Bull

Concepts de Gestion du Système AIX
Système d’exploitation et unités

AIX

REFERENCE
86 F2 21KX 02

Bull

Bull

Concepts de Gestion du Système AIX
Système d’exploitation et unités

AIX

Logiciel

Mai 2000

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

REFERENCE
86 F2 21KX 02

Imprimé en France

Vos suggestions sur la forme et le fond de ce manuel seront les bienvenues. Une feuille destinée à recevoir vos remarques se trouve à la fin de ce document.

Pour commander d'autres exemplaires de ce manuel ou d'autres publications techniques Bull, veuillez utiliser le bon de commande également fourni en fin de manuel.

Marques déposées

Toutes les marques déposées sont la propriété de leurs titulaires respectifs.

AIX® est une marque déposée d'IBM Corp. et est utilisée sous licence.

UNIX est une marque déposée licenciée exclusivement par X/Open Company Ltd.

An 2000

Le produit documenté dans ce manuel est agréé pour l'An 2000.

A propos de ce manuel

Ce manuel fournit les informations permettant à l'administrateur d'assimiler les tâches quotidiennes de gestion du système d'exploitation AIX. Il présente également les outils AIX d'administration système. Utilisez ce manuel conjointement au document *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*, 86 F2 21KX.

Remarque : les informations contenues dans ce manuel sont également disponibles sur le "Hypertext Library for AIX 4.3" CD-ROM. Cette documentation en ligne est conçue pour être utilisée avec un navigateur web version 3.2 HTML compatible.

Utilisateurs concernés

Ce manuel est destiné aux personnes responsables de la gestion du système sur l'ordinateur et du système d'exploitation. Ces personnes sont censées en connaître les commandes de base.

L'administrateur est supposé familiarisé avec les informations et les concepts des documents suivants :

- *AIX 4.3 Guide de l'utilisateur : système d'exploitation et unités*, 86 F2 97HX
- *AIX 4.3 Guide de l'utilisateur : communications et réseaux*, 86 F2 98HX
- *AIX 4.3 Guide d'installation*, 86 F2 43GX

Mode d'emploi

La structure de ce manuel permet la recherche rapide des informations. Vous trouverez, dans l'ordre, les informations relatives au(x) :

- généralités sur les différents groupes et rubriques de tâches,
- tâches de configuration,
- tâches de maintenance,
- dépannage.

Remarque : Les sections relatives au dépannage sont utiles lorsque vous connaissez la cause du problème. Si vous rencontrez un problème dont vous ne connaissez pas la cause, reportez-vous au manuel *AIX Version 4.3 - Guide de résolution des incidents et références*.

Contenu du manuel

Ce manuel se compose des chapitres et annexes suivants :

- Le chapitre 1 "Gestion du système AIX", présente les principaux outils de gestion du système AIX et les caractéristiques spécifiques du système.
- Le chapitre 2 "Démarrage et arrêt du système", développe les concepts et les procédures de démarrage et d'arrêt du système.
- Le chapitre 3 "Protection du système", est consacré aux fonctions relatives à la sécurité, y compris TCB (Trusted Computing Base), la commande **virscan** (détection de virus), l'audit et le contrôle d'accès.
- Le chapitre 4 "Rôles administratifs", donne des indications sur les rôles et les droits ainsi que les procédures permettant de définir et de maintenir les rôles.

- Le chapitre 5 ”Administration des utilisateurs et des groupes”, développe les fonctions relatives à l’administration des utilisateurs et illustre les procédures de gestion de groupes d’utilisateurs.
- Le chapitre 6 ”Volumes logiques”, présente les concepts et procédures relatives à la gestion du stockage sur volumes logiques.
- Le chapitre 7 ”Systèmes de fichiers”, décrit les concepts et procédures de gestion des fichiers des répertoires et des systèmes de fichiers.
- Le chapitre 8 ”Espace de pagination et mémoire virtuelle,” développe les aspects pratiques de l’affectation d’espace de pagination, la création et la mise à jour de l’espace de pagination du système, et décrit le programme Virtual Memory Manager (gestionnaire de mémoire virtuelle).
- Le chapitre 9 ”Sauvegarde et restauration”, présente les commandes et concepts relatifs à la sauvegarde et à la restauration des données.
- Le chapitre 10 ”Environnement système”, développe les composants de l’environnement de base et leur exploitation. En outre, sont indiquées des instructions concernant la modification du message du jour, les messages diffusés aux utilisateurs, et l’utilisation des profils.
- Le chapitre 11 ”NLS”, fournit les indications nécessaires à la gestion du système dans les langues voulues.
- Le chapitre 12 ”Gestion des processus”, présente les processus système et leur exploitation.
- Le chapitre 13 ”SRC et sous-systèmes”, est dédié aux fonctions du contrôleur SRC et à son exploitation.
- Le chapitre 14 ”Comptabilité système”, présente les concepts servant à l’exploitation des commandes et des sous-routines du système de comptabilité.
- Le chapitre 15 ”Web-based System Manager” décrit le Web-based System Manager dans des environnements autonome et Client–Serveur.
- Le chapitre 16 “SMIT”, décrit l’exploitation et la structure de l’outil SMIT (System Management Interface Tool). SMIT est une interface utilisateur permettant de créer des commandes et de modifier les tâches de gestion de système. On peut l’utiliser soit en environnement ASCII, soit en environnement Windows.
- Le chapitre 17 ”CDE Desktop” décrit le CDE Desktop.
- Le chapitre 18 ”Service de recherche documentaire” vous permet de rechercher sur votre serveur de documentation des documents HTML en ligne qui ont été indexés. Cette section donne des informations sur l’installation et la configuration, ainsi que la création de vos propres index pour rechercher des documents créés par l’utilisateur.
- Le chapitre 19 ”Gestion de Power Management”, décrit la gestion système et les tâches utilisateur de Power Management.
- Le chapitre 20 ”Unités”, présente brièvement les méthodes employées par le système d’exploitation pour gérer nombre d’unités.
- Le chapitre 21 ”Unité de Bande”, est dédié aux fonctions de gestion des unités de bande.
- L’annexe A ”AIX pour administrateurs système BSD”, contient des informations à l’intention des administrateurs familiarisés avec le système d’exploitation 4.3 BSD UNIX ou System V. Ce chapitre décrit aussi bien les différences que les similitudes entre ces deux systèmes.

Conventions typographiques

Les conventions typographiques adoptées dans ce manuel sont les suivantes :

Gras	Commandes, sous-routines, mots-clés, fichiers, structures, répertoires et autres éléments dont le nom est prédéfini par le système, ainsi que les objets graphiques (tels que boutons, labels, icônes). Identifie également les objets graphiques (tels que boutons, labels, icônes).
Italique	Paramètres dont la valeur ou le nom est fourni par l'utilisateur.
`Espacement fixe`	Exemples de valeurs spécifiques, de texte affiché, de code programme, messages système, ou données entrées par l'utilisateur.

ISO 9000

Ce produit répond aux normes qualité ISO 9000.

Autres sources d'information sur la gestion de système

Documentation connexe

Un seul manuel ne prétend pas couvrir tout l'environnement informatique. Aussi, nous sommes-nous appliqués à présenter une bibliographie répondant aux besoins et aux centres d'intérêt des administrateurs système, axée sur les différents aspects de leur travail :

Voici une liste de documents traitant de sujets connexes :

- *AIX 4.3 Guide d'administration : communications et réseaux*, 86 F2 31JX, traitant de l'administration et de la maintenance de réseau.
- *AIX 4.3 Guide d'installation*, 86 F2 60AP
- Résolution des incidents et messages :
  - *AIX Version 4.3 - Guide de résolution des incidents et références*, 86 F2 32JX
  - *AIX - Guide des messages*, 86 F2 33JX
- *AIX General Programming Concepts : Writing and Debugging Programs*, 86 A2 34JX, présente les outils de programmation et les interfaces disponibles pour écrire et mettre au point les programmes d'application.
- *AIX Communications Programming Concepts*, 86 A2 35JX, donne des informations sur les concepts et les procédures relatives aux outils de programmation des communications.
- *AIX 4.3 Initiation*, 86 F2 75HX.
- Contrôle et ajustement des performances système :
  - *AIX - Guide d'optimisation*, 86 F2 72AP, décrit les outils de contrôle et d'ajustement des performances fournis dans la version de base du système d'exploitation.
  - *Performance Toolbox 1.2 and 2.1 for AIX: User's Guide*, 86 A2 10AQ, décrit les outils de contrôle supplémentaires fournis dans Performance Toolbox for AIX.
- *AIX 4.3 Guide d'administration : installation via un réseau*, 86 F2 17HX, traite de la configuration et de la maintenance des stations sans disque.
- *Distributed SMIT 2.2 for AIX: Guide and Reference*, 86 A2 09AQ, traite de DSMIT (Distributed System Management Interface Tool).
- *Common Desktop Environment 1.0: Advanced User's and System Administrator's Guide*, 86 A2 85AT, traite des tâches avancées de personnalisation de l'apparence et du comportement de l'environnement CDE (Common Desktop Environment).
- Object Data Manager (ODM) Overview dans *AIX General Programming Concepts : Writing and Debugging Programs*.

Support AIX pour la spécification X/Open UNIX95

Depuis AIX version 4.2, le système d'exploitation est conçu pour prendre en charge la spécification X/Open UNIX95 pour la portabilité des systèmes d'exploitation basés sur UNIX. Un certain nombre d'interfaces, dont certaines courantes, ont été ajoutées ou améliorées pour répondre à cette spécification. Depuis la version 4.2, AIX est encore plus ouvert et portable pour les applications.

En outre, la compatibilité avec les versions antérieures d'AIX est préservée. Et ceci, grâce à la création d'une variable d'environnement, qui permet de définir l'environnement du système pour chaque système, utilisateur ou process.

Pour connaître la meilleure façon de développer une application UNIX95 portable, reportez-vous à la spécification X/Open UNIX95, disponible sur CD-ROM en même temps que la copie papier de *AIX Commands Reference*, CEDOC 86 A2 38JX à 86 A2 43JX, ou en commandant *Go Solo: How to Implement and Go Solo with the Single Unix Specification*, qui inclut également cette spécification X/Open UNIX95 sur CD-ROM.

Bibliographie

Il est fait référence aux commandes et aux fichiers utilisés dans le système d'exploitation dans les documents suivants :

- *AIX Commands Reference*, 86 A2 38JX à 86 A2 43JX (document de 6 volumes répertoriant les commandes dans l'ordre alphabétique).
- *AIX Files Reference*, 86 A2 79AP (informations relatives aux fichiers disponibles dans le système d'exploitation).

Les manuels suivants contiennent des informations utiles quant à la gestion quotidienne :

- *AIX - Aide-mémoire*, 86 F2 55AP (description succincte des commandes courantes, avec un bref récapitulatif).
- *AIX - Guide de l'utilisateur : éditeur INed* (description de l'éditeur Ined).

Commande de documentation

Pour commander ce CD-ROM, adressez-vous à Bull Electronics Angers S.A CEDOC, à l'adresse indiquée sur le formulaire de remarques, à la fin de ce manuel.

Pour commander d'autres exemplaires de ce manuel, précisez la référence CEDOC 86 F2 21KX.

Reportez-vous à *AIX - Bibliographie* pour plus d'informations sur les manuels susceptibles de vous intéresser et sur la façon de vous les procurer.

Table des matières

Chapitre 1. Gestion du système AIX

La personne chargée de la gestion du système est l'administrateur système (ainsi appelé dans la littérature UNIX). Malheureusement, seule une faible part des activités de l'administrateur système sont suffisamment simples pour être définies comme étant des tâches d'administration. Ce manuel et la documentation connexe ont pour objectif de les assister dans leurs nombreuses fonctions.

Objectifs de l'administrateur système

Voici les trois principaux objectifs de l'administrateur système :

- veiller à l'efficacité du système,
- assurer la sécurité des données,
- gérer les règles d'exploitation du système définies par son propriétaire.

Pour atteindre ces objectifs, les connaissances de l'administrateur relatives à la structure et aux interactions matérielles/logicielles doivent dépasser le cadre de son champ d'activité habituel. L'administrateur doit en effet maîtriser l'environnement interconnecté de la plupart des systèmes actuels et l'impact de cet environnement sur les fonctions et les performances du système local.

Concept de système

Pour satisfaire les besoins de l'utilisateur, un système est constitué d'un certain nombre de composants matériels, logiciels et de données, l'ensemble étant interactif. En voici les principaux éléments, y compris les fonctions de gestion :

- Unités de disque contrôlant :
  - le regroupement et la subdivision de l'espace disque,
  - l'emplacement des données et des programmes pour optimiser les performances,
  - la taille de l'espace affecté à différentes opérations.
- Programmes d'application dédiés :
  - au contrôle de programmes sensibles et coûteux,
  - à l'installation et à l'ajustement des principales applications.
- Données d'application contrôlant :
  - l'accès aux données sensibles,
  - la mise en œuvre des copies de sauvegarde.
- Mémoire et processeurs individuels dédiés :
  - à la garantie du respect des priorités de l'entreprise quant à l'exploitation des ressources,
  - au contrôle de l'accès au système par les utilisateurs et les groupes,
  - à la mise au point du système d'exploitation pour optimiser l'exploitation des ressources disponibles.
- Réseaux locaux dédiés :
  - à la garantie de la mise au point des réseaux pour obtenir des performances optimales,
  - au contrôle des mécanismes d'adressage du réseau.

- Terminaux locaux dédiés au contrôle :
  - de l'interconnexion terminaux/processeurs,
  - de la configuration des terminaux et des processeurs pour garantir des performances optimales.
- Connexions aux autres réseaux pour assurer :
  - une configuration adéquate des ponts et des passerelles vers les autres réseaux,
  - la compatibilité de l'interaction entre les réseaux distants et les systèmes locaux.
- Accès aux/à partir des systèmes distants pour contrôler :
  - les autorisations d'accès dans les deux sens,
  - réguler la charge de travail imposée par les connexions distantes.
- Accès à des données distantes pour contrôler :
  - les méthodes et droits d'accès.

Gestion de système propre à AIX

AIX fournit sa propre version spécifique d'aide à la gestion de système, axée sur la facilité d'emploi, l'amélioration de la sécurité et de l'intégrité. Ces caractéristiques uniques sont présentées dans ce chapitre, à savoir :

- les interfaces disponibles,
- les fonctions uniques du système d'exploitation,
- la commande **man**.

Interfaces disponibles

AIXfournit en option, en plus de l'administration standard de type ligne de commande, les interfaces suivantes :

- SMIT (System Management Interface Tool), interface utilisateur permettant de créer et d'exécuter des commandes à partir d'options, et d'assurer :

  L'interface SMIT vous permet d'assurer :

  - l'installation, la mise à jour et la maintenance du logiciel,
  - la configuration des unités,
  - la configuration des unités de disque pour le stockage en groupes de volumes et en volumes logiques,
  - la création et l'extension de systèmes de fichiers et d'espaces de pagination,
  - la gestion des utilisateurs et des groupes,
  - la configuration des réseaux et des applications de communication,
  - l'impression,
  - l'identification des incidents,
  - l'organisation des travaux,
  - Gère les environnements système.

  Reportez–vous à la section SMIT (System Management Interface) – généralités Pour plus de détails, reportez-vous au chapitre 16. SMIT, page 16-1.

- DSMIT (Distributed System Management Interface Tool), interface utilisateur ASCII, permettant d'exécuter des tâches d'administration système sur des "grappes" de stations de travail, y compris des machines Sun/OS 4.1.3 et HP/UX 9.0. DSMIT est un produit en option. Ce produit peut être vendu séparément. Pour plus de détails, reportez-vous au manuel *Distributed SMIT 2.2 for AIX: Guide and Reference*.

Fonctions uniques du système d’exploitation

Ces fonctions sont présentées ci-après de façon succincte.

Logical Volume Manager (LVM)

LVM permet aux volumes logiques d’étendre plusieurs volumes physiques. Pour l’utilisateur, les données des volumes logiques semblent contiguës, mais ne le sont pas nécessairement sur le volume physique. Avec cette fonction, la taille et l’emplacement des systèmes de fichiers, de l’espace de pagination, et des autres volumes logiques peuvent être modifiés.

Pour plus de détails, reportez-vous à ”Stockage sur volumes logiques - généralités”, page 6-2.

