

VisualAge Pacbase



Pactables - Windows 2000 ou NT

Version 3.5



VisualAge Pacbase



Pactables - Windows 2000 ou NT

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.

Deuxième édition (Avril 2007)

La présente édition s'applique à :

- VisualAge Pacbase Version 3.5

Vous pouvez nous adresser tout commentaire sur ce document (en indiquant sa référence) via le site Web de notre Support Technique à l'adresse suivante : <http://www.ibm.com/software/awdtools/vapacbase/support.html> ou en nous adressant un courrier à :

IBM Paris Laboratory
1, place Jean-Baptiste Clément

93881 Noisy-le-Grand, France.

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

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

Table des matières

Notices	v	INTA - Présentation générale	23
Marques	vii	INTA - Entrée utilisateur	24
Chapitre 1. Préambule	1	INTA - Description des étapes	24
Chapitre 2. Les composants de Pactables	3	INTA - Script d'exécution	25
Présentation générale	3	GETT - Génération de tables	26
Les macro-structures du module TUF-TP.	3	GETT - Présentation générale	26
Les fichiers système	4	GETT - Description des étapes	26
Les fichiers évolutifs	4	GETT - Script d'exécution	27
Limitations standard	6	UPTA - Mise à jour des tables	28
Les modules d'accès utilisateur	6	UPTA - Présentation générale	28
Chapitre 3. Installation	7	UPTA - Entrée utilisateur	28
Prérequis	7	UPTA - Description des étapes	29
Matériel et logiciel	7	UPTA - Script d'exécution	30
Espace disque	7	PRTA - Edition des tables	32
Installation	8	PRTA - Présentation générale	32
Présentation générale	8	PRTA - Entrée utilisateur	33
Installation du système	8	PRTA - Description des étapes	34
Installation du référentiel	9	PRTA - Script d'exécution	34
Base Pactables	9	IMTA - Importation de tables	36
Paramètres de configuration	9	IMTA - Présentation générale	36
Déréférencer une base	9	IMTA - Entrée utilisateur	37
Connexion	10	IMTA - Description des étapes	38
Lancement et Arrêt d'un serveur	10	IMTA - Script d'exécution	39
Connexion d'un émulateur 3270	11	RETA - Réorganisation des tables	40
Tests d'utilisation	12	RETA - Présentation générale	40
Chapitre 4. Réinstallation/Suppression des composants	15	RETA - Entrée utilisateur	40
Présentation générale	15	RETA - Description des étapes	41
Lancement de la procédure	15	RETA - Script d'exécution	43
Chapitre 5. Les procédures batch	17	SVTA - Sauvegarde	45
Présentation générale	17	SVTA - Présentation générale	45
Classification des procédures	17	SVTA - Description des étapes	45
Anomalies d'exécution	18	SVTA - Script d'exécution	46
Liste des "run-time errors"	19	TCTA - Transposition d'une autre plateforme	47
Gestion des erreurs dans les procédures	20	TCTA - Présentation générale	47
Définition d'une procédure	20	TCTA - Description des étapes	48
Lancement d'une procédure	22	TCTA - Script d'exécution	49
INTA - Initialisation de tables	23	RSTA - Restauration	50
		RSTA - Présentation générale	50
		RSTA - Description des étapes	51
		RSTA - Script d'exécution	52
		LDTA - Liste descriptifs de tables	53
		LDTA - Présentation générale	53
		LDTA - Entrée utilisateur	53
		LDTA - Description des étapes	53
		LDTA - Script d'exécution	53

PMTA - Mise à jour des paramètres	54	TUTA - Description des étapes	63
PMTA - Présentation générale	54	TUTA - Script d'exécution	63
PMTA - Entrée utilisateur	55	Gestionnaire de tables réparties	64
PMTA - Description des étapes.	56	CDT1-CDT2 - Comparaison de descriptifs	64
PMTA - Script d'exécution	57	CDT1 - Entrée utilisateur.	65
EXTA - Extraction de tables	59	CDT1 - Description des étapes	65
EXTA - Présentation générale	59	CDT1 - Script d'exécution	66
EXTA - Entrée utilisateur.	59	CDT2 - Description des étapes	67
EXTA - Description des étapes	60	CDT2 - Script d'exécution	67
EXTA - Script d'exécution	60	CVTA - Mise à niveau contenus tables	68
TUTA - Mise en exploitation	62	CVTA - Entrée utilisateur	68
TUTA - Présentation générale	62	CVTA - Description des étapes.	69
TUTA - Entrée utilisateur	62	CVTA - Script d'exécution	69

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.

L'installation de cette version ne nécessite aucune reprise.

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 SYS/PGM)
- 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	Répertoire SYS/SKEL
Nom interne	PAC7TE

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	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	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	DATA/"Nom Base"/BASE
Nom interne	PAC7TG

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

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

La sauvegarde (TC)

Caractéristique	Valeur
Organisation	Séquentielle
Longueur	1061
Utilisation	Batch
Localisation	DATA/"Nom Base"/SAVE
Nom interne	PAC7TC

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 fichier d'extension '.DLL' compilés dans l'environnement Windows/NT de Pactables, qui peuvent donc être utilisés tels quels avec le même runtime Micro Focus. (Sous le répertoire "racine"\Pactables\SYS\PGM.)

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

Chapitre 3. Installation

Prérequis

Matériel et logiciel

- Architecture : Un serveur Windows/NT, Windows 2000, Windows XP ou Windows 2003.
- Processeur : Poste supportant soit Windows NT server (4.0 minimum), soit Windows 2000, soit Windows XP, soit Windows 2003.
- Mémoire : 96 Mo. La mémoire nécessaire peut augmenter en fonction du nombre de serveurs installés sur la même machine.
- Logiciels :
 - Microsoft Windows Script, version 5.1 minimum. Vous pouvez la télécharger à partir de l'URL suivante :
www.microsoft.com/downloads
- Pour la version MICROFOCUS :
MICRO FOCUS Application Server
- 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 de 6,5 millions d'octets environ.

Installation

Présentation générale

Le produit est livré sur support CD-Rom.

L'installation requiert les droits Administrateur NT.

