



VisualAge Pacbase 2.5

**Station de Travail VisualAge
Pacbase
Manuel d'exploitation**

DSEXP000251F

Remarque

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section "Remarques" de la page suivante.

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

<http://www.software.ibm.com/ad/vapacbase/support.htm>

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

1ère Edition (août 1998)

La présente édition s'applique à :

- VisualAge Pacbase Version 2.5

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.software.ibm.com/ad/vapacbase/support.htm>

ou en nous adressant un courrier à :

IBM Paris Laboratory
Support VisualAge Pacbase
30, rue du Château des Rentiers
75640 PARIS Cedex 13
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, 1999. Tous droits réservés.

REMARQUES

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 :

Intellectual Property and Licensing
International Business Machines Corporation
North Castle Drive, Armonk, New-York 10504-1785
USA

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
Département SMC
30, rue du Château des Rentiers
75640 PARIS Cedex 13
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, PACBASE, RACF, RS/6000, SQL/DS, TeamConnection 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.

Table des Matières

1	INSTALLATION DE LA STATION DE TRAVAIL	9
1.1	STATION DE TRAVAIL ET SERVEUR	9
1.2	INSTALLATION DE LA PARTIE SERVEUR	10
1.2.1	<i>Touches de fonction</i>	10
1.2.2	<i>Installation d'une méthode</i>	10
1.3	ARBORESCENCE DE LA STATION DE TRAVAIL : ETAT INITIAL ET ÉVOLUTION.....	12
1.3.1	<i>Variantes d'installation</i>	14
1.3.2	<i>Evolution de l'arborescence : Réseau utilisateur</i>	15
1.4	INSTALLATION DE LA PARTIE MICRO : GÉNÉRALITÉS	17
1.4.1	<i>Support d'installation</i>	17
1.4.2	<i>Environnement d'installation</i>	17
1.5	INSTALLATION DE LA PARTIE MICRO : VERSIONS DOS WINDOWS ET WIN-OS/2.....	18
1.5.1	<i>Lancement de l'installation</i>	18
1.5.2	<i>Sélection de la Station et/ou des méthodes à installer</i>	18
1.5.3	<i>Installation d'une nouvelle Station</i>	19
1.5.4	<i>Installation des méthodes</i>	21
1.6	PARAMÉTRAGE MICRO : VERSION DOS WINDOWS 3.X.....	24
1.7	PARAMÉTRAGE MICRO : VERSION DOS WINDOWS 95 ET NT	25
1.8	PARAMÉTRAGE MICRO : VERSION OS/2 WIN-OS/2 (À PARTIR DE 2.1).....	26
2	PERSONNALISATION DE LA STATION DE TRAVAIL.....	27
2.1	SCRIPTS DE CONNEXION ET DE DÉCONNEXION	27
2.1.1	<i>Modes de connexion</i>	27
2.1.2	<i>Présentation</i>	28
2.1.3	<i>Répertoire une base</i>	29
2.1.4	<i>Scripts de connexion</i>	29
2.1.5	<i>Conseils pratiques</i>	30
2.1.6	<i>Cas particulier : Connexion à VisualAge Pacbase (serveurs OS/2, Windows NT ou UNIX)</i>	30
2.1.7	<i>Modes d'utilisation des scripts</i>	30
2.1.8	<i>Mode test</i>	31
2.1.9	<i>Mode exploitation</i>	31
2.1.10	<i>Affichage d'écrans serveur</i>	32
2.1.11	<i>Exemples de scripts IBM-CICS</i>	33
2.1.12	<i>Exemples de scripts BULL-DPS7</i>	36
2.1.13	<i>Exemples de scripts BULL-DPS8</i>	38
2.1.14	<i>Exemples de scripts OS/2, Windows NT ou UNIX</i>	40
2.2	PARAMÉTRAGE DES COMBINAISONS DE TOUCHES ÉQUIVALENTES AUX TOUCHES FONCTION F11 ET F12	41
3	LANCEMENT DE LA STATION DE TRAVAIL.....	43
3.1	FICHER PACBASE.DAT.....	43
3.2	FICHIERS DE PARAMÉTRAGE OPTIONNELS	44
3.3	PROCÉDURE DE LANCEMENT	44
3.3.1	<i>Répertoire de lancement : \SPAC\NNNL</i>	45
3.3.2	<i>Lancement : PEXEC.EXE</i>	46
4	UTILITAIRES DE LA STATION DE TRAVAIL.....	47
4.1	STIMULATEUR EXTERNE	47
4.1.1	<i>Fonctionnalités</i>	47
4.1.2	<i>Activation</i>	47
4.2	APPLICATION ÉDITION LOCALE DE RAPPORTS AU FORMAT RTF	49
4.3	UTILITAIRE DE "NETTOYAGE" DE LA STATION DE TRAVAIL	50

5	ANNEXE LE LANGAGE DE CONNEXION	51
5.1	STRUCTURE D'UN SCRIPT	52
5.2	MOTS RÉSERVÉS	52
5.3	DÉCLARATIONS	53
5.3.1	<i>Types de variables</i>	53
5.3.2	<i>Noms de variables</i>	53
5.3.3	<i>Valeurs des variables</i>	54
5.3.4	<i>Étiquettes</i>	54
5.3.5	<i>Commentaires</i>	55
5.3.6	<i>Espacements et retours à la ligne</i>	55
5.3.7	<i>Corps du programme</i>	56
5.3.8	<i>Instructions</i>	56
5.4	AFFECTATION	56
5.5	EXPRESSIONS ET OPÉRATEURS	57
5.5.1	<i>Priorité des opérateurs</i>	58
5.5.2	<i>Traitement des opérateurs associatifs</i>	58
5.6	BRANCHEMENTS.....	59
5.7	STRUCTURES DE CONTRÔLE.....	59
5.8	ÉCRITURE DES EXPRESSIONS.....	60
5.9	FONCTIONS	62
5.9.1	<i>Appel d'une fonction</i>	62
5.9.2	<i>Paramètres d'une fonction</i>	62
5.10	FONCTIONS DE COMMUNICATION.....	63
5.11	FONCTIONS D'ENTRÉE-SORTIE.....	67
5.12	FONCTIONS TECHNOLOGIQUES.....	70
5.13	ERREURS	71
5.13.1	<i>Erreurs de contexte</i>	71
5.13.2	<i>Erreurs de syntaxe</i>	72
5.13.3	<i>Erreurs à l'exécution</i>	73

Préambule

Ce manuel présente l'installation et l'exploitation de la Station de Travail VisualAge Pacbase (serveurs : mainframe, OS/2, Windows NT ou UNIX).

Il se compose des chapitres qui décrivent respectivement :

- La procédure d'installation de la Station, le paramétrage des environnements graphiques à partir desquels elle peut être utilisée et une description de la structure sur disque dur résultant de son implantation.
- Les scripts de connexion permettant l'accès à la base de spécifications depuis la Station.
- Le lancement de la Station, à partir des environnements graphiques ou directement depuis le système.
- Les utilitaires qui facilitent le lancement et l'arrêt de la Station de Travail et permettent d'éditer des Rapports au format RTF.
- Le langage dont l'utilisateur doit se servir pour créer ou modifier ses scripts de connexion

Pour obtenir les détails concernant les configurations, tant matérielle que logicielle, nécessaires à l'implantation de la Station de Travail, consulter votre support client.

1 Installation de la Station de Travail

1.1 Station de Travail et serveur

La Station de Travail fonctionne avec VisualAge Pacbase qui peut être installé sur :

- un serveur mainframe ;
- un serveur de réseau local OS/2, Windows NT ou UNIX.

Pour que la Station de Travail puisse fonctionner, VisualAge Pacbase doit au préalable avoir été correctement installé sur le serveur choisi.

Pour mener à bien l'installation de VisualAge Pacbase sur votre serveur, consultez le *Manuel d'Exploitation* lié à votre plateforme serveur (Vol. I - *Environnement & Installation*, Chapitre *Installation*).

Lors de la première utilisation de la Station de Travail, assurez-vous que les modules suivants sont bien installés :

- Dictionnaire;
- Gestionnaire de Documentation Personnalisée;
- Personnalisation.

1.2 Installation de la partie serveur

1.2.1 Touches de fonction

Pour chaque base VisualAge Pacbase installée sur mainframe, OS/2, Windows NT ou UNIX que vous voulez rendre accessible par la Station de Travail, si les touches de fonction ne sont pas affectées ou si elles sont différentes des options standard, vous devez :

- Soit les définir selon les options standard;
- Soit (au minimum) affecter la touche PF7 à l'inhibition de la mise à jour implicite.

La procédure REST permet de définir les touches de fonction pour chaque base. Pour plus d'informations sur cette procédure, consultez le *Manuel d'Exploitation* lié à votre plateforme serveur (Vol. II - *Procédures Batch : Guide de l'administrateur*).

1.2.2 Installation d'une méthode

Pour chaque base VisualAge Pacbase installée sur mainframe, OS/2, Windows NT ou UNIX que vous voulez rendre accessible par la Station de Travail, vous devez en outre installer la méthode de conception ou de réalisation choisie dans une bibliothèque dédiée. "Installer une méthode" signifie alimenter la base à partir d'entités spécifiques, qui permettent la mise en oeuvre de cette méthode.

Ces entités sont des Entités Utilisateur pré-définies. Leur définition et leur description ne doivent *en aucun cas* être modifiées après l'installation de la méthode. Pour assurer cela, vous pouvez par exemple interdire aux utilisateurs de la Station de Travail la mise à jour dans la bibliothèque où vous installez la méthode. Cette bibliothèque devra alors être de haut niveau hiérarchique, afin que les utilisateurs de la Station de Travail puissent travailler avec les entités de la méthode dans des bibliothèques dépendantes.

Les supports d'installation du serveur contiennent des fichiers de mouvements liés aux méthodes. Chacun d'eux contient l'ensemble des entités d'une méthode. Pour installer une méthode sur le serveur, vous devez exécuter la procédure UPDT avec en entrée le fichier adéquat.

REMARQUE :

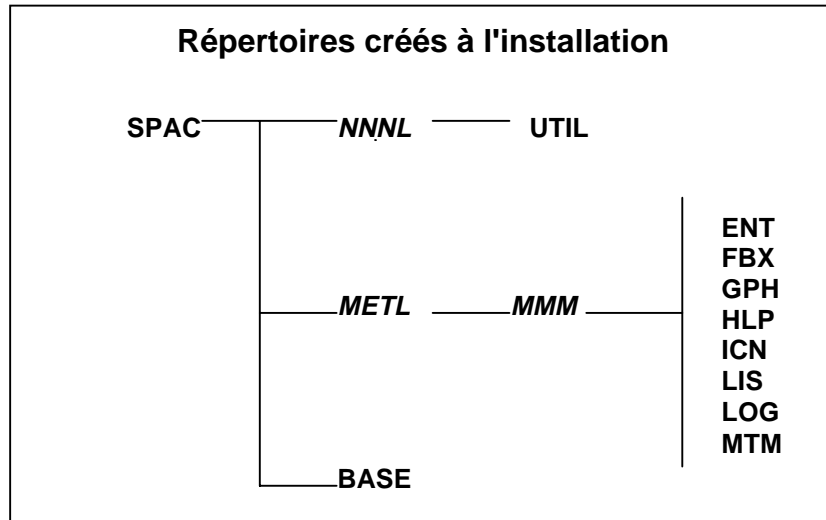
Pour les méthodes client, les fichiers UPDT sont fournis à l'installation dans le répertoire des fichiers méthode : \SPAC\METL\MMM. Pour plus d'informations sur ce répertoire, consultez le Sous-chapitre *Arborescence de la Station de Travail*.

Ce type d'installation est documenté dans le Manuel d'Exploitation VisualAge Pacbase spécifique à votre plateforme (Vol. I - *Environnement et Installation*).

Vous devez par ailleurs assurer l'intégration des choix méthode dans VisualAge Pacbase. Ces choix sont fournis à l'installation sous forme de fichiers de mouvements à utiliser en entrée de la procédure PARM de mise à jour des paramètres utilisateur.

Pour plus de détails, consultez le *Manuel d'Exploitation* de votre plateforme serveur (Vol. II - *Procédures Batch - Guide de l'administrateur, chapitre "Gestion de la Base", sous-chapitre "PARM : Gestion des Paramètres Utilisateur"*).

1.3 Arborescence de la Station de Travail : Etat initial et évolution



Dans ce schéma :

- *NNN* désigne le numéro de version;
- *L* est le code langue (F pour français et A pour anglais);
- *MMM* est le code de la méthode : MER (MERISE), ADM (SSADM), DON (YSM), FAA (IFW) ou OMT.

