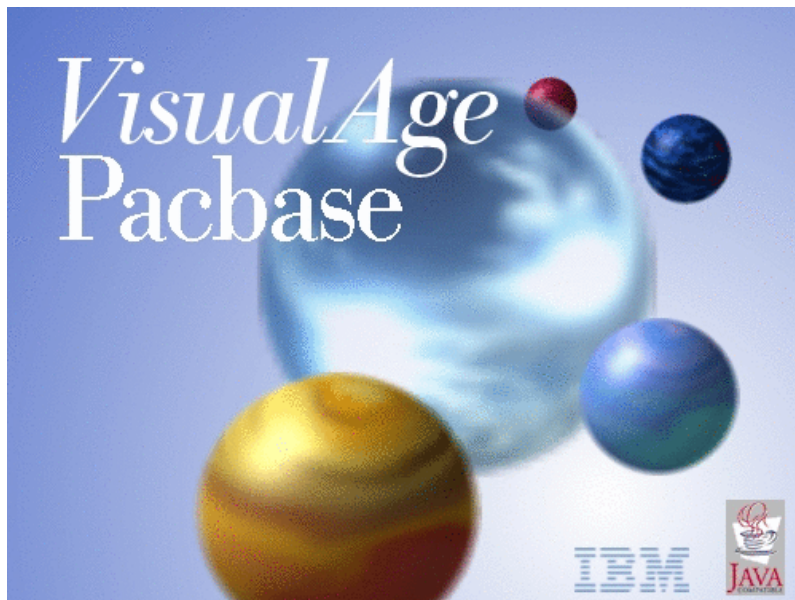


VisualAge Pacbase



Pactables - UNIX

Version 3.5



VisualAge Pacbase



Pactables - UNIX

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 v.

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.

Première édition (Juillet 2006)

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 1983,2006. All rights reserved.

Table des matières

Notices	v	Chapitre 5. Les procédures batch	29
Marques	vii	Présentation générale	29
Chapitre 1. Préambule	1	Classification des procédures	29
Chapitre 2. Les composants de Pactables	3	Anomalies d'exécution	30
Présentation générale	3	Liste des "runtime errors".	31
Les macro-structures du module TUF-TP.	3	Gestion des erreurs dans les procédures.	32
Les fichiers système	4	Lancement des procédures	32
Les fichiers évolutifs	5	Structure d'une procédure	33
Limitations standard	7	Les paramètres	33
Les modules d'accès utilisateur	7	Les variables d'environnement	34
Chapitre 3. Installation	9	Affichage et vérification des paramètres.	35
Prérequis	9	Assignation et codification des fichiers	36
Matériel et logiciel	9	Conseils d'utilisation	38
Espace disque	9	Remarques générales	38
Installation système	9	Gestion des fichiers temporaires	39
Présentation générale	9	Gestion des fichiers sauvegardés	39
Déroulement de l'installation	10	INTA - Initialisation de tables	40
Mise à jour du fichier "/etc/services"	11	INTA - Présentation générale	40
Création d'un utilisateur Unix spécifique	12	INTA - Entrée utilisateur	40
Modification des fichiers de configuration	12	INTA - Description des étapes	41
Définition du code page	13	INTA - Script d'exécution	41
Installation à partir d'un CDROM.	14	GETT - Génération de tables	42
Lancement de la procédure d'installation	14	GETT - Présentation générale	42
Liste des éléments installés	16	GETT - Description des étapes	43
Un élément du système : Le serveur TP.	17	GETT - Script d'exécution	43
Configuration	17	UPTA - Mise à jour des tables	45
Lancement	17	UPTA - Présentation générale	45
Description des commandes.	18	UPTA - Entrée utilisateur	45
Installation du référentiel	22	UPTA - Description des étapes	46
Base Pactables	22	UPTA - Script d'exécution	47
Suppression d'une base	24	PRTA - Edition des tables	49
Connexion	25	PRTA - Présentation générale	49
Tests d'utilisation	25	PRTA - Entrée utilisateur	50
Chapitre 4. Réinstallation Serveur	27	PRTA - Description des étapes	51
Réinstallation	27	PRTA - Script d'exécution	51
Présentation générale	27	IMTA - Importation de tables	54
Lancement de la procédure de réinstallation.	27	IMTA - Présentation générale	54
		IMTA - Entrée utilisateur.	55
		IMTA - Description des étapes	55
		IMTA - Script d'exécution	56
		RETA - Réorganisation des tables	58
		RETA - Présentation générale	58
		RETA - Entrée utilisateur.	59
		RETA - Description des étapes	60
		RETA - Script d'exécution	61

SVTA - Sauvegarde	64	EXTA - Extraction de tables	81
SVTA - Présentation générale	64	EXTA - Présentation générale	81
SVTA - Description des étapes	65	EXTA - Entrée utilisateur	82
SVTA - Script d'exécution	65	EXTA - Description des étapes	82
TCTA - Transposition d'une autre plateforme	67	EXTA - Script d'exécution	83
TCTA - Présentation générale	67	TUTA - Mise en exploitation	85
TCTA - Description des étapes	68	TUTA - Présentation générale	85
TCTA - Script d'exécution	69	TUTA - Entrée utilisateur	85
RSTA - Restauration	72	TUTA - Description des étapes	86
RSTA - Présentation générale	72	TUTA - Script d'exécution	86
RSTA - Description des étapes	72	Gestionnaire de tables réparties (Option	
RSTA - Script d'exécution	73	DTM)	87
LDTA - Liste descriptifs de tables	75	CDT1-CDT2 - Comparaison de descriptifs	87
LDTA - Présentation générale	75	CDT1 - Entrée utilisateur	88
LDTA - Entrée utilisateur	75	CDT1 - Description des étapes	88
LDTA - Description des étapes	75	CDT1 - Script d'exécution	89
LDTA - Script d'exécution	75	CDT2 - Description des étapes	90
PMTA - Mise à jour des paramètres	77	CDT2 - Script d'exécution	91
PMTA - Présentation générale	77	CVTA - Mise à niveau contenus tables	92
PMTA - Entrée utilisateur	77	CVTA - Entrée utilisateur	92
PMTA - Description des étapes	79	CVTA - Description des étapes	93
PMTA - Script d'exécution	79	CVTA - Script d'exécution	93

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

Conseils d'utilisation du manuel pour l'installation

Si vous disposez de la version antérieure de Pactables, l'installation de la version 3.5 doit être entièrement distincte de toute version antérieure du système quant au choix des paramètres d'installation. Elle doit être testée à l'aide du jeu d'essai fourni sur le support d'installation.

Chapitre 2. Les composants de Pactables

Présentation générale

Le module Pactables a pour but la gestion d'un certain nombre de données permanentes soit en mode conversationnel, soit en mode batch (se reporter au Manuel de Référence Pactables).

Dans ce but, il nécessite l'utilisation de deux types de ressources :

- Des répertoires dans lesquels seront stockés les programmes constitutifs du module, ainsi que les paramètres nécessaires à son fonctionnement.
- Des fichiers permanents qui matérialisent les données manipulées par le module Pactables. Ces fichiers peuvent être classés en 2 catégories :
 - Les fichiers 'système', qui restent stables lors de l'utilisation du module Pactables.
 - Les fichiers 'évolutifs', qui sont manipulés par l'utilisateur, et dont le volume varie en fonction des mises à jour effectuées.

Remarques

Le module Pactables est implanté indépendamment des autres modules de VisualAge Pacbase.

Sa mise en oeuvre nécessite la définition et la description de données dans le Dictionnaire VisualAge Pacbase. La procédure d'extraction des données définies dans le Dictionnaire et nécessaires au fonctionnement du module Pactables est décrite dans le Manuel d'Exploitation de VisualAge Pacbase 3.5.

Les macro-structures du module TUF-TP

- Les Macro-structures sont les suivantes :

Code	Signification
AATUFA	Description de la rubrique de table
AATUFL	Liste 'LT' ou 'LH'
AATUFS	Liste 'LS' ou 'LC'
AATUFU	Liste des paramètres utilisateur
AATUFX	Liste des postes d'une table

Ces Macro-structures sont destinées à être utilisées dans les programmes applicatifs transactionnels utilisateur faisant appel au module TUF-TP.

Elles permettent d'y inclure la description des zones de communications nécessaires à l'appel du sous-programme BVPFT90 du module TUF-TP.

Ces Macros-structures sont livrées sous forme de mouvements de mise à jour de VisualAge Pacbase. Elles peuvent être téléchargées via le Support VisualAge Pacbase à l'adresse suivante : <http://www.ibm.com/software/awdtools/vapacbase/support.html>

Elles doivent donc être remontées dans la bibliothèque VisualAge Pacbase utilisée pour le développement des transactions utilisateur, en prenant les mouvements en entrée des procédures UPDP ou UPDT de VisualAge Pacbase.

Les fichiers système

Ils constituent le système proprement dit. Ils ne sont pas touchés par les manipulations quotidiennes et doivent faire l'objet d'un rechargement lors d'une réimplantation. Ils comprennent :

- Les modules exécutables batch et TP
(installés sous le répertoire \$TABDIR/system/gnt)
- Un fichier contenant les libellés d'erreurs et la documentation automatique du module Pactables : TE

Caractéristique	Valeur
Taille	Environ 900 enregistrements soit environ 1,2 Méga-octets
Organisation	Indexée
Longueur	90
Clé	17 (position 1)
Utilisation	Batch et TP
Localisation	\$TABDIR/system/skel
Nom interne	PAC7TE

Remarque

\$TABDIR est un paramètre de l'installation.

Les fichiers évolutifs

Ils contiennent les informations de l'utilisateur et sont gérés par le système, soit en mode conversationnel, soit en mode batch.

Les deux premiers constituent les fichiers Pactables proprement dit :

Le fichier des descriptifs des tables (TD)

Caractéristique	Valeur
Organisation	Indexée
Longueur	240
Clé	21 (position 1)
Utilisation	Batch et TP
Localisation	\$TABDIR/data/"nom_base"/base
Nom interne	PAC7TD

Le fichier du contenu des tables (TV)

Caractéristique	Valeur
Organisation	Indexée
Longueur	80 à 1100
Clé	35 (position 5)
Utilisation	Batch et TP
Localisation	\$TABDIR/data/"nom_base"/base
Nom interne	PAC7TV

Le troisième fichier contient les 'paramètres' propres à l'utilisateur, indispensables au bon fonctionnement du système. Il est géré par une procédure batch particulière.

Le fichier des paramètres utilisateur (TG)

Caractéristique	Valeur
Organisation	Indexée
Longueur	85
Clé	22 (position 1)
Utilisation	Batch et TP
Localisation	\$TABDIR/data/"nom_base"/base

Caractéristique	Valeur
Nom interne	PAC7TG

On y trouve les codes utilisateur et leurs autorisations d'accès.

Un quatrième fichier constitue la sauvegarde des fichiers Tables utilisateur décrits ci-dessus.

La sauvegarde (TC)

Caractéristique	Valeur
Organisation	Séquentielle
Longueur	1061
Utilisation	Batch
Localisation	\$TABDIR/data/"nom_base"/save
Nom interne	PAC7TC

Remarque sur la localisation du fichier TD

Pour les sites disposant de VisualAge Pacbase et Pactables, le fichier TD est de préférence localisé avec la base de spécifications VisualAge Pacbase. C'est sous ce répertoire que les procédures d'installation de VisualAge Pacbase et Pactables assignent le fichier TD, via les fichiers PAC7TD.ini (situés sous les répertoires config de VisualAge Pacbase et de Pactables).

Les utilisateurs souhaitant déplacer le fichier TD devront modifier les deux fichiers PAC7TD.ini.

Pour les sites ne disposant que de Pactables, le fichier TD sera localisé avec les autres fichiers de la base Pactables.

Remarque

\$TABDIR et "nom_base" sont des paramètres de l'installation.

Limitations standard

Longueur maximum pour un poste de table : 999 caractères.

Longueur maximum pour la clé d'une table : 20 caractères.

Nombre maximum de rubriques d'une table : 40

Nombre de postes d'une table : illimité.

Les modules d'accès utilisateur

Des sous-programmes permettent aux applications utilisateur d'accéder au module Pactables :

Code	Signification
BVPTA800	Module d'accès optimisé
BVPTA900	Module d'accès généralisé
BVPFT00	Module TUF
BVPFT90	Module TUF

Ces programmes sont livrés sous la forme de '.gnt' compilés dans l'environnement UNIX de Pactables, qui peuvent donc être utilisés tels quels avec le même runtime Micro Focus ou AcuCobol. Ils sont installés sous le répertoire \$TABDIR/system/gnt.

Ces sous-programmes peuvent être utilisés dans les applications utilisateur batch ou conversationnelles.

Chapitre 3. Installation

Prérequis

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.

L'espace disque nécessaire à l'installation des serveurs est d'approximativement 6,5 millions d'octets.

Installation système

Présentation générale

Le produit est livré sur CDROM contenant les éléments suivants :

- TABINST.Vnnx,
- TABBASE.xxx

("Vnnx" 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 TABINST.Vnnx :

Le fichier TABINST.Vnnx contient les commandes d'installation ou de réinstallation de Pactables.

Ces commandes sont écrites en langage shell.

Description de TABBASE.xxx :

Les fichiers TABBASE.xxx sont des fichiers tar compressés contenant les fichiers d'exécution et d'exploitation de Pactables.

