VisualAge Pacbase

Information Support Technique Fichiers complémentaires à VisualAge Pacbase

Version 3.0

VisualAge Pacbase

Information Support Technique Fichiers complémentaires à VisualAge Pacbase

Version 3.0

Note

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Notices» à la page v.

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

http://www.ibm.com/software/ad/vapacbase/productinfo_f.htm

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

troisième édition (Octobre 2002)

La présente édition s'applique à :
• VisualAge Pacbase Version 3.0

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

IBM Paris Laboratory
1, place Jean–Baptiste Clément
93881 Noisy-le-Grand, France.

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

Table des matières

Notices . . . . . **v**

Marques . . . . . **vii**

Chapitre 1. Utilitaires **1**
Programme d'accès aux constantes Pacbase . . 1
Plateforme IBM/MVS 1
Plateforme BULL/GCOS7 3
Plateforme Windows/NT 4
Plateformes UNIX 5
MSP pour dates,T.U.F, et libellé d'erreurs. . . 7
Interface de sécurité Top Secret (IBM/MVS) . . 8
RACF et AD-WORKENCH : EXIT USER SECURITY (IBM/MVS) 9
Présentation 9
Mise en oeuvre 10
Mise à jour du catalogue DB2 (IBM/MVS) . . 12
Aide à la reprise 2.5 => 3.0 /3.5 13
Procédure UTAG (2.5) 13
Procédure UTFG (2.5) 14
Procédure UTSD (2.5) 15
Compléments pour l'amélioration 20874 (modification de génération de l'opérateur UNS) 15
Procédure UTU1 17
Procédure UTU2 17

Chapitre 2. P.A.F. - P.U.F. - P.Q.C. **19**
Description des tables P.A.F 19
EXEMPLES D'UTILISATION. 20
PLATEFORME IBM/MVS-CICS. 20
Exemple d'application : PAF-TP et P.U.F. 20
P.Q.C. 21

Chapitre 3. Gestionnaire multi-écrans (Zar980). **23**
Généralités 23
Micro Focus DOS. 23
Fonctionnalités 24
Micro Focus Unix. 28
Présentation du lot XP250V02_UIX . . . 28
Modes de Compilation 30
Configuration du poste 32
Jeu d'essai 41
Compaq VMS. 48
Architecture des Applications 48
Mise en oeuvre 49
Bull Gcos7. 53
Bull Gcos8. 54
MVS/CICS. 54
MVS/IMS. 55
HP3000. 55
UNISYS-A. 55
ICL. 56

Chapitre 4. Compléments A&D Workbench 57
OPEN JADE & TIDY 57

Chapitre 5. Annexes **59**
Transferts de fichiers 59
Exécutables IBM/MVS 59
C.U. BULL/GCOS7 59
Fichiers textes 59

Notices

Ce document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM. Cela ne signifie pas qu'IBM ait l'intention de les annoncer dans tous les pays où la compagnie est présente. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante : IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk NY 10504–1785, U.S.A.

Les détenteurs de licences du présent produit souhaitant obtenir des informations sur celui-ci à des fins : (i) d'échange d'informations entre des programmes développés indépendamment et d'autres programmes (y compris celui-ci) et (ii) d'utilisation mutuelle des informations ainsi échangées doivent s'adresser à : IBM Paris Laboratory, SMC Department, 1 place J.B.Clément, 93881 Noisy-Le-Grand Cedex, France. De telles informations peuvent être mises à la disposition du Client et seront soumises aux termes et conditions appropriés, y compris dans certains cas au paiement d'une redevance.

IBM peut modifier ce document, le produit qu'il décrit ou les deux.

Marques

IBM est une marque d'International Business Machines Corporation, Inc. AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, VisualAge Pacbase, RACF, RS/6000, SQL/DS et VisualAge sont des marques d'International Business Machines Corporation, Inc. dans certains pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.

UNIX est une marque enregistrée aux Etats-Unis et/ou dans d'autres pays et utilisée avec l'autorisation exclusive de la société X/Open Company Limited.

D'autres sociétés peuvent être propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

Chapitre 1. Utilitaires

Package : util.zip

Programme d'accès aux constantes Pacbase

Cet utilitaire permet d'extraire les valeurs des Constantes Pacbase présentes dans les programmes générés.

Version :

Valable toutes versions.

Plateformes :

IBM MVS, BULL GCOS7, WINDOWS/NT, UNIX

Plateforme IBM/MVS

Nature des composants

Exécutable et Jcl

Liste des composants :

\mvs\bvpcst_mvs : sous-programme exécutable

\mvs\bvpcst_mvs_jcl.rtf : exemple de JCL d'exécution

Mise en oeuvre :

Transférer l'exécutable en binaire dans une bibliothèque de load-module.

Insérer l'appel du sous-programme dans un programme.

Exemple d'appel dans un programme

TRAITEMENTS PROGRAMME P UTICST recherche constante pac

```
OPE                                NVTY CONDITION
N   RECHERCHE CONSTANTE PAC        05BL
M   SPACE        LK04
M   MB00-COCAR   LK04-NAPRO
M   MB00-SUITE   LK04-PROGE
```

```
CAL 'BVPCST'  USING LK04
MES LK04-PROGE ' : ERREUR  >>>>> '   99IT LK04-CODERE
     LK04-CODERE                          NOT = ZERO
MES LK04-PROGE ' : ' LK04-ZGCST      99EL
```

Description de la zone de communication :

```
 01    LK04.
  10          XENTRE
   11         NAPRO        X     (B=Batch,T=TP)
   11         PROGE        X(8)
  10          XSORTI
   11         CODERE       X     (0=OK, 6=non trouvé)
   11         ZGCST
    12        SESSI        X(5)
    12        APPLI        XXX
    12        DAT8G        X(8)
    12        PROGR        X(6)
    12        CODUTI       X(8)
    12        TIMGN        X(8)
    12        CPCOB1       X(8)
    12        CTRAN        X(4)
    12        DATGNC       X(10)
```

Exemple de Jcl d'exécution :

Remonter le jcl dans une bibliothèque de sources :

```
 //UTICST EXEC PGM=UTICST
 //STEPLIB  DD DSN=PT$PDV.EXPL20.LOAD,DISP=SHR
 //DFHRPL   DD DSN=PT$PST.PB250.MBR8,DISP=SHR
 //SYSOUT   DD SYSOUT=H
 //SYSUDUMP DD SYSOUT=H
 //SYSABOUT DD SYSOUT=H
 //PAC7MB   DD *
   PACG3C
   PACG3S
   PACG4S
   PACG8C
   PACG8S
 /*
```

Plateforme BULL/GCOS7

Nature des composants :

Sources JCL

Liste des composants :

\gcos7\cu-utixcu : JCL d'installation

\gcos7\paccste : exemple de JCL d'exécution

SPECIFICITES GCOS7 :

Le module UTIXCU permet d'extraire les constantes Pacbase pour des Compil-Units ou des Load-Modules.

Les Share-Modules ne sont pas traités.

LIMITES :

Pour les CU ou LM modules, quelques cas ne fonctionnent pas (moins de 1%), dus à une césure d'enregistrement.

Pour un LM, UTIXCU considère toujours que le premier programme du fichier est le programme principal. Les suivants sont considérés comme les sous-programmes et présentés avec une indentation.

Mise en oeuvre :

Transférer les jcls dans une bibliothèque de sources

Lancer le jcl d'installation qui crée le programme UTIXCU dans la Bibliothèque de Compil-Unit renseignée dans le paramètre 1.

Créer le load module UTIXCU avec l'Editeur de Liens du site.

Exemple de Jcl d'exécution (PACCSTE)

Paramètre 1: nom du Compil-unit à extraire (ou * pour tous)

Paramètre 2: Bibliothèque à explorer

Paramètre 3: Type de la bibliothèque à explorer : CU ou LM

Paramètre 4: Bibliothèque de Load-modules contenant UTIXCU

```
COMM '***** PRINT PACBASE CONSTANTS *****';

COMM ' Parameters: ';

COMM ' 1: Selected CU - star convention allowed';

COMM ' 2: program library to explore';

COMM ' 3: program library type: CU or LM';

COMM ' 4: LM library containing utility UTIXCU';

VL *,SRCHLIB,CU,LMLIB;

LIB &3 IL1=&2;

LMN &3 OUTFILE=(TCHIK,TEMPRY,END=PASS),

COM='MV IL1:'&1';',

PRTFILE=DUMMY;

STEP UTIXCU FILE=&4;

ASG BV TCHIK,TEMPRY,END=DEASSIGN;

ASG BZ SYS.OUT;

ESTP;
```

REMARQUE :

Si on veut récupérer les mouvements (BZ) dans un fichier, les caractéristiques sont les suivantes :

80 caractères, fixe, bloqué.

Plateforme Windows/NT

Nature des composants :

Exécutable, Script Visual Basic, Fichier de Commandes

Liste des composants :

\wnt\bvpdate.exe : exécutable

\wnt\procinsl.cmd : exemple de Fichier de commande (version pacbase < 3.0)

\wnt\insl.vbs : exemple de Script Visual Basic (version pacbase >= 3.0)

Principe de fonctionnement :

Les exécutables à explorer, avec leur chemin d'accès complet sont fournis sous forme de liste dans un fichier.

La localisation de ce fichier est fournie à l'utilitaire par l'intermédiaire d'une variable d'environnement de nom ″ LIST ″.

L'utilitaire prend pour argument le nom du fichier compte-rendu d'extraction.

ex : set LIST= ListCob.txt

Bvpdate Report.txt

Mise en oeuvre :

- Environnement Pacbase < 3.0

  Transférer en mode binaire, le programme bvpdate dans le répertoire ″...\batch\pgm ″ d'une installation Pacbase.

  Transférer, la procédure de lancement PROCINSL.CMD dans le répertoire ″...\batch\proc ″ .

  Lancer le fichier de commandes ″ PROCINSL.CMD ″ pour l'extraction des constantes Pacbase.

  Le fichier compte rendu d'extraction se trouve dans le répertoire ″...\tmp″.
- Environnement Pacbase à partir de 3.0

  Transférer en mode binaire, le programme bvpdate dans le répertoire ″...\server\sys\pgm ″ d'une installation Pacbase.

  Transférer, la procédure de lancement INSL.VBS dans le répertoire ″...\system\proc ″ .

  Lancer le script ″ INSL.VBS ″ pour l'extraction des constantes Pacbase.

  Le fichier compte rendu d'extraction se trouve dans le répertoire ″...\Server\Config ″.

Plateformes UNIX

Nature des composants :

Exécutables, Shells

Liste des composants :

\unix\procdate : exemple de Shell de lancement Pacbase < 3.0

\unix\insl : exemple de Shell de lancement Pacbase >= 3.0

\unix\bvpdate_aix : exécutable système aix

\unix\bvpdate_sunos: exécutable système SunOS

\unix\bvpdate_osf1 : exécutable système OSF1

\unix\bvpdate_hpux : exécutable système HP-UX

\unix\bvpdatelinux : exécutable système Linux

Principe de fonctionnement :

Les exécutables à explorer, avec leur chemin d'accès complet sont fournis sous forme de liste dans un fichier.

La localisation de ce fichier est fournie à l'utilitaire par l'intermédiaire d'une variable d'environnement de nom ″ LIST ″.