La procédure d'installation crée les répertoires suivants :

- \SPAC : racine de l'arborescence de la Station de Travail que vous pouvez installer à l'emplacement de votre choix.
- \SPAC\NNNL : ce répertoire contient notamment :
 - tous les fichiers exécutables (.EXE) de la version *NNNL*;
 - le fichier PACBASE.DAT, qui contient tous les paramètres d'exécution de la Station de Travail;
 - le fichier OK.VER, témoin d'installation de la version;
 - les fichiers de paramétrage du Gestionnaire de Communications, GSxxxxxx.PRM et GSxxxxxx.TAB (à ce sujet, cf. le manuel *Gestionnaire de Communications et Utilitaire PACLINK*);
 - le fichier CGIPAC.FON, qui contient la police utilisée par le Gestionnaire de Communications en mode TRACE (le mode TRACE est documenté dans le manuel *Gestionnaire de Communications et Utilitaire PACLINK*, Chapitre *Le Gestionnaire de Communications*, sous-chapitre *Description de la fenêtre du Gestionnaire de Communications*);

- \SPAC\NNNL\UTIL : ce répertoire contient :
 - des utilitaires de lancement direct de la Station de Travail;
 - le fichier NEWSL.TXT (*L* correspond à la langue de la méthode) affichable par n'importe quel éditeur de texte, qui contient un texte sur les nouveautés de la version de la Station.
- \SPAC\METL\MMM : ce répertoire contient tous les paramètres décrivant la méthode *MMM* en langue *L*. Ces fichiers de paramètres sont de 8 types, chacun d'eux se trouvant dans un sous-répertoire propre :
 - ENT : 1 fichier .ENT par entité gérée dans la méthode;
 - FBX : 1 fichier .FBX par boîte *A partir de...*;
 - GPH : 1 fichier .GPH par descriptif de graphe;
 - HLP : fichiers d'aide de la Station de Travail;
 - ICN : 1 fichier .ICN par icône appellable dans les graphes;
 - LIS :
 - . 2 fichiers .LIS par module (liste des entités et liste des métamodèles utilisés par le module);
 - . 3 fichiers supplémentaires : RELATION.LIS, EXTMQ.LIS et MAQLO.LIS;
 - LOG : fichiers .LOG pour l'interface avec le serveur;
 - MTM : 1 fichier .MTM par sous-schéma du métamodèle.
- \SPAC\BASE : ce répertoire contient :
 - le fichier BASE.LST, contenant la liste des bases qui seront proposées à l'utilisateur lors de la connexion à la Station de Travail;
 - les scripts de connexion (.SCI) et de déconnexion (.SCF)¹ permettant d'accéder aux bases présentes sur le serveur ou de les quitter. Remarque : le contenu de ces fichiers dépend des paramètres d'exploitation de votre ou vos site(s) serveur(s). Vous devez donc mettre à jour leur contenu selon ces paramètres. Pour plus d'informations, consultez le Chapitre *Personnalisation de la Station de Travail*, Sous-chapitre *Scripts de connexion et de déconnexion*;
 - des fichiers de paramètres .PRO qui mémorisent les préférences de l'utilisateur. En général, ces fichiers ne doivent être ni modifiés, ni supprimés.

Le répertoire BASE sert également de racine à l'ensemble des répertoires de données des utilisateurs, qui se constitueront au fur et à mesure de leur travail avec la Station de Travail.

¹ Extension non affichée sous Windows NT

1.3.1 Variantes d'installation

Vous pouvez répartir les composants de la Station de Travail en fonction de l'architecture matérielle dont vous disposez, et de l'organisation que vous souhaitez mettre en place. Les types de composants de la Station et leur localisation dans les répertoires sont les suivants :

- Les fichiers exécutables des applications, dans le répertoire *NNNL* (où *NNN* correspond au numéro de version et *L* à la langue);
- Les fichiers méthodologiques, dans le répertoire *METL\MMM* et ses sous-répertoires (où *MMM* correspond à la méthode et *L* à la langue);
- Les fichiers utilisateur, dans le répertoire *BASE* et ses sous-répertoires.

Vous pouvez installer la Station de Travail sur micro seulement, en réseau ou de façon mixte (une partie sur micro et une partie sur le réseau). L'installation de ces répertoires peut se faire selon les configurations suivantes (il s'agit de quelques exemples, la liste n'est pas exhaustive) :

- Installation de tous les répertoires sur le même disque dur, disque du micro ou du serveur de réseau local. Avec cette configuration :
 - sur micro, toutes les Stations de Travail sont indépendantes;
 - sur le réseau, il y aura une Station de Travail unique partagée par tous les utilisateurs. Les micros qui y accéderont doivent tous avoir le même système de communication avec le serveur. Attention, des problèmes d'accès concurrents peuvent apparaître, puisque les fichiers utilisateur (répertoire *BASE* et ses sous-répertoires) sont communs à tous. Les éventuels conflits d'accès sont gérés par le gestionnaire de réseau local.
- Installation de la partie "non-utilisateur" (qui peut au besoin être protégée par le système) sur le réseau, et de la partie "utilisateur" (répertoire *BASE* et ses sous-répertoires) sur le micro.
Avec ce type de configuration, les données utilisateur sont indépendantes, et les problèmes d'accès concurrents sont donc réduits. Dans ce cas également, les micros doivent tous avoir le même système de communication avec le serveur.
- Installation de la partie "non-utilisateur" sur le réseau, et de la partie "utilisateur" plus quelques exécutables spécifiques sur le micro. Cette configuration permet de stocker sur un disque les exécutables communs à toutes les Stations de Travail et sur un autre ceux spécifiques à chaque Station de Travail. On peut ainsi gérer des différences entre les Stations de Travail (les systèmes de communication par exemple).

On appelle :

- "disque des exécutables" le disque qui contient les exécutables spécifiques à une Station de Travail;
- "disque des exécutables par défaut" le disque qui contient la partie commune à toutes les Stations de Travail.

Avec une telle configuration, le premier disque accédé est celui des exécutables "non défaut". Si certains des exécutables sont introuvables, le disque par défaut est alors accédé.

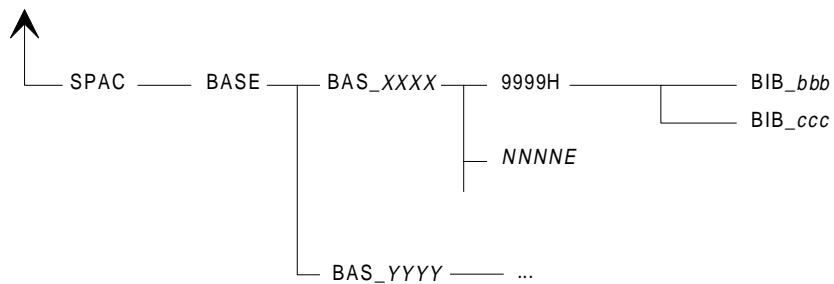
PEXEC.EXE et CONNX.EXE, qui servent au démarrage de la Station de Travail, doivent rester sur l'unité "non défaut" ainsi que le fichier PACBASE.DAT.

Selon la configuration choisie, il faut modifier les paramètres disque EXE, disque EXE (défaut), disque système et disque données dans le fichier PACBASE.DAT. Les fichiers correspondants doivent être copiés ou déplacés manuellement, en conformité avec les valeurs modifiées de ces paramètres.

Pour plus d'informations sur le fichier PACBASE.DAT, consulter le paragraphe correspondant dans le chapitre *Lancement de la Station de Travail*.

1.3.2 Evolution de l'arborescence : Réseau utilisateur

En cours d'utilisation de la Station de Travail, la partie de l'arborescence située sous le répertoire \SPAC\BASE s'enrichit automatiquement en fonction des bases, des sessions et des bibliothèques auxquelles l'utilisateur accède.



Le réseau utilisateur, utilisant le répertoire \SPAC\BASE comme racine, se développe de la manière suivante :

- Pour chaque nouvelle base à laquelle l'utilisateur se connecte, un sous-répertoire est créé sous \SPAC\BASE, intitulé BAS_XXXX, où XXXX est le code de la base de connexion. Par exemple, si l'utilisateur accède à la base D280, un sous-répertoire BAS_D280 est automatiquement créé s'il n'existe pas déjà.

Théoriquement, la liste des bases accessibles, répertoriées dans le fichier \SPAC\BASE\BASE.LST, doit être en phase avec la liste des répertoires BAS_XXXX. Si l'utilisateur élimine une ou plusieurs lignes du fichier BASE.LST, il est conseillé de détruire les répertoires correspondants, devenus inutiles. ATTENTION : cette opération ne doit être effectuée qu'une fois que les données utilisateur contenues dans ces répertoires ont été remontées sur le serveur, s'il y a lieu.

- Lorsque l'utilisateur se connecte à une base, il choisit de s'y connecter en une session et une bibliothèque données.
 - Session choisie :
 - Si l'utilisateur se connecte en session courante, un sous-répertoire 9999H est ajouté sous BAS_XXXX;
 - Si l'utilisateur se connecte en session historisée, un sous-répertoire NNNNE est alors ajouté sous BAS_XXXX :
 - NNNN est le numéro de la session historisée;
 - E prend la valeur H ou T (type d'historisation).
 - Bibliothèque choisie : Dans une session donnée, pour chaque bibliothèque à laquelle l'utilisateur accède, un sous-répertoire BIB_BBB est créé (BBB est le code de la bibliothèque d'accès). Ce sous-répertoire est placé sous le répertoire 9999H ou NNNNE correspondant à la session de connexion. Les répertoires BIB_BBB contiennent toutes les descriptions des occurrences d'entités descendues du serveur en local pour consultation ou modification, ainsi que tous les brouillons créés sous la Station de Travail, à savoir :
 - un fichier EEEDz.LST par type de description d'une entité gérée localement, contenant la liste des occurrences présentes sur le micro dans ce répertoire. EEE représente le code local de l'entité et Dz le code local du type de description (exemple : TXTD1.LST, où TXT est le code local de l'entité Texte, et D1 le code local du type de description "contenu du texte" de cette entité). C'est ce fichier qui permet de mettre à jour les listes locales proposées par la Station de Travail (voir le *Manuel de Référence* au sujet des listes locales).
 - un fichier OOOOODz.EEE par occurrence d'entité présente en local. Ce fichier contient la description de type Dz de l'occurrence OOOOOO de l'entité EEE (exemple : PTMA00D1.TXT contient la description locale de type D1 de l'occurrence PTMA00 de l'entité Texte, dont le code local est TXT).

La Station de Travail ne détruit pas automatiquement les répertoires correspondant à des sessions et/ou bibliothèques qui ne sont plus utilisées. L'utilisateur doit pratiquer de temps en temps une purge de son arborescence, pour éviter la prolifération de répertoires inutiles. ATTENTION : les mêmes précautions doivent être prises qu'en cas de suppression d'un répertoire BAS_XXXX (décrite plus haut).

1.4 Installation de la partie micro : Généralités

1.4.1 Support d'installation

La partie micro-ordinateur de la Station de Travail est constituée d'un CD ROM contenant des fichiers compactés. Ces fichiers seront automatiquement décompactés par la procédure d'installation.

1.4.2 Environnement d'installation

L'installation de la partie micro de la Station de Travail peut s'effectuer sur divers environnements, selon l'environnement d'exécution désiré :

Systeme d'exploitation (exécution)	Interface Graphique Utilisateur (GUI)	Installation sous Windows via Gestionnaire de Programmes
DOS	Windows 3.1 ou 3.11 Windows 95 Windows for Workgroups	SETUP.EXE
Windows NT		
OS/2 2.1 WARP	Win-OS/2 (similaire à Windows 3.1)	

1.5 Installation de la partie micro : versions DOS Windows et Win-OS/2

1.5.1 Lancement de l'installation

L'installation peut être effectuée directement à partir du CD-ROM d'installation ou à partir d'un répertoire du réseau où le contenu du CD-ROM aura préalablement été copié.

Pour lancer l'installation de la Station et / ou d'une méthode :

- insérez le CD ROM dans le lecteur et attendez quelques secondes. La première fenêtre qui s'affiche alors vous demande de sélectionner la langue d'installation, puis une deuxième fenêtre vous demande de sélectionner le produit que vous voulez installer. Sélectionnez Station de Travail (Pacdesign, Pacbench, Méthodes).
- ou dans le Gestionnaire de fichiers, positionnez-vous sur l'unité du CD-ROM, qui contient les fichiers d'installation, et sélectionnez le sous-répertoire *pdnnn* (*nnn* représente le numéro de version de la Station), puis :
 - ♦ double-cliquez sur le fichier SETUP.EXE. Dans ce cas, une fenêtre s'ouvre, vous demandant d'indiquer la langue de l'installation;
 - ♦ ou sélectionnez le sous-répertoire *fra* (pour français) ou *eng* (pour anglais) puis double-cliquez sur le fichier SETUP.EXE.



Vous pouvez arrêter l'installation de la Station à tout moment en cliquant sur l'icône "Exit" qui se trouve en bas à droite de votre écran, en appuyant sur la touche fonction F3 ou en appuyant sur le bouton Annuler se trouvant au bas de chaque fenêtre.

1.5.2 Sélection de la Station et/ou des méthodes à installer

La fenêtre suivante affiche une liste comprenant la Station de Travail et différentes méthodes. Vous sélectionnez une ou plusieurs lignes de cette liste, selon vos besoins.

Si vous sélectionnez la Station et au moins une méthode, l'installation de la Station sera automatiquement suivie de celle de la ou des méthode(s). Si vous sélectionnez plusieurs méthodes, celles-ci seront installées les unes après les autres, dans l'ordre de la liste.

Si vous sélectionnez l'installation d'une "autre méthode", non citée dans la liste, vous devez disposer de la disquette d'installation.



Vous ne pouvez installer de méthode que si la Station est installée.

Pour sélectionner une ligne de la liste, cliquez dans la partie située à gauche de la liste avec le bouton gauche de la souris. Pour la désélectionner, cliquez sur la marque de sélection pour la faire disparaître.