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

Déroulement de l'installation

Il est recommandé de faire une sauvegarde générale du système avant de commencer l'installation de Pactables 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 Pactables.

Entre l'installation du système et celle de la base Pactables 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 "tabadmin" (voir le paragraphe "Installation du Référentiel" dans ce chapitre) se trouvant dans le répertoire :

```
$TABDIR/system/install
```

ATTENTION : Pactables 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 runtime COBOL est indispensable au fonctionnement du produit, il doit être impérativement installé avant l'installation de Pactables.

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

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 Pactables. 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, un numéro de port doit être préalablement choisi. Soit 52040 le numéro de port choisi.

- le numéro de port 52040/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 52040
```

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

```
tablex 52040/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 Pactables.

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 login UNIX sous lequel sera installé Pactables.

Ce login sera celui de l'administrateur Pactables.

Modification des fichiers de configuration

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

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

```
TABDIR="$HOME/tablex"  
export TABDIR  
PATH="$TABDIR/system/bin:$TABDIR/system/proc  
:$COBDIR/bin:$PATH"  
export PATH
```

Les variables d'environnement COBDIR, COBPATH, ACUDIR et CODE_PREFIX doivent être ajoutées, selon le runtime :

Pour Microfocus :

```
COBDIR=<chemin d'accès au répertoire du runtime COBOL>  
export COBDIR  
COBPATH="$TABDIR/system/gnt"  
export COBPATH
```

Pour AcuCobol :

```
ACUDIR=<chemin d'accès au répertoire du runtime COBOL>  
export ACUDIR  
CODE_PREFIX="$TABDIR/system/acu"  
export CODE_PREFIX
```

Si la variable TMPDIR (répertoire temporaire du runtime 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:$COBDIR/coblib:$TABDIR/system/bin
export LD_LIBRARY_PATH
```

Si vous utilisez Server Express, LD_LIBRARY_PATH devra contenir :

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

Si vous utilisez Acucobol, LD_LIBRARY_PATH devra contenir :

```
LD_LIBRARY_PATH=/usr/lib:$ACUDIR/lib:$TABDIR/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=$TABDIR/system/bin:$SHLIB_PATH
```

Le runtime COBOL correspond au répertoire où se trouvent un certain nombre d'exécutables et de bibliothèques 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 runtime 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 Pactables.

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 Pactables.

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

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 TABBASE.xxx sous le répertoire d'installation.

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

```
PT350"code_lot"."code_version"
```

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

Exemple sur AIX :

```
cd $HOME/PT350AIX.Vnn (nn = numéro de version)
```

et taper au clavier la commande suivante :

```
sh tabinst.Vnnx (nn = numéro de version, x =numéro de sous-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 runtime COBOL utilisé,
- choix du code langue,
- création du répertoire d'installation,
- ventilation des fichiers issus de TABBASE.xxx,
- saisie du nom et création de la base de test,
- création du fichier journal_tabininstall.

Contrôle de cohérence des variables d'environnement :

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

Affichage du menu d'installation :

Le menu d'installation a la forme suivante :

```
*****
                          Pactables
*****
inst      : installation de la version
*****
Type the command name or 'x' to quit TABINST.Vnnx :
```

description des commandes :

- La commande 'inst' déclenche l'installation.

Saisir la commande et suivre les instructions fournies par TABINST.Vnnx.

Choix du runtime COBOL utilisé :

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

Consulter l'administrateur UNIX pour connaître le runtime utilisé.

Création du répertoire d'installation :

TABINST.Vnnx impose le nom (tablex) au répertoire d'installation pour deux raisons :

- bien isoler Pactables des autres applications et donc en contrôler plus facilement l'évolution,

- permettre à TABINST.Vnnx de contrôler le type de traitement (installation ou réinstallation) à effectuer.

Ventilation des fichiers :

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

Saisie et création de la base de test :

Le nom de la base doit être donné lors de l'installation.

Il est saisi de manière interactive en répondant aux questions concernant la base (4 caractères alphanumériques maximum).

Ce nom va permettre la création des sous-répertoires spécifiques de la base :

- \$TABDIR/config/"nom_base"
- \$TABDIR/data/"nom_base"/base
- \$TABDIR/data/"nom_base"/save
- \$TABDIR/data/"nom_base"/script
- \$TABDIR/data/"nom_base"/tmp
- \$TABDIR/data/"nom_base"/users

Mise à jour du fichier journal_tabinstall :

Le fichier journal_tabinstall contient certaines informations résultant de l'utilisation de tabinst. Lors d'une installation ces informations sont :

- la date d'installation,
- le numéro de release,
- le numéro de version,
- la date de création de la base de test.

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 programmes
- les procédures
- l'utilitaire de création de base
- les modèles des scripts de lancement

Suite à l'installation sont créés les répertoires suivants :

- \$TABDIR/config,
- \$TABDIR/data,
- \$TABDIR/system.

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 \$TABDIR/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.

Lancement

Le listener, dont les exécutables (tabtp, tabserver, tablaunch) se trouvent dans le répertoire \$TABDIR/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 (tabtp).

Deux modes sont possibles :

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

Le mode "commande" consiste à taper au clavier :

```
tabtp <commande>
```

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

```
tabtp -s
```

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

Par exemple, le fichier `com_table` contient les commandes suivantes :

```
# affichage de l'état des listeners
tabtp info
# lancement du listener PTST
tabtp start PTST
# affichage d'informations sur le listener PTST pendant dix
# secondes
tabtp info PTST
# lancement de la commande de purge de la station 003
# dépendant du serveur PTST
tabtp purge PTST 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 `TABDIR` doit être initialisée.

La variable d'environnement `COBPATH` doit contenir le chemin d'accès aux modules TP "`$(TABDIR)/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 "\$TABDIR/config/[nom_base]/BVPSEVER.ini". Sa valeur par défaut : "\$TABDIR/data/[nom_base]/tmp/server".

EXEMPLES :

- Lancement du mode debug sur un listener PTST

```
tabtp debug PTST on          (en mode commande)
Pactables : debug PTST on   (en mode "shell")
```
- Arrêt du mode debug sur le listener PTST

```
tabtp debug PTST off        (en mode commande)
Pactables : debug PTST 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 tabtp -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"

tabtp help start (en mode commande)

Pactables : 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 (tabtp) est en mode commande, la syntaxe de la commande purge est la suivante :

```
"tabtp 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 "Pactables :." (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 "Pactables :."

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 :

```
"tabtp 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 PTST

```
tabtp shutdown PTST      (en mode commande)
Pactables : shutdown PTST (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 \$TABDIR/config/"nom_base". Le programme de lancement du listener, tablaunch, est démarré

et lance le process "tabserver". En cas de problèmes (blocage du listener), il est nécessaire dans un premier temps d'utiliser la commande :

```
tabtp purge_server "nom_du_listener".
```

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

EXEMPLE : lancement du listener de nom PTST

```
tabtp start PTST      (en mode commande)
Pactables: start PTST (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 PTST

```
tabtp stop PTST      (en mode commande)
Pactables: stop PTST (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 PTST

```
tabtp view PTST      (en mode commande)
Pactables: view PTST (en mode "shell")
```

Installation du référentiel

Base Pactables

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

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

\$TABDIR/system/install.

Les phases de cette installation sont les suivantes :

Affichage du menu d'installation :

Le menu d'installation a la forme suivante :

```
*****
                          Pactables
*****
crebase      :  creation of a new database
*****
Type the command name or 'x' to quit tabadmin :
```

description des commandes :

- La commande 'crebase' déclenche l'installation.

Saisir la commande et suivre les instructions fournies par tabadmin.

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).

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

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

Création sous-répertoires spécifiques de la Base :

- \$TABDIR/data/PTST/base
- \$TABDIR/data/PTST/save

- \$TABDIR/data/PTST/script
- \$TABDIR/data/PTST/tmp
- \$TABDIR/data/PTST/users
- \$TABDIR/config/PTST

Installation de la base de test :

Deux bases de tests sont livrées : une base française livrée dans le répertoire \$TABDIR/system/install/basefra et une base anglaise livrée dans le répertoire \$TABDIR/system/install/baseeng.

Ces bases de tests contiennent :

- dans le répertoire base, les fichiers utilisateurs TD, TV et TG,
- dans le répertoire save, le fichier de sauvegarde TC,
- dans le répertoire script, les scripts de lancement des procédures.

Lors de l'installation de la base de test, ces fichiers sont copiés respectivement dans les répertoires :

- \$TABDIR/data/"nom_base"/base,
- \$TABDIR/data/"nom_base"/save,
- \$TABDIR/data/"nom_base"/script.

Mise à jour du fichier journal_tabinstall :

Ajout dans le fichier journal_tabinstall des informations concernant la création de la Base :

- nom de la Base,
- date de création.

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 :

- \$TABDIR/data/[nom_de_base]
- \$TABDIR/config/[nom_de_base]

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

Connexion

Pour que les terminaux puissent se connecter à Pactables, il faut que le listener soit lancé.

Il est possible de se connecter sur le serveur TP à l'aide d'un émulateur 3270.

Pour accéder à la base Pactables en mode 3270 via le serveur TP, il est nécessaire de configurer l'émulateur en indiquant :

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

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

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

Ces codes pages sont positionnés automatiquement, lors de la création de la base de données.

Lors du lancement de l'émulateur, le système demande de saisir un code transaction :

- TABPARM pour accéder à la transaction Pactables des paramètres utilisateur,
- TABLES pour accéder à la base Pactables de spécifications.

Tests d'utilisation

Ces tests comprennent les étapes suivantes :

- Tests d'utilisation en TP,
- Test de mise à jour, édition et réorganisation batch,
- Test de génération de tables.

Le jeu d'essai livré à l'installation comprend 3 tables :

- 'TEMPER' sans historique,
- 'CUSTOM' avec historiques du 01/03/1985 et du 10/03/1985,
- 'ARTICL' avec historique du 15/01/1987.

Tests de la transaction des paramètres utilisateur :

- Lancer la transaction Administrateur. Le code utilisateur défini dans la base de test est celui du gestionnaire : code utilisateur '*****' et mot de passe 'SUPER'.
- Utiliser ce code pour entrer dans la transaction, définir d'autres utilisateurs, puis modifier le mot de passe du code utilisateur du gestionnaire de la base.

Tests TP du module Pactables :

- Lancer un serveur TP ; y connecter un poste de travail.
- Effectuer la consultation de tous les écrans.
- Effectuer des mises à jour.

Tests batch :

- Exécuter la procédure PRTA.
- Exécuter la procédure EXTA.
- Fermer le serveur TP.
- Exécuter la procédure UPTA.
- Réorganisation des tables de test :
 - Sauvegarder TV et TD (COPY sous des noms différents).
 - Exécuter la réorganisation (RETA) qui comprend :
 - Réorganisation de TV (Prog. PTA400 et PTA410),
 - Réorganisation de TD (Prog. PTA420),
 - Constitution du fichier sauvegarde TC (PTA430),
 - Restaurer les fichiers TV et TD (RSTA),
 - Exécuter une édition (PRTA) pour vérification.
 - Lancer le serveur TP et une transaction. Effectuer quelques tests en TP pour contrôle après réorganisation.

Test de génération de tables (Procédure GETT) :

- Sortir de la transaction et fermer le serveur TP.
- Exécuter l'extraction sous VA Pac (GETA ou GETD).
- Exécuter la procédure GETT.
- Vérifier l'exécution.
- Lancer le serveur TP et une transaction. Faire quelques tests de vérification.

Chapitre 4. Réinstallation Serveur

Réinstallation

Présentation générale

Une réinstallation de la partie système de Pactables 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 le fichier système des libellés d'erreur 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 \$TABDIR/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

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

- Les scripts de lancement opérationnels, dans \$TABDIR/data/[nom_de_base]/script, ne sont pas touchés. Pour avoir la dernière version, lors de la réinstallation, il faut se référer aux modèles de scripts de lancement, dans le répertoire /system/install/basefra ou baseeng/script.

Lancement de la procédure de réinstallation

ATTENTION : Le serveur de la Base Pactables doit être arrêté.

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

```
PT350"code_matériel"."code_version"
```

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

```
sh tabinst.Vnnx (nn = numéro de version, x = numéro de sous-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 runtime COBOL utilisé (si nécessaire),
- ventilation des fichiers issus de TABBASE.xxx,
- mise à jour du fichier journal_tabinstall.

Contrôle de cohérence des variables d'environnement :

Voir le sous-chapitre "Installation système".

Affichage du menu de réinstallation :

Le menu de réinstallation a la forme suivante :

```
*****
                          Pactables
*****
reinst      : version reinstallation
*****
Type the command name or 'x' to quit TABINST.Vnnx :
```

Description des commandes :

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

Saisir la commande et suivre les instructions fournies par TABINST.Vnnx.

Choix du runtime COBOL utilisé :

Voir le sous-chapitre "Installation système".

Mise à jour du fichier journal_tabinstall :

Ajout dans le fichier journal_tabinstall des informations concernant la réinstallation :

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

Chapitre 5. Les procédures batch

Présentation générale

Les traitements BATCH associés au module Pactables sont regroupés en procédures. L'objet des chapitres suivants est de présenter chacune des procédures et d'en préciser les conditions d'exécution.