L'utilitaire prend pour argument le nom du fichier compte-rendu d'extraction.

ex :

LIST= ListCob.txt

export LIST

bvpdate Report.txt

Mise en oeuvre pour extraire les constantes

d'une version VisualAge Pacbase :

- Environnement Pacbase < 3.0

  Transférer en mode binaire, le programme bvpdate dans le répertoire ″$PACDIR\bin″ de l'installation Pacbase.

  Transférer, le shell de lancement PROCDATE dans le répertoire ″ $PACDIR\batch\proc ″.

  Vérifier que l'exécutable ″ bvpdate ″ et le shell de lancement disposent bien des droits UNIX d'exécution, au besoin ajouter ces droits par la commande UNIX :

  ″ chmod u+x bvpdate ″ pour l'exécutable par exemple.

  Lancer le shell de lancement ″ PROCDATE ″ pour l'extraction des constantes Pacbase.

Le fichier compte rendu d'extraction se trouve dans le répertoire " $PACDIR\tmp ".

- Environnement Pacbase à partir de 3.0

  Transférer en mode binaire, le programme bvpdate dans le répertoire " $PACDIR\system\bin " de l'installation Pacbase.

  Transférer le shell de lancement INSL dans le répertoire "$PACDIR\system\proc " .

  Lancer le shell de lancement " INSL " pour l'extraction des constantes Pacbase.

  Le fichier compte rendu d'extraction se trouve dans le répertoire "\$PACDIR\Config ".

MSP pour dates,T.U.F, et libellé d'erreurs

Nature des composants : source Pacbase

Liste des composants : \fr\bvputifr

Les mouvements permettent de charger les programmes et MSP suivants :

Macro-Structures Controle de Dates

P AADS10

Macro-Structures Pactables (TUF)

P AATUFA : Description Rubrique

P AATUFL : Liste LT ou LH

P AATUFS : Liste LS ou LC

P AATUFX : Liste des postes

Libellés d'erreur Utilisateur (Dialogue)

P UTEMLD : Indexation Fichier Libellé d'erreurs utilisateur

Plateformes

Valable toutes plateformes.

Version

3.0

Mise en oeuvre

Remonter le fichier texte sur le site cental. Lancer la procédure UPDT avec ce fichier en entrée , après adaptation de la carte d'identification de l'utilisateur.

Interface de sécurité Top Secret (IBM/MVS)

Sous-programmes Interface de Sécurité Top-Secret

Ces sous-programmes permettent la mise sous contrôle de la base VisualAge Pacbase par le système de sécurité TOP-SECRET.

Plateforme

IBM MVS/CICS

Version

3.0

Nature des composants :

Sources COBOL

Liste des composants :

\mvs\bvptss : sous-programme interface batch

\mvs\bvptssc : sous-programme interface TP

Mise en oeuvre :

- Remonter les sources sur le site central
- BVPTSS et BVPTSSC doivent être compilés avec la bibliothèque TSS ″OPMAT″ dans la ligne SYSLIB de la compilation Assembleur.
- BVPTSSC est un programme CICS et doit être translaté avant compilation et link-edit.
- BVPTSSC et le programme TSSCAI (Computer Associates) doivent être déclarés dans la CSD de CICS et se trouver dans une bibliothèque de load-modules de la DFHRPL.
- Définition de la classe de ressources :

  TSS ADD (RDT) RESCLASS(cccc) RESCODE(xx)

  cccc = code de la classe de ressources correspondant à VA Pacbase

  xx = code hexadécimal qui indique le type de ressource
- Création des ressources

TSS ADD(nom-dept) cccc(nbib) cccc(nbib) ...

nom-dept = nom du département

n = niveau d'autorisation

bib = code bibliothèque

- Définition des autorisations d'accès

  TSS PERMIT(code-utilisateur) cccc(nbib)

  TSS PERMIT(code-utilisateur) cccc(nbib)

RACF et AD-WORKENCH : EXIT USER SECURITY (IBM/MVS)

Programme EXIT USER SECURITY pour une base VAPacbase sous contrôle RACF.

Ce programme est livré comme exemple de contrôle de la validité de la connexion d'un utilisateur depuis le WORKBENCH, lorsque la base serveur MVS/CICS est sous contrôle RACF et que le protocole de communication utilisé est CICS SOCKET TCP/IP.

Plateforme :

IBM MVS/CICS (communication TCP/IP Socket)

Version :

3.0

Présentation

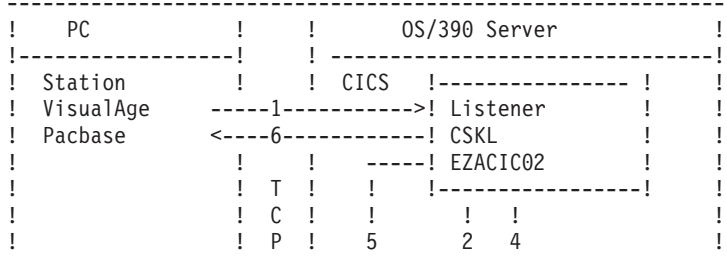
Documentations de référence IBM :

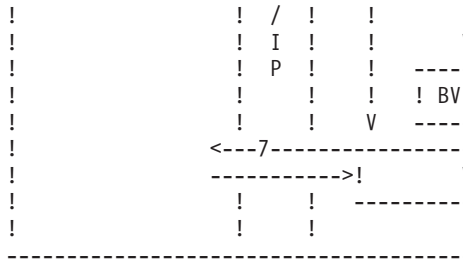
OS/390 SecureWay Communications Server

IP CICS Sockets Guide

6.6.3 Writing Your Own Security Link Module for the Listener

SCHEMA DE L'ARCHITECTURE





DESCRIPTION DES ECHANGES :

1: L'Application demande une connexion avec un code utilisateur, un mot de passe, et une transaction.

2: EZACIC02 lance BVPCICSE avec le code utilisateur /mot de passe.

3: BVPCICSE appelle RACF pour vérifier l'autorisation

4: BVPCICSE retourne au Listener

5: Rejet du message si RACF l'a signalé invalide

6: Démarrage de la transaction sous le code utilisateur vérifié par RACF (pas de mot de passe,le code utilisateur est valorisé à 'CICSDFLT')

7: Plusieurs transactions tournent en mode conversationnel.

Mise en oeuvre

Nature des composants :

Load module TP.

Liste des composants :

\mvs\bvpcicse : programme TP

Mise en oeuvre :

- Remonter l'exécutable sur le site central dans la bibliothèque de load modules TP.
- Mettre à jour la CSD CICS (clause Execkey):
  - Par job batch : DFHCSDUP

```
DEFINE PROGRAM(BVPCICSE) GROUP(XXXXXX)
DESCRIPTION(Cics socket exit program)
    LANGUAGE(COBOL) RELOAD(NO) RESIDENT(NO) USAGE(NORMAL)
    USELPACOPY(NO) STATUS(ENABLED) CEDF(YES) DATALOCATION(Y)
    EXECKEY(CICS) EXECUTIONSET(FULLAPI)
```

- ou Par la transaction CEDA

```
CEDA  AL PROG(BVPCICSE) GR(PB300)
  OVERTYPE TO MODIFY
   CEDA  ALter PROGram( BVPCICSE )
    PROGram        : BVPCICSE
    Group          : PB300
    DEscription  ==>
    Language     ==>                  CObol ! Assembler !
    RELoad       ==> No               No ! Yes
    RESident     ==> No               No ! Yes
    USAge        ==> Normal           Normal ! Transient
    USElpacopy   ==> No               No ! Yes
    Status       ==> Enabled          Enabled ! Disabled
    RSl            : 00               0-24 ! Public
    CEdf         ==> Yes              Yes ! No
    DAtalocation ==> Below            Below ! Any
    EXECKey      ==> Cics             User ! Cics
    COncurrency  ==> Quasirent        Quasirent ! Threadsa
   REMOTE ATTRIBUTES
    DYnamic      ==> No               No ! Yes
 +  REMOTESystem ==>
```

- Paramétrer le listener CICS (clause SECEXIT)

  Sous CICS, saisir la commande 'EZAC AL'

```
EZAC,ALter
ENTER ONE OF THE FOLLOWING

CICS         ===>                     Enter Yes!No
LISTENER     ===>  Y                  Enter Yes!No
```

Aprés avoir sélectionné le Listener:

```
EZAC,ALter,LISTENER
 ENTER ALL FIELDS

APPLID       ===> A6ECCSXP    APPLID of CICS Syste
NAME         ===> CSKL        TRANSACTION NAME OF LISTENER
```

On obtient l'écran suivant dans lequel il faut préciser le code de l'exit-user :

```
EZAC,ALter,LISTENER
OVERTYPE TO ENTER

APPLID       ===> A6ECCSXP            APPLID of CICS Syste
TRANID       ===> CSKL                Transaction Name of
PORT         ===> 09957               Port Number of Liste
IMMEDIATE    ===> YES                 Immediate Startup
BACKLOG      ===> 040                 Backlog Value for Li
NUMSOCK      ===> 100                 Number of Sockets in
MINMSGL      ===> 004                 Minimum Message Leng
ACCTIME      ===> 030                 Timeout Value for AC
GIVTIME      ===> 010                 Timeout Value for GI
REATIME      ===> 300                 Timeout Value for RE
FASTRD       ===> YES                 Read Immediately
TRANTRN      ===> YES                 Translate TRNID
```

```
TRANUSR      ===> YES                  Translate User Data
SECEXIT      ===> BVPCICSE             Name of Security Ex
WLM groups   ===> CICSSEXP     ===>              ==
```

Mise à jour du catalogue DB2 (IBM/MVS)

Nature des composants :

Source programme COBOL

Liste des composants :

\mvs\bvprs12

Application interactive SQL sous VA Pacbase

Lors du choix B.....GN, le système génère le SQL de mise à niveau et appelle le programme BVPRS12. Le module BVPRS12, est disponible sous forme de programme source.

Plateforme :

IBM MVS/DB2

Version :

3.0

Mise en oeuvre :

Remonter le source COBOL sur le site central

Le module BVPRS12, livré sous forme de programme source Cobol doit être préparé et installé comme tout programme DB2 classique :

- Préparation par préprocesseur donnant le source Cobol et les DBRM (de même nom que le source Cobol) : xxRS12 xx représente le paramètre $ROOT radical des programmes T.P.
- Compilation et link-edit du source Cobol donnant le programme exécutable (xxRS12).
- Le plan est construit avec le DBRM, à l'aide de l' application TSO/DB2I (c'est l'opération du BIND),
- La table RCT de CICS est mise à jour avec le code de la transaction VA Pac et le code du plan précédemment construit.

Aide à la reprise 2.5 => 3.0 /3.5

Nature des composants : exécutables et Jcls

Liste des composants :

\mvs\utagfgsd.bin : éxécutables bvptu570 bvptufgu bvputsd

\mvs\utagfgsd.jcl : procédures UTAG UTFG UTSD

Plateformes

MVS/CICS

Version

2.5

Mise en oeuvre

Remonter les load-modules dans la bibliothèque des load- modules batch (voir annexe)

Remonter le fichier .jcl dans un fichier séquentiel ou un membre de bibliothèque.