Pour installer la Station et toutes les méthodes proposées, cliquez sur le bouton "Tout sélectionner".

Le bouton "Effacer tout" permet de décocher toutes les lignes sélectionnées.

Les évolutions de la version installée apparaissent dans la fenêtre suivante. Elles sont stockées :

- dans le fichier *NEWSF.TXT* (français) ou *NEWSA.TXT* (anglais) du répertoire *UTIL* pour les nouveautés de la Station,
- dans le fichier *metF.TXT* ou *metA.TXT* ('met' représentant la méthode), directement sous la racine du répertoire de la méthode pour les nouveautés d'une méthode.

De plus, vous pourrez créer, en fin d'installation, un icône "Nouveautés" dans le groupe Station de Travail/Méthode pour permettre à l'utilisateur de consulter à tout moment les évolutions de la version installée.

1.5.3 Installation d'une nouvelle Station

Les deux premiers choix de cette fenêtre sont importants si vous voulez installer une Station alors qu'il existe déjà une version de celle-ci sur le poste.

Si vous choisissez "Installation nouvelle", l'ancienne installation sera conservée, si elle ne se trouve pas sur le même répertoire que la nouvelle.



Si vous installez la Station au même emplacement, la nouvelle installation écrasera l'ancienne, sans message d'avertissement.

Si vous choisissez "Remplacement", vous devrez indiquer le chemin complet des exécutables de la Station que vous voulez remplacer. Vous pourrez alors modifier l'emplacement de certains fichiers si vous le désirez.

Vous pouvez utiliser le choix "Utiliser la sauvegarde" si les paramètres d'installation de la Station ont été sauvegardés lors d'une installation précédente (voir paragraphe *Récapitulation des paramètres*). Ce choix vous permet de récupérer ces paramètres en indiquant l'unité et le chemin complet d'accès au fichier de sauvegarde *SPACSAVE.PRM*. Tous les paramètres seront alors renseignés à l'identique, sans que vous ayez à les saisir de nouveau.

1.5.3.1 Chemin des exécutables, données méthodologiques et utilisateur

Par défaut, l'utilitaire d'installation vous propose d'installer tous les fichiers sur le disque C. Mais vous pouvez modifier un, deux ou les trois chemins d'accès en sélectionnant les lignes correspondantes (clic avec le bouton gauche de la souris dans la case du début de la ligne).

Si vous appuyez sur Suivant, des fenêtres de sélection vous permettent d'indiquer les nouveaux chemins d'accès.



Les chemins d'accès indiqués sont automatiquement complétés par le répertoire de la Station *\SPAC* ; vous ne devez donc pas l'inclure dans les chemins d'accès.

1.5.3.2 Paramètres de communication

Vous devez ensuite indiquer les paramètres de communication, c'est à dire :

- Type de serveur : serveur avec lequel la Station de Travail doit fonctionner (par exemple Mainframe);
- Protocole de communication sous lequel ce serveur fonctionne (par exemple IBM 3270 ou BULL DKU71xx);
- Variante du protocole de communication : vous devez choisir dans la liste en fonction du mode serveur. (Pour plus d'informations sur les types de communications, consultez le manuel *Gestionnaire de Communications et Utilitaire PACLINK*);
- Mode de connexion : ce paramètre permet de mettre en place les modes de connexion qui seront autorisés pour accéder à la Station de Travail.
Les deux possibilités sont les suivantes :
 - connexion locale et serveur;
 - connexion serveur forcée.

L'autorisation maximale "connexion locale et serveur" permet à tout utilisateur de la Station de Travail soit de se connecter localement et travailler sur les données qui ont déjà été descendues du serveur sur son poste, soit de se connecter au serveur. L'autorisation "connexion serveur forcée" impose de se connecter au serveur. Dans ce cas, les fichiers de reconnaissance des utilisateurs ne sont pas créés sur le micro.

1.5.3.3 Récapitulation des paramètres

La fenêtre suivante récapitule les paramètres indiqués pour l'installation.

Si vous voulez les conserver pour d'autres installations, cliquez sur "Sauvegarder". Ainsi, lors d'une prochaine installation, tous les paramètres seront automatiquement renseignés à l'identique, sans que vous ayez à les saisir à nouveau.

Quand vous cliquez sur "Sauvegarder", une fenêtre de sélection vous permet d'indiquer l'unité et le chemin d'accès complet au fichier de sauvegarde de la Station (SPACSAVE.PRM). Si le support de lancement n'est pas le CD-ROM, le répertoire de sauvegarde proposé est celui du programme SETUP.EXE.

Si vous voulez modifier certains paramètres, cliquez sur "Modifier". En faisant défiler l'ensemble des fenêtres de la procédure d'installation, vous pourrez modifier les informations erronées.

Si les paramètres vous satisfont, cliquez sur "Installer" pour démarrer l'installation.

Si l'installation s'est bien passée un message vous indique que l'installation est correcte.

1.5.3.4 Création des icônes

A la fin de l'installation, une fenêtre s'ouvre proposant la liste des dossiers du Gestionnaire de Programme déjà installés. Vous devez y sélectionner un dossier Windows où insérer les icônes de la Station ou saisir le nom d'un nouveau dossier ("Station VisualAge Pacbase" par défaut).

Un message vous demande ensuite si vous voulez installer le groupe correspondant.

Si vous choisissez "Oui", vous devez choisir les icônes que vous voulez voir apparaître dans ce groupe :

- Nouveautés: fichier texte contenant les nouveautés de la version installée
- PACBASE.DAT: fichier contenant les paramètres nécessaires au fonctionnement de la Station. Voir le chapitre *Lancement de la Station de Travail*.
- Station de Travail : icône de lancement de la Station
- Arrêt de la Station : icône de fermeture de la Station
- PACLINK : icône de lancement de l'émulateur de terminal. Pour des détails sur cet utilitaire consultez le manuel *Gestionnaire de Communications et Utilitaire PACLINK*.

1.5.4 Installation des méthodes

L'installation d'une méthode est un prérequis au fonctionnement de la Station.

Pour qu'une méthode puisse être installée, la Station de Travail doit déjà être installée.

1.5.4.1 Choix de la méthode à installer

Vous pouvez demander l'installation d'une méthode en même temps que celle de la Station après que celle-ci soit installée, de manière indépendante. Ceci vous permet d'ajouter des méthodes ou d'en changer facilement.



Reportez-vous au paragraphe *Sélection de la Station et/ou des méthodes à installer* pour plus de renseignements.

1.5.4.2 Chemin de la Station de Travail

Vous devez indiquer le chemin complet d'accès (incluant \SPAC\NNNL) au fichier PEXEC.EXE.

Si vous n'indiquez qu'un chemin partiel, l'installation s'achève sur un message d'erreur.

Si les paramètres d'installation de la méthode ont été sauvegardés lors d'une installation précédente (voir paragraphe *Récapitulation des paramètres*), vous pouvez les récupérer en indiquant l'unité et le chemin complet d'accès au fichier de sauvegarde METHSAVE.PRM. Tous les paramètres seront alors renseignés à l'identique, sans que vous ayez à les saisir de nouveau.

1.5.4.3 Chemin de la méthode

En fonction du chemin de la Station de Travail indiqué dans la fenêtre précédente, l'utilitaire d'installation vous propose par défaut un chemin d'accès à la méthode, que vous pouvez modifier.

Si la méthode n'existe pas dans le chemin choisi, la fenêtre suivante précise qu'il s'agit d'une "nouvelle installation".

Si la méthode est déjà installée dans le chemin choisi, la fenêtre suivante affiche le chemin d'accès à la précédente installation de la méthode. Si vous souhaitez remplacer une méthode installée à un autre emplacement, vous devez indiquer le chemin complet d'accès à l'ancienne méthode.

1.5.4.4 Récapitulation des paramètres

La fenêtre suivante récapitule les paramètres indiqués pour l'installation.

Si vous voulez les conserver pour d'autres installations, cliquez sur "Sauvegarder". Ainsi, lors d'une prochaine installation, tous les paramètres seront automatiquement renseignés à l'identique, sans que vous ayez à les saisir à nouveau.

Quand vous cliquez sur "Sauvegarder", une fenêtre de sélection vous permet d'indiquer l'unité et le chemin d'accès complet au fichier de sauvegarde de la méthode METHSAVE.PRM. Si le support de lancement n'est pas le CD-ROM, le répertoire de sauvegarde proposé est celui du programme SETUP.EXE.



Si la méthode que vous installez existe déjà dans le répertoire indiqué, un message prévient que le contenu du répertoire sera détruit et remplacé par la nouvelle installation.

Si vous voulez modifier certains paramètres, cliquez sur "Modifier". En faisant défiler l'ensemble des fenêtres de la procédure d'installation, vous pourrez modifier les informations erronées.

Si les paramètres vous satisfont, cliquez sur "Installer" pour démarrer l'installation.

Si l'installation s'est bien passée un message vous indique que l'installation est correcte.

A la fin de l'installation, un message vous demande de sélectionner les modules que vous voulez installer : Pacdesign et/ou Pacbench.

1.5.4.5 Création des icônes

Lorsque l'installation est terminée, vous pouvez choisir d'installer un groupe Windows et les icônes de la méthode. Une fenêtre s'ouvre proposant la liste des dossiers du Gestionnaire de Programme déjà installés. Vous devez y sélectionner un dossier où insérer les icônes de la méthode ou saisir le nom d'un nouveau dossier.

Un message vous demande ensuite si vous voulez installer le groupe correspondant.



Si vous avez créé un groupe personnalisé lors de l'installation de la Station, celui-ci apparaîtra dans la liste des groupes proposés pour la méthode.

Si vous choisissez "Oui", vous devez choisir les icônes que vous voulez voir apparaître dans le groupe :

Nouveautés: fichier texte contenant les nouveautés de la version installée

PACBASE.DAT: fichier contenant les paramètres nécessaires au fonctionnement de la Station. Voir le chapitre *Lancement de la Station de Travail*.

Si l'installation s'est bien passée un message vous indique que l'installation est correcte.

1.6 Paramétrage micro : version DOS Windows 3.x

- Fichier CONFIG.SYS : De manière générale, il faut éviter de surcharger le fichier CONFIG.SYS. En effet, chaque ordre entraîne le chargement d'un programme résident, ce qui diminue d'autant la mémoire centrale disponible. Toutefois, un bon réglage du micro-ordinateur, au niveau de ce fichier, permet d'améliorer les performances de la Station de Travail. Voici quelques valeurs recommandées pour y parvenir :
 - SHELL=C:COMMAND.COM /E:512 /P
 - FILES=20 et BUFFERS=10 suffisent au fonctionnement de la Station de Travail;
 - SmartDrive, dont l'installation est automatique lors de l'installation de Windows 3.x, doit entièrement être configuré en Extended, d'où l'absence de fichier EMM.SYS.

SmartDrive est plus rapide qu'un disque virtuel et surtout plus "intelligent" : il gère son espace mémoire en tenant compte des fréquences d'utilisation des fichiers qu'il contient. Avec ce système, le premier accès à un fichier (exécutable ou données) est fait sur le disque dur (le témoin clignote). Le fichier est ensuite chargé dans le cache disque et les accès ultérieurs n'entraînent plus d'accès disque (relativement lents).

- Fichier AUTOEXEC.BAT : quelques réglages de la Station de Travail s'effectuent en déclarant de nouveaux paramètres d'environnement dans le fichier AUTOEXEC.BAT :
 - SET TEMP=X:\PACTMP permet d'indiquer qu'un répertoire PACTMP (à créer s'il n'existe pas déjà) doit contenir l'ensemble des fichiers temporaires générés par la Station de Travail. X: désigne une partition quelconque du disque dur;
 - Pour éviter les conflits d'accès, l'option SHARE doit être active. Il est conseillé de l'intégrer à l'AUTOEXEC.BAT. Vous pouvez aussi l'intégrer au fichier CONFIG.SYS (reportez-vous à la documentation DOS).
ATTENTION : la Station de Travail, lors de son lancement, ne contrôle pas que l'option SHARE est active;
 - Si le caractère '\$' n'est pas disponible sur le clavier, set DOLLAR=NNN (où NNN est la valeur ANSI du caractère désiré). La Station de Travail et le serveur utilisent le caractère '\$' pour des opérations telles que : caractère de contrôle dans les descriptions de Texte, caractère d'appel de la liste d'occurrences serveur d'entités utilisateur.

1.7 Paramétrage micro : version DOS Windows 95 et NT

Comme pour toutes les applications Windows, l'installation crée un groupe de programmes spécifique à la Station. Vous pouvez utiliser les raccourcis créés à cette occasion pour lancer la Station (menu *Démarrer*, choix *Programmes*).

1.8 Paramétrage micro : version OS/2 WIN-OS/2 (à partir de 2.1)

- Fichier CONFIG.SYS : ajouter un paramètre d'environnement dans le fichier CONFIG.SYS :
 - SET TEMP=X:\PACTMP permet d'indiquer qu'un répertoire PACTMP (à créer s'il n'existe pas déjà) doit contenir l'ensemble des fichiers temporaires générés par la Station de Travail. X désigne une partition quelconque du disque dur.

REMARQUE :

En OS/2 2.1, l'interface WIN-OS/2 se présente de façon identique à Windows 3.1.