System Resource Controller (SRC)

SRC fournit un ensemble de commandes et de sous-routines de création et de contrôle des sous-systèmes. Dans le traitement informatique, sa conception permet une intervention humaine minimale. Il fournit un mécanisme de contrôle des processus du sous-système opérant à partir d’une ligne de commande standard et avec l’interface C. Par le biais de scripts shell, de commandes ou de programmes utilisateur, vous pouvez lancer, arrêter ou réunir des informations sur l’état des process sous-système.

Pour plus de détails, reportez-vous à ”SRC - généralités”, page 14-2.

Object Data Manager (ODM)

ODM est un gestionnaire de données dédié au stockage des données du système. Nombre de fonctions de gestion système font appel à ODM. Les données servant à nombre de fonctions SMIT et de commandes sont stockées et actualisées comme des objets, avec leurs caractéristiques associées. Les données système gérées par ODM comprennent :

- des informations sur la configuration des unités ;
- des menus, des sélecteurs et des dialogues d’écrans SMIT ;
- des VPD (données techniques essentielles) pour les procédures d’installation et de mise à jour ;
- des informations relatives à la configuration des communications ;
- des informations relatives aux ressources système.

Base des données techniques essentielles (SWVPD)

Certaines informations concernant les logiciels et leurs options installables sont actualisées dans la base de données SWVPD (données techniques essentielles du logiciel). Cette base est constituée d’un ensemble de commandes et de classes d’objets ODM, dédiées à la maintenance des données relatives au logiciel. Par le biais de ces commandes, l’utilisateur peut formuler des requêtes (commande **lslpp**) et vérifier (commande **lppchk**) les produits logiciels installés. Les classes d’objets ODM définissent la portée et le format des données actualisées.

La commande **installp** fait appel à ODM pour actualiser dans la base de données SWVPD :

- le nom du logiciel installé ;
- sa version ;
- son niveau d’édition, indiquant les modifications de son interface de programmation externe ;
- son niveau de modification, indiquant les modifications n’affectant pas l’interface de programmation externe ;
- son niveau de correction, indiquant les mises à jour mineures qui feront ultérieurement l’objet d’un nouveau niveau de modification ;
- l’identification de la correction ;

- les noms, les totaux de contrôle, et la taille des fichiers constituant le logiciel ou l'option ;
- l'état d'installation du logiciel : disponible, en cours d'application, appliqué, en cours de validation, validé, en cours de rejet ou retiré.

Commande man

La commande **man** sert essentiellement à rechercher des informations sur les commandes, les sous-routines et les fichiers. Par exemple, si vous souhaitez des informations sur la commande **gprof**, entrez :

```
>man gprof
```

La plupart des informations affichées proviennent de fichiers HTML formatés. Pour trouver une explication sur un indicateur ou la syntaxe d'une commande, nombre d'administrateurs système font de préférence appel à la commande **man** plutôt que de démarrer une session avec le navigateur web.

Pour plus de détails sur la commande **man**, reportez-vous au manuel *AIX Commands Reference*. Reportez–vous également à la commande **man** pour administrateurs système BSD 4.3, page A-25.

AIX Mises à jour

Vous trouverez des informations détaillées sur les mises à jour de logiciel, ou mises à jour de service à la section "Installation de logiciels en option et de mises à jour de service" du manuel *AIX Installation Guide*.

Chapitre 2. Démarrage et arrêt du système

Ce chapitre est consacré aux activités de démarrage du système, notamment l'amorçage, la création d'images ou de fichiers d'amorçage et la définition du niveau d'exécution du système. L'arrêt du système, au moyen des commandes **reboot** et **shutdown**, est également traité.

Les sujets abordés sont les suivants :

Démarrage du système

A l'amorçage du système d'exploitation de base, le système lance un ensemble complexe de tâches. En conditions normales d'exploitation, ces tâches s'exécutent automatiquement. Pour plus d'informatins sur l'amorçage du système, reportez–vous aux sections suivantes :

- Description du processus d'amorçage, page 2-3
- Identification des problèmes d'amorçage dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Amorçage du système

Vous demandez l'amorçage du système dans certaines situations : par exemple, pour entériner l'installation d'un nouveau logiciel, pour réinitialiser les unités périphériques, pour exécuter des routines de maintenance (telles que la vérification des systèmes de fichiers), ou pour relancer le système après une panne ou une interruption. Vous trouverez des informations sur ces procédures aux sections :

- Amorçage d'un système non installé dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.
- Réamorçage d'un système en cours d'exploitation dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.
- Amorçage à l'issue d'une panne système dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Création d'images d'amorçage

Lors de la première installation, la commande **bosboot** crée une image d'amorçage à partir d'une image du système de fichiers disque en RAM et du noyau du système d'exploitation. L'image d'amorçage est transférée sur un support donné, un disque dur par exemple. Au réamorçage, l'image d'amorçage est chargée en mémoire à partir de son support.

Pour plus de détails, reportez-vous à "Création d'images d'amorçage" dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Identification et modification du niveau d'exécution du système

Le niveau d'exécution spécifie l'état du système et définit les processus à démarrer. Par exemple, avec un niveau 3, tous les processus définis pour opérer à ce niveau sont lancés. Juste avant la fin de phase d'amorçage système du processus d'amorçage, le niveau d'exécution est lu dans l'entrée initdefault du fichier **/etc/inittab**. Il peut être modifié par le biais de la commande **init**. Le fichier **/etc/inittab** contient un article par processus, qui définit les niveaux d'exécution. A l'amorçage du système, la commande **init** lit le fichier **/etc/inittab** pour définir les processus à démarrer. Vous trouverez des informations sur ces procédures aux sections :

- Identification des niveaux d'exécution dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.
- Modification du niveau d'exécution dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.
- Modification du fichier /etc/inittab dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Description du processus d'amorçage

Lors du processus d'amorçage, le système teste le matériel, charge puis exécute le système d'exploitation et configure les unités. L'amorçage du système d'exploitation nécessite les ressources suivantes :

- une *image d'amorçage* chargeable après la mise sous tension ou la réinitialisation de la machine,
- l'accès aux systèmes de fichiers racine et **/usr**.

Il existe trois types d'amorçage :

A partir d'un disque	La machine est démarrée pour être exploitée dans des conditions normales avec le sélecteur de mode en position normale. Pour plus de détails, reportez-vous à "Description du traitement de l'amorçage", page 2-4.
A partir d'un réseau sans disque	La station de travail sans disque ou sans données est démarrée à distance, par l'intermédiaire du réseau. La machine est démarrée pour être exploitée dans des conditions normales avec le sélecteur de mode en position normale. Un ou plusieurs serveurs de fichiers distants fournissent les fichiers et les programmes nécessaires.
Amorçage de maintenance	La machine est démarrée à partir d'un disque, d'un réseau, d'une bande ou d'un CD–ROM, avec le sélecteur de mode en position de maintenance. On parle, dans ce cas, d'amorçage en *mode maintenance*. En mode maintenance, l'administrateur système peut exécuter différentes tâches, par exemple, installer un nouveau logiciel ou une mise à jour, et lancer des tests de diagnostic. Pour plus de détails, reportez-vous à "Description du processus d'amorçage de maintenance", page 2-9.

Pendant l'amorçage à partir d'un disque, le système recherche l'image d'amorçage sur un disque local défini à l'installation du système d'exploitation. Au cours du processus, le système configure toutes les unités trouvées dans la machine et initialise le logiciel de base nécessaire pour l'exploitation du système, tel que LVM (Logical Volume Manager). En fin de processus, les systèmes de fichiers sont montés et prêts à l'emploi. Pour plus de détails sur le système de fichiers utilisé lors de l'amorçage, reportez-vous à "Description du système de fichiers RAM", page 2-10.

Ce traitement s'applique également, dans les grandes lignes, aux clients de réseau sans disque. Ils requièrent également une image d'amorçage et l'accès à l'arborescence du fichier système d'exploitation. Ce type de client sans disque ne possédant pas de système de fichiers local y accède à distance.

Description du traitement de l’amorçage

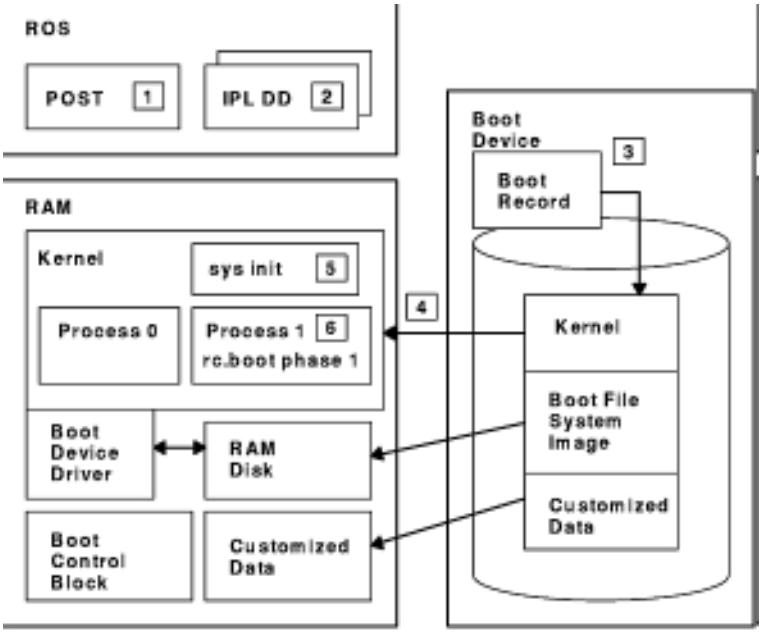
Lors du démarrage du système pour une exploitation dans des conditions normales, l’amorçage s’effectue généralement à partir du disque. Le système trouve sur le disque toutes les données nécessaires au processus d’amorçage.

Lorsque le système est démarré par une mise sous tension (“à froid”) ou redémarré par la commande **reboot** ou **shutdown** (“à chaud”), un certain nombre d’événements ont lieu avant qu’il ne soit prêt à l’emploi. Ces événements sont répartis en trois phases :

1. Initialisation du noyau ROS (Read Only Storage),
2. Configuration des unités de base,
3. Amorçage du système.

Phase d’initialisation du noyau ROS

La figure ci-après illustre la phase d’initialisation du noyau ROS, qui est antérieure au lancement du processus d’amorçage du système.



ROS Kernel Init Phase

Cette phase comprend les étapes suivantes :

1. Le micro-processeur BUMP (Bring–Up MicroProcessor) vérifie la carte système mère. Le contrôle passe au ROS, qui exécute le POST (Power–On Self–Test = test à la mise sous tension).
2. L’IPL (Initial Program Load = chargement initial du programme) ROS vérifie la liste d’amorçage utilisateur, qui répertorie les unités d’amorçage disponibles. Vous pouvez adapter cette liste à vos besoins au moyen de la commande **bootlist**. Si la liste en NVRAM n’est pas valide, ou si le système ne trouve pas d’unité d’amorçage valide, c’est la liste d’amorçage par défaut qui est vérifiée. Dans les deux cas, la première unité d’amorçage valide rencontrée est utilisée pour le démarrage du système. Les unités sont vérifiées dans l’ordre de la liste (à condition qu’elle soit valide) située, le cas échéant, au niveau de la NVRAM. A défaut de liste, toutes les cartes et unités du bus sont vérifiées.

Dans les deux cas, les unités sont vérifiées en boucle (continue) jusqu'à trouver une unité d'amorçage valide.

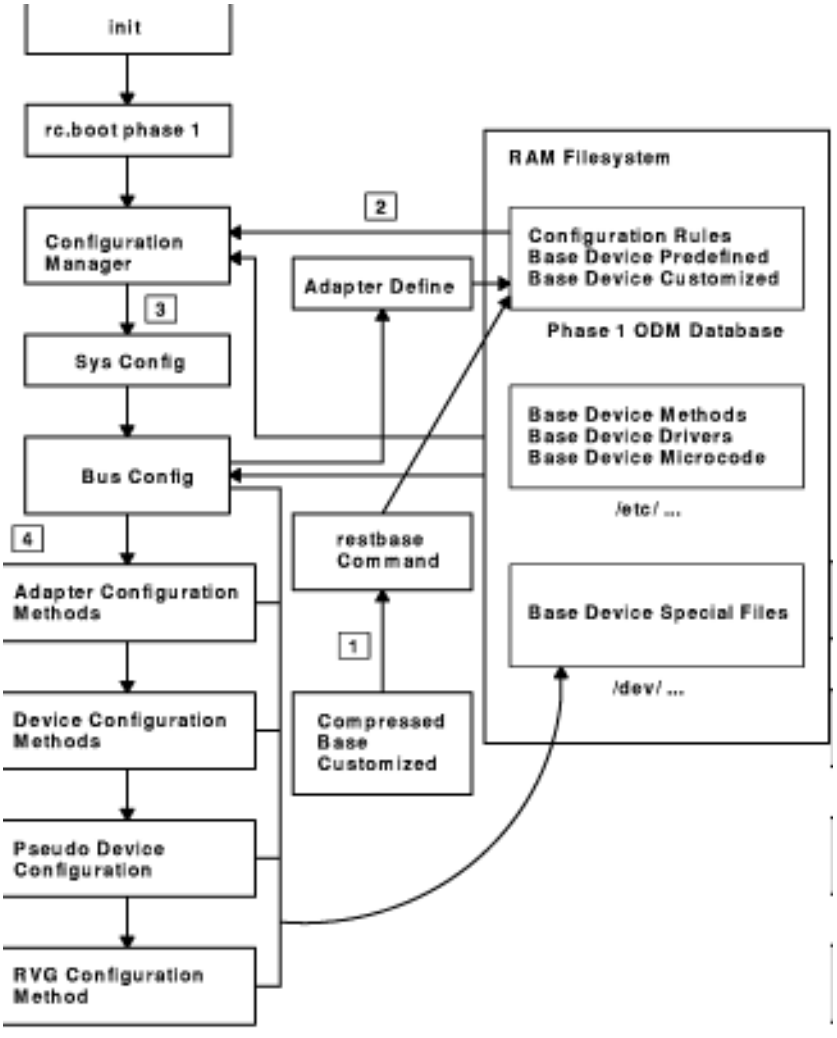
Remarque : Le système se charge de la maintenance d'une liste d'amorçage par défaut, située en ROS, et d'une liste d'amorçage utilisateur, stockée en NVRAM, pour l'amorçage normal. Font aussi l'objet de maintenance deux listes d'amorçage distinctes (une liste utilisateur et une liste par défaut), pour l'amorçage avec le sélecteur sur maintenance.

3. Lorsqu'une unité d'amorçage valide est trouvée, c'est le premier article d'amorçage ou le premier PSN (numéro de secteur de programme) qui est vérifié ; si cet article est valide, il est lu en mémoire et ajouté au bloc de contrôle de l'IPL en mémoire. Les principales données de l'article d'amorçage comprennent la position de démarrage de l'image d'amorçage sur l'unité d'amorçage, la taille de cette image et les instructions relatives à l'emplacement mémoire sur laquelle charger l'image d'amorçage.
4. L'image d'amorçage est lue sur l'unité d'amorçage, en mode séquentiel, et chargée en mémoire, à partir de l'emplacement indiqué dans l'article d'amorçage. L'image d'amorçage sur disque est constituée du noyau, d'un système de fichiers RAM et de données sur l'unité de base (personnalisée).
5. Le contrôle passe ensuite au noyau, qui commence l'initialisation du système.
6. Le processus 1 lance **init**, qui exécute la phase 1 du script **rc.boot**.

A l'issue de cette phase, la configuration de l'unité de base est lancée.

Phase de configuration de l’unité de base

La figure ci-après illustre la phase de configuration de l’unité de base.