Dupliquer le jcl d'installation effectuant l'éxécution du programme de paramétrage MM1JCL.

Renseigner en SYSUT1 le fichier .jcl remonté

Renseigner en SYSUT2 un fichier séquentiel ou un membre de bibliothèque qui recevra le JCL paramétré ($prefjD5P)

Exécuter MM1JCL.

Exécuter ensuite le jcl $prefjD5P obtenu: les procédures $prefjUTAG, $prefjUTFG, $prefjUTSD, seront crées dans la bibliothèque des procédures.

Procédure UTAG (2.5)

Principe

Cette procédure est un utilitaire qui permet de constituer la liste des entrées de la procédure REAG à partir de l'image séquentielle des commandes d' édition-génération.

Les utilisateurs inexistants dans le fichier AE ,les bibliothèques et sessions inexistantes dans la base sont recherchés. Des lignes d'épuration des commandes de ces utilisateurs, bibliothèques ou sessions sont constituées.

Condition d'exécution

Aucune

Résultat obtenu

Un fichier mouvement est créé qui constitue la liste des entrées utilisateurs de la procédure REAG.

Une ligne '*' avec code utilisateur et mot de passe .

Si la carte * n'est pas renseignée, un message d'anomalie est émis et la procédure ne peut s'effectuer.

Procédure UTFG (2.5)

Principe

Cette procédure est un utilitaire qui permet d' attribuer aux types des formats guide d'une base 2.5 les valeurs utilisées dans une base 3.0.

L'extraction des fiches des formats guide s'effectue dans les sessions précisées sur les cartes en entrée. Chaque format guide est typé avec la valeur rencontrée sur son descriptif. Le type pourra prendre les valeurs 'G'(complément de génération), 'O' (option de dialogue) ou 'C'(commentaires). Le type 'C' est attribué pour toutes les valeurs différentes de 'O' ou 'G'. Si sur un même descriptif, des types différents ont été saisis, un message d'incohérence apparait dans l'état avec une demande d'intervention manuelle.

Condition d'exécution

Aucune

Edition obtenue

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

Résultat obtenu

Un fichier mouvement est créé qui sera mis à jour par la procédure UPDT dans la base 2.5.

Pas de ligne spécifique à cet extracteur, mais autant de lignes '*' que de sessions de la base à extraire.

Si la carte * n'est pas renseignée, un message d'anomalie est émis et la procédure ne peut s'effectuer.

Procédure UTSD (2.5)

Principe

Cette procédure est un utilitaire qui permet de constituer un fichier mouvement pour la procédure UPDT afin d'associer un mot-clé à un type de structure de données.

Condition d'exécution

Aucune

Résultat obtenu

Un fichier mouvement est créé qui constitue la liste des entrées utilisateurs de la procédure UPDT.

Une ligne '*' avec code utilisateur et mot de passe .

Une ligne 'G' suivie du type de structure de donnée et du mot-clé associé à ce type (sur 13 caractères). Il y a autant de lignes 'G' que nécessaire.

Si la carte * n'est pas renseignée, un message d'anomalie est émis et la procédure ne peut s'effectuer.

Compléments pour l'amélioration 20874 (modification de génération de l'opérateur UNS)

OBJET :

Avant le change 20874, un '.' était systématiquement généré à la fin de la ligne précédant un ordre unstring. Ceci provoquait une erreur si l'ordre UNS était sur une ligne 99EL.

Ceci a été corrigé (C 20874).

Certains utilisateurs ont utilisé cette anomalie de génération : ils n'ont pas mis de 99BL sur une ligne comportant un UNS et suivant un conditionnement car la ligne se comportait comme si on avait un 99BL). Avec la correction du C20874, ce type de ligne se retrouve sous le conditionnement alors qu'elle ne l'était pas avant.

Des utilitaires ont donc été créés pour détecter ce genre de lignes pour que l'utilisateur puisse les corriger éventuellement.

Cette opération se déroule en 2 étapes.

.1ère : procédure : UTU1

Extraction des lignes de traitement spécifiques (P) contenant l'opérateur 'UNS' et n'ayant aucune information dans niveau-conditionnement.

L'utilisateur vérifiera si des lignes ne sont pas à garder en l'état, et les supprimera du fichier résultat. Les enregistrements résultats sont définis comme suit : AAA BBBBBBBB CCCCCC DD EE FFF AAA = appli, BBBBBBBB = session , CCCCCC = code programme , DD = code fonction , EE = code sous-fonction , FF = n° ligne

.2ème : procédure : UTU2 Pour toutes les lignes du fichier précédant, on forcera '99BL' dans la zone niveau-condition.

Nature des composants : exécutables et Jcls

Liste des composants :

\mvs\uns1uns2.bin : éxécutables bvptuns1 bvptuns2

\mvs\utu1utu2.jcl : procédures UTU1 UTU2

Plateformes

MVS/CICS

Version

3.0

Mise en oeuvre

Remonter les load-modules dans la bibliothèque des load- modules batch (voir annexe)

Remonter le fichier .jcl dans un fichier séquentiel ou un membre de bibliothèque.

Dupliquer le jcl d'installation effectuant l'éxécution du programme de paramétrage MM1JCL.

Renseigner en SYSUT1 le fichier .jcl remonté

Renseigner en SYSUT2 un fichier séquentiel ou un membre de bibliothèque qui recevra le JCL paramétré ($prefjD5P)

Exécuter MM1JCL.

Exécuter ensuite le jcl $prefjD5P obtenu: les procédures $prefjUTU1, $prefjUTU2 seront crées dans la bibliothèque des procédures.

Procédure UTU1

Principe

Cette procédure est un utilitaire qui permet de d'extraire les lignes 'P' des programmes contenant l' opérateur 'UNS' et n'ayant aucune information dans niveau- conditionnement. L'utilisateur vérifiera le fichier en sortie. Il devra supprimer de ce fichier toutes les lignes à garder en l'état. Pour toutes les lignes conservées dans le fichier, la zone niveau-conditionnement sera forcée à '99BL' si on exécute la procédure UTU2.

Condition d'exécution

Mettre le nom du fichier résultat des lignes à pointer dans le jcl de lancement : NOMUT='... '

Résultat obtenu

Un fichier des lignes 'P' UNS à pointer --> NOMUT

Pas d'entrées utilisateur.

Procédure UTU2

Principe

Cette procédure est un utilitaire qui permet de mettre à jour les enregistrements extraits par la procédure UTU1. Pour toutes les lignes du fichier en entrée, on force '99BL' dans la zone niveau-conditionnement.

Condition d'exécution

Exécuter auparavant la procédure UTU1.

Mettre le nom du fichier résultat de la procédure UTU1 dans le jcl de lancement : NOMUT='... '

Résultat obtenu

Nouvelle image séquentielle du réseau.

Pas d'entrées utilisateur.

Chapitre 2. P.A.F. - P.U.F. - P.Q.C.

Package : Paf.zip

Le module Pacbase Access Facility est soumis à droit d' acquisition. Il permet d'accéder par programme ou requête utilisateur aux données de la base VisualAge Pacbase, en batch ou en T.P.

Version :

3.0

Plateformes :

Toutes plateformes.

Documentations de référence :

Manuel de référence PAF

Description des tables PAF

Description des tables P.A.F

Nature des composants

mouvements Pacbase UPDT

Liste des composants

\fr\bvppaffr : description des tables PAF - français

Ce fichier contient la description des segments utilisés pour décrire les entités du référentiel Pacbase.

Mise en oeuvre :

- Remonter le fichier texte sur le site cental.
- Lancer la procédure UPDT avec ce fichier en entrée , après adaptation de la carte d'identification de l' utilisateur.

REMARQUE : il est conseillé de charger les mouvements dans une bibliothèque centrale indépendante.

Ils peuvent être installés dans une base autre que celle à explorer.

EXEMPLES D'UTILISATION.

PLATEFORME IBM/MVS-CICS.

Exemple d'application : PAF-TP et P.U.F.
Ceci est un exemple de programmes transactionnels mettant en oeuvre PAF et PUF TP.

Version :

3.0

Plateformes :

Toutes plateformes.

Plateforme cible :

MVS-CICS

Nature des composants

mouvements Pacbase UPDP

Liste des composants

mvs-cics\pufext : exemple d'application transactionnelle PUF

Mise en oeuvre :

- Remonter le fichier texte sur le site cental.
- Lancer la procédure UPDP avec ce fichier en entrée , après adaptation de la carte d'identification de l' utilisateur (carte ASSIGN)

  REMARQUE : Les mouvements doivent être chargés dans une bibliothèque applicative sous celle contenant le référentiel (tables PAF).
- Générer les écrans PUINIT, PU0200, PU0300, PU3ERR
- Les préprocesser (procédure PPAF)
- Les compiler/ link-editer.
- Déclarer les programmes et les transaction au CICS.

L'application est composée de :

Ecran ″Mire d'accueil PUF″ (PUINIT)

- Cet écran sert à saisir le contexte de connexion à la base : Code utilisateur/ mot de passe, code de la transaction Va Pac, bibliothèque, session...
- Il sert aussi de menu :

  La touche fonction F2 permet d'accéder à la liste des programmes, la touche F3 à la liste des écrans.

Ecran ″Liste des programmes″ (PU0200)

- Cet écran permet de consulter et mettre à jour la liste des programmes. Il n'y a pas de positionnement dans la liste.
- Mise à jour :

  Saisir un code action sur les occurences et modifier les données voulues : cela déclenche l'ordre PUF INSERT qui mets à jour le fichier de travail SYSPAF

  Lorsque l'on veut déclencher la mise à jour de la base, saisir 'O' dans la zone de bas d'écran : cela déclenche l'ordre PUF CALPUF .

  Si une / des erreurs sont détectées lors de l'étape de mise à jour de la base, on se débranche sur un écran dédié à la visualisation des erreurs PUF : PU3ERR

Ecran ″Liste des écrans″ (PU0300)

- Cet écran permet de consulter et mettre à jour la liste des ecrans : mêmes fonctionnalités que la liste des programmes.

Ecran de ″Liste des erreurs PUF″ (PU3ERR)

- Cet écran permet de consulter la liste des erreurs de mise à jour de la base.

P.Q.C.

Le module Pacbase Quality Control est soumis à droit d' acquisition. PACBENCH QUALITY CONTROL est un module destiné à évaluer et contrôler la qualité des applications développées à l'aide de VisualAge Pacbase. Le fichier suivant est nécessaire quand on a acquis ce module avec l'option 'personnalisation'.

Version :

3.0

Plateformes :

Toutes plateformes.

Documentations de référence :

Manuel de référence PQC

Nature des composants

mouvements Pacbase UPDT

Liste des composants

\fr\bvpqcfr : source des règles PQC - français

Ce fichier contient la description de la Meta-Entité utilisée pour décrire les règles de qualité, ainsi que les Entités Utilisateurs décrivant les règles standard.

Mise en oeuvre :

- Remonter le fichier texte sur le site cental.
- Lancer la procédure UPDT avec ce fichier en entrée , après adaptation de la carte d'identification de l' utilisateur.

  REMARQUE : il est conseillé de charger les mouvements dans une bibliothèque centrale indépendante.

  Ils peuvent être installés dans une base autre que celle à controler.

