'accueil de la Documentation vous permet d'identifier la dernière édition disponible du présent document.

#### Cinquième édition (Novembre 2007)

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 International Business Machines Corporation 1983,2007. All rights reserved.

# Table des matières

<b>Notices</b> . . . . .	<b>vii</b>	<b>Chapitre 4. Installation/Réinstallation des composants Client</b> . . . . .	<b>29</b>
<b>Marques</b> . . . . .	<b>ix</b>	Informations préalables . . . . .	29
<b>Chapitre 1. Préambule</b> . . . . .	<b>1</b>	Répertoire racine . . . . .	29
Objet du manuel . . . . .	1	Lancement de l'installation . . . . .	30
Architecture du produit . . . . .	1	Principes et architectures de la communication . . . . .	30
Contenu de la livraison . . . . .	2	Administrator & Developer workbench . . . . .	33
Bibliographie . . . . .	3	Pour la fonction Publication : Open Jade et Tidy. . . . .	35
<b>Chapitre 2. Prérequis</b> . . . . .	<b>5</b>	Outils eBusiness . . . . .	36
Prérequis partie Serveur . . . . .	5	Station de Travail VisualAge Pacbase. . . . .	38
Matériel et logiciel . . . . .	5	Web Application Models (WAM) . . . . .	43
Espace disque. . . . .	5	Middleware . . . . .	44
Prérequis partie Client . . . . .	6	Mise à jour des paramètres de la communication . . . . .	46
Matériel. . . . .	6	Le fichier bases.ini . . . . .	46
Espace disque. . . . .	6	Le fichier vaplocat.ini . . . . .	48
Logiciel . . . . .	6	Mise à jour, Ajout, Désinstallation de composants . . . . .	53
Communications . . . . .	7	<b>Chapitre 5. Tests</b> . . . . .	<b>55</b>
<b>Chapitre 3. Installation de la partie Serveur</b> <b>9</b>	<b>9</b>	Liste des utilitaires principaux . . . . .	55
Installation . . . . .	9	Tests de l'installation . . . . .	55
Présentation générale . . . . .	9	Tests de mise à jour TP/batch, Edition-Génération . . . . .	56
Déroulement de l'installation. . . . .	9	Tests des procédures de la Base Administration . . . . .	56
Installation du Système . . . . .	10	Tests des procédures de la Base de Développement . . . . .	56
Mise à jour du fichier "/etc/services" . . . . .	10	Tests des utilitaires d'extraction . . . . .	57
Création d'un utilisateur UNIX spécifique . . . . .	11	<b>Chapitre 6. Réinstallation Serveur</b> . . . . .	<b>59</b>
Modification des fichiers de configuration . . . . .	11	Réinstallation . . . . .	59
Définition du code page . . . . .	13	Présentation générale . . . . .	59
Installation à partir d'un CDROM. . . . .	13	Lancement de la procédure de réinstallation. . . . .	59
Lancement de la procédure d'installation . . . . .	14	<b>Chapitre 7. Reprises et échanges entre bases 2.n et 3.n</b> . . . . .	<b>61</b>
Liste des éléments installés . . . . .	16	Reprises des bases des versions 2.0 et 2.5 . . . . .	61
Un élément du Système : Le Serveur TP . . . . .	16	Préambule . . . . .	61
Configuration . . . . .	16	Remarques . . . . .	61
Lancement . . . . .	16	Opérations à effectuer. . . . .	62
Description des commandes. . . . .	18	Reprise des paramètres utilisateur (PE25) . . . . .	64
Installation du Référentiel . . . . .	22		
Base Administration . . . . .	22		
Base de Développement . . . . .	23		
Saisie de la clé d'accès . . . . .	26		
Sauvegarde de la Base administration . . . . .	26		
Suppression d'une Base . . . . .	26		
Liste des dates des composants . . . . .	26		
INSL - Script d'exécution. . . . .	27		

PE25 - Présentation générale . . . . .	64	UV25 - Entrées / Traitements /	
PE25 - Entrées / Traitements /		Résultats . . . . .	118
Résultats . . . . .	65	UV25 - Description des étapes . . . . .	118
PE25 - Description des étapes . . . . .	66	UV25 - Script d'exécution . . . . .	120
PE25 - Script d'exécution . . . . .	68	Echange mouvements MB entre base 2.n	
Reprise de la Base de Développement		et 3.n (MB25) . . . . .	123
(PC25) . . . . .	71	MB25 - Présentation générale . . . . .	123
PC25 - Présentation générale . . . . .	71	MB25 - Description des étapes . . . . .	124
PC25 - Remarques sur les données		MB25 - Script d'exécution . . . . .	124
reprises . . . . .	71	Echange mouvements GY entre base 2.n	
PC25 - Entrées / Traitements /		et 3.n (GY25) . . . . .	126
Résultats . . . . .	73	GY25 - Présentation générale . . . . .	126
PC25 - Description des étapes . . . . .	73	GY25 - Description des étapes . . . . .	126
PC25 - Script d'exécution . . . . .	77	GY25 - Script d'exécution . . . . .	127
Reprise commandes d'édition-génération		Echange mouvements MB entre base 3.n	
(PG20) . . . . .	83	et 2.n (MB30) . . . . .	128
PG20 - Présentation générale . . . . .	83	MB30 - Présentation générale . . . . .	128
PG20 - Entrées / Traitements /		MB30 - Description des étapes . . . . .	129
Résultats . . . . .	84	MB30 - Script d'exécution . . . . .	129
PG20 - Description des étapes . . . . .	84	Echange mouvements GY entre base 3.n	
PG20 - Script d'exécution . . . . .	88	et 2.n (GY30) . . . . .	131
Reprise commandes d'édition-génération		GY30 - Présentation générale . . . . .	131
(PG25) . . . . .	93	GY30 - Description des étapes . . . . .	131
PG25 - Présentation générale . . . . .	93	GY30 - Script d'exécution . . . . .	132
PG25 - Entrées / Traitements /		Procédures - Tableau récapitulatif des	
Résultats . . . . .	94	changements . . . . .	133
PG25 - Description des étapes . . . . .	94	Reprises des bases de la version 3.0. . . . .	136
PG25 - Script d'exécution . . . . .	98	Opérations à effectuer . . . . .	136
Reprise mouvements PJ (PJ25) . . . . .	103	<b>Chapitre 8. Les composants . . . . .</b>	<b>137</b>
PJ25 - Présentation générale . . . . .	103	Composants de la partie Serveur. . . . .	137
PJ25 - Description des étapes . . . . .	104	Présentation générale . . . . .	137
PJ25 - Script d'exécution . . . . .	104	Documentation en ligne. . . . .	137
Reprise du fichier PP (PP25) . . . . .	105	Squelettes de génération . . . . .	137
PP25 - Présentation générale . . . . .	105	Base Administration . . . . .	138
PP25 - Entrées / Traitements /		Fichiers de la Base Administration . . . . .	138
Résultats . . . . .	106	Sauvegarde de la Base Administration . . . . .	138
PP25 - Description des étapes . . . . .	107	Base de Développement. . . . .	138
PP25 - Script d'exécution . . . . .	109	Fichiers de la Base de Développement . . . . .	138
Reprise PEI (RPP2) . . . . .	112	Fichiers de sauvegarde Base de	
RPP2 - Présentation générale . . . . .	112	Développement . . . . .	139
RPP2 - Entrées / Traitements /		Modules - fichiers spécifiques. . . . .	139
Résultats . . . . .	113	Pac/Impact . . . . .	139
RPP2 - Description des étapes. . . . .	113	DSMS. . . . .	140
RPP2 - JCL d'exécution . . . . .	116	PAF . . . . .	140
Reprise mots de passe (UTMP) . . . . .	116	Bibliothèques et fichiers	
UTMP - Présentation générale. . . . .	116	complémentaires . . . . .	140
UTMP - Description des étapes . . . . .	116	<b>Chapitre 9. Annexes . . . . .</b>	<b>143</b>
UTMP - Script d'exécution . . . . .	116		
Reprise paramètres Pac/Transfert (UV25) . . . . .	117		
UV25 - Présentation générale . . . . .	117		

Installation du Modèle de la Base	
Administration. . . . .	143
VINS - Présentation générale . . . . .	143
VINS - Entrées / Traitements / Résultats	143
VINS - Description des étapes . . . . .	144
VINS - Script d'exécution . . . . .	145
Installation du Modèle de la Base de	
Développement . . . . .	148
VINS - Présentation générale . . . . .	148
VINS - Entrées / Traitements / Résultats	148
VINS - Description des étapes . . . . .	149
VINS - Script d'exécution . . . . .	151
Utilitaires d'initialisation des fichiers	
GS/HE/ZS . . . . .	153
LDGS - Présentation générale . . . . .	153
LDGS - Description des étapes . . . . .	153
LDGS - Script d'exécution . . . . .	154
LDHE - Présentation générale. . . . .	154
LDHE - Description des étapes . . . . .	155
LDHE - Script d'exécution . . . . .	155
LDZS - Présentation générale . . . . .	155
LDZS - Description des étapes . . . . .	156
LDZS - Script d'exécution . . . . .	156
Utilitaires d'aide à la reprise . . . . .	157
UTU1 UTU2 - Adaptation des opérateurs	
'UNS' . . . . .	157
UTU1 - Extraction ligne 'P' avec 'UNS'	157
UTU1 - Entrées Utilisateur . . . . .	157
UTU1 - Description des étapes . . . . .	157
UTU1 - Script d'exécution . . . . .	158
UTU2 - Mise à jour ligne 'P' avec	
'UNS' . . . . .	158
UTU2 - Entrées Utilisateur . . . . .	159
UTU2 - Description des étapes . . . . .	159
UTU2 - Script d'exécution . . . . .	159
UTM1 UTM2 - Migration anciennes méta	
entités . . . . .	160
UTM1 - Description des étapes . . . . .	163
UTM1 - Script d'exécution . . . . .	163
UTM2 - Entrées Utilisateur. . . . .	164
UTM2 - Description des étapes . . . . .	164
UTM2 - Script d'exécution . . . . .	166





---

## 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 ou cartouche, selon l'environnement, pour l'installation de la partie Serveur,
- CD-ROM pour l'installation des composants Client,

- CD-ROM 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 UNIX
- Mémoire : Mémoire RAM (64 Mo). De la mémoire supplémentaire peut s'avérer nécessaire selon le nombre de serveurs installés sur la même machine.
- Logiciels - Runtime Cobol
  - Pour la version MICROFOCUS :  
MICRO FOCUS Application Server
  - Pour la version Acucorp :  
ACUCOBOL-GT

#### 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, la variable PATH doit être complétée par le chemin correspondant au runtime AcuCobol (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	170
Total Système	119
Total fichiers Utilisateur de tests d'installation	50
Total fichiers Administrateur de tests d'installation	1

---

## 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): 256 Mo (512 Mo conseillé).
- Logiciel : Microsoft Windows Script Host (Version 5.1 au minimum).