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

C:\Program Files\IBM\VisualAge Pacbase\

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

1. installation du Système, qui crée le répertoire \SYS\
2. installation d'une Base Pactables, qui crée le répertoire \CONFIG\[nom_base]\ et le répertoire \DATA\[nom_base]\

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

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

ATTENTION :

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

Installation du système

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

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

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

L'installation copie :

- le fichier de conversions, le fichier des messages produits par les procédures en cas d'erreur ou pour information

- les programmes
- les procédures
- l'utilitaire de création de base
- les modèles des scripts de lancement
- le fichier des libellés d'erreur.

Installation du référentiel

Base Pactables

Vous pouvez installer une Base Pactables soit directement après l'installation système, soit indépendamment via le 'Menu Démarrer' dans lequel vous pouvez accéder au raccourci [Create New Database] du groupe [VisualAge Pacbase 3.5 Pactables Server].

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

- Le nom de la Base (8 car.).
- Le code de la Base (4 car.).
- Le numéro de port de la Base (compris entre 49152 et 65535, unique pour chaque Base).
- Le code langue de la Base (par défaut, Anglais).

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

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

Ainsi, vous pouvez modifier les répertoires de sauvegarde, de fichiers temporaires ou bien encore le répertoire utilisateur du produit dans lequel sont stockés notamment les différents compte-rendus d'exécution des procédures.

Paramètres de configuration

Pour chaque base, on crée un répertoire de Configuration avec

- BvpServer.ini, qui contient les paramètres du serveur conversationnel.

Déréférencer une base

Mise à disposition d'une procédure BVPDELDB pour déréférencer une base sur une plateforme Windows.

Cette procédure effectue les opérations suivantes :

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

Connexion

Lancement et Arrêt d'un serveur

On installe le Listener en mode Service NT.

Les terminaux peuvent ainsi se connecter à Pactables.

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

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

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

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

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

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

- MODE TRACE

Différents niveaux de trace peuvent être mis en oeuvre :

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

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

EXEMPLE :

SRV_TRACE=1 pour une trace de niveau 1

SRV_TRACE=3 pour une trace de niveau 1 et 2

SRV_TRACE=5 pour une trace de niveau 1 et 4

Les fichiers résultat de la trace sont de deux types :

- srv[numéro_process].txt

pour tracer le listener (BvpServer.exe).

- dial[numéro_process].txt

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

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

.../data/[nom_de_base]/tmp

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

SRV_TRACE_DEL : "ON" (valeur par défaut)

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

SRV_TRACE_DEL : "OFF"

on conserve toutes les traces produites.

Connexion d'un émulateur 3270

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

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

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

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

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

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

Code transaction des paramètres utilisateur : "TABPARM".

Code transaction d'accès au module pactables : "PACTABLES".

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. BVPTA400 et BVPTA410),
 - Réorganisation de TD (Prog. BVPTA420),
 - Constitution du fichier sauvegarde TC (BVPTA430),
 - 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/Suppression des composants

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

Lancement de la procédure

Le CD du produit Pactables vous propose également les options suivantes :

- "Update" pour mettre à jour le produit Pactables déjà installé.
- "Remove" pour désinstaller le produit Pactables.

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

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 à 12, accompagné d'un message émis sur le fichier .LOG de la procédure.

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

```

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

```

La présence de ce message est conditionnée par le positionnement, au préalable, de la variable "NoDisp" à "NO" dans Intb.vbs.

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
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 "run-time errors"

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

Numéro	Signification
-----	-----

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

Gestion des erreurs dans les procédures

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.

Définition d'une procédure

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

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

- une zone <resource id> constituée des entrées (141 lignes maximum) ou la définition d'un fichier des entrées par `WshEnv("BVP_Input") = "full file name"` (c'est la valeur prioritaire),
- le code utilisateur,
- le nom externe de la base (répertoire sous \DATA),
- les assignations complètes des fichiers paramétrables de la procédure, exemple : `WshEnv("PTA310_PAC7NK") = "External table file"`.

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

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

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

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

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

Ils sont supprimés à la fin de l'exécution de la procédure. Si vous ne désirez pas cette suppression, il faut, au préalable, positionner la variable "DelDo" à "NO" dans Intb.vbs. Alors, à vous de les détruire manuellement. Cette variable une fois valorisée ("NO" ou "YES") est valable pour TOUTES les procédures.

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

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

ATTENTION :

Dans Pactables pour Acucorp, lorsqu'une procédure comportant un tri cobol s'exécute anormalement, il arrive que les fichiers temporaires de tri ne soient pas automatiquement supprimés. Ces fichiers, au nombre de 16, dont le nom commence par "T" , se trouvent dans le répertoire temporaire de WINDOWS (\TMP ou \TEMP) ou directement sous "C:\". Ils possèdent la date et l'heure du lancement de la procédure.

Lancement d'une procédure

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

- via une ligne de commande,
- en double-cliquant dessus,
- via le 'Menu Démarrer'. Accédez au raccourci : [[nom_base] Database Utilities] du groupe [VisualAge Pacbase 3.5 Pactables Server] de [nom-base] et saisissez [Nom_Script].

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

Pour chaque procédure on définit plusieurs éléments nécessaires à son exécution.

1. Soit dans le Script de lancement : Ce script de lancement est en Windows Scripting identifié comme suit : BVPxxxx.wsf.
 - a. Code utilisateur ,
 - b. Nom externe de la base (répertoire sous \DATA) ,
 - c. Les données en entrée sont décrites dans la zone <resource id> du script et récupérées dans une variable d'environnement BVP_Resource. Le volume des données de cette zone est limité à 141 lignes maximum, mais son utilisation peut être remplacée par celle d'un fichier désigné dans la variable d'environnement BVP_Input. La limitation du volume des données est ainsi éliminée.
 - d. Les fichiers autres que le fichier des entrées peuvent être assignés "en substitution" en définissant une variable d'environnement avec :
 - . "NomStep_NomFichier" pour une assignation substituée dans une étape (ou "step"),
 - . ou "NomProc_NomFichier" pour une assignation substituée dans toute la procédure.Exemple : Définition de la variable d'environnement
WshEnv("PTA310_PAC7NK") = Fil_In with pathname pour une assignation substituée dans l'étape PTA310 de la procédure IMTA.
 - e. Les messages d'information ou d'erreur sont
 - soit affichés à l'écran (MsgTyp = 1),
 - soit édités dans un fichier (MsgTyp =2) ; c'est la valeur par défaut. La variable d'environnement BVP_Msg permet de paramétrer le choix.