Chapitre 3. Gestionnaire multi-écrans (Zar980).

Généralités

Package : multi_screen.zip

Ces utilitaires permettent d'utiliser la variante Multi-ecrans soit pour le module Dialogue Standard, soit pour le module Pacbase Web Connection.

Le programme de mise en forme du message est ZAR980 .

Un ensemble d'outils est livré pour chaque cible de génération :

Bull-Gcos7, Bull-Gcos8, Compaq-VMS, HP3000, IBM-CICS, ICL, IBM-IMS, Microfocus-DOS, Unisys-A series, Microfocus-Unix.

Version :

Valable toutes versions.

Plateformes :

Ces composants sont indépendants de la plateforme de développement, car ils concernent les applications générées.

Documentations de référence :

Manuel de référence DIALOGUE

Guide du développeur Pabase WEB Connection

Guide du développeur PAW

Micro Focus DOS.

Nature des composants :

Sources COBOL

Liste des composants :

zarmf1 : source programme cobol ZAR980

scrcodif : source sous-programme gestion clavier

scriopar : source sous-programme gestion clavier

scrpeint : source sous-programme gestion clavier

scrsaisi : source sous-programme gestion clavier

Mise en oeuvre :

Remonter les fichiers texte sur le serveur.

Tous ces programmes doivent être compilés (sous forme de .GNT ou de .EXE pour MS/DOS) avec les options de compilation suivantes obligatoires :

ASSIGN ″EXTERNAL″

SEQUENTIAL ″LINE″

NOIBMCOMP

Fonctionnalités

LE SOUS-PROGRAMME ZAR980

Ce sous-programme prend en charge :

- La simulation d'un écran synchrone :
  - saisie pleine page (tabulation, gestion du curseur),
  - transmission du message par utilisation de touches (<ENTREE>, touches fonction).
- La gestion des couleurs, en particulier celle du fond de l'écran.

FONCTIONS EMULEES

Transmission :

```
-------------------------------------------------------------
!MNEMONIQUE !  DESCRIPTION               ! TOUCHE CLAVIER !
-------------------------------------------------------------
!           !                            !                !
! ENTER     ! Equivalent TRANSMIT        ! Ctrl-CR        !
!           !                            !                !
! CLEAR     ! Effacement écran           ! Alt-F10        !
!           !                            !                !
! PA1       ! Non utilisé                ! Alt-F01        !
!           !                            !                !
! PA2       !  -   -   -                 ! Alt-F02        !
!           !                            !                !
! PA3       !  -   -   -                 ! Alt-F03        !
```

```
!           !                              !                !
! PF1...PF10! Touches fonctions            ! F01...F10      !
!PF11...PF20!                              ! Maj-F01...F10  !
!PF21...PF24!                              ! Ctrl-F01...F04 !
!           !                              !                !
-------------------------------------------------------------
```

Tabulation :

```
-------------------------------------------------------------
!MNEMONIQUE !  DESCRIPTION                ! TOUCHE CLAVIER !
-------------------------------------------------------------
!           !                              !                !
! TAB       ! Tabulation avant            ! TAB            !
!           !                              !                !
! BACKTAB   ! Tabulation arrière          ! Maj-TAB        !
!           !                              !                !
! NEWLINE   ! Zone suivante sur ligne     ! CR             !
!           ! suivante                     !                !
!           !                              !                !
-------------------------------------------------------------
```

Positionnement :

```
-------------------------------------------------------------
!MNEMONIQUE !  DESCRIPTION                ! TOUCHE CLAVIER !
-------------------------------------------------------------
!           !                              !                !
! HOME      ! Positionnement sur le pre-  ! HOME ou        !
!           ! mier champ saisissable      ! Ctrl-PGup      !
!           !                              !                !
! FIN       ! Positionnement sur le der-  ! FIN  ou        !
!           ! nier champ saisissable      ! Ctrl-PGdown    !
!           !                              !                !
! DEB-FLD   ! Positionnement sur le pre-  ! Ctrl- <--      !
!           ! mier caractère du champ     !                !
!           !                              !                !
! FIN-FLD   ! Positionnement sur le der-  ! Ctrl- -->      !
!           ! nier caractère du champ     !                !
!           !                              !                !
-------------------------------------------------------------
```

Déplacement :

```
-------------------------------------------------------------
!MNEMONIQUE !  DESCRIPTION                ! TOUCHE CLAVIER !
-------------------------------------------------------------
! HAUT      ! Déplacement vers le haut    !   î            !
!           !                              !                !
! BAS       ! Déplacement vers le bas     !   !            !
!           !                              !   V            !
!           !                              !                !
! GAUCHE    ! Déplacement vers la gauche  !  <--           !
!           !                              !                !
```

```
! DROITE    ! Déplacement vers la droite  !  -->          !
!           !                             !               !
-----------------------------------------------------------
```

Action :

```
-------------------------------------------------------------
!MNEMONIQUE !  DESCRIPTION               ! TOUCHE CLAVIER !
-------------------------------------------------------------
!           !                            !                !
! BACKSPACE ! Effacement du caractère pré-! BACKSPACE      !
!           ! cédent et recul du curseur  !                !
!           ! d'une position              !                !
!           !                            !                !
! INS       ! Insertion de caractères    ! INSERT         !
!           !                            !                !
! DEL       ! Annulation d'un caractère  ! DELETE         !
!           !                            !                !
! ERASE-EOF ! Effacement de fin de zone  ! Ctrl-FIN       !
!           !                            !                !
!ERASE-INPUT! Effacement de tous les     ! Ctrl-HOME      !
!           ! champs saisissables        !                !
!           !                            !                !
! RECOVER   ! Réaffiche l'écran obtenu à ! ESCAPE         !
!           ! l'entrée de la transaction  !                !
!           !                            !                !
-------------------------------------------------------------
```

GESTION DES COULEURS

Les valeurs prises par défaut pour la gestion des couleurs ainsi que certaines caractéristiques du clavier peuvent être modifiées, en créant un fichier paramètres de nom logique FPARAM. Ce fichier séquentiel sera lu au début de la transaction et les valeurs par défaut seront remplacées par celles trouvées dans le fichier.

Un fichier paramètres différent peut être créé pour chaque dialogue.

La structure de ce fichier est la suivante :

```
-------------------------------------------------------------
! Posi.! Long.! Description                 ! Valeurs     !
!------!------!-----------------------------!-------------!
!      !      !                             !             !
!  1   !  2   ! Code du dialogue            !             !
!      !      !                             !             !
!  3   !  1   ! Type d'écran :    Monochrome ! 'M' (défaut)!
!      !      !                      Couleur ! 'C'         !
!      !      !           Monochrome dégradé ! 'G'         !
!      !      !                             !             !
!  4   !  1   ! Couleur fond d'écran Blanc(*)! 'W' (défaut)!
!      !      !                             !             !
!  5   !  1   ! Couleur du pinceau :  Noir(*)! 'N' (défaut)!
!      !      !                             !             !
```

```
!  6   !  1   ! Couleur fond 25ème ligne :   !             !
!      !      !                      Blanc(*)! 'W' (défaut)!
!      !      !                              !             !
!  7   !  1   ! Couleur pinceau 25ème ligne  !             !
!      !      !                       Noir(*)! 'N' (défaut)!
!      !      !                              !             !
!  8   !  1   ! Effacement écran en début    !             !
!      !      ! d'affichage :            Non ! 'N' (défaut)!
!      !      ! (Affichage plus rapide   Oui ! 'O' ou 'Y'  !
!      !      !  si pas d'effacement,        !             !
!      !      !  les zones fixes n'étant     !             !
!      !      !  pas réaffichées)            !             !
!      !      !                              !             !
```

(*) : Valeurs des couleurs : Blanc = 'W', Noir = 'N',
Jaune = 'Y', Vert = 'G', Turquoise = 'T',
Bleu = 'B', Rouge = 'R', Rose = 'P'.

```
! Posi.! Long.! Description                  ! Valeurs     !
!------!------!-----------------------------!-------------!
!      !      !                              !             !
!  9   !   1  ! Retour automatique du chariot!             !
!      !      ! en fin de zone :         Oui ! 'O' ou 'Y'  !
!      !      !                              !  (Défaut)   !
!      !      !                          Non ! 'N'         !
!      !      !                              !             !
!  10  !   1  ! Retour automatique du chariot!             !
!      !      ! en fin de dernière zone sai- !             !
!      !      ! sissable :               Non ! 'N' (défaut)!
!      !      !                          Oui ! 'O' ou 'Y'  !
!      !      !                              !             !
!  11  !   1  ! En mode insertion, autorise  !             !
!      !      ! la perte du caractère en fin !             !
!      !      ! de zone s'il n'est pas à     !             !
!      !      ! blanc :                  Non ! 'N' (défaut)!
!      !      !                          Oui ! 'O' ou 'Y'  !
!      !      !                              !             !
!  12  !   1  ! Couleur d'affichage des zo-  !             !
!      !      ! nes dont l'attribut de pré-  !             !
!      !      ! sentation est "underlined" : !             !
!      !      !                       Rouge(*)! 'R' (défaut)!
!      !      !                              !             !
!  13  !   1  ! Saisie des caractères en     !             !
!      !      ! ASCII :                      !             !
!      !      !       jusqu'à la valeur 'FF'! 'Y' (défaut)!
!      !      !       jusqu'à la valeur '7F'! 'N'         !
!      !      !                              !             !
!  14  !  67  ! Zone non utilisée            !             !
!      !      !                              !             !
```

(*) : Valeurs des couleurs : Blanc = 'W', Noir = 'N',
Jaune = 'Y', Vert = 'G', Turquoise = 'T',
Bleu = 'B', Rouge = 'R', Rose = 'P'.

Micro Focus Unix.

Nature des composants :

Fichiers textes, Fichier compressé ISO

Liste des composants :

\xp250v02.uix\readme_F.v02: notice d'installation francaise

\xp250v02.uix\readme_E.v02: notice d'installation anglaise

\xp250v02.uix\zarbase.iso : fichier des composants compressé

\xp250v02.uix\zarinst.v02 : shell de lancement de l' installation

INSTALLATION :

Pour installer le lot sur une machine UNIX :

Décharger en mode binaire les fichiers suivants

ZARINST.V02 (zarinst.v02)

ZARBASE.ISO (zarbase.iso)

Lancer la procedure d'installation

sh zarinst.v02 -E (version Anglaise)

ou

sh zarinst.v02 -F (version Francaise)

Présentation du lot XP250V02_UIX

```
*********************************************************
                  GENERATION DIALOGUE MICRO FOCUS UNIX
   ENVIRONNEMENT : UIX
                            REFERENCE : XP250V02UIX
                              VERSION : V02
                                 DATE : 02/04/2002
 *********************************************************
```

Ce lot technique est composé pour l'essentiel de deux utilitaires éxécutables dans un environnement UNIX et dans le cadre d'applications générées par le module Dialogue en Cobol Micro Focus UNIX.

L'un permet la communication d'une application dialogue habillée par PAW; l'autre assure la gestion de l'écran et du clavier de terminal UNIX.