2 *Personnalisation de la Station de Travail*

2.1 **Scripts de connexion et de déconnexion**

2.1.1 **Modes de connexion**

La Station de Travail peut être utilisée de deux façons :

- en liaison avec le serveur (mode serveur),
- de façon autonome (mode local), c'est le mode par défaut.



Certains sites travaillent en "mode serveur forcé" et n'ont pas la possibilité d'utiliser le mode local. C'est une option à positionner à l'installation.

Dans le cas général, et mise à part la première connexion, l'utilisateur peut se connecter en "mode local" puis passer en "mode serveur" (consultez le *Manuel de Référence* pour de plus amples détails). Au moment de la connexion en "mode serveur" (au lancement de la Station de Travail ou plus tard), une procédure de connexion au serveur est lancée automatiquement.

Pour que l'accès en "mode local" soit possible, il faut qu'une première connexion en "mode serveur" ait été réalisée antérieurement. Les codes des utilisateurs et les mots de passe sont alors connus au niveau local et peuvent être contrôlés sans connexion au serveur.

2.1.2 Présentation

Par le biais des gestionnaires de communication, la Station de Travail assure une connexion automatique entre le micro-ordinateur et le serveur, pour l'une des bases répertoriées. La boîte de connexion est représentée ci-dessous.

The screenshot shows a graphical user interface for connecting to a VisualAge Pacbase workstation. The window is titled "Station de travail VisualAge Pacbase". It features several input fields and controls:

- Utilisateur:** A "Code" field containing "PDCFI" and a "Mot de passe" field containing "*****". There is a "Modification" checkbox and a "Nouveau mot de passe" field.
- Bases:** A list of databases with columns for name, description, and file name. The entries are:

BABY	VisualAge pour PACBASE 1.6	[PACBASE.DAT]
D100	Base Test pour Station 2.0 Mode Dossier	
D7ME	Base Test pour STATION	
D9ME	Base Test pour PACBASE etc...	[PACCICS.DAT]
- Bibliothèque:** A dropdown menu showing "T3A" and "variante microfocus anglais".
- Session:** A dropdown menu.
- Site de connexion:** Radio buttons for "Local" and "Serveur" (which is selected).
- Contrôle DSMS:** Fields for "Code produit" and "Numéro amélioration".
- Buttons:** "OK" and "Quitter" buttons at the bottom.

A label "Liste des bases répertoriées" points to the "Bases" list.

Pour qu'un nom de base apparaisse dans la boîte de connexion, il faut que cette base soit répertoriée dans le fichier \SPAC\BASE\BASE.LST.

Pour que l'accès serveur à une base répertoriée puisse aboutir, vous devez créer un script de connexion à cette base. Un script de déconnexion doit également exister pour pouvoir quitter la base, après utilisation.

2.1.3 Répertoire une base

Chaque base accessible doit être répertoriée dans le fichier \SPAC\BASE\BASE.LST sous forme d'une ligne, qui doit être formatée de la manière suivante :

- Code de la base : un à quatre caractères (selon les plates-formes), ne comprenant ni blanc, ni point, ni astérisque. Ce code doit commencer en première colonne du fichier;
- Libellé de la base : à partir de la cinquième colonne du fichier, vous pouvez saisir un libellé, qui indiquera à l'utilisateur de quelle base il s'agit. Pour répertorier une nouvelle base il suffit d'ajouter, à l'aide d'un éditeur de texte, une telle ligne dans le fichier \SPAC\BASE\BASE.LST.

Si vous supprimez une ligne du fichier, la base correspondante ne sera plus proposée dans la boîte de connexion.

Exemple : pour obtenir la boîte de connexion représentée ci-dessus, il a fallu saisir dans le fichier \SPAC\BASE\BASE.LST les informations suivantes :

```
BABY VisualAge pour Pacbase 1.6 [PACBASE.DAT]
D100 Base Test pour Station 2.0 Mode Dossier
D7ME Base Test pour STATION
D9ME Base Test pour PACBASE etc. [PACCICS.DAT]
```

2.1.4 Scripts de connexion

Les procédures de connexion au serveur sont très variables d'un site à un autre, en particulier lorsqu'un même site dispose de plusieurs systèmes différents, ou de logiciels gérant la confidentialité et la protection des informations.

Un langage interprété permet de créer les scripts de connexion et de déconnexion. Ce langage est décrit en annexe du présent Manuel.

Pour chaque base *BBBB* répertoriée dans le fichier \SPAC\BASE\BASE.LST, il doit exister dans le répertoire \SPAC\BASE deux fichiers scripts associés :

- Un script de connexion, dont le nom est *BBBB.SCI*. Ce script doit contenir les instructions qui établissent la connexion au moniteur TP (CICS, IMS, TDS, ...), et qui envoient le code transaction VisualAge Pacbase permettant d'aboutir à la grille de connexion, qui sera remplie par la Station de Travail;
- Un script de déconnexion, dont le nom est *BBBB.SCF*¹. Ce script doit contenir les instructions permettant de quitter la transaction VisualAge Pacbase et éventuellement le moniteur TP.

¹ Extension non affichée sous Windows NT

Exemple : à une base D280 répertoriée correspondra un script de connexion D280.SCI et un script de déconnexion D280.SCF.

Des exemples de scripts sont fournis dans le répertoire \SPAC\BASE (fichiers FX_...SCI, et FX_...SCF pour des exemples en français et fichiers AX_...SCI, et AX_...SCF pour des exemples en anglais). D'autres font l'objet de paragraphes spécifiques du présent Manuel.

REMARQUE : Si vous êtes connecté via Top Secret, la connexion et la déconnexion s'effectuent sans utiliser les fichiers "Scripts" .SCI et .SCF.

2.1.5 Conseils pratiques

Il peut s'avérer utile de prévoir, au début du script de connexion, une première phase de déconnexion, de sorte que l'utilisateur puisse se connecter à la Station de Travail quel que soit l'état initial de son terminal (connecté ou non au serveur).

Il est très important de tenir compte des variations possibles des temps de réponse, pour déterminer les valeurs de temporisation. Il est donc vivement conseillé de multiplier les tests, en particulier lorsque les temps de réponse sont plus longs qu'habituellement.

2.1.6 Cas particulier : Connexion à VisualAge Pacbase (serveurs OS/2, Windows NT ou UNIX)

La connexion et la déconnexion "réelles" sont assurées, via le serveur OS/2, Windows NT ou le serveur UNIX, par des assignations de fichiers. Pour de plus amples détails sur ces fichiers, consultez le manuel *Gestionnaire de Communications et Utilitaire PACLINK*.

Les fichiers de scripts de connexion et de déconnexion doivent exister, car la Station de Travail vérifie leur présence et teste leur validité. Cependant, comme les opérations de connexion/déconnexion ne dépendent pas de ces scripts, ils sont réduits à leur plus simple expression (voir les exemples du paragraphe "Exemples de Scripts").

2.1.7 Modes d'utilisation des scripts

Une base étant répertoriée et les scripts associés créés, l'accès à la base par l'intermédiaire de la boîte de connexion peut se faire de deux manières distinctes :

- En mode test : la majeure partie des instructions contenues dans les scripts s'affichent à l'écran au fur et à mesure de leur utilisation, permettant de déceler d'éventuelles erreurs;
- En mode exploitation : les scripts, jugés corrects après une phase de test, sont lancés sans que les instructions en cours défilent à l'écran.

Pour changer de mode, il faut, à l'aide de n'importe quel éditeur de texte, éditer le fichier \SPAC\NNN\L\PACBASE.DAT (où *NNN* correspond au numéro de version installée et *L* à la langue) et en modifier la ligne 011, intitulée Mode d'exécution connexion, de la façon suivante :

- 011 Mode d'exécution connexion [T] pour le mode test;
- 011 Mode d'exécution connexion [E] pour le mode exploitation.

2.1.8 Mode test

Lorsque l'utilisateur se connecte à la Station de Travail ou s'en déconnecte, le script associé s'exécute, et s'affiche dans une boîte de dialogue de la manière suivante :

- Le message "connexion en cours" est affiché pendant toute l'exécution du script;
- Sous ce message apparaît la phase du traitement, qui correspond au libellé associé à une commande OUTPUT dans le script;
- Toutes les commandes du script défilent dans la boîte située sous les deux messages précédents. Les commandes exécutées sont visualisées, mais les structures (IF, WHILE, FOR etc.) ne le sont pas;
- A la fin de l'exécution du script et si une erreur a été détectée, le message "le script n'a pas abouti" est affiché. Si tout s'est normalement déroulé, un bouton SUITE... permet d'enchaîner et de démarrer la Station de Travail.

2.1.9 Mode exploitation

Une fois les scripts bien rodés, leur exécution peut avoir lieu en mode exploitation, de la manière suivante :

- Le message "connexion en cours" est affiché pendant toute l'exécution du script;
- Sous ce message apparaît la phase du traitement, qui correspond au libellé associé à une commande OUTPUT dans le script;
- A la fin de l'exécution du script de connexion, si tout s'est déroulé normalement, la fenêtre du Gestionnaire de la Station de Travail est lancée automatiquement. Si une erreur utilisateur ou système est décelée, l'exécution du script est interrompue et la boîte de connexion est réaffichée. Si une erreur est décelée au niveau des contrôles de droit d'accès au serveur, l'utilisateur peut les modifier dans la boîte de connexion et la connexion est reprise au niveau de ces contrôles.

2.1.10 Affichage d'écrans serveur

Il est possible d'incorporer aux scripts des ordres DISPLAY, qui ouvrent une fenêtre affichant l'écran du serveur en mode 25x80, permettant par exemple un meilleur diagnostic en cas d'erreur d'exécution du script.

Suite à un ordre DISPLAY, l'exécution du script est suspendue jusqu'à la fermeture de la fenêtre associée.

ATTENTION : si dans un script vous écrivez un ordre WRITE (écriture de données en direction du serveur) suivi d'un ordre DISPLAY, ce dernier efface les données écrites par le WRITE. Pour ne pas perdre les données écrites, l'ordre DISPLAY doit être placé *après* l'ordre SEND qui les envoie au serveur.

Il est conseillé, lors du basculement du mode test en mode exploitation, de modifier les scripts afin d'éliminer tous les DISPLAY de test, hormis ceux devant être activés suite aux erreurs utilisateur ou système (moniteur transactionnel (CICS, TDS...) inactif, serveur inactif, etc.).

2.1.11 Exemples de scripts IBM-CICS

- Connexion : Obtention de la grille de connexion (grille Saint-Marc), effacement de l'écran, saisie du code CICS sur lequel on désire travailler. Attente de la mire Welcome; saisie du code utilisateur et du mot de passe dans la boîte de dialogue script, effacement de l'écran. Envoi du code transaction sign-on CICS (CESN), alimentation des valeurs saisies dans la grille de transaction CICS, attente de la mire ***** et connexion à VisualAge Pacbase. Ce script est particulier à un site défini, et doit bien entendu être adapté pour chaque site.

```

prog V280SCI
integer cr ln cl
integer cpt

string USER PASS NEWPASS

begin

OUTPUT ("Obtention grille Saint-Marc", 12)
ln = 0
cl = 0
SEND ("REST")
cr = SEARCH (ln, cl, "VECQUERIE", 9, 2)
if (cr <> 1)
  begin
    SEND ("CLS")
    WRITEC ("CESF", 0)
    SEND ("ENT")
    ln = 0
    cl = 0
    cr = SEARCH (ln, cl, "VECQUERIE", 9, 3)
    if (cr <> 1)
      begin
        cr = SEARCH (ln, cl, "INCONNU", 7, 2)
        if (cr <> 1)
          begin
            ERROR ("PROBLEME SITE CENTRAL", "Vous êtes sûrement sous TSO", 1)
            DISPLAY ()
            EXIT (12)
          end
        end
      end
    end
  end

OUTPUT ("Connexion à CICS", 12)
SEND ("REST")
SEND ("CLS")
WRITEC ("CICST", 0)
SEND ("ENT")
ln = 0
cl = 0
cr = SEARCH (ln, cl, "WELCOME", 7, 25)
if (cr <> 1)
  begin
    ERROR ("PROBLEME SITE CENTRAL", "Cics Test non lancé", 1)
    DISPLAY ()
    EXIT (12)
  end
end

```



```

cr = 0
cpt = 1
while (cr <> 1)
  begin
    if (cpt > 3)
      begin
        ERROR ("Nombre d'essais limité à 3", "Signon invalide", 1)
        EXIT (15)
      end
    OUTPUT ("Signon CICS", 12)
    INPUT ("Identification système", "Utilisateur CICS", USER, 8, 1, "Mot de passe CICS",
    PASS, 8, 0, "Nouveau mot de passe", NEWPASS, 8, 0)
    SEND ("CLS")
    if ( USER == "ex" or USER == "EX" )
      begin
        OUTPUT("Déconnexion de CICST", 12)
        WRITEC("CESF", 0)
        SEND("ENT")
        EXIT(16)
      end
    WRITEC ("CESN", 2)
    SEND ("ENT")
    ln = 0
    cl = 0
    cr = SEARCH (ln, cl, "PASSWORD", 8, 12)
    if (cr <> 1)
      begin
        ERROR ("Problème CICS", "Ecran CESN non obtenu", 0)
        EXIT (15)
      end
    WRITE (5, 25, USER, 1)
    WRITE (7, 25, PASS, 3)
    if (NEWPASS <> " ")
      begin
        WRITE (7, 63, NEWPASS, 3)
        WRITE (9, 63, NEWPASS, 3)
      end
    SEND ("ENT")
    ln = 0
    cl = 0
    cr = SEARCH (ln, cl, "EXPIRED", 7, 3)
    if (cr == 1)
      begin
        ERROR ("Changement de mot de passe", "Entrez votre nouveau mot de passe", 0)
        cr = 0
      end
    else
      begin
        ln = 0
        cl = 0
        cr = SEARCH (ln, cl, "*****", 6, 3)

        cpt = cpt + 1
      end
    end

    OUTPUT ("Connexion PACBASE", 12)
    SEND ("CLS")
    WRITEC ("V280", 0)