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

2. Soit dans la partie commune à toutes les procédures : Cette partie commune est une inclusion VBScript, dans chaque procédure, ayant pour nom "Intb.vbs".
 - a. Récupération des arguments : [base] et [user]
 - b. Affectation de 2 paramètres avec 2 valeurs possibles :
DelDo = "YES"/"NO" :
les fichiers temporaires sont supprimés (YES) ou non (NO)
NoDisp = "YES"/"NO" :
les "display" cobol ne sont pas affichés (YES) ou sont affichés par cmd (NO)
 - c. Constitution du fichier MB, données en entrée, comportant le Code Utilisateur, le Mot de passe et éventuellement les entrées utilisateur de la procédure.

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

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

Remarques :

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

Désormais, on définit le fichier TC (sauvegarde de la Base) par une variable d'environnement : WshEnv("BVP_SaveName"). Cette variable permet de créer le paramètre de définition de l'assignation PAC7TC. Si cette variable d'environnement n'est pas présente dans le script, la valeur par défaut est : [Rep_SAVE & "\TC]

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

(*) Réservé à d'autres plateformes qu'UNIX et WNT.

INTA - Description des étapes

Initialisation des fichiers : BVPTAINI

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 : INTAED.txt	Etat	Compte-rendu d'initialisation

INTA - Script d'exécution

```
REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *                   - TABLE INITIALIZATION -
REM *
REM * -----
REM *
REM * INPUT
REM * COL 1-36   : INSTALLATION LABEL
REM * COL 37    : LANGUAGE VERSION PARAMETER
REM *           "E" ENGLISH
REM *           "F" FRENCH
REM * COL 38    : NOT USED
REM * COL 39-50 : FUNCTION KEYS ASSIGNMENTS
REM * COL 56-57 : "NN" NUMBER OF LINES PER PRINTOUT PAGE
REM *
REM * -----
<job id=INTA>

<script language="VBScript">
MyProc = "INTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTAINI"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MD") = Fic_Input
Call BvpEnv("PTAINI","PAC7ED",Rep_USR & "\INTAED.txt")
Call RunCmdLog ("BVPTAINI")
Call Err_Cod(Return , 0 , "PTAINI")

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

</script>
</job>
```

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.

Le paramètre MD du JCL doit être valorisé avec le DSN du fichier PAC7MD issu du step PACT40 de la procédure GETA ou GETD si la description est destinée à un Pactables d'une version supérieure ou égale à 2.0.

Sinon, il doit être valorisé avec le DSN du fichier PAC7ND issu du step PACT45.

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 : BVPTA250

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 : WTK.tmp	Sortie	Fichier en sortie
PAC7ET	Rép. user : GETTET250.txt	Etat	Erreur entrée/sortie sur fichiers

Edition des descriptifs : BVPTA290

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base: TD	Entrée	Fichier de description de tables
PAC7TE	Rép. tmp : WTK.tmp	Entrée	Fichier mouvement de demande d'édition

Code	Nom physique	Type	Libellé
PAC7ID	Rép. user : GETTID290.txt	Etat	Edition des descriptifs

GETT - Script d'exécution

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLE GENERATION -
REM *
REM * -----
REM * INPUT : RESULT OF GETA OR GETD PROCEDURE
REM *
REM * -----
<job id=GETT>

<script language="VBScript">
MyProc = "GETT"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTA250"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MD") = Fic_Input
Call BvpEnv("PTA250","PAC7TK",Rep_TMP & "\WTK.tmp")
Call BvpEnv("PTA250","PAC7ET",Rep_USR & "\GETTET250.txt")
Call RunCmdLog ("BVPTA250")
Call Err_Cod(Return , 0 , "PTA250")

Call Msg_Log (Array("1022" , "PTA290"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTA290","PAC7TE",Rep_TMP & "\WTK.tmp")
Call BvpEnv("PTA290","PAC7ID",Rep_USR & "\GETTID290.txt")
Call RunCmdLog ("BVPTA290")
Call Err_Cod(Return , 0 , "PTA290")

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

</script>
</job>

```

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 BVPTA302 est fournie à partir de la version 2.0.

Lors des mises à jour, le programme BVPTA300 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 BVPTA302 fourni doit être renommé et utilisé en lieu et place du programme BVPTA300.

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 : BVPTA300

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 : WDE.tmp	Sortie	Demande d'édition
PAC7ET	Rép. user : UPTAET.txt	Etat	Compte-rendu des mouvements
PAC7MT	Rép. tmp : WMT.tmp		Mouvements mis en forme

Mise en forme de l'édition : BVPTA350

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 : WDE.tmp	Entrée	Demande d'édition
PAC7ET	Rép. tmp : WET.tmp	Sortie	Fichier d'édition
PAC7EX	Rép. user : UPTAEX.txt	Etat	Statistiques d'édition

Edition : BVPTA360

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

UPTA - Script d'exécution

```
REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLE UPDATE -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE
REM * COL 11-18  : PASSWORD
REM *
REM * --> ONE "A" TYPE LINE PER TABLE TO UPDATE
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
REM * COL 17    : NOT USED
REM * COL 18    : SUB-SYSTEM NUMBER
REM *           : " " : NO SUB-SYSTEM SPECIFIED
REM *           : 1 TO 0 : SUB-SYSTEM NUMBER
REM * COL 19    : DATA DELIMITER
REM *           : " " : CONSIDER AS "/" BY DEFAULT
REM * --> ONE "V" TYPE LINE TO UPDATE TABLE DATA
REM * COL 1     : ACTION CODE
REM *           : "C" : CREATION
REM *           : "M" : MODIFICATION
REM *           : "D" : DELETION
REM * COL 2     : "V" : LINE CODE
```



```

REM * COL 3      : CONTINUATION LINE
REM *           : " " : FIRST DATA LINE
REM *           : "-" : ITEM DATA CONTINUATION
REM * COL 4-80   : TABLE DATA SEPARATED BY THE DELIMITER
REM *           : INDICATED ON THE "A" TYPE LINE
REM *
REM * -----
<job id=UPTA>

<script language="VBScript">
MyProc = "UPTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTA300"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MS") = Fic_Input
Call BvpEnv("PTA300","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA300","PAC7MT",Rep_TMP & "\WMT.tmp")
Call BvpEnv("PTA300","PAC7ET",Rep_USR & "\UPTAET.txt")
Call RunCmdLog ("BVPTA300")
Call Err_Cod(Return , 0 , "PTA300")