Le premier utilitaire se trouve sous le répertoire ZARPAW quant au second il est implanté sous le répertoire ZARTERM. Pour chaque utilitaire une documentation de mise en oeuvre est fournie :

- sous le répertoire ZARPAW le fichier zarpaw.doc,

- sous le répertoire ZARTERM le fichier zarterm.doc.

LISTE DES FICHIERS LIVRES SOUS LE REPERTOIRE ZARPAW :

Documentation : zarpaw.doc

Programme zar980 : servzar.c (source en langage C) zar980.c (" ")

pactypes.h (en-tete pour zar980)

scrstruc.h (" ")

zar980.h (" ") comp_zar980 (script de compilation)

LISTE DES FICHIERS LIVRES SOUS LE REPERTOIRE ZARTERM :

Documentation : zarterm.doc

Programme zar980 : zar980.c (source du gestionnaire d'affichage en langage C)

pacparam.h (en-tete pour zar980)

pactypes.h (" ")

scrstruc.h (" ")

zar980.h (" ")

comp_zar980 (script de compilation (en mode normal d'exploitation)

Tests du terminal :

touche.c (tests clavier : source c)

video.c (tests ecran : source c) Executables et objets : touche (tests clavier : exécutable)

video (tests ecran : exécutable)

zar980.o (gestionnaire d'affichage : objet)

Les exécutables et objets sont livrés pour les systèmes UNIX suivants :

AIX avec le COBOL Micro-Focus Server Express 2.0.10 dans le sous répertoire AIX/2010

HP-UX 10.20 avec le COBOL Micro-Focus HP 11.30 dans le sous répertoire HP-UX/113

IRIX avec le COBOL Micro-Focus 3.1.39 dans le sous répertoire IRIX

OSF1 4.0 avec le COBOL Micro-Focus 4.1.10 dans le sous répertoire OSF1/4110

OSF1 5.1 avec le COBOL Micro-Focus Server Express 2.0.11 dans le sous répertoire OSF1/2011

SunOS avec le COBOL Micro-Focus 4.0.20 dans le sous répertoire SunOS

Modes de Compilation

Mise en oeuvre

Compilation du programe C Zar980

Trois modes de compilation sont proposés, qui sont :

- le mode debug en vue d'utiliser le debogueur cobol ANIMATOR,
- le mode normal d'exploitation,
- le mode degradé, qui ne prend en compte aucune touche fonction

REMARQUE : dans les descriptions suivantes, <ansi flags> correspondent aux options permettant la compilation en mode C-ANSI.

Compilation Mode Debug

En mode debug, compilez zarterm.c avec la suppression de l' appel à l'éditeur de liens (conservation du fichier zarterm.o) :

cc <ansi flags> -c -D_MF zarterm.c

puis lancez les commandes :

cob -x -e "" zarterm.o -lcurses

mv zarterm zar980

Vous obtenez ainsi le fichier zar980, qui contient le Run Time System, le fichier zarterm.o, la bibliothèque curses.

Enfin, pour lancer l'exécution du programme COBOL faisant appel au zar980 en C, tapez les commandes suivantes :

COBSW=+A-F

export COBSW

zar980 <program>

Compilation Mode Exploitation

En mode normal d'exploitation compilez zarterm.c avec la suppression de l'appel à l'éditeur de liens (conservation du fichier zarterm.o) :

cc <ansi flags> -c -D_CURSES zarterm.c

puis lancez les commandes :

cob -x -e "" zarterm.o -lcurses

mv zarterm zar980

Vous obtenez ainsi le fichier zar980, qui contient le Run Time System, le fichier zarterm.o, la bibliothèque curses.

Enfin, pour lancer l'exécution du programme COBOL faisant appel au zar980 en C tapez, les commandes suivantes :

zar980 <program>

Compilation Mode Dégradé

En mode dégradé zarterm.c est compilé selon le même principe que pour les autres modes, mais en plus la commande contient une option de compilation (-D_NOKEYPAD) qui inhibe les touches fonction du zar980 :

cc <ansi flags> -c -D_CURSES -D_NOKEYPAD zarterm.c

puis lancez les commandes :

cob -x -e "" zarterm.o -lcurses

mv zarterm zar980

Vous obtenez ainsi le fichier zar980, qui contient le Run Time System, le fichier zarterm.o, la bibliothèque curses.

Enfin, pour lancer l'exécution du programme COBOL faisant appel au zar980 en C, tapez les commandes suivantes :

zar980 <program>

NOTE : Sous AIX, si la version du Run Time COBOL est supérieure a 3.2.20, vous devez assigner la variable d'environnement LIBPATH avant le lancement de la commande ″cob″.

Exemple : LIBPATH=/lib:/usr/lib:$COBDIR/coblib

NOTE : Sous IRIX, SINIX, SunOS ou OSF1, vous devez assigner la variable LIBPATH avant le lancement de la commande ″cob″.

Exemple : LD_LIBRARY_PATH=$COBDIR/coblib

Configuration du poste

CONFIGURATION du POSTE de TRAVAIL

Pour une utilisation correcte, il faut procéder à la configuration du terminal, du clavier, des touches fonctions et de l'écran (couleurs et attributs de présentation).

Configuration du terminal

Le programme zar980 en langage C a été écrit par IBM pour assurer la gestion de l'écran et du clavier dans les applications générées par le module Dialogue en Cobol Micro Focus Unix.

Le source C livré doit etre compilé puis linké soit avec les écrans Cobol générés soit avec le Run Time System Micro Focus COBOL/2.

Le programme zar980 utilise la bibliothèque Unix curses de gestion du clavier et de l'écran et est conforme à la norme XPG3 (il n'utilise que des fonctions décrites dans XPG3 : X/OPEN PORTABILITY GUIDE 3).

Ce source C peut également être compilé en vue d'utilisation du debogueur Cobol Micro Focus Animator pour l'application générée.

Dans ce cas, le programme utilise les routines de gestion du clavier et de l'écran fournies par Micro Focus et non les fonctions de la bibliothèque curses.

Configuration du clavier

L'objet de ce chapître est de vérifier que toutes les touches Fonction de votre clavier sont correctement configurées, puis de vous aider à modifier la configuration de votre terminal si nécessaire.

On appelle touche fonction une touche du clavier permettant l'exécution d'une fonction particulière, c'est-à-dire toutes les touches autres que celles permettant la saisie des caractères alphanumériques.

Présentation des touches fonction

Le programme zar980 prend en compte les touches fonction lorsqu'elles sont décrites dans une base de données Unix appelée terminfo.

La base de données terminfo contient la configuration d'un certain nombre de terminaux.

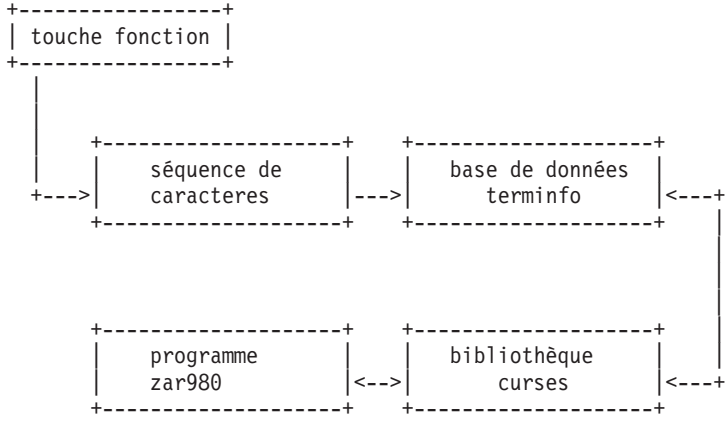
La configuration d'un terminal est constituée d'un ensemble d'attributs.

Chaque attribut se caractérise par un nom et une valeur; la valeur peut être :

- Un booleen indiquant que le type de terminal possède ou non la caractéristique correspondante;
- Un nombre entier spécifiant une caractéristique particulière du type de terminal;
- Une chaîne de caractères correspondant à la séquence de caractères a envoyer au terminal pour réaliser une fonction particulière.

Pour exploiter les configurations décrites dans terminfo, le programme zar980 utilise une bibliothèque de fonctions appelée curses.

Le schéma suivant résume l'enchaînement des opérations :



N.B. : Pour de plus amples informations sur terminfo tapez la commande ″man terminfo″ (si votre manuel d'utilisation en ligne contient des informations sur terminfo).

Le programme zar980 utilise les touches fonction suivantes :

nom d'attribut dans curses(X).h	**nom d'attribut dans terminfo**	**description/interprétation par le programme zar980**	
KEY_UP	kcuu1	flèche vers le haut déplace le curseur vers le haut de l'écran	(1)
KEY_DOWN	kcud1	flèche vers le bas déplace le curseur vers le bas de l'écran	(2)
KEY_LEFT	kcub1	flèche vers la gauche déplace le curseur vers la gauche de l'écran	(3)
KEY_RIGHT	kcuf1	flèche vers la droite déplace le curseur vers la droite de l'écran	(4)
KEY_HOMET	khome	flèche oblique en général positionne sur la premiere zone de saisie de l'écran	(5)
KEY_LL	kll	touche FIN en général positionne sur la dernière zone de saisie de l'écran	(6)
KEY_IC	kich1	touche INSER en général active/désactive mode insertion	(7)
KEY_DC	kdch1	touche SUPPR en général supprime un caractère	(8)
KEY_BACKSPACE	kbs	touche backspace supprime un caractère	(9)
KEY_ENTER	cud1	touche ENTER	(10)
KEY_F(0)	kf0	touche fonction associée à la fonction PF0 du zar980	(11)
KEY_F(1)	kf1	touche fonction associée à la fonction PF1 du zar980	(12)
KEY_F(2)	kf2	touche fonction associée à la fonction PF2 du zar980	(13)
KEY_F(3)	kf3	touche fonction associée à la fonction PF3 du zar980	(14)
KEY_F(4)	kf4	touche fonction associée à la fonction PF4 du zar980	(15)

nom d'attribut dans curses(X).h	nom d'attribut dans terminfo	description/interprétation par le programme zar980	
KEY_F(5)	kf5	touche fonction associée à la fonction PF5 du zar980	(16)
KEY_F(6)	kf6	touche fonction associée à la fonction PF6 du zar980	(17)
KEY_F(7)	kf7	touche fonction associée à la fonction PF7 du zar980	(18)
KEY_F(8)	kf8	touche fonction associée à la fonction PF8 du zar980	(19)
KEY_F(9)	kf9	touche fonction associée à la fonction PF9 du zar980	(20)
KEY_F(10)	kf10	touche fonction associée à la fonction PF10 du zar980	(21)
KEY_F(11)	kf11	touche fonction associée à la fonction PF11 du zar980	(22)
KEY_F(12)	kf12	touche fonction associée à la fonction PF12 du zar980	(23)
KEY_F(13)	kf13	touche fonction associée à la fonction PF13 du zar980	(24)
KEY_F(14)	kf14	touche fonction associée à la fonction PF14 du zar980	(25)
KEY_F(15)	kf15	touche fonction associée à la fonction PF15 du zar980	(26)
KEY_F(16)	kf16	touche fonction associée à la fonction PF16 du zar980	(27)
KEY_F(17)	kf17	touche fonction associée à la fonction PF17 du zar980	(28)
KEY_F(18)	kf18	touche fonction associée à la fonction PF18 du zar980	(29)
KEY_F(19)	kf19	touche fonction associée à la fonction PF19 du zar980	(30)
KEY_F(20)	kf20	touche fonction associée à la fonction PF20 du zar980	(31)
KEY_F(21)	kf21	touche fonction associée à la fonction PF21 du zar980	(32)
KEY_F(22)	kf22	touche fonction associée à la fonction PF22 du zar980	(33)
KEY_F(23)	kf23	touche fonction associée à la fonction PF23 du zar980	(34)

nom d'attribut dans curses(X).h	nom d'attribut dans terminfo	description/interprétation par le programme zar980	
KEY_F(24)	kf24	touche fonction associée à la fonction PF24 du zar980	(35)

**** TABLEAU 1 ****

NOTE : : Sur système COMPAQ TRUE 64 seules 10 touches fonction du type KEY_F(i) (i=0,...,24) peuvent être programmées.