### Espace disque

Espace disque nécessaire à l'installation :

- 38 Mo pour Administrator & Developer workbench
- 30 Mo pour eBusiness Tools
- 68 Mo pour le JDK
- 15 Mo pour le Middleware
- 6 Mo pour WAM
- 16 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 121 Mo (38 Mo pour AD workbench, 68 Mo pour le JDK et 15 Mo pour le Middleware).

### Logiciel

Les composants Client de VisualAge Pacbase requièrent un poste équipé d'un Windows 32 bits, au choix :

- Windows 98,
- Windows/NT version 4.0 avec Service Pack 3,
- Windows 2000,
- Windows XP.

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 UNIX, le protocole de communication est TCP-IP Socket.



---

## Chapitre 3. Installation de la partie Serveur

---

### Installation

#### Présentation générale

Le produit est livré sur CDROM contenant les éléments suivants :

- PACINST.Vnn,
- PACBASE.xxx

("Vnn" représente le numéro de la version installée).

("xxx" représente la version de compilateur utilisée).

**ATTENTION :** suivant le type de support et le système UNIX, le nom des fichiers d'installation peut être en majuscules ou en minuscules.

#### Description de PACINST.Vnn :

Le fichier PACINST.Vnn contient les commandes d'installation ou de réinstallation de VA Pac.

Ces commandes sont écrites en langage shell.

#### Description de PACBASE.xxx :

Les fichiers PACBASE.xxx sont des fichiers tar compressés contenant les fichiers d'exécution et d'exploitation de VA Pac.

Lors de l'installation ou la réinstallation, le choix du Run-Time permet d'installer des fichiers compatibles avec le Run-Time utilisé.

#### Déroulement de l'installation

Il est recommandé de faire une sauvegarde générale du système avant de commencer l'installation de VA Pac et de suivre scrupuleusement les différentes étapes de l'installation :

- mise à jour du fichier "/etc/services",
- création d'un login spécifique au produit,
- création ou modification des variables d'environnement,
- installation du système, à partir du CD-ROM,
- installation de la Base d'Administration,

- installation de la Base de Développement.

Entre l'installation du système et celle des bases, on exécute le Mklink. Si celui-ci ne marche pas, il suffit alors de corriger l'erreur, de "linker" les exécutables au runtime si nécessaire. Puis de relancer l'installation des bases par l'exécution de la procédure "pacadmin" (voir le paragraphe "Installation du Référentiel dans ce chapitre) se trouvant dans le répertoire :

PACDIR/system/install

**ATTENTION :** VA Pac doit être installé sur des disques présents physiquement sur la machine UNIX et surtout pas sur des disques montés par NFS.

Pour les versions Microfocus ou Acucorp , le run-time COBOL est indispensable au fonctionnement du produit, il doit être impérativement installé avant l'installation de VisualAge Pacbase.

**ATTENTION :** Pour l'installation de la version Acucobol sur HP-UX, il est nécessaire de modifier la variable système [nlocks], "max Number of File LOCKS", pour lui donner une valeur supérieure à 9000.

---

## Installation du Système

### Mise à jour du fichier `"/etc/services"`

L'utilisation de l'interface de communication "socket" pour la communication entre le serveur et les postes clients nécessite la réservation de numéros de port de communication dans le fichier UNIX `"/etc/services"` .

Ce fichier doit être mis à jour AVANT l'installation des Bases de Développement. Dans la mesure où cette installation peut être enchaînée à l'installation du système, la mise à jour du fichier doit être effectuée AVANT l'installation du système.

Pour chaque base, un numéro de port de communication différent doit être choisi. Ce choix ne peut être arbitraire.

Un certain nombre de ports est réservé à des applications réseau "internet" standard ou réservés pour des applications ultérieures. D'autres ports sont alloués de façon dynamique.

Il est donc nécessaire de choisir un numéro de port qui n'est pas réservé dans `"/etc/services"` et qui n'est pas en cours d'utilisation c'est-à-dire alloué dynamiquement.

Exemple de mise en oeuvre :

Pour chaque serveur TP, y compris celui de la base ADMIN , un numéro de port doit être préalablement choisi. Soit 52030 le numéro de port choisi.

- le numéro de port 52030/tcp ne doit pas être dans le fichier `"/etc/services"`;

- le numéro de port ne doit pas être en cours d'utilisation. La commande UNIX suivante permet de s'en assurer :

```
netstat -an | grep 52030
```

- si les deux conditions préalables sont vérifiés, l'administrateur UNIX (root) doit mettre à jour le fichier `"/etc/services"` par la ligne suivante :

```
paclanx 52030/tcp
```

Le numéro de port choisi pour chaque serveur TP doit être soigneusement noté car cette information est nécessaire lors de la création des Bases de Développement.

## Création d'un utilisateur UNIX spécifique

La plupart des logiciels sous UNIX s'installent sous le compte d'un utilisateur spécifique.

L'intérêt est d'isoler au maximum les différents logiciels installés sur une machine, pour des raisons de sécurité.

NE PAS INSTALLER SOUS LE COMPTE UNIX "root".

L'administrateur UNIX (root) doit donc créer un utilisateur UNIX sous lequel sera installé VA Pac.

Cet utilisateur sera celui de l'administrateur VA Pac.

Il devra être autorisé à utiliser la commande 'at'.

## Modification des fichiers de configuration

Se connecter sous le login de l'administrateur VA Pac et modifier les fichiers de configuration du shell (.profile, .kshrc ou .login suivant le cas).

Les variables d'environnement PACDIR et PATH seront créées ou modifiées

```
PACDIR="$HOME/paclanx"
export PACDIR
PATH="$PACDIR/system/bin:$PACDIR/system/proc
:$COBDIR/bin:$PATH"
export PATH
```

Les variables d'environnement COBDIR, COBPATH, ACUDIR et CODE\_PREFIX doivent être ajoutées, selon le Run-Time :

Pour Microfocus :

```
COBDIR=<chemin d'accès au répertoire du Run-Time COBOL>
export COBDIR
COBPATH="$PACDIR/system/gnt"
export COBPATH
```

Pour AcuCobol :

```
ACUDIR=<chemin d'accès au répertoire du Run-Time COBOL>
export ACUDIR
CODE_PREFIX="$PACDIR/system/acu"
export CODE_PREFIX
```

Si la variable TMPDIR (répertoire temporaire du Run-Time COBOL) est assignée, le répertoire indiqué doit exister.

Pour les versions COMPAQ/TRUE64 (DIGITAL), SUN, HP-UX ou LINUX, de Microfocus, il faut créer ou modifier la variable LD\_LIBRARY\_PATH :

```
LD_LIBRARY_PATH=/usr/lib:$COBDIR/coblib:$PACDIR/system/bin
export LD_LIBRARY_PATH
```

Si vous utilisez Server Express, LD\_LIBRARY\_PATH devra contenir :

```
LD_LIBRARY_PATH=/usr/lib:$COBDIR/lib:$PACDIR/system/bin
export LD_LIBRARY_PATH
```

Si vous utilisez Acucobol, LD\_LIBRARY\_PATH devra contenir :

```
LD_LIBRARY_PATH=/usr/lib:$ACUDIR/lib:$PACDIR/system/bin
export LD_LIBRARY_PATH
```

Pour la version AIX, il faut utiliser la variable LIBPATH à la place de LD\_LIBRARY\_PATH.

Pour la version HP-UX, en plus de la variable LD\_LIBRARY\_PATH, il est nécessaire de positionner la variable SHLIB\_PATH comme suit :

```
export SHLIB_PATH=$PACDIR/system/bin:$SHLIB_PATH
```

Le Run-Time COBOL correspond au répertoire où se trouvent un certain nombre d'exécutables et de librairies nécessaires à l'exécution des programmes COBOL.



Pour Microfocus, COBDIR peut prendre en général les valeurs /usr/lib/cobol, /usr/lpp/cobol ou /opt/cobol (voir votre document d'installation du compilateur COBOL et les manuels COBOL où se trouve décrite la variable d'environnement COBDIR).

Pour Acucobol, la variable ACUDIR contient le répertoire d'installation du compilateur ou du run-time COBOL.

Une fois ces modifications effectuées, relancer l'initialisation de la configuration en tapant au clavier ".profile" et vérifier par la commande UNIX "set" la prise en compte des modifications des fichiers de configuration.

Si les variables d'environnement ne sont pas correctement initialisées, se déconnecter puis se reconnecter au compte de l'administrateur VA Pac.

### **Définition du code page**

Le code page du référentiel est un code page IBM-923 équivalent au code page ISO8859-1 avec support du caractère Euro (ISO8859-15).

#### Positionnement du code "LANG" de l'utilisateur UNIX

Afin de visualiser correctement les comptes rendus comportant des caractères nationaux, l'utilisateur devra définir un code page compatible avec celui du référentiel du produit VisualAge Pacbase.

Vous devez vous référer à votre documentation du système UNIX pour définir un code page compatible ISO8859-1 ou ISO8859-15 pour le support du caractère Euro.

En règle générale, le positionnement de la variable "LANG" suffit à définir le code page utilisé.

Par exemple pour le support des caractères nationaux français on a :

- sous AIX, SOLARIS, OSF1 :  
export LANG=fr\_FR.ISO8859-1
- sous HP-UX :  
export LANG=fr\_FR.iso8859-1

### **Installation à partir d'un CDROM**

**ATTENTION :** Monter éventuellement le driver du lecteur CDROM sur un répertoire du système.

## Lancement de la procédure d'installation

Pour assurer une mise en place correcte du logiciel, il est nécessaire de disposer d'un espace disque équivalent à cinq fois le fichier PACBASE.xxx sous le répertoire d'installation.

Après le téléchargement, sous le répertoire de connexion de l'administrateur VisualAge Pacbase, ou le "montage" du CDROM, se positionner sous le répertoire :

```
PB350"code_lot"."code_version"
```

qui se trouve dans \$HOME ou dans le répertoire de "montage" du CDROM.

Exemple sur AIX :

```
cd $HOME/PB350AIX.Vnn (nn = numéro de version)
```

et taper au clavier la commande suivante :

```
sh pacinst.Vnn (nn = numéro de version)
```

suivie d'un retour-chariot (touche Entrée ou Return).

Les étapes de la procédure d'installation sont les suivantes :

- contrôle de cohérence sur les variables d'environnement,
- affichage du menu d'installation,
- éventuellement, choix du run-time COBOL utilisé,
- choix du code langue de génération,
- création du répertoire d'installation,
- ventilation des fichiers issus de PACBASE.xxx,
- création du fichier journal\_pacinstall,
- création de la base d'Administration,
- création de la base de test livrée en standard (BVAP).

Contrôle de cohérence des variables d'environnement :

Avant de lancer la procédure PACINST.Vnn, l'Administrateur doit bien vérifier les modifications effectuées au niveau du (des) fichier(s) de configuration, même si la procédure PACINST.Vnn effectue certains contrôles de cohérence.

Affichage du menu d'installation :

Le menu d'installation a la forme suivante :

```
*****
```

```
VisualAge Pacbase
```

```
*****
inst      :  installation de la version
*****
Type the command name or 'x' to quit PACINST.Vnn :
```

description des commandes :

- La commande 'inst' déclenche l'installation.

Saisir la commande et suivre les instructions fournies par PACINST.Vnn.