Pour chaque procédure, on trouvera :

- Une présentation générale comprenant :
 - l'objet de la procédure,
 - les conditions d'exécution,
 - les actions à entreprendre en cas d'anomalie d'exécution.
- La description des entrées utilisateur, des traitements et des résultats obtenus, ainsi que d'éventuelles recommandations d'utilisation.
- La description des étapes :
 - les notations ou paramètres utilisés,
 - la liste des fichiers utilisés (intermédiaires et permanents),
 - les codes retours éventuels émis par chaque étape.

Classification des procédures

Les procédures associées aux traitements batch sont les suivantes :

- L'initialisation des fichiers Pactables (INTA),
- La génération de tables (GETT),
- La mise à jour des tables (UPTA),
- L'édition des tables (PRTA),
- L'importation de tables (IMTA),
- La réorganisation des tables (RETA),
- La sauvegarde (SVTA),
- La restauration (RSTA),
- L'édition de listes des descriptifs de tables (LDTA),
- La mise à jour des paramètres utilisateur (PMTA),
- L'extraction des données (EXTA),
- La mise en exploitation des tables (TUTA).

Pour l'option Gestionnaire de tables réparties DTM :

- La comparaison des descriptifs (CDT1, CDT2),

- L'extraction de tables pour mise à niveau (CVTA).

Pour les migrations en provenance d'une autre plateforme :

- La reprise des fichiers sur Pactables (TCTA).

Rappel

Le module Pactables ne comporte pas de journal des mouvements de mise à jour.

Anomalies d'exécution

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

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

```

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

```

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

Quelques valeurs courantes du code opération et du status :

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

Status	Libellé
21	Erreur de séquence

Status	Libellé
22	Clé en double
23	Enregistrement non trouvé
24	Fichier trop petit
30	Erreur système
34	Fichier trop petit (séquentiel)
92	Erreur logique (par exemple, ouverture fichier déjà ouvert)
93	Fichier resté ouvert en TP
95	Fichier non défini/mal défini

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

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

```
Run Time Error nnn
```

où nnn est le numéro de l'erreur.

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

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

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

Liste des "runtime errors"

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

Numéro	Signification
-----	-----
004	Nom de fichier incorrect.
005	Désignation d'unité incorrecte.
007	Plus d'espace disque disponible.
009	Répertoire inexistant ou saturé.

```

013      Fichier non trouvé.
026      Erreur d'entrée-sortie due à un disque corrompu.
027      Unité indisponible.
028      Plus d'espace disque disponible.
033      Erreur physique d'entrée-sortie.
105      Erreur d'allocation mémoire.
116      Impossibilité d'allouer de la mémoire.
135      Fichier non trouvé.
150      Programme interrompu par l'utilisateur.
157      Mémoire insuffisante pour charger le programme.
170      Programme système non trouvé.
173      Programme appelé non trouvé.
188      Nom de fichier trop long.
198      Mémoire insuffisante pour charger le programme.
207      Machine inexistante sur le réseau.
208      Erreur réseau.
209      Erreur réseau.
221 !
222 !>  Erreur pendant un tri.
223 !

```

Gestion des erreurs dans les procédures

Dès lors qu'une erreur est détectée dans une étape, les étapes suivantes ne sont pas exécutées. On affiche alors le nom du programme en erreur et si possible le type d'erreur décelée.

La procédure affiche ensuite le message :

```
"Press Return to carry on"
```

L'utilisateur doit alors intervenir pour terminer l'arrêt de la procédure, ce qui permet de visualiser l'erreur dans le cas d'enchaînement de procédures.

(La variable d'environnement NOBVPERR positionnée à "yes" supprime l'affichage de ce message et l'intervention de l'utilisateur.)

La procédure s'arrête avec un code retour différent de zéro. Ce code est récupérable dans la variable Return tout de suite après la commande de lancement de la procédure, pour permettre d'empêcher l'exécution d'une suite si plusieurs procédures sont enchaînées.

Lancement des procédures

Les fichiers de commande des procédures sont créés sous le répertoire \$TABDIR/system/proc au moment de l'installation.

Il y a deux façons de lancer une procédure batch :

- Directement avec le fichier de commande de la procédure, suivi de paramètres :

```
procédure "code base" -i "fichier entrée utilisateur" +t
```

- u "répertoire utilisateur" -t "répertoire temporaire"
- ar l'intermédiaire d'un script de lancement :
Ce script, écrit en shell UNIX positionne des variables d'environnement (éventuellement l'entrée utilisateur) et lance le fichier de commande de la procédure.
Un exemple de script opérationnel est fourni pour la plupart des procédures et pour chaque base créée, dans le répertoire :
`$TABDIR/data/"code base"/script`

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

Structure d'une procédure

Le gestionnaire de la base est parfois amené à modifier les fichiers de commandes des procédures batch.

Par exemple, s'il souhaite séparer les fichiers TD et TV sur deux disques ou déplacer le fichier TE, les modifications induites dans les fichiers de commandes pourraient être considérables.

C'est pourquoi les procédures sont conçues pour faciliter toute modification de l'installation standard et pour minimiser les adaptations des procédures liées aux contraintes d'exploitation.

L'objet de ce sous-chapitre est d'analyser une procédure batch afin d'en expliquer le fonctionnement et de guider ainsi l'utilisateur dans ses éventuelles adaptations.

Les paramètres

- Le code de la base (sur 4 caractères).
Il est obligatoire.
- Le nom complet du fichier des entrées utilisateur :
Il est obligatoire dans le cas d'un lancement direct de la procédure.
- Le paramètre "+t" :
Il est facultatif et permet d'empêcher l'effacement par défaut des fichiers temporaires.
- Le répertoire utilisateur :
Il est facultatif et permet de remplacer l'assignation par défaut du répertoire utilisateur.
- Le répertoire temporaire :
Il est facultatif et permet de remplacer l'assignation par défaut du répertoire des fichiers temporaires.

Les variables d'environnement

- BVPINPUT :

Cette variable contient l'entrée utilisateur ; l'assignation s'effectue de la façon suivante :

```
BVPINPUT=`cat <<eof
1ere ligne utilisateur
2eme ligne utilisateur
.
.
eof`
export BVPINPUT
```

Si les caractères : \$ ` ' " sont utilisés, ils doivent être précédés de deux \.

Cette variable est sans effet si le paramètre -i est utilisé.

- TABUTI :

Cette variable contient le code utilisateur qui servira à l'assignation des répertoires "users" et "tmp". Elle est positionnée par défaut avec le code utilisateur Pactables défini dans l'entrée utilisateur.

Elle est obligatoire si l'entrée utilisateur n'est pas renseignée ou si elle ne comporte pas de code utilisateur Pactables.

L'assignation s'effectue de la façon suivante :

```
TABUTI="code utilisateur"
export TABUTI
```

- TABBASE :

Cette variable contient le code de la base, qui se substitue au code entré en paramètre de la procédure.

L'assignation s'effectue de la façon suivante :

```
TABBASE="code base"
export TABBASE
```

- NOBVPPAUSE :

Cette variable, si elle est positionnée à "yes", inhibe la suspension de l'exécution lors de l'affichage des messages d'information de la procédure.

L'assignation s'effectue de la façon suivante :

```
NOBVPPAUSE="yes"
export NOBVPPAUSE
```

- NOBVPERR :

Cette variable, si elle est positionnée à "yes", inhibe la suspension de l'exécution lors de l'affichage des messages d'erreur de la procédure.

L'assignation s'effectue de la façon suivante :

```
NOBVPERR="yes"
export NOBVPERR
```

- "procédure"_INPUT :

Cette variable permet d'indiquer le nom complet (répertoire et nom) du fichier contenant l'entrée utilisateur.

L'assignation s'effectue de la façon suivante :

```
"procédure"_INPUT="répertoire/fichier"  
export "procédure"_INPUT
```

- Des variables d'environnement permettent également de remplacer les assignations par défaut des fichiers temporaires et des fichiers utilisateurs produits (états ou fichiers résultats), soit dans toute la procédure, soit seulement dans une étape de la procédure.

L'assignation s'effectue de la façon suivante :

```
"procédure"_"code fichier"="répertoire/fichier"  
export "procédure"_"code fichier"
```

ou

```
"étape"_"code fichier"="répertoire/fichier"  
export "étape"_"code fichier"
```

Le nom des étapes et les codes fichiers sont décrits dans paragraphe "Description des étapes" de chaque procédure.

Affichage et vérification des paramètres

L'exécution d'une procédure débute par l'exécution du fichier de commande :

```
. $TABDIR/system/proc/TABINIT.ini
```

Ce fichier est créé à l'installation dans le répertoire \$TABDIR/system/proc et contrôle les paramètres en fonction de la procédure.

S'il détecte une anomalie, TABINIT.ini affiche le message d'erreur correspondant et stoppe la procédure avec un code retour égal à 20.

Sinon l'exécution de la procédure se poursuit par l'affichage des assignations des répertoires.

Afin de visualiser cet affichage, au moins pendant les tests de l'installation, l'exécution est suspendue par le message suivant :

```
***** Check your parameters *****  
Press Control_C to stop the execution  
Press Return to carry on
```

Si l'on ne souhaite pas suspendre l'exécution, il faut positionner la variable d'environnement NOBVPPAUSE à "yes".

Assignment et codification des fichiers

Chaque étape nécessite qu'on lui assigne les fichiers adéquats.

- LES FICHIERS CONSTITUANT LA BASE

Ces assignments sont effectuées via l'appel de fichiers de commandes, créés à l'installation sous le répertoire :

```
$TABDIR/config/"nom_base"
```

Exemple de l'assignment du fichier TD :

```
. $TABDIR/config/$1/PAC7TD.ini
```

L'intérêt fondamental de ces fichiers est de centraliser en un lieu unique l'assignment de chaque fichier de la base.

L'utilisateur souhaitant modifier la localisation standard d'un fichier n'aura qu'à adapter le fichier d'assignment.

Remarque : les mêmes fichiers sont utilisés lors du lancement des listeners.

- LE FICHER SAUVEGARDE

De même que pour les fichiers de la base, cette assignment est effectuée via l'appel de fichiers de commandes, créés à l'installation sous le répertoire :

```
$TABDIR/config/"nom_base"
```

Exemple de l'assignment du fichier TC :

```
. $TABDIR/config/$1/PACSAVTC.ini
```

Par défaut, le fichier TC est localisé sous le répertoire \$TABDIR/data/\$1/save.

Toutes les procédures batch qui utilisent le fichier de sauvegarde sont uniformisées quant au nom des fichiers :

```
sauvegarde en entrée (consultée) = TC
```

```
sauvegarde en sortie (créée par la procédure) = TC.NEW
```

Ceci simplifie la gestion de ce fichier (voir par exemple le paragraphe "Gestion des fichiers sauvegardes" ci-après).

- LES ETATS EN SORTIE ET LES FICHIERS RESULTATS

La localisation des états et des fichiers résultats est déterminée par l'appel du fichier de commande PACUSERS.ini :

```
. $TABDIR/config/$1/PACUSERS.ini
```

Ce fichier est créé lors de la création de chaque base dans le répertoire :

```
$TABDIR/config/'nom_base' .
```

Il contient :

```
# Command file for assignment of TABUSERS environment
# variable
# ( 'users' directory )
# Description of parameters : $0      = database name
#                                $TABUTI = Pactables user code
TABUSERS=$TABDIR/data/$1/users/$TABUTI
export TABUSERS
```

L'utilisation du paramètre -u se substitue à cette assignation par défaut.

L'exécution d'une procédure crée dans le répertoire \$TABUSERS, un sous-répertoire "code procédure"_"numéro de process".

Dans le cas de PRTA, le numéro de process est remplacé par le numéro de job.

Le nom des états en sortie commence par le code de la procédure dont ils sont issus.

Plus précisément, les états sont codifiés sur neuf caractères plus une extension (.txt), de la façon suivante :

- les 4 premiers caractères correspondent au code de la procédure,
- les 2 suivants correspondent aux 2 derniers caractères du fichier (ID dans PAC7ID),
- les 3 derniers représente les 3 derniers caractères du code programme (290 dans PTA290).

Exemple : procédure LDTA, programme PTA290

```
état PAC7ID --> LDTAID290.txt
```

Pour la codification des fichiers résultats, se référer au paragraphe "Description des étapes" de chaque procédure.

- LES FICHIERS TEMPORAIRES

La localisation des fichiers temporaires est déterminée par l'appel du fichier de commande PACTMP.ini :

```
. $TABDIR/config/$1/PACTMP.ini
```

Ce fichier est créé lors de la création de chaque base dans le répertoire :

\$TABDIR/config/'nom_base'

Il contient :

- l'assignation du répertoire des fichiers temporaires TABTMP,
- l'espace disque nécessaire pour les tris SORTFLAG,
- différentes variables spécifiques à chaque runtime (Microfocus ou AcuCobol),
- le répertoire des fichiers temporaires de tri TMPDIR

L'utilisation du paramètre -t se substitue à cette assignation par défaut.

L'exécution d'une procédure crée dans le répertoire \$TABTMP, un sous-répertoire "code procédure"_"numéro de process".

Dans le cas de PRTA, le numéro de process est remplacé par le numéro de job.

Pour la codification des fichiers temporaires, se référer au paragraphe "Description des étapes" de chaque procédure.