Programmation des touches Fonction

Le programme 'touche', utilisable sur une large gamme de terminaux, a pour objet de vous aider à configurer votre terminal (source C et exécutable livrés).

Avant toute modification de la base de données terminfo, vous devez :

- sauvegarder le repertoire terminfo,
- connaître le fonctionnement de l'utilitaire de production de la base terminfo (commande UNIX tic en general),
- demander au constructeur du matériel :
  - quel type d'émulation de terminal est le mieux adapté à votre terminal (contenu de la variable d'environnement Unix TERM),
  - si votre terminal peut supporter l'ensemble des attributs utilisés par zar980 (voir TABLEAU 1),
  - la marche à suivre pour modifier la base de données,
  - la séquence de caractères envoyée par les touches fonctions

A noter : le programme touche peut avoir un comportement imprévisible lorsque le terminal sur lequel il est executé n'est pas correctement configuré.

Objet du programme touche

Le programme touche vous permet :

- de contrôler que, pour une configuration de terminal donnée, les touches fonction sont reconnues dans terminfo (choix 1 : Configuration touche dans terminfo),
- de connaître la séquence de caractères des touches fonction ne comportant pas la séquence Ctrl (choix 0 : séquence touche),
- de connaître la séquence Ctrl des touches fonction particulières (choix 2 : Séquence Ctrl-x).

Mise en oeuvre du programme touche

Tapez au clavier : touche -f (-f pour la version francaise)

puis <RETURN> (touche retour chariot)

L'exécutable affiche :

Votre terminal est configuré en xxx

Séquence touche tapez 0 <RETURN>

Configuration touche dans terminfo tapez 1 <RETURN>

Séquence CTRL-x tapez 2 <RETURN>

avec xxx = VT, VT52, VT100, VT102, VT125, VT200, VT220, VT240, VT241, VT300, VT320, VT330, VT340, VT400, VT240, VT241, VT300, VT320, VT330, VT340, VT400, VT420, LK201, LK401, xterm ...

Dans les choix 1 et 2, l'activation de la touche fonction renvoie directement l'information demandée. Dans le choix 0, l'activation de la touche fonction doit être suivie de <RETURN>.

Sortie du programme : tapez sur la barre d'espacement puis <RETURN>.

Méthode d'utilisation du programme touche

Dans un premier temps, contrôlez si les touches fonction sont reconnues dans terminfo pour l'émulation de terminal active sur votre terminal.

Choisissez l'option 1 du programme touche, puis activez chaque touche fonction décrite dans la colonne 3 du TABLEAU 1 :

- si la touche vous renvoie l'attribut de curses(X).h colonne 1 du TABLEAU 1) alors la touche a sa correspondance dans terminfo,
- si ce n'est pas le cas, essayez plusieurs types d' émulation de terminal X (en positionnant la variable TERM à différentes valeurs), puis retenez l'émulation la mieux adaptée à votre terminal.

Dans un deuxième temps, si toutes les touches fonction ne sont pas prises en compte au niveau terminfo ou mal configurées (cas ou il n'y a pas correspondance entre les colonnes 1 et 3) ou si la touche décrite colonne 3 n'existe pas sur votre clavier (dans ce cas remplacez cette touche fonction par une autre non décrite dans le TABLEAU 1) lancez l'exécutable touche, choisissez l'option 0 ou 2 et notez la séquence de caractères envoyée par chaque touche fonction, qui devra être prise en compte au niveau de terminfo.

Prise en compte des touches fonction au niveau terminfo

Il est vivement conseillé de ne pas modifier les configurations existantes au niveau de terminfo, mais plutôt de créer une nouvelle configuration à partir d'une configuration de terminal existante.

Exemple :

Vous avez testé votre terminal avec le programme touche en configuration vt300, qui vous a renvoyé le message suivant :

Votre terminal est configuré en vt300

Séquence touche tapez 0 <RETURN>

Configuration touche dans terminfo tapez 1 <RETURN>

Séquence CTRL-x tapez 2 <RETURN>

Vous avez testé la configuration de vos touches fonction au niveau terminfo en utilisant l'option 1 du menu ci-dessus.

Le résultat est le suivant :

nom d'attribut dans curses(X).h	**description de la touche utilisée par zar980**	
KEY_UP	flèche vers le haut	(1)
KEY_DOWN	flèche vers le bas	(2)
KEY_LEFT	flèche vers la gauche	(3)
KEY_RIGHT	flèche vers la droite	(4)
	flèche oblique en général	(5)
	touche FIN en général	(6)
	touche INSER en général	(7)
	touche SUPPR en général	(8)
KEY_BACKSPACE	touche backspace	(9)
KEY_ENTER	touche ENTER	(10)
KEY_F(0)	touche fonction associée à la fonction PF0 du zar980	(11)
KEY_F(6)	touche fonction associée à la fonction PF0 du zar980	(12)
KEY_F(2)	touche fonction associée à la fonction PF2 du zar980	(13)

nom d'attribut dans curses(X).h	**description de la touche utilisée par zar980**	
.	.	.
.	.	.
.	.	.
KEY_F(24)	touche fonction associée à la fonction PF24 du zar980	(35)

A partir de ce tableau on constate :

- les touches (1),(2),(3),(4),(9),(10),(11),(12),(13),..., (35) sont déclarées dans la base terminfo,
- la touche (12) y est définie incorrectement,
- les touches (5),(6),(7),(8) sont absentes de terminfo pour un terminal défini en vt300.

L'opération suivante consiste à déterminer la séquence de caractères envoyée par chaque touche fonction (5), (6), (7), (8), (12).

Apres le lancement de l'exécutable touche avec l'option 0, on obtient le tableau ci-dessous :

séquence de caractères envoyée par la touche fonction	**description de la touche utilisée par zar980**	
\E[1~	flèche oblique en général	(5)
\E[4~	touche FIN en général	(6)
\E[2~	touche Inser en général	(7)
\E[3~	touche Suppr en général	(8)
\E[18~	touche fonction associée à la fonction PF1 de zar980	(12)

A la suite de cette opération vous créez dans le fichier source de la base de données de terminfo la séquence suivante :

vtcgi

khome=\E[1~,

kll=\E[4~,

kich1=\E[2~,

kdch1=\E[3~,

kf2=\E[18~,

use=vt300,

Votre nouvelle configuration s'appelle vtcgi et hérite de la configuration vt300.

Il vous suffira au niveau du login utilisateur de déclarer dans le fichier de configuration .login ou .profile la variable TERM sous la forme :

export TERM=vtcgi

Configuration de l'écran

Vérifier la configuration de l'écran consiste à vérifier que la gestion des attributs d'intensité et de présentation est correctement effectuée par le terminal.

Le programme zar980 traite les attributs suivants :

INTENSITE	PRESENTATION
.double brillance	.inversion vidéo
.intensité normale	.clignotement
.souligné	.couleur

Comme pour les touches fonction, la base de données terminfo peut contenir leur représentation :

nom d'attribut dans curses(X).h	**nom d'attribut dans terminfo**	**description**	
A_BLINK	blink	clignotement	(1)
A_BOLD	bold	double brillance	(2)
A_DIM	dim	demi-intensité	(3)
A_REVERSE	rev	inversion vidéo	(4)
A_UNDERLINE	smul	souligné (debut)	(5)
	rmul	souligné (fin)	

**** TABLEAU 2 ****

Restriction : sur terminal hpterm, le mode clignotement n' est pas supporté.

Le programme 'video' (source C et exécutable livrés) affiche une liste de libellés sur lesquels sont positionnés différents attributs d'intensité et de présentation, permettant donc de tester si votre terminal supporte l' ensemble de ces attributs.

Aux attributs cités ci-dessus zarterm prend en charge les attributs de couleur qui sont dans la base de données terminfo :

nom d'attribut dans curses(X).h	**nom d'attribut dans terminfo**	**description**
COLOR_WHITE	colf0	caractère en blanc
COLOR_RED	colf1	caractère en rouge
COLOR_GREEN	colf2	caractère en vert
COLOR_YELLOW	colf3	caractère en jaune(marron)
COLOR_BLUE	colf4	caractère en bleu
COLOR_MAGENTA	colf5	caractère en magenta
COLOR-CYAN	colf6	caractère en cyan
COLOR_BLACK	colf7	caractère en noir
	colb0	fond en noir
	colb1	fond en rouge
	colb2	fond en vert
	colb3	fond en jaune
	colb4	fond en bleu
	colb5	fond en magenta
	colb6	fond en cyan
	colb7	fond en blanc

Jeu d'essai

Présentation du Jeu d'essai

Le fichier MBUPDT.ZAR contient un exemple de dialogue (de code MA), sous forme de mouvements batch. Pour réaliser les tests proposés, le jeu d'essai fourni dans MBUPDT.ZAR doit être integré dans votre base de spécifications par la procédure UPDT de mise à jour batch.

Le jeu d'essai permet d'une part de se familiariser avec l'emploi de zar980 et d'autre part de tester l'environnement clavier-écran.

Compilation du dialogue

Après avoir genéré le Dialogue MA et ses écrans :

MA00ME Menu general

MA00M1 Menu 1 (navigation par code opération)

MA00M2 Menu 2 (navigation par touche fonction)

MA01CA Fiche cassette (menu 1)

MA01FI Liste des films (menu 1)

MA02CA Fiche cassette (menu 2)

MA02FI Liste des films (menu 2)

MAHELP Ecran souffleur

Compilez ces programmes à l'aide du script shell comp_cobol suivant :

COBOPT='-C NOANIM

-C ASSIGN=EXTERNAL

-C NOBOUND

-C DEFAULTBYTE=32

-C NOLIST

-C NESTCALL

-C SEQUENTIAL=LINE

-C NOTRACE

-C NOTRUNC

-C NOWARNING

-N NOHPOPTIMIZE (sur HP)

-N NOCHECK (sur AIX/BOSX)

-N NOBOUNDOPT' (sur AIX/BOSX)

export COBOPT

cob -v -u $COBOPT MA.cbl

cob -v -u $COBOPT MA00ME.cbl

cob -v -u $COBOPT MA00M1.cbl

cob -v -u $COBOPT MA00M2.cbl

cob -v -u $COBOPT MA01CA.cbl

cob -v -u $COBOPT MA01FI.cbl

cob -v -u $COBOPT MA02CA.cbl

cob -v -u $COBOPT MA02FI.cbl

cob -v -u $COBOPT MAHELP.cbl

Puis copiez les fichiers .gnt produits sous un répertoire référencé par la variable COBPATH.

