

Platine Terminal AXEL

AX3000 modèles asynchrones

*Mise en œuvre sous
UNIX*

La reproduction et la traduction de ce manuel, ou d'une partie de ce manuel, sont interdites. Pour tout renseignement complémentaire, s'adresser à :



Zone d'activité d'Orsay-Courtabœuf
16 Avenue du Québec
BP 728
91962 LES ULIS Cedex
Tél. : 01.69.28.27.27
Fax : 01.69.28.82.04

Les informations contenues dans ce document ne sont données qu'à titre indicatif ; elles peuvent être modifiées sans préavis. AXEL ne peut, en aucun cas, être tenu responsable des erreurs qui pourraient s'y être glissées.

© - 1994-1996 - AXEL - Tous droits réservés

1 - PRESENTATION ET GENERALITES	1
1.1 - GENERALITES.....	2
1.2 - CONFIGURATION.....	4
1.2.1 - Set-up de la Platine Terminal AX3000	4
1.2.2 - Configuration système	4
1.3 - IMPRIMANTE LOCALE	5
1.3.1 - Configuration.....	5
1.3.2 - Utilisation	6
2 - MISE EN ŒUVRE SOUS UNIX/XENIX SCO.....	9
2.1 - INSTALLATION LOGICIELLE.....	10
2.1.1 - Paramétrage du set-up	10
2.1.2 - Configuration système	11
2.2 - LE MULTISCREEN.....	15
2.2.1 - Déclaration de mscreen	15
2.2.2 - Activation de mscreen	16
2.2.3 - Utilisation de mscreen	17
2.3 - GESTION DE L'IMPRIMANTE LOCALE	19
2.3.1 - Installation d'une imprimante locale	19
2.3.2 - Déclaration et utilisation de l'imprimante locale	19
2.4 - GESTION SCANCODE	21
2.4.1 - Déclaration pour une utilisation permanente	22
2.4.2 - Déclaration pour une utilisation ponctuelle.....	24
2.4.3 - Déclaration pour l'utilisation de Word.....	26
3 - MISE EN ŒUVRE SOUS IBM AIX.....	27
3.1 - INSTALLATION LOGICIELLE.....	28
3.1.1 -Paramétrage du set-up	28
3.1.2 - Configuration système	29
3.1.3 - Utilisation des caractères composés	30
3.2 - GESTION DE L'IMPRIMANTE LOCALE	33
3.2.1 - Installation d'une imprimante locale	33
3.2.2 - Utilisation de l'imprimante locale	33

4 - LE MULTISCREEN.....	35
4.1 - PRESENTATION.....	36
4.2 - INSTALLATION.....	37
4.3 - UTILISATION.....	38
4.4 - LES OPTIONS.....	39
4.4.1 - Option b : choix des pttys.....	39
4.4.2 - Option c : bordure d'écran.....	39
4.4.3 - Option e : sortie interdite.....	40
4.4.4 - Option f : commandes initiales.....	40
4.4.5 - Option h : aide en ligne.....	41
4.4.6 - Option k : commentaire.....	41
4.4.7 - Option l : multilogin.....	42
4.4.8 - Option m : mode muet.....	44
4.4.9 - Option n : nombre de vues.....	44
4.4.10 - Option p : préaffectation des pseudo-terminaux.....	45
4.4.11 - Option s : ligne status.....	46
4.4.12 - Option t : pas de test du tty.....	46
4.4.13 - Option v : Unix SCO 3.2v4.0.....	46
4.4.14 - Option w : non régénération de shell.....	46
4.4.15 - Option x : buffer circulaire.....	47
4.5 - CONSEILS D'UTILISATION.....	48
4.5.1 - Multilogin ou Multishell ?.....	48
4.5.2 - Création d'un utilisateur multivue.....	48
4.5.3 - Ligne status.....	49
4.5.4 - Quitter le multiscreen.....	50
4.6 - LIMITATIONS DE LA NORME ANSI.....	51

**- 1 -
PRESENTATION
ET GENERALITES**

Ce chapitre a pour objet la présentation des principes généraux d'installation et d'utilisation de la Platine Terminal AX3000 dans les différents environnements UNIX.

Le nombre des différents systèmes UNIX disponibles étant important, ce chapitre fournit des principes généraux de mise en œuvre de la Platine Terminal AXEL et d'une éventuelle imprimante locale (imprimante directement connectée à l'un des ports auxiliaires de la Platine).

Les chapitres 2 et 3 reprennent en détail la mise en œuvre de la Platine AX3000 sous les systèmes d'exploitation les plus couramment utilisés :

- XENIX SCO, UNIX SCO 3.2v2, 3.2v4 et Release 5,
- et IBM AIX,

Le chapitre 4 détaille la fonctionnalité de multiscreen mise en œuvre au moyen du logiciel `axmscreen` fourni avec la Platine.

1.1 - GENERALITES

L'un des paramètres importants de la configuration d'un terminal est le choix de l'émulation. Sous UNIX, les deux émulations les plus utilisées sont :

- VT220 et
- ANSI.

Chacune de ces deux émulations possèdent les caractéristiques suivantes :

- VT220 : cette émulation a l'avantage d'être présente sur la quasi totalité des systèmes UNIX.
Bien que monochrome d'origine, il est possible de l'utiliser en **couleur** grâce à la fonction de coloriage des attributs vidéo qu'offre la Platine Terminal (voir *Le guide de l'utilisateur*).
- ANSI : cette émulation, plus récente que VT220, intègre directement la couleur (sur certains UNIX cependant, l'émulation ANSI peut n'être que monochrome).
En général, la console principale des micro-ordinateurs fonctionnant sous UNIX (SCO, UNIXWARE...) est gérée en émulation ANSI ; ceci garantit à la Platine un **comportement identique à celui de cette console principale**.

L'émulation choisie doit être précisée à deux niveaux :

- le set-up de la Platine Terminal : option **<F10>** (set-up prédéfinis),
- le système UNIX : mise à jour de la variable TERM.

Le tableau suivant donne les valeurs conseillées du set-up prédéfini et de TERM selon le système d'exploitation utilisé :

Système UNIX	Set-up prédéfini	TERM
Unix SCO 3.2v2	UNIX SCO 3.2.2	ansi
Unix SCO 3.2v4	UNIX SCO 4.0	ansi
Unix SCO Release 5	UNIX SCO 4.0	ansi
Xenix SCO	XENIX SCO	ansi
IBM AIX 3.1	ANSI RS6000	hft-c
IBM AIX 3.2	ANSI RS6000	hft-c-old
IBM AIX 4.x	ANSI RS6000	hft-c-old
Unix Ware	UNIX SVR4	AT386
Unix Interactive	ANSI INTERACTIVE	AT386
Solaris 2.3	UNIX SVR4	AT386
Autre Unix	ANSI	ansi
	VT220	vt220

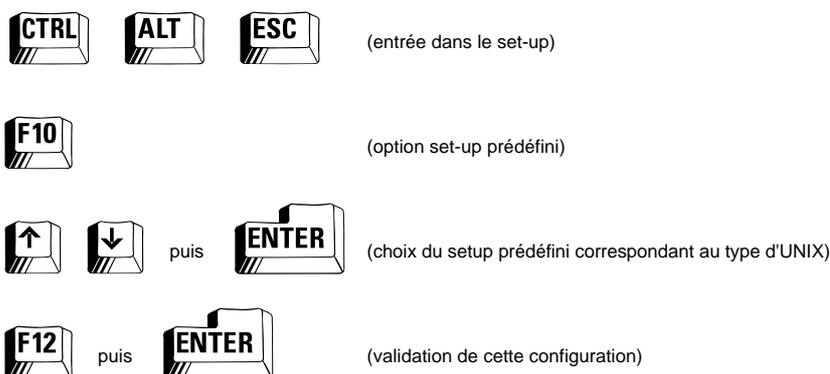
1.2 - CONFIGURATION

1.2.1 - Set-up de la Platine Terminal AX3000

L'un des points forts de Platine Terminal réside dans le fait que c'est elle qui s'adapte au système d'exploitation utilisé et non le contraire. Ceci s'effectue en une seule opération, ce qui permet une installation **rapide et fiable** sur site.

En effet, la fonction de **set-up prédéfini** permet de configurer automatiquement la Platine en mettant à jour tous les paramètres d'environnement (nombre de lignes, valeurs des touches de fonction...) nécessaires à son bon fonctionnement dans un système d'exploitation donné.

La Platine AX3000 peut ainsi être complètement initialisée en effectuant les seules opérations suivantes :



La sélection d'un set-up prédéfini positionne également les paramètres de communication (vitesse, parité...) du port principal selon des valeurs par défaut. Il est bien sûr possible de modifier ensuite, si nécessaire, individuellement ces paramètres. Pour plus de renseignements, consultez *Le guide de l'utilisateur*.

1.2.2 - Configuration système

La valeur de la variable d'environnement TERM doit être impérativement mise à jour. Cette variable décrit le comportement du terminal (nombre de lignes et

de colonnes, valeur des touches de fonctions...) au travers des fichiers `/etc/termcap` et `/usr/lib/terminfo`.