```

```
SEND ("ENT")
ln = 24
cl = 8
cr = SEARCH (ln, cl, "CH:", 3, 12)
if (cr <> 1)
  begin
    ERROR ("PROBLEME SITE CENTRAL", "Pacbase inactif", 1)
    DISPLAY ()
    PAUSE (3)
    EXIT (16)
  end
end
end
```

- Déconnexion :

```
PROG V280SCF
BEGIN
  OUTPUT ("DECONNEXION DE PACBASE", 12)
  SEND ("CLS")
  OUTPUT ("DECONNEXION DE CICST", 12)
  WRITEC ("CESF", 0)
  SEND ("ENT")
END
```

2.1.12 Exemples de scripts BULL-DPS7

- Connexion : Obtention du "\$*\$" par une série de transmissions, attente de la demande de saisie du sign-on TDS, envoi du CN incluant le code utilisateur et le mot de passe; attente de la mire d'acceptation comprenant la chaîne "Ready", envoi du code transaction VisualAge Pacbase et attente de la chaîne "CH:" caractéristique de la mire *VisualAge Pacbase*. Ce script est particulier à un site défini, et doit bien entendu être adapté pour chaque site.

```

INTEGER CR LN CL
INTEGER CPT

BEGIN
OUTPUT ("OBTENTION PROMPT $*$", 12)
CR=0
CPT=0
WHILE (CR <> 1)
  BEGIN
    IF (CPT > 10)
      BEGIN
        ERROR ("PROBL. SERVEUR", "$*$ NON OBTENU", 1)
        DISPLAY ( )
        EXIT (12)
      END
    SEND ("ENT")
    LN = 0
    CL = 0
    CR = SEARCH (LN, CL, "$*$", 3, 1)
    CPT = CPT+1
  END
OUTPUT ("SIGNON TDS", 12)
CR = 0
CPT= 1
WHILE (CR <> 1)
  BEGIN
    IF (CPT > 3)
      BEGIN
        ERROR ("NOMBRE D'ESSAIS LIMITE A 3", "SIGNON
INVALIDE", 1)
        EXIT (15)
      END
    INPUT ("IDENTIFICATION TDS", "UTILISATEUR TDS",USER, 8, 1, "MOT
DE &
PASSE TDS", PASS,8,0)
    SIGNON = ""
    CONCAT(SIGNON, "$*$CN STMTDS4 -USER ", USER, " -PW",
PASS, &
"-STR ?A")
    WRITEM (SIGNON,0)
    SEND ("ENT")
    OUTPUT ("CONNEXION A TDS", 12)
    LN = 0
    CL=0
    CR = SEARCH (LN, CL, "CONVE", 5, 3)
    IF (CR<> 1)
      BEGIN
        LN = 0
        CL=0
        CR = SEARCH (LN, CL, "READY", 5, 3)
      END
    CPT = CPT+1
  END
END

```

```

OUTPUT ("CONNEXION VISUALAGE PACBASE", 12)
WRITEM ("PB00 /192", 0)
SEND ("ENT")
LN = 24
CL=8
CR = SEARCH (LN, CL, "CH:", 3, 45)
IF (CR<> 1)
    BEGIN
        ERROR ("PROBLEME SERVEUR", "VISUALAGE PACBASE INACTIF", 1)

        DISPLAY ()
        EXIT(16)
    END
END

```

- Déconnexion :

```

INTEGER CR LN CL

BEGIN
    OUTPUT ("DECONNEXION DE VISUALAGE PACBASE", 12)
    WRITE (24, 5, "FT", 0)
    SEND ("ENT")
    LN = 24
    CL = 8
    CR = SEARCH (LN, CL, "CH:", 3, 45)
    WRITE (24, 5, "FT", 0)
    SEND ("ENT")

    OUTPUT ("DECONNEXION DE TDS", 12)
    LN = 0
    CL = 0
    CR = SEARCH (LN, CL, "CONVE:", 5, 20)
    WRITEM (3*$LO, 0)
    SEND ("ENT")
END

```

2.1.13 Exemples de scripts BULL-DPS8

- Connexion : Obtention du \$\$\$ par une série de transmissions, envoi du CN à la machine et au TP adéquat, attente de la demande de saisie du logical id, envoi d'un logical id, puis du code transaction VisualAge Pacbase et attente de la chaîne "CH:" caractéristique de la mire *VisualAge Pacbase*. Ce script est particulier à un site défini, et doit bien entendu être adapté pour chaque site.

```

INTEGER CR LN CL
INTEGER CPT

BEGIN
OUTPUT ("OBTENTION PROMPT $$$", 12)
CPT=0
CR=0
WHILE (CR==0)
  BEGIN
    IF      (CPT > 6)
      BEGIN
        ERROR ("PROBLEME SERVEUR", "$$$ NON OBTENU",
1)
          DISPLAY ()
          EXIT (12)
        END
      SEND ("ENT")
      LN = 0
      CL = 0
      CR = SEARCH (LN, CL, " $$$", 3, 3)
      CPT = CPT+1
    END
  OUTPUT ("CONNEXION _ TP8", 12)*****
  WRITEM ("CN-SC XXXX -MB PC", 0)
  SEND ("ENT")
  LN = 0
  CL = 0
  CR = SEARCH (LN, CL, "ID", 2, 15)
  IF (CR <> 1)
    BEGIN
      ERROR ("PROBLEME SERVEUR", "TP8 NON ACTIF", 1)
      DISPLAY ()
      EXIT (12)
    END
  DISPLAY ()

```

```

OUTPUT ("ENVOI LOGICAL ID", 12)
WRITEM ("P001", 0)
SEND ("ENT")
LN = 0
CL = 0
CR = SEARCH (LN, CL, "CONNECT", 7, 12)
IF (CR<> 1)
    BEGIN
        ERROR ("PROBLEME SERVEUR", "ID INVALIDE", 1)
        DISPLAY ()
        EXIT (12)
    END
DISPLAY ()

OUTPUT ("CONNEXION VISUALAGE PACBASE", 12)
WRITEC ("PB00", 0)
SEND ("ENT")
LN = 24
CL = 8
CR = SEARCH (LN, CL, "CH:", 3, 3)
IF (CR<> 1)
    BEGIN
        ERROR ("PROBLEME SERVEUR", " VISUALAGE PACBASE INACTIF", 1)

        DISPLAY ()
        EXIT (16)
    END
END

```

- Déconnexion :

```

INTEGER CR LN CL

```

```

BEGIN
    OUTPUT ("DECONNEXION DE VISUALAGE PACBASE", 12)
    WRITE (24, 5, "FT", 0)
    SEND ("ENT")
    LN = 24
    CL = 8
    CR = SEARCH (LN, CL, "CH:", 3, 45)
    WRITE (24, 5, "FT", 0)
    SEND ("ENT")

    OUTPUT ("DECONNEXION DE TDS", 12)
    LN = 0
    CL = 0
    CR = SEARCH (LN, CL, "CONVE:", 5, 20)
    WRITEM ("*$LO", 0)
    SEND ("ENT")
END

```

2.1.14 Exemples de scripts OS/2, Windows NT ou UNIX

- Connexion :

```
PROG PLANSCI  
BEGIN  
END
```

- Déconnexion :

```
PROG PLANSCF  
BEGIN  
END
```

2.2 Paramétrage des combinaisons de touches équivalentes aux touches fonction F11 et F12

Dans certains cas, énumérés dans les tableaux ci-dessous, vous devez modifier quelques lignes du fichier EMPAC.PRO pour pouvoir activer normalement les combinaisons de touches ayant la même fonctionnalité que les touches F11 et F12.

Ce fichier, automatiquement créé au lancement de l'Emulateur dans le répertoire \SPAC\BASE, contient en colonnes 9, 10 et 11, le paramétrage des touches fonction.

Vous devez modifier les trois chiffres des colonnes 9, 10 et 11 en fonction de la nationalité de configuration de votre clavier dans les cas suivants :

- Vous utilisez une Station de Travail française et un clavier qui n'est pas de configuration française;
 - Vous utilisez une Station de Travail anglaise de version inférieure à la 1.2 V00 et un clavier qui n'est pas de configuration française.
- REMARQUE "----" signifie que vous ne devez rien modifier.

Numéro de ligne	Valeur par défaut	Etats-Unis Gde-Bret. Canada	Allemagne Suisse	Espagne Italie Portugal	Norvège Suède Finlande Danemark	Pays-Bas	Belgique
24	219	189	----	----	187	----	----
25	187	----	221	221	219	191	189
36	219	189	----	----	187	----	----
37	187	----	221	221	219	191	189

Pour les Stations de Travail anglaises de version 1.2 V00 ou plus récentes avec un clavier qui n'est pas de configuration anglaise ou américaine, vous devez modifier les colonnes 9, 10 et 11 de la façon suivante, en fonction de la nationalité de configuration de votre clavier :

Numéro de ligne	Valeur par défaut	France	Allemagne Suisse	Espagne Italie Portugal	Norvège Suède Finlande Danemark	Pays-Bas	Belgique
24	189	219	219	219	187	219	219
25	187	----	221	221	219	191	189
36	189	219	219	219	187	219	219
37	187	----	221	221	219	191	189

ATTENTION La première ou les deux premières lignes du fichier EMPAC.PRO sont différentes des autres lignes et sont optionnelles. L'une ou l'autre ou les deux peuvent être absentes. Si tel est le cas, retranchez un ou deux aux numéros de ligne mentionnés dans les deux tableaux ci-dessus.

3 Lancement de la Station de Travail

3.1 Fichier PACBASE.DAT

Les paramètres nécessaires au fonctionnement de la Station de Travail sont rassemblés dans le fichier PACBASE.DAT. La procédure d'installation de la Station de Travail crée ce fichier dans le répertoire \SPAC\WNNL (où *NNN* désigne le numéro de version et *L* le code langue de la version implantée).

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

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

Voici un exemple de fichier PACBASE.DAT :

```
001 Version Station          [250F]
002 Serveur                  [PACBASE]
003 Gestionnaire de communication [GSEXTW]
004 Paramètres du gestionnaire [GSEXTW]
005 Système d'exploitation    [DOS]
006 Méthode utilisée         [MER]
007 Disque EXE                [C]
008 Disque EXE (défaut)      [C]
009 Disque Données système   [C]
010 Disque Données utilisateur [C]
011 Mode d'exécution connexion [E]
```

Le fichier PACBASE.DAT est initialisé avec les paramètres fournis lors de la première installation de la version *NNNL*. Si une réinstallation de cette même version a lieu sans destruction préalable, le contenu de ce fichier n'est pas mis à jour. L'utilisateur doit alors le modifier manuellement (si besoin).

Le type de serveur est codifié de la manière suivante :

```
Mainframe : [PACBASE]
Unix      : [PACBASE/X]
OS/2     : [PACBASE/OS2]
Windows NT : [PACBASE/WNT]
```

REMARQUE : Si vous êtes connecté via Top Secret, vous devez ajouter dans le fichier PACBASE.DAT une ligne 012 indiquant le chemin d'accès au fichier local qui contient ces codes utilisateurs, afin que la Station puisse les retrouver.

3.2 Fichiers de paramétrage optionnels

Vous pouvez définir des fichiers de paramétrage optionnels pour adapter et choisir les configurations de votre site.

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

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

En cas de réinstallation de la Station, les fichiers *.DAT que vous avez créés ainsi que les fichiers entre crochets dans le fichier BASE.LST seront sauvegardés avant le remplacement de la Station et replacés dans leur répertoire d'origine à la fin de la réinstallation.

3.3 Procédure de lancement

Le lancement de la Station de Travail peut s'effectuer directement sous le système d'exploitation, ou depuis l'interface graphique utilisateur (GUI) installé sur le poste.

Selon l'environnement sur lequel la Station de Travail est installée, les manipulations permettant d'effectuer le lancement varient. Ces différences reposent uniquement sur les différences de fonctionnement entre les environnements. Par exemple, pour changer de répertoire :

- sous DOS, il est nécessaire d'utiliser la commande CD (Change Directory) avec le nom du répertoire (ex. CD \SPAC\NNNL).
- sous l'interface utilisateur Windows, c'est le Gestionnaire de fichiers qui le permet par sélection des répertoires voulus dans l'arborescence affichée dans une fenêtre.

Dans ce chapitre nous indiquons, dans des termes généraux et indépendants des environnements sur lesquels la Station de Travail peut fonctionner, les actions à effectuer sur le répertoire et le programme de lancement. Pour le détail des manipulations liées à votre environnement, consultez les documentations d'utilisation du système d'exploitation et de l'interface graphique sous lesquels votre Station de Travail doit fonctionner.