Conseils d'utilisation

L'objet de ce sous-chapitre est de sensibiliser le responsable de la base aux spécificités des procédures Pactables exécutées sous le système UNIX.

Remarques générales

1. Chaque procédure nécessite qu'on lui transmette des paramètres. Tous les paramètres prévus à l'appel d'une procédure sont obligatoirement présents, même s'ils ne sont pas utilisés.
2. Lorsqu'une entrée utilisateur est prévue dans une procédure, même si elle est facultative, le fichier mouvement correspondant doit être présent lors de l'exécution de la procédure.
Pour une entrée utilisateur directement saisie dans le script, si les caractères : \$ ` ' " sont utilisés, ils doivent être précédés de deux \.
3. Aucune protection n'est assurée dans le cas où une procédure BATCH mettant à jour les fichiers système ou évolutifs de la base est lancée pendant que des utilisateurs mettent à jour de manière interactive ces mêmes fichiers. Une seule personne (le gestionnaire de la base) doit avoir la possibilité de lancer des procédures batch mettant à jour les bases. Il doit donc assurer lui-même la protection des données de la base (en fermant les serveurs TP par exemple).
4. Les fichiers temporaires de travail créés par les procédures batch sont automatiquement détruits en fin de procédure, sauf si une étape se déroule mal et envoie un code retour différent de 0.

5. Les procédures batch doivent être soumises à partir de la machine UNIX.
6. La présence de caractères spéciaux dans les codes entités est fortement déconseillée, par exemple le caractère EURO. Ce dernier constitue un réel problème sur ACU.

Gestion des fichiers temporaires

L'utilisateur se reportera, pour chaque procédure, au chapitre correspondant pour une description détaillée de ces fichiers.

Dans tous les cas, il faut veiller à libérer suffisamment d'espace disque, sous le répertoire utilisateur choisi, pour un bon déroulement de la procédure.

Fichiers temporaires de tri :

Lorsqu'un programme exécute un tri, les routines COBOL appelées utilisent également un fichier temporaire, indépendant de ceux cités ci-dessus.

Ce fichier est créé par défaut dans le répertoire /usr/tmp.

Sa taille peut être égale à 3 ou 4 fois celle du fichier à trier.

Si le répertoire par défaut est trop petit, la variable TMPDIR permet d'assigner un autre répertoire pour les fichiers temporaires de tri :

```
TMPDIR=/tmp2
export TMPDIR
```

Gestion des fichiers sauvegardes

Toutes les procédures créant une sauvegarde appellent un fichier de commandes en fin de procédure sans erreur.

Ce fichier est localisé sous le répertoire \$TABDIR/ config/"nom_base" et est nommé TCBACKUP.ini. Il est créé à la création de la base et contient :

```
Script de rotation des fichiers de sauvegarde de la base
. $TABDIR/config/PTST/PACSAVTC.ini
if [ -f "$PACSAVTC" ]
then
    mv -f $PACSAVTC $PACSAVTC'-1'
fi
mv -f $PACSAVTCNEW $PACSAVTC
```

Caractéristiques du fichier TCBACKUP :

- procède par 'mv' pour éviter des copies pouvant être très longues, des fichiers sauvegardes,

- garantit que le fichier TC est bien la dernière sauvegarde.

Ce fichier ne prétend pas couvrir toutes les contraintes d'exploitation de tous les sites. Le gestionnaire de la base devra en général l'adapter, en tenant compte des caractéristiques ci-dessus.

INTA - Initialisation de tables

INTA - Présentation générale

Cette procédure permet l'initialisation des fichiers contenant les descriptifs et contenus des Tables.

Important :

Cette procédure est donc utilisée pour l'initialisation de nouveaux fichiers physiques, et non pour initialiser de nouvelles tables dans un contexte existant (pour cette dernière opération, se reporter au chapitre "Les procédures batch", sous-chapitre "GETT- Génération de tables").

INTA - Entrée utilisateur

Pos.	Lon.	Valeur	Signification
1	36		Libellé de l'installation
37	1		Code langue
		'F'	Français
		'E'	Anglais
38	1		Inutilisé
39	12		Signification des touches fonctions
51	4	cccc	Classe pour système de sécurité
55	1		Type de système de sécurité
		' '	Pas de système de sécurité
		'R'	RACF
		'S'	TOP SECRET
56	2	nn	Nombre de lignes par page d'édition
58	1		Type de contrôle de ressource
		' '	Définition ressources tables système de sécurité
		'P'	Définition ressources dans VA Pac
59	1		Blocage du code utilisateur
		' '	Possibilité autre code utilisateur
		'N'	Interdiction autre code utilisateur

INTA - Description des étapes

Initialisation des fichiers : PTAINI

Code	Nom physique	Type	Libellé
PAC7MD		Entrée	Fichier mouvement
PAC7TD	Rép. base : TD	Sortie	Fichier de description de tables
PAC7TV	Rép. base : TV	Sortie	Fichier des contenus de tables
PAC7ED	Rép. user : INTAEDINI.txt	Etat	Compte-rendu d'initialisation

INTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) INTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - TABLE INITIALIZATION -
# *
# * -----
# *
# * INPUT
# * COL 1-36   : INSTALLATION LABEL
# * COL 37    : LANGUAGE VERSION PARAMETER
# *           "E" ENGLISH
# *           "F" FRENCH
# * COL 38    : NOT USED
# * COL 39-50 : FUNCTION KEYS ASSIGNMENTS
# * COL 51-54 : "CCCC" SECURITY SYSTEM CLASS
# * COL 55    : SECURITY SYSTEM TYPE
# *           BLANK : NOT USED
# *           "R"   RACF
# *           "S"   TOP SECRET
# * COL 56-57 : "NN" NUMBER OF LINES PER PRINTOUT PAGE
# * COL 58    : TYPE OF RESOURCE CONTROL
# *           BLANK : TABLE RESOURCES SECURITY SYSTEM
# *           "P"   : RESOURCE IN VA PAC
# * COL 59    : LOCK OF USER'S CODE
# *           : BLANK : OTHER USER'S CODE AUTHORIZED
# *           : "N"   : OTHER USER'S CODE UNAUTHORIZED
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "INTA"
echo "=====
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
```

```

TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7MD=$TABINPUT
export PAC7MD
PAC7ED=`TABENV BVPTAINI PAC7ED $TABUSERS/INTAEDINI.txt`
export PAC7ED
TABMSG 1009 "BVPTAINI"
rtstab BVPTAINI
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTAINI"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

GETT - Génération de tables

GETT - Présentation générale

Cette procédure reprend les descriptions des tables extraites de la base VisualAge Pacbase pour mettre à jour le fichier des descriptifs de tables et initialiser les tables générées dans le fichier du contenu des tables.

Condition d'exécution

Cette procédure doit nécessairement être précédée par la procédure d'extraction sous VisualAge Pacbase (GETD ou GETA), qui lui fournit le fichier des descriptions extraites en entrée.

Entrée utilisateur

Résultat d'extraction de GETD ou GETA.

GETT - Description des étapes

Mise à jour des fichiers tables : PTA250

Code	Nom physique	Type	Libellé
PAC7MD		Entrée	Fichier mouvement issu de GETD ou GETA
PAC7TD	Rép. base : TD	Entrée Sortie	Fichier de description de tables
PAC7TV	Rép. base : TV	Entrée Sortie	Fichier des contenus de tables
PAC7TK	Rép. tmp : TK	Sortie	Fichier en sortie
PAC7ET	Rép. user : GETTET250.txt	Etat	Erreur entrée/sortie sur fichiers

Edition des descriptifs : PTA290

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base: TD	Entrée	Fichier de description de tables
PAC7TE	Rép. tmp : TK	Entrée	Fichier mouvement de demande d'édition
PAC7ID	Rép. user : GETTID290.txt	Etat	Edition des descriptifs

GETT - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) GETT BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *                   - TABLE GENERATION -
# *
# * -----
# *
# * INPUT : RESULT OF GETA OR GETD PROCEDURE
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "GETT"
echo "
=====
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
```

```

echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7MD=$TABINPUT
export PAC7MD
PAC7TK=`TABENV BVPTA250 PAC7TK $TABTMP/TK`
export PAC7TK
PAC7ET=`TABENV BVPTA250 PAC7ET $TABUSERS/GETTET250.txt`
export PAC7ET
TABMSG 1009 "BVPTA250"
rtstab BVPTA250
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA250"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
PAC7TE=`TABENV BVPTA290 PAC7TE $TABTMP/TK`
export PAC7TE
PAC7ID=`TABENV BVPTA290 PAC7ID $TABUSERS/GETTID290.txt`
export PAC7ID
TABMSG 1009 "BVPTA290"
rtstab BVPTA290
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA290"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

UPTA - Mise à jour des tables

UPTA - Présentation générale

Cette procédure effectue la mise à jour batch des tables et l'édition des tables mises à jour.

Condition d'exécution

Aucune

Note importante

Une deuxième version du programme de mise à jour PTA302 est fournie à partir de la version 2.0.

Lors des mises à jour, le programme PTA300 appelle éventuellement les sous-programmes de contrôle utilisateur pour effectuer des contrôles complémentaires. L'option de génération de ces sous-programmes par défaut est sans gestion de siècle.

A partir de la version 2.0, si les sous-programmes de contrôle utilisateur sont générés avec l'option gestion du siècle, le nouveau programme PTA302 fourni doit être renommé et utilisé en lieu et place du programme PTA300.

Dans tous les cas, TOUS les sous-programmes de contrôle utilisateur doivent être générés avec la même option.

UPTA - Entrée utilisateur

- Une ligne '*' par utilisateur :

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

- Une ligne 'A' par table à mettre à jour :

Pos.	Lon.	Valeur	Signification
2	1	'A'	Code carte
3	6	tttttt	Numéro de la table
9	8	JJMMSSAA	Date de l'historique
17	1		Inutilisé
18	1		Numéro du sous-système

Pos.	Lon.	Valeur	Signification
		' '	Pas de sous-système précisé
		1 à 0	Numéro de sous-système
19	1		Délimiteur des données
		' '	Pris comme '/' par défaut

- Des lignes 'V' de données de la table à mettre à jour :

Pos.	Lon.	Valeur	Signification
1	1		Code mouvement
		'C'	Création
		'M'	Modification
		'A'	Annulation
2	1	'V'	Code carte
3	1		Continuation des données
		' '	Première ligne des données
		'-'	Suite des données du poste
4	77		Données de la table séparées par le délimiteur indiqué au niveau de la ligne 'A'

UPTA - Description des étapes

Mise à jour des tables : PTA300

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier de description de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreurs
PAC7TG	Rép. base : TG	Entrée	Fichier des paramètres utilisateur
PAC7TV	Rép. base : TV	Entrée Sortie	Fichier des contenus de tables
PAC7MS		Entrée	Mouvements de mise à jour
PAC7DE	Rép. tmp : DE	Sortie	Demande d'édition
PAC7ET	Rép. user : UPTAET300.txt	Etat	Compte-rendu des mouvements
PAC7MT	Rép. tmp : MT		Mouvements mis en forme

Mise en forme de l'édition : PTA350

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier de description de tables
PAC7TV	Rép. base : TV	Entrée	Fichier des contenus de tables
PAC7DE	Rép. tmp : DE	Entrée	Demande d'édition
PAC7ET	Rép. tmp : ET	Sortie	Fichier d'édition
PAC7EX	Rép. user : UPTAEX350.txt	Etat	Statistiques d'édition

Edition : PTA360

Code	Nom physique	Type	Libellé
PAC7TD	Rép base : TD	Entrée	Fichier de description de tables
PAC7ET	Rép. tmp : ET	Entrée	Fichier d'édition
PAC7EY	Rép. user : UPTAEY360.txt	Etat	Edition des tables

UPTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) UPTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *               - TABLE UPDATE -
# *
# * -----
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE
# * COL 11-18  : PASSWORD
# *
# * --> ONE "A" TYPE LINE PER TABLE TO UPDATE
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
# * COL 17     : NOT USED
# *
# * COL 18     : SUB-SYSTEM NUMBER
# *           : " " : NO SUB-SYSTEM SPECIFIED
# *           : 1 TO 0 : SUB-SYSTEM NUMBER
# * COL 19     : DATA DELIMITER
# *           : " " : CONSIDER AS "/" BY DEFAULT
# * --> ONE "V" TYPE LINE TO UPDATE TABLE DATA
# * COL 1      : ACTION CODE
# *           : "C" : CREATION
```

```

# *           : "M" : MODIFICATION
# *           : "D" : DELETION
# * COL 2     : "V" : LINE CODE
# * COL 3     : CONTINUATION LINE
# *           : " " : FIRST DATA LINE
# *           : "-" : ITEM DATA CONTINUATION
# * COL 4-80  : TABLE DATA SEPARATED BY THE DELIMITER
# *           : INDICATED ON THE "A" TYPE LINE
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "UPTA"
echo "          ====="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7MS=$TABINPUT
export PAC7MS
PAC7DE=`TABENV BVPTA300 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7MT=`TABENV BVPTA300 PAC7MT $TABTMP/MT`
export PAC7MT
PAC7ET=`TABENV BVPTA300 PAC7ET $TABUSERS/UPTAET300.txt`
export PAC7ET
TABMSG 1009 "BVPTA300"
rtstab BVPTA300
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA300"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7DE=`TABENV BVPTA350 PAC7DE $TABTMP/DE`

```

```

export PAC7DE
PAC7ET=`TABENV BVPTA350 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EX=`TABENV BVPTA350 PAC7EX $TABUSERS/UPTAEX350.txt`
export PAC7EX
TABMSG 1009 "BVPTA350"
rtstab BVPTA350
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA350"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
PAC7ET=`TABENV BVPTA360 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EY=`TABENV BVPTA360 PAC7EY $TABUSERS/UPTAEY360.txt`
export PAC7EY
TABMSG 1009 "BVPTA360"
rtstab BVPTA360
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA360"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

PRTA - Edition des tables

PRTA - Présentation générale

Cette procédure effectue l'édition batch des tables.