La valeur à attribuer à cette variable doit être en accord avec le set-up prédéfini choisi lors de la configuration de la Platine (voir tableau page 3).

En général, la valeur de cette variable TERM est positionnée au niveau des fichiers `/etc/profile` ou `.profile`. Sous certains UNIX (UNIX SCO par exemple), il est aussi possible de préciser la valeur de TERM dans un fichier qui associe nom de voie (device) et variable TERM.

1.3 - IMPRIMANTE LOCALE

Définition : une imprimante locale est une imprimante directement connectée à un terminal. Ceci permet d'avoir une imprimante proche d'un terminal sans utiliser de voie série supplémentaire.

1.3.1 - Configuration

Une imprimante locale se connecte à la Platine AX3000 :

- soit sur le port parallèle (modèles 35E, 40 et 40B),
- soit sur le port auxiliaire série AUX1 (prise 9 pts ou RJ45 selon modèle).

Cette imprimante doit être déclarée au niveau du set-up de la Platine (<Ctrl><Alt><Echap>) dans l'écran consacré aux ports auxiliaires (<F4>).

Le paramètre 'Port préférentiel' permet de sélectionner le port auxiliaire de l'imprimante. Utilisez la touche <Espace> pour sélectionnez le 'Port parallèle' ou le 'Port série'.

Si le port utilisé est le port parallèle, la configuration est terminée. Tapez <F12> puis <RC> pour sauvegarder et quitter.

Si le port utilisé est le port série, il est nécessaire de préciser la fonction de ce port. Utilisez les flèches verticales pour sélectionner le paramètre 'Utilisation du port série'. Au moyen de la touche <Espace>, sélectionnez la valeur 'Imprimante'.

Cette opération positionne automatiquement les paramètres de communication (vitesse, parité...) du port auxiliaire série selon des valeurs par défaut. Il est bien sûr possible de modifier ensuite, si nécessaire, individuellement ces paramètres. Pour plus de renseignements, consultez *Le guide de l'utilisateur*.

1.3.2 - Utilisation

Les deux utilisations possibles de l'imprimante locale sont :

- l'opération de copie écran (i.e. 'hardcopy'),
- l'impression de fichiers.

a - Copie écran

Cette opération restant totalement locale au niveau de la Platine (pas d'accès à l'ordinateur hôte), aucune configuration du système n'est nécessaire.

Cette fonctionnalité est directement accessible au moyen de la touche **<ImprEcran>**.

b - Impression de documents

Le principe de fonctionnement d'une imprimante locale consiste à rediriger les informations que reçoit le terminal hôte vers l'imprimante. Il est donc nécessaire de différencier les données destinées à l'imprimante de celles destinées à l'écran pour affichage.

C'est pour cela que toutes les données à destination de l'imprimante locale sont encadrées par une 'séquence de connexion' et une 'séquence de déconnexion'.

Pour la Platine AX3000, en émulation ANSI ou VT220, les valeurs de ces séquences sont les suivantes :

- séquence de connexion : Esc [5i,
- séquence de déconnexion : Esc [4i,

Note : ces deux séquences sont représentées respectivement par les directives 'mc4' et 'mc5' dans un fichier terminfo.

En général, sur les systèmes UNIX, il n'existe pas de commande exploitant cette notion d'impression locale. Il est donc nécessaire de créer un script spécifique.

Avec l'éditeur de texte `vi`, créez, dans le répertoire `/usr/bin`, le fichier `localprint` suivant :

```
if [ -z "$1" ]; then
    echo "Usage : $0 fichier"
    exit
fi
echo "Impression de $1 en cours..."
echo "\033[5i\c"
cat $1
echo "\014\c"
echo "\033[4i\c"
echo "OK"
```

Le fichier ainsi créé doit être positionné avec l'autorisation d'exécution. Pour cela tapez la commande suivante :

```
# chmod +x /usr/bin/localprint
```

Pour imprimer un fichier sur l'imprimante locale de la Platine Terminal AXEL, lancez la commande suivante :

```
$ localprint nomfic
```

Note : cette commande ne peut être lancée qu'à partir du terminal auquel est connectée l'imprimante locale. Durant l'utilisation de l'imprimante locale, il est déconseillé de se servir du clavier du terminal.

- 2 -
MISE EN ŒUVRE
SOUS UNIX/XENIX SCO

Ce chapitre décrit les opérations à effectuer pour la mise en œuvre des différentes fonctionnalités de la Platine Terminal AXEL sous les systèmes XENIX SCO, UNIX SCO 3.2v2, 3.2v4 et Release 5.

2.1 - INSTALLATION LOGICIELLE

L'installation d'une Platine Terminal AXEL comporte deux phases :

- le paramétrage du set-up,
- la configuration du système d'exploitation.

2.1.1 - Paramétrage du set-up

La première étape de l'installation d'une Platine Terminal AXEL est la mise à jour du set-up. Grâce à la fonction de **set-up prédéfini**, cette opération peut être effectuée très rapidement :



(entrée dans le set-up)



(option set-up prédéfini)



puis



(choix du setup prédéfini : XENIX SCO, UNIX SCO 3.2.2,
UNIX SCO 4.0 ou SCO OPENSERVER)



puis



(validation de cette configuration)

La Platine AX3000 est alors complètement configurée selon le **set-up prédéfini** choisi, et est prête à fonctionner sous les systèmes UNIX/XENIX SCO à la vitesse de 38400 bauds. Si nécessaire, les éléments composant ce paramétrage automatique peuvent être modifiés individuellement (changement de la vitesse par exemple).

Pour plus d'informations concernant le set-up de la Platine Terminal AXEL, consultez *Le guide de l'utilisateur*.

2.1.2 - Configuration système

L'installation logicielle de la Platine Terminal AXEL sous environnement UNIX/XENIX SCO se déroule en quatre phases :

- choix et configuration d'une voie série,
- activation de la voie série,
- déclaration du type de terminal,
- affectation éventuelle d'un fichier clavier.

Dans la suite de ce chapitre, on suppose que le port série sur lequel doit être connectée la Platine Terminal est déjà déclaré. En effet, l'installation et la déclaration d'un port série sont indépendantes de l'installation et la déclaration du périphérique qui lui est associé.

La sortie série standard COM1 sera utilisée pour illustrer les explications qui suivent, la procédure étant la même pour n'importe quel autre port série.

Sous les systèmes UNIX, à chaque port série est associé un, ou plusieurs fichiers spéciaux, appelés 'devices'. C'est par l'intermédiaire de ces fichiers spéciaux que sont lues ou écrites les données sur le port série.

Ainsi, deux devices sont associés à la COM1 :

- /dev/tty1a,
- et /dev/tty1A.

En général, les devices comportant une lettre majuscule sont dédiés à la gestion de modems (/dev/tty1A par exemple).

Les devices ne comportant que des lettres minuscules sont dédiés à la gestion de terminaux ou d'imprimantes (/dev/tty1a par exemple).

a - Configuration d'une voie série

Dans le fichier `/etc/gettydefs` sont définies les caractéristiques (notamment les paramètres de transmission) des différents périphériques pouvant être connectés à l'ordinateur : imprimantes, terminaux, modems...).

Ce fichier établit une relation entre une variable (ici un caractère) et les caractéristiques d'un périphérique. Par exemple :

- 2 est utilisé pour un modem à 1200 bauds,
- 6 est utilisé pour un terminal ou un modem à 9600 bauds,
- m est utilisé pour un terminal à 9600 bauds,
- n est utilisé pour un terminal à 19200 bauds,
- o est utilisé pour un terminal à 38400 bauds.

Par défaut, le device `/dev/tty1a` est associé au paramètre `m` (vitesse transmission de 9600 bauds).

C'est le fichier `/etc/inittab` qui établit la correspondance entre les devices et leurs paramètres (sous XENIX, ce fichier se nomme `/etc/ttys`).

Recherchez dans ce fichier, le mnémonique de la voie utilisée. La ligne correspondant à la COM1 doit se trouver sous cette forme :

```
sela:23:off:/etc/getty tty1a m
```

Remplacez par exemple, le 'm' par un 'o' pour changer la vitesse de transmission de 9600 bauds à 38400 bauds.

Sauvegardez ces modifications.

Le fichier `/etc/inittab` (ou `/etc/ttys`) étant le résultat de la concaténation de plusieurs fichiers du répertoire `/etc/conf/init.d`, les modifications effectuées au niveau de `/etc/inittab` (ou `/etc/ttys`) doivent être reportées au niveau du fichier présent dans ce répertoire et correspondant à la voie série. Ex. pour COM1 : `/etc/conf/init.d/sio`.

b - Activation de la voie série

Ceci est réalisé par la commande `enable` suivie du nom du device sur lequel est connecté le terminal.