#### Choix du run-time COBOL utilisé :

Cette proposition est effectuée sur certaines plateformes lors de l'installation.

Consulter l'administrateur UNIX pour connaître le run-time utilisé.

#### Choix de la langue de génération :

Le code langue de génération est "fr" pour français et "en" pour anglais.

#### Création du répertoire d'installation :

PACINST.Vnn impose le nom (pacanx) au répertoire d'installation pour deux raisons :

- bien isoler VA Pac des autres applications et donc en contrôler plus facilement l'évolution,
- permettre à PACINST.Vnn de contrôler le type de traitement (installation ou réinstallation) à effectuer.

#### Ventilation des fichiers :

Au cours de cette étape, PACINST.Vnn crée les répertoires décrits plus loin dans ce document ("Description des répertoires créés") et extrait les fichiers de PACBASE.xxx.

#### Mise à jour du fichier journal\_pacinstall :

Le fichier journal\_pacinstall contient certaines informations résultant de l'utilisation de PACINST.Vnn. Lors d'une installation ces informations sont :

- la date d'installation,
- le numéro de release,
- le numéro de version,
- le code langue de génération.

### 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.

## Liste des éléments installés

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

## Un élément du Système : Le Serveur TP

### Configuration

Lors de son lancement, chaque listener exécute un fichier de commande de type "BVPSEVER.ini" pour assigner les variables d'environnement dont il a besoin.

Ce fichier est créé lors de la création du listener sous le répertoire \$PACDIR/config/"nom\_de\_la\_base" et contient en particulier les variables d'environnement suivantes :

- BVPSOCKET : numéro du port (socket),
- SRV\_DIR : répertoire contenant les fichiers de trace du listener,
- SEMLOCK : verrou de sérialisation des mises à jour concurrentes sur la base utilisateur,
- SEMADMIN : verrou de sérialisation des mises à jour concurrentes sur la base administrateur "ADMI".

### Lancement

Le listener, dont les exécutables (pactp, bvpserver, palaunch) se trouvent dans le répertoire \$PACDIR/system/bin, permet de :

- activer/désactiver le(s) listener(s),
- fournir des informations concernant le(s) listener(s) ,

- purger une (des) station(s) rattachée(s) à un listener ,
- purger le ou les listener(s).

Pour effectuer les opérations ci-dessus on lance l'exécution de l'interpréteur de commandes du listener (pactp).

Deux modes sont possibles :

- le mode "commande",
- le mode "shell".

Le mode "commande" consiste à taper au clavier :

```
pactp <commande>
```

Pour rentrer en mode "shell" il suffit de taper au clavier :

```
pactp -s
```

L'intérêt du mode "commande" est de pouvoir insérer des commandes du listener dans un fichier de commandes.

Par exemple, le fichier com\_paclan contient les commandes suivantes :

Í affichage de l'état des listeners

```
pactp info
```

Í lancement du listener BVAP

```
pactp start BVAP
```

Í affichage d'informations sur le listener BVAP pendant dix

Í secondes

```
pactp info BVAP
```

Í arrêt du listener ADMI

```
pactp shutdown ADMI
```

Í lancement de la commande de purge de la station 003

Í dépendant du serveur BVAP

```
pactp purge BVAP 003
```

L'avantage du mode "shell" est d'éviter de relancer l'interpréteur à chaque commande, celui-ci se mettant en attente de la prochaine commande.

Les commandes disponibles sont :

- debug (activation/désactivation du mode débogueur)
- exit (sortie du mode "shell"),
- help (aide sur la syntaxe d'une commande),
- info (informations sur le(s) listener(s)),
- purge (purge du poste),

- purge\_server (purge du listener),
- shutdown (arrêt du listener sans confirmation),
- start (démarrage du listener),
- stop (arrêt du listener avec confirmation),
- view (affichage de l'état du listener).

La description détaillée des commandes se trouve ci-dessous, dans le paragraphe 'Description des commandes'.

### Conditions d'exécution

La variable d'environnement PACDIR doit être initialisée.

La variable d'environnement COBPATH doit contenir le chemin d'accès aux modules TP "\$PACDIR/system/gnt".

(voir le chapitre "INSTALLATION de la partie Serveur" - "Installation du Système" - "Modification des fichiers de Configuration").

### **Description des commandes**

#### Commande debug

Cette commande permet d'activer ou de désactiver un mode "debug" sur le fonctionnement du listener. Le nom du listener doit être passé en paramètre, suivi de "on" ou "off" pour activer ou désactiver le mode debug.

Les fichiers résultat sont de deux types :

srv[numéro\_process].txt

pour tracer le fonctionnement du listener en écoute de nouvelles connexions,  
dial[numéro\_process].txt

pour tracer le fonctionnement de chaque connexion au listener.

Ainsi à chaque connexion au listener correspond une trace.

Ces fichiers se trouvent dans le répertoire désigné par SRV\_DIR., variable d'environnement assignée dans le fichier de configuration "\$PACDIR/config/[nom\_base]/BVPSEVER.ini". Sa valeur par défaut : "\$PACDIR/data/[nom\_base]/tmp/server".

EXEMPLES :

- Lancement du mode debug sur un listener BVAP  
pactp debug BVAP on (en mode commande)

- ```

VA Pac : debug BVAP on (en mode "shell")

```
- Arrêt du mode debug sur le listener BVAP

```

pactp debug BVAP off (en mode commande)
VA Pac : debug BVAP off (en mode "shell")

```

- Différents niveaux de trace peuvent être mis en oeuvre :

#### 1- Niveau 1

trace minimum permettant de suivre le fonctionnement du listener avec les appels au moniteur de communication COBOL ,

#### 2- Niveau 2

trace détaillée du fonctionnement du listener,

#### 3- Niveau 4

trace des messages échangés entre le listener et le poste client.

La commande "debug on" met en oeuvre une trace de Niveau 1 sur un listener actif. Pour utiliser un autre niveau de trace, il est nécessaire de relancer le listener en ayant positionné la variable SRV\_TRACE dans "BVPSERVER.ini".

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

La commande "debug off" a pour effet d'arrêter la production d'une trace pour les nouvelles connexions du listener.

#### Commande exit

Cette commande permet de sortir du mode "shell" (interpréteur de commandes) précédemment activé par la commande pactp -s.

#### Commande help

Cette commande permet d'afficher l'aide concernant une commande d'administration du listener. Si on ne lui passe pas de paramètre, la liste des

commandes disponibles est affichée. Si on spécifie une commande particulière, l'aide de cette commande est affichée à l'écran.

EXEMPLE : demande d'affichage de la syntaxe de la commande

```
start
pactp help start      (en mode commande)
VA Pac : help start  (en mode "shell")
```

### Commande info

Cette commande permet d'afficher des informations sur le(s) listener(s).

Cette commande suivie du nom de la base (info "nom\_base") donne des informations suivantes :

- actif (si le listener est actif),
- inactif (si le listener est inactif),
- Error (si le listener est arrêté anormalement).

### Commande purge

Cette commande permet de purger un poste (c'est-à-dire mettre fin à la connexion d'un poste).

Si l'interpréteur (pactp) est en mode commande, la syntaxe de la commande purge est la suivante :

"pactp purge <nom\_base> <numéro du poste à purger>".

En mode "shell" il existe deux syntaxes de la commande purge :

- "purge <numéro du poste à purger>" si le prompt est différent de "VA Pac : " (le prompt prenant la valeur du nom du listener, suite par exemple à l'utilisation de la commande view),
- "purge <nom\_base> <numéro du poste à purger>" si le prompt est "VA Pac :".

### Commande purge\_server

Cette commande permet de purger un listener en cas d'anomalie, visualisée par un état "Error" lors de la commande "info". La syntaxe de la commande purge\_server est la suivante :

"pactp purge\_server <nom\_base>".

En mode "shell" la syntaxe de la commande est:



"`purge_server <nom_base>`".

N.B.: Cette commande supprime le ou les process du listener ainsi que les ressources IPC utilisées (sémaphores, mémoires partagées).

### Commande shutdown

Cette commande permet d'arrêter un listener. Le nom de la base doit être passé en paramètre de la commande.

EXEMPLE : arrêt du listener de nom BVAP

```
pactp shutdown BVAP      (en mode commande)
VA Pac  : shutdown BVAP (en mode "shell")
```

### Commande start

Cette commande permet de lancer un listener. Le nom de la base doit être passé en paramètre de la commande start. Le fichier d'initialisation BVPSEVER.ini doit être présent dans le répertoire \$PACDIR/config/"nom\_base". Le programme de lancement du listener, paclaunch, est démarré et lance le process "bvpsver". En cas de problèmes (blocage du listener), il est nécessaire dans un premier temps d'utiliser la commande :

```
pactp purge_server "nom_du_listener".
```

Si cette commande est inopérante, on peut toujours tuer le process "bvpsver" au moyen de la commande "kill -15", ou "kill -9" suivi du numéro de process (PID).

EXEMPLE : lancement du listener de nom BVAP

```
pactp start BVAP      (en mode commande)
PACBASE : start BVAP (en mode "shell")
```

### Commande stop

Cette commande permet d'arrêter un listener. Le nom de la base doit être passé en paramètre de la commande stop. Une confirmation de l'arrêt du listener est demandée.

EXEMPLE : arrêt du listener de nom BVAP

```
pactp stop BVAP      (en mode commande)
PACBASE : stop BVAP (en mode "shell")
```

### Commande view

Cette commande permet d'afficher les informations concernant un listener sur une base donnée : la liste des postes connectés, l'adresse IP du poste client, le numéro PID du process client lancé par le listener, le nom du programme COBOL exécuté et son temps d'exécution 'Elapse' en millisecondes, ainsi que des informations concernant l'état de la sémaphore permettant de synchroniser les accès concurrents à la base.

EXEMPLE : affichage d'informations du listener BVAP

```
pactp view BVAP      (en mode commande)
VA Pac  : view BVAP (en mode "shell")
```

---

## Installation du Référentiel

### Base Administration

Cette étape crée la Base d'Administration, qui contrôle les Bases de Développement. On ne peut installer qu'une seule Base d'Administration, elle se nomme "ADMI".

L'installation de la Base d'Administration se fait, soit à la suite de celle du système, soit plus tard, à l'aide de la procédure "pacadmin" se trouvant dans le répertoire \$PACDIR/system/install.

Les phases de cette installation sont les suivantes :

#### Affichage du menu d'installation :

Le menu d'installation a la forme suivante :

```
*****
                          VisualAge Pacbase
*****
  creadmin   :  creation of the administration database
*****
Type the command name or 'x' to quit pacadmin :
```

description des commandes :

- La commande 'creadmin' déclenche l'installation.

Saisir la commande et suivre les instructions fournies par pacadmin.

#### Saisie du code langue de la base :

Ce code peut être :

- en (anglais)

- fr (français)

Saisie du numéro de socket du listener :

Le numéro est compris entre 49152 et 65535.

Il doit être déclaré dans le fichier /etc/services (voir paragraphe 'MISE A JOUR DU FICHER "/etc/services").