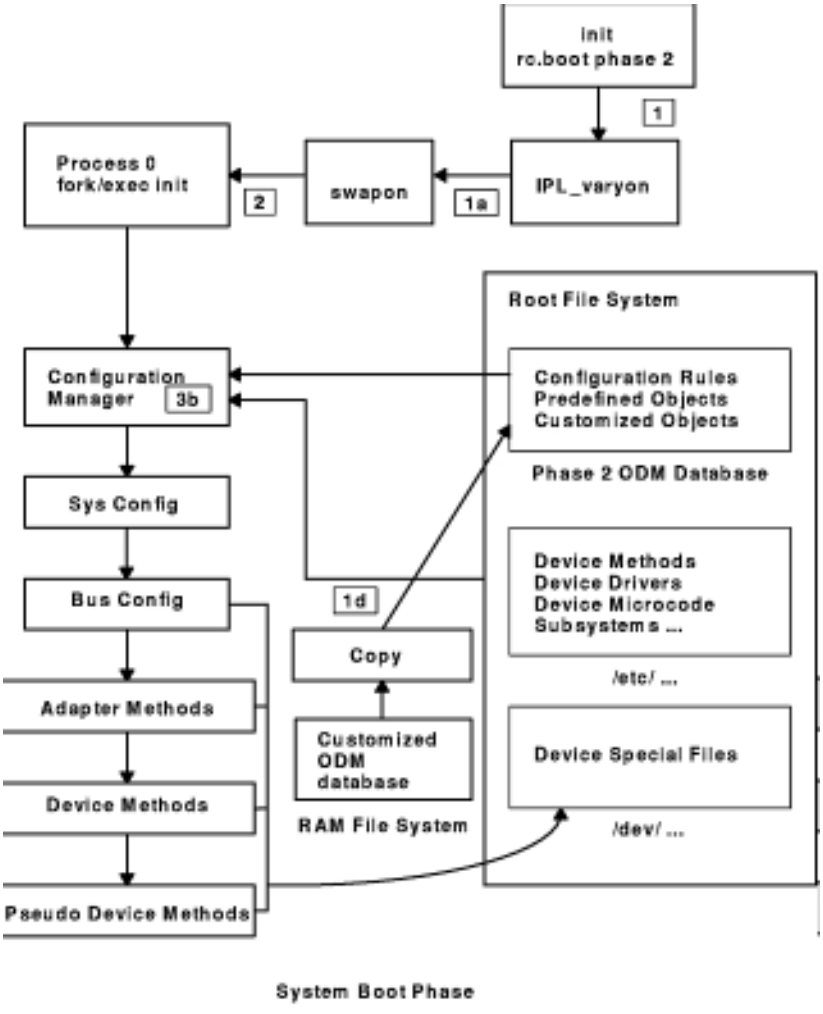
Base Device Configuration Phase

Le processus **init** lance le script **rc.boot**. La phase 1 de ce script se charge de la configuration de l’unité de base. Elle comprend les différentes étapes suivantes :

1. Le script d’amorçage appelle le programme **restbase** pour créer la base de données ODM personnalisée dans le système de fichiers RAM, à partir des données compressées personnalisées.
2. La script lance le gestionnaire de configuration, qui recherche les règles de configuration – phase 1 pour configurer les unités de base.
3. Le gestionnaire de configuration lance les méthodes de configuration du système (**sys**), du bus (**bus**), du disque (**disk**), de SCSI, de LVM et de RVG.
4. Les méthodes de configuration chargent les gestionnaires d’unités, créent les fichiers spéciaux et mettent à jour les données personnalisées dans la base de données ODM.

Phase d’amorçage du système

La figure ci-après illustre la phase d’amorçage du système.



System Boot Phase

1. Le processus **init** lance la phase 2 du script **rc.boot** comprenant les étapes suivantes :
   a. Appel du programme **ipl_varyon**, qui met RVG en ligne.
   b. Montage des systèmes de fichiers disque sur le système de fichiers RAM.
   c. Exécution de **swapon** pour lancer la pagination.
   d. Copie des données personnalisées de la base ODM du système de fichiers RAM dans la base ODM du système de fichiers disque.
   e. Démontage des montages temporaires des systèmes de fichiers disque, puis montages permanents de root, **/usr** et **/var**.
   f. Sortie du script **rc.boot**.
2. Le processus d’amorçage passe ensuite du système de fichiers RAM au système de fichiers disque.

3. Le processus init exécute les processus définis dans les articles du fichier **/etc/inittab**. Selon l'une des instructions de ce fichier, la phase 3 du script **rc.boot** est exécutée ; elle comprend les étapes suivantes :

   a. Montage du système de fichiers disque **/tmp**

   b. Lancement du gestionnaire de configuration – phase 2 pour configurer les unités restantes

   c. Exécution de la commande **savebase** pour sauvegarder les données personnalisées sur le volume logique d'amorçage

   d. Sortie du script **rc.boot**.

En fin de processus, le système est monté et prêt à l'emploi.

Description du processus d’amorçage de maintenance

Ce type d’amorçage est requis par certaines opérations spécifiques, telles que l’installation d’un nouveau logiciel ou d’une mise à jour, les tests de diagnostic ou la maintenance. Dans ce cas, le système démarre à partir d’un support amorçable (CD–ROM, bande), d’un réseau ou d’une unité de disque avec le sélecteur de mode en position maintenance.

La séquence de ce type d’amorçage est semblable à celle d’un amorçage normal. Les événements peuvent être soulignés comme suit :

1. L’OCS (On–Chip Sequencer) vérifie la carte système mère.
2. Le contrôle passe au ROS, qui exécute le POST.
3. Le ROS vérifie la liste d’amorçage utilisateur. Vous pouvez adapter cette liste à vos besoins au moyen de la commande **bootlist**. Si la liste en NVRAM n’est pas valide, ou si le système ne trouve pas d’unité d’amorçage valide, c’est la liste d’amorçage par défaut qui est vérifiée. Dans les deux cas, la première unité d’amorçage valide rencontrée est utilisée pour le démarrage du système.

   Remarque : Le système se charge de la maintenance d’une liste d’amorçage par défaut, située en ROS, et d’une liste d’amorçage utilisateur, stockée en NVRAM, pour l’amorçage normal. Font aussi l’objet de maintenance deux listes d’amorçage distinctes (une liste utilisateur et une liste par défaut), pour l’amorçage avec le sélecteur sur maintenance.
4. Lorsqu’une unité d’amorçage valide est trouvée, c’est le premier article d’amorçage ou le premir PSN (numéro de secteur de programme) qui est vérifié ; si cet article est valide, il est lu en mémoire et ajouté au bloc de contrôle de l’IPS en mémoire. Les principales données de l’article d’amorçage comprennent la position de démarrage de l’image d’amorçage sur l’unité d’amorçage, la taille de cette image et le décalage par rapport au point d’entrée, pour lancer l’exécution quand l’image est en mémoire.
5. L’image d’amorçage est lue sur l’unité d’amorçage, en mode séquentiel, et chargée en mémoire, à partir de l’emplacement indiqué dans l’article d’amorçage.
6. Le contrôle passe ensuite au noyau, qui lance l’exécution des programmes dans le système de fichiers RAM.
7. La base de données ODM identifie les unités présentes, puis la commande **cfgmgr** configure, de façon dynamique, toutes les unités trouvées, y compris les disques destinés à contenir le système de fichiers racine.
8. Si le système est amorcé à partir d’un CD–ROM, d’une bande ou du réseau, le groupe de volumes rootvg (RVG), qui n’existera pas nécessairement, n’est pas mis en ligne (comme dans le cas de l’installation du système d’exploitation sur un nouveau système). La configuration du réseau peut alors commencer. Avec ce type d’amorçage, la pagination n’a pas lieu.

A l’issue de ce processus, le système est prêt pour l’installation, la maintenance ou les diagnostics.

Note: Si le système est amorcé à partir du disque, le RVG est mis en ligne, les systèmes de fichiers disque racine et disque utilisateur sont montés dans le système de fichiers RAM. Un menu s’affiche, permettant d’opter pour un mode de diagnostic ou pour le mode mono-utilisateur. Ce dernier permet de poursuivre le processus d’amorçage et sa sélection est possible avec un niveau “S” d’exécution d’init. Le système est alors prêt pour la maintenance, les mises à jour de logiciel ou l’exécution de la commande **bosboot**.

Description du système de fichiers RAM

Le système de fichiers RAM, qui fait partie de l'image d'amorçage, réside en mémoire et contient tous les programmes assurant la poursuite du processus d'amorçage. Les fichiers de ce système déterminent le type d'amorçage.

Un système de fichiers RAM d'amorçage de maintenance ne contient pas toujours les routines du volume logique, du fait que le groupe de volumes rootvg n'est pas nécessairement mis en ligne. Toutefois, lors d'un amorçage à partir d'un disque, la mise ne ligne du groupe de volumes rootvg et l'activation de la pagination sont souhaitables aussi rapidement que possible. Malgré les différences entre ces deux scénarios d'amorçage, la structure du système de fichiers RAM ne varie pas de manière sensible.

La commande **init** du système de fichiers RAM utilisée lors de l'amorçage est en fait le programme **ssh** (simple shell). Ce programme contrôle le processus d'amorçage en appelant le script **rc.boot** dont la première étape consiste à déterminer à partir de quelle unité la machine a été amorcée. L'unité d'amorçage détermine les unités à configurer dans le système de fichiers RAM. Si la machine est amorcée à partir du réseau, les unités du réseau doivent être configurées pour permettre le montage à distance des systèmes de fichiers client. En cas d'amorçage à partir d'un bande ou d'un CD–ROM, la console affiche les menus d'installation de BOS. Quand **rc.boot** a identifié l'unité d'amorçage, les routines de configuration correspondantes sont appelées dans le système de fichiers RAM. **ssh** appelle **rc.boot** à deux reprises, c'est-à-dire pour les deux phases de configuration de l'amorçage. Un troisième appel de **rc.boot** se produit au moment de l'appel de la commande **init** pendant un amorçage disque ou réseau. Une strophe de **rc.boot** figurant dans le fichier **inittab** se charge de la configuration finale de la machine.

En raison de la variété des types d'unités à configurer, il existe un système de fichiers RAM distinct par unité d'amorçage. Un fichier prototype est associé à chaque type d'unité d'amorçage. Le fichier prototype est un modèle de fichiers qui constituent le système de fichiers RAM. La commande **mkfs** est utilisée par la commande **bosboot** pour créer le système de fichiers RAM au moyen de différents fichiers de prototype. Pour plus de détails, reportez-vous à la commande **bosboot**.

Description du processus de fermeture

Vous avez la possibilité de fermer le système sous contrôle dans les situations suivantes :

- après l’installation d’un nouveau logiciel ou la modification de la configuration logicielle,
- lors d’un incident matériel,
- en cas d’interruption anormale persistante,
- en cas de performances décroissantes,
- si un système de fichiers est vraissemblablement endommagé.

Chapitre 3. Protection du système

Ce chapitre traite des aspects évolués de la sécurité du système. Cette notion est développée dans le manuel *AIX 4.3 Guide de l'utilisateur : système d'exploitation et unités*.

Les sujets abordés sont les suivants :

Gestion de la sécurité

Pour assurer la protection des ressources de données, une gestion appropriée du système est primordiale. La sécurité AIXest fondée sur le contrôle d'accès (mise en oeuvre et actualisation). Un administrateur système en est responsable ; il est chargé de la configuration des différents éléments suivants :

Contrôle d'accès aux ressources protégées

Protège le caractère privé, l'intégrité et la disponibilité des données.

Identification et authentification, page 3-2

Définit les modalités d'identification des utilisateurs et du contrôle de l'authenticité de l'identité.

Base TCB (Trusted Computing Base), page 3-12

Renforce les mesures de sécurité.

Audit, page 3-17

Consigne et analyse les événements.

Différents aspects

La sécurité vise principalement la détection et la prévention en matière de violation de l'accès aux données. Elle englobe les accès fondamentaux développés ci-après.

Gestion utilisateur

La gestion utilisateur consiste à créer des utilisateurs et des groupes, et à définir leurs attributs, notamment l'attribut majeur d'authentification. L'utilisateur représente l'agent principal du système. Les attributs utilisateur permettent le contrôle des droits d'accès, de l'environnement, de l'authentification et du mode d'accès aux comptes utilisateur (notamment en ce qui concerne l'emplacement, les tranches horaires).

Un groupe représente plusieurs utilisateurs partageant des droits d'accès aux ressources protégées. Il possède un ID et est constitué de membres et d'administrateurs. Généralement, le créateur du groupe est le premier administrateur.

Le système d'exploitation prend en charge les attributs utilisateur standard ci–après, généralement enregistrés dans les fichiers **/etc/passwd** et **/etc/group**.

Authentication Information (authentification)	Mot de passe.
Credentials (identité)	ID utilisateur, groupe principal et ID groupe supplémentaire.
Environment (environnement)	Environnement shell ou personnel.

Au besoin, le système d'exploitation peut exercer un contrôle plus intense, avec des attributs étendus, et interdir l'accès public aux données relatives à la sécurité.

Certains utilisateurs et groupes peuvent être administrateurs. Ils sont créés et modifiés exclusivement au niveau utilisateur racine.

Contrôle du compte utilisateur

Un ensemble d'attributs est associé à chaque utilisateur. Ces attributs sont définis à partir de valeurs par défaut lors de la création de l'utilisateur avec la commande **mkuser**. La commande **chuser** permet de les modifier. Voici quelques exemples d'attributs utilisateur :

ttys	limite certains comptes à des zones protégées (physiquement).
expires	dédié à la gestion des comptes étudiant et visiteur, et permet de désactiver temporairement des comptes.
logintimes	limite la connexion utilisateur à certaines tranches horaires, par exemple, aux heures de bureau.

L'ensemble complet des attributs utilisateur est défini dans les fichiers **/usr/lib/security/mkuser.default**, **/etc/security/user**, **/etc/security/limits** et **/etc/security/lastlog**. Certains attributs contrôlent le mode de connexion de l'utilisateur et peuvent, selon des conditions spécifiées, verrouiller le compte utilisateur (pour empêcher d'autres connexions).

Une fois son compte verrouillé, l'utilisateur ne peut plus s'y connecter tant que l'administrateur n'a pas redéfini l'attribut **unsuccessful_login_count** dans le fichier**/etc/security/lastlog** à une valeur inférieure au nombre de tentatives de connexion, avec la commande **chsec**.

```
chsec –f /etc/security/lastlog –s username –a
     unsuccessful_login_count=0
```

Cette commande permet de modifier les valeurs par défaut, en éditant la strophe par défaut dans le fichier approprié, tel que **/etc/security/user**, **/usr/lib/security/mkuser.default** ou **/etc/security/limits**. Nombre de valeurs par défaut définissent un comportement standard.

Identification et authentification

L'identification et l'authentification constituent l'identité de l'utilisateur. Celui–ci, pour se connecter au système, doit décliner le nom utilisateur d'un compte et, le cas échéant, le mot de passe associé (les comptes d'un système sécurité sont soit assortis au mot de passe, soit invalidés). Si le mot de passe est correct, l'utilisateur est connecté au compte correspondant et dispose des droits d'accès et des privilèges de ce compte. Les mots de passe utilisateur sont actualisés dans les fichiers **/etc/passwd** et **/etc/security/passwd**.

D'autres méthodes d'authentification intégrées au système sont disponibles par le biais du paramètre **SYSTEM** figurant dans **/etc/security/user**. Par exemple, l'environnement DCE (Distributed Computing Environment) requiert également l'authentification du mot de passe, mais ne le valide pas selon le modèle de chiffrement utilisé dans **etc/passwd** et **/etc/security/passwd**. Pour l'authentification dans l'environnement DCE, la strophe correspondante est définie dans **/etc/security/user** à `SYSTEM=DCE`.

Les autres valeurs de **SYSTEM** sont **compat**, **files** et **NONE**. Le jeton **compat** sert à effectuer la résolution du nom (puis son authentification) auprès de la base de données locale, et, si elle n'aboutit pas, auprès de la base NIS (Network Information Services) ; le jeton **files** spécifie l'utilisation exclusive des fichiers locaux pour l'authentification ; et le jeton **NONE** désactive la méthode d'authentification. Pour désactiver complètement l'authentification, le jetong **NONE** doit figurer dans les lignes **SYSTEM** et **authl** de la strophe utilisateur.

Il est possible de définir d'autres jetons compatibles avec l'attribut **SYSTEM** dans **/etc/security/login.cfg**.

> **Remarque :** L'authentification de l'utilisateur racine doit toujours être effectuée par le biais du fichier sécurité local. L'entrée de l'attribut **SYSTEM** pour cet utilisateur est définie à **SYSTEM = "compat"** dans **/etc/security/user**.