Dans le cas de la COM1 :

```
# enable ttyla
```

Les fichiers `/etc/inittab` et `/etc/conf/init.d/sio` sont mis à jour

La voie série devient ainsi active jusqu'à sa désactivation au moyen de la commande `disable`.

c - Déclaration du type de terminal

L'émulation utilisée par la Platine AX3000 est conforme à l'entrée `ansi` des fichiers `termcap` et `terminfo`. Cette entrée `ansi` (utilisée par la console principale) étant standard dans UNIX/XENIX SCO, la Platine Terminal est directement utilisable sans installation de contrôleurs spécifiques.

Le type d'émulation d'un terminal peut être décrit de deux façons :

- Affectation et exportation de la variable d'environnement **TERM** :

```
$ TERM=ansi  
$ export TERM
```

Ces deux commandes peuvent être lancées, soit à partir d'un script (le fichier `.profile` par exemple), soit à partir de l'interpréteur de commandes.

- Ajout de la ligne suivante dans le fichier `/etc/ttytype` pour mettre en correspondance device et type d'émulation :

```
ansi      ttyla
```

d - Gestion des caractères composés

Sur UNIX/XENIX SCO, Il est possible d'affecter à un terminal (ou même à la console principale), un fichier clavier. Ce fichier, qui est en fait un filtre, permet notamment la gestion des caractères composés (^ et e donnent ê).

Note : les caractères composés peuvent être gérés en local par la Platine (option <F2> du set-up) ce qui évite un paramétrage au niveau système d'exploitation. Pour plus d'information, consultez le *Guide Utilisateur*.

L'affectation d'un fichier clavier peut être effectuée de deux façons :

- Utilisation de la commande `mapchan` :

```
$ mapchan /usr/lib/mapchan/cons.ibm
```

Cette commande peut être lancée, soit à partir d'un script (le fichier `.profile` par exemple), soit à partir de l'interpréteur de commandes.

- Ajout de la ligne suivante dans le fichier `/etc/default/mapchan` pour mettre en correspondance device et fichier clavier :

```
cons.ibm      ttyla
```

Le fichier paramètre `cons.ibm` utilisé n'est donné qu'à titre d'exemple. Le répertoire `/usr/lib/mapchan` contient tous les fichiers claviers disponibles.

Note : pour obtenir sur la Platine un comportement identique à celui de la console principale, il est nécessaire d'affecter à la Platine le même fichier clavier qu'à la console principale (si celle-ci en possède un).

Pour plus de précisions, consultez :

<i>User's Reference</i> (C)	: ENABLE, DISABLE, STTY
<i>User's Reference</i> (M)	: GETTY, UNGETTY, LOGIN, MAPCHAN
<i>User's Reference</i> (F)	: GETTYDEFS, MAPCHAN
<i>User's Reference</i> (ADM)	: MKDEV

2.2 - LE MULTISCREEN

Le multiscreen, disponible en standard sur la console principale, doit être explicitement déclaré pour un terminal. Ce multiscreen est obtenu au moyen de l'utilitaire `mscreen` qui est une fourniture standard de UNIX/XENIX SCO.

Une Platine AX3000 peut supporter jusqu'à 8 vues simultanément, et c'est elle, et non pas la machine centrale, qui sauvegarde automatiquement le contexte propre à chaque vue (ceci permet un changement de vue instantané).

Le contexte de chaque vue comprend :

- l'affichage de l'écran (les caractères et leurs attributs),
- les attributs courants (couleur, position et aspect du curseur...).

Note : AXEL a réalisé un logiciel de multiscreen spécifique à la Platine Terminal AX3000. Ce logiciel, appelé `axmscreen`, offre des fonctionnalités supplémentaires par rapport à `mscreen` (le multishell et les commandes initiales notamment). Pour plus d'informations, consultez le chapitre 4.

2.2.1 - Déclaration de mscreen

Pour gérer le multiscreen, UNIX/XENIX SCO s'appuie, d'une part sur le type d'émulation du terminal (variable `TERM`), et d'autre part sur le contenu du fichier `/etc/mscreencap` qui décrit le comportement du terminal durant le multiscreen.

le multiscreen sur un terminal **ansi** n'est pas géré en standard. Les lignes suivantes doivent être rajoutées à la fin du fichier `/etc/mscreencap` :

```
ansi:\
:who,Alt-F9,\E[<i,:\
:help,Alt-F10,\E[<j,:\
:stop,Alt-F11,\E[<k,\E[0z:\
:quit,Alt-F12,\E[<l,\E[0z:\
:,Alt-F1,\E[<a,\E[1z:\
:,Alt-F2,\E[<b,\E[2z:\
:,Alt-F3,\E[<c,\E[3z:\
:,Alt-F4,\E[<d,\E[4z:\
:,Alt-F5,\E[<e,\E[5z:\
:,Alt-F6,\E[<f,\E[6z:\
:,Alt-F7,\E[<g,\E[7z:\
:,Alt-F8,\E[<h,\E[8z:
```

Sauvegardez le fichier `/etc/mscreencap` ainsi modifié.

Note : l'ensemble de ces lignes est fourni sous la forme d'un fichier (`axcap`) sur la disquette de multiscreen AXEL. Une fois ce fichier recopié sur disque, il suffit de taper la commande `'cat axcap >> /etc/mscreencap'` pour mettre à jour le fichier `mscreencap`.

2.2.2 - Activation de mscreen

La gestion du multiscreen s'appuie sur la notion de pseudo-terminal (un pseudo-terminal pour une vue). Un pseudo-terminal étant considéré comme un device, chaque session du multiscreen est vue par Unix comme reliée à une voie série physique indépendante.

Les pseudo-terminaux utilisés par `mscreen` se nomment `/dev/ttypx`, `/dev/ttyqx`, `/dev/ttyrx` et `/dev/ttysx` (où `x` varie de 0 à f).

a - Activation sous UNIX SCO 3.2v2 et XENIX SCO

De même que pour l'activation de la voie d'une carte, l'activation des pseudo-terminaux se fait par l'intermédiaire de la commande `enable` avec le nom du device approprié.

Cette commande doit être lancée pour chaque device utilisé. Par exemple :

```
# enable tty0
```

b - Activation sous UNIX SCO 3.2v4 et Release 5

Dans la configuration standard du système d'exploitation UNIX SCO 3.2v4 aucun pseudo-terminal n'est déclaré. Il est donc nécessaire de spécifier le nombre total de pseudo-terminaux utilisés par les terminaux multiscreen avant de recompiler le noyau. Pour cela, lancez la commande suivante :

```
# mkdev pty
```

L'utilisation du logiciel `mscreen` ne sera possible qu'après avoir réinitialisé la machine (commande `shutdown`).

2.2.3 - Utilisation de mscreen

Pour utiliser le service de multiscreen, lancez la commande :

```
$ mscreen -n x
```

où `x` est le nombre de vues désirées (de 1 à 8).

Cette commande peut être lancée soit par un script (`.profile` par exemple), soit depuis l'interpréteur de commande d'UNIX.

Une fois le multiscreen actif, les combinaisons de touches opérationnelles sont les suivantes :

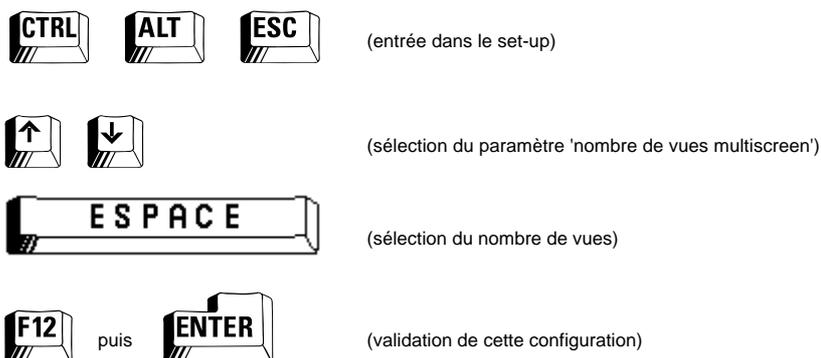
- **<Alt><F1>** à **<Alt><F8>** : changement de vue.
- **<Alt><F9>** : affichage du nom du pseudo-terminal courant (similaire à la commande **who**).
- **<Alt><F10>** : affichage des commandes disponibles (aide),
- **<Alt><F11>** : permet de quitter le logiciel `mscreen` avec un status de retour égal à 1.
- **<Alt><F12>** : permet de quitter le logiciel `mscreen` avec un status de retour égal à 0.
- **<Ctrl><ImprEcran>** : passage à la vue suivante (vue 1 à vue 2, vue 2 à vue 3, ..., vue 8 à vue 1).

Par défaut, le logiciel `mScreen` fournit jusqu'à 8 vues. Il est possible de limiter ce nombre de vues au moyen de l'option `'-n x'` où `x` est ce nombre de vues (de 1 à 8).

Exemple : `mScreen -n 4`

La combinaison de touches **<Ctrl><ImprEcr>** permet de passer automatiquement à la vue suivante (vue 1 à vue 2, ..., vue 8 à vue 1). Pour optimiser cette fonction de permutation de vue, il est nécessaire de déclarer au niveau de la Platine Terminal le nombre de vues réellement utilisées.