Création sous-répertoires spécifiques de la Base :

- \$PACDIR/data/ADMI/base
- \$PACDIR/data/ADMI/journal
- \$PACDIR/data/ADMI/save
- \$PACDIR/data/ADMI/script
- \$PACDIR/data/ADMI/tmp
- \$PACDIR/data/ADMI/users
- \$PACDIR/config/ADMI

Exécution des procédures BATCH d'initialisation :

- INGU : initialisation du fichier GU,
- REST : restauration de la base,
- VINS : initialisation du méta-modèle.

INGU : initialisation du fichier des utilisateurs

Cette procédure n'est exécutée que lors de la première installation. Il 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.

Mise à jour du fichier journal\_pacinstall :

Ajout dans le fichier journal\_pacinstall des informations concernant la création de la Base :

- nom de la Base et date de création.

## **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. Le libellé saisi n'est mis à jour que si la base est nouvelle.

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.

On peut installer plusieurs Bases de Développement, chacune ayant son propre environnement.

L'installation d'une Base de Développement se fait, soit à la suite de celle du système, soit plus tard, à l'aide de la procédure "pacadmin" se trouvant dans le répertoire :

\$PACDIR/system/install.

Les phases de cette installation sont les suivantes :

Affichage du menu d'installation :

Le menu d'installation a la forme suivante :

```
*****
                          VisualAge Pacbase
*****
crebase      :  creation of a new database
*****
Type the command name or 'x' to quit pacadmin :
```

description des commandes :

- La commande 'crebase' déclenche l'installation.

Saisir la commande et suivre les instructions fournies par pacadmin.

Saisie du code de la Base (sauf pour la Base de test) :

Le code de la Base est saisi sur 4 caractères et son libellé sur 30 caractères (caractères alphanumériques et en majuscules).

La base doit être déclarée dans la Base d' Administration, sinon il y a mise à jour de ces informations pour déclaration dans la Base Administration.

Lors de l'installation de la première Base (Base de test), le code de la Base est fixe : BVAP.

#### Saisie du code langue de la Base :

Ce code peut être :

- en (anglais)
- fr (français)

#### Saisie du numéro de socket du listener :

Le numéro de socket est un nombre de 5 chiffres, compris entre 49152 et 65535.

Il faut reporter manuellement ce numéro dans le fichier vaplocat.ini (voir chapitre 4 : FICHIERS de COMMUNICATION).

Il doit être déclaré dans le fichier /etc/services (voir paragraphe 'MISE A JOUR DU FICHER "/etc/services").

#### Création sous-répertoires spécifiques de la Base :

- \$PACDIR/data/BVAP/base
- \$PACDIR/data/BVAP/journal
- \$PACDIR/data/BVAP/save
- \$PACDIR/data/BVAP/script
- \$PACDIR/data/BVAP/tmp
- \$PACDIR/data/BVAP/users
- \$PACDIR/config/BVAP

#### Exécution des procédures BATCH d'initialisation :

- UPGP : mise à jour dans la Base Administration du code et du libellé de la Base de Développement,
- REST : chargement de la Base de test,
- VINS : initialisation du méta-modèle,
- LDHE : initialisation du fichier HE,
- LDZS : initialisation du fichier ZS,

- LDGS : initialisation du fichier GS,
- INFQ : initialisation du fichier FQ,
- INFP : initialisation du fichier FP.

Mise à jour du fichier journal\_pacinstall :

Ajout dans le fichier journal\_pacinstall des informations concernant la création de la Base :

- nom de la Base et date de création.

### **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.

### **Suppression d'une Base**

Pour pouvoir supprimer une base, il faut se positionner sur le répertoire de connexion de l'utilisateur d'installation pour détruire les répertoires suivants :

PACDIR/data/[nom\_de\_base]

et

PACDIR/config/[nom\_de\_base]

On utilise la commande : `rm -r [Nom_de_Répertoire]`

### **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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) INSL BATCH PROCEDURE
# * -----
# *          VISUALAGE PACBASE
# *
# * -----
# *          - LIST OF INSTALLED PROGRAMS AND FILES -
# *
# * -----
# *
# *
# * INPUT:
# * - ONE LINE : BVP PREFIX OF THE PRODUCT
# *   (COL.3 LENGTH 3 )
# *
# * -----
# *
# Parameter control
# . $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "INSL"
echo "          ====="
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
echo ""
OUTFILE=~BVPENV INSL OUTFILE $PACUSERS/INSL.txt`
for i in `cat $PACINPUT`
do
  if [ -d "$i" ]
  then
    cd $i
    ls * >> $PACTMP/LIST
  else
    echo $i >> $PACTMP/LIST
  fi
done
LIST=$PACTMP/LIST
export LIST
```

```
LOGFILE=$PACUSERS/INSL.log
export LOGFILE
BVPMSG 1009 "bvupdate"
bvupdate $OUTFILE > $LOGFILE 2>&1
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "bvupdate"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN
```



---

## 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, une version 4.7 ou suivante de Netscape ou une version 5.5 ou suivante de Internet Explorer.
- 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 MVS 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**

La fonction Publication requiert l'installation sur votre poste de deux utilitaires "open source" : Open Jade et Tidy.

Vous pouvez les télécharger via le Support VisualAge Pacbase, à l'adresse suivante :

<http://www.ibm.com/software/awdtools/vapacbase/support.html>

Sous la rubrique "Self Help" puis "Download", cliquez sur le lien "Downloads VisualAge Pacbase en français". Suivez les indications fournies et exploitez les liens vers Open Jade et Tidy.

Installez ces deux utilitaires dans un (ou deux) répertoire(s) de votre choix, sachant que vous devrez impérativement indiquer ces deux répertoires dans des Préférences ad-hoc de AD workbench.

---

## 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".

depuis le poste utilisateur de la Station de Travail 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 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.

Le répertoire USERS doit être visible et accessible depuis le poste de l'utilisateur sur lequel fonctionne la Station de Travail. Pour ce faire, vous pouvez utiliser le Partage de Fichiers, ou encore une unité réseau accessible aux deux postes.

Le fichier GPMON.PRM doit contenir le chemin d'accès au répertoire USERS accessible en local par l'utilisateur de la Station de Travail.

#### 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=XXXXXXXX  
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
TRANSACTION 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) 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

---

### Réinstallation

#### Présentation générale

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.

D'une manière générale, seuls les fichiers du système (libellés d'erreur, squelettes de génération...) et les programmes sont touchés par une nouvelle sous-version.

#### REMARQUES :

- La procédure de réinstallation ne crée pas les répertoires, qui sont censés être identiques à ceux créés lors de la première installation de la version.
- Elle ne copie pas les procédures batch (répertoire \$PACDIR/system/proc) pour ne pas écraser d'éventuelles adaptations au site, sauf si la nouvelle sous-version peut fonctionner avec les anciennes procédures.

Les nouvelles procédures sont copiées dans le répertoire

\$PACDIR/system/proc.Vnn (nn = numéro de version)

- Les scripts de lancement opérationnels, dans PACDIR/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 les répertoires /system/install/admin/script et /system/install/base/script.

#### Lancement de la procédure de réinstallation

**ATTENTION :** Les serveurs de la Base d'Administration et des Bases de Développement doivent être arrêtés.

Après le téléchargement, sous le répertoire de connexion de l'Administrateur VisualAge Pacbase, ou le "montage" du CDROM, se positionner sous le répertoire :

```
PB300"code_matériel"."code_version"
```

Comme dans le cas d'une installation (voir le sous-chapitre "Installation de VA Pac"), taper au clavier la commande :

```
sh pacinst.Vnn (nn = numéro de version)
```

suivie d'un retour-chariot (touche Entrée ou Return).

#### Description des étapes :

Les étapes de la procédure de réinstallation sont les suivantes :

- contrôle de cohérence sur les variables d'environnement (fichier .profile, .kshrc ou .login),
- affichage du menu de réinstallation,
- choix du run-time COBOL utilisé (si nécessaire),
- ventilation des fichiers issus de PACBASE.xxx,
- mise à jour du fichier journal\_pacinstall.

#### Contrôle de cohérence des variables d'environnement :

Voir le sous-chapitre "Installation".

#### Affichage du menu de réinstallation :

Le menu de réinstallation a la forme suivante :

```
*****  
                               VisualAge Pacbase  
*****  
reinst      : version reinstallation  
*****  
Type the command name or 'x' to quit PACINST.Vnn :
```

Description des commandes :

- La commande 'reinst' permet la réinstallation complète de la nouvelle version.

Saisir la commande et suivre les instructions fournies par PACINST.Vnn.

#### Choix du run-time COBOL utilisé :

Voir le sous-chapitre "Installation".

#### Mise à jour du fichier journal\_pacinstall :

Ajout dans le fichier journal\_pacinstall des informations concernant la réinstallation :

- la date de réinstallation,
- le numéro de release,
- le numéro de version.



---

## 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 "bvptrans" 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.

#### Remarque

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.

## 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 : WGZ ou 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 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	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour

Code	Nom physique	Type	Libellé
PAC7RB	Rép. tmp : WRB	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. tmp : WRY	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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PE25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF PE FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PE25"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7EN=~BVPENV PTU920 PAC7EN `dirname $PACSAVPC`/OLDPE`
BVPMSG 1044 PE 2.5 $PAC7EN
echo "-----"
echo ""
BVPPAUSE
STATUS=~pactp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != "Inactive" -a "$STATUS" != "" ]
```



```

then
    BVPMSG 1012 "PE25"
    BVPMSG 1037 $1
    BVPERR
    exit 12
fi
BVPMKDIR
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACSAVPC.ini
PAC7EN=~BVPENV PTU920 PAC7EN `dirname $PACSAVPC`~/OLDPE~
export PAC7EN
PAC7ET=~BVPENV PTU920 PAC7ET $PACUSERS/PE25ET920.txt~
export PAC7ET
PAC7GY=~BVPENV PTU920 PAC7GY $PACTMP/WGY~
export PAC7GY
PAC7MB=$PACINPUT
export PAC7MB
BVPMSG 1009 "BVPTU920"
rtspac BVPTU920
RETURN=?
case $RETURN in
0)
;;
*)
    BVPMSG 1012 "BVPTU920"
    BVPMSG 1025
    BVPERR
    BVPRTMP
    exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=~BVPENV PAF900 PAC7GY $PACTMP/WGY~
export PAC7GY
echo " CHECKP    4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=~BVPENV PAF900 PAC7ME $PACTMP/WME~
export PAC7ME
PAC7MV=~BVPENV PAF900 PAC7MV $PACTMP/WMV~
export PAC7MV
PAC7MW=~BVPENV PAF900 PAC7MW $PACTMP/WMW~
export PAC7MW
PAC7MX=~BVPENV PAF900 PAC7MX $PACTMP/WMX~
export PAC7MX

```

```

PAC7MY=~BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=~BVPENV PACA15 PAC7IE $PACUSERS/PE25IEA15.txt`
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/PE25IFA15.txt`
export PAC7IF
PAC7ME=~BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY`
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
;;
2)
BVPMSG 1012 "BVPACA15"
BVPMSG 1054
BVPERR
BVPRMTMP
exit $RETURN
;;
4)
BVPMSG 1012 "BVPACA15"