3.3.1 Répertoire de lancement : \SPAC\WNNL

Avant de procéder au lancement de la Station de Travail, il est nécessaire de se placer sur le répertoire \SPAC\WNNL du disque X: où la Station de Travail a été installée. En effet, les programmes d'initialisation de la Station de Travail doivent accéder à certains fichiers (entre autres PACBASE.DAT) qui se trouvent dans ce répertoire.

Le contenu de ce répertoire est détaillé dans le Sous-chapitre *Arborescence initiale de la Station de Travail et évolutions possibles* du Chapitre *Installation de la Station de Travail*.

3.3.2 Lancement : PEXEC.EXE

Pour lancer la Station de Travail, exécutez le programme PEXEC.EXE. Ce programme s'exécute sous l'interface graphique utilisateur (GUI) correspondant à la version installée : Windows, Presentation Manager ou Win-OS/2.

Par défaut, le programme PEXEC.EXE prend en compte le fichier de paramètre PACBASE.DAT.

Cependant, vous pouvez spécifier un fichier optionnel de paramétrage que vous avez préalablement défini :

- Donnez le nom de ce fichier en paramètre du programme PEXEC.EXE dans la ligne de commande :

EXEMPLE : C:\SPAC\250F\PEXEC.EXE PACBASE.BIS

Les paramètres du fichier PACBASE.DAT ne sont alors pas pris en compte.

- ou, plus spécifiquement, associez le nom de ce fichier à une base dans le fichier BASE.LST. Vous devez encadrer entre crochets le nom du fichier (ex : V280 Base de Validation [PACBASE.TER]). Ce nom apparaîtra dans la liste des bases, à côté du libellé de la base. Ces nouveaux paramètres écraseront les paramètres du fichier (PACBASE.DAT ou PACBASE.BIS) qui ont été utilisés pour lancer la Station de Travail.



Les fichiers optionnels de paramétrage (ici PACBASE.BIS ou PACBASE.TER) doivent exister dans le même répertoire que PACBASE.DAT. Ils doivent être constitués à l'image de PACBASE.DAT (avec des valeurs différentes dans les paramètres). Le nom de ces fichiers est libre. PACBASE.DAT ne doit pas être détruit.

4 Utilitaires de la Station de Travail

4.1 Stimulateur externe

Cet utilitaire vous permet de lancer des applications de la Station de Travail sans passer par les menus de la Station.

Son utilisation est recommandée quand vous travaillez sous le contrôle d'un outil de gestion de projet, tel que LCM ou Microfocus WorkBench.

Vous pouvez le lancer à partir du Gestionnaire de Programmes de Windows.

4.1.1 Fonctionnalités

Le stimulateur ne peut fonctionner que si le Gestionnaire de la Station de Travail est actif.

Il permet :

- d'accéder aux définitions et aux descriptions d'occurrences (mais pas aux listes ni aux chaînages).
- d'ouvrir les fenêtres de service suivantes: Bibliothèques-sessions, Edition-génération, Emulateur, Extraction de données, Journal, Recherche sur mots-clés, et Suivi des Editions-génération (pour la Station de Travail installée sur serveur OS/2, Windows NT ou UNIX uniquement).

4.1.2 Activation

Pour activer cet utilitaire, sélectionnez le choix **Exécuter** dans le menu **Fichier** du Gestionnaire de Programme de Windows et saisissez la ligne de commande suivante :

- Dans le cas de l'accès à une description :

```
C:\SPAC\250FWSTIM MODULE EEE XXXXXX Dd,
```

où :

MODULE est le module de travail;

EEE le code de l'entité;

XXXXXX le code de l'occurrence;

Dd le code de la description;

(EEE et Dd sont les codes locaux; pour la liste des codes locaux, référez-vous à l'aide en ligne).

- Dans le cas de l'accès à une Définition :

C:\SPAC\250F\WSTIM MODULE EEE XXXXXX DEF,

où :

MODULE est le module de travail;

EEE le code de l'entité;

XXXXXX le code de l'occurrence;

DEF le code de la définition;

(EEE est un code local. Pour la liste des codes locaux, référez-vous à l'aide en ligne).

- Dans le cas de l'ouverture d'une fenêtre de service :

C:\SPAC\250F\WSTIM XXXXX,

où XXXXX est le code de l'application à lancer (BIBSS pour Bibliothèques-sessions, EDGEN pour Edition-génération, EMPAC pour Emulateur, EXTMQ pour Extraction de données, JOURN pour Journal, WSRCH pour Recherche sur mots-clés et GPMON pour Suivi des Editions-génération).

Pour représenter les différents stimulateurs (ex : un stimulateur de l'application Edition-Génération, un autre de l'application Journal ...) dans le Gestionnaire de Programmes, vous pouvez sélectionner n'importe quel icône de la Station de Travail.



Dans le cas où le stimulateur externe est activé à partir de LCM, ce dernier est capable de déterminer la durée de travail effectuée sur la description demandée et d'en déduire la charge de travail effectuée et le "reste à faire", nécessaires au suivi du projet.

4.2 Application Edition locale de rapports au format RTF

L'application Edition locale ILRTF.EXE remplace l'utilitaire d'impression locale IMPLO.EXE et élargit son champ d'action. Il est automatiquement installé avec les applications de la Station de Travail, et il est rangé dans le sous-répertoire SPAC\NNNL correspondant au sous-répertoire de la Station de Travail (version française ou anglaise).

Le fichier de paramétrage de cet utilitaire est ILRTF.PRO. Il se trouve dans le sous-répertoire SPAC\BASE.

Si le traitement de texte que vous allez utiliser n'est pas en version française, vous devez spécifier, dans la **ligne 6** du fichier ILRTF.PRO, l'équivalent du style "Titre" de votre traitement de texte. Cette information est indispensable pour que le logiciel de traitement de texte reconnaisse ce style et puisse ainsi faire la numérotation des titres dans le rapport. En l'absence de cette ligne, le nom de style pour les titres est "Titre".

Par ailleurs et pour information, le fichier ILRTF.PRO contient des informations exploitées par l'application ILRTF.EXE. A la première utilisation de l'application, les champs de la fenêtre "Conversion d'un fichier GDP au format RTF" sont vides. Puis à chaque utilisation, les chemins des fichiers saisis dans chaque champ sont mémorisés dans le fichier ILRTF.PRO. Ainsi, les fichiers mémorisés alimenteront une liste déroulante pour chaque champ de l'application. Lorsque la liste déroulante devient trop longue, vous avez la possibilité de supprimer des chemins de fichiers dans ILRTF.PRO.

Les informations suivantes permettent d'alimenter ces champs :

- Les **lignes 1** contiennent les chemins des fichiers qui ont été saisis dans le champ "GDP source".
- Les **lignes 2** contiennent les chemins des fichiers qui ont été saisis dans le champ "Feuille de styles".
- Les **lignes 3** contiennent les chemins des fichiers qui ont été saisis dans le champ "Cible".
- La **ligne 4** contient le chemin du fichier PACBASE.DAT (ou équivalent) qui a été saisi dans le champ "Fichier de configuration de la Station de Travail".
- La **ligne 5** contient le chemin du fichier de traitement de texte utilisé qui a été saisi dans le champ "Traitement de texte associé" de la boîte de dialogue indiquant les résultats de la génération.

NB La **ligne 7** indique le style des titres qui a été spécifié à l'aide des 2 boutons radio de la fenêtre:

1 indique le style standard.

2 indique le style GDP.

4.3 Utilitaire de "nettoyage" de la Station de Travail

Un utilitaire est fourni pour fermer toute application restée ouverte après un arrêt inopiné de la Station de Travail. En effet, il est toujours possible que des problèmes externes au fonctionnement de la Station de Travail provoquent une interruption.

Dans ce cas, lancez l'utilitaire KILLP.EXE situé dans le répertoire \SPACWNNL.

5 ANNEXE *Le langage de Connexion*

Introduction

Les procédures de connexion et de déconnexion sont basées sur un processus d'échange entre l'utilisateur et le système. L'utilisateur saisit une commande sur son poste et la soumet au système, qui réagit en conséquence : soit il traite la commande et attend la suite, soit il la rejette et le signale à l'utilisateur.

La connexion aux sites centraux s'avérant parfois fastidieuse, un langage interprété permet de l'établir automatiquement par l'écriture de programmes, les scripts de connexion, capables d'assurer un échange avec le système, de tester sa réponse et de réagir en conséquence.

5.1 Structure d'un script

La structure d'un script de connexion est :

```

PROG Nom_de_script
  déclaration(s)
BEGIN
  instruction(s)
END

```

La première ligne du script doit avoir la structure suivante :

```

PROG nom_de_script

```

Elle permet d'indiquer le nom du programme.

Le mot-clé PROG peut être écrit en minuscules ou en majuscules. Le nom du programme est une suite de lettres et de chiffres commençant par une lettre, de 16 caractères au maximum. Il ne doit pas être un mot réservé. (La liste des mots réservés est donnée ci-dessous.)

Aucune autre information ne peut figurer sur cette ligne.

5.2 Mots réservés

Les mots-clés et les noms de fonctions sont réservés. Ils ne peuvent être utilisés comme noms de programmes et de variables ou comme étiquettes.

AND	BEGIN	BOOLEAN
BREAK	DO	ELSE
END	FALSE	FOR
GOTO	IF	ENTIER
NOT	OR	PROG
CHAINE	TRUE	UNTIL
WHILE		

Les mots réservés suivants sont les noms des fonctions prédéfinies. Ils doivent être entrés en majuscules :

CONCAT	DISPLAY	ERROR
EXIT	CURPOS	CPYSCR
INPUT	INTSTR	OUTPUT
PAUSE	READ	READC
RPARAM	SEARCH	SEND
WRITE	WRITEC	WRITEM

5.3 Déclarations

La partie déclarative du programme permet de définir les variables et étiquettes utilisées dans les instructions. Elle se situe entre l'en-tête et le corps du programme (si aucune variable n'est utilisée, cette partie peut être supprimée).

- déclaration de variable :

```
Type_de_variable      Nom_de_variable
```

- déclaration d'étiquette :

```
$label
```

5.3.1 Types de variables

Il existe trois types de variables : entier (noté integer), chaîne (notée string), booléen (noté boolean).

Une déclaration de variable s'écrit donc:

```
      STRING      nom_de_variable
ou   INTEGER      nom_de_variable
ou   BOOLEAN      nom_de_variable
```

Plusieurs variables du même type peuvent être déclarées sur une même ligne:

```
      STRING      nom1 nom2
```

est équivalent à:

```
      STRING      nom1
      STRING      nom2
```

5.3.2 Noms de variables

Un nom de variable est une suite de caractères alphanumériques, commençant par une lettre. La longueur en est limitée à 16 caractères. Toute variable utilisée dans le programme doit impérativement être déclarée. Deux variables ne peuvent pas avoir le même nom. Un nom de variable ne doit être ni un mot-clé, ni le nom d'une fonction (voir ci-dessus la liste des mots réservés).

Les noms de variable doivent être suivis d'un caractère blanc ou d'un autre caractère non alphanumérique. Une déclaration se termine impérativement par un retour à la ligne.

5.3.3 Valeurs des variables

L'indication de la valeur d'une variable s'écrit :

```
nom_de_variable=Valeur
```

Les entiers sont compris entre (-maxint) et (maxint -1), maxint dépendant de la machine utilisée ($2^{15} = 32768$ sur IBM PC). Ils sont manipulés par les opérateurs décrits dans la section INSTRUCTIONS.

Les chaînes de caractères ont une longueur maximum de 132 caractères. Elles sont entrées entre guillemets. Elles peuvent comprendre des blancs. Lorsque l'un des caractères de la chaîne est le guillemet lui-même, il doit être doublé.

Lorsqu'une chaîne ne tient pas sur une seule ligne, il est nécessaire d'insérer un "&", avant le retour à la ligne. Il peut uniquement y avoir des blancs ou des marques de tabulation entre le caractère de continuation et le retour à la ligne, sinon le "&" est considéré comme un caractère quelconque de la chaîne. La suite de la chaîne commence à la colonne 1 de la ligne suivante.

Exemples:

```
STRING="Il dit ""Bonjour""."
```

Le contenu de la chaîne est :

Il dit "Bonjour".

```
STRING="Cette chaîne se continue&  
sur la ligne suivante."
```

Le contenu de la chaîne est :

Cette chaîne se continue sur la ligne suivante.

```
STRING="Cette chaîne contient un &(et commercial)."
```

Le contenu de la chaîne est :

Cette chaîne contient un &(et commercial).

Les booléens ne peuvent prendre que deux valeurs, notées TRUE pour vrai et FALSE pour faux. Ils peuvent être combinés avec des opérateurs logiques (voir ci-dessous). Ce sont les booléens qui entrent en compte dans toutes les structures conditionnelles.

5.3.4 Etiquettes

Les étiquettes utilisées dans les instructions de branchement (GOTO \$XXX) doivent également être déclarées. Le nom d'une étiquette est constitué de chiffres, 3 au maximum.

Déclaration d'étiquette :

```
$nom_d'étiquette
```

Aucune autre information ne peut figurer sur la ligne. Toutes les étiquettes du programme doivent avoir des noms distincts.