Par défaut, le nombre de vues disponibles est égal à 8. Pour modifier ce nombre, il est nécessaire d'intervenir au niveau du set-up de la Platine de la façon suivante :



Pour plus de précisions, consultez :

- User's Reference(C)* : ENABLE, DISABLE
- User's Reference(M)* : MSCREEN
- User's Reference(HW)* : SCREEN

2.3 - GESTION DE L'IMPRIMANTE LOCALE

Définition : une imprimante locale est une imprimante directement connectée à un terminal. Ceci permet d'avoir une imprimante proche du terminal sans utiliser de voie série supplémentaire.

Remarque : une imprimante est en général beaucoup plus lente qu'un terminal (9600 bauds pour l'un, 38400 bauds ou plus pour l'autre) et donc beaucoup plus rapidement saturée (i.e. XOFF). Comme l'imprimante locale et le terminal hôte utilisent la même voie série, la saturation de l'un des deux entraîne l'arrêt de la communication pour les deux. **On peut dire que la vitesse des deux est égale à la vitesse du périphérique le plus lent.**

Solution AXEL : l'utilisation d'une carte multivoie intelligente AXEL avec une Platine AX3000 résout non seulement ce problème de ralentissement mais permet de déclarer l'imprimante locale comme imprimante système (i.e. accessible par la commande `lp`). La déclaration de cette configuration spécifique est décrite dans la *notice d'installation des cartes V605/V610/V810 sous UNIX SCO*.

La suite de ce chapitre ne concerne donc que le fonctionnement standard d'une imprimante locale connectée à une Platine Terminal, l'ensemble étant relié à une voie série quelconque.

2.3.1 - Installation d'une imprimante locale

L'installation d'une imprimante locale sur une Platine AXEL sous UNIX/XENIX SCO est conforme aux indications données au chapitre 1.3 de ce document.

2.3.2 - Déclaration et utilisation de l'imprimante locale

Comme cela a été précisé en début de chapitre, une imprimante locale utilise la voie série du terminal pour recevoir des informations, et donc ralentit celui-ci.

Une imprimante locale peut être utilisée :

- soit pour imprimer des copies d'écran (i.e. 'hardcopy'),
- soit pour imprimer le contenu de fichiers (commande `lprint`).

a - Copie écran

Cette opération restant totalement locale au niveau de la Platine (pas d'accès à l'ordinateur hôte), aucune configuration du système n'est nécessaire.

Cette fonctionnalité est directement accessible au moyen de la touche **<ImprEcran>**.

b - Impression de documents

La commande `lprint` permet l'impression d'un fichier sur l'imprimante locale. Quelques modifications doivent être apportées à la configuration du système UNIX/XENIX SCO pour utiliser cette commande.

Le principe de fonctionnement d'une imprimante locale consiste à rediriger les informations que reçoit le terminal hôte vers l'imprimante. Il est donc nécessaire de différencier les données destinées à l'imprimante de celles destinées à l'écran pour affichage.

C'est pour cela que toutes les données à destination de l'imprimante locale sont encadrées par une 'séquence de connexion' et une 'séquence de déconnexion'.

Pour la Platine AX3000, en émulation UNIX/XENIX SCO, les valeurs de ces séquences sont les suivantes :

- séquence de connexion : Esc [5i,
- séquence de déconnexion : Esc [4i.

L'émulation **ansi** standard ne reconnaît pas ces deux séquences. Le fichier `/etc/termcap` (décrivant l'émulation **ansi**) doit donc être modifié.

- Editez le fichier `/etc/termcap` avec l'éditeur de texte **vi**.

- Recherchez l'entrée **ansi** :

```
li|ansi|xxxxx|xxxx:\
:xx:xxxx:xxx:\
:xxx:xx:xxx:\
:xx:xxx:xxx:
```

- Insérez la ligne suivante dans la description :

```
:PN=\E[5i:PS=\E[4i:\
```

Rq : cette ligne ne doit se trouver ni en première, ni en dernière position dans la description du terminal.

- Quittez vi en sauvegardant le fichier `/etc/termcap`.

La configuration de l'imprimante locale est terminée. Pour imprimer un document sur l'imprimante locale, utilisez la commande suivante :

```
$ lprint nomfic
```

Note : cette commande ne peut être lancée qu'à partir du terminal auquel est connectée l'imprimante locale. Durant l'utilisation de l'imprimante locale, il est déconseillé de se servir du clavier du terminal.

Si l'imprimante n'effectue pas les sauts de lignes, tapez les commandes suivantes avant l'impression :

```
$ FORMS=X
$ export FORMS
```

Pour plus de précisions, consultez :

<i>User's Reference</i> (ADM)	: LPRINT
<i>User's Reference</i> (C)	: LP
<i>User's Reference</i> (F)	: TERMCAP
<i>System Administrator's Guide</i>	: Adding a Local Printer

2.4 - GESTION SCANCODE

La version UNIX SCO 3.2v4 (et suivantes) introduit la notion de terminal avec une gestion scancode du clavier. Dans les versions antérieures de UNIX SCO, seule la gestion ASCII était possible.

La déclaration de la gestion scancode sur la Platine AX3000 peut être effectuée selon trois méthodes :

- de façon permanente : la gestion scancode est opérationnelle dès le login.
- de façon ponctuelle : la gestion scancode n'est effective que pour une durée limitée durant une session.
- durant l'utilisation d'un logiciel : la gestion scancode n'est effective que durant l'utilisation de ce logiciel (Word par exemple).

Avertissement : la gestion scancode impose au terminal de fonctionner avec un contrôle de flux DTR ou XPC (et non pas XON/XOFF). Vérifiez que la sortie série sur laquelle est connectée la Platine gère correctement l'un de ces deux contrôles de flux.

2.4.1 - Déclaration pour une utilisation permanente

a - Déclaration au niveau AX3000

Le paramétrage de la Platine AX3000 doit être modifié pour accepter la gestion scancode :



(entrée dans le set-up)



(option clavier)



(mise à jour de 'Codes' à 'Géographiques (scancodes)')



(validation de cette configuration)

Note : le contrôle de flux XON/XOFF étant incompatible avec une gestion scancode du clavier, la mise à jour du paramètres 'Codes' au niveau du set-up de la Platine entraîne le positionnement du contrôle de flux à XPC (si celui-ci était XON/XOFF).

b - Déclaration au niveau de UNIX SCO 3.2v4

Une modification du fichier `/etc/gettydefs` doit être effectuée pour créer une entrée acceptant la gestion scancode.

Ce fichier contient en standard une entrée pour un terminal scancode à 9600 bauds (entrée `sc_m`).

Voici à titre d'exemple la ligne à rajouter dans le fichier `/etc/gettydefs` pour la déclaration d'un terminal scancode à 38400 bauds :

```
sc_o # B38400 HUPCL SCancode # B38400 CS8 SANE HUPCL
TAB3 ECHOE IXANY SCancode # \r\n@!login: # sc_o
```

La deuxième étape est la déclaration des voies destinées à recevoir les entrées scancodes. Cette modification se fait dans le fichier `/etc/inittab`.

Désactivez tout d'abord la voie à modifier (ex : `disable tty1a`).

Puis au moyen de l'éditeur `vi`, recherchez dans le fichier `/etc/inittab` le mnémonique de la voie utilisée et remplacez le caractère de codage (o par exemple) par `sc_o`. Sauvegardez ces modifications puis sortez de l'éditeur.

Réactivez la voie au moyen de la commande `enable`.

La gestion du scancode, ainsi déclarée, fonctionne pour un clavier QWERTY. Pour des claviers d'un autre type (AZERTY par exemple), il est nécessaire d'utiliser une autre table de codage que celle des claviers QWERTY.

Le répertoire `/usr/lib/keyboard` contient les tables de codage scancode de chacun des claviers reconnus. Par exemple, le fichier à utiliser pour un clavier français est `ps.ibm.fra`.

Il est possible d'affecter à une voie déclarée en scancode, un type de clavier précis par l'intermédiaire du fichier `/etc/default/mapkey`.

Dans ce fichier, est affecté à chaque voie un type de clavier. Par exemple, si un terminal, connecté à la COM1, intègre la gestion scancode d'un clavier français, il faut rajouter la ligne suivante dans le fichier :

```
tty1a      ps.ibm.fra
```

Faites de même pour chaque voie associée à un terminal utilisant la gestion scancode.

La prise en compte de ces modifications ne sera effective qu'après avoir éteint puis rallumé l'ordinateur (commande `shutdown`).

Si toutefois le fichier `/etc/default/mapkey` n'est toujours pas pris en compte après avoir réinitialisé, lancez la commande suivante :

```
$ mapkey -a
```

Cette commande doit être lancée, avant l'utilisation d'un terminal géré en scancode.

2.4.2 - Déclaration pour une utilisation ponctuelle

Il peut être intéressant de n'utiliser la gestion scancode qu'occasionnellement (pour certains logiciels par exemple) et de fonctionner en standard en gestion ASCII.