Call Msg_Log (Array("1022" , "PTA350"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
Call BvpEnv("PTA350","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA350","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA350","PAC7EX",Rep_USR & "\UPTAEX.txt")
Call RunCmdLog ("BVPTA350")
Call Err_Cod(Return , 0 , "PTA350")

Call Msg_Log (Array("1022" , "PTA360"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTA360","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA360","PAC7EY",Rep_USR & "\UPTAEY.txt")
Call RunCmdLog ("BVPTA360")
Call Err_Cod(Return , 0 , "PTA360")

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

```

```
Call DeleteFldr(Rep_TMP)
```

```
</script>
```

```
</job>
```

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. (sauf par l'utilisateur "*****", d'une manière générale il est déconseillé de créer un code utilisateur comportant des caractères spéciaux).

Dans ce cas, vous devez saisir dans l'écran LJ le script nécessaire au lancement de la procédure.

Voici un exemple :

```
TY LIGNE LIBELLE
TJ 111111 <job id = BVPPRTA>
TJ 111222 <script language="VBScript">
TJ 111333 Dim WshSh
TJ 111444 Set WshSh = WScript.CreateObject("WScript.Shell")
TJ 111450 Set WshEnv = WshSh.Environment("PROCESS")
TJ 111460 Set Args = Wscript.Arguments
TJ 111470 </script>
TJ 111490 <resource id="PRTA"><![CDATA[
TJ 611111 ]]></resource>
TJ 611222 <script language="VBScript">
TJ 611444 Dim FSO
TJ 611450 Set FSO=CreateObject("Scripting.FileSystemObject")
TJ 611460 Rep = "HKLM\SOFTWARE\IBM" & _
TJ 611470 "\BVP VisualAge Pacbase 3.5\Pactables"
TJ 611490 Rep_PROC = WshSh.RegRead (Rep & "\BVP_SYS\PROC")
TJ 612111 USER=Args(0) 'user code
TJ 612222 BASE=Args(1) 'database code
```

```

TJ 612333 'JOB=ARGS(2) 'NUMÉRO DE JOB
TJ 612444 WshEnv("BVP_Resource") = getresource("PRTA")
TJ 621111 PROC=Rep_PROC & "\prta.wsf"
TJ 621333 CMD=chr(34) & PROC & chr(34) & " " & BASE & _
TJ 621444 " " & USER
TJ 621470 RetPrta = WshSh.Run(CMD , 1 , TRUE )
TJ 621490 </script>
TJ 621500 </job>

```

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
		'P'	Edition paramètres généraux
		'S'	Liste des sous-schémas et sous-systèmes
		'X'	Liste des postes avec historiques
2	1	'A'	Code carte
3	6	ttttt	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

Pos.	Lon.	Valeur	Signification
		' '	Edition des rubriques groupées
		'O'	Edition des rubriques séparées

PRTA - Description des étapes

Extraction des tables à éditer : BVPTA320

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 : WDE.tmp	Sortie	Demandes d'édition
PAC7XE	Rép. user : PRTAXE320.txt	Etat	Compte-rendu des mouvements

Mise en forme de l'édition : BVPTA350

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 : WDE.tmp	Entrée	Demandes d'édition
PAC7ET	Rép. tmp : WET.tmp	Sortie	Fichier d'édition
PAC7EX	Rép. user : PRTAEX350.txt	Etat	Statistiques d'édition

Edition : BVPTA360

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

PRTA - Script d'exécution

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----

```

```

REM *                - TABLE PRINT -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2          : "*" : LINE CODE
REM * COL 3-10       : USER CODE
REM * COL 11-18      : PASSWORD
REM *
REM * --> ONE "A" TYPE LINE PER TABLE TO BE PRINTED
REM * COL 1          : ACTION CODE
REM *                : "E" : TABLE PRINTING
REM *                : "H" : LIST OF HISTORICAL ACCOUNTS
REM *                : "L" : LIST OF THE TABLES
REM *                : "S" : LIST OF SUB-SCHEMAS AND SUB-SYSTEMS
REM *                : "X" : TABLE CONTENTS WITH HISTORICAL
REM *                :     ACCOUNTS
REM * COL 2          : "A" : LINE CODE
REM * COL 3-8        : TABLE NUMBER (TTTTTT)
REM * COL 9-16       : HISTORICAL ACCOUNT DATE OR DATE OF THE
REM *                : REFERENCE DESCRIPTION
REM * COL 17         : SUB-SCHEMA SELECTION
REM *                : BLANK : NO SUB-SCHEMA SELECTION
REM *                : 1 TO 0 : SELECTED SUB-SCHEMA NUMBER
REM * COL 18         : SUB-SYSTEM SELECTION
REM *                : BLANK : NO SUB-SYSTEM SELECTION
REM *                : 1 TO 0 : SELECTED SUB-SYSTEM NUMBER
REM * COL 19         : PRINT OPTION OF THE KEY'S DATA ELEMENTS
REM *                : BLANK : CONCATENED DATA ELEMENTS
REM *                : "0"  : SEPARATED DATA ELEMENTS
REM *
REM * -----
<job id=PRTA>

<script language="VBScript">
MyProc = "PRTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTA320"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7CA") = Fic_Input
Call BvpEnv("PTA320","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA320","PAC7XE",Rep_USR & "\PRTAXE320.txt")
Call RunCmdLog ("BVPTA320")

```

```

Call Err_Cod(Return , 0 , "PTA320")

Call Msg_Log (Array("1022" , "PTA350"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
Call BvpEnv("PTA350","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA360","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA360","PAC7EX",Rep_USR & "\PRTAEX350.txt")
Call RunCmdLog ("BVPTA350")
Call Err_Cod(Return , 0 , "PTA350")

Call Msg_Log (Array("1022" , "PTA360"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTA360","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA360","PAC7EY",Rep_USR & "\PRTAEY360.txt")
Call RunCmdLog ("BVPTA360")
Call Err_Cod(Return , 0 , "PTA360")

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

</script>
</job>

```

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 BVPTA312 est fourni à partir de la version 2.0.

Lors des mises à jour, le programme BVPTA310 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 BVPTA312 fourni doit être renommé et utilisé à la place du programme BVPTA310.