5.3.5 Commentaires

Les commentaires, placés entre ":", ne sont pas interprétés. Ils peuvent être insérés partout, sauf dans une chaîne de caractères. Il est possible d'écrire plusieurs lignes de commentaires en plaçant un ":" au début de la première ligne, et un second à la fin de la dernière ligne. Les instructions incluses dans un commentaire sont ignorées.

Exemples :

```

PROG      NOM_DE_PROGRAMME
:CE PROGRAMME A UN COMMENTAIRE ETALE SUR
PLUSIEURS LIGNES :
BEGIN END
PROG NOM_DE_PROG      :CE PROGRAMME A PLUSIEURS LIGNES:
BEGIN                  :DE COMMENTAIRE DANS LA MARGE:
END
PROG NOM_DE_PROG      :CE COMMENTAIRE RECOUVRE LE MOT
BEGIN                  CLE BEGIN, IL Y AURA DONC ERREUR :
END
PROG      NOM_DE_PROGRAMME
CHAINE    NOM_DE_CHAINE
BEGIN     NOM_DE_CHAINE="ARTHUR:COMMENTAIRE:"
END

```

La chaîne vaut ARTHUR:COMMENTAIRE: et non ARTHUR.

5.3.6 Espacements et retours à la ligne

Les espacements et les tabulations sont équivalents, et marquent la fin d'une suite de caractères. Dans la suite de ce chapitre, seront soulignés les cas où cette séparation est indispensable. Un mot-clé immédiatement suivi d'un caractère alphanumérique n'est pas reconnu. Par exemple, un en-tête ne peut s'écrire : PROGnom

Un retour à la ligne marque la fin d'une instruction. Les cas où le retour à la ligne est obligatoire seront indiqués. Il peut s'avérer nécessaire de continuer une instruction sur plusieurs lignes, en particulier lorsqu'une fonction a plusieurs paramètres (voir ci-après). En ce cas, il est possible d'indiquer que le retour à la ligne ne représente pas la fin de l'instruction en insérant un "&" avant le retour à la ligne. La suite de la ligne ne doit pas comprendre de caractère autre que tabulation et blanc : tout autre caractère inséré après le "&" provoque une erreur.

Exemple :

```

INPUT("sign-on CICS","Code Utilisateur",&UT1,6,1,)
équivaut à :
INPUT("sign-on CICS","Code Utilisateur",UT1,6,1,)

```

Lorsque le retour à la ligne est autorisé, il est possible de laisser une ou plusieurs lignes blanches, sauf dans la structure de contrôle IF ELSE (voir plus loin) qui n'admet pas de ligne blanche avant le ELSE.

5.3.7 Corps du programme

Le corps du programme est délimité par le mot-clé BEGIN et le mot-clé END. Il est constitué d'une suite d'instructions et de structures, dont les syntaxes sont décrites plus loin.

5.3.8 Instructions

Les instructions constituant le corps du programme sont constituées d'appels de fonctions, de sauts à une autre partie du programme et d'affectations de valeurs aux expressions.

5.4 Affectation

L'affectation permet d'attribuer à une variable la valeur d'une expression. Elle s'écrit :

```
nom_de_variable = expression
```

Aucune autre information ne peut figurer sur la ligne.

L'expression peut être soit une autre variable, soit une constante de type entier, chaîne ou booléen, soit un appel de fonction, soit enfin une expression où des opérateurs sont à des expressions, selon les règles édictées dans le paragraphe qui suit.

5.5 Expressions et Opérateurs

Les expressions simples sont du type :

- nom_de_variable
- constante entière
- constante littérale
- constante booléenne

Des opérateurs arithmétiques et logiques permettent d'obtenir des expressions complexes :

expression : expressionA Opérateur expressionB

Des opérateurs arithmétiques combinés avec des opérandes entiers permettent d'obtenir des entiers:

- expressionA	opposé de expressionA
expressionA + expressionB	somme des expressions
expressionA - expressionB	différence des expressions
expressionA * expressionB	produit des expressions
expressionA / expressionB	quotient des expressions

Des opérateurs logiques permettent d'obtenir des résultats booléens (dont la valeur est soit TRUE, soit FALSE) :

- $expA < expB$ donne TRUE si $expA$ est inférieur à $expB$; $expA$ et $expB$ doivent être entiers.
- $expA > expB$ donne TRUE si $expA$ est supérieur à $expB$; $expA$ et $expB$ doivent être entiers.
- $expA == expB$ donne TRUE si $expA$ est égal à $expB$; $expA$ et $expB$ doivent être de même type.
- $expA <> expB$ donne TRUE si $expA$ est différent d' $expB$; $expA$ et $expB$ doivent être de même type.
- $expA$ and $expB$ donne TRUE si $expA$ et $expB$ sont vrais ; $expA$ et $expB$ doivent être booléens.
- $expA$ or $expB$ donne TRUE si $expA$ ou $expB$ est vrai ; $expA$ et $expB$ doivent être booléens.
- not $expA$ donne TRUE si $expA$ est FALSE et vice versa ; $expA$ doit être booléen.

5.5.1 Priorité des opérateurs

Chaque opérateur est doté d'une priorité et de propriétés relatives à l'associativité. L'opération correspondant à l'opérateur de plus grande priorité est d'abord effectuée, puis on passe à l'opérateur de seconde priorité etc. Si deux opérateurs de même priorité sont adjacents, on se réfère à leur associativité. Les opérateurs sont indiqués ci-dessous dans un ordre de priorité décroissante.

-	unaire
* /	associatifs
- +	associatifs
< > <> ==	non associatifs
NOT	non associatif
AND OR	associatifs

5.5.2 Traitement des opérateurs associatifs

Pour évaluer $A + B + C$
 on effectue $\text{intermédiaire} = A + B$
 et le résultat est : $\text{intermédiaire} + C$.

Deux opérateurs non associatifs ne peuvent pas être adjacents. Ainsi le produit de A par la somme de B et de C doit s'écrire :

$$A*B+A*C$$

Les parenthèses "(" et ")" permettent de changer l'ordre d'évaluation. Une expression entre parenthèses équivaut à une variable temporaire, les parenthèses les plus profondes étant évaluées en priorité. Ainsi :

$$A=(B+C-D)*(E/F)$$

est équivalent à :

$$\text{intermédiaire}_1=B+C$$

$$\text{intermédiaire}_2=\text{intermédiaire}_1-D$$

$$\text{intermédiaire}_3=E/F$$

$$A=\text{intermédiaire}_2*\text{intermédiaire}_3$$

Les parenthèses permettent également de rendre les expressions plus lisibles sans modifier l'ordre d'évaluation. Leur utilisation est recommandée pour les expressions dont les opérandes sont de types différents, par exemple:

$$(a>0) \text{ and } (a<11)$$

5.6 Branchements

L'instruction GOTO \$XXX permet de ne pas effectuer le programme en séquence en allant directement à l'endroit désigné par l'étiquette \$XXX.

Le blanc entre "GOTO" et "\$" est facultatif, et le retour à la ligne obligatoire.

Tout branchement n'est pas licite : on ne peut entrer dans un bloc d'instructions par un branchement, même si l'on peut en sortir.

5.7 Structures de contrôle

Les structures de contrôle permettent d'effectuer des choix et des itérations.

Les structures de contrôle sont :

IF condition	WHILE condition	DO	FOR condition
ELSE	UNTIL condition		
bloc			

Une condition est une expression de type booléen encadrée par des parenthèses et s'écrit :

(expression)

La condition est vraie si et seulement si ce booléen vaut TRUE.

La condition de la structure FOR est particulière ; elle est décrite ci-dessous.

La condition et le mot-clé qui la précède doivent être écrits sur la même ligne. Ils peuvent être séparés par un blanc.

Un bloc d'instructions regroupe plusieurs instructions qui sont exécutées en séquence en fonction de critères spécifiés par une structure de contrôle. Ces instructions peuvent elles-mêmes être des structures de contrôle.

Le bloc d'instructions est précédé du mot-clé BEGIN et suivi du mot-clé END.

Les branchements peuvent se faire uniquement depuis l'intérieur d'un bloc vers l'extérieur ; les branchements depuis l'extérieur dans un bloc ne sont pas possibles.

5.8 Ecriture des expressions

Un bloc d'instructions s'écrit :

```
BEGIN instructions    END
```

Un retour à la ligne est obligatoire après le bloc. On peut en insérer avant le bloc, après BEGIN, avant et après les instructions.

IF ELSE

```
IF    condition
      BEGIN
      instructions
      END
ELSE
      BEGIN
      instructions
      END
```

Quand l'interpréteur rencontre cette structure, il évalue la condition, puis exécute le premier bloc si elle est vraie, le second dans le cas contraire.

La partie ELSE BEGIN instructions END est facultative. Dans ce cas, le programme poursuit immédiatement en séquence si la condition est fausse.

On peut insérer un seul retour à la ligne entre le premier END et ELSE.

WHILE

```
WHILE condition
      BEGIN
      instructions
      END
```

Le bloc est exécuté tant que la condition est vraie. Si la condition est fausse dès la première évaluation, le bloc n'est jamais exécuté. Il faut bien entendu s'assurer que la condition évolue dans le temps vers la valeur FALSE, si l'on ne veut pas engendrer une boucle infinie.

DO UNTIL

```
DO
    BEGIN
        instructions
    END
UNTIL condition
```

Le bloc est exécuté tant que la condition est fausse. Il est exécuté au moins une fois. Il faut bien entendu s'assurer que la condition évolue dans le temps vers la valeur TRUE, si l'on ne veut pas engendrer une boucle infinie.

FOR

```
FOR (expressionA, nom_de_variable, expressionB)
    BEGIN
        Instructions
    END
```

La variable "nom_de_variable" doit être de type entier : elle sert de compteur. ExpressionA est évaluée une fois ; sa valeur doit être un entier, c'est la borne inférieure. De même, expressionB fournit la borne supérieure. Le compteur prend la valeur d'ExpressionA ; si expressionB lui est supérieure ou égale, le bloc est exécuté. Le compteur est incrémenté et est à nouveau comparé à la borne supérieure.

Si expressionA est supérieure à expressionB, le bloc n'est pas exécuté, et le programme reprend en séquence à la fin du bloc.

Cette structure permet d'exécuter un bloc un nombre fixé de fois ($\text{expressionB} - \text{expressionA} + 1$). Elle utilise un compteur, qui repère l'itération en cours. Ce compteur ne peut être utilisé en tant que variable dans le bloc : il ne peut ni subir d'affectation ni servir de compteur pour une boucle FOR imbriquée. S'il est utilisé par ailleurs dans le programme, sa valeur antérieure est perdue à l'entrée de la boucle FOR.

Rupture de boucle

Le mot-clé BREAK permet de sortir de la structure FOR, WHILE ou DO UNTIL qui l'englobe. Cela permet de sortir prématurément d'une structure itérative, en général à la suite d'un test. Un BREAK en dehors de toute structure FOR, WHILE ou DO UNTIL provoque une erreur.

Le mot BREAK doit être suivi d'un retour à la ligne.

5.9 Fonctions

Les fonctions peuvent être utilisées comme instructions ou comme expressions. Certaines fonctions rendent une valeur (code retour). D'autres sont destinées à inter-agir avec l'environnement (affichage sur la sortie standard par exemple), et permettent au programme de dialoguer avec l'extérieur.

5.9.1 Appel d'une fonction

L'appel d'une fonction est de la forme :

```
nom_de_fonction (liste_de_paramètres)
```

Pour chaque fonction, il sera précisé si elle rend une valeur ou non et le type de la valeur le cas échéant. Les fonctions qui rendent des valeurs peuvent être utilisées, soit dans une affectation, soit comme instruction, mais la valeur du code retour est alors perdue. En revanche, une fonction ne rendant pas de valeur ne peut figurer que dans une instruction.

5.9.2 Paramètres d'une fonction

Les paramètres sont de deux types : les paramètres constants et les paramètres modifiables. Un paramètre constant n'est pas modifié par la fonction. Ce peut être une expression ou une variable. Un paramètre modifiable est transformé par la fonction. Ces modifications sont indiquées dans la description des fonctions.

Les paramètres doivent être séparés par une virgule ; ils peuvent être de type entier, chaîne ou booléen. Leur nombre et le type de chacun d'entre eux sont indiqués pour chaque fonction et doivent être respectés (sous peine d'erreur à l'exécution).



Dans la description des fonctions, le mot CNT indique un paramètre constant, le mot VAR un paramètre modifiable. CR représente le code retour.