Trois opérations sont nécessaires à la mise en service de cette fonctionnalité.

a - Modification de l'émulation ansi

L'émulation **ansi** utilisée par la Platine AX3000 ne reconnaît pas en standard le changement de gestion clavier (ASCII ↔ scancode) et le changement de contrôle de flux associé (XON/XOFF ↔ XPC).

Pour modifier cette émulation, effectuez les opérations suivantes :

```
# cd /usr/lib/terminfo  
# infocmp ansi > ansi.src
```

Editez le fichier `ansi.src` pour ajouter la ligne suivante (ni au début, ni à la fin du fichier) :

```
smsc=\E[<0A, rmsc=\E[<1A, xonc=e, xoffc=g,
```

Sortez de l'éditeur en sauvegardant ces modifications, puis lancez la commande suivante :

```
# tic ansi.src
```

L'émulation **ansi** ainsi modifiée accepte alors le passage de la gestion ASCII à la gestion scancode (et vice versa) en utilisant le contrôle de flux approprié.

b - Mise à jour du fichier /etc/ttytype

La deuxième opération à effectuer est de déclarer pour le terminal l'association device/émulation.

Par exemple, pour une Platine AX3000 (émulation ansi) connectée à la COM1, vous devez ajouter la ligne suivante (si elle n'existe pas déjà) dans le fichier `/etc/ttytype` :

```
ansi          ttyla
```

c - Changement de gestion clavier

Le passage de la gestion clavier ASCII à la gestion clavier scancode est effectuée au moyen de la commande `scanon`.

En standard, la gestion scancode est prévue pour un clavier de type QWERTY. Pour changer le type du clavier (par exemple AZERTY), il est nécessaire d'utiliser la commande `mapkey` suivi du nom du fichier de codage correspondant (ce fichier se trouve dans le répertoire `/usr/lib/keyboard`).

Ainsi pour obtenir la gestion scancode d'un clavier français, tapez la commande suivante :

```
$ mapkey /usr/lib/keyboard/ps.ibm.fra
```

Pour simplifier la tâche de l'utilisateur, il est possible de regrouper les deux commandes `scanon` et `mapkey` dans un script.

Utilisez la commande `scanoff` pour repasser en gestion ASCII.

2.4.3 - Déclaration pour l'utilisation de Word

Le logiciel Word peut gérer le changement de gestion clavier au travers d'un fichier paramètre (`/usr/lib/word/termscan`) similaire par son contenu au fichier `/etc/termcap`.

Pour pouvoir utiliser Word depuis un terminal avec un clavier géré en scancode, il est nécessaire de modifier le contenu de l'entrée **ansi** du fichier `termscan`.

Editez le fichier et ajoutez la ligne suivante (ni au début, ni à la fin du descriptif ansi) :

```
ZO=\E[<0A:ZF=\E[<1A:XN=\145:XF=\147:\
```

Sortez de l'éditeur en sauvegardant ces modifications.

Le terminal sera ainsi utilisé avec une gestion clavier ASCII pour toutes les applications, seul le lancement du logiciel Word initialisera la gestion scancode. A la sortie du logiciel Word, la gestion ASCII sera à nouveau rétablie.

- 3 -
MISE EN ŒUVRE
SOUS IBM AIX

Ce chapitre décrit les opérations à effectuer pour la mise en œuvre des différentes fonctionnalités de la Platine Terminal AXEL sous le système AIX IBM RS6000.

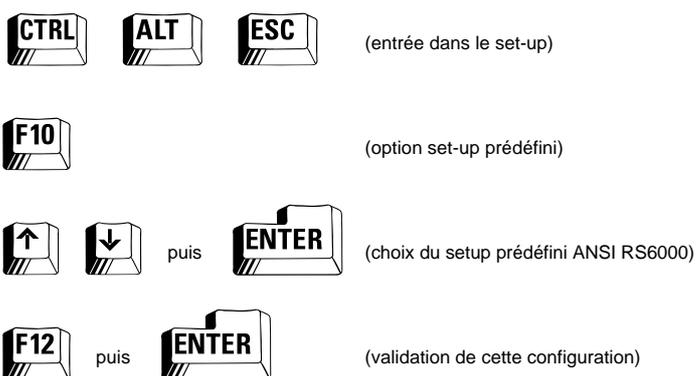
3.1 - INSTALLATION LOGICIELLE

L'installation d'une Platine Terminal AXEL se déroule en deux phases :

- configuration du set-up,
- configuration du système d'exploitation.

3.1.1 -Paramétrage du set-up

La première étape de l'installation d'une Platine Terminal AXEL est la mise à jour du set-up. Grâce à la fonction de **set-up prédéfini**, cette opération peut être effectuée très rapidement :



La Platine AX3000 est alors complètement configurée selon le **set-up prédéfini** choisi, et est prête à fonctionner sous les systèmes AIX IBM RS6000 à la vitesse de 38400 bauds. Si nécessaire, les éléments composant ce paramétrage automatique peuvent être modifiés individuellement (changement de la vitesse par exemple).

Pour plus d'informations concernant le set-up de la Platine Terminal AXEL, consultez *Le guide de l'utilisateur*.

3.1.2 - Configuration système

La Platine AX3000 se déclare au sein du système AIX IBM comme n'importe quel terminal asynchrone.

Pour ce faire, il est important de donner à la variable d'environnement TERM une valeur correspondant à l'émulation choisie au niveau du set-up de la Platine.

L'émulation ANSI RS6000 utilisée par la Platine Terminal est de type **ansi couleur**. La valeur de la variable TERM doit être :

- **hft-c** pour IBM AIX 3.1,
- **hft-c-old** pour IBM AIX 3.2 et IBM AIX 4.x.

La déclaration des paramètres d'un terminal s'effectue au moyen de l'utilitaire `smit`, dans le menu consacré aux TTYs.

Lancez la commande suivante :

```
# smit tty
```

Sélectionnez l'option '**Ajout TTY**' (ou '**Modif/Affich caractéristiques TTY**' pour modifier les caractéristiques d'une voie déjà déclarée).

Après avoir sélectionné le type de liaison (RS232 ou RS422), la carte parente et le numéro du port, l'écran de saisie principal apparaît.

Parmi tous les paramètres requis dans cet écran de saisie, voici la valeur de ceux correspondant au set-up prédéfini ANSI RS6000 de la Platine AXEL :

- Vitesse de transmission : 38400 bauds
- Parité : none
- Bits par caractère : 8
- Nombre de bits d'arrêt : 1
- Type de terminal : hft-c (ou hft-c-old)

Après saisie de ces nouvelles valeurs, appuyez sur **<RC>** pour valider les modifications et revenir au menu précédent de `smit`.

La configuration au niveau du système AIX IBM RS6000 est terminée.

Note : les paramètres de communication (la vitesse par exemple) peuvent être modifiés à condition de le faire, d'une part au niveau du set-up de la Platine AX3000, et d'autre part au niveau de l'utilitaire `smit`.

3.1.3 - Utilisation des caractères composés

Définition : un caractère composé est un caractère obtenu par l'appui successif sur plusieurs touches (en général deux ou trois). Ainsi le caractère 'â' est obtenu par l'appui sur la touche '^' suivi de la touche 'a'.

Note : cette fonction peut être gérée en local par la Platine (option **<F2>** du set-up) ce qui évite un paramétrage au niveau système d'exploitation. Pour plus d'information, consultez le *Guide Utilisateur*.

Le codage des caractères composés s'effectue au travers d'un fichier paramètre. Ce fichier doit être créé dans le répertoire `/etc/nls/termmap`. En général ce type de fichier est suffixé par `'.ti'`.

Au moyen de l'éditeur de texte `vi`, créez le fichier `ax3000.ti` dans le répertoire `/etc/nls/termmap`. Voici le contenu de ce fichier :

```
# â ê î ô û
\xb0\x61:\x83
\xb0\x65:\x88
\xb0\x69:\x8c
\xb0\x6f:\x93
\xb0\x75:\x96
# à è ì ò ù
\x60\x61:\x85
\x60\x65:\x8a
\x60\x69:\x8d
\x60\x6f:\x95
\x60\x75:\x97
# ä ë ï ö ü ÿ Ä Ö Ü
\xb1\x61:\x84
\xb1\x65:\x89
\xb1\x69:\x8b
\xb1\x6f:\x94
\xb1\x75:\x81
\xb1\x79:\x98
\xb1\x41:\x8e
\xb1\x4f:\x99
\xb1\x55:\x9a
# ñ Ñ
\x7e\x6e:\xa4
\x7e\x4e:\xa5
```

Deux méthodes sont possibles pour utiliser ce fichier et obtenir ainsi les caractères composés.

a - L'utilitaire `setmaps`

C'est la méthode la plus simple. Il suffit de lancer la commande suivante au début de chaque session :

```
$ setmaps -i ax3000.in
```

Cette commande peut être insérée dans le script du login (fichier `.profile`).

b - L'utilitaire smit

Cette méthode permet d'affecter définitivement le fichier `ax3000.in` à la **voie tty** qu'utilise la Platine Terminal.