Pour plus de détails sur la protection des mots de passe, reporte-vous au manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Restrictions relatives au mot de passe

Seule “l’éducation” de l’utilisateur permet une gestion efficace des mots de passe. Toutefois, pour augmenter la sécurité, AIXa prévu des restrictions, configurées par l’administrateur, contraignant au choix du mot de passe et à son changement à fréquence régulière. Ces restrictions sont enregistrées dans **/etc/security/user** et s’appliquent dès qu’un nouveau mot de passe utilisateur esdt défini. Elles sont définies par utilisateur et non globalement. En les conservant dans la strophe par défaut du fichier **/etc/security/user**, les mêmes restrictions s’appliquent à l’ensemble des utilisateurs. Il est préférable que tous les mots de passe doent protégés de la même façon.

Le système d’exploitation permet en outre d’étendre les restrictions sur les mots de passe : avec l’attribut **pwdchecks** du fichier **/etc/security/user**, l’administrateur peut ajouter de nouvelles sous-routines (appelées méthodes) au code corerspondant aux restrictions. Par ce biais, une politique locale (sur site) de restrictions peut être mise en oeuvre, applicable au niveau du système d’exploitation. Voir “Restrictions étendues”, page 3-5 pour plus d’informations.

Les restrictions doivent être judicieuses. Par exemple, une taille limitée du mot de passe (permettant de le deviner facilement) ou la sélection obligatoire de mots de passe compliqués (obligeant l’utilisateur à les écrire) peuvent compromettre la sécurité. En définitive, la sécurité des mots de passe repose sur les utilisateurs. La meilleure politique consiste en des restrictions simples, assorties d’instructions adaptées et d’un audit occasionnel (pour vérifier que les mots de passe sont uniques).

Voici les restrictions applicables :

minage	durée de vie minimale du mot de passe – exprimée en nombre de semaines – avant modification.
maxage	durée de vie maximale du mot de passe – exprimée en nombre de semaines – avant modification.
maxexpired	durée maximale au–delà de **maxage**, avant modification exigée par le système – excepté au niveau racine.
minalpha	taille minimale du nouveau mot de passe – exprimée en nombre de caractères alphabétiques.
minother	taille minimale du nouveau mot de passe – exprimée en nombre de caractères non–alphabétiques. (caractères ASCII imprimables, non–alphabétiques et différents des points de code de langue).
minlen	taille minimale du nouveau mot de passe – exprimée en nombre de caractères.

Remarque : La taille minimale d’un mot de passe correspond à **minlen** ou à **minalpha** plus **minother** (la plus grande des deux). La taille maximale est de huit caractères. La somme de **minalpha** et **minother** ne peut excéder huit, sinon **minother** est réduit en conséquence (8 moins **minalpha**).

maxrepeats	nombre maximal d’occurrences du même caractère dans le nouveau mot de passe.
mindiff	nombre minimal de caractères obligatoirement différents entre l’ancien et le nouveau mot de passe.
histexpire	durée – exprimée en nombre de semaines – pendant laquelle l’utilisateur ne peut réutiliser un ancien mot de passe.
histsize	nombre d’anciens mots de passe à ne pas réutiliser.

Remarque : Si **histexpire** et **histsize** sont tous deux définis, le système retient le nombre de mots de passe remplissant les deux conditions, dans la limite de 50 par utilisateur. Les mots de passe nuls ne sont pas pris en compte.

dictionlist liste de fichiers dictionnaire vérifiés lors d'un changement de mot de passe. Ces fichiers contiennent les mots de passe non-attribuables.

pwdchecks liste des méthodes externes de restrictions utilisées lors d'un changement de mot de passe.

Valeurs des attributs du mot de passe

valeurs des restrictions	valeurs recommandées	valeurs par défaut	valeurs maximum
minage	0	0	52
maxage	8	0	52
maxexpired	4	–1	52
minalpha	4	0	8
minother	1	0	8
minlen	6	0	8
mindiff	3	0	8
maxrepeats	1	8	8
histexpire	26	0	260*
histsize	0	0	50
dictionlist	NA	NA	NA
pwdchecks	NA	NA	NA
*50 mots de passe maximum sont retenus. NA = non applicable			

Un mot de passe doit être difficile à devinier et facile à mémoriser ; tenez-en compte dans la définition des restrictions. L'utilisation de mots de passe difficiles à mémoriser oblige l'utilisateur à les écrire, ce qui compromet la sécurité du système.

Sur un système avec traitement de texte, l'administrateur peut utiliser le fichier **/usr/share/dict/words** à la place de **dictionlist** et, dans ce cas, affecter la valeur 0 à **minother** (les mots du dictionnaire qui appartiennent à la catégorie **minother** étant peu nombreux, **minother** avec une valeur égale à 1 ou plus rendrait inutile la majorité des mots du dictionnaire).

Restrictions étendues

Les règles du programme acceptant ou refusant un mot de passe (restrictions sur la composition du mot de passe) peuvent être étendues pour définir des restrictions quant au site. Pour ce faire, l'administrateur système ajoute des sous-routines (appelées méthodes) exécutées pendant le changement d'un mot de passe. L'attribut **pwdchecks** du fichier **/etc/security/user** spécifie les méthodes appelées.

Le manuel *AIX Technical Reference* décrit **pwdrestrict_method**, interface de sous-routine fournissant les règles applicables aux méthodes de restrictions sur la composition des mots de passe. Pour étendre correctement les restrictions sur la composition des mots de passe, l'administrateur système doit programmer cette interface lorsqu'il écrit une méthode de restriction. Ceci suppose certaines précautions, étant donné l'impact direct sur les commandes **login**, **passwd** et **su**, et sur d'autres programmes. L'emploi du mauvais code peut facilement compromettre la sécurité du système. Par conséquent, n'utilisez qu'un code absolument sûr.

Journal des ID utilisateur

Tous les événements d'audit enregistrés pour un utilisateur donné sont libellés avec l'ID spécifié et doivent être examinés quand vous générez des audits. Pour plus de détails, reportez-vous au manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Règles applicables à la sécurité du système

Les règles suivantes s'adressent aux administrateurs qui mettent en oeuvre et actualisent la sécurité du système.

Introduction

Avertissement : Chaque environnement d'exploitation peut avoir ses propres besoins en matière de sécurité. Ceux-ci n'étant pas forcément traités ici. Il incombe donc à l'administrateur de les mettre en oeuvre pour assurer la sécurité du système.

Les règles données ne prétendent pas s'appliquer à tous les environnements d'exploitation, ni suffire à assurer totalement la sécurité d'un système, un seul ensemble de règles de sécurité ne pouvant satisfaire tous les besoins.

La politique de sécurité gagne à être planifiée avant de commencer à exploiter le système, sa mise en oeuvre en cours d'exploitation n'étant pas une économie de temps !

Les règles de sécurité concernent les catégories suivantes :

Sécurité de base

Chaque système doit maintenir le niveau de sécurité conforme à la politique adoptée en matière de sécurité de base.

Comptes utilisateur

De nombreux attributs peuvent être définis puor chaque compte utilisateur, y compris les attributs de mot de passe et de connexion (pour consulter la liste des attributs configurables, reportez–vous au tableau "Gestion des utilisateurs et des groupes" dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*). Voici quelques recommandations :

- Un ID utilisateur distinct par utilisateur est préconisé : les outils de protection et de comptabilité ne sont exploitables qu'avec des ID utilisateur uniques.
- Employez des noms utilisateur significatifs : les noms réels sont les plus adaptés, d'autant que la plupart des applications de courrier électronique prennent l'ID utilisateur comme libellé pour le courrier entrant.
- Utilisez le Web-based System Manager ou l'interface SMIT pour ajouter, modifier ou supprimer des utilisateurs. Ces opérations peuvent être effectuées à partir de la ligne de commande, mais ces interfaces permettent d'éviter les erreurs mineures.

- N’anticipez pas pour attribuer un mot de passe de compte utilisateur tant que l’utilisateur concerné n’est pas prêt à se connecter au système ; avec le symbole * (astérisque) dans la zone mot de passe du fichier **/etc/passwd**, les informations du compte sont conservées, mais personne ne peut se connecter sur ce compte.
- Ne modifiez pas les ID utilisateur définis par le système : ils lui sont nécessaires pour une exploitation correcte ; ils sont répertoriés dans le fichier **/etc/passwd**.
- Ne passez aucun paramètre **admin** d’ID utilisateur à **true**. Seul l’utilisateur racine est autorisé à le faire dans le fichier **/etc/security/user**.

Groupes et propriété de fichier

A la création d’un fichier, le système d’exploitation affecte à l’ID utilisateur du nouveau fichier l’ID utilisateur effectif du processus qui l’a créé. L’ID groupe du fichier est soit l’ID groupe effectif du processus, soit l’ID groupe du répertoire contenant le fichier, sur la base du bit SGID (Set Group ID) de ce répertoire.

Le propriétaire du fichier peut être changé par le biais de la commande **chown**.

La commande **id** affiche l’UID (ID utilisateur), le GID (ID groupe) et les noms de tous les groupes dont vous êtes membre.

Dans les listings de fichier (tels que ceux générés par la commande **li** ou **ls**), les trois groupes d’utilisateurs sont toujours représentés dans l’ordre suivant : utilisateur, groupe, autre. Pour trouver votre nom de groupe, utilisez la commande **groups** qui affiche tous les groupes d’un ID utilisateur.

Pour plus de détails sur les modes d’accès aux fichiers et aux répertoiers, reportez-vous à “Propriété des fichiers et groupes utilisateur” dans *AIX 4.3 System User’s Guide: Operating System and Devices*.

Groupes

Un groupe représente plusieurs utilisateurs partageant des droits d’accès aux ressources protégées. Planifiez-les avant de les créer : une fois créés, il est moins facile d’en modifier l’organisation. Il existe trois types de groupes : utilisateur, administrateur système et les groupes définis par le système.

Groupes utilisateur

Il est conseillé de créer le moins de groupes utilisateur possible.

Le groupe réunit généralement les utilisateurs partageant les mêmes fichiers : tels que les collaborateurs d’un même service ou d’un même projet.

Par exemple, le personnel d’une petite société d’ingénierie, constitué d’employés de bureau, d’administrateurs système et d’ingénieurs peut, dans un premier temps, être réparti en deux groupes utilisateurs : un groupe BUREAU et un groupe INGENIEUR ; par la suite, si un sous–groupe d’ingénieurs démarre un projet particulier, un nouveau groupe peut être créé, par exemple le groupe PROJET, les ID des ingénieurs concernés étant ajoutés à ce groupe. Ainsi, certains utilisateurs font partie de plusieurs groupes, mais ils ne peuvent être membres que d’un seul groupe principal. Vous pouvez changer les utilisateurs de groupe principal avec la commande **newgrp**.

Dans des systèmes simples, il est en outre conseillé de ne pas définir **admin** lors de la création de groupes. Un groupe avec **admin=true** dans le fichier **/etc/security/group** peut être géré uniquement par l’utilisateur racine.

Groupes administrateur système

Les administrateurs système sont, de préférence, membres du groupe SYSTEM. Cette appartenance leur permet d’effectuer certaines tâches de maintenance sans avoir besoin des droits d’accès racine.

Groupes définis par le système

Plusieurs groupes sont définis par le système : Le groupe STAFF est réservé, par défaut, aux utilisateurs non administratifs créés dans le système. Pour changer de groupe par défaut, utilisez la commande **chsec** pour éditer le fichier **/usr/lib/security/mkuser.default**.

Le groupe SECURITY est doté de privilèges limités pour la gestion de la sécurité. Ses membres ont accès aux programmes et aux fichiers du répertoire **/etc/security**. Les membres du groupe SECURITY peuvent modifier la plupart des attributs des utilisateurs et des groupes non administratifs, tels que le shelle de connexion utilisateur ou l'appartenance à un groupe non administratif.

La plupart des systèmes n'ont pas besoin du groupe SECURITY ; ce groupe est adapté aux systèmes multi-utilisateur comptant un grand nombre d'utilisateurs. Les administrateurs peuvent effectuer les mêmes tâches que les membres du groupe SECURITY avec la commande **su** pour obtenir les privilèges racine.

Les autres groupes définis par le système permettent de contrôler certains sous–systèmes. Pour considérer l'appartenance de certains utilisateurs à ces groupes, reportez-vous aux informations sur les sous-systèmes. Le fichier **/etc/group** contient les groupes et les utilisateurs définis par le système.

Systèmes de fichiers

A chaque objet système de fichiers (y compris fichiers, répertoires, fichiers spéciaux, fichiers de liens et tubes) est associé un mécanisme de sécurité. La liste de contrôle d'accès ou ACL (Access Control List) est le plus courant. Le contrôle de la sécurité peut en outre être assuré par les mécanismes suivants :

ACL de base	Autorisations accordées au propriétaire, au groupe et aux autres utilisateurs, contrôlées par la commande **chmod**. Pour plus de détails, reportez-vous à "Mode d'accès aux fichiers et aux répertoires" dans le manuel *AIX 4.3 System User's Guide: Operating System and Devices*.
ACL étendu	Contrôle d'accès plus pointu que celui offert par l'ACL de base. Pour plus de détails, reportez-vous à "Listes de contrôle d'accès (ACL)" dans *AIX 4.3 System User's Guide: Operating System and Devices*.
Etat des ACL étendus	L'ACL étendu doit être activé pour un objet système de fichiers ; sinon, ce type d'ACL n'est pas pris en compte.
ID propriétaire	ID du propriétaire de l'objet système de fichiers ; cet ID utilisateur est l'unique bénéficiaire des autorisations du propriétaire de l'objet.
ID groupe	ID du groupe associé à l'objet. Seuls les membres de ce groupe sont les uniques bénéficiaires des autorisations du groupe associé à l'objet.
Bit de rappel	Lorsque défini pour un répertoire, seul le propriétaire du répertoire ou du fichier peut supprimer ou renommer un fichier du répertoire (même si d'autres utilisateurs ont l'autorisation d'écriture dans le répertoire). Il peut être assorti de l'indicateur **t** avec la commande **chmod**.
Bit TCB	Lorsque défini pour un objet système de fichiers, identifie cet objet comme faisant partie de la base TCB (Trusted Computing Base).
umask	Le paramètre d'environnement **umask** indique les autorisations attribuées par défaut à tout fichier ou répertoire créé.
Etat du système de fichiers	Le montage d'un système de fichiers autorisé en lecture-écriture ou seulement en lecture est possible.

Pour les objets système de fichiers, respectez les règles suivantes :

- Evitez d'utiliser les ACL étendus. Les ACL de base sont généralement suffisants pour gérer la plupart des systèmes. N'utilisez les ACL étendus que si vous avez besoin du contrôle supplémentaire ; dans ce cas, soyez méthodique. L'actualisation des entrées d'un grand nombre d'ACL étendus prend du temps. Ne les utilisez pas si votre réseau est hétérogène : ils sont seulement reconnus par les systèmes AIX.
- Le bit de rappel ne s'emploie que sur les répertoires autorisés en écriture à tous les utilisateurs.
- Définissez 740 autorisations pour protéger les fichiers utilisateur **.profile**.
- N'autorisez pas l'accès en écriture sur les répertoires système aux utilisateurs.
- Ne modifiez pas les autorisations d'accès à des fichiers ou à des répertoires installés en tant que partie du système. Ce type de modification a des répercussions sur l'intégrité du système.

Accès racine

Avertissement : Le compte racine ne doit pas être partagé ; il doit être protégé par un mot de passe. Seul l'administrateur système doit connaître ce mot de passe. Les tâches de gestion système requérant des privilèges racine doivent être effectués par l'administrateur, celui-ci opérant uniquement en tant qu'utilisateur racine, puis repassant sur son compte utilisateur habituel. Prendre l'habitude de travailler au niveau racine peut endommager le système : en effet, à ce niveau, nombre de protections sont annulées dans le système.

L'administrateur doit posséder le mot de passe racine pour obtenir l'autorisation racine avec la commande **su**. Le mot de passe du compte racine doit être attribué immédiatement après l'installation du système.

L'authentification du compte racine doit toujours être effectuée par le biais des fichiers de sécurité locaux.