```

```

BVPMSG 1055
BVPERR
BVPRMTMP
exit $RETURN
;;
*)
BVPMSG 1012 "BVPACA15"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## 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 : WPB	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 (+1)	Sortie	Image séquentielle du réseau (version actuelle)
PAC7ET	Rép. user : PC25ET914	État	Compte-rendu de la reprise

### **PC25 - Script d'exécution**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PC25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *
# *      RETRIEVAL OF PC FILE
# *
```

```

# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PC25"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7MC=`BVPENV PTU911 PAC7MC `dirname $PACSAVPC`/OLDPC`
BVPMSG 1044 PC 2.5 $PAC7MC
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7MC" ]
then
  BVPMSG 1034 $PAC7MC
  RETURN=1
  BVPRMTPM
  exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7ET=`BVPENV PTU911 PAC7ET $PACUSERS/PC25ET911.txt`
export PAC7ET
PAC7MB=$PACINPUT
export PAC7MB
. $PACDIR/config/$1/PACSAVPC.ini
PAC7MC=`BVPENV PTU911 PAC7MC `dirname $PACSAVPC`/OLDPC`
export PAC7MC
PAC7PB=`BVPENV PTU911 PAC7PB $PACTMP/WPC`
export PAC7PB
PAC7PE=`BVPENV PTU911 PAC7PE $PACTMP/WPE`
export PAC7PE
PAC7PF=`BVPENV PTU911 PAC7PF $PACTMP/WPF`
export PAC7PF
PAC7PG=`BVPENV PTU911 PAC7PG $PACTMP/WPG`
export PAC7PG
PAC7PL=`BVPENV PTU911 PAC7PL $PACTMP/WPL`
export PAC7PL
PAC7PM=`BVPENV PTU911 PAC7PM $PACTMP/WPM`
export PAC7PM
PAC7PZ=`BVPENV PTU911 PAC7PZ $PACTMP/WPZ`
export PAC7PZ
BVPMSG 1009 "BVPTU911"
rtspac BVPTU911
RETURN=$?
case $RETURN in
0)

```

```

;;
*)
  BVPMSG 1012 "BVPTU911"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7PB=~BVPENV PTU909 PAC7PB $PACTMP/WPC`
export PAC7PB
PAC7PI=~BVPENV PTU909 PAC7PI $PACTMP/WPI`
export PAC7PI
PAC7PL=~BVPENV PTU909 PAC7PL $PACTMP/WPL`
export PAC7PL
PAC7PM=~BVPENV PTU909 PAC7PM $PACTMP/WPM`
export PAC7PM
BVPMSG 1009 "BVPTU909"
rtspac BVPTU909
RETURN=$?
case $RETURN in
0)
  ;;
*)
  BVPMSG 1012 "BVPTU909"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----
PAC7PG=~BVPENV PTU92A PAC7PG $PACTMP/WPG`
export PAC7PG
PAC7PE=~BVPENV PTU92A PAC7PE $PACTMP/WPH`
export PAC7PE
PAC7PM=~BVPENV PTU92A PAC7PM $PACTMP/WPM`
export PAC7PM
BVPMSG 1009 "BVPTU92A"
rtspac BVPTU92A
RETURN=$?
case $RETURN in
0)
  ;;
*)
  BVPMSG 1012 "BVPTU92A"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----

```

```

PAC7PE=`BVPENV PTU92B PAC7PE $PACTMP/WPH`
export PAC7PE
PAC7PK=`BVPENV PTU92B PAC7PK $PACTMP/WPK`
export PAC7PK
BVPMSG 1009 "BVPTU92B"
rtspac BVPTU92B
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU92B"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
PAC7PB=`BVPENV PTU92C PAC7PB $PACTMP/WPC`
export PAC7PB
PAC7PK=`BVPENV PTU92C PAC7PK $PACTMP/WPK`
export PAC7PK
PAC7PD=`BVPENV PTU92C PAC7PD $PACTMP/WPD`
export PAC7PD
PAC7KP=`BVPENV PTU92C PAC7KP $PACTMP/WKP`
export PAC7KP
BVPMSG 1009 "BVPTU92C"
rtspac BVPTU92C
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU92C"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7PB=`BVPENV PTU919 PAC7PB $PACTMP/WPD`
export PAC7PB
PAC7PD=`BVPENV PTU919 PAC7PD $PACTMP/WPB`
export PAC7PD
PAC7PE=`BVPENV PTU919 PAC7PE $PACTMP/WPE`
export PAC7PE
PAC7PM=`BVPENV PTU919 PAC7PM $PACTMP/WPM`
export PAC7PM
PAC7PO=`BVPENV PTU919 PAC7PO $PACTMP/WPO`
export PAC7PO
PAC7ZP=`BVPENV PTU919 PAC7ZP $PACTMP/WEP`
export PAC7ZP

```

```

PAC7KP=~BVPENV PTU919 PAC7KP $PACTMP/WKP`
export PAC7KP
BVPMSG 1009 "BVPTU919"
rtspac BVPTU919
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU919"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7PB=~BVPENV PTU912 PAC7PB $PACTMP/WPC`
export PAC7PB
PAC7PZ=~BVPENV PTU912 PAC7PZ $PACTMP/WPZ`
export PAC7PZ
PAC7ZP=~BVPENV PTU912 PAC7ZP $PACTMP/WZP`
export PAC7ZP
BVPMSG 1009 "BVPTU912"
rtspac BVPTU912
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU912"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7PB=~BVPENV PTU913 PAC7PB $PACTMP/WPB`
export PAC7PB

PAC7PD=~BVPENV PTU913 PAC7PD $PACTMP/WPR`
export PAC7PD
PAC7PX=~BVPENV PTU913 PAC7PX $PACTMP/WEP`
export PAC7PX
PAC7PZ=~BVPENV PTU913 PAC7PZ $PACTMP/WZP`
export PAC7PZ
PAC7ZP=~BVPENV PTU913 PAC7ZP $PACTMP/WZX`
export PAC7ZP
BVPMSG 1009 "BVPTU913"
rtspac BVPTU913
RETURN=$?
case $RETURN in

```

```

0)
;;
*)
  BVPMSG 1012 "BVPTU913"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----
PAC7PZ=`BVPENV PTU91A PAC7PZ $PACTMP/WXZ`
export PAC7PZ
PAC7ZP=`BVPENV PTU91A PAC7ZP $PACTMP/WXZ`
export PAC7ZP
BVPMSG 1009 "BVPTU91A"
rtspac BVPTU91A
RETURN=$?
case $RETURN in
0)
;;
*)
  BVPMSG 1012 "BVPTU91A"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7ET=`BVPENV PTU914 PAC7ET $PACUSERS/PC25ET914.txt`
export PAC7ET
. $PACDIR/config/$1/PACSAVPC.ini
PACSAVPC=`BVPENV PTU914 PAC7PC $PACSAVPC`
PAC7PC=$PACSAVPC.NEW
export PAC7PC
PAC7PD=`BVPENV PTU914 PAC7PD $PACTMP/WPB`
export PAC7PD
PAC7PF=`BVPENV PTU914 PAC7PF $PACTMP/WPF`
export PAC7PF
PAC7PI=`BVPENV PTU914 PAC7PI $PACTMP/WPI`
export PAC7PI
PAC7PM=`BVPENV PTU914 PAC7PM $PACTMP/WPM`
export PAC7PM
PAC7PO=`BVPENV PTU914 PAC7PO $PACTMP/WPO`
export PAC7PO
PAC7ZP=`BVPENV PTU914 PAC7ZP $PACTMP/WXZ`
export PAC7ZP
BVPMSG 1009 "BVPTU914"
rtspac BVPTU914
RETURN=$?
case $RETURN in
0)
;;

```

```

*)
  BVPMSG 1012 "BVPTU914"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPMSG 1016 "PCBACKUP.ini"
sh $PACDIR/config/$1/PCBACKUP.ini
BVPRMTMP
exit $RETURN

```

## 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 : OLDGP	Entrée	Commandes édition-génération ancienne version
PAC7OU	Rép. tmp : WPG	Sortie	Commandes édition-génération reformatées (longueur=150)

Traitement des commandes d'édition génération : PTU921

Code	Nom physique	Type	Libellé
PAC7PG	Rép. tmp : WPG	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)



Code	Nom physique	Type	Libellé
PAC7GZ	Rép. tmp : WGZ	Sortie	Mouvements des lignes de Script (longueur=310)
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 : WGZ ou 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 Administration : PACA15

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

Code	Nom physique	Type	Libellé
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	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	Rép. tmp : WRB	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. tmp : WRY	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
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
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	Rép. tmp : WRB	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. tmp : WRY	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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PG20 BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# *     RETRIEVAL OF PG FILE SINCE 2.0
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
```

```

echo ""
echo "-----"
BVPMSG 1004 "PG20"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7IN=`BVPENV PTU908 PAC7IN `dirname $PACSAVPC\~/OLDPG`
BVPMSG 1044 PG 2.0 $PAC7IN
echo "-----"
echo ""
BVPPAUSE
STATUS=`pactp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != " Inactive" -a "$STATUS" != "" ]
then
    BVPMSG 1012 "PG20"
    BVPMSG 1037 $1
    BVPERR
    exit 12
fi
BVPMKDIR
if [ ! -f "$PAC7IN" ]
then
    BVPMSG 1034 $PAC7IN
    RETURN=1
    BVPRMTMP
    exit $RETURN
fi
# -----
. $PACDIR/config/$1/PACSAVPC.ini
PAC7IN=`BVPENV PTU908 PAC7IN `dirname $PACSAVPC\~/OLDPG`
export PAC7IN
PAC7OU=`BVPENV PTU908 PAC7OU $PACTMP/WPG`
export PAC7OU
BVPMSG 1009 "BVPTU908"
rtspac BVPTU908
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU908"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7ET=`BVPENV PTU921 PAC7ET $PACUSERS/PG20ET921.txt`
export PAC7ET

```

```

PAC7GY=~BVPENV PTU921 PAC7GY $PACTMP/WGY`
export PAC7GY
PAC7GZ=~BVPENV PTU921 PAC7GZ $PACTMP/WGZ`
export PAC7GZ
PAC7MB=$PACINPUT
export PAC7MB
PAC7PG=~BVPENV PTU921 PAC7PG $PACTMP/WPG`
export PAC7PG
BVPMSG 1009 "BVPTU921"
rtspac BVPTU921
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU921"
BVPMSG 1025
BVPERR
BVPMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=~BVPENV PAF900 PAC7GY $PACTMP/WGZ`
export PAC7GY
echo " CHECKP 4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=~BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=~BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=~BVPENV PAF900 PAC7MW $PACTMP/WMW`
export PAC7MW
PAC7MX=~BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7MY=~BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
;;
esac

```

```

# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AJ.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/ADMI/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/ADMI/SEMLOCK.ini
PAC7IE=~BVPENV PACA15 PAC7IE $PACUSERS/PG20IEADM.txt~
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/PG20IFADM.txt~
export PAC7IF
PAC7ME=~BVPENV PACA15 PAC7ME $PACTMP/WME~
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACTMP/WMV~
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB~
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY~
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
;;
2)
BVPMSG 1012 "BVPACA15"
BVPMSG 1054
;;
4)
BVPMSG 1012 "BVPACA15"
BVPMSG 1055
;;
*)
BVPMSG 1012 "BVPACA15"
BVPMSG 1025
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=~BVPENV PAF900 PAC7GY $PACTMP/WGY~
export PAC7GY
echo " CHECKP 4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY

```