L'utilitaire `smit` n'accepte que certains noms de fichiers paramètres. Comme le fichier `ax3000.in` n'est pas reconnu en standard, il faut lui donner le nom d'un fichier reconnu par `smit` (par exemple `vt220.in`).

Pour cela, lancez les commandes suivantes :

```
# cd /usr/nls/termmap
# mv vt220.in vt220.SAV
# mv ax3000.in vt220.in
```

Puis lancez l'utilitaire :

```
# smit tty
```

Accédez à l'écran de modification de la voie tty utilisée par la Platine Terminal et modifiez la rubrique 'Fichier mappe d'entrée' en lui attribuant `vt220.in` :

```
.....      ....
Fichier mappe d'entrée      [vt220.in]
Fichier mappe de sortie     [none]
.....      ....
```

Sauvegardez ces modifications en tapant **<RC>**.

3.2 - GESTION DE L'IMPRIMANTE LOCALE

Une imprimante locale est une imprimante directement connectée à un terminal. Cela permet d'avoir une imprimante sur le même site que le terminal sans utiliser de voie série supplémentaire.

3.2.1 - Installation d'une imprimante locale

L'installation d'une imprimante locale sur une Platine AXEL sous IBM AIX est conforme aux indications données au chapitre 1.3 de ce document.

3.2.2 - Utilisation de l'imprimante locale

Pour les versions IBM AIX 3.1, la notion d'imprimante locale reconnue comme imprimante système n'existe pas. L'utilisation de l'imprimante locale est donc conforme aux indications données au chapitre 1.3 de ce document.

Pour les versions IBM AIX 3.2 et suivantes, la notion d'imprimante locale reconnue comme imprimante système est intégrée au système d'exploitation.

La déclaration d'une imprimante locale est effectuée grâce à l'utilitaire `smit`. Une file d'attente est ainsi associée au device (tty) utilisé par la Platine.

Pour qu'une telle imprimante fonctionne, l'émulation **terminfo** doit inclure les directives `mc5` et `mc4` représentant respectivement les séquences de connexion et de déconnexion de l'imprimante locale.

L'émulation **hft-c** (ou `hft-c-old`) ne reconnaissant pas ces deux séquences en standard, le fichier `/usr/share/lib/terminfo/ibm.ti` (décrivant les émulations **hft-c**) doit être modifié.

- Placez vous dans le répertoire `/usr/share/lib/terminfo`
- Editez le fichier `ibm.ti` avec l'éditeur de texte `vi`.

- Recherchez l'entrée hft-c-old :

```
hft-c-old,xxxxx,xxxx,  
xx,xxxx,xxx,  
xxx,xx,xxx,  
xx,xxx,xxx,
```

- Insérez la ligne suivante dans la description (ni au début, ni à la fin du descriptif hft-c-old) :

```
mc5=\E[5i, mc4=\E[4i,
```

- Quittez vi en sauvegardant ces modifications.

- Les modifications effectuées dans `ibm.ti` doivent être recompilées :

```
# tic ibm.ti
```

L'imprimante est alors prête à être utilisée au moyen de la commande `lp`.

**- 4 -
LE MULTISCREEN**

Ce chapitre décrit l'installation et l'utilisation du logiciel de multiscreen axmscreen fourni avec la Platine.

4.1 - PRESENTATION

Le multiscreen est une fonctionnalité qui permet d'avoir, sur un même poste de travail, jusqu'à 8 sessions accessibles simultanément par simple combinaison de touches. Cette fonctionnalité est disponible sur les Platines Terminal AXEL grâce au logiciel `axmscreen`.

Cette fonctionnalité de multiscreen est utilisable sous deux formes :

- le multishell : sur chacune des 8 sessions est lancé un shell, ayant les mêmes droits que le shell d'où a été exécuté `axmscreen`,
- le multilogin : sur les 8 sessions est demandé un login (nom utilisateur et mot de passe), permettant ainsi d'avoir un environnement différent sur chacune des vues.

Plusieurs versions du logiciel `axmscreen` sont disponibles, correspondant à autant de types d'UNIX différents. Le logiciel `axmscreen` fonctionne déjà sur :

- UNIX SCO (XENIX, 3.2v2, 3.2v4 et Release 5),
- IBM AIX,
- UNIX AT&T (Interactive, Unix WARE...),
- HP 9000,
- SINIX (Siemens),
- AViiON (Data Général),
- MOTOROLA.

Ce service de multiscreen est basé sur l'utilisation de pseudo-terminaux (pty). Leur déclaration au sein du noyau UNIX est indispensable pour le fonctionnement du logiciel `axmscreen`.

A noter que le logiciel `axmscreen` ne fonctionne que sur les Platines Terminal AXEL car il utilise la capacité de la Platine à sauvegarder le contexte de plusieurs écrans.

4.2 - INSTALLATION

Loggez-vous comme utilisateur `root` et tapez les commandes suivantes :

```
# cd /tmp
# tar xvf /dev/fdx ax*
```

(où `fdx` est le device du lecteur de disquette)

Les fichiers suivants sont copiés sur disque :

- <code>axmscreen.AIX</code>	: IBM AIX
- <code>axmscreen.SCO3</code>	: XENIX et UNIX SCO 3.2v2
- <code>axmscreen.SCO4</code>	: UNIX SCO 3.2v4 et Release 5
- <code>axmscreen.SVR4</code>	: UNIXWARE
- <code>axmscreen.DG</code>	: AViiON
- <code>axmscreen.SNX</code>	: SINIX
- <code>axmscreen.HP</code>	: HP 9000
- <code>axmscreen.MOTO88</code>	: MOTOROLA 8800
- <code>axmscreen.MOTO68</code>	: MOTOROLA 6800
- <code>axmscreen.INTER</code>	: UNIX INTERACTIVE

Copiez le fichier correspondant au système d'exploitation utilisé sous le nom `axmscreen` dans le répertoire `/usr/bin`. Exemple :

```
# cp axmscreen.AIX /usr/bin/axmscreen
```

Le logiciel est immédiatement opérationnel, car étant dédié à la Platine Terminal AXEL, aucun fichier de configuration n'est nécessaire.

4.3 - UTILISATION

La commande `axmscreen` peut être lancée, soit directement à partir du prompt Unix, soit insérée dans un fichier script (`.profile` par exemple), suivie des éventuelles options (voir chapitre suivant).

Après l'activation du logiciel `axmscreen`, le changement de vue s'effectue par les combinaisons de touches suivantes :

- **<Alt><F1>** : vue 1
- **<Alt><F2>** : vue 2
- ...
- **<Alt><F8>** : vue 8
- **<Ctrl><ImprEcran>** : passage à la vue suivante (vue 1 à vue 2, vue 2 à vue 3, ..., vue 8 à vue 1).

Les autres combinaisons de touches possibles sont :

- **<Alt><F9>** (`who`) : affiche le numéro de la vue courante (1 à 8) ainsi que le nom du pseudo-terminal lié à cette vue.
- **<Alt><F10>** (`help`) : affiche la liste des combinaisons de touches valides et les commandes correspondantes.
- **<Alt><F11>** (`quit`) : permet de quitter le logiciel **axmscreen** avec un status de retour égal à 1.
- **<Alt><F12>** (`exit`) : permet de quitter le logiciel **axmscreen** avec un status de retour égal à 0.

Note : ces combinaisons de touches peuvent être changées au moyen de l'option **<Alt><F2>** du set-up de la Platine.

4.4 - LES OPTIONS

La syntaxe de lancement de `axmscreen` est la suivante :

```
axmscreen [-bcefhklmnpstvw] [file] [x]
```

La suite du chapitre détaille chacune de ces options.

4.4.1 - Option b : choix des pttys

Le logiciel `axmscreen` est basé sur l'existence des pseudo-terminaux (ptty). Un pseudo-terminal est un device dont le nom peut varier selon les systèmes d'exploitation.

Pour pallier à d'éventuels problèmes, l'option '-b' permet de paramétrer le nom générique des pseudo-terminaux du système d'exploitation utilisé.

L'option '-b' doit être suivie d'un paramètre (trois valeurs possibles) :

- '-b 0' : utilisation des `ttypx` (x exprimé en décimal varie de 0 à 63),
- '-b 1' : utilisation des `ttypx` (x en hexadécimal varie de 0 à 3F),
- '-b 2' : utilisation des `ttypx`, `ttyqx`, `ttyrx` et `ttysx` (x exprimé en hexadécimal varie de 0 à F).

Par défaut, le comportement du logiciel est celui de l'option '-b 0'.

Note : cette option n'est pas disponible pour la version UNIX AT&T et est incompatible avec l'utilisation de l'option '-p' (voir chapitre 4.4.9)

4.4.2 - Option c : bordure d'écran

Une fois le logiciel `axmscreen` activé, la combinaison de touches <Alt><F9> permet d'identifier la vue active en affichant son numéro (1 à 8).

Une autre méthode permettant d'identifier la vue active, consiste à associer à chacune des vues une couleur de bordure d'écran. Ceci est obtenu grâce à l'utilisation de l'option '-c'.