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	ttttt	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 : BVPTA310

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	Entrée	Fichier table externe
PAC7DE	Rép. tmp : WDE.tmp	Sortie	Demandes d'édition
PAC7ET	Rép. user : IMTAET310.txt	Etat	Compte-rendu

Mise en forme de l'édition : BVPTA350

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 : WDE.tmp	Entrée	Demandes d'édition
PAC7ET	Rép. tmp : WET.tmp	Sortie	Fichier d'édition
PAC7EX	Rép. user : IMTAEX350.txt	Etat	Statistiques d'édition

Edition : BVPTA360

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

IMTA - Script d'exécution

```
REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLE IMPORT -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE
REM * COL 11-18  : PASSWORD
REM *
REM * --> ONE "A" TYPE LINE PER TABLE TO BE IMPORTED
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER TO BE IMPORTED (TTTTTT)
REM * COL 9-16   : TABLE DATE (OPTIONAL) (DDMMCCYY)
REM *
REM * -----
<job id=IMTA>

<script language="VBScript">
MyProc = "IMTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTA310"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MV") = Fic_Input
Call BvpEnv("PTA310","PAC7NK",RepT_USR & "\TABF")
Call BvpEnv("PTA310","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA310","PAC7ET",Rep_USR & "\IMTAET310.txt")
Call RunCmdLog ("BVPTA310")
Call Err_Cod(Return , 0 , "PTA310")

Call Msg_Log (Array("1022" , "PTA350"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
Call BvpEnv("PTA350","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA350","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA350","PAC7EX",Rep_USR & "\IMTAEX350.txt")
Call RunCmdLog ("BVPTA350")
```

```

Call Err_Cod(Return , 0 , "PTA350")

Call Msg_Log (Array("1022" , "PTA360"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTA360","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA360","PAC7EY",Rep_USR & "\IMTAEY360.txt")
Call RunCmdLog ("BVPTA360")
Call Err_Cod(Return , 0 , "PTA360")

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

</script>
</job>

```

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	ttttt	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 à 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 : BVPTA400

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

Code	Nom physique	Type	Libellé
PAC7DR		Entrée	Demande de réorganisation
PAC7TX	Rép. tmp : WTX.tmp	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 : BVPTA410

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

Réorganisation des descriptifs : BVPTA420

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
PAC7TS	Rép. tmp : WTS.tmp	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 : BVPTA430

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

Sauvegarde de TG : BVPTASVG

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

RETA - Script d'exécution

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLE REORGANIZATION -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE IDENTIFYING PACTABLES MANAGER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : PACTABLES MANAGER CODE
REM * COL 11-18  : PACTABLES MANAGER PASSWORD
REM *
REM * --> ONE "A" TYPE LINE PER HISTORICAL ACCOUNT TO KEEP
REM *   OR DELETE
REM * COL 1      : ACTION CODE
REM *           : "S" : HISTORICAL ACCOUNT TO PURGE
REM *           : "G" : HISTORICAL ACCOUNT TO KEEP
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
REM * COL 19     : OPTION
REM *           : " " : SEE DOCUMENTATION FOR MORE
REM *           : "X" : EXPLANATIONS
REM *           : "X"
REM *
REM * -----
<job id=RETA>

```

```

<script language="VBScript">
MyProc = "RETA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTA400"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7DR") = Fic_Input
Call BvpEnv("PTA400","PAC7TX",Rep_TMP & "\WTX.tmp")
Call BvpEnv("PTA400","PAC7DE",Rep_USR & "\MVRETA.txt")
Call BvpEnv("PTA400","PAC7IR",Rep_USR & "\RETAIR400.txt")
Call RunCmdLog ("BVPTA400")
Call Err_Cod(Return , 0 , "PTA400")

Call Msg_Log (Array("1022" , "PTA410"))
'-----
Call BvpEnv("PTA410","PAC7TX",Rep_TMP & "\WTX.tmp")
Call BvpEnv("PTA410","PAC7TW",Rep_TMP & "\WTW.tmp")
WshEnv("PAC7MB") = Fic_Input
Call RunCmdLog ("BVPTA410")
Call Err_Cod(Return , 0 , "PTA410")

Call Msg_Log (Array("1022" , "PTA420"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTA420","PAC7TS",Rep_TMP & "\WTS.tmp")
Call BvpEnv("PTA420","PAC7DE",Rep_USR & "\MVRETA.txt")
Call BvpEnv("PTA420","PAC7ML",Rep_USR & "\MWRETA.txt")
Call RunCmdLog ("BVPTA420")
Call Err_Cod(Return , 0 , "PTA420")

Call Msg_Log (Array("1022" , "PTA430"))
'-----
WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
Call BvpEnv("PTA430","PAC7TW",Rep_TMP & "\WTW.tmp")
Call BvpEnv("PTA430","PAC7TS",Rep_TMP & "\WTS.tmp")
Call RunCmdLog ("BVPTA430")
Call Err_Cod(Return , 0 , "PTA430")

Call Msg_Log (Array("1022" , "PTASVG"))
'-----
WshEnv("PAC7TG") = Rep_BASE & "\TG"

```

```

WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
Call RunCmdLog ("BVPTASVG")
Call Err_Cod(Return , 0 , "PTASVG")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\TC")

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

</script>
</job>

```

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 : BVPTASVT

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

Sauvegarde de TV : BVPTASVV

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 : BVPTASVG

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

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLES BACKUP -
REM *
REM * -----
<job id=SVTA>

<script language="VBScript">
MyProc = "SVTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTASVT"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
Call RunCmdLog ("BVPTASVT")
Call Err_Cod(Return , 0 , "PTASVT")

Call Msg_Log (Array("1022" , "PTASVV"))
'-----
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
Call RunCmdLog ("BVPTASVV")
Call Err_Cod(Return , 0 , "PTASVV")

Call Msg_Log (Array("1022" , "PTASVG"))
'-----
WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
Call RunCmdLog ("BVPTASVG")
Call Err_Cod(Return , 0 , "PTASVG")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\TC")