5.10 Fonctions de communication

PAUSE

Temporisation

cr PAUSE (Par1)

Par1	cnt entier	temps en secondes
------	------------	-------------------

cr	entier	1
----	--------	---

SEND

Envoi d'une touche

cr SEND (Par1)

Par1	cnt chaîne	touche à envoyer
cr	entier	1 envoi normal
		2 touche inconnue
		3 pas de réponse du site central

TOUCHES TOUS MATERIELS

ENT	enter
GCHE	déplacement curseur vers la gauche
DRTE	déplacement curseur vers la droite
HAUT	déplacement curseur vers le haut
BAS	déplacement curseur vers le bas
TAR	tabulation arrière (écran formaté)
TAV	tabulation avant (écran formaté)
PF1 à PF24	Touches fonctions

PROTOCOLE 3270 SEULEMENT

ATTN	attn
ATTR	identifiants
CLS	clear
CURS	cursor select
DEL	supprime un caractère
DUP	dup
EEOF	erase EOF
ERA	erase
HOME	début de ligne
PA1	AP1
PA2	AP2
PRN	print
REST	restaure
RSET	reset
SYST	appel système

WRITE

Ecriture d'une chaîne à l'écran

cr WRITE (Parm1, Parm2, Parm3 , Parm4)

Parm1	cnt entier	ligne	
Parm2	cnt entier	colonne	
Parm3	cnt chaîne	chaîne à écrire	
Parm4	cnt entier	modification du MDT (mode 3270)	
		affichage de la chaîne en mode test	
0	pas de modification du MDT, chaîne visible		
1	modification du MDT, chaîne visible		
2	pas de modification du MDT, chaîne masquée		
3	modification du MDT, chaîne masquée		
cr	entier	1	écriture réussie
		2	erreur écriture

WRITEC

Ecriture d'une chaîne à l'écran à la position du curseur

cr WRITEC (Parm1, Parm2)

Parm1	cnt chaîne	chaîne à écrire	
Parm2	cnt entier	modification du MDT (mode 3270)	
		affichage de la chaîne en mode test	
0	pas de modification du MDT, chaîne visible		
1	modification du MDT, chaîne visible		
2	pas de modification du MDT, chaîne masquée		
3	modification du MDT, chaîne masquée		
cr	entier	1	écriture réussie
		2	erreur écriture

(Si le curseur n'est pas sur une zone saisissable, l'écriture échoue.)

WRITEM

Ecriture d'une chaîne en mode non formaté - BULL seulement

cr WRITEM (Parm1, Parm2)

Parm1	cnt chaîne	chaîne à écrire	
Parm2	cnt entier	en mode test :0 = chaîne visible	
		2 = ch. masquée	
cr	entier	1	écriture réussie
		2	erreur écriture

READ

Lecture d'une chaîne

cr READ (Parm1, Parm2, Parm3, Parm4)

Parm1	cnt entier	ligne
Parm2	cnt entier	colonne
Parm3	var chaîne	chaîne recevant le résultat de la lecture
Parm4	cnt entier	longueur de la zone de lecture

cr	entier	1	lecture réussie
----	--------	---	-----------------

READC

Lecture d'une chaîne sur l'écran à la position du curseur

cr READC (Parm1, Parm2)

Parm1	var chaîne	chaîne recevant le résultat de la lecture
Parm2	cnt entier	longueur de la zone de lecture

cr	entier	1	lecture réussie
----	--------	---	-----------------

SEARCH

Recherche d'une chaîne de caractères sur l'écran

cr SEARCH (Parm1, Parm2, Parm3, Parm4, Parm5)

Parm1	var entier	ligne de début de recherche (toutes lignes : valeur 0)
Parm2	var entier	colonne de début de recherche (toutes colonnes : valeur 0)
Parm3	cnt chaîne	chaîne à rechercher
Parm4	cnt entier	longueur de la chaîne
Parm5	cnt entier	temps maximum de recherche en secondes.

cr	entier	0	chaîne non trouvée
		1	chaîne trouvée

CURPOS

Lit la position du curseur sur l'écran serveur

curpos (Parm1, Parm2)

Parm1	var entier	numéro de ligne
Parm2	var entier	numéro de colonne

DISPLAY

Affichage de l'écran serveur

cr DISPLAY()

A utiliser en phase de test ou en cas d'erreur, afin que l'utilisateur puisse vérifier la cause de l'erreur. L'écran affiché n'est pas saisissable.

Aucun Paramètre

cr entier 1

CPYSCR

Copie l'écran du serveur sur un fichier PC

Cette commande n'interrompt pas l'exécution du script; elle est surtout utile pour les tests et en cas d'erreur pour permettre à l'utilisateur de retrouver la cause d'erreur.

Les écrans se cumulent dans le fichier dont la destruction est à la charge de l'utilisateur.

cr CPYSCR(Parm1)

Parm1 var chaîne nom complet du fichier dos
(U:\PATH\PREFIX.EXT)

cr entier 1

5.11 Fonctions d'entrée-sortie

INPUT

Création d'une boîte de dialogue

```
cr INPUT (Parm1,ParmN,ParmN+1,ParmN+2,1ParmN+3,&
          ParmN,ParmN+1,ParmN+2,1ParmN+3,&
          ...
          ParmN,ParmN+1,ParmN+2,1ParmN+3)
```

Cette fonction a un nombre variable de paramètres :

Parm1 cnt chaîne titre de la boîte de dialogue
 Les paramètres se regroupent ensuite par quatre, chaque groupe correspondant à une ligne de dialogue dans la fenêtre; le nombre de lignes n'est pas limité.

ParmN cnt chaîne libellé de la ligne
 ParmN+1 var chaîne variable mémorisant la valeur saisie par l'utilisateur
 ParmN+2 cnt entier nombre de caractères de la zone saisissable
 ParmN+3 cnt entier affichage des caractères saisis :
 1 OUI
 2 NON (Mot de Passe)

cr entier 1 OK

Exemple :

```
INPUT("Sign-on CICS","Code Utilisateur",UTI,6,1,"Mot de Passe",PAS,8,0)
```

Cette commande permet de construire la boîte de dialogue suivante:

The diagram shows a rectangular dialog box with a title bar at the top containing the text "Sign-on CICS". Below the title bar, there are two rows of labels and input fields. The first row has the label "Code Utilisateur" followed by a rectangular input field. The second row has the label "Mot de Passe" followed by a rectangular input field. At the bottom center of the dialog box, there is an oval-shaped button containing the text "OK".

Le code utilisateur (UTI) visible, est sur 6 caractères, le mot de passe (PAS) invisible est sur 8 caractères. La chaîne correspondant au titre ("Sign-on CICS") est tronquée à 50 caractères.

OUTPUT

Message devant apparaître à l'utilisateur

cr OUTPUT (Parm1, Parm2)

Parm1	cnt chaîne	message à afficher
Parm2	cnt entier	type d'affichage
0 ou >6		le message s'intègre dans la boîte de connexion, permet de matérialiser les différentes phases de la connexion.
1 à 6		boîte de message en surimpression sur la boîte de connexion ; utiliser plutôt l'instruction ERROR (voir cette fonction pour la signification des valeurs)
cr	entier	1 si la valeur du paramètre 2 est 0 ; voir les codes retour de ERROR dans les autres cas.

ERROR

Message d'erreur dans une boîte de dialogue

cr ERROR (cnt chaîne, cnt chaîne, cnt entier)

Parm1	cnt chaîne	en-tête de la boîte
Parm2	cnt chaîne	message d'erreur
Parm3	cnt entier	boutons accessibles à l'utilisateur :

Windows français	Windows anglais
1 OK	1 OK
2 OK et Annuler	2 OK Cancel
3 Abandon Reprise Ignore	3 Abort Retry Ignore
4 Oui Non Annuler	4 Yes No Cancel
5 Oui Non	5 Yes No
6 Reprise Annuler	6 Retry Cancel

cr entier valeur du bouton pressé par l'utilisateur :

Windows français	Windows anglais
1 OK	1 OK
2 Annuler	2 Cancel
3 Abandon	3 Abort
4 Reprise	4 Retry
5 Ignore	5 Ignore
6 Oui	6 Yes
7 Non	7 No

RPARAM

Lit, durant l'exécution du script, les paramètres indiqués pour la connexion à VisualAge Pacbase

cr RPARAM(cnt entier,var chaîne)

Parm1	cnt entier	type de paramètre
Parm2	var chaîne	chaîne recevant le paramètre
cr	entier	1 OK
		2 Valeur de Parm1 erronée
Type de paramètre		code utilisateur
0		1 mot de passe
		2 code base
		3 code bibliothèque
		4 numéro de session
		(session courante : 9999)
		5 type de session (session
		(session historisée : H
		session test : T)
		6 code produit DSMS
		7 numéro d'amélioration DSMS

5.12 Fonctions technologiques

EXIT

Sortie de l'exécution du script

exit (Parm1)

Parm1 cnt entier valeur différente de 0 et de 1

CONCAT

Effectue une concaténation de chaînes de caractères

cr CONCAT (Parm1,ParmM, ... ,ParmX)

Parm1 var chaîne chaîne recevant le résultat de la
concaténation

ParmM à
ParmX cnt chaîne chaînes à concaténer

cr entier longueur de la chaîne concaténée

Si la chaîne recevant le résultat n'est pas vide, elle est également concaténée ; assurez vous que la longueur de la chaîne concaténée est bien inférieure ou égale à 132 (longueur maximum d'une chaîne).

INTSTR

Convertit un entier en une chaîne

cr INTSTR (Parm1, Parm2)

Parm1 var chaîne chaîne résultat
Parm2 cnt entier entier à convertir

cr entier longueur de la chaîne résultat

5.13 ERREURS

Lorsque l'interpréteur détecte une erreur, il interrompt l'exécution du script et édite un message comprenant le plus souvent le numéro de la ligne erronée et, si possible, un diagnostic.

Les erreurs se répartissent en trois groupes : les erreurs de contexte, les erreurs de syntaxe et les erreurs à l'exécution. Les deux premiers comprennent les contrôles effectués lors de la lecture du code source : un suivi des types des variables, de leur utilisation, des branchements et des étiquettes et le contrôle de la syntaxe. Le troisième groupe correspond aux erreurs survenant lors de l'exécution même du programme.

5.13.1 Erreurs de contexte

La principale caractéristique des erreurs de contexte est de ne pas provoquer l'arrêt de la lecture du programme : lorsqu'une erreur de contexte est détectée, un message d'erreur est édité, et la lecture du programme se poursuit. En revanche, le programme ne sera pas exécuté, il doit auparavant être corrigé.

L'interpréteur détecte les incohérences ou le non-respect des règles concernant les expressions, les affectations, les paramètres. Ces règles ont été indiquées dans la description du langage.

Au niveau des variables, plusieurs contrôles sont effectués. L'utilisation de la valeur d'une variable non initialisée (c'est-à-dire n'ayant subi ni affectation directe, ni utilisation comme paramètre par variable d'une fonction prédéfinie) provoque une erreur. L'utilisation d'une variable non déclarée de même. Si une variable est déclarée mais pas utilisée dans le programme, le fait est signalé, sans qu'il y ait erreur.

Les branchements inconditionnels (GOTO) sont soumis à des règles de contexte strictes. L'utilisation d'étiquettes non déclarées provoque une erreur, ainsi que les branchements vers l'intérieur d'un bloc d'instructions. Si une étiquette est déclarée mais pas utilisée dans le programme, le fait est signalé, sans qu'il y ait erreur.

5.13.2 Erreurs de syntaxe

Lorsque la **syntaxe** n'est pas respectée, l'interpréteur rend immédiatement la main à l'utilisateur et édite un message d'erreur. Le message d'erreur indique le numéro de la ligne où l'erreur est constatée. Ce numéro correspond, non pas à la ligne qu'il faut corriger, mais à celle à partir de laquelle on peut affirmer sans ambiguïté que la syntaxe est incorrecte. Dans l'exemple, l'erreur ne sera détectée qu'à la dernière ligne, alors qu'il manque un END à la ligne 12 :

```
1   PROG NOM_DE_PROG
2   CHAINE SELECTEUR
3   BEGIN
4       SELECTEUR="OUVERT"
5       IF (SELECTEUR=="OUVERT")
6           BEGIN
7               WRITEM ("LANCEMENT")
8           END
9       ELSE
10          BEGIN
11              WRITEM ("ATTENTE")
12              PAUSE (25)
13  END
```

L'erreur ne peut être détectée plus tôt car l'interpréteur ne peut déterminer à l'avance le nombre de lignes contenues dans le bloc «ELSE».

Les erreurs les plus communément rencontrées sont :

- l'oubli ou l'ajout d'un délimiteur de bloc, BEGIN ou END ;
- l'oubli d'un passage à la ligne obligatoire ;
- l'ajout d'une ligne blanche avant le ELSE ;
- l'ajout ou l'oubli de parenthèses dans les expressions complexes (vérifier qu'elles vont bien par paires) ;
- l'oubli des parenthèses pour une condition ;
- l'oubli du END final.

5.13.3 Erreurs à l'exécution

Lors de l'exécution du programme peuvent apparaître des problèmes non prévisibles à la seule vue du code.

Selon la longueur, la complexité du code et la présence d'algorithmes erronés, les limitations physiques de la machine peuvent entrer en ligne de compte. Ces limitations visent la mémoire et le temps maximal d'exécution des routines d'un utilisateur ; elles sont très dépendantes du site et du système d'exploitation. Leur dépassement (boucle infinie par exemple), provoque sur certains sites une erreur explicite, et sur d'autres un comportement aberrant de la machine.

Les calculs arithmétiques peuvent engendrer des dépassements de la valeur absolue maximale pour un entier (32767). Ceci mène à des résultats erronés mais pas toujours de façon évidente (le calcul de $a+b-c$, avec a , b et c grands peut faire intervenir une erreur dans le calcul de $a+b$, alors que le calcul de $a+(b-c)$ peut se dérouler sans problème). L'interpréteur arrête l'exécution dans tout dépassement arithmétique (le message d'erreur n'indique pas le numéro de la ligne fautive).