```

PAC7ME=`BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=`BVPENV PAF900 PAC7MW $PACTMP/WMW`
export PAC7MW
PAC7MX=`BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7MY=`BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=`BVPENV PACA15 PAC7IE $PACUSERS/PG20IEA15.txt`
export PAC7IE
PAC7IF=`BVPENV PACA15 PAC7IF $PACUSERS/PG20IFA15.txt`
export PAC7IF
PAC7ME=`BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=`BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=`BVPENV PACA15 PAC7RY $PACTMP/WRY`
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
;;
2)
BVPMSG 1012 "BVPACA15"
BVPMSG 1054
BVPERR

```



```

BVPRMTMP
exit $RETURN
;;
4)
BVPMMSG 1012 "BVPACA15"
BVPMMSG 1055
BVPERR
BVPRMTMP
exit $RETURN
;;
*)
BVPMMSG 1012 "BVPACA15"
BVPMMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMMSG 1010
BVPRMTMP
exit $RETURN

```

## 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)

Code	Nom physique	Type	Libellé
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
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 : WGZ ou 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 Administration : PACA15

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

Code	Nom physique	Type	Libellé
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	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	Rép. tmp : WRB	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. tmp : WRY	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
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
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	Rép. tmp : WRB	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. tmp : WRY	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

### PG25 - Script d'exécution

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PG25 BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# *     RETRIEVAL OF PG FILE SINCE 2.5
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
```

```

echo ""
echo "-----"
BVPMSG 1004 "PG25"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7PG=~BVPENV PTU921 PAC7PG `dirname $PACSAVPC`~/OLDPG~
BVPMSG 1044 PG 2.5 $PAC7PG
echo "-----"
echo ""
BVPPAUSE
STATUS=~pactp info $1 | grep "Server Status" | cut -d: -f2~
if [ "$STATUS" != " Inactive" -a "$STATUS" != "" ]
then
    BVPMSG 1012 "PG25"
    BVPMSG 1037 $1
    BVPERR
    exit 12
fi
BVPMKDIR
if [ ! -f "$PAC7PG" ]
then
    BVPMSG 1034 $PAC7PG
    RETURN=1
    BVPRMTMP
    exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PACSAVPC.ini
PAC7ET=~BVPENV PTU921 PAC7ET $PACUSERS/PG25ET921.txt~
export PAC7ET
PAC7GY=~BVPENV PTU921 PAC7GY $PACTMP/WGY~
export PAC7GY
PAC7GZ=~BVPENV PTU921 PAC7GZ $PACTMP/WGZ~
export PAC7GZ
PAC7MB=$PACINPUT
export PAC7MB
PAC7PG=~BVPENV PTU921 PAC7PG `dirname $PACSAVPC`~/OLDPG~
export PAC7PG
BVPMSG 1009 "BVPTU921"
rtspac BVPTU921
RETURN=$?
case $RETURN in
0)
;;
*)
    BVPMSG 1012 "BVPTU921"
    BVPMSG 1025
    BVPERR
    BVPRMTMP

```

```

exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=`BVPENV PAF900 PAC7GY $PACTMP/WGZ`
export PAC7GY
echo " CHECKP    4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=`BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=`BVPENV PAF900 PAC7MW $PACTMP/WMW`
export PAC7MW
PAC7MX=`BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7MY=`BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AJ.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/ADMI/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/ADMI/SEMLOCK.ini
PAC7IE=`BVPENV PACA15 PAC7IE $PACUSERS/PG25IEADM.txt`
export PAC7IE
PAC7IF=`BVPENV PACA15 PAC7IF $PACUSERS/PG25IFADM.txt`
export PAC7IF
PAC7ME=`BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PACA15 PAC7MV $PACTMP/WMV`

```



```

export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB~
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY~
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=?
case $RETURN in
0)
;;
2)
BVPMSG 1012 "BVPACA15"
BVPMSG 1054
;;
4)
BVPMSG 1012 "BVPACA15"
BVPMSG 1055
;;
*)
BVPMSG 1012 "BVPACA15"
BVPMSG 1025
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=~BVPENV PAF900 PAC7GY $PACTMP/WGY~
export PAC7GY
echo " CHECKP 4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=~BVPENV PAF900 PAC7ME $PACTMP/WME~
export PAC7ME
PAC7MV=~BVPENV PAF900 PAC7MV $PACTMP/WMV~
export PAC7MV
PAC7MW=~BVPENV PAF900 PAC7MW $PACTMP/WMW~
export PAC7MW
PAC7MX=~BVPENV PAF900 PAC7MX $PACTMP/WMX~
export PAC7MX
PAC7MY=~BVPENV PAF900 PAC7MY $PACTMP/WMY~
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025

```

```

;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=~BVPENV PACA15 PAC7IE $PACUSERS/PG25IEA15.txt`
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/PG25IFA15.txt`
export PAC7IF
PAC7ME=~BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY`
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
;;
2)
BVPMSG 1012 "BVPACA15"
BVPMSG 1054
BVPERR
BVPRMTMP
exit $RETURN
;;
4)
BVPMSG 1012 "BVPACA15"
BVPMSG 1055
BVPERR
BVPRMTMP
exit $RETURN
;;
*)
BVPMSG 1012 "BVPACA15"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac

```

```
# -----  
BVPMSG 1010  
BVPRMTMP  
exit $RETURN
```

## 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.

#### 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
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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PJ25 BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# *     RETRIEVAL OF PJ FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PJ25"
echo "===== "
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
. $PACDIR/config/$1/PACSAVPJ.ini
PAC7PJ=~BVPENV PTU918 PAC7PJ `dirname $PACSAVPJ`/OLDPJ`
BVPMSG 1044 PJ 2.5 $PAC7PJ
```

```

echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7PJ" ]
then
  BVPMMSG 1034 $PAC7PJ
  RETURN=1
  BVPRMTMP
  exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AY.ini
PAC7ET=~BVPENV PTU918 PAC7ET $PACUSERS/PJ25ET918.txt~
export PAC7ET
. $PACDIR/config/$1/PACSAVPJ.ini
PAC7JP=~BVPENV PTU918 PAC7JP $PACSAVPJ~
export PAC7JP
PAC7PJ=~BVPENV PTU918 PAC7PJ `dirname $PACSAVPJ`~/OLDPJ~
export PAC7PJ
BVPMMSG 1009 "BVPTU918"
rtspac BVPTU918
RETURN=?
case $RETURN in
0)
  ;;
*)
  BVPMMSG 1012 "BVPTU918"
  BVPMMSG 1025
  BVPPER
  BVPRMTMP
  exit $RETURN
  ;;
esac
# -----
BVPMMSG 1010
BVPRMTMP
exit $RETURN

```

## 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

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
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	Rép. tmp : WRB	Sortie	Mouvements erronés UPDT (longueur=80)



Code	Nom physique	Type	Libellé
PAC7RY	Rép. tmp : WRY	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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PP25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF PP FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PP25"
echo "======"
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7PP=~BVPENV PTU923 PAC7PP `dirname $PACSAVPC`~/OLDPP~
BVPMSG 1044 PP 2.5 $PAC7PP
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7PP" ]
then
  BVPMSG 1034 $PAC7PP
```

```

RETURN=1
BVPRMTMP
exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACSAVPC.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
PAC7PP=~BVPENV PTU923 PAC7PP `dirname $PACSAVPC`~/OLDPP`
export PAC7PP
PAC7ET=~BVPENV PTU923 PAC7ET $PACUSERS/PP25ET923.txt`
export PAC7ET
PAC7GY=~BVPENV PTU923 PAC7GY $PACTMP/WGY`
export PAC7GY
PAC7MB=$PACINPUT
export PAC7MB
BVPMSG 1009 "BVPTU923"
rtspac BVPTU923
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU923"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=~BVPENV PAF900 PAC7GY $PACTMP/WGY`
export PAC7GY
echo " CHECKP 4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=~BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=~BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=~BVPENV PAF900 PAC7MW $PACTMP/WMW`
export PAC7MW
PAC7MX=~BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX

```

```

PAC7MY=~BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=~BVPENV PACA15 PAC7IE $PACUSERS/PP25IEA15.txt`
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/PP25IFA15.txt`
export PAC7IF
PAC7ME=~BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY`
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=?
case $RETURN in
0)
;;
2)
BVPMSG 1012 "BVPACA15"
BVPMSG 1054
BVPERR
BVPRMTMP
exit $RETURN
;;
4)
BVPMSG 1012 "BVPACA15"

```

```

BVPMMSG 1055
BVPERR
BVPRMTMP
exit $RETURN
;;
*)
BVPMMSG 1012 "BVPACA15"
BVPMMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMMSG 1010
BVPRMTMP
exit $RETURN

```

## Reprise PEI (RPP2)

### RPP2 - Présentation générale

#### Principe

Cette procédure permet de reprendre l'environnement PEI à partir d'une version 2.0 ou 2.5 et met à jour la Base de développement de la version installée.

Elle prend en compte les mouvements d'environnements pour lesquels des entités sont générées.

#### Condition d'exécution

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

#### Résultat obtenu

Une Base de développement intégrant les environnements migrés.

#### Remarques

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

L'environnement cible doit exister au préalable.

## RPP2 - Entrées / Traitements / Résultats

Une ligne \* avec Code utilisateur et Mot de passe, ainsi que les codes des environnements sources et de l'environnement cible.