```



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

</script>
</job>
```

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 : BVPTATC1

Code	Nom physique	Type	Libellé
PAC7TC	Rép. save : TC.INI	Entrée	Fichier sauvegarde
PAC7SD	Rép. tmp : WSD.tmp	Sortie	Image séquentielle des descriptifs
PAC7SV	Rép. tmp : WSV.tmp	Sortie	Image séquentielle des contenus
PAC7SG	Rép. tmp : WSG.tmp	Sortie	Image séquentielle des paramètres

Tri des descriptifs de tables : BVPTATCD

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

Tri des contenus des tables : BVPTATCV

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

Tri des paramètres utilisateur : BVPTATCG

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

Reconstitution de la sauvegarde TC : BVPTATC2

Code	Nom physique	Type	Libellé
PAC7AD	Rép. tmp : WAD.tmp	Entrée	Image séquentielle des descriptifs
PAC7AV	Rép. tmp : WAV.tmp	Entrée	Image séquentielle des contenus
PAC7AG	Rép. tmp : WAG.tmp	Entrée	Image séquentielle des paramètres
PAC7TC	Rép. save : TC-new	Sortie	Sauvegarde TC au format EBCDIC

TCTA - Script d'exécution

```
REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *   - TABLES TRANSFER FROM ANOTHER PLATFORM -
REM *
REM * -----
<job id=TCTA>

<script language="VBScript">
MyProc = "TCTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTATC1"))
'-----
WSHENV("PAC7TC") = REP_SAVE & "\TC.INI"
Call BvpEnv("PTATC1","PAC7SD",Rep_TMP & "\WSD.tmp")
Call BvpEnv("PTATC1","PAC7SG",Rep_TMP & "\WSG.tmp")
Call BvpEnv("PTATC1","PAC7SV",Rep_TMP & "\WSV.tmp")
Call RunCmdLog ("BVPTATC1")
Call Err_Cod(Return , 0 , "PTATC1")

Call Msg_Log (Array("1022" , "PTATCD"))
'-----
Call BvpEnv("PTATC1","PAC7SD",Rep_TMP & "\WSD.tmp")
Call BvpEnv("PTATC1","PAC7AD",Rep_TMP & "\WAD.tmp")
```

```

Call RunCmdLog ("BVPTATCD")
Call Err_Cod(Return , 0 , "PTATCD")

Call Msg_Log (Array("1022" , "PTATCV"))
'-----
Call BvpEnv("PTATC1","PAC7SV",Rep_TMP & "\WSV.tmp")
Call BvpEnv("PTATC1","PAC7AV",Rep_TMP & "\WAV.tmp")
Call RunCmdLog ("BVPTATCV")
Call Err_Cod(Return , 0 , "PTATCV")

Call Msg_Log (Array("1022" , "PTATCG"))
'-----
Call BvpEnv("PTATC1","PAC7SG",Rep_TMP & "\WSG.tmp")
Call BvpEnv("PTATC1","PAC7AG",Rep_TMP & "\WAG.tmp")
Call RunCmdLog ("BVPTATCG")
Call Err_Cod(Return , 0 , "PTATCG")

Call Msg_Log (Array("1022" , "PTATC2"))
'-----
WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
Call BvpEnv("PTATC2","PAC7AD",Rep_TMP & "\WAD.tmp")
Call BvpEnv("PTATC2","PAC7AG",Rep_TMP & "\WAG.tmp")
Call BvpEnv("PTATC2","PAC7AV",Rep_TMP & "\WAV.tmp")
Call RunCmdLog ("BVPTATC2")
Call Err_Cod(Return , 0 , "PTATC2")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\TC")

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

</script>
</job>

```

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 : BVPTARST

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 : BVPTARSG

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

RSTA - Script d'exécution

```
REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLES RESTORATION -
REM *
REM * -----
<job id=RSTA>

<script language="VBScript">
MyProc = "RSTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTARST"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\\TD"
WshEnv("PAC7TC") = Rep_SAVE & "\\TC"
Call RunCmdLog ("BVPTARST")
Call Err_Cod(Return , 0 , "PTARST")

Call Msg_Log (Array("1022" , "PTARSV"))
'-----
WshEnv("PAC7TV") = Rep_BASE & "\\TV"
WshEnv("PAC7TC") = Rep_SAVE & "\\TC"
Call RunCmdLog ("BVPTARSV")
Call Err_Cod(Return , 0 , "PTARSV")

Call Msg_Log (Array("1022" , "PTARSG"))
'-----
WshEnv("PAC7TC") = Rep_SAVE & "\\TC"
WshEnv("PAC7TG") = Rep_BASE & "\\TG"
Call RunCmdLog ("BVPTARSG")
Call Err_Cod(Return , 0 , "PTARSG")

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

</script>
</job>
```

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 : BVPTA290

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

```
REM * -----  
REM *   VA PACTABLES  
REM *  
REM * -----  
REM *           - LIST OF TABLE DESCRIPTIONS -  
REM *  
REM * -----  
REM *  
REM * INPUT  
REM * --> ONE "Z" TYPE LINE PER PRINT REQUEST
```

```

REM * COL 2      : "Z" : LINE CODE
REM * COL 5-8    : PRINT REQUEST
REM *           : "TLS" : LIST OF TABLE DESCRIPTIONS
REM *           : "TDS" : TABE DESCRIPTION
REM * COL 9-14   : TABLE NUMBER (TTTTTT)
REM * COL 23-30  : HISTORICAL ACCOUNT DATE (MMDDCCYY)
REM *
REM * -----
<job id=LDTA>

<script language="VBScript">
MyProc = "LDTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTA290"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TE") = Fic_Input
Call BvpEnv("PTA290","PAC7ID",Rep_USR & "\LDTAID290.txt")
Call RunCmdLog ("BVPTA290")
Call Err_Cod(Return , 0 , "PTA290")