Condition d'exécution

Cette procédure consulte les fichiers Pactables ; elle peut être exécutée même si l'accès au TP reste ouvert.

Remarque

Cette procédure peut être lancée directement en TP.

(Voir dans le Manuel de Référence Pactables la description des écrans de lancement des éditions batch).

PRTA - Entrée utilisateur

- Une ligne '*' par utilisateur :

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

- Une ligne 'A' par table à éditer :

Pos.	Lon.	Valeur	Signification
1	1		Code mouvement
		'E'	Edition de la table
		'H'	Liste des historiques
		'L'	Liste des tables
		'S'	Liste des sous-schémas et sous-systèmes
		'X'	Liste des postes avec historiques
2	1	'A'	Code carte
3	6	tttttt	Numéro de la table
9	8	JJMMSSAA	Date de l'historique ou date de descriptif de référence (si code mouvement 'X')
17	1		Sélection du sous-schéma
		' '	Pas de sélection de sous-schéma
		1 à 0	Numéro du sous-schéma sélectionné
18	1		Sélection du sous-système
		' '	Pas de sélection de sous-système
		1 à 0	Numéro du sous-système sélectionné
19	1		Option d'édition des rubriques de la clé
		' '	Edition des rubriques groupées
		'O'	Edition des rubriques séparées

PRTA - Description des étapes

Extraction des tables à éditer : PTA320

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier de description de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreurs
PAC7TV	Rép. base : TV	Entrée	Fichier des contenus de tables
PAC7TG	Rép. base : TG	Entrée	Fichier des paramètres utilisateur
PAC7CA		Entrée	Mouvements de demande d'édition
PAC7DE	Rép. tmp : DE	Sortie	Demandes d'édition
PAC7XE	Rép. user : PRTAXE320.txt	Etat	Compte-rendu des mouvements

Mise en forme de l'édition : PTA350

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier de description de tables
PAC7TV	Rép. base : TV	Entrée	Fichier des contenus de tables
PAC7DE	Rép. tmp : DE	Entrée	Demandes d'édition
PAC7ET	Rép. tmp : ET	Sortie	Fichier d'édition
PAC7EX	Rép. user : PRTAEX350.txt	Etat	Statistiques d'édition

Edition : PTA360

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier de description de tables
PAC7ET	Rép. tmp : ET	Entrée	Fichier d'édition
PAC7EY	Rép. user : PRTAEY360.txt	Etat	Edition des tables

PRTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) PRTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - TABLE PRINT -
# *
# * -----
# *
# * INPUT
```

```

# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE
# * COL 11-18  : PASSWORD
# *
# * --> ONE "A" TYPE LINE PER TABLE TO BE PRINTED
# * COL 1      : ACTION CODE
# *           : "E" : TABLE PRINTING
# *           : "H" : LIST OF HISTORICAL ACCOUNTS
# *           : "L" : LIST OF THE TABLES
# *           : "S" : LIST OF SUB-SCHEMAS AND SUB-SYSTEMS
# *           : "X" : TABLE CONTENTS WITH HISTORICAL
# *           :       ACCOUNTS
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# * COL 9-16   : HISTORICAL ACCOUNT DATE OR DATE OF THE
# *           : REFERENCE DESCRIPTION
# * COL 17     : SUB-SCHEMA SELECTION
# *           : BLANK : NO SUB-SCHEMA SELECTION
# *           : 1 TO 0 : SELECTED SUB-SCHEMA NUMBER
# * COL 18     : SUB-SYSTEM SELECTION
# *           : BLANK : NO SUB-SYSTEM SELECTION
# *           : 1 TO 0 : SELECTED SUB-SYSTEM NUMBER
# * COL 19     : PRINT OPTION OF THE KEY'S DATA ELEMENTS
# *           : BLANK : CONCATENED DATA ELEMENTS
# *           : "O"  : SEPARATED DATA ELEMENTS
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "PRTA"
echo "          ====="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7CA=$TABINPUT
export PAC7CA
PAC7DE=~TABENV BVPTA320 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7XE=~TABENV BVPTA320 PAC7XE $TABUSERS/PRTAXE320.txt`
export PAC7XE
TABMSG 1009 "BVPTA320"
rtstab BVPTA320

```

```

RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA320"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7DE=`TABENV BVPTA350 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7ET=`TABENV BVPTA350 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EX=`TABENV BVPTA350 PAC7EX $TABUSERS/PRTAEX350.txt`
export PAC7EX
TABMSG 1009 "BVPTA350"
rtstab BVPTA350
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA350"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
PAC7ET=`TABENV BVPTA360 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EY=`TABENV BVPTA360 PAC7EY $TABUSERS/PRTAEY360.txt`
export PAC7EY
TABMSG 1009 "BVPTA360"
rtstab BVPTA360
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA360"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac

```

```
# *****  
TABMSG 1010  
TABRMTMP  
exit $RETURN
```

IMTA - Importation de tables

IMTA - Présentation générale

Cette procédure permet d'importer des tables externes dans les fichiers Pactables existants.

L'utilisateur doit d'abord entrer la description de la Table à importer dans la base VisualAge Pacbase et générer cette description (Procédures GETA/GETT).

Une fois ces opérations effectuées, la Table externe peut alors être importée par la procédure IMTA.

Le format en entrée de IMTA de la Table à importer est un fichier séquentiel comportant un enregistrement par poste de la table, dont le contenu correspond à la description effectuée dans la base VisualAge Pacbase (format d'entrée).

La longueur de l'enregistrement de ce fichier est de 999 caractères (longueur maximum d'un poste de Table).

Condition d'exécution

Cette procédure mettant à jour le fichier Tables TV, l'accès au TP doit être fermé, sauf pour les matériels autorisant la concurrence batch/TP.

RECOMMANDATION pour les plates-formes où l'espace disque alloué aux fichiers est fixe :

Dans le cas d'importation d'une table de taille importante, il peut être nécessaire de procéder, avant l'exécution de cette procédure, à une sauvegarde et à un rechargement, soit pour agrandir le fichier TV, soit pour le réorganiser physiquement de façon à rendre disponible tout le 'free space' prévu initialement.

Restriction

La procédure ne permet d'importer qu'une seule table par exécution.

Note importante

Un programme supplémentaire PTA312 est fourni à partir de la version 2.0.

Lors des mises à jour, le programme PTA310 appelle éventuellement des sous-programmes de contrôle utilisateur pour effectuer des contrôles supplémentaires. L'option de génération de ces sous-programmes par défaut est sans gestion de siècle.

A partir de la version 2.0, si les sous-programmes de contrôle utilisateur sont générés avec l'option gestion du siècle, le nouveau programme PTA312 fourni doit être renommé et utilisé à la place du programme PTA310.

Dans tous les cas, TOUS les sous-programmes de contrôle utilisateur doivent être générés avec la même option.

IMTA - Entrée utilisateur

- Une ligne '*' utilisateur :

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

- Une ligne 'A' renseignant la table à importer :

Pos.	Lon.	Valeur	Signification
2	1	'A'	Code carte
3	6	tttttt	Numéro de la table à importer
9	8	JJMMSSAA	Date de la table (facultatif)

IMTA - Description des étapes

Contrôle et mise à jour des tables : PTA310

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier de description de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreurs
PAC7TG	Rép. base : TG	Entrée	Fichier des paramètres utilisateur
PAC7TV	Rép. base : TV	Entrée Sortie	Fichier des contenus de tables
PAC7MV		Entrée	Mouvements de demande
PAC7NK	Rép. user : TABF.txt	Entrée	Fichier table externe
PAC7DE	Rép. tmp : DE	Sortie	Demandes d'édition

Code	Nom physique	Type	Libellé
PAC7ET	Rép. user : IMTAET310.txt	Etat	Compte-rendu

Mise en forme de l'édition : PTA350

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier de description de tables
PAC7TV	Rép. base : TV	Entrée	Fichier des contenus de tables
PAC7DE	Rép. tmp : DE	Entrée	Demandes d'édition
PAC7ET	Rép. tmp : ET	Sortie	Fichier d'édition
PAC7EX	Rép. user : IMTAEX350.txt	Etat	Statistiques d'édition

Edition : PTA360

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier de description de tables
PAC7ET	Rép. tmp : ET	Entrée	Fichier d'édition
PAC7EY	Rép. user : IMTAEY360.txt	Etat	Edition des tables

IMTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) IMTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - TABLE IMPORT -
# *
# * -----
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE
# * COL 11-18  : PASSWORD
# *
# * --> ONE "A" TYPE LINE PER TABLE TO BE IMPORTED
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER TO BE IMPORTED (TTTTTT)
# * COL 9-16   : TABLE DATE (OPTIONAL) (DDMMCCYY)
# *
# * -----
# Parameter control
# . $TABDIR/system/proc/TABINIT.ini
echo ""
```

```

echo "-----"
TABMSG 1004 "IMTA"
echo "          ====="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7MV=$TABINPUT
export PAC7MV
PAC7NK=`TABENV BVPTA310 PAC7NK `dirname $TABUSERS`/TABF.txt`
export PAC7NK
PAC7DE=`TABENV BVPTA310 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7ET=`TABENV BVPTA310 PAC7ET $TABUSERS/IMTAET310.txt`
export PAC7ET
TABMSG 1009 "BVPTA310"
rtstab BVPTA310
RETURN=?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA310"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7DE=`TABENV BVPTA350 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7ET=`TABENV BVPTA350 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EX=`TABENV BVPTA350 PAC7EX $TABUSERS/IMTAEX350.txt`
export PAC7EX
TABMSG 1009 "BVPTA350"
rtstab BVPTA350
RETURN=?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA350"