Le tableau suivant donne la correspondance entre numéro de vue et couleur de bordure d'écran :

N° de vue	Couleur
1	Bleu
2	Jaune brillant
3	Vert
4	Cyan
5	Rouge
6	Magenta
7	Jaune clair
8	Blanc

4.4.3 - Option e : sortie interdite

La sortie du logiciel s'effectue par l'intermédiaire des combinaisons de touches <Alt><F11> ou <Alt><F12>. L'option '-e' permet d'interdire leur utilisation, pour éviter une sortie accidentelle du logiciel.

Si malgré tout, vous êtes amené à devoir supprimer les process créés par `axmscreen`, utilisez la commande suivante :

```
# kill -s 15 pid
```

où `pid` est le numéro du process père de `axmscreen` (dont le père est en général un shell).

Note : utilisez le signal 15 (et non pas le signal 9) avec la commande `kill`.

4.4.4 - Option f : commandes initiales

Dans le cas d'une utilisation du logiciel en multishell, il est possible de lancer automatiquement sur chacune des vues une commande initiale.

Cette liste de commandes (une par vue) est stockée dans un fichier dont le nom est passé en paramètre avec l'option '-f'.

Ce fichier de commandes contient 8 lignes maximum. Chaque ligne contient la commande affectée à sa vue :

1^{ère} ligne → vue 1
2^{ème} ligne → vue 2
... ..
8^{ème} ligne → vue 8

Si une vue n'a pas de commande initiale, la ligne qui lui est affectée doit commencer par un tiret (caractère 2Dh).

Exemple : le fichier `cmd`

```
smit  
-  
client
```

La prise en compte de ce fichier est effectué par la commande :

```
# axmscreen -f cmd
```

Dans cet exemple, l'utilitaire `smit` est lancé sur la vue 1 et l'application `client` sur la vue 3. Les vues 2, 4, 5, 6, 7 et 8 n'ont pas de commande initiale (c'est le shell qui est actif).

4.4.5 - Option h : aide en ligne

Cette option permet d'afficher la syntaxe de lancement du logiciel `axmscreen`, ainsi qu'une description sommaire de chacune des options.

4.4.6 - Option k : commentaire

Il est possible d'utiliser avec `axmscreen` une ligne status (voir ci-après option '-s') pour visualiser les vues disponibles et la vue active. L'option '-k' permet de spécifier une chaîne de caractères qui sera affichée sur la droite de cette ligne status.

Exemple :

```
NOM=`tty`  
axmscreen -n 4 -s -k $NOM
```

4.4.7 - Option I : multilogin

La fonctionnalité de multishell du logiciel `axmscreen`, peut être remplacée par celle de multilogin. C'est à dire que chacune des vues attend un login (nom utilisateur et mot de passe). Cette fonctionnalité est activée par l'option '-l'.

Note : il est impossible d'avoir sur un même terminal, des vues avec des logins et des vues avec des commandes initiales (multishell). Les options '-l' et '-f' sont incompatibles.

Le login étant un process un peu particulier, cette fonctionnalité de multilogin demande un traitement spécifique selon le système d'exploitation.

Note : la fonction de multilogin n'est pas disponible pour les versions de `axmscreen` sous SIEMENS et HP 9000.

a - Unix SCO 3.2v4 et Release 5

Le login est généré par le programme `/etc/utmp_getty`. Les droits d'utilisation de ce programme se trouvent dans le fichier `/tcb/files/no_luid/cmdtable`. Afin d'autoriser le multilogin à tous les utilisateurs, il est nécessaire de vérifier le contenu de ce fichier.

Editez le fichier `/tcb/files/no_luid/cmdtable`. Si la ligne qui donne les droits de `utmp_getty` est de la forme suivante :

```
utmp_getty:/etc/utmp_getty:root
```

vous devez remplacer le mot 'root' par '*' (caractère 2Ah). Quittez l'éditeur de texte en sauvegardant cette modification.

Note : dans le cas où la version du système d'exploitation est **3.2v4.0**, il est nécessaire d'associer l'option '-v' à la notion de multilogin. Pour les versions supérieures, cette option ne doit pas être spécifiée.

Exemple :

```
axmscreen -n 4 -l -v   pour Unix SCO 3.2v4.0
axmscreen -n 4 -l     pour Unix SCO 3.2v4.2
```

b - Unix AT&T

Le login sur ce système d'exploitation est effectué par le process 'ttypmon'. Ce process ne peut être lancé que par un programme disposant des droits maximum (utilisateur **root**).

L'obtention de ces droits est effectuée au moyen de la commande `su` lancée au début de l'exécution du logiciel.

Par défaut, cette commande demande le mot de passe de l'utilisateur **root**. Pour éviter ceci, il est nécessaire de modifier le fichier `/etc/default/su`. Au moyen de `vi`, éditez `/etc/default/su` et ajoutez à la fin du fichier, la ligne suivante :

```
PROMPT=No
```

Sauvegardez le fichier; il est alors possible d'utiliser l'option multilogin du logiciel `axmscreen`.

c - Autres Unix

La notion de login implique l'existence d'un process en 'background' attendant la saisie du nom utilisateur pour chacune des vues. Ce process se nomme `/etc/getty` et doit exister pour chacun des pseudo-terminals utilisés (il est facile de vérifier son existence avec la commande `ps -ef`).

Si les process `/etc/getty` ne sont pas présents pour les pseudo-terminals utilisés, le fichier `/etc/inittab` doit être modifié.

Le fichier `/etc/inittab` contient les noms des voies sur lesquelles un login doit être actif. Il faut donc ajouter dans ce fichier, le nom de TOUS les pseudo-terminals devant utiliser le login.

La structure des lignes à rajouter est la suivante :

```
ptyx:2:respawn:/etc/getty /dev/ttypx m
```

où x est le numéro du pseudo-terminal.

Une fois le fichier sauvegardé, tapez la commande `init q`, qui demande au système d'exploitation de prendre en compte ces modifications immédiatement.

Vérifiez le résultat des modifications par la commande `ps -ef`.

4.4.8 - Option m : mode muet

Lors du lancement de `axmscreen`, des messages d'aide sont automatiquement affichés (nombre de vues, touches actives...).

Il est possible de supprimer ces messages en utilisant l'option '-m'.

4.4.9 - Option n : nombre de vues

Par défaut, le logiciel `axmscreen` fournit jusqu'à 8 vues. Il est possible de limiter ce nombre de vues au moyen de l'option '-n x' où `x` est ce nombre de vues (de 1 à 8).

Exemple :

```
$ axmscreen -n 4
```

La combinaison de touches `<Ctrl><ImprEcr>` permet de passer automatiquement à la vue suivante (vue 1 à vue 2, ..., vue 8 à vue 1). Pour optimiser cette fonction de permutation de vue, il est nécessaire de déclarer au niveau de la Platine Terminal le nombre de vues réellement utilisées.

Le set-up de la Platine doit donc être modifié :



(entrée dans le set-up)



(sélection du paramètre 'nombre de vues multiscreen')



(sélection du nombre de vues)



4.4.10 - Option p : préaffectation des pseudo-terminals

L'option '-p' du logiciel `axmscreen` permet de choisir par avance les pseudo-terminals à utiliser pour chaque vue du multiscreen. Cette liste de pseudo-terminals est stockée dans un fichier dont le nom est passé en paramètre.

Le fichier de pseudo-terminals contient 8 lignes maximum. Chaque ligne contient le pty affecté à sa vue :

1^{ère} ligne → vue 1
2^{ème} ligne → vue 2
... ..
8^{ème} ligne → vue 8

Exemple : le fichier `predef`

```
ptyp2  
ptyp5  
ptyp6  
ptyp8
```

La prise en compte de ce fichier est effectué par la commande :

```
$ axmscreen -p predef
```

Avec ce fichier `predef`, 4 vues seront créées (`ptyp2` affecté à la vue 1, `ptyp5` affecté à la vue 2, `ptyp6` affecté à la vue 3 et `ptyp8` affecté à la vue 4).

Note : cette option n'est pas disponible pour la version UNIX AT&T et est prioritaire par rapport à l'option '-n' (qui fixe le nombre de vue). Ainsi, la commande '`axmscreen -n 8 -p predef`' ne crée que 4 vues.

4.4.11 - Option s : ligne status

L'option '-s' permet d'utiliser le logiciel `axmscreen`, avec une ligne status au bas de l'écran indiquant à la fois, le nombre de vues déclarées et la vue active.

Note : l'option '-k' offre la possibilité d'ajouter un commentaire sur cette ligne status (voir chapitre 4.4.6).

Généralement, les entrées **ansi** d'un fichier `terminfo` ou `termcap` décrivent le comportement de terminaux avec des écrans de 25 lignes. Pour utiliser la lignes status, il est nécessaire de créer une entrée **ansi de 24 lignes** (pour plus de précisions, consultez le chapitre 4.5.3).

4.4.12 - Option t : pas de test du tty

Par défaut, le logiciel `axmscreen` teste lors de son lancement, la nature du tty sur lequel il est exécuté. Si ce tty est reconnu comme étant un pseudo-terminal, le logiciel s'interrompt.