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

</script>
</job>

```

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
		'M'	Modification
		'A'	Annulation
2	8	uuuuuuuu	Code utilisateur
10	2	'TC'	Code carte
12	6	ttttt	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

Pos.	Lon.	Valeur	Signification
	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 : BVPTA100

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 : WNU.tmp	Sortie	Demande d'édition des paramètres

Code	Nom physique	Type	Libellé
PAC7ET	Rép. user : PMTAET100.txt	Etat	Compte-rendu des mouvements

Edition des paramètres utilisateur : BVPTA120

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 : WNU.tmp	Entrée	Demande d'édition
PAC7ET	Rép. user : PMTAET120.txt	Etat	Edition des paramètres utilisateur

PMTA - Script d'exécution

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - USER PARAMETER UPDATE -
REM *
REM * -----
REM * INPUT
REM * --> ONE "TA" LINE : USER PARAMETER UPDATING
REM * COL 1      : ACTION CODE
REM *           : " " : CREATION OR MODIFICATION
REM *           : "C" : CREATION
REM *           : "M" : MODIFICATION
REM *           : "D" : DELETION
REM * COL 2-9    : USER CODE (UUUUUUUU)
REM * COL 10-11 : "TA" : LINE CODE
REM * COL 12-19 : PASSWORD (PPPPPPPP)
REM * COL 20    : GENERAL ACCESS AUHTORIZATION
REM *           : "0" : NO GENERAL ACCESS AUTHORIZATION
REM *           : "1" : READ ONLY ACCESS AUTHORIZATION
REM *           : "2" : READ WRITE AUTHORIZATION ON TABLES
REM *           : "3" : READ WRITE AUTHORIZATION ON USER
REM *           :      CODE
REM *
REM * --> ONE "TC" LINE : ACCESS AUTHORIZATIONS PER TABLE
REM * COL 1      : ACTION CODE
REM *           : " " : CREATION OR MODIFICATION
REM *           : "C" : CREATION
REM *           : "M" : MODIFICATION
REM *           : "D" : DELETION
REM * COL 2-9    : USER CODE (UUUUUUUU)
REM * COL 10-11 : "TC" : LINE CODE
REM * COL 12-17 : TABLE CODE (TTTTTT)
REM * COL 18-20 : LINE NUMBER (NNN)
REM * COL 21-80 : ACCESS AUTHORIZATIONS : 20 ACCESS

```

```

REM *           : AUTHORIZATIONS MAY BE ENTERED IN THIS
REM *           : FIELD, WITH, FOR EACH AUTHORIZATION, THE
REM *           : SUB-SCHEMA NUMBER (N), THE SUB-SYSTEM
REM *           : NUMBER (N), THE AUTHORIZATION (0,1,2).
REM *           : "*" IS REQUIRED FOR ALL SUB-SCHEMAS AND
REM *           : SUB-SYSTEMS.
REM *
REM * --> ONE "TJ" LINE : CONTROL CARDS
REM * COL 1       : ACTION CODE
REM *           : " " : CREATION OR MODIFICATION
REM *           : "C" : CREATION
REM *           : "M" : MODIFICATION
REM *           : "D" : DELETION
REM * COL 2-9     : USER CODE (UUUUUUUU)
REM * COL 10-11   : "TJ" : LINE CODE
REM * COL 12-17   : TABLE CODE (TTTTTT)
REM * COL 18-20   : SCRIPT LINE NUMBER
REM *           :LT600000 : CONTROL CARD IN FRONT OF PROGRAM
REM *           :GT599999 : CONTROL CARD IN BACK OF PROGRAM
REM * COL 21-80  : CONTENT OF THE SCRIPT LINE
REM *
REM * -----
<job id=PMTA>

```

```

<script language="VBScript">
MyProc = "PMTA"
</script>

```

```

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

```

```

<script language="VBScript">

```

```

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

```

```

Call Msg_Log (Array("1022" , "PTA100"))

```

```

'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7MV") = Fic_Input
Call BvpEnv("PTA100","PAC7NU",Rep_TMP & "\WNU.tmp")
Call BvpEnv("PTA100","PAC7ET",Rep_USR & "\PMTAET100.txt")
Call RunCmdLog ("BVPTA100")
Call Err_Cod(Return , 0 , "PTA100")

```

```

Call Msg_Log (Array("1022" , "PTA120"))

```

```

'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
Call BvpEnv("PTA120","PAC7NU",Rep_TMP & "\WNU.tmp")
Call BvpEnv("PTA120","PAC7ET",Rep_USR & "\PMTAET120.txt")
Call RunCmdLog ("BVPTA120")
Call Err_Cod(Return , 0 , "PTA120")

```

```

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

</script>
</job>

```

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	ttttt	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 : BVPTA150

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 : WEX.tmp	Sortie	Mouvements extraits
PAC7ET	Rép. user : EXTAET150.txt	Etat	Compte-rendu des mouvements

Edition des mouvements extraits : BVPTA160

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7EX	Rép. tmp : WEX.tmp	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

```
REM * -----  
REM *   VA PACTABLES  
REM *  
REM * -----  
REM *           - TABLE EXTRACTION -  
REM *  
REM * -----  
REM *  
REM * INPUT  
REM * --> ONE "*" TYPE LINE PER USER  
REM * COL 2      : "*" : LINE CODE  
REM * COL 3-10   : USER CODE (UUUUUUUU)  
REM * COL 11-18  : PASSWORD (PPPPPPPP)  
REM *  
REM * --> ONE "A" LINE TYPE PER TABLE TO EXTRACT  
REM * COL 2      : "A" : LINE CODE
```

```

REM * COL 3-8      : TABLE NUMBER (TTTTTT)
REM * COL 9-16    : HISTORICAL ACCOUNT DATE (DDMMCCYY)
REM * COL 17      : NOT USED
REM * COL 18      : SUB-SYSTEM SELECTION
REM *              : BLANK : NO SUB-SYSTEM SELECTION
REM *              : 1 TO 0 : NUMBER OF SELECTED SUB SYSTEM
REM * COL 19      : DATA DELIMITEUR
REM *              : BLANK : "/" BY DEFAULT
REM *
REM * -----
<job id=EXTA>

<script language="VBScript">
MyProc = "EXTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTA150"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MV") = Fic_Input
Call BvpEnv("PTA150","PAC7EX",Rep_TMP & "\WEX.tmp")
Call BvpEnv("PTA150","PAC7ET",Rep_USR & "\EXTAET150.txt")
Call RunCmdLog ("BVPTA150")
Call Err_Cod(Return , 0 , "PTA150")

Call Msg_Log (Array("1022" , "PTA160"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTA160","PAC7EX",Rep_TMP & "\WEX.tmp")

'One for each procedure : Rep_USR
'One for all the procedure : RepT_USR

'Call BvpEnv("PTA160","PAC7NU",Rep_USR & "\NUTAB.txt")
Call BvpEnv("PTA160","PAC7NU",RepT_USR & "\NUTAB.txt")

Call BvpEnv("PTA160","PAC7ET",Rep_USR & "\EXTAET160.txt")
Call RunCmdLog ("BVPTA160")

If Return = 0 Then
Call Msg_Log (Array("1058"))
End If
If Return = 8 Then
Call Msg_Log (Array("1059"))
End If