Variable d'environnement PATH

La variable d'environnement **PATH** constitue un contrôle de sécurité important. Elle indique les répertoires où rechercher les commandes. A l'échelle du système, la valeur de **PATH** par défaut est spécifiée dans le fichier **/etc/profile** ; normalement chaque utilisateur possède une valeur**PATH** dans le fichier utilisateur **$HOME/.profile**. Dans le fichier **.profile**, la valeur de **PATH** se substitue à celle de **PATH** à l'échelon du système ou lui ajoute des répertoires.

Des modifications illicites de **PATH** peuvent permettre à un utilisateur d'espionner d'autres utilisateurs (y compris des utilisateurs racine). Des programmes espions (également appelés Cheval de Troie) remplacent les commandes système pour s'emparer des informations qui leu sont destinées, telles que les mots de passe utilisateur.

Supposons, par exemple, qu'un utilisateur modifie la valeur **PATH** pour que le système recherche le répertoire **/tmp** dès qu'une commande est lancée. Il met ensuite dans ce répertoire un programme nommé **su** qui retrouve le mot de passe racine comme le fait la commande **su**. Le programme **/tmp/su** adresse alors le mot de passe à cet utilisateur et appelle la vraie commande **su** avant de sortir. Avec un tel scénario, n'importe quel utilisateur racine s'étant servi de la commande **su** aurait pu fournir le mot de passe racine à son insu. Il existe nombre d'autres scénarios frauduleux permettant d'obtenir des informations confidentielles en modifiant les valeurs **PATH**.

Pour prévenir tout problème avec la variable **PATH**, voici quelques conseils simples à l'intention des administrateurs et des utilisateurs :

- Dans le doute, indiquez des chemins d'accès complets : ainsi, la variable **PATH** n'est pas prise en compte.
- Ne placez jamais le répertoire courant (indiqué par un **.** point) dans la valeur **PATH** de l'utilisateur racine. Ne l'indiquez jamais dans **/etc/profile**.
- L'utilisateur racine se sert de la valeur **PATH** du fichier **/etc/profile**. N'indiquez que des répertoires autorisés en écriture au niveau racine. En outre, il est recommandé de ne pas créer de fichiers **.profile** dans le répertoire racine (**/**). Créez-les uniquement dans les répertoires utilisateur **$HOME**.
- Les modifications apportées à des fichiers utilisateur **.profile** doivent être approuvées par l'administrateur système. A défaut, elles pourraient autoriser, à l'insu de leur auteur, des accès interdits. Les autorisations d'un fichier utilisateur **.profile** doivent être définies à 740.
- Les administrateurs système ne doivent pas lancer la commande **su** pour obtenir les privilèges racine à partir d'une session utilisateur, la valeur de **PATH** utilisateur indiquée dans le fichier **.profile** étant effective. La définition des fichiers **.profile** utilisateur est indifférente. Les administrateurs système doivent se connecetr en tant qu'utilisateur racine sur la machine utilisateur ou employer la commande :

  ```
  su – root
  ```

  Ceci garantit l'utilisation de l'environnement racine pendant la session. Si l'administrateur système opère au niveau racine dans une autre session utilisateur, il doit indiquer le chemin d'accès complet pendant cette session.
- Empêchez les modifications de la variable d'environnement **IFS** (Input Fiels Separator) dans les fichiers **/etc/profile**. Soyez attentifs aux éventuelles modifications de la variable **IFS** dans le fichier **.profile** par un quelconque utilisateur. Cette variable peut servir à changer la valeur **PATH**.

Sécurité étendue

Ces règles confèrent au système un plus grand niveau de protection, mais requièrent une maintenance plus conséquente. C'est pourquoi nombre d'administrateurs ne les appliquent que très peu, voire pas du tout.

Comptabilité

La comptabilité système n'est pas directement liée à la sécurité. Toutefois, les informations qu'elle permet de collecter contribuent largement à détecter les problèmes relatifs à la sécurité. Il est conseillé d'activer une comptabilité de base sur le système, comme expliqué dans "Mise en œuvre d'un système de comptabilité" au sein du manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*, n'incluant pas nécessairement la comptabilité disque et impression. Ces fonctions génèrent quantité de données et ne sont pas essentielles en matière de sécurité.

Audit

L'audit des petits systèmes n'est généralement pas indispensable. Pour les grands systèmes multi-utilisateurs, il fournit nombre d'informations utiles sur l'activité du système. Pour plus de détails, reportez-vous à "Audit-Généralités", page 3-17.

Base TCB

La base TCB (Trusted Computing Base) permet un contrôle plus pointu des programmes protégés. Elle améliore en outre la sécurité du système.

La plupart des systèmes ne font appel qu'à deux composants de cette base : la commande **tcbck** et le fichier de configuration par défaut **/etc/security/sysck.cfg**. La commande **tcbck** utilise les données du fichier **/etc/security/sysck.cfg** pour comparer l'état de la sécurité des éléments-clés avec la base de données du fichier **sysck.cfg**.Il incombe aux administrateurs de protéger le fichier **syscj.cfg** et d'exécuter régulièrement la commande **tcbck**.

Le contrôle des plus grands systèmes par la base TCB est beaucoup plus intensif. TCB fournit un ensemble protégé de composants système. La gestion administrative est toutefois plus lourde. Pour plus de détails, reportez-vous à "Base TCB-Généralités", page 3-12.

Rôles administratifs

AIX version 4.2.1 (et ultérieures) permet à l'administrateur système d'affecter à un utilisateur un ensemble de rôles administratifs avec différents niveaux de droits. Pour plus de détails, reportez-vous à "Rôles administratifs", page 4-1.

Sécurité des réseaux et communications

La sécurité des réseaux et des communications est décrite dans le manuel *AIX 4.3 System Management Guide: Communications and Networks*.

- Sécurité de TCP/IP
- Sécurité des commandes TCP/IP
- Description de la sécurité BNU
- Base NTCB (Network Trusted Computing Base)
- Sécurité de NIS
- Configuration du système sécurité NFS

Base TCB - Généralités

La base TCB (Trusted Computing Base) est un composant du système renforçant les règles de sécurité. Elle comprend tout le matériel du système. Dans un premier temps, il est intéressant pour l'administrateur d'en connaître les composants logiciels.

Nombre de fonction de TCB s'activent en option à l'installation. Sélectionner **oui** sur l'option **Inst. base info. sécurisée** du menu Installation et paramètres active le chemin d'accès sécurisé, le shelle sécurisé et la vérification de l'intégrité du système (commande **tcbck**. L'option **non** désactive ces fonctions. Celles-ci ne sont activables qu'au moment de l'installation.

Le logiciel de TCB se compose :

- du noyau (système d'exploitation),
- des fichiers de configuration contrôlant les activités du système,
- de tout programme assorti de privilèges ou de droits d'accès pour modifier le noyau ou les fichiers de configuration.

La plupart des fichiers système ne sont accessibles qu'à l'utilisateur racine ; toutefois, les membres d'un groupe administratif peuvent aussi avoir accès à certains de ces fichiers. Seul l'utilisateur racine est autorisé à modifier le noyau du système d'exploitation. TCB contientles programmes sécurisés suivants :

- tous les programmes **setiud** racine,
- tous les programmes **setgid** de groupes administratifs,
- tout programme exploité exclusivement par l'utilisateur racine ou un membre du groupe système,
- tout programme obligatoirement exécuté par l'administrateur sur le chemin d'accès sécurisé de communication (par exemple, la commande **ls**).

Dans le système d'exploitation, l'administrateur système peut placer des marques sur des fichiers sécurisés pour repérer facilement qu'ils sont intégrés à la base TCB (commande **chtcb**).

L'administrateur système doit veiller à n'ajouter que des logiciels sécurisés dans la base TCB. Il doit notamment s'assurer :

- qu'il a fait l'objet de tests complets ;
- que son code programme a été vérifié ;
- qu'il provient d'une source sécurisée où il a été testé et où le code programme a été vérifié.

L'administrateur système doit évaluer à quel point sécuriser un programme, en prenant en compte la valeur des ressources de données du système et en décidant du degré de sécurité nécessaire pour un programme à installer avec des privilèges.

Programme de vérification tcbck

Un élément important du programme **tcbck** est l'attribut **program** qui figure dans le fichier **/etc/security/sysck.cfg**. Cet attribut édite un programme associé pouvant vérifier d'autres états. La vérification est plus pointue et plus souple qu'avec les autres attributs.

Utilisez ces programmes pour vérifier l'intégrité et la cohérence du contenu d'un fichier et sa relation avec les autres fichiers. Ces programmes ne se limitent pas à un fichier particulier.

Par exemple, supposez un programme, **/etc/profile**, qui vérifie que les fichiers utilisateur **.profile** sont protégés en écriture, excepté pour leur utilisateur. Ce programme doit avoir les caractéristiques suivantes :

- propriété au niveau **racine**,
- membre d'un goupe **système**,
- mode 0750,
- marque d'appartenance à la base TCB.

Ce programme peut être ajouté au vérificateur de sécurité du système en entrant :

```
tcbck –a /etc/profile
”program=/etc/profile” class=profiles \ owner group mode
```

Cette commande génère dans la base de données l'entrée suivante :

```
/etc/profile:

class = profiles

owner = root

group = system

mode = TCB,rwxr–x––

program = ”/etc/profile”
```

La commande **tcbck** suivante vérifie l'installation du programme **/etc/profile** et exécute le programme :

```
tcbck –t profiles
```

Les programmes de vérification **tcbck** doivent remplir les conditions suivantes :

- accepter les indicateurs **–n**, **–y**, **–p** et **–t** et les gérer de la même façon que la commande **sysck** ;
- renvoyer 0 pour signifier l'absence d'erreur et écrire tous les messages d'erreur en erreur standard ;
- à noter que ces programmes sont exécutés avec un ID utilisateur effectif à 0 et qu'ils possèdent, par conséquent, tous les privilèges. Ils doivent être écrits et inspectés comme des programmes setuid racine.

Programme de vérification TCB

Le système d'exploitation fournit les programmes de vérification TCB suivants :

pwdck	contrôle la cohérence interne et mutuelle des fichiers **/etc/passwd** et **/etc/security/passwd**.
grpck	contrôle la cohérence interne et mutuelle des fichiers **/etc/group** et **/etc/security/group**.
usrck	vérifie dans le fichiers de la base de données utilisateur l'exactitude des définitions de tout ou partie des utilisateurs.

Installatino et mise à jour du système sécurisé

L'installation et la mise à jour d'un programme consistent à importer des fichiers dans le système, généralement à créer de nouveaux répertoires pour le programme et parfois à reconfigurer le système. Pour la sécurité, le programme peut avoir à ajouter des comptes utilisateur, à définir de nouveaux événements d'ausit et à attribuer des privilèges à un des fichiers programme.

Le programme d'installation le plus simple consiste à installer une nouvelle arborescence de sous-répertoire (à partir de **/usr/lpp**) et à ajouter au besoin de nouveaux liens symboliques dans le répertoire **/usr/bin**. Toutefois, deux questions se posent :

- La configuration système doit généralement être modifiée ; en outre, certaines commandes du programme requièrent la définition de privilèges administratifs dont le niveau est à déterminer.
- Pour éviter toute interférence entre procédures d'installation, chaque programme doit être installé sous un domaine d'accès distinct.

Sécuriser installations et mises à jour fait appel à deux stratégies. La première consiste à décrire et à limiter les privilèges et les droits d'accès pendant l'installation et la mise à jour, réduisant les risques d'endommagement que présentent des modules d'installation non fiables. Avec la seconde stratégie, l'ensemble du process peut être sou mis à audit, en procédant par analyse du suivi d'audit du système une fois l'installation ou la mise à jour du programme terminée, à moins que l'audit ne soit interactif. La commande **watch** permet d'effectuer un audit interactif. Elle permet d'exécuter un programme spécifié et affiche, le cas échéant, les enregistrements d'audit générés pendant l'exécution de ce programme.

Cette approche offre une grande souplesse à l'installation, tout en fournissant un niveau de sécurité élevé. Elle est efficace, même si la sécurité est plutôt un travail de détection que de prévention. En outre, le process étant interactif, l'installation d'un programme "malsain" peut être rapidement interrompue.

Règles relatives aux modes et à la propriété des fichiers

Commandes utilisateur standard

Exécutables par tout utilisateur sans SUID (ID utilisateur) ni SGID (ID groupe), ces commandes ne requièrent pas de propriétaire ni de groupe. Pour être exécutées sur le chemin d'accès sécurisé (par exemple, avec les commandes **vi**, **grep** et **cat**), elles doivent être assorties d'un bit TCB. Voici un exemple portant sur la propriété et les modes :

```
owner:  bin     r–x
group:  bin     r–x
others:         r–x
```

Commandes utilisateur administratif

Exécutables uniquement au niveau racine, ce type de commandes est réservé aux membres d'un groupe administratif et aux membres spécifiés dans les rubriques de l'ACL étendu. Elles exécutent généralement des opérations privilégiées pour lesquelles le SUID peut être requis. Voici un exemple portant sur la propriété et les modes :
`owner: root` `r–x` (SUID parfois requis)
`group: system r–x` (SGID parfois requis)
`others:`(rubriques ACL étendu parfois requises)

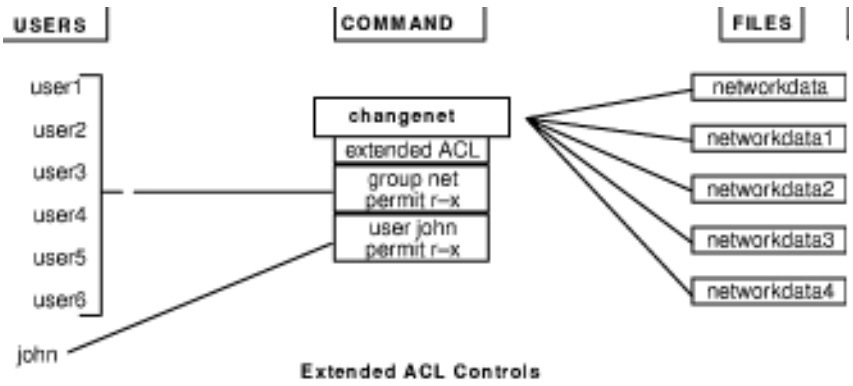
A titre d'exemple de scénario type de commande utilisateur administratif, prenez des fichiers réseau contenant des informations importantes sur la configuration du réseau :

```
owner:  root            rw–
group:  netgroup        rw–
others:                 –––
```

Modifiez–en les données au moyen de la commande **changenet**. `owner:`
`root r–x`
` group: netgroup ––s`
` others:`(rubriques ACL étendu)
`permit r–x g:net`
` permit r–x u:john`

Dans cet exemple, le groupe **netgroup** est un groupe administratif privilégié dépourvu de membres. Les fichiers réseau ne sont autorisés en lecture-écriture qu'aux process exécutés avec ce groupe (ou l'utilisateur racine). La commande **changenet**, qui est le SGID **netgroup**, permet de modifier les fichiers réseau. Elle n'est exécutable que par les utilisateurs racine, les membres du groupe `net` et l'utilisateur nommé `john`.

La figure ci-après illustre commande l'ACL étendu est associé à la commande et non aux fichiers de données.



La commande **changenet** représente la passerelle vers les fichiers de configuration du réseau. L'ACL étendu qui lui est associé est le gardien n'autorisant l'accès qu'à certains utilisateurs.

Cet exemple suppose un nombre de fichiers de données supérieur au nombre de programmes opérant sur ces fichiers. Dans le cas contraire, il serait plus pertinent d'associer l'ACL aux fichiers de données.

Fichiers de configuration

L'exemple suivant illustre le groupe **admin** avec le privilège administratif et aucun membre. Le nom exact de ce groupe dépend du type de fichier de configuration s'appliquant au groupe (par exemple, audit, authentification et courrier).

```
owner:  root    rw–
group:  admin   rw–
others:         r––
mode:           TCB
```

La plupart des fichiers de configuration sont généralement accessibles en lecture à l'ensemble des utilisateurs, excepté les données d'audit et d'authentification.

Fichiers spéciaux d'unités

Les unités ne doivent pas être accessibles en lecture-écriture aux utilisateurs normaux. La seule exception concerne les terminaux (écriture autorisée pour l'envoi de messages entre utilisateurs) et les unités de disquette (accessibles en lecture-écriture pour les transferts de fichiers).