NOTE : sous AIX, si la version du run time COBOL est supérieure a 3.2.20, vous devez assigner la variable d' environnement LIBPATH avant le lancement de la commande ″cob″.

Exemple : LIBPATH=/lib:/usr/lib:$COBDIR/coblib

Sous IRIX , SINIX , SunOS ou OSF1, vous devez assigner la variable LIBPATH avant le lancement de la commande ″cob″.

Exemple : LD_LIBRARY_PATH=$COBDIR/coblib

Lancement de l'application

Exécutez le fichier de commandes testma, qui assigne les fichiers nécessaires puis lance le dialogue MA.

NOTE : sous AIX, si la version du Run Time COBOL est supérieure à 3.2.20, vous devez assigner la variable d'environnement LIBPATH avant le lancement du fichier de commandes testma.

Exemple : LIBPATH=/lib:/usr/lib:$COBDIR/coblib

Sous IRIX , SINIX , SunOS ou OSF1, vous devez assigner la variable LIBPATH avant le lancement du fichier de commandes testma.

Exemple : LD_LIBRARY_PATH=$COBDIR/coblib

Les fichiers nécessaires à l'application sont les suivants :

- fparam fichier de paramètres du zar980 (livre)
- LE fichier des libellés d'erreur de l'application (livre)
- HE fichier sauvegarde si appel du help (créé par l'appli.)
- CA fichier des cassettes (créé par l'application)
- VI fichier des films (créé par l'application)

Le dialogue est très simple, il permet de consulter ou mettre à jour un fichier de cassettes video et un fichier de films. Il est découpé en deux branches, l'une permettant de naviguer dans l'application par la saisie d'un code opération, l'autre par touches fonction.

La saisie du code opération et du code action est obligatoire et doit toujours être effectuée en majuscules.

Les valeurs du code opération sont décrites dans les écrans; les valeurs du code action sont :

C (création), M (modification), A (annulation).

Appel de la documentation écran : PF6 , rubrique : PF7

Tabulation : touche de tabulation ou touches Ctrl et i simultanément

Dans la suite, le numéro entre parenthèses qui suit les touches citées fait référence au numéro du TABLEAU 1.

Tests de la configuration du clavier

L'objet de cette section est de vérifier le bon comportement des touches de votre clavier à l'aide du jeu d'essai fourni. Pour chaque touche décrite dans le tableau 1 vous devez retrouver le même comportement au niveau de la touche de votre clavier.

Ces tests s'organisent en deux temps : navigation dans l'application par la zone 'CODE OPERATION' et test des touches de déplacement du curseur, puis test des 'PF-keys'.

Navigation dans l'application par le 'CODE OPERATION'

Au lancement de l'application, le menu suivant s'affiche :

```
                              MENU GENERAL
                              ============
                        M1 : MENU 1 (code operation)
                        M2 : MENU 2 (touche fonction)
                        FT : FIN DE TRANSACTION
                              CODE OPERATION:
```

Le curseur est positionné sur la zone 'CODE OPERATION'.

Saisissez M1 dans cette zone pour afficher le menu suivant :

```
                              MENU VIDEO1
                              ===========
                      CA : FICHE CASSETTE
                      FI : LISTE DES FILMS
                      ME : RETOUR MENU GENERAL
                      M1 : RETOUR MENU VIDEO1
                      FT : FIN DE TRANSACTION
CODE OPERATION:         NUMERO DE CASSETTE:
```

Puis, dans les écrans :

- MA = Mise à jour
- SU = Suite de l'affichage

Vous allez créer une fiche cassette. Le curseur est toujours positionné sur le 'CODE OPERATION'. Saisissez CA et 001 (num. cassette) :

```
                       FICHE CASSETTE

CODE ACTION.......:
NUMERO DE CASSETTE: 001           TYPE CASSETTE.....:
                                  MARQUE CASSETTE...:
                                  DUREE CASSETTE....:
                                  DUREE RESTANTE....:

A PO TITRE DU FILM                GE DUR CO V

ME=MENU, M1=MENU1, CA=FICHE CASSETTE, FI=LISTE FILMS, MA=MAJ
 FIN DE LISTE

CODE OPERATION: MA    NUMERO DE CASSETTE:
```

Saisissez les informations suivantes :

```
CODE ACTION.......: C
NUMERO DE CASSETTE: 001           TYPE CASSETTE.....: VHS
MARQUE CASSETTE...: TDK
DUREE CASSETTE....: 240
DUREE RESTANTE....:

A PO TITRE DU FILM                          GE DUR VE V
C 01 Robin des Bois                         AV 120 VF 4
C 02 Metropolis                             FI 110 VO 4
```

Positionnez-vous ensuite sur la zone 'CODE OPERATION' pour y saisir FI (liste des films) :

```
LISTE DES FILMS

A NUM PO TITRE DU FILM                    GE VE DURV
  001 01 Robin des Bois                   AV VF 1204
  001 02 Metropolis                       FI VO 1104
ME=MENU, M1=MENU1, CA=FICHE CASSETTE, FI=LISTE FILMS, MA=MAJ
FIN DE LISTE

CODE OPERATION: MA    NUMERO DE CASSETTE:
```

Utilisez cet écran pour tester les touches 'Flèches' (1 à 4) de déplacement du curseur. Vous pouvez poursuivre vos tests sur les deux éérans ci-dessus, qui permettent la consultation et la mise à jour.

Lorsque cette première série de tests est terminée, retournez au menu general par le 'CODE OPERATION' ME :

```
                              MENU GENERAL
                              ============
                    M1 : MENU 1 (code opération)
                    M2 : MENU 2 (touche fonction)
                    FT : FIN DE TRANSACTION
                              CODE OPERATION:
```

Navigation dans l'application par les 'PF-keys'

Accédez au 'Menu Video2' :

```
                  MENU VIDEO2
                  ===========
PF-03 : FICHE CASSETTE
PF-04 : LISTE DES FILMS
PF-01 : RETOUR MENU GENERAL
PF-02 : RETOUR MENU VIDEO2
PF-05 : FIN DE TRANSACTION
Puis, dans les écrans :  ENTREE= Mise à jour
                         PF-08 = Suite de l'affichage
                         PF-06 = Doc. écran
                         PF-07 = Doc. rubrique
                         NUMERO DE CASSETTE:
```

A partir de ce menu vous pouvez tester les touches fonction proposées.

La saisie dans les écrans est la même que pour les écrans du 'Menu Video1'.

Tests de la configuration de l'écran

L'objet de cette section est de vérifier que les attributs d'intensité et de présentation sont correctement pris en compte par le terminal.

L'écran 'MENU GENERAL' propose differents attibuts d' intensité et de présentation supportés par le programme zar980 :

- le titre 'MENU GENERAL' est en double brillance de couleur cyan,
- le choix 1 (menu 1) est en intensité normale, souligné et de couleur jaune,
- le choix 2 (menu 2) est en intensité normale, clignotant et de couleur bleu,
- le choix 3 (fin) est en double brillance, souligné et de couleur rouge,
- le libellé 'CODE OPERATION' est en inversion video de couleur verte,
- la première zone de saisie est invisible.

Vous pouvez modifier ces attributs et ceux des autres Ecrans dans la base de spécifications, puis re-générer le Dialogue pour des tests complémentaires. Enfin, si l'affichage de ces différents attributs ne donne pas entière satisfaction, vous pouvez modifier le fichier de paramètres du zar980 FPARAM.

Liste des systèmes UNIX sur lesquels zar980 a été testé

AIX

BOS

BOSX

HP-UX

IRIX

NCR

OSF1

SINIX

SunOS

ULTRIX

Compaq VMS.

Nature des composants :

sources COBOL, sources assembleur, scripts

Liste des composants :

zardec : source programme cobol DEC (caractères)

zarde2 : source programme cobol DEC (champs)

zartrm : source programme assembleur DEC

scrdec : source sous-programme assembleur DEC

pacvmss : source programme cobol DIGITAL VMS

vmsutil : scripts VMS et fichiers divers

Architecture des Applications

ARCHITECTURE DES APPLICATIONS

L'activation d'un programme à partir d'un autre programme se fait uniquement par l'ordre COBOL "CALL", qui suit les règles standard d'appel des sous-programmes.

Ceci implique l'utilisation d'un programme spécifique pour l'enchaînement des programmes, le MONITEUR.

La gestion des appels des différents programmes est assurée par un MONITEUR d'enchaînement, qui devra être généré pour chaque application.

ACCES AUX FICHIERS

Les accès aux fichiers RMS (Record Management Service) ainsi que la gestion des blocages des ressources fichiers sont gérés automatiquement par le module DIALOGUE, selon la règle de gestion des blocages manuels ("Manual Record Locking").

INTERFACE UTILISATEUR

La gestion de l'interface utilisateur (affichage-reception) est assurée par un sous-programme fourni (dit ZAR980), qui est appelé par chaque programme de l'application.

Plusieurs types de communication avec une application PACBASE-VMS sont possibles :

- Avec des écrans VT :
  - en mode caractères, chaque caractère saisi est immédiatement traité par ZAR980.
  - en mode champs, les données sont transmises à ZAR980 par champs entiers, ce qui tend à optimiser les communications.
- Avec des stations de travail munies du module PAW :
  - par une liaison DECNET, les données sont transmises à ZAR980 par pages entières.
  - par une liaison TCP/IP, les données sont transmises à ZAR980 par pages entières.

Les communications VT-champs, DECNET et TCP/IP sont assurées par un même programme ZAR980 (source ZARDE2), la distinction se faisant par un paramètre externe. Ceci permet à une image-VMS unique de gérer tous les types d'utilisateurs.

Par contre, la communication VT-caractères requiert un programme ZAR980 distinct (source ZARDEC), et donc une image-VMS distincte.

Mise en oeuvre

Avec le module DIALOGUE, sont fournis les fichiers nécessaires à la mise en oeuvre des applications produites par les utilisateurs :

ZARDEC: : Source COBOL du programme ZAR980 en mode VT-caractères.

SCRDEC: : Source assembleur complémentaire au programme ZAR980 en mode VT-caractères.

ZARDE2: : Source COBOL du programme ZAR980 en mode VT-champs, DECNET et TCP/IP.

ZARTRM: : Source assembleur complémentaire au programme ZAR980 en mode VT-champs, DECNET et TCP/IP.

PACVMSS: : Source COBOL du serveur DECNET des applications PACBASE-VMS.

VMSUTIL: : Exemples de DCLs de compilations, link, déclarations TCP/IP, lancements de serveur DECNET, lancements d'application PACBASE-VMS, définition du clavier VT-champs.

Les exemples cités dans les paragraphes suivants sont tous fournis dans le fichier VMSUTIL.

Mode VT-caractères

1. Compilation des programmes ZARDEC et SCRDEC

   Voir l'exemple COMPZARCHAR.COM
2. Compilation des programmes de l'application

   Voir l'exemple COMPAPPLI.COM
3. Link de l'image de l'application

   Voir l'exemple LINKAPPLICHAR.COM
4. Lancement de l'application

   Les utilisateurs doivent se connecter d'abord à une session VMS, puis à l'application.