```

```

TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
PAC7ET=`TABENV BVPTA360 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EY=`TABENV BVPTA360 PAC7EY $TABUSERS/IMTAEY360.txt`
export PAC7EY
TABMSG 1009 "BVPTA360"
rtstab BVPTA360
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA360"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

RETA - Réorganisation des tables

RETA - Présentation générale

A partir de la base Pactables, cette procédure reconstitue le fichier sauvegarde contenant les nouveaux fichiers des descriptifs et des contenus des tables, images réorganisées des fichiers initiaux TD et TV.

RETA supprime les enregistrements annulés logiquement en mise à jour, en réorganisant les historiques des fichiers en fonction des demandes de l'utilisateur (Voir le 'Manuel de Référence Pactables'). Les enregistrements annulés logiquement peuvent être gardés sur option.

Pour les programmes utilisateur écrits en Cobol II, elle affecte le signe aux données numériques signées positives absent des versions antérieures.

Condition d'exécution

Pour des raisons de cohérence de la base, l'accès au TP doit être fermé.

RETA - Entrée utilisateur

- Une ligne '*' du gestionnaire de tables :

Pos.	Lon.	Valeur	Signification
2	1	'*'	Code carte
3	8	uuuuuuuu	Code de l'administrateur de la base
11	8	pppppppp	Mot de passe du gestionnaire

- Une ligne 'A' par historique à garder ou à supprimer :

Pos.	Lon.	Valeur	Signification
1	1		Code mouvement
		'S'	Historique à supprimer
		'G'	Historique à conserver
2	1	'A'	Code carte
3	6	tttttt	Numéro de la table
9	8	JJMMSSAA	Date historique
19	1		Option
		' '	- En cas de code action égal à 'G', conservation de l'historique dont la date est égale celle indiquée. En l'absence de la date, tous les historiques sont conservés.
			- En cas de code action égal à 'S', suppression de l'historique dont la date est égale à celle indiquée.
		'<'	- En cas de code action égal à 'G', conservation de tous les historiques dont la date est strictement antérieure à la date indiquée.
			- En cas de code action égal à 'S', suppression de tous les historiques dont la date est strictement antérieure à la date indiquée.
		'>'	- En cas de code action égal à 'G', conservation de tous les historiques dont la date est postérieure ou égale à la date indiquée.
			- En cas de code action égal à 'S', suppression de tous les historiques dont la date est postérieure ou égale à la date indiquée.

Les codes mouvements 'G' et 'S' sont exclusifs.

Pour des explications plus détaillées sur ces entrées, se reporter au Manuel de référence Pactables.

RETA - Description des étapes

Réorganisation des contenus des tables : PTA400

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier de description de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreurs
PAC7TG	Rép. base : TG	Entrée	Fichier des paramètres utilisateur
PAC7TV	Rép. base : TV	Entrée	Fichier des contenus de tables
PAC7DR		Entrée	Demande de réorganisation
PAC7TX	Rép. tmp : TX	Sortie	Fichier des contenus réorganisés
PAC7DE	Rép. user : MVRETA.txt	Sortie	Fichier des liste tables réorganisées
PAC7IR	Rép. user : RETAIR400.txt	Etat	Compte-rendu des mouvements

Remarque

Le fichier PAC7DE (fichier liste tables réorganisées), qui a pour description les demandes d'édition, peut être gardé. Une fois la réorganisation effectuée, ce fichier, en entrée de la procédure PRTA appliquée sur les fichiers réorganisés, peut ainsi permettre l'édition de toutes les tables qui ont été conservées, pour vérification du bon déroulement de la réorganisation.

Codes retour :

- 0 : pas d'erreur détectée
- 4 : erreur sur une carte 'A'

Validation des contenus des tables : PTA410

Code	Nom physique	Type	Libellé
PAC7MB		Entrée	Mouvements de demande de réorganisation
PAC7TX	Rép. tmp : TX	Entrée	Fichier des contenus réorganisés
PAC7TW	Rép. tmp : TW	Sortie	Fichier des contenus validés

Réorganisation des descriptifs : PTA420

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7DE	Rép. user : MVRETA.txt	Entrée	Fichier liste tables réorganisées

Code	Nom physique	Type	Libellé
PAC7TS	Rép. tmp : TS	Sortie	Fichier des descriptifs réorganisés
PAC7ML	Rép. user : MWRETA.txt	Sortie	Demande d'édition des descriptifs

Remarque

Le fichier PAC7ML (demande d'édition des descriptifs) doit être gardé et mis en entrée de la procédure LDTA, pour obtenir l'édition des descriptifs qui ont été conservés, pour vérification du bon déroulement de la réorganisation.

Constitution du fichier sauvegarde : PTA430

Code	Nom physique	Type	Libellé
PAC7TW	Rép tmp : TW	Entrée	Fichier des contenus validés
PAC7TS	Rép. tmp : TS	Entrée	Fichier des descriptifs réorganisés
PAC7TC	Rép. save : TC	Sortie	Fichier sauvegarde résultat de la réorganisation

Sauvegarde de TG : PTASVG

Code	Nom physique	Type	Libellé
PAC7TG	Rép. base : TG	Entrée	Fichier paramètres utilisateur
PAC7TC	Rép. save : TC	Sortie	Sauvegarde tables

RETA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) RETA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - TABLE REORGANIZATION -
# *
# * -----
# *
# * INPUT
# * --> ONE "*" TYPE LINE IDENTIFYING PACTABLES MANAGER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : PACTABLES MANAGER CODE
# * COL 11-18  : PACTABLES MANAGER PASSWORD
# *
# * --> ONE "A" TYPE LINE PER HISTORICAL ACCOUNT TO KEEP
# *      OR DELETE
# * COL 1      : ACTION CODE
# *           : "S" : HISTORICAL ACCOUNT TO PURGE
```

```

# *           : "G" : HISTORICAL ACCOUNT TO KEEP
# * COL 2     : "A" : LINE CODE
# * COL 3-8   : TABLE NUMBER (TTTTTT)
# * COL 9-16  : HISTORICAL ACCOUNT DATE (DDMMCCYY)
# * COL 19    : OPTION
# *           : " " : SEE DOCUMENTATION FOR MORE
# *           : "X" : EXPLANATIONS
# *           : "X"
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "RETA"
echo "          ====="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TV.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7DR=$TABINPUT
export PAC7DR
PAC7DE=~TABENV BVPTA400 PAC7DE $TABUSERS/MVRETA.txt~
export PAC7DE
PAC7TX=~TABENV BVPTA400 PAC7TX $TABTMP/TX~
export PAC7TX
PAC7IR=~TABENV BVPTA400 PAC7IR $TABUSERS/RETAIR400.txt~
export PAC7IR
TABMSG 1009 "BVPTA400"
rtstab BVPTA400
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA400"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
PAC7TX=~TABENV BVPTA410 PAC7TX $TABTMP/TX~
export PAC7TX
PAC7TW=~TABENV BVPTA410 PAC7TW $TABTMP/TW~

```



```

export PAC7TW
PAC7MB=$TABINPUT
export PAC7MB
TABMSG 1009 "BVPTA410"
rtstab BVPTA410
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA410"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
PAC7TS=`TABENV BVPTA420 PAC7TS $TABTMP/TS`
export PAC7TS
PAC7DE=`TABENV BVPTA420 PAC7DE $TABUSERS/MVRETA.txt`
export PAC7DE
PAC7ML=`TABENV BVPTA420 PAC7ML $TABUSERS/MWRETA.txt`
export PAC7ML
TABMSG 1009 "BVPTA420"
rtstab BVPTA420
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA420"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
PAC7TW=`TABENV BVPTA430 PAC7TW $TABTMP/TW`
export PAC7TW
PAC7TS=`TABENV BVPTA430 PAC7TS $TABTMP/TS`
export PAC7TS
TABMSG 1009 "BVPTA430"
rtstab BVPTA430
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA430"

```

```

TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TG.ini
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
TABMSG 1009 "BVPTASVG"
rtstab BVPTASVG
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTASVG"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
echo ""
echo "Calling the file TCBACKUP.CMD"
. $TABDIR/config/$1/TCBACKUP.ini
TABRMTMP
exit $RETURN

```

SVTA - Sauvegarde

SVTA - Présentation générale

La procédure SVTA effectue la sauvegarde des descriptifs et des contenus des Tables, ainsi que des paramètres utilisateur dans un fichier séquentiel unique (TC).

Condition d'exécution

Afin de garder une certaine cohérence dans les données, il est préférable de fermer l'accès au TP.

Entrée utilisateur

Aucune.

SVTA - Description des étapes

Sauvegarde de TD : PTASVT

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7TC	Rép. save : TC	Sortie	Sauvegarde des fichiers tables

Sauvegarde de TV : PTASVV

Code	Nom physique	Type	Libellé
PAC7TV	Rép. base : TV	Entrée	Fichier des contenus de tables
PAC7TC	Rép. save : TC.NEW	Sortie	Sauvegarde des fichiers Tables

Sauvegarde de TG : PTASVG

Code	Nom physique	Type	Libellé
PAC7TG	Rép. base : TG	Entrée	Fichier paramètres utilisateur
PAC7TC	Rép. save : TC.NEW	Sortie	Sauvegarde des fichiers Tables

SVTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) SVTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - TABLES BACKUP -
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "SVTA"
echo "           ====="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
STATUS=`tabtp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != " Inactive" -a "$STATUS" != "" ]
then
```

```

        TABMSG 1012 "SVTA"
        TABMSG 1037 $1
        TABERR
        TABRMTMP
        exit $RETURN
    fi
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
TABMSG 1009 "BVPTASVT"
rtstab BVPTASVT
RETURN=$?
case $RETURN in
0)
;;
*)
    TABMSG 1012 "BVPTASVT"
    TABMSG 1025
    TABERR
    TABRMTMP
    exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TV.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
TABMSG 1009 "BVPTASVV"
rtstab BVPTASVV
RETURN=$?
case $RETURN in
0)
;;
*)
    TABMSG 1012 "BVPTASVV"
    TABMSG 1025
    TABERR
    TABRMTMP
    exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TG.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
TABMSG 1009 "BVPTASVG"
rtstab BVPTASVG
RETURN=$?
case $RETURN in
0)
;;
*)
    TABMSG 1012 "BVPTASVG"

```

```

TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
echo ""
echo "Calling the file TCBACKUP.ini"
. $TABDIR/config/$1/TCBACKUP.ini
TABRMTMP
exit $RETURN

```

TCTA - Transposition d'une autre plateforme

TCTA - Présentation générale

Le but de cette procédure est la récupération sur une base Pactables cible, de bases Pactables en provenance d'autres plates-formes (plates-formes origines), pour les adapter à une installation sous un autre environnement.

La sauvegarde de la base est triée selon les critères du format de la plate-forme cible (ASCII ou EBCDIC).

Si la version sur le site d'origine est identique à la version de l'environnement cible, les opérations à effectuer sont les suivantes :

- Sauvegarde sur le site d'origine (procédure SVTA),
- Transfert du fichier TC issu de SVTA sur la plate-forme cible,
- Reprise du fichier sur la plate-forme cible (procédure TCTA),
- Restauration de la base (procédure RSTA) avec en entrée le fichier TC constitué dans l'étape précédente.

Si, à l'inverse, le site d'origine est d'une version antérieure et que cette version nécessite une reprise, la sauvegarde TC doit être reprise au nouveau format sur le site d'origine avant son transfert sur le site cible.

Condition d'exécution

Aucune. Cependant, lire attentivement les remarques page suivante.

Entrée utilisateur

Aucune.

Remarques

1. Transfert de la sauvegarde

Le transfert de la sauvegarde TC depuis le site d'origine vers le réseau local où sera installé Pactables est à la charge de l'utilisateur. Paramètres à spécifier lors du transfert : les fichiers sont des données (DATA), ils doivent être convertis au format ASCII et doivent contenir des caractères de contrôle pour les fins d'enregistrements (CRLF en général, ou LINE_FEED).

2. Espace disque

La procédure TCTA, décrite ci-après, consiste essentiellement à trier la sauvegarde TC selon une séquence ASCII. Le tri est effectué dans 3 programmes différents pour minimiser l'espace disque nécessaire. Néanmoins, la procédure a besoin de 4 à 5 fois l'équivalent du fichier de départ.

La procédure ne prévoit pas de détruire le fichier TC.INI issu du site central. Il est possible de détruire ce fichier à la fin de la première étape, si l'utilisateur a besoin de place disque supplémentaire.

Enfin, pour des raisons de temps d'exécution, il est conseillé d'assigner les fichiers de travail et de tri sur le disque du serveur Pactables qui exécute les programmes.

TCTA - Description des étapes

Découpe de la sauvegarde TC : PTATC1

Code	Nom physique	Type	Libellé
PAC7TC	Rép. save : TC	Entrée	Fichier sauvegarde
PAC7SD	Rép. tmp : SD	Sortie	Image séquentielle des descriptifs
PAC7SV	Rép. tmp : SV	Sortie	Image séquentielle des contenus
PAC7SG	Rép. tmp : SG	Sortie	Image séquentielle des paramètres

Tri des descriptifs de tables : PTATCD

Code	Nom physique	Type	Libellé
PAC7SD	Rép. tmp : DS	Entrée	Image séquentielle des descriptifs
PAC7AD	Rép. tmp : AD	Sortie	Descriptifs triés

Tri des contenus des tables : PTATCV

Code	Nom physique	Type	Libellé
PAC7SV	Rép. tmp : SV	Entrée	Image séquentielle contenus
PAC7AV	Rép. tmp : AV	Sortie	Contenus des tables triés

Tri des paramètres utilisateur : PTATCG

Code	Nom physique	Type	Libellé
PAC7SG	Rép. tmp : SG	Entrée	Image séquentielle des paramètres
PAC7AG	Rép. tmp : AG	Sortie	Paramètres utilisateur triés

Reconstitution de la sauvegarde TC : PTATC2

Code	Nom physique	Type	Libellé
PAC7AD	Rép. tmp : AD	Entrée	Image séquentielle des descriptifs
PAC7AV	Rép. tmp : AV	Entrée	Image séquentielle des contenus
PAC7AG	Rép. tmp : AG	Entrée	Image séquentielle des paramètres
PAC7TC	Rép. save : TC.NEW	Sortie	Sauvegarde TC au format EBCDIC

TCTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) TCTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *   - TABLES TRANSFER FROM ANOTHER PLATFORM -
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "TCTA"
echo "======"
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
. $TABDIR/config/$1/PACSAVTC.ini
TABMSG 1044 $PACSAVTC.INI
TABMSG 1045 $PACSAVTC
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTC.INI
export PAC7TC
PAC7SD=`TABENV BVPTATC1 PAC7SD $TABTMP/SD`
export PAC7SD
PAC7SG=`TABENV BVPTATC1 PAC7SG $TABTMP/SG`
```

```

export PAC7SG
PAC7SV=~TABENV BVPTATC1 PAC7SV $TABTMP/SV~
export PAC7SV
TABMSG 1009 "BVPTATC1"
rtstab BVPTATC1
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTATC1"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
PAC7SD=~TABENV BVPTATCD PAC7SD $TABTMP/SD~
export PAC7SD
PAC7AD=~TABENV BVPTATCD PAC7AD $TABTMP/AD~
export PAC7AD
TABMSG 1009 "BVPTATCD"
rtstab BVPTATCD
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTATCD"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
PAC7SV=~TABENV BVPTATCV PAC7SV $TABTMP/SV~
export PAC7SV
PAC7AV=~TABENV BVPTATCV PAC7AV $TABTMP/AV~
export PAC7AV
TABMSG 1009 "BVPTATCV"
rtstab BVPTATCV
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTATCV"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac

```



```

# *****
PAC7SG=`TABENV BVPTATCG PAC7SG $TABTMP/SG`
export PAC7SG
PAC7AG=`TABENV BVPTATCG PAC7AG $TABTMP/AG`
export PAC7AG
TABMSG 1009 "BVPTATCG"
rtstab BVPTATCG
RETURN=?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTATCG"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
PAC7TC=$PACSAVTCNEW
export PAC7TC
PAC7AD=`TABENV BVPTATC2 PAC7AD $TABTMP/AD`
export PAC7AD
PAC7AG=`TABENV BVPTATC2 PAC7AG $TABTMP/AG`
export PAC7AG
PAC7AV=`TABENV BVPTATC2 PAC7AV $TABTMP/AV`
export PAC7AV
TABMSG 1009 "BVPTATC2"
rtstab BVPTATC2
RETURN=?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTATC2"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
echo ""
echo "Calling the file TCBACKUP.ini"
. $TABDIR/config/$1/TCBACKUP.ini
TABRMTMP
exit $RETURN

```

RSTA - Restauration

RSTA - Présentation générale

La procédure RSTA permet de reconstruire les descriptifs et contenus des tables, ainsi que les paramètres utilisateur à partir de l'image séquentielle obtenue par la procédure de sauvegarde (SVTA).