Chemin d’accès sécurisé des communications

Ce chemin d’accès du système d’exploitation sécurise les communications entre les utilisateurs et la base TCB. Pour le démarrer, utilisez la clé SAK (Secure Attention Key). Elle donne l’accès des seuls process sécurisés au terminal utilisateur. L’utilisateur fait appel à ce type d’accès pour entrer des données confidentielles (par exemple, un mot de passe). L’administrateur y fait aussi appel pour mettre en oeuvre un environnement d’administration protégé.

Remarque : Si l’option **Inst. base info. sécurisée** n’a pas été sélectionnée au cours de l’installation initiale, le chemin d’accès sécurisé de communication est désactivé. Pour l’activer correctement, le système doit être réinstallé.

Ce chemin d’accès est conçu sur la base :

- d’un interpréteur de commande sécurisé (commande **tsh**) exécutant uniquement les commandes signalées par une marque comme étant membres de la base TCB ;
- d’un accès au terminal limité aux programmes sécurisés ;
- d’une séquence de clés appelée SAK, permettant à l’utilisateur de demander un accès sécurisé aux communications.

Après activation de SAK, la commande **init** lance soit la commande **getty**, soit la commande **shell** qui :

- passe le terminal à un autre propriétaire et à un autre mode pour n’autoriser que les process exécutés par un utilisateur donné à ouvrir le terminal ;
- lance une sous-routine **frevoke** pour invalider tous les appels **open** antérieurs (demandes d’ouverture du terminal).

Cette opération garantit l’accès au terminal uniquement par la commande **getty** ou **shell**.

Interpréteur de commande sécurisé

La commande **getty** n’exécute la commande **shell** en réponse à l’activation de la clé SAK que si l’utilisateur est déjà connecté au terminal. Elle implante les modes du terminal et exécute le shelle sécurisé (commande **tsh**).

La commande **tsh** fournit un sous-ensemble de fonctions shell standard (Korn). Le shell sécurisé n’exécute que les programmes sécurisés (par exemple, les programmes assortis du bis TCB). La commande **shell** intégrée permet à l’utilisateur d’exécuter le shell de connexion utilisateur au lieu de faire appel à l’accès sécurisé des communications.

Audit - généralités

Le sous-système d'audit donne à l'administrateur système le moyen d'enregistrer les informations relatives à la sécurité, puis de les analyser pour détecter les violations (réelles et potentielles) des règles de sécurité. Il présente trois fonctions : Il présente trois fonctions, configurables par l'administrateur. Chacune de ces fonctions peut être configurée par l'administrateur système.

Détection des événements

Cette fonction est distribuée dans la base TCB, tant dans le noyau (code superviseur) que dans les programmes sécurisés (code utilisateur). Toute occurrence système relative à la sécurité représente un événement "auditable". Toute modification de l'état de la sécurité, toute violation (réelle ou sous forme de tentative) du contrôle d'accès et/ou des règles de sécurité constitue une occurrence système relative à la sécurité. Les modules de programmes ou de noyau détectant les événements auditables les signalent à l'enregistreur d'audit du système, intégré au noyau et accessible par une sousroutine (pour les audits de programmes sécurisés) ou par un appel de procédure à l'intérieur du noyau (pour les audits superviseur). Doivent être enregistrés le nom de l'événement, une indication mentionnant s'il a abouti, ou non et toute information spécifique susceptible d'intéresser l'audit de sécurité.

Configurer cette fonction consiste à activer/désactiver la détection d'événement, à un niveau global (à l'échelon du système) ou local (process). Pour contrôler la détection d'événement au niveau global, utilisez la commande **audit**. Le contrôle de la détection d'événement au niveau local se fait au moyen d'audit d'utilisateurs sélectionnés pour des groupes d'événements d'audit (classes d'audit).

Collecte d'informations

Cette fonction comprend la consignation des événements sélectionnés "auditables", exécutée par l'enregistreur d'audit intégré au noyau. Ce dernier fournit un SVC (sous-routine) et une interface d'appel de procédure (intégrée au noyau) qui enregistre les événements auditables.

L'enregistreur d'audit se charge entièrement de la création des enregistrements d'audit, formés d'un en-tête de données communes à tous les événements (nom de l'événement, utilisateur responsable, date et heure de l'événement, état renvoyé), et d'un suivi d'audit contenant les informations spécifiques de l'événement. Il intègre successivement chaque enregistrement au suivi d'audit du noyau, dans le ou les modes suivants :

mode BIN	Le suivi est inscrit dans d'autres fichiers, pour la sécurité et pour le stockage sur le long terme.
mode STREAM	Le suivi est inscrit dans un tampon circulaire lu de façon synchronisée par une pseudo-unité d'audit. Dans ce mode, la réponse est immédiate.

Vous pouvez configurer la collecte des données du côté frontal (enregistrement des événements), et dorsal (traitement du suivi du noyau). L'enregistrement des événements peut être défini par utilisateur ; ceux-ci sont enregistrés dans le noyau lorsqu'ils se produisent. Côté dorsal, les modes peuvent être configurés individuellement : l'administrateur peut ainsi choisir le traitement dorsal le plus adapté à chaque environnement. En outre, il peut définir le mode d'audit BIN pour la fermeture du système en cas d'indident.

Traitement des données

Pour le traitement du suivi d'audit du noyau, le système d'exploitation propose plusieurs options. En mode BIN, il est possible de compresser, de filtrer et/ou de formatter les données, dans la mesure où la combinaison de ces options avant l'archivage est, le cas échéant, compatible avec le système. La compression est effectuée en codage Huffman. Le filtrage est effectué en langage SQL (Standard Query Language) – comme la sélection d'enregistrement d'audit (avec la commande **auditselect**) – et permet la rétention et l'affichage sélectifs du suivi d'audit. Pour analyser le suivi d'audit, l'imprimer et générer des rapports périodiques sur la sécurité, vous pouvez en formater le contenu. Le mode STREAM permet de gérer la surveillance des menaces en temps réel. Les options de ce mode sont contrôlées par des programmes distincts qui peuvent être appelés comme des process démon pour filtrer les suivis en mode BIN ou STREAM, sachant que certains programmes de filtrage sont plus adaptés à un mode qu'à l'autre.

Sélection d'événement d'audit

L'ensemble des événements système auditables définit les occurrences à auditer et la granularité de l'audit fourni. Comme expliqué plus haut, les événements concernés sont obligatoirement relatifs à la sécurité du système. Dans la définition des événements auditables, le niveau détail doit être soigneusement évalué : en effet, un niveau trop faible peut générer une collecte trop importante d'informations, et un niveau surestimé, empêcher de comprendre la logique des informations sélectionnées. La définition des événements bénéficie de la similitude de certains événements détectés. Un événement détecté est un événement, de diverses origines, susceptible d'être soumis à audit. Le principe repose sur la similitude des propriétés relatives à la sécurité : si elles sont semblables, les événements détectés sont classifiés comme des événements identiques, auditables. En voici la classification :

Evénements relatifs aux règles de sécurité

- Sujets
  - création de process
  - suppression de process
  - définition des attributs de sécurité : ID groupe et utilisateur
  - groupe de process, terminal de contrôle
- Objets
  - création
  - suppression
  - ouverture (y compris de process en tant qu'objets)
  - fermeture (y compris de process en tant qu'objets)
  - définition des attributs de sécurité : propriétaire, groupe, ACL
- Import/Export
  - import ou export d'un objet
- Comptabilité
  - ajout d'un utilisateur, modification d'attributs utilisateur dans la base des mots de passe
    Base de données
  - ajout d'un groupe, modification d'attributs dans la base des groupes
    Base de données
  - connexion utilisateur
  - déconnexion utilisateur
  - modification de l'authentification utilisateur
  - configuration terminal : accès sécurisé
  - configuration authentification
  - administration de l'audit : sélection des événements et des suivis d'audit, activation
    activation/désactivation, définition des classes d'audit utilisateur
- Gestion système
  - exploitation des privilèges
  - configuration système de fichiers
  - définition et configuration d'unité
  - définition des paramètres de configuration système
  - IPL et fermeture système (traitement standard)
  - configuration RAS
  - configuration d'autres systèmes
- Violation (potentielle) de la sécurité
  - refus d'autorisation d'accès
  - incidents relatifs aux privilèges
  - erreurs système détectées par les diagnostics
  - (tentative de) modification de la base TCB.

Configuration

Le sous-système d'audit est assorti d'une variable d'état globale indiquant si le sous-système est activé ou non. En outre, une variable d'état locale est affectée à chaque process définissant l'enregistrement ou non par le sous-système des informations relatives aux process. Ces variables (globale et locales) déterminent si les événements sont détectés par les modules et programmes de la base TCB. En désactivant l'audit TCB d'un process spécifique, celuici peut gérer son propre audit sans passer outre les règles de comptabilité du système. Le programme sécurisé autorisé à effectuer son propre audit bénéficie d'une collecte d'informations plus efficace.

Collecte d'informations

Elle concerne les modes de sélection d'événements et de suivi d'audit du noyau. Les composants TCB qui détectent les événements "auditables" font appel à ces interfaces. En outre, cette routine fournit des interfaces de configuration au sous-système d'audit pour contrôler la routine de l'enregistreur.

Journal d'audit

Les événements "auditables" sont enregistrés par une ou deux interfaces, l'état utilisateur et l'état superviseur. La partie état utilisateur de la base TCB fait appel à la sous-routine **auditlog** ou **auditwrite**, tandis que la partie état superviseur se sert d'un ensemble d'appels de procédure noyau.

L'enregistreur d'audit affecte un préfixe spécifique de l'événement à l'en-tête de chaque enregistrement. L'en-tête identifie l'utilisateur et le process pour lesquels l'événement fait l'objet d'un audit, ainsi que l'heure de cet événement. Le code qui détecte l'événement en fournit le type, le code retour ou l'état, et, en option, des informations complémentaires sur l'événement (queue de l'événement). Ces informations sont les noms d'objet (par exemple, fichiers dont l'accès a été refusé ou terminal tty utilisé par des tentatives échouées de connexion), des paramètres de sous-routine et d'autres informations modifiées.

La définition des événements est symbolique et non numérique. Cela réduit le risque de collision de noms, sans qu'un schéma d'enregistrement d'événement soit nécessaire. En outre, les sous-routines étant susceptibles d'être soumises à audit, la définition extensible du noyau, dépourvue de numéro SVC fixe, ne facilite pas la numérotation des enregistrements d'événements :

Format des enregistrements d'audit

Les enregistrements d'audit sont constitués d'un en-tête commun et des suivis d'audit spécifiques de l'événement. Les structures d'en-têtes sont définies dans le fichier **/usr/include/sys/audit.h**. Dans les suivis d'audit, le format des informations dépend de la base d'événements associée; il figure dans le fichier **/etc/security/audit/events**.

Pour en garantir l'exactitude, la collecte des informations de l'en-tête d'audit est généralement effectuée par la routine de connexion, tandis que les informations des suivis d'audit sont fournies par le code qui détecte l'événement. L'enregistreur d'audit ne connaît pas la structure ou la sémantique des suivis d'audit. Par exemple, lorsque la commande **login** détecte une connexion échouée, elle enregistre l'événement, y compris le terminal sur lequel il s'est produit et consigne l'enregistrement dans la queue d'audit au moyen de la sous-routine **auditlog**. L'enregistreur d'audit (composant du noyau) enregistre les données spécifiques du sujet (ID utilisateur, ID process, heure) dans un en-tête et l'ajoute aux autres informations. L'appelant ne fournit que le nom de l'événement et les zones résultantes dans l'en-tête.

Configuration de l’enregistreur

L’enregistreur d’audit est chargé de fabriquer de toutes pièces l’enregistrement d’audit. Vous devez sélectionner les événements d’audit à consigner.

Sélection d’événement d’audit

Elles sont de deux types : par process et par objet.

Audit par process

Pour une sélection efficace d’événements de process, l’administrateur système a la possibilité de définir des classes d’audit. Une classe d’audit est un sous-ensemble d’événements d’audit de la base. Les classes d’audit permettent un regroupement logique des événements d’audit de la base.

L’administrateur définit un ensemble de classes d’audit par utilisateur du système, qui détermine les événements relatifs à l’utilisateur concerné susceptibles d’être enregistrés. Tout process exécuté par l’utilisateur est étiqueté avec ses classes d’audit.

Audit par objet

Le système d’exploitation permet l’audit des accès aux objets par leur nom, c’estàdire l’audit d’objets spécifiques (généralement des fichiers). En outre, il est possible de préciser le mode d’audit : La plupart des objets ne présentant pas d’intérêt du point de vue de la sécurité, ce type d’audit permet de ne sélectionner que les accès aux objets pertinents. de cette façon, uniquement les accès dans le mode spécifié (lecture/écriture/exécution) et les résultats (succès/échec) sont enregistrés.

Modes de suivi d’audit du noyau

Le mode BIN ou STREAM peut être défini pour la connexion noyau, indiquant où inscrire le suivi d’audit du noyau. Avec le mode BIN, un ou plusieurs descripteurs de fichier auxquels ajouter les enregistrements doivent être associés à l’enregistreur d’audit du noyau (avant le démarrage de l’audit).

Le mode BIN consiste à inscrire les enregistrements d’audit dans des fichiers alternés. Au démarrage de l’audit, deux descripteurs de fichier sont associés au noyau, ainsi qu’une taille maximale de casier. Ce mode suspend le process d’appel et démarre l’inscription des enregistrements d’audit dans le premier descripteur de fichier. Une fois le premier casier saturé, si le second descripteur est valide, passe au deuxième casier et réactive le process appelant. L’inscription se poursuit dans le second casier jusqu’à l’appel suivant avec un autre descripteur valide. A ce stade, si le second casier est saturé, l’opération repasse au premier casier et le retour du process appelant est immédiat. Sinon, le process appelant est suspendu, et le noyau poursuit l’inscription des enregistrements dans le second casier jusqu’à saturation. Le traitement continue de cette façon jusqu’à désactivation de l’audit.

STREAM est un mode beaucoup plus simple. Le noyau inscrit les enregistrements dans un tampon circulaire. Ensuite, il repasse au début dès que la fin du tampon est atteinte. Les informations sont lues par les process via une pseudo-unité appelée **/dev/audit**. Quand un process ouvre cette unité, un nouveau canal est créé. En option, il est possible de spécifier comme une liste de classes d’audit les process à lire sur le canal.

L’objectif principal de ce mode est de permettre une lecture opportune du suivi d’audit, souhaitable pour gérer les menaces en temps réel. Ce mode permet aussi de créer un suivi immédiat, destiné à un support papier, pour prévenir toute falsification du suivi d’audit, possible dès lors que ce dernier est enregistré sur un support inscriptible.

Chapitre 4. Rôles administratifs

Rôles administratifs

AIX version 4.3 permet d'attribuer une partie des droits utilisateur racine à des utilisateurs non racine. Les tâches utilisateur racine sont affectées d'autorisations distinctes, regroupées en rôles. Ce sont ces rôles qui sont affectés à divers utilisateurs.

Ce chapitre aborde les points suivants :

Rôles - Généralités

Un rôle est composé d'autorisations qui permettent à un utilisateur d'exécuter des fonctions normalement réservées à l'utilisateur racine.

Liste des rôles valides :

Ajout et retrait d'utilisateurs	Donne à un utilisateur les droits racine pour un rôle : ajout et retrait d'utilisateurs, modification des informations sur un utilisateur, modification des classes d'audit, gestion des groupes et modification des mots de passe. Tout utilisateur habilité à exécuter des tâches d'administration d'utilisateurs doit être membre du groupe **security**.
Modification des mots de passe utilisateur	Permet de modifier des mots de passe.
Gestion des rôles	Permet à un utilisateur de créer, modifier, supprimer et afficher les rôles. L'utilisateur doit appartenir au groupe **security**.
Sauvegarde et restauration	Permet à un utilisateur de sauvegarder et de restaurer des systèmes de fichiers et des répertoires. Ce rôle requiert des autorisations pour activer la sauvegarde et la restauration d'un système.
Sauvegarde seulement	Permet à un utilisateur de sauvegarder seulement des systèmes de fichiers et des répertoires. L'utilisateur doit détenir les droits requis pour activer la sauvegarde d'un système.
Exécution des diagnostics	Permet à un utilisateur, à un ingénieur commercial ou à un employé du service d'assistance technique d'exécuter des tâches de diagnostic. L'utilisateur doit avoir **system** comme groupe principal ainsi qu'un groupe comportant la commande **shutdown**. **Remarque :** les utilisateurs exerçant ce rôle peuvent modifier la configuration du système, mettre à jour le microcode, etc. Ils doivent être conscients des responsabilités incombant à ce rôle.
Arrêt du système	Permet à un utilisateur de fermer, réamorcer et arrêter le système.

