

VisualAge Pacbase



# Information Support Technique Fichiers complémentaires à VisualAge Pacbase Sources ZAR980 pour cibles UNIX

*Version 3.0*





VisualAge Pacbase



# Information Support Technique Fichiers complémentaires à VisualAge Pacbase Sources ZAR980 pour cibles UNIX

*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/awdtools/vapacbase/productinfo\\_f.htm](http://www.ibm.com/software/awdtools/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.

### **première édition (avril 2003)**

La présente édition s'applique à :

- VisualAge Pacbase Version 3.5

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

© Copyright International Business Machines Corporation 1983,2003. All rights reserved.

---

## Table des matières

Notices . . . . .	v	Chapitre 2. Modes de Compilation . . . . .	5
Marques . . . . .	vii	Chapitre 3. Configuration du poste . . . . .	9
Chapitre 1. Présentation du lot XP250UIX. . . . .	1	Chapitre 4. Jeu d'essai . . . . .	19



---

## 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. Présentation du lot XP250UIX

### Objet :

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

### Nature des composants :

Fichiers textes, Fichier compressé ISO

### Liste des composants :

readme\_F.v02: notice d'installation française

readme\_E.v02: notice d'installation anglaise

zarbase.iso : fichier des composants compressé

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)

\*\*\*\*\*

GENERATION DIALOGUE MICRO FOCUS UNIX

ENVIRONNEMENT : UNIX

REFERENCE : XP250V02UNIX

VERSION : V02

DATE : 02/04/2002

\*\*\*\*\*

Ce lot technique est composé pour l'essentiel de deux utilitaires exécutable 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

---

## Chapitre 2. Modes de Compilation

### Mise en oeuvre

Compilation du programme C Zar980

Trois modes de compilation sont proposés, qui sont :

- le mode debug en vue d'utiliser le débogueur cobol ANIMATOR,
- le mode normal d'exploitation,
- le mode dégradé, 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 à 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



---

## Chapitre 3. 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 être compilé puis lié 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 débogueur 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 chapitre 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.

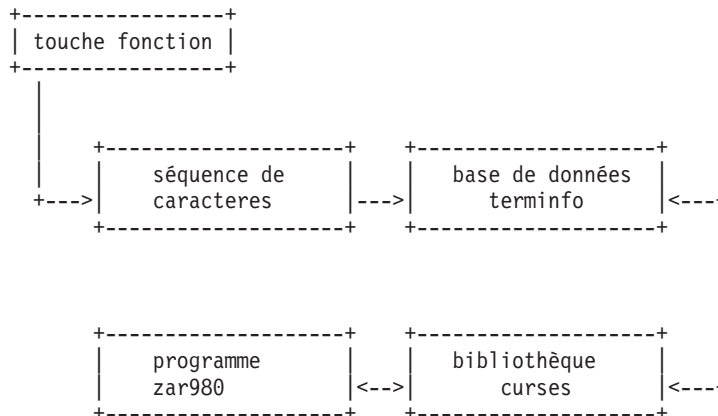
La configuration d'un terminal est constituée d'un ensemble d'attributs.

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 à 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)

<b>nom d'attribut dans curses(X).h</b>	<b>nom d'attribut dans terminfo</b>	<b>description/interprétation par le programme zar980</b>	
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 première 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)
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)

nom d'attribut dans curses(X).h	nom d'attribut dans terminfo	description/interprétation par le programme zar980	
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)
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, VT420, 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'espace 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 où 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)
.	.	.
.	.	.
.	.	.
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).

Après 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

<b>nom d'attribut dans curses(X).h</b>	<b>nom d'attribut dans terminfo</b>	<b>description</b>
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

---

## Chapitre 4. 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 intégré 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 généré 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 écrans ci-dessus, qui permettent la consultation et la mise à jour.

Lorsque cette première série de tests est terminée, retournez au menu général 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 différents attributs 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 vidéo de couleur verte,
- la première zone de saisie est invisible.

Vous pouvez modifier ces attributs et ceux des autres écrans 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







Référence : SUPS0200001F - 6286

Imprimé en France

(1P) P/N: SUPS0200001F - 6286