Condition d'exécution

Afin de garder une certaine cohérence dans les données, l'accès au TP doit être fermé.

REMARQUE concernant les plates-formes où l'espace disque alloué aux fichiers est figé :

Cette procédure recréant les fichiers, il peut être intéressant de réajuster préalablement la taille des différents fichiers en fonction de leur évolution estimée.

Ces modifications doivent se faire dans la bibliothèque de paramètres système.

Anomalies d'exécution

Se reporter au sous-chapitre "Anomalies d'exécution" du chapitre "Les procédures batch".

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

Entrée utilisateur

Aucune.

RSTA - Description des étapes

Restauration de TD : PTARST

Code	Nom physique	Type	Libellé
PAC7TC	Rép. save : TC	Entrée	Sauvegarde tables
PAC7TD	Rép. base : TD	Sortie	Fichier des descriptifs de tables

Restauration de TV : PTARSV

Code	Nom physique	Type	Libellé
PAC7TC	Rép. save : TC	Entrée	Sauvegarde tables
PAC7TV	Rép. base : TV	Sortie	Fichier des contenus de tables

Restauration de TG : PTARSG

Code	Nom physique	Type	Libellé
PAC7TC	Rép. save : TC	Entrée	Sauvegarde tables
PAC7TG	Rép. base : TG	Sortie	Fichier paramètres utilisateur

RSTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) RSTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - TABLES RESTORATION -
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "RSTA"
echo "           ====="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
STATUS=`tabtp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != "Inactive" -a "$STATUS" != "" ]
then
    TABMSG 1012 "RSTA"
    TABMSG 1037 $1
    TABERR
    TABRMTMP
    exit $RETURN
fi
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTC
export PAC7TC
```

```

TABMSG 1009 "BVPTARST"
rtstab BVPTARST
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTARST"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TV.ini
PAC7TC=$PACSAVTC
export PAC7TC
TABMSG 1009 "BVPTARST"
rtstab BVPTARST
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTARST"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TG.ini
PAC7TC=$PACSAVTC
export PAC7TC
TABMSG 1009 "BVPTARST"
rtstab BVPTARST
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTARST"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

LDTA - Liste descriptifs de tables

LDTA - Présentation générale

La procédure LDTA permet d'éditer les descriptifs de tables.

Condition d'exécution

Cette procédure consulte le fichier TD qui peut rester accessible au TP.

LDTA - Entrée utilisateur

Une ligne 'Z' par demande d'édition :

Pos.	Lon.	Valeur	Signification
2	1	'Z'	Code carte
5	4		Code de la demande :
		'TLS '	Demande de liste de descriptifs
		'TDS '	Demande d'édition de descriptif
9	6	ttttt	Numéro de la table
23	8	JJMMSSAA	Date historique du descriptif

Remarque

Les mouvements en entrée de cette procédure ne sont pas contrôlés ; les demandes erronées sont ignorées.

LDTA - Description des étapes

Edition des descriptifs : PTA290

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7TE		Entrée	Demande d'édition
PAC7ID	Rép. user : LDTAID290.txt	Etat	Edition des descriptifs

LDTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) LDTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - LIST OF TABLE DESCRIPTIONS -
# *
# * -----
# *
```

```

# * INPUT
# * --> ONE "Z" TYPE LINE PER PRINT REQUEST
# * COL 2      : "Z" : LINE CODE
# * COL 5-8    : PRINT REQUEST
# *           : "TLS" : LIST OF TABLE DESCRIPTIONS
# *           : "TDS" : TABE DESCRIPTION
# * COL 9-14   : TABLE NUMBER (TTTTTT)
# * COL 23-30  : HISTORICAL ACCOUNT DATE (MMDCCYY)
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "LDTA"
echo "          ====="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
PAC7TE=$TABINPUT
export PAC7TE
PAC7ID=`TABENV BVPTA290 PAC7ID $TABUSERS/LDTAID290.txt`
export PAC7ID
TABMSG 1009 "BVPTA290"
rtstab BVPTA290
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA290"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

PMTA - Mise à jour des paramètres

PMTA - Présentation générale

La procédure PMTA permet de mettre à jour les codes utilisateur, leurs mots de passe, les autorisations d'accès et les cartes de contrôle de demandes d'édition.

Lorsque l'entrée utilisateur contient une ligne 'TA' avec le code utilisateur administrateur de la base, la procédure PMTA édite l'ensemble des paramètres utilisateur.

Condition d'exécution

Cette procédure met à jour le fichier TG qui doit être fermé au TP, sauf pour les matériels permettant la concurrence Batch/TP.

PMTA - Entrée utilisateur

- Ligne 'TA' : mise à jour du code utilisateur

Pos.	Lon.	Valeur	Signification
1	1		Code action
		blanc	Création ou modification
		'C'	Création
		'M'	Modification
		'A'	Annulation
2	8	uuuuuuuu	Code utilisateur
10	2	'TA'	Code carte
12	8	pppppppp	Mot de passe
20	1		Niveau d'accès global
		'0'	Pas d'accès global autorisé
		'1'	Consultation seule
		'2'	Mise à jour des tables autorisée
		'3'	M-à-j codes utilisateur autorisée

- Ligne 'TC' : Autorisations d'accès par table

Pos.	Lon.	Valeur	Signification
1	1		Code action
		blanc	Création ou modification
		'C'	Création

Pos.	Lon.	Valeur	Signification
		'M'	Modification
		'A'	Annulation
2	8	uuuuuuuu	Code utilisateur
10	2	'TC'	Code carte
12	6	tttttt	Code de la table
18	3	nnn	Numéro de ligne
21	60		Autorisations d'accès, codifiées sous la forme de 20 répétitions de combinaisons :
	1	n	Numéro de sous-schéma
	1	n	Numéro de sous-système
	1	x	Autorisation (0, 1 ou 2) ('*' signifiant tous sous-schémas ou tous sous-systèmes)

- Lignes 'TJ' : Cartes de contrôle

Pos.	Lon.	Valeur	Signification
1	1		Code action
		blanc	Création ou modification
		'C'	Création
		'M'	Modification
		'A'	Annulation
2	8	uuuuuuuu	Code utilisateur
10	2	'TJ'	Code carte
12	6		Numéro de ligne de JCL
		< 600000	Lignes de JCL en tête de flot
		> 599999	Lignes de JCL en fin de flot
18	69		Contenu de la ligne de JCL

Note

L'annulation du code utilisateur entraîne l'annulation de toutes ses autorisations d'accès et de toutes ses lignes de JCL.

Il doit exister dans la base au moins un code administrateur de base de niveau d'accès égal à 3. La suppression du dernier code administrateur est interdite.

PMTA - Description des étapes

Mise à jour des paramètres utilisateur : PTA100

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreur
PAC7TG	Rép. base : TG	Entrée Sortie	Fichier des paramètres utilisateur
PAC7MV		Entrée	Mouvements de demande d'extraction
PAC7NU	Rép. tmp : NU	Sortie	Demande d'édition des paramètres
PAC7ET	Rép. user : PMTAET100.txt	Etat	Compte-rendu des mouvements

Edition des paramètres utilisateur : PTA120

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7TG	Rép. base : TG	Entrée	Fichier des paramètres utilisateur
PAC7NU	Rép. tmp : NU	Entrée	Demande d'édition
PAC7ET	Rép. user : PMTAET120.txt	Etat	Edition des paramètres utilisateur

PMTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) PMTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - USER PARAMETER UPDATE -
# *
# * -----
# *
# * INPUT
# * --> ONE "TA" LINE : USER PARAMETER UPDATING
# * COL 1      : ACTION CODE
# *           : " " : CREATION OR MODIFICATION
# *           : "C" : CREATION
# *           : "M" : MODIFICATION
# *           : "D" : DELETION
# * COL 2-9    : USER CODE (UUUUUUUU)
# * COL 10-11  : "TA" : LINE CODE
# * COL 12-19  : PASSWORD (PPPPPPPP)
# * COL 20     : GENERAL ACCESS AUHTORIZATION
```

```

# *          : "0" : NO GENERAL ACCESS AUTHORIZATION
# *          : "1" : READ ONLY ACCESS AUTHORIZATION
# *          : "2" : READ WRITE AUTHORIZATION ON TABLES
# *          : "3" : READ WRITE AUTHORIZATION ON USER
# *          :      CODE
# *
# * --> ONE "TC" LINE : ACCESS AUTHORIZATIONS PER TABLE
# * COL 1      : ACTION CODE
# *          : " " : CREATION OR MODIFICATION
# *          : "C" : CREATION
# *          : "M" : MODIFICATION
# *          : "D" : DELETION
# * COL 2-9    : USER CODE (UUUUUUUU)
# * COL 10-11  : "TC" : LINE CODE
# * COL 12-17  : TABLE CODE (TTTTTT)
# * COL 18-20  : LINE NUMBER (NNN)
# * COL 21-80  : ACCESS AUTHORIZATIONS : 20 ACCESS
# *          : AUTHORIZATIONS MAY BE ENTERED IN THIS
# *          : FIELD, WITH, FOR EACH AUTHORIZATION, THE
# *          : SUB-SCHEMA NUMBER (N), THE SUB-SYSTEM
# *          : NUMBER (N), THE AUTHORIZATION (0,1,2).
# *          : "*" IS REQUIRED FOR ALL SUB-SCHEMAS AND
# *          : SUB-SYSTEMS.
# *
# * --> ONE "TJ" LINE : CONTROL CARDS
# * COL 1      : ACTION CODE
# *          : " " : CREATION OR MODIFICATION
# *          : "C" : CREATION
# *          : "M" : MODIFICATION
# *          : "D" : DELETION
# * COL 2-9    : USER CODE (UUUUUUUU)
# * COL 10-11  : "TJ" : LINE CODE
# * COL 12-17  : TABLE CODE (TTTTTT)
# * COL 18-20  : SCRIPT LINE NUMBER
# *          :LT600000 : CONTROL CARD IN FRONT OF PROGRAM
# *          :GT599999 : CONTROL CARD IN BACK OF PROGRAM
# * COL 21-80  : CONTENT OF THE SCRIPT LINE
# *
# * -----
# Parameter control
# . $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "PMTA"
echo "
=====
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini

```

```

. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MV=$TABINPUT
export PAC7MV
PAC7NU=`TABENV BVPTA100 PAC7NU $TABTMP/NU`
export PAC7NU
PAC7ET=`TABENV BVPTA100 PAC7ET $TABUSERS/PMTAET100.txt`
export PAC7ET
TABMSG 1009 "BVPTA100"
rtstab BVPTA100
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA100"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7NU=`TABENV BVPTA120 PAC7NU $TABTMP/NU`
export PAC7NU
PAC7ET=`TABENV BVPTA120 PAC7ET $TABUSERS/PMTAET120.txt`
export PAC7ET
TABMSG 1009 "BVPTA120"
rtstab BVPTA120
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA120"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

EXTA - Extraction de tables

EXTA - Présentation générale

La procédure EXTA extrait les données de tables sous forme de mouvements de mise à jour batch.

Condition d'exécution

Cette procédure consulte les fichiers Pactables qui peuvent rester accessibles au TP.