   Voir les exemples EXECAPPLIVT.COM et EXECAPPLI.COM

   Gestion du clavier VT.

- la touche de déplacement vers le bas déplace le curseur sur le premier caractère de la première zone variable située après la fin de la ligne courante.
- la touche de déplacement vers le haut déplace le curseur sur le premier caractère de la dernière zone précédant le début de la ligne courante.
- la touche-fonction PF4 est réservée à l'effacement de fin de zone.
- la touche PF12 est réservée à la tabulation horizontale arrière ('BACKSPACE').
- la touche F13 est réservée à l'effacement de toute la zone ('LINEFEED').
- les touches PF1, PF2, PF3, F7 à F11 et F15 à F20 peuvent être utilisées dans les programmes.
- le curseur ne peut se déplacer qu'à l'intérieur des zones variables.

Mode VT-champs

1. Compilation des programmes ZARDE2 et ZARTRM

   Voir l'exemple COMPZARFIELD.COM
2. Compilation des programmes de l'application

   Voir l'exemple COMPAPPLI.COM
3. Link de l'image de l'application

   Voir l'exemple LINKAPPLIFIELD.COM
4. Lancement de l'application

   Les utilisateurs doivent se connecter d'abord à une session VMS, puis à l'application.

   Voir les exemples EXECAPPLIVT.COM et EXECAPPLI.COM

   Symboles externes utilisés.

   Le comportement de l'application peut être modifié par des symboles externes :

PACBASE_TYPCOM

indique le mode de communication à utiliser: 'VT' commande le mode VT-champs.

PACBASE_TIMEOUT

indique, en secondes, sur 8 caractères numériques, le temps d'attente maximum de l'application (ex: 00003600 = 1h).

La valeur par défaut est 09999999. Si le temps maximum est atteint, ZAR980 renvoie à l'application le code erreur '14'.

PACBASE_AUXKEY

indique l'utilisation du clavier numérique en mode fonctions (valeur 'Y').

PACBASE_KPARAM

personalisation du clavier.

'WRITE' : écriture des valeurs par défaut dans le fichier KPARAM.DAT.

'READ' : prise en compte des valeurs spécifiées dans le fichier KPARAM.DAT.

ATTENTION : les modifications de valeurs doivent respecter l'ordre des définitions et le cadrage des valeurs.

Les valeurs par défaut sont fournies dans l'exemple KPARAM.DAT.

Mode de communication DECNET

En mode DECNET, l'utilisateur doit se connecter à l'application à partir d'une station de travail munie du module PAW.

L'application micro communique avec un 'serveur DECNET' (fourni dans le fichier PACVMSS), qui gère le lancement et l'arrêt des applications PACBASE-VMS, ainsi que la communication.

1. Compilation des programmes ZARDE2 et ZARTRM

   Voir l'exemple COMPZARFIELD.COM
2. Compilation des programmes de l'application

   Voir l'exemple COMPAPPLI.COM
3. Link de l'image de l'application

   Voir l'exemple LINKAPPLIFIELD.COM
4. Compilation et link du serveur DECNET

   Voir l'exemple COMPDNETSERV.COM
5. Lancement du serveur DECNET

   Le serveur DECNET doit être activé et arrêté à partir d'une session VMS. On doit le lancer en process détaché par la procédure d'initialisation DNETINIT.

   Voir les exemples DNETINIT.COM et DNETSERV.COM.

6. Lancement des applications par le serveur DECNET

   Voir les exemples EXECAPPLIDECNET.COM et EXECAPPLI.COM

7. Arrêt du serveur DECNET

   Voir l'exemple DNETSTOP.COM

   Symboles externes utilisés.

   Le comportement de l'application peut être modifié par des symboles externes :

   PACBASE_TYPCOM

   indique le mode de communication à utiliser: 'DECNET' commande le mode DECNET.

   PACBASE_TIMEOUT

   indique, en secondes, sur 8 caractères numériques, le temps d'attente maximum de l'application (ex: 00003600 = 1h).

   La valeur par défaut est 09999999. Si le temps maximum est atteint, ZAR980 renvoie à l'application le code erreur '35'.

   NAME

   identification du serveur, utilisée dans les communications avec le serveur DECNET.

   La valeur de ce paramètre doit correspondre à celle du paramètre NAME de la procédure DNETSERV.COM.

   D'autres paramètres externes indiquent au serveur DECNET les commandes à exécuter pour chaque application PACBASE-VMS. Ces paramètres figurent dans le fichier d'exécution du serveur DECNET. Voir l'exemple DNETSERV.COM.

   NAME

   identification du serveur, utilisée dans les communications avec les utilisateurs et avec les applications.

   La valeur de ce paramètre doit correspondre à celle du paramètre 'P0' des communications de la station.

   'appl'

   fichier de commandes associé à l'application PACBASE_VMS. Le nombre de paramètres 'appl' n'est pas limité. La valeur du paramètre 'P1' des communications de la station doit correspondre à un des paramètres 'appl' définis au lancement du serveur DECNET.

   'appl'LOG

   nom du fichier compte-rendu associé à l'application 'appl'.

   TRACE

   mode trace. Valeurs : TRACE ou NOTRACE.

Mode de communication TCP/IP

En mode TCP/IP, l'utilisateur doit se connecter à l'application à partir d'une station de travail munie du module PAW.

L'application micro communique avec l'application par l'intermédiaire de l'interface TCP/IP de VMS, appelée UCX.

1. Compilation des programmes ZARDE2 et ZARTRM

   Voir l'exemple COMPZARFIELD.COM
2. Compilation des programmes de l'application

   Voir l'exemple COMPAPPLI.COM
3. Link de l'image de l'application

   Voir l'exemple LINKAPPLIFIELD.COM
4. Déclaration des ports UCX

   Le numéro du port UCX associé à l'application PACBASE-VMS doit figurer dans le paramètre 'P2' des communications de la station.

   Voir l'exemple TCPIPUCX.COM
5. Fichier de commandes de l'application

   Voir les exemples EXECAPPLITCPIP.COM et EXECAPPLI.COM

   Symboles externes utilisés.

   Le comportement de l'application peut être modifié par des symboles externes :

   PACBASE_TYPCOM

   indique le mode de communication à utiliser: 'TCPIP' commande le mode TCP/IP.

   PACBASE_TIMEOUT

   indique, en secondes, sur 8 caractères numériques, le temps d'attente maximum de l'application (ex: 00003600 = 1h).

   La valeur par défaut est 09999999. Si le temps maximum est atteint, ZAR980 renvoie à l'application le code erreur '58'.

Bull Gcos7.

Nature des composants :

Sources COBOL

Liste des composants :

zarg7 : source programme cobol ZAR980

Webg7 : source programme cobol ZAR980 pour PACWEB

Présentation de la variante multi-écrans

Cette variante de langage permet d'obtenir un programme transactionnel destiné à des écrans QUESTAR, VIP 7700 et 7800 et IBM 3270.

Cette variante ne génère pas une description physique de l'écran, celle-ci étant effectuée par un sous-programme à partir d'une table contenant une description logique de l' écran; le sous-programme standard est ZAR980.

Le sous-programme génère la grille en fonction du type d' écran qui lui est indiqué.

NOTE : Un programme généré avec cette variante n'est pas compatible avec un programme généré avec une autre variante.

CODIFICATION DU TYPE D'ECRAN

Le type d'écran utilisé doit être alimenté par l'utilisateur au moyen des lignes -P pour le premier écran du dialogue ; sa valeur est transférée aux écrans suivants par l'intermédiaire de la zone de communication.

NOTE : Si le dialogue n'a pas de premier écran, la zone est initialisée à zéro dans tous les programmes.

Bull Gcos8.

Nature des composants :

Sources COBOL

Liste des composants :

zarg8 : source programme cobol ZAR980

Webg8 : source programme cobol ZAR980 pour PACWEB

MVS/CICS.

Nature des composants :

Sources COBOL

Liste des composants :

zarcvs : source programme cobol II ZAR980 MVS et VSE

webcvs : source programme cobol ZAR980 pour PACWEB

Présentation de la variante multi-écrans

Le formatage du message physique pour envoi, et la remise en forme du message logique à recevoir sont assurés par un sous-programme, en fonction du type de terminal utilisé.

Le sous-programme (PRCGI) est fourni pour des terminaux type 3270; l'utilisateur doit écrire le sous-programme (PRUSER) pour d'autres types de terminaux. Le débranchement à l'un ou à l'autre est prévu dans le programme en fonction d'une variable.

MVS/IMS.

Nature des composants :

Source programme COBOL

Liste des composants :

zarims : source programme cobol ZAR980

HP3000.

Nature des composants :

Sources COBOL

Liste des composants :

hpform : source programme cobol ZAR980

UNISYS-A.

Nature des composants :

Sources COBOL

Liste des composants :

zarbur : source programme cobol ZAR980

ICL.

Nature des composants :

Sources COBOL

Liste des composants :

zaricl : source programme cobol ZAR980

Chapitre 4. Compléments A&D Workbench

Package : Adwb_util_3.0.exe

OPEN JADE & TIDY

Installation des logiciels OPENJADE et TIDY nécessaires au fonctionnement de l'outil PUBLISHING de la station A&D WORKBENCH.

Version :

3.0

Plateformes :

WINDOWS/NT , WINDOWS/2000

Pré-requis :

Installation de A&D Workbench

Nature des composants :

Exécutable Windows

Liste des composants :

Adwb_util_3.0.exe

Mise en oeuvre :

Exécuter Adwb_util_3.0 .exe ;

Donner le répertoire racine de l'installation de ADWorkbench.

(par exemple c:\program files\ibm\visualage pacbase)

Chapitre 5. Annexes

Transferts de fichiers

Exécutables IBM/MVS

Transfert d'éxécutable vers un serveur IBM/MVS

Procédure pour remonter le load-module xxxxxxxx :

1. Sous TSO : IND$FILE du fichier XXXXXXXX (paramètres de transfert : binary, fixe bloqué 80 caractères) dans un fichier séquentiel (ex: user.xxxxxxxx)
2. Sous l'écran de commande TSO :

   saisir RECEIVE INDA puis transmit

   saisir le nom du fichier séquentiel (ex: 'user.xxxxxxxx')

   puis transmit

   indiquer le nom de la bibliothèque de load module (ex: da ('ex.pac25.mbr8'))

C.U. BULL/GCOS7

Transfert de programme vers un serveur Bull Gcos7

1. Transférer le fichier xxxx sur la machine Gcos7 par un utilitaire de votre choix, dans un fichier séquentiel ou un membre de bibliothèque SL.
2. Exécuter le fichier xxxx, qui met à jour la bibliothèque de CU : EJ xxxx VL=(nom-de-la-librairie-de-cu-à-mettre-à-jour)
3. Linker le programme :
   - Pour un sous-programme TP : lancer la génération du TDS
   - Pour un programme Batch ou TP : utiliser l'éditeur de lien du site

Fichiers textes

Le transfert de fichier texte est utilisé pour remonter les fichiers de type :

Sources programmes

Scripts

JCL

Mouvements Pacbase

...

Les paramètres sont ASCII, et CRLF.

Référence : SUPSRC00001F - 6113

Imprimé en France

(1P) P/N: SUPSRC00001F - 6113