Ceci a été fait pour éviter que `axmscreen` soit lancé sur une vue de `axmscreen` !

Ce test peut être inhibé par l'utilisation de l'option '-t'. Cela est nécessaire dans le cas où la Platine est utilisée sur un serveur de terminal dans un réseau Ethernet (les voies du serveur de terminal sont en effet des pseudo-terminaux).

4.4.13 - Option v : Unix SCO 3.2v4.0

Option OBLIGATOIRE en complément de l'option '-1' (multilogin) sous Unix SCO 3.2v4.0. Pour plus de précisions, consultez le chapitre 4.4.7.

4.4.14 - Option w : non régénération de shell

Cette option est à utiliser en complément de la fonctionnalité de `multishell`.

Le fait de taper <Ctrl><D> (ou `exit`), durant le `multishell`, termine le shell en cours sur la vue active. Ce shell est automatiquement régénéré (avec sa commande initiale éventuelle) par le logiciel `axmscreen`.

Cette fonctionnalité de régénération de shell peut être désactivée au moyen de l'option '-w'.

Dans ce cas, un shell terminé n'est pas régénéré et la vue concernée devient inaccessible. La vue en question est libellée 'killed' au niveau de la commande 'who' du logiciel (combinaison de touches <Alt><F9>), et disparaît de l'éventuelle ligne status.

4.4.15 - Option x : buffer circulaire

Une vue inactive est une vue sur laquelle l'utilisateur ne travaille pas. Dans une session, si n vues sont utilisées, il y a, à tout moment, une seule vue active (la vue courante) et $n-1$ vues inactives.

Quel est le comportement d'une vue inactive ?

Par défaut, l'affichage de caractères sur une vue inactive est impossible. En effet, une vue inactive qui reçoit des caractères à afficher est rapidement bloquée (i.e. XOFF). Elle sera débloquée lorsqu'elle deviendra active.

Pour éviter ces blocages, il est prévu d'affecter à chacune des vues un buffer circulaire au moyen de l'option '-x'. Ainsi les caractères que reçoit une vue inactive sont stockés dans le buffer circulaire correspondant ; au moment où la vue est réactivée, le contenu du buffer est affiché.

La taille par défaut de ces buffers circulaires est de 2 Ko par vue. Il est possible de spécifier la taille au moyen de la syntaxe '-x n' où n est la taille en Ko des buffers par vue (n varie de 1 à 9).

A noter que cette gestion par buffer circulaire ne conserve que les dernières données envoyées à une vue inactive. La réactivation d'une vue peut donc faire apparaître un affichage incomplet (cf. chapitre 4.6).

4.5 - CONSEILS D'UTILISATION

4.5.1 - Multilogin ou Multishell ?

Le multilogin permet d'avoir sur chacune des vues un environnement UNIX différent (lié aux noms utilisateurs) alors que le multishell duplique le même shell sur les 8 vues.

Grâce à la notion de commandes initiales, le multishell permet le lancement automatique de différents logiciels sur chacune des vues

Sur la plupart des Unix, la notion de multilogin nécessite l'ajout de lignes dans le fichier `/etc/inittab`. La modification de ce fichier doit être très rigoureuse car le fichier `/etc/inittab` est utilisé lors du boot de la machine. De plus, les pseudo-terminaux spécifiés dans le fichier `/etc/inittab` doivent être les mêmes que ceux utilisés par `axmscreen` sous peine de dysfonctionnement (importance de l'option '-p').

4.5.2 - Création d'un utilisateur multivue

L'activation du multiscreen étant une simple ligne de commande, il est intéressant de placer celle-ci dans un script pour un lancement automatique.

Le principe est donc de créer un compte utilisateur dont le fichier `.profile` lance le logiciel `axmscreen` :

```
...
axmscreen -n 4 -s -f batch
exit
```

Ainsi, après la saisie du nom utilisateur et du mot de passe de ce compte, le terminal se trouve automatiquement en multiscreen. Lorsque l'utilisateur quitte le multiscreen (`<Alt><F11>` ou `<Alt><F12>`), le compte est fermé et une nouvelle demande de login apparaît.

4.5.3 - Ligne status

Les options '-s' et '-k' permettent la gestion d'une ligne status durant l'utilisation du multiscreen. Cette ligne status permet de connaître à tout moment le nombre de vues disponible et le numéro de la vue active.

Des problèmes d'émulation peuvent se poser avec l'utilisation de la ligne status.

Cette ligne status occupant la 25^{ème} ligne de l'écran, il ne reste plus que 24 lignes utiles pour les applications. Cela implique que le fichier `terminfo` utilisé spécifie le comportement d'un terminal de 24 lignes.

Si le système d'exploitation UNIX ne contient pas en standard un fichier `terminfo` décrivant un terminal de 24 lignes, il est nécessaire de modifier le fichier `terminfo` utilisé.

La description des émulations se situe, en général, dans un fichier suffixé par `.src` ou `.ti` dans le répertoire `/usr/lib/terminfo`.

Editez le fichier de description avec l'éditeur `vi`. Une fois la description de l'émulation trouvée, positionnez la variable fixant le nombre de lignes (`li#` ou `lines#` selon l'Unix) à 24. Sauvegardez ces modifications.

Le fichier ainsi modifié doit être recompilé. Lancez la commande suivante (où `nomfic` est le nom du fichier source modifié) :

```
# tic nomfic
```

Ce choix d'utiliser la 25^{ème} ligne de l'écran comme ligne status doit être en concordance avec la valeur du paramètre 'nombre de lignes' du set-up de la Platine Terminal.

Ce paramètre peut avoir deux valeurs :

- 25 lignes,
- 24 + 1 ligne message.

Le lancement du logiciel `axmscreen` avec l'option `'-s'`, force la mise à jour de ce paramètre à '24 + 1 ligne message'. Par contre en sortie du logiciel, l'ancienne valeur n'est pas restaurée.

Voici les opérations à effectuer pour modifier la valeur de ce paramètre :



(entrée dans le set-up)



(sélection du paramètre 'nombre de lignes')



(positionnement '25 lignes' ou '24 + 1 ligne message')



(validation de cette configuration)

4.5.4 - Quitter le multiscreen

La sortie du logiciel `axmscreen` s'effectue en standard au travers des combinaisons de touches `<Alt><F11>` ou `<Alt><F12>`.

Pour éviter une sortie involontaire, il est possible d'inhiber ces combinaisons (option `'-e'`). La sortie du logiciel est alors réalisée par la commande `kill` (signal 15 sur le process `axmscreen`).

Le script suivant (disponible sous le nom `end_axmscr` dans la disquette AXEL) recherche le numéro du process concerné et exécute la commande `kill` :

```
echo "*** axmscreen terminator"
if [ -z "$DAD_AX" ]; then
    echo -n "Enter the tty name:"
    read DAD_AX
fi
PID_AX=`ps -t $DAD_AX | grep "axmscr" | awk '{print $1}' `
if [ -z "$PID_AX" ]; then
    echo "Nobody to kill !!!"
else
    echo "kill -s 15 "$PID_AX
    kill -s 15 $PID_AX
fi
```

La variable `DAD_AX` peut être mise à jour dans le fichier `.profile` :

```
DAD_AX=`basename \`tty\``
```

Pour sortir de `axmscreen`, lancez le script `end_axmscr` depuis une des vues du logiciel de multisesion. Si la variable `DAD_AX` n'est pas renseignée, le nom du device sur lequel est connectée la Platine doit être saisi (`tty1a` par ex.).

IMPORTANT : ce script ne fonctionne que pour le mode `multishell`.

4.6 - LIMITATIONS DE LA NORME ANSI

L'émulation ANSI n'est pas réentrante. Cela signifie qu'une séquence escape 'coupée' par une autre séquence escape ou par des caractères quelconques est perdue ou provoque des anomalies d'affichage. En ANSI, les caractères d'une séquence escape non reconnue (car coupée ou incomplète) sont affichés.

Cette spécification de l'émulation ANSI peut perturber l'utilisation du logiciel `axmscreen` dans les deux cas suivants :

- **Le changement de vue** : si la séquence de changement de vue (combinaison `<Alt><Fx>`) coupe une séquence escape qui s'affiche sur une vue au moment où on la rend inactive, l'affichage peut être perturbé. La solution consiste à changer de vue lorsque l'écran est 'stable'.
- **L'utilisation de buffers (option '-x')** : après un changement de vue, le logiciel vide le buffer affecté à la nouvelle vue active. Ce buffer étant géré comme un buffer circulaire, il est possible que les premiers caractères de la première séquence escape à afficher soient absents. Les caractères de la deuxième moitié de cette séquence escape coupée seront alors affichés tel quels. Ceci peut se produire si les buffers circulaires utilisés ont une taille insuffisante. La solution consiste donc à augmenter cette taille au moyen de l'option `'-x n'`.



Zone d'activité d'Orsay-Courtabœuf
16 Avenue du Québec - BP 728 - 91962 LES ULIS Cedex
Tél. : 01.69.28.27.27 - Fax : 01.69.28.82.04