EXTA - Entrée utilisateur

- Une ligne '*' par utilisateur :

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

- Une ligne 'A' par table à extraire :

Pos.	Lon.	Valeur	Signification
2	1	'A'	Code carte
3	6	tttttt	Numéro de la table
9	8	JJMMSSAA	Date de l'historique
17	1		Inutilisé
18	1		Sélection du sous-système
		' '	Pas de sélection de sous-système
		1 à 0	Numéro du sous-système sélectionné
19	1		Délimiteur des données
		' '	Pris comme '/' par défaut

EXTA - Description des étapes

Extraction des données des tables : PTA150

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreur
PAC7TV	Rép. base : TV	Entrée	Fichier des contenus de tables
PAC7TG	Rép. base : TG	Entrée	Fichier des paramètres utilisateur
PAC7MV		Entrée	Mouvements de demande d'extraction
PAC7EX	Rép. tmp : EX	Sortie	Mouvements extraits
PAC7ET	Rép. user : EXTAET150.txt	Etat	Compte-rendu des mouvements

Edition des mouvements extraits : PTA160

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7EX	Rép. tmp : EX	Entrée	Mouvements extraits
PAC7NU	Rép. user : NUTAB.txt	Sortie	Mouvements extraits
PAC7ET	Rép. user : EXTAET160.txt	Etat	Edition des données extraites

Codes retour :

- 0 : pas de délimiteur dans les données
- 8 : présence de délimiteur dans une table au moins
- 12 : présence de délimiteur dans toutes les tables

EXTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) EXTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - TABLE EXTRACTION -
# *
# * -----
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE (UUUUUUUU)
# * COL 11-18  : PASSWORD (PPPPPPPP)
# *
# * --> ONE "A" LINE TYPE PER TABLE TO EXTRACT
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
# * COL 17     : NOT USED
# * COL 18     : SUB-SYSTEM SELECTION
# *           : BLANK : NO SUB-SYSTEM SELECTION
# *           : 1 TO 0 : NUMBER OF SELECTED SUB SYSTEM
# * COL 19     : DATA DELIMITEUR
# *           : BLANK : "/" BY DEFAULT
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "EXTA"
echo "
=====
TABMSG 1047 "$TABBASE"
```

```

TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TV.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MV=$TABINPUT
export PAC7MV
PAC7EX=`TABENV BVPTA150 PAC7EX $TABTMP/EX`
export PAC7EX
PAC7ET=`TABENV BVPTA150 PAC7ET $TABUSERS/EXTAET150.txt`
export PAC7ET
TABMSG 1009 "BVPTA150"
rtstab BVPTA150
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA150"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
PAC7EX=`TABENV BVPTA160 PAC7EX $TABTMP/EX`
export PAC7EX
PAC7NU=`TABENV BVPTA160 PAC7NU `dirname $TABUSERS`/NUTAB.txt`
export PAC7NU
PAC7ET=`TABENV BVPTA160 PAC7ET $TABUSERS/EXTAET160.txt`
export PAC7ET
TABMSG 1009 "BVPTA160"
rtstab BVPTA160
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTA160"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac

```

```
# *****
TABMSG 1010
TABRMTMP
exit $RETURN
```

TUTA - Mise en exploitation

TUTA - Présentation générale

La procédure TUTA extrait des tables sous forme de tables sans historique destinées à être mises en exploitation.

La procédure crée deux nouveaux fichiers qui contiennent les descriptifs et le contenu des tables sélectionnées. Chaque table sélectionnée n'a qu'un seul descriptif et qu'une seule version de données.

Condition d'exécution

Cette procédure recrée deux fichiers AD et AV respectivement images réorganisées de TD et TV.

La procédure effectue la définition de ces deux fichiers dans la deuxième étape.

TUTA - Entrée utilisateur

- Une ligne '*' :

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

- Eventuellement une ligne 'A' par table à sélectionner :

Pos.	Lon.	Valeur	Signification
2	1	'A'	Code carte
3	6	ttttt	Numéro de la table
9	8	JJMMSSAA	Date de l'historique

L'absence de la ligne 'A' permet au demandeur de mettre en exploitation toutes les tables auxquelles il a accès à la date du jour. Une date différente peut être indiquée sur une ligne 'A' unique sans numéro de table.

TUTA - Description des étapes

Mise en exploitation des tables : PTAU80

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreur
PAC7TV	Rép. base : TV	Entrée	Fichier des contenus de tables
PAC7TG	Rép. base : TG	Entrée	Fichier des paramètres utilisateur
PAC7MX		Entrée	Mouvements de demande
PAC7AD	Rép. base : AD	Sortie	Fichier des descriptifs de tables
PAC7AV	Rép. base : AV	Sortie	Fichier des contenus de tables
PAC7ET	Rép. user : TUTAETU80.txt	Etat	Compte-rendu des mouvements

TUTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) TUTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - DIRECT CONSULTATION OF TABLES -
# *
# * -----
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE (UUUUUUUU)
# * COL 11-18  : PASSWORD (PPPPPPPP)
# *
# * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
# *
# * -----
# Parameter control
# . $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "TUTA"
echo "
=====
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
```



```

echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TV.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MX=$TABINPUT
export PAC7MX
. $TABDIR/config/$1/PAC7AD.ini
. $TABDIR/config/$1/PAC7AV.ini
PAC7ET="TABENV BVPTAU80 PAC7ET $TABUSERS/TUTAETU80.txt"
export PAC7ET
TABMSG 1009 "BVPTAU80"
rtstab BVPTAU80
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTAU80"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

Gestionnaire de tables réparties (Option DTM)

CDT1-CDT2 - Comparaison de descriptifs

Le Gestionnaire de Tables Réparties est un utilitaire optionnel. A ce titre, son utilisation est soumise à un contrat d'acquisition.

Comparaison de descriptifs et mise à niveau

La comparaison de descriptifs de tables consiste à comparer deux états du fichier des descriptifs afin d'en extraire les différences sous la forme d'un fichier séquentiel intermédiaire (procédure CDT1).

Ce fichier est utilisé pour mettre à jour le fichier des descriptifs en retard, dit fichier 'esclave' (procédure CDT2).

Condition d'exécution

La procédure CDT1 consulte les fichiers Pactables qui peuvent rester accessibles au TP.

En revanche, la procédure CDT2, à partir du résultat de la procédure CDT1, met à niveau les fichiers TD et TV dits 'esclaves'. Ceux-ci doivent donc être inaccessibles en TP.

CDT1 - Entrée utilisateur

- Une ligne '*' par utilisateur :

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

- Une ligne 'A' par table à comparer :

Pos.	Lon.	Valeur	Signification
2	1	'A'	Code carte
3	6	tttttt	Numéro de la table

La présence d'une seule ligne 'A' sans numéro de table permet de comparer les descriptifs de toutes les tables.

CDT1 - Description des étapes

Contrôle des mouvements : PTAD05

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier 'maître' des descriptifs de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreur
PAC7TG	Rép. base : TG	Entrée	Fichier des paramètres utilisateur
PAC7MV		Entrée	Mouvements de demande de comparaison
PAC7MX	Rép. tmp : MX	Sortie	Mouvements valides de demande comparaisons
PAC7ET	Rép. user : CDT1ETD05.txt	Etat	Compte-rendu des mouvements

Comparaison des descriptifs et extraction : PTAD10

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier 'maître' des descriptifs de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreur

Code	Nom physique	Type	Libellé
PAC7TS	Rép. user : TD	Entrée	Fichier 'esclave' des descriptifs de tables
PAC7MX	Rép. tmp : MX	Entrée	Mouvements valides
PAC7TX	Rép. user : MBCDT2.txt	Sortie	Résultat de la comparaison extrait pour entrée de la procédure CDT2
PAC7ET	Rép. user : CDT1ETD10.txt	Etat	Edition de l'extraction

CDT1 - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) CDT1 BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *   - DISPATCHED TABLE MANAGEMENT -
# *   - TABLE DESCRIPTION COMPARISON -
# * -----
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE (UUUUUUUU)
# * COL 11-18  : PASSWORD (PPPPPPPP)
# *
# * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "CDT1"
echo "====="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MV=$TABINPUT
export PAC7MV
```

```

PAC7MX=~TABENV BVPTAD05 PAC7MX $TABTMP/MX~
export PAC7MX
PAC7ET=~TABENV BVPTAD05 PAC7ET $TABUSERS/CDT1ETD05.txt~
export PAC7ET
TABMSG 1009 "BVPTAD05"
rtstab BVPTAD05
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTAD05"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
PAC7TS=~TABENV BVPTAD10 PAC7TS `dirname $TABUSERS`~/TD~
export PAC7TS
PAC7MX=~TABENV BVPTAD10 PAC7MX $TABTMP/MX~
export PAC7MX
PAC7TX=~TABENV BVPTAD10 PAC7TX `dirname $TABUSERS`~/MBCDT2.txt~
export PAC7TX
PAC7ET=~TABENV BVPTAD10 PAC7ET $TABUSERS/CDT1ETD10.txt~
export PAC7ET
TABMSG 1009 "BVPTAD10"
rtstab BVPTAD10
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTAD10"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

CDT2 - Description des étapes

Mise à niveau des fichiers 'esclaves', Prise en compte du fichier extrait par CDT1 : PTAD20

Code	Nom physique	Type	Libellé
PAC7TD	Rép. user : TD	Entrée	Fichier 'esclave' des descriptifs de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreur
PAC7TX	Rép. user : MBCDT2.txt	Entrée	Résultat extrait de la comparaison dans la procédure CDT1
PAC7TV	Rép. user : TV	Sortie	Fichier des contenus de tables liés au fichier descriptifs 'esclave'
PAC7ET	Rép. user : CDT2ETD20.txt	Etat	Compte-rendu de la mise à jour

CDT2 - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) CDT2 BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *   - DISPATCHED TABLE MANAGEMENT -
# *   - UPDATE OF TABLE DESCRIPTIONS AFTER COMPARISON -
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "CDT2"
echo "          ====="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TE.ini
PAC7TD=`TABENV BVPTAD20 PAC7TD `dirname $TABUSERS`/TD`
export PAC7TD
PAC7TV=`TABENV BVPTAD20 PAC7TV `dirname $TABUSERS`/TV`
export PAC7TV
PAC7TX=`TABENV BVPTAD20 PAC7TX `dirname $TABUSERS`/MBCDT2.txt`
export PAC7TX
PAC7ET=`TABENV BVPTAD20 PAC7ET $TABUSERS/CDT2ETD20.txt`
export PAC7ET
TABMSG 1009 "BVPTAD20"
rtstab BVPTAD20
RETURN=$?
case $RETURN in
0)
```

```

;;
*)
TABMSG 1012 "BVPTAD20"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN

```

CVTA - Mise à niveau contenus tables

La procédure CVTA consiste à extraire sous forme de mouvements de mise à jour batch des données de tables modifiées à une date donnée ou entre deux dates précisées.

Condition d'exécution

Cette procédure consulte les fichiers Pactables ; elle peut être exécutée même si les fichiers sont ouverts sous TP.

CVTA - Entrée utilisateur

- Une ligne '*' par utilisateur :

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

- Une ligne 'A' par table à comparer :

Pos.	Lon.	Valeur	Signification
1	1	'S'	Code mouvement
2	1	'A'	Code carte
3	6	tttttt	Numéro de la table
9	8	JJMMSSAA	Date de m-à-j (borne inférieure)
17	2		Non utilisé
19	1	'/'	Délimiteur
20	1		Non utilisé
21	8	JJMMSSAA	Date de m-à-j (borne supérieure)

La présence d'une seule ligne 'A' sans numéro de table permet d'extraire les postes modifiés de toutes les tables auxquelles l'utilisateur a accès.

CVTA - Description des étapes

Comparaison des données des tables : PTAV10

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7TE	Système - Rép. skel : TE	Entrée	Fichier des libellés d'erreur
PAC7TV	Rép. base : TV	Entrée	Fichier des contenus de tables
PAC7TG	Rép. base : TG	Entrée	Fichier des paramètres utilisateur
PAC7MV		Entrée	Mouvements de demande de sélection
PAC7EX	Rép. tmp : EX	Sortie	Résultat de la comparaison
PAC7ET	Rép. user : CVTAETV10.txt	Etat	Compte-rendu des mouvements

Extraction des mouvements de mise à niveau : PTAV20

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7EX	Rép. tmp : EX	Entrée	Résultat de la comparaison
PAC7NU	Rép. user : MBUPTA.txt	Sortie	Mouvements de mise à niveau pour proc. UPTA
PAC7ET	Rép. user : CVTAETV20.txt	Etat	Edition des données extraites

CVTA - Script d'exécution

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) CVTA BATCH PROCEDURE
# * -----
# *   VA PACTABLES
# *
# * -----
# *           - DIRECT CONSULTATION OF TABLES -
# *
# * -----
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE (UUUUUUUU)
# * COL 11-18  : PASSWORD (PPPPPPPP)
# *
# * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
# * COL 1      : "S" : TRANSACTION CODE
# * COL 2      : "A" : LINE CODE
```

```

# * COL 3-8      : TABLE NUMBER (TTTTTT)
# * COL 9-16    : UPDATE DATE : BEGINNING (DDMMCCYY)
# * COL 17-18   : NOT USED
# * COL 19      : "/" : DELIMITER
# * COL 20      : NOT USED
# * COL 21-28   : UPDATE DATE : END (DDMMCCYY)
# *
# * -----
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----"
TABMSG 1004 "CVTA"
echo "
=====
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TV.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MV=$TABINPUT
export PAC7MV
PAC7EX=`TABENV PAC7EX BVPTAV10 $TABTMP/EX`
export PAC7EX
PAC7ET=`TABENV PAC7ET BVPTAV10 $TABUSERS/CVTAETV10.txt`
export PAC7ET
TABMSG 1009 "BVPTAV10"
rtstab BVPTAV10
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTAV10"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
. $TABDIR/config/$1/PAC7TD.ini
PAC7EX=`TABENV PAC7EX BVPTAV20 $TABTMP/EX`
export PAC7EX
PAC7NU=`TABENV BVPTAD20 PAC7NU `dirname $TABUSERS`/MBUPTA.txt`
export PAC7NU
PAC7ET=`TABENV PAC7ET BVPTAV20 $TABUSERS/CVTAETV20.txt`
export PAC7ET

```



```
TABMSG 1009 "BVPTAV20"
rtstab BVPTAV20
RETURN=$?
case $RETURN in
0)
;;
*)
TABMSG 1012 "BVPTAV20"
TABMSG 1025
TABERR
TABRMTMP
exit $RETURN
;;
esac
# *****
TABMSG 1010
TABRMTMP
exit $RETURN
```




Référence : DETIX000351F - 7347

Imprimé en France