Autorisations

Les autorisations sont des attributs conférant des droits à un utilisateur. Ces autorisations lui permettent d'exécuter certaines tâches. Par exemple, un utilisateur détenant l'autorisation UserAdmin peut créer un utilisateur administratif via la commande **mkuser**. Un utilisateur non détenteur de ce droit ne peut créer d'utilisateur administratif.

Il existe deux types d'autorisations :

Autorisation de base	Permet à un utilisateur d'exécuter une commande spécifique. Par exemple, l'autorisation RoleAdmin est une autorisation de base permettant à un administrateur d'exécuter la commande **chrole**. A défaut de cette autorisation, la commande s'achève sans modifier les définitions du rôle.
Modificateur d'autorisation	Etend les droits d'un utilisateur. Par exemple, UserAdmin est un modificateur d'autorisation qui étend les droits d'un administrateur membre du groupe **security**. Sans cette autorisation, la commande **mkuser** ne crée que des utilisateurs non administrateurs. Avec cette autorisation, la commande **mkuser** crée également des utilisateurs non administrateurs.

Les autorisations sont les suivantes :

Backup	Effectue une sauvegarde du système.	
	La commande suivante fait appel à l'autorisation Backup :	
	Backup	Sauvegarde fichiers et systèmes de fichiers. L'administrateur doit détenir l'autorisation Backup.
Diagnostics	Permet à un utilisateur d'exécuter des tâches de diagnostic. Cette autorisation est également requise pour exécuter ces tâches directement à partir de la ligne de commande.	
	La commande suivante fait appel à l'autorisation Diagnostics :	
	diag	Exécute le programme de diagnostics sur des ressources sélectionnées. Si l'administrateur des utilisateurs n'a pas l'autorisation Diagnostics, le programme s'interrompt.
GroupAdmin	Exécute les fonctions de l'utilisateur racine sur les données d'un groupe.	
	Les commandes suivantes font appel à l'autorisation GroupAdmin :	
	chgroup	Modifie les informations de groupe. A défaut de l'autorisation GroupAdmin, l'utilisateur ne peut modifier que les informations relatives à un groupe non administratif.
	chgrpmem	Administre tous les groupes. A défaut de l'autorisation GroupAdmin, l'administrateur de groupe ne peut modifier que l'appartenance au groupe qu'il administre ou un utilisateur du groupe de sécurité pour administrer un groupe non administratif quelconque.

	chsec	Modifie les données d'un groupe administratif dans les fichiers **/etc/group** et **/etc/security/group**. L'utilisateur peut également modifier les valeurs de la strophe **default:** valeurs des strophes. A défaut de l'autorisation GroupAdmin, l'utilisateur ne peut modifier que les données d'un groupe non administratif dans les fichiers **/etc/group** et **/etc/security/group**.
	mkgroup	Crée un groupe. A défaut de l'autorisation GroupAdmin, l'utilisateur ne peut créer que des groupes non administratifs.
	rmgroup	Supprime un groupe. A défaut de l'autorisation GroupAdmin, l'utilisateur ne peut créer que des groupes non administratifs.
ListAuditClasses	Affiche la liste de classes d'audit valides. L'administrateur d'utilisateur qui se sert de cette autorisation n'a pas besoin d'être utilisateur **root** ou membre du groupe **audit**. Entrez le raccourci **smit mkuser** ou **smit chuser** pour afficher la liste des classes d'audit disponibles pour créer ou modifier un utilisateur. Indiquez la liste des classes d'audit dans le champ AUDIT classes.	
PasswdAdmin	Exécute les fonctions de l'utilisateur racine sur les données d'un mot de passe. Les commandes suivantes font appel à l'autorisation PasswdAdmin :	
	chsec	Modifie les attributs **lastupdate** et **flags** de tous les utilisateurs. A défaut de l'autorisation PasswdAdmin, la commande **chsec** permet à l'administrateur de modifier les attributs **lastupdate** et **flags** des seuls utilisateurs non administratifs.
	lssec	Affiche les attributs **lastupdate** et **flags** de tous les utilisateurs. A défaut de l'autorisation PasswdAdmin, la commande **lssec** permet à l'administrateur d'afficher les attributs **lastupdate** et **flags** des seuls utilisateurs non administratifs.
	pwdadm	Modifie les mots de passe de tous les utilisateurs. L'administrateur doit appartenir au groupe **security**.
PasswdManage	Exécute des fonctions d'administration des mots de passe sur les utilisateurs non administratifs. Les commandes suivantes font appel à l'autorisation PasswdManage :	
	pwdadm	Modifie le mot de passe d'un utilisateur non administratif. L'administrateur doit être membre du groupe **security** ou détenir l'autorisation PasswdManage.
UserAdmin	Exécute les fonctions de l'utilisateur racine sur les données d'un groupe. Seuls les utilisateurs détenteurs de l'autorisation UserAdmin peuvent modifier les informations de rôle d'un utilisateur. Vous ne pouvez accéder ou modifier les informations d'audit avec cette autorisation. Les commandes suivantes font appel à l'autorisation UserAdmin :	

	chfn	Modifie le champ gecos (general information) d'un utilisateur quelconque. Si l'utilisateur ne détient pas l'autorisation UserAdmin, mais qu'il appartient au groupe **security**, il peut modifier le champ gecos de n'importe quel utilisateur non administratif. Sinon, il ne peut modifier que son propre champ gecos.
	chsec	Modifie les données d'un utilisateur administratif dans les fichiers **/etc/passwd**, **/etc/security/environ**, **/etc/security/lastlog**, **/etc/security/limits** et **/etc/security/user**, attributs de rôle compris. L'utilisateur peut également modifier les valeurs de la strophe **default:** et le fichier **/usr/lib/security/mkuser.default**, à l'exclusion des attributs auditclasses.
	chuser	Modifie les informations (excepté l'attribut auditclasses) d'un utilisateur quelconque. Si l'utilisateur ne détient pas l'autorisation UserAdmin, il peut modifier les informations utilisateur des seuls utilisateurs non administratifs, exception faite des attributs auditclasses et de rôle.
	mkuser	Crée un utilisateur, excepté son attribut auditclasses. Si l'utilisateur ne détient pas l'autorisation UserAdmin, il ne peut créer que des utilisateurs non administratifs, exception faite des attributs auditclasses et de rôle.
	rmuser	Supprime un utilisateur. Si l'utilisateur ne détient pas l'autorisation UserAdmin, il ne peut créer que des utilisateurs non administratifs.
UserAudit	Permet à l'utilisateur de modifier les informations d'audit utilisateur.	
	Les commandes suivantes font appel à l'autorisation UserAudit :	
	chsec	Modifie l'attribut auditclasses du fichier **mkuser.default** des utilisateurs non administrateurs. Si l'utilisateur détient l'autorisation UserAdmin, il peut également modifier l'attribut auditclasses du fichier **mkuser.default** des utilisateurs administrateurs et non administrateurs.
	chuser	Modifie l'attribut auditclasses d'un utilisateur non administrateur. Si l'administrateur détient l'autorisation UserAdmin, il peut également modifier l'attribut auditclasses de tous les utilisateurs.
	lsuser	Affiche l'attribut auditclasses d'un utilisateur non administrateur s'il est utilisateur **root** ou membre du groupe **security**. S'il détient l'autorisation UserAdmin, il peut également afficher l'attribut auditclasses de tous les utilisateurs.

	mkuser	Crée un utilisateur et autorise l'administrateur à affecter l'attribut auditclasses d'un utilisateur non administrateur. Si l'utilisateur détient l'autorisation UserAdmin, il peut également modifier l'attribut auditclasses de tous les utilisateurs.
RoleAdmin	Exécute les fonctions de l'utilisateur racine sur les données d'un mot de passe. Les commandes suivantes font appel à l'autorisation RoleAdmin :	
	chrole	Affiche un rôle. Si l'administrateur ne détient pas l'autorisation RoleAdmin, la commande est arrêtée.
	lsrole	Affiche un rôle.
	mkrole	Affiche un rôle. Si l'administrateur ne détient pas l'autorisation RoleAdmin, la commande est arrêtée.
	rmrole	Suppression d'un rôle Si l'administrateur ne détient pas l'autorisation RoleAdmin, la commande est arrêtée.
Restore	Effectue une restauration du système.	
	La commande suivante fait appel à l'autorisation Restore :	
	Restore	Restaure les fichiers sauvegardés. L'administrateur doit détenir l'autorisation Restore.

Pour consulter la liste des correspondances entre commandes et autorisations, reportez–vous à la section "Liste commandes/autorisations" dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Chapitre 5. Administration des utilisateurs et des groupes

Administration des utilisateurs et des groupes

Ce chapitre, consacré à l'administration des utilisateurs et des groupes, comporte également des informations sur les quota disque.

Système de quota disque - généralités

Le système de quota disque permet à l'administrateur de contrôler le nombre de fichiers et de blocs de données affectables aux utilisateurs et aux groupes. Les sections suivantes décrivent ce système, sa mise en œuvre et son exploitation :

Concept

Le système de quota disque, fondé sur le concept du système Berkeley Disk Quota System, représente un moyen efficace de contrôler l'exploitation de l'espace disque. Vous pouvez le définir pour les utilisateurs et pour les groupes. Il est actualisé pour tout système de fichiers journalisé.

Ce système définit des limites sur la base de trois paramètres modifiables avec la commande **edquota** :

- soft limit : limite inférieure (des utilisateurs ou des groupes),
- hard limit : limite supérieure (des utilisateurs ou des groupes),
- quota grace period : "délai de grâce".

Le paramètre *soft limit* définit le nombre minimal de blocs disque de 1 Ko ou de fichiers. Le paramètre *hard limit* définit le nombre maximal. Le paramètre quota grace period autorise l'utilisateur à dépasser la limite inférieure sur une courte durée (une semaine, par défaut). Passé ce délai, si l'utilisateur ne parvient pas à rétablir la limite inférieure, le système considère cette limite comme le maximum autorisé et n'affecte pas d'espace de stockage supplémentaire à l'utilisateur. Pour rétablir la condition initiale, l'utilisateur peut supprimer autant de fichiers que nécessaire.

Le suivi des quotas des utilisateurs et des groupes est enregistré dans les fichiers **quota.user** et **quota.group** résidant dans les répertoires racine de systèmes de fichiers où les quotas sont activés. Ces fichiers sont créés avec les commandes **quotacheck** et **edquota**, et lues avec les commandes de quota.

Reprise sur dépassement de quota

Pour réduire l'exploitation du système de fichiers après un dépassement de quota, vous avez le choix entre différentes méthodes :

- abandonner le processus qui a généré le dépassement, supprimer des fichiers pour rétablir le quota, puis relancer le programme qui était en cours.
- si vous travaillez avec un éditeur (par exemple, vi), vérifier l'espace avec la séquence d'échappement shell, supprimer des fichiers puis poursuivre le traitement en cours (avec ce procédé, vous ne perdez pas le fichier en cours) ; ou, si vous utilisez le shell C ou le shell Korn, suspendre l'éditeur en appuyant sur Ctrl-Z, exécuter les commandes de système de fichiers, puis reprendre le traitement, avec la commande **fg** (foreground).
- transférer temporairement le fichier en cours dans un système de fichiers non saturé, supprimer des fichiers, puis rapatrier le fichier.

Mise en œuvre

Avant de mettre en œuvre le système de quota disque, prenez les points suivants en compte :

- l'espace disque du système est limité.
- la sécurité du système de fichiers devra être accrue.
- les niveaux d'exploitation disque sont importants (comme dans nombre d'universités).

En principe, si votre environnement n'est pas concerné par ces points, définir des limites d'exploitation du disque en mettant le système de quota en œuvre n'est pas pertinent.

Le système de quota est plutôt adapté aux systèmes de fichiers contenant des fichiers et des répertoires personnels utilisateur. Ce système n'est applicable qu'aux systèmes de fichiers journalisés.

> **Remarque :** Il est recommandé de ne pas appliquer le système de quota au système de fichiers **/tmp**.

Chapitre 6. Volumes logiques

Ce chapitre décrit les concepts de gestion du stockage sur volumes logiques. Ce chapitre traite des sujets suivants :

Stockage sur volume logique - généralités

La gestion du stockage sur disque (fixe) fait appel à une hiérarchie de structures. Chaque unité de disque fixe, appelée *volume physique* (ou PV) porte un nom, tel que `/dev/hdisk0`. Chaque volume logique en cours d'utilisation fait partie d'un *groupe de volumes* (VG). Tous les volumes physiques d'un groupe de volumes sont divisés en *partitions physiques* (PP) de même taille (2 Mo par défaut dans les groupes de volumes comportant des volumes physiques de capacité inférieure à 300 Mo, 4 Mo dans les autres cas). Pour l'affectation de l'espace, chaque volume physique est divisé en cinq régions (outer_edge, inner_edge, outer_middle, inner_middle et center). Dans chaque région, le nombre de partitions physiques varie en fonction de la capacité totale de l'unité de disque. Si le groupe de volumes est créé avec l'option –B dans la commande mkvg, ces limites passent à 128 volumes physiques et 512 volumes logiques.

Dans chaque groupe de volumes, un ou plusieurs *volumes logiques* sont définis (LV). Les volumes logiques regroupent des données situées sur des volumes physiques. Pour l'utilisateur, les données sur volumes logiques semblent contiguës, alors qu'elles ne le sont pas nécessairement sur le volume physique. Ceci permet de modifier la taille ou l'emplacement des systèmes de fichiers, de l'espace de pagination et d'autres volumes logiques, de fractionner des volumes physiques multiples et de reproduire leur contenu pour un stockage de données plus souple et plus accessible.

Chaque volume logique est constitué d'une ou de plusieurs *partitions logiques* (LP). Une partition logique correspond à une partition physique minimum. Si l'écriture miroir est spécifiée pour le volume logique, des partitions physiques supplémentaires sont affectées pour stocker les copies supplémentaires de chaque partition logique. Les partitions logiques sont numérotées les unes à la suite des autres, ce qui ne signifie pas que les partitions physiques correspondantes sont consécutives ou contiguës.

Les volumes logiques ont des fonctions multiples, telles que la pagination, mais chaque volume logique dans lequel résident des données ou des programmes standard utilisateur ou système contient un système de fichiers journalisés (JFS) unique. Les JFS sont constitués d'un pool de blocs de la taille d'une page (4 ko). Pour l'écriture de données dans un fichier, un ou plusieurs blocs supplémentaires sont affectés au fichier. Ces blocs ne sont pas nécessairement contigus. Sous AIX 4.1, un système de fichiers donné peut être défini avec une taille de fragment inférieure à 4 ko (512 octets, 1 ko, 2 ko).