```

```

If Return = 12 Then
Call Msg_Log (Array("1060"))
End If
Call Err_Cod(Return , 0 , "PTA160")

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

</script>
</job>

```

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	tttttt	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 : BVPTAU80

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 : TUTAET.txt	Etat	Compte-rendu des mouvements

TUTA - Script d'exécution

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *   - DIRECT CONSULTATION OF TABLES -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE (UUUUUUUU)
REM * COL 11-18  : PASSWORD (PPPPPPPP)
REM *
REM * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
REM *
REM * -----
<job id=TUTA>

<script language="VBScript">
MyProc = "TUTA"
</script>

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

<script language="VBScript">

```

```

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

Call Msg_Log (Array("1022" , "PTAU80"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MX") = Fic_Input
WshEnv("PAC7AD") = Rep_BASE & "\AD"
WshEnv("PAC7AV") = Rep_BASE & "\AV"
Call BvpEnv("PTAU80","PAC7ET",Rep_USR & "\TUTAET.txt")
Call RunCmdLog ("BVPTAU80")
Call Err_Cod(Return , 0 , "PTAU80")

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

</script>
</job>

```

Gestionnaire de tables réparties

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 : BVPTAD05

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 : WMX.tmp	Sortie	Mouvements valides de demande comparaisons
PAC7ET	Rép. user : CDT1ETD05.txt	Etat	Compte-rendu des mouvements

Comparaison des descriptifs et extraction : BVPTAD10

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
PAC7TS	Rép. user : TD	Entrée	Fichier 'esclave' des descriptifs de tables
PAC7MX	Rép. tmp : WMX.tmp	Entrée	Mouvements valides

Code	Nom physique	Type	Libellé
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

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *       - DISPATCHED TABLE MANAGEMENT -
REM *       - TABLE DESCRIPTION COMPARISON -
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE (UUUUUUUU)
REM * COL 11-18  : PASSWORD (PPPPPPPP)
REM *
REM * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM *
REM * -----
<job id=CDT1>

<script language="VBScript">
MyProc = "CDT1"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTAD05"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
Call BvpEnv("PTAD05","PAC7TD", Rep_BASE & "\TD")
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7MV") = Fic_Input
Call BvpEnv("PTAD05","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PTAD05","PAC7ET",Rep_USR & "\CDT1ETD05.txt")
Call RunCmdLog ("BVPTAD05")
Call Err_Cod(Return , 0 , "PTAD05")

Call Msg_Log (Array("1022" , "PTAD10"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"

```

```

Call BvpEnv("PTAD10","PAC7TD",Rep_BASE & "\TD")
Call BvpEnv("PTAD10","PAC7TS",Rep_USR & "\TD")
Call BvpEnv("PTAD10","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PTAD10","PAC7TX",Rep_USR & "\MBCDT2.txt")
Call BvpEnv("PTAD10","PAC7ET",Rep_USR & "\CDT1ETD10.txt")
Call RunCmdLog ("BVPTAD10")
Call Err_Cod(Return , 0 , "PTAD10")

```

```

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

```

```

</script>
</job>

```

CDT2 - Description des étapes

Mise à niveau des fichiers 'esclaves', Prise en compte du fichier extrait par CDT1 : BVPTAD20

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 : CDT2ET.txt	Etat	Compte-rendu de la mise à jour

CDT2 - Script d'exécution

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *   - DISPATCHED TABLE MANAGEMENT -
REM *   - UPDATE OF TABLE DESCRIPTIONS AFTER COMPARISON -
REM * -----
<job id=CDT2>

<script language="VBScript">
MyProc = "CDT2"
</script>

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

<script language="VBScript">

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

```

```

Call Msg_Log (Array("1022" , "PTAD20"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
Call BvpEnv("PTAD20","PAC7TD",Rep_BASE & "\TD")
Call BvpEnv("PTAD20","PAC7TV",RepT_USR & "\TV")
Call BvpEnv("PTAD20","PAC7TX",RepT_USR & "\MBCDT2.txt")
Call BvpEnv("PTAD20","PAC7ET",Rep_USR & "\CDT2ET.txt")
Call RunCmdLog ("BVPTAD20")
Call Err_Cod(Return , 0 , "PTAD20")

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

</script>
</job>

```

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

Pos.	Lon.	Valeur	Signification
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 : BVPTAV10

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 : WEX.tmp	Sortie	Résultat de la comparaison
PAC7ET	Rép. user : CVTAETV10.txt	Etat	Compte-rendu des mouvements

Extraction des mouvements de mise à niveau : BVPTAV20

Code	Nom physique	Type	Libellé
PAC7TD	Rép. base : TD	Entrée	Fichier des descriptifs de tables
PAC7EX	Rép. tmp : WEX.tmp	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

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - DIRECT CONSULTATION OF TABLES -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE (UUUUUUUU)

```

```

REM * COL 11-18 : PASSWORD (PPPPPPPP)
REM *
REM * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
REM * COL 1      : "S" : TRANSACTION CODE
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM * COL 9-16   : UPDATE DATE : BEGINNING (DDMMCCYY)
REM * COL 17-18  : NOT USED
REM * COL 19     : "/" : DELIMITER
REM * COL 20     : NOT USED
REM * COL 21-28  : UPDATE DATE : END (DDMMCCYY)
REM *
REM * -----
<job id=CVTA>

<script language="VBScript">
MyProc = "CVTA"
</script>

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTAV10"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MV") = Fic_Input
Call BvpEnv("PTAV10","PAC7EX",Rep_TMP & "\WEX.tmp")
Call BvpEnv("PTAV10","PAC7ET",Rep_USR & "\CVTAETV10.txt")
Call RunCmdLog ("BVPTAV10")
Call Err_Cod(Return , 0 , "PTAV10")

Call Msg_Log (Array("1022" , "PTAV20"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTAV10","PAC7EX",Rep_TMP & "\WEX.tmp")
Call BvpEnv("PTAV20","PAC7NU",RepT_USR & "\MBUPTA.txt")
Call BvpEnv("PTAV20","PAC7ET",Rep_USR & "\CVTAETV20.txt")
Call RunCmdLog ("BVPTAV20")
Call Err_Cod(Return , 0 , "PTAV20")

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

</script>
</job>

```




Référence : DETNT000352F - 7618

Imprimé en France