Si la ligne \* est absente ou incorrecte, 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	10	1234567890	Codes sur un caractère des environnements devant être repris
29	2	EE	Code de l'environnement cible de la reprise

## RPP2 - Description des étapes

Prise en compte des entrées : PTU001

Reprise PEI étape 1 : PTU950

Code	Nom physique	Type	Libellé
			Sauvegarde fichier PEI de l'ancienne version
PAC7PP		Entrée	
			Entrée utilisateur
PAC7MB		Entrée	
			Mouvements intermédiaires pour PTU955
PAC7GY		Sortie	

Reprise PEI étape 2 : PTU955

Code	Nom physique	Type	Libellé
			Mouvements intermédiaires issus de PTU950
PAC7GY		Sortie	

Code	Nom physique	Type	Libellé
			Mouvements des environnements de production (longueur=310)
PAC7GY		Sortie	

#### 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

Code	Nom physique	Type	Libellé
PAC7AY	Rép. base : AY	Sortie	Extension de la Base de Développement
PAC7AJ	Rép. journal : AJ	Sortie	Journal de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	Rép. tmp : WRB	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. tmp : WRY	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

## RPP2 - JCL d'exécution

### 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.

##### 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

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

#### UTMP - Description des étapes

##### Reprise des mots de passe 2.N : PTURMP

Code	Nom physique	Type	Libellé
PAC7CE	Rép. save : OLDPE	Entrée	Paramètres utilisateur ancienne version
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration

#### UTMP - Script d'exécution

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTMP BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# *     PASSWORD RETRIEVAL
# *
```



```

# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTMP"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7CE=~BVPENV PTURMP PAC7CE `dirname $PACSAVPC`~/OLDPE~
BVPMSG 1044 PE 2.5 $PAC7CE
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
. $PACDIR/config/$1/PACGGU.ini
PAC7CE=~BVPENV PTURMP PAC7CE `dirname $PACSAVPC`~/OLDPE~
export PAC7CE
BVPMSG 1009 "BVPTURMP"
rtspac BVPTURMP
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTURMP"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## 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
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 : WGZ ou 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 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

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
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. tmp : WMV	Entrée	Mouvements de mise à jour
PAC7RB	Rép. tmp : WRB	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. tmp : WRY	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

```

#!/bin/sh
#@(#)VA Pac xxx xxx (R) UV25 BATCH PROCEDURE
# * -----
# *          VISUALAGE PACBASE
# *
# * -----
# *          RETRIEVAL OF UV FILE
# *
# * -----

```

```

# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UV25"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PAC7AR.ini
PAC7UV=~BVPENV PTU922 PAC7UV `dirname $PAC7AR`/OLDUV~
BVPMSG 1044 UV 2.5 $PAC7UV
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7UV" ]
then
    BVPMSG 1034 $PAC7UV
    RETURN=1
    BVPRMTMP
    exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7UV=~BVPENV PTU922 PAC7UV `dirname $PAC7AR`/OLDUV~
export PAC7UV
PAC7ET=~BVPENV PTU922 PAC7ET $PACUSERS/UV25ET922.txt~
export PAC7ET
PAC7GY=~BVPENV PTU922 PAC7GY $PACTMP/WGY~
export PAC7GY
PAC7MB=$PACINPUT
export PAC7MB
BVPMSG 1009 "BVPTU922"
rtspac BVPTU922
RETURN=$?
case $RETURN in
0)
;;
*)
    BVPMSG 1012 "BVPTU922"
    BVPMSG 1025
    BVPPER
    BVPRMTMP
    exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/ADMI/PAC7AY.ini

```

```

. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=`BVPENV PAF900 PAC7GY $PACTMP/WGY`
export PAC7GY
echo " CHECKP      4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=`BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=`BVPENV PAF900 PAC7MW $PACTMP/WMW`
export PAC7MW
PAC7MX=`BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7MY=`BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AJ.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/ADMI/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=`BVPENV PACA15 PAC7IE $PACUSERS/UV25IEA15.txt`
export PAC7IE
PAC7IF=`BVPENV PACA15 PAC7IF $PACUSERS/UV25IFA15.txt`
export PAC7IF
PAC7ME=`BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=`BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=`BVPENV PACA15 PAC7RY $PACTMP/WRY`

```

```

export PAC7RY
BVPMMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
;;
2)
BVPMMSG 1012 "BVPACA15"
BVPMMSG 1054
BVPERR
BVPRMTMP
exit $RETURN
;;
4)
BVPMMSG 1012 "BVPACA15"
BVPMMSG 1055
BVPERR
BVPRMTMP
exit $RETURN
;;
*)
BVPMMSG 1012 "BVPACA15"
BVPMMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMMSG 1010
BVPRMTMP
exit $RETURN

```

## 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**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) MB25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
```



```

# *
# * -----
# *                RETRIEVAL OF MB FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "MB25"
echo "                ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
PAC7MB=~BVPENV PTU926 PAC7MB `dirname $PACUSERS~/OLDMB`
export PAC7MB
BVPMSG 1044 MB 2.5 $PAC7MB
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7MB" ]
then
  BVPMSG 1034 $PAC7MB
  RETURN=1
  BVPRMTMP
  exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7DD=~BVPENV PTU926 PAC7DD $PACUSERS/MB25DD926.txt`
export PAC7DD
PAC7EF=~BVPENV PTU926 PAC7EF $PACUSERS/MB25EF926.txt`
export PAC7EF
PAC7MB=~BVPENV PTU926 PAC7MB `dirname $PACUSERS~/OLDMB`
export PAC7MB
PAC7ME=~BVPENV PTU926 PAC7ME $PACUSERS/MB25ME`
export PAC7ME
PAC7MV=~BVPENV PTU926 PAC7MV $PACUSERS/MB25MV`
export PAC7MV
BVPMSG 1009 "BVPTU926"
rtspac BVPTU926
RETURN=?
case $RETURN in
0)
;;
*)
  BVPMSG 1012 "BVPTU926"
  BVPMSG 1025

```

```

BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## 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

Code	Nom physique	Type	Libellé
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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) GY25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF GY FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "GY25"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
PAC7GY=~BVPENV PTU927 PAC7GY `dirname $PACUSERS`~/OLDGY~
export PAC7GY
BVPMSG 1044 GY 2.5 $PAC7GY
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7GY" ]
then
  BVPMSG 1034 $PAC7GY
```

```

RETURN=1
BVPRMTMP
exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7DD=`BVPENV PTU927 PAC7DD $PACUSERS/GY25DD927.txt`
export PAC7DD
PAC7EF=`BVPENV PTU927 PAC7EF $PACUSERS/GY25EF927.txt`
export PAC7EF
PAC7GY=`BVPENV PTU927 PAC7GY `dirname $PACUSERS`/OLDGY`
export PAC7GY
PAC7ME=`BVPENV PTU927 PAC7ME $PACUSERS/GY25ME`
export PAC7ME
PAC7MV=`BVPENV PTU927 PAC7MV $PACUSERS/GY25MV`
export PAC7MV
BVPMSG 1009 "BVPTU927"
rtspac BVPTU927
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU927"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## 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, Mot-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 : OLDMB	Entrée	Mouvements MB de la nouvelle version
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**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) MB30 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF MB FILE
# *
```

```

# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "MB30"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
PAC7MB=`BVPENV PTU928 PAC7MB `dirname $PACUSERS`/OLDMB`
export PAC7MB
BVPMSG 1044 MB 3.5 $PAC7MB
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7MB" ]
then
    BVPMSG 1034 $PAC7MB
    RETURN=1
    BVPRMTMP
    exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7DD=`BVPENV PTU928 PAC7DD $PACUSERS/MB30DD928.txt`
export PAC7DD
PAC7EF=`BVPENV PTU928 PAC7EF $PACUSERS/MB30EF928.txt`
export PAC7EF
PAC7MB=`BVPENV PTU928 PAC7MB `dirname $PACUSERS`/OLDMB`
export PAC7MB
PAC7ME=`BVPENV PTU928 PAC7ME $PACUSERS/MB30ME`
export PAC7ME
PAC7MV=`BVPENV PTU928 PAC7MV $PACUSERS/MB30MV`
export PAC7MV
BVPMSG 1009 "BVPTU928"
rtspac BVPTU928
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU928"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;

```

```

esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## 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, Mot-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
			Données de la Base de Développement
PAC7AR	Rép. base : AR	Entrée	
			Index de la Base de Développement

Code	Nom physique	Type	Libellé
PAC7AN		Entrée	
			Données extensions de la Base de Développement
PAC7AY		Entrée	
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 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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) GY30 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF GY FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "GY30"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
PAC7GY=`BVPENV PTU929 PAC7GY \dirname $PACUSERS\`OLDGY`
export PAC7GY
BVPMSG 1044 GY 3.5 $PAC7GY
echo "-----"
echo ""
BVPPAUSE
```



```

BVPMKDIR
if [ ! -f "$PAC7GY" ]
then
  BVPMSG 1034 $PAC7GY
  RETURN=1
  BVPRMTMP
  exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7DD=~BVPENV PTU929 PAC7DD $PACUSERS/GY30DD929.txt`
export PAC7DD
PAC7EF=~BVPENV PTU929 PAC7EF $PACUSERS/GY30EF929.txt`
export PAC7EF
PAC7GY=~BVPENV PTU929 PAC7GY `dirname $PACUSERS\~/OLDGY`
export PAC7GY
PAC7ME=~BVPENV PTU929 PAC7ME $PACUSERS/GY30ME`
export PAC7ME
PAC7MV=~BVPENV PTU929 PAC7MV $PACUSERS/GY30MV`
export PAC7MV
BVPMSG 1009 "BVPTU929"
rtspac BVPTU929
RETURN=?
case $RETURN in
0)
;;
*)
  BVPMSG 1012 "BVPTU929"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## 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)

<b>Procédure</b>	<b>Remarques</b>
INQJ	Initialisation du fichier journal d'archivage du module SCM
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
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
RPP2	Reprise PEI
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

Procédure	Remarques
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
UXSR	UTIXSR	Intégrée dans procédure PACS
CRYP	PACU99	
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	
SVAG	PTU550	
GET0	PACT11	
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	

Procédure	Programmes	Remarques
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. 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

<b>Fichier</b>	<b>Contenu ou format</b>	<b>Remarques</b>
BVPQCRA	Fichier séquentiel anglais	Règles standard
BVPQCRF	Fichier séquentiel français	Règles standard



---

## Chapitre 9. 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	Rép. tmp : WMX	Sortie	Entités non extraites
PAC7IA	Rép. user : VINSIA	Etat	Edition générale de l'enchaînement des programmes
PAC7EE	Rép. user : VINSEE	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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) VINS BATCH PROCEDURE
# * -----
# *      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"
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "VINS"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
. $PACDIR/config/$1/PAC7AJ.ini
BVPMSG 1015 "`dirname $PAC7AJ`"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPPAUSE
STATUS=`pactp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != "Inactive" -a "$STATUS" != "" ]
then
    BVPMSG 1012 "VINS"
    BVPMSG 1037 $1
    BVPPER
    exit 12
fi
BVPMKDIR
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
PAC7BM=`BVPENV PVINS PAC7BM $PACTMP/WBM`
export PAC7BM
PAC7DD=`BVPENV PVINS PAC7DD $PACUSERS/VINSDD.txt`
export PAC7DD
PAC7EE=`BVPENV PVINS PAC7EE $PACUSERS/VINSEE.txt`
export PAC7EE
PAC7EQ=`BVPENV PVINS PAC7EQ $PACUSERS/VINSEQ.txt`
export PAC7EQ

```

```

PAC7ER=~BVPENV PVINS PAC7ER $PACUSERS/VINSER.txt`
export PAC7ER
PAC7ES=~BVPENV PVINS PAC7ES $PACTMP/WES`
export PAC7ES
PAC7EU=~BVPENV PVINS PAC7EU $PACUSERS/VINSEU.txt`
export PAC7EU
PAC7EZ=~BVPENV PVINS PAC7EZ $PACUSERS/VINSEZ.txt`
export PAC7EZ
PAC7IA=~BVPENV PVINS PAC7IA $PACUSERS/VINSIA.txt`
export PAC7IA
PAC7IE=~BVPENV PVINS PAC7IE $PACUSERS/VINSIE.txt`
export PAC7IE
PAC7IF=~BVPENV PVINS PAC7IF $PACUSERS/VINSIF.txt`
export PAC7IF
PAC7IG=~BVPENV PVINS PAC7IG $PACUSERS/VINSIG.txt`
export PAC7IG
PAC7IH=~BVPENV PVINS PAC7IH $PACUSERS/VINSIH.txt`
export PAC7IH
. $PACDIR/config/$1/PACVINS.ini
PAC7MA=~BVPENV PVINS PAC7MA $PACVINS`
export PAC7MA
PAC7MB=$PACINPUT
export PAC7MB
PAC7MR=~BVPENV PVINS PAC7MR $PACTMP/WMR`
export PAC7MR
PAC7MX=~BVPENV PVINS PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7TD=~BVPENV PVINS PAC7TD $PACUSERS/TD.txt`
export PAC7TD
PAC7WD=~BVPENV PVINS PAC7WD $PACTMP/WWD`
export PAC7WD
SYSEXT=~BVPENV PVINS SYSEXT $PACTMP/WSY`
export SYSEXT
BVPMSG 1009 "BVPVINS"
rtspac BVPVINS
RETURN=$?
case $RETURN in
0)
;;
4)
BVPMSG 1008 $PACUSERS
;;
*)
BVPMSG 1012 "BVPVINS"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

---

## 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'



Pos.	Lon.	Valeur	Signification
		' '	'T' + '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

Code	Nom physique	Type	Libellé
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
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	Rép. tmp : WMX	Sortie	Entités non extraites
PAC7IA	Rép. user : VINSIA	Etat	Edition générale de l'enchaînement des programmes
PAC7EE	Rép. user : VINSEE	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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) VINS BATCH PROCEDURE
# * -----
# *      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"
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "VINS"
echo "======"
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
. $PACDIR/config/$1/PAC7AJ.ini
BVPMSG 1015 "`dirname $PAC7AJ`"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPVPAUSE
STATUS=`pactp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != " Inactive" -a "$STATUS" != "" ]
then
    BVPMSG 1012 "VINS"
    BVPMSG 1037 $1
```

```

        BVPERR
        exit 12
    fi
BVPMKDIR
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
PAC7BM=`BVPENV PVINS PAC7BM $PACTMP/WBM`
export PAC7BM
PAC7DD=`BVPENV PVINS PAC7DD $PACUSERS/VINSDD.txt`
export PAC7DD
PAC7EE=`BVPENV PVINS PAC7EE $PACUSERS/VINSEE.txt`
export PAC7EE
PAC7EQ=`BVPENV PVINS PAC7EQ $PACUSERS/VINSEQ.txt`
export PAC7EQ
PAC7ER=`BVPENV PVINS PAC7ER $PACUSERS/VINSER.txt`
export PAC7ER
PAC7ES=`BVPENV PVINS PAC7ES $PACTMP/WES`
export PAC7ES
PAC7EU=`BVPENV PVINS PAC7EU $PACUSERS/VINSEU.txt`
export PAC7EU
PAC7EZ=`BVPENV PVINS PAC7EZ $PACUSERS/VINSEZ.txt`
export PAC7EZ
PAC7IA=`BVPENV PVINS PAC7IA $PACUSERS/VINSIA.txt`
export PAC7IA
PAC7IE=`BVPENV PVINS PAC7IE $PACUSERS/VINSIE.txt`
export PAC7IE
PAC7IF=`BVPENV PVINS PAC7IF $PACUSERS/VINSIF.txt`
export PAC7IF
PAC7IG=`BVPENV PVINS PAC7IG $PACUSERS/VINSIG.txt`
export PAC7IG
PAC7IH=`BVPENV PVINS PAC7IH $PACUSERS/VINSIH.txt`
export PAC7IH
. $PACDIR/config/$1/PACVINS.ini
PAC7MA=`BVPENV PVINS PAC7MA $PACVINS`
export PAC7MA
PAC7MB=$PACINPUT
export PAC7MB
PAC7MR=`BVPENV PVINS PAC7MR $PACTMP/WMR`
export PAC7MR
PAC7MX=`BVPENV PVINS PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7TD=`BVPENV PVINS PAC7TD $PACUSERS/TD.txt`
export PAC7TD
PAC7WD=`BVPENV PVINS PAC7WD $PACTMP/WWD`
export PAC7WD
SYSEXT=`BVPENV PVINS SYSEXT $PACTMP/WSY`
export SYSEXT

```

```

BVPMMSG 1009 "BVPVINS"
rtspac BVPVINS
RETURN=$?
case $RETURN in
0)
;;
4)
BVPMMSG 1008 $PACUSERS
;;
*)
BVPMMSG 1012 "BVPVINS"
BVPMMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMMSG 1010
BVPRMTMP
exit $RETURN

```

---

## 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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) LDGS BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# *     INITIALIZATION OF THE GS FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "LDGS"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
echo "-----"
echo ""
BVPPAUSE
# -----
. $PACDIR/config/$1/PAC7GS.ini
BVPMSG 1009 "BVPTLDGS"
rtspac BVPTLDGS
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTLDGS"
BVPERR
exit $RETURN
;;
esac
# -----
BVPMSG 1010
exit $RETURN
```

## 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**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) LDHE BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMMSG 1004 "LDHE"
echo "          ====="
BVPMMSG 1047 "$BVPBASE"
BVPMMSG 1005 "$PACDIR/config/$1"
echo "-----"
echo ""
BVPPAUSE
# -----
. $PACDIR/config/$1/PAC7HE.ini
BVPMMSG 1009 "BVPTLDHE"
rtspac BVPTLDHE
RETURN=?
case $RETURN in
0)
;;
*)
BVPMMSG 1012 "BVPTLDHE"
BVPERR
exit $RETURN
;;
esac
# -----
BVPMMSG 1010
exit $RETURN
```

### **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**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) LDZS BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "LDZS"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
echo "-----"
echo ""
BVPPAUSE
# -----
. $PACDIR/config/$1/PAC7ZS.ini
BVPMSG 1009 "BVPTLDTS"
rtspac BVPTLDTS
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTLDTS"
BVPERR
```



```

        exit $RETURN
    ;;
esac
# -----
BVPMSG 1010
exit $RETURN

```

---

## 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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTU1 BATCH PROCEDURE
# * -----
# *          VISUALAGE PACBASE
# *
# * -----
# *          EXTRACTION OF LINES "P"
# *          WITH OPERATOR "UNS"
# *          WITHOUT LEVEL-CONDITION TYPE
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTU1"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
echo "-----"
echo ""
BVPPAUSE
BVPKDIR
# -----
UTUTO=`BVPENV PTUNS1 UTUTO `dirname $PACUSERS`/UTUTI`
export UTUTO
. $PACDIR/config/$1/PACSAVPC.ini
PAC7PC=`BVPENV PTUNS1 PAC7PC $PACSAVPC`
export PAC7PC
BVPMSG 1009 "BVPTUNS1"
rtspac BVPTUNS1
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUNS1"
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN
```

## 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**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTU2 BATCH PROCEDURE
# * -----
# *          VISUALAGE PACBASE
# *
# * -----
# *          UPDATE LINES "P"
# *          WITH OPERATOR "UNS"
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTU2"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
```

```

BVPMSG 1073 "$PACUSERS"
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
UTUTI=~BVPENV PTUNS2 UTUTI `dirname $PACUSERS`/UTUTI`
export UTUTI
. $PACDIR/config/$1/PACSAVPC.ini
PACSAVPC=~BVPENV PTUNS2 PAC7PC $PACSAVPC`
PAC7PC=$PACSAVPC
export PAC7PC
PAC7CP=$PACSAVPC.NEW
export PAC7CP
BVPMSG 1009 "BVPTUNS2"
rtspac BVPTUNS2
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUNS2"
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPMSG 1016 "PCBACKUP.ini"
sh $PACDIR/config/$1/PCBACKUP.ini
BVPRMTMP
exit $RETURN

```

## 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.

#### 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 constraint 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 en session 1 et en inter-bibliothèque.

- 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**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTM1 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      CONVERSION OF OLD META-ENTITIES
# *      ANALYSIS OF THE DATABASE RELATIONS
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTM1"
echo "====="
```

```

BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
PAC7EQ=~BVPENV PTUME1 PAC7EQ $PACUSERS/UTM1EQ.txt`
export PAC7EQ
PAC7ME=~BVPENV PTUME1 PAC7ME `dirname $PACUSERS~/UTM1ME.txt`
export PAC7ME
BVPMSG 1009 "BVPTUME1"
rtspac BVPTUME1
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUME1"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## 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	/dev/null	Sortie	Journal de la Base de Développement
PAC7AE	Système - Rép. skel : AE	Entrée	Libellés d'erreur
PACGGN	Base Admin - Rép. base : AN	Entrée	Index de la Base Administration
PACGGR	Base Admin - Rép. base : AR	Entrée	Données de la Base Administration
PACGGY	Base Admin - Rép. base : AY	Entrée	Extension de la Base Administration
PACGGU	Base Admin - Rép. base : GU	Entrée	Utilisateurs de la Base Administration
PAC7DC	Rép. base : DC	Entrée	Fichier DSMS éléments de la Base de Développement
PAC7ME	Rép. tmp : WME	Entrée	Fichier de travail
PAC7MV	Rép. user : UTM2MV	Entrée	Mouvements de mise à jour

Code	Nom physique	Type	Libellé
PAC7RB	Rép. tmp : WRB	Sortie	Mouvements erronés UPDT (longueur=80)
PAC7RY	Rép. tmp : WRY	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

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTM2 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      CONVERSION OF OLD META-ENTITIES
# *      THE RELATIONS AND THEIR CALLS
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTM2"
echo "      ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
PAC7ME=`BVPENV PTUME3 PAC7ME \dirname $PACUSERS\`/UTM1ME.txt`
export PAC7ME
BVPMSG 1009 "BVPTUME3"
```

```

rtspac BVPTUME3
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUME3"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
PAC7ET=~BVPENV PTUME2 PAC7ET $PACUSERS/UTM2ET.txt`
export PAC7ET
PAC7MV=~BVPENV PTUME2 PAC7MV $PACUSERS/UTM2MV.txt`
export PAC7MV
PAC7ME=~BVPENV PTUME2 PAC7ME `dirname $PACUSERS\~/UTM1ME.txt`
export PAC7ME
PAC7MB=$PACINPUT
export PAC7MB
BVPMSG 1009 "BVPTUME2"
rtspac BVPTUME2
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUME2"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=~BVPENV PACA15 PAC7IE /dev/null`
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/UTM2IFA15.txt`
export PAC7IF

```

```

PAC7ME=~BVPENV PACA15 PAC7ME /dev/null`
if [ "$PAC7ME" = "/dev/null" ]
then
  PAC7ME=$PACTMP/ME
  touch $PAC7ME
fi
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACUSERS/UTM2MV.txt`
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB /dev/null`
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY /dev/null`
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
  ;;
2)
  BVPMSG 1012 "BVPACA15"
  BVPMSG 1054
  BVPERR
  BVPRMTMP
  exit $RETURN
  ;;
4)
  BVPMSG 1012 "BVPACA15"
  BVPMSG 1055
  BVPERR
  BVPRMTMP
  exit $RETURN
  ;;
*)
  BVPMSG 1012 "BVPACA15"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
  ;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```





Référence : DELIX001355F - 7652

Imprimé en France