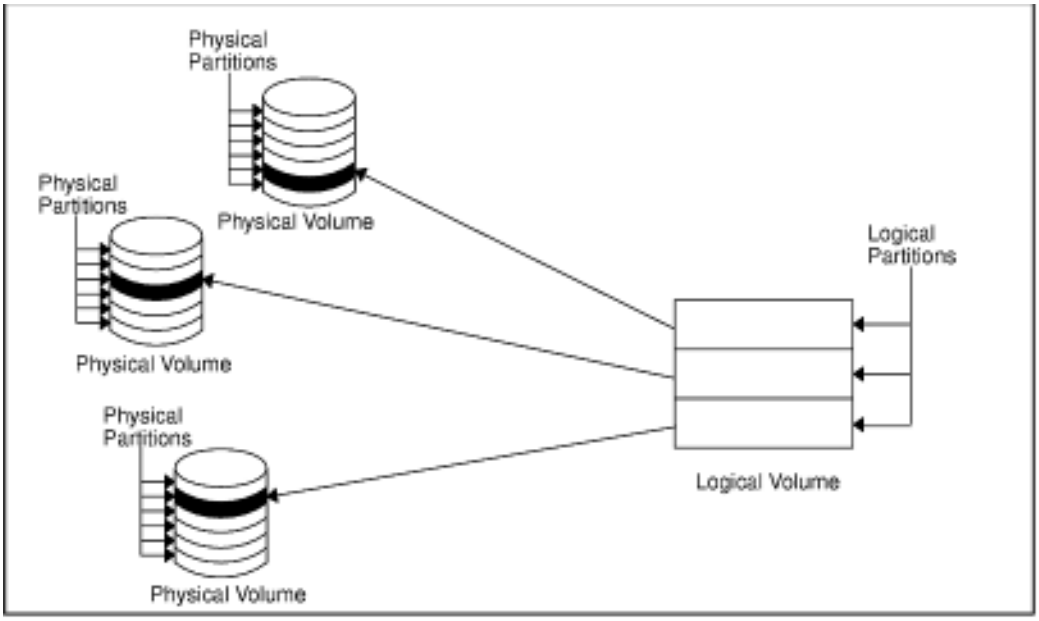
Après installation, le système possède un groupe de volumes (le groupe racine rootvg) formé d'un ensemble de base des volumes logiques nécessaires au démarrage du système et de tout autre volume spécifié dans le script d'installation. Tout volume physique supplémentaire connecté au système peut être ajouté à un groupe de volumes (avec la commande **extendvg**). Il peut être ajouté soit au groupe rootvg, soit à un autre groupe de volumes (défini avec la commande **mkvg**). Vous pouvez personnaliser les volumes logiques avec des commandes ou avec SMIT.

Dans les généralités, vous trouverez des informations sur :

- processus vary-on, page 6-7
- quorums, page 6-8
- mise en fonction forcée, page 6-8,
- groupes de volumes à l'état "nonquorum", page 6-9.

Concepts de stockage sur volume logique

Le stockage sur volume logique est fondé sur cinq concepts de base : volumes physiques, groupes de volumes, partitions physiques, volumes logiques et partitions logiques. Les relations entre ces différents concepts sont illustrées ci-après.



Volume Group

A volume group composed of three physical volumes with maximum range specified. The logical volume (which can span physical volumes) is composed of logical partitions allocated onto physical partitions.

Volumes physiques

Le disque doit être défini comme un volume physique; en outre, il doit être à l'état disponible avant son affectation à un groupe de volumes. Des données de configuration et d'identification sont écrites sur le volume physique. Ces données sont notamment l'identificateur unique (à l'échelle du système) de volume physique. Pour devenir un volume physique, le disque est divisé en *blocs physiques* de 512 octets. Le disque est défini en tant que volume physique via la commande **mkdev** ou **chdev**, ou avec l'interface SMIT (procédure d'ajout d'un volume physique).

Lors du premier démarrage du système après la connexion d'un nouveau disque, le système d'exploitation détecte le disque et recherche dans l'article d'amorçage s'il possède un identificateur unique de volume physique. Dans l'affirmative, le disque est désigné comme un volume physique portant un nom (généralement sous la forme **hdisk***x*, *x* représentant un nombre unique à l'échelle du système) associé à ce disque à titre permanent (jusqu'à l'annulation éventuelle de sa définition).

Groupes de volumes

A ce stade, le volume physique doit faire partie d'un seul groupe de volumes. Ce dernier peut regrouper de 1 à 32 volumes physiques de taille et de type divers. Il n'appartient qu'à un seul groupe de volumes par système; le système admet 255 groupes de volumes maximum.

Quand un volume physique est affecté à un groupe de volumes, les blocs physiques du support de stockage sont structurés en partitions physiques dont la taille est définie à la création du groupe de volumes. Les partitions physiques sont décrites ci-après.

Un groupe de volumes est automatiquement créé à l'installation du système (groupe racine appelé **rootvg).** rootvg contient un ensemble de base des volumes logiques nécessaires au démarrage du système et tout autre volume spécifié dans le script d'installation. En outre, il comprend un espace de pagination, un journal, des données d'amorçage, une mémoire de vidage, chacun de ces éléments figurant sur un volume logique distinct. Les attributs de rootvg sont différents des attributs (définis par les utilisateurs) des autres groupes de volumes. Par exemple, rootvg ne peut pas être importé ni exporté. Vous devez connaître les caractéristiques spécifiques de rootvg pour exécuter toute commande ou procédure sur ce groupe de volumes.

Pour créer un nouveau groupe de volumes, utilisez la commande **mkvg**. Pour y ajouter un volume physique, utilisez la commande **extendvg** et pour en supprimer un, la commande **reducevg**. Voici la liste des autres commandes disponibles : **chvg** (modifications), **lsvg** (liste des groupes de volumes), **exportvg** (suppression), **importvg** (installation), **reorgvg** (restructuration), **syncvg** (synchronisation), **varyonvg** (mise en fonction) et **varyoffvg** (mise hors fonction).

Sur les petits systèmes, un seul groupe de volumes peut suffire à contenir tous les volumes physiques connectés. Sauf si vous souhaitez, pour des raisons de sécurité, en créer plusieurs, pour répartir les diverses autorisations. En outre, les groupes de volumes séparés facilitent les opérations de maintenance pendant lesquels les groupes non concernés restent actifs. rootvg, étant toujours en ligne, ne doit contenir qu'un nombre minimal des volumes nécessaires et suffisants pour l'exploitation du système.

Pour transférer des données entre volumes physiques *du même groupe de volumes*, utilisez la commande **migratepv**. Elle permet de libérer un volume physique pour le supprimer de son groupe. Par exemple, vous pouvez transférer des données d'un volume à remplacer.

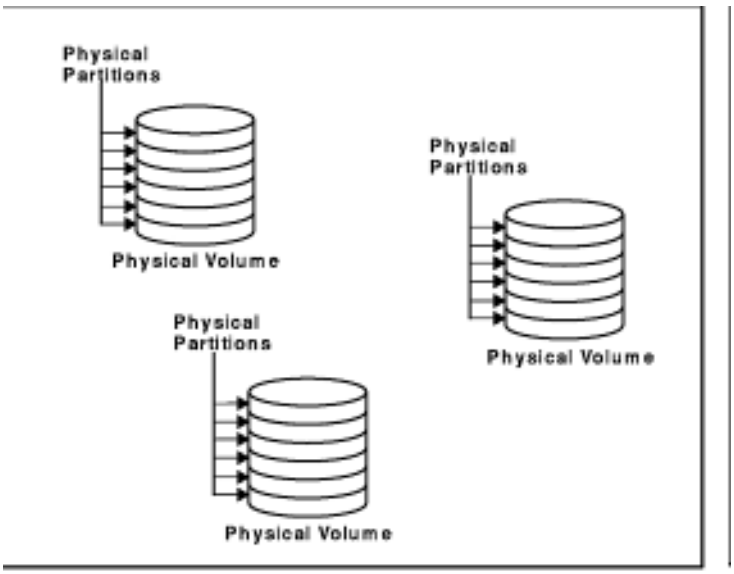
Un groupe de volumes créé avec des limites de volume physique et logique inférieures peut être converti vers un gros format pouvant supporter plus de volumes physiques (jusqu'à 128) et plus de volumes logiques (jusqu'à 512). Pour cela, il faut qu'il y ait suffisamment de partitions disponibles sur chaque volume physique du groupe de volumes pour l'extension de la zone VGDA (Volume group descriptor area). Le nombre de partitions disponibles nécessaires dépend de la taille de la zone VGDA actuelle et de la taille de la partition physique. La VGDA étant située sur le bord du disque et nécessite un espace continu, les partitions disponibles doivent se trouver sur le bord du disque. Si ces partitions sont affectées à l'utilisateur, elles seront envoyées sur d'autres partitions disponibles du disque. Le reste des partitions physiques sera renuméroté afin de prendre en compte la perte de partitions pour utilisation par VGDA. Cela modifiera les affectations des partitions logiques aux partitions physiques dans tous les volumes physiques de ce groupe de volumes. Si vous avez sauvegardé les affectations des volumes logiques pour une éventuelle opération de restauration, vous devez regénérer les affectations à la fin de l'opération de conversion. Ainsi, si la sauvegarde du groupe de volumes est faite avec l'option affectation et que vous voulez restaurer le système avec ces affectations, l'opération de restauration risque d'échouer puisque le numéro de partition n'existe plus (en raison de la réduction). Il est recommandé de faire la sauvegarde avant la conversion et juste après la conversion si vous utilisez l'option affectation. L'espace VGDA étant augmenté de manière sensible, chaque opération de mise à jour de la VGDA (création d'un volume logique, modification d'un volume logique, ajout d'un volume physique, etc.) peut prendre beaucoup de temps.

Remarque : Une fois que vous avez créé un gros groupe de volumes ou que vous avez converti un groupe de volumes à un gros format de groupe de volumes, vous ne pouvez pas le ramener à un niveau inférieur à AIX Version 4.3.2.

Partitions physiques

Lorsque vous ajoutez un volume physique à un groupe de volumes, ce volume est partagé en unités d'espace contiguës, de taille égale, appelées *partitions physiques*. La partition physique représente la plus petite unité d'affectation d'espace de stockage contigu sur un volume physique.

Les volumes physiques héritent de la taille de partition physique du groupe de volumes, définie au moment de la création de ce groupe (par exemple, avec la commande **mkvg -s**). Groupe de volumes contenant trois volumes physiques : présente la relation entre les partitions physiques sur les volumes physiques et les groupes de volumes.



A Volume Group Containing Three Physical Volumes

Volumes logiques

Vous pouvez créer des volumes logiques dans un groupe de volumes existant. Pour l'utilisateur et les applications, le *volume logique*, bien que résidant sur des partitions physiques non contiguës ou sur plusieurs volumes physiques, correspond à un volume disque unique, contigu et extensible. Pour créer des volumes logiques supplémentaires, utilisez la commande **mklv**. Elle permet de spécifier le nom du volume logique et de définir ses caractéristiques, y compris le nombre et l'emplacement des partitions logiques. Pour modifier un volume logique existant (nom et caractéristiques), utilisez la commande **chlv**, et pour en augmenter le nombre de partitions logiques, la commande **extendlv**. La taille maximale par défaut d'un volume logique à la création est de 128 partitions logiques. Toutefois, vous pouvez augmenter ce nombre avec la commande **chlv**.

Remarque : Après la création d'un volume logique, son état (LV STATE), accessible par la commande **lslv**, est `closed` (fermé). Il devient `open` (ouvert), par exemple, quand un système de fichiers a été créé dans le volume logique et monté.

Vous pouvez copier les volumes logiques (commande **cplv**), en afficher la liste (commande **lslv**), les supprimer (commande **rmlv**), et augmenter/réduire le nombre de copies qu'ils gèrent (commandes **mklvcopy** et **rmlvcopy**). En outre, lorsque le groupe de volumes est restructuré, les volumes logiques peuvent être réaffectés.

Le système vous permet de définir jusq'à 256 volumes logiques (512 dans le cas d'un gros groupe de volumes) par groupe de volumes, mais en réalité, le nombre à spécifier dépend de la capacité de stockage physique définie pour le groupe et de la taille affectée aux volumes logiques.

Partitions logiques

Lors de la création d'un volume logique, vous devez spécifier le nombre voulu de *partitions logiques*. Une partition logique correspond à une, à deux ou à trois partitions physiques, selon le nombre d'instances de données à gérer. Une seule instance signifie une seule copie du volume logique (valeur par défaut). Dans ce cas a lieu un mappage direct entre la partition logique et la partition physique. Toute instance, y compris la première, est qualifiée de copie. L'emplacement des partitions physiques (par exemple, leur proximité physique) se définit lors de la création du volume logique (au moyen d'options).

Systèmes de fichiers

Le volume logique définit l'affectation d'espace disque jusqu'au niveau de la partition physique. Des niveaux plus pointus de gestion de données sont obtenus par des composants logiciels de niveau plus élevé, tels que le gestionnaire de mémoire virtuelle ou le système de fichiers. Ainsi, l'ultime étape de l'évolution du disque est la création de *systèmes de fichiers*. Vous pouvez en créer un par volume logique. Pour créer un système de fichiers, utilisez la commande **crfs**. Pour plus de détails, reportez-vous à "Systèmes de fichiers - généralités", page 7-2.

Restrictions

La gestion du stockage sur volume logique a ses limites, indiquées dans le tableau ci–après. Bien que le nombre maximal de volumes physiques soit par défaut de 32 par groupe de volumes (128 dans le cas d'un gros groupe de volumes), vous pouvez définir le nombre maximal de groupes de volumes (variable définie par l'utilisateur) avec la commande **mkvg**. Pour le groupe rootvg, cette variable est automatiquement définie à la valeur maximum par le système lors de l'installation.

```
MAXPVS: 32 (128 big volume group)
MAXLVS: 255 (512 big volume group)
```

Restrictions	
Groupe de volumes	255 par système
Volume physique	(MAXPVS / facteur groupe de volumes) par groupe de volumes
Partition physique	(1016 x facteur groupe de volumes) par volume physique (jusqu'à 1024 Mo par taille)
Volume logique	MAXLVS par groupe de volumes
Partition logique	(MAXPVS * 1016) par volume logique

Si vous aviez déjà créé un groupe de volumes avant l'entrée en vigueur de la restriction des partitions physiques à 1016 par volume physique, les partitions anciennes du groupe de volumes ne sont pas correctement repérées tant que vous n'avez pas converti le groupe de volumes dans un format pris en charge. Pour cela, vous pouvez utiliser la commande **chvg –t**. Le facteur qui permet d'adapter le plus grans disque au groupe de volumes est choisi par défaut.

Par exemple, si vous avez créé un groupe de volumes avec un disque de 9 Go et une taille de partitionde 4 Mo, le groupe de volumes contient environ 2250 partitions. Avec un facteur de conversion de 3 (1016 * 3 = 3048), les 2250 partitions seront repérées correctement. En utilisant un facteur de conversion plus grand, vous pourriez utiliser une plus grand disque de partitions jusqu'à 1016* facteur. Vous pouvez également utiliser une facteur de conversion plus grand lorsque vous créez un groupe de volumes avec un plus grand disque et une petite taille de partition.

Ces opérations réduisent le nombre total de disques que vous pouvez ajouter à un groupe de volumes. Le nouveau nombre maximum de disques que vous pouvez ajouter est égal à 32/facteur. Par exémple, un facteur de 2 fait passer le nombre maximum de disques dans un groupe de volumes à 16 (32/2).

Remarque : Une fois que vous avez converti un groupe de volumes, vous ne pouvez pas le ramener à un niveau inférieur à AIX Version 4.3.1.

Gestionnaire de volumes logiques (LVM)

Le gestionnaire de volumes logiques (LVM) comprend le jeu de commandes du système d'exploitation, les sous-routines de bibliothèque et les autres outils de mise en œuvre et de contrôle du stockage sur volume logique. LVM contrôle les ressources disque par le mappage des données des disques *physiques* avec une vue *logique*, plus souple et plus simple, de l'espace de stockage. Pour ce faire, LVM fait appel à une couche du code du gestionnaire d'unités à un niveau supérieur à celui des gestionnaires d'unités de disque traditionnels.

LVM comprend le gestionnaire d'unités de *volume logique* (LVDD) et la bibliothèque d'interface de sous-routines LVM. LVDD est un gestionnaire de pseudo-unités contrôlant et traitant l'ensemble des entrées/sorties. Il traduit les adresses logiques en adresses physiques et envoie les demandes d'E/S aux gestionnaires d'unités spécifiques. La *bibliothèque d'interface de* sous-routines *LVM* contient des routines dédiées aux commandes de gestion du système, pour l'exécution de tâches afférentes à la gestion des volumes logiques et physiques. L'interface de programmation de la bibliothèque permet d'étendre les fonctions des commandes de gestion système pour volumes logiques.

Pour plus de détails sur le fonctionnement de LVM, reportez-vous à "Understanding the Logical Volume Device Driver" dans *AIX Version 4 Kernel Extensions and Device Support Programming Concepts* et à "Logical Volume Programming Overview" dans *AIX Version 4 General Programming Concepts: Ecriture et mise au point de programmes*.

Concepts de quorum

Les sections suivantes décrivent le processus vary-on et le quorum, par lesquels LVM garantit qu'un groupe de volumes est prêt pour l'exploitation et contient les données les plus à jour.

Processus vary-on

Les commandes **varyonvg** et **varyoffvg** activent et désactivent (en le mettant ou non en fonction) un groupe de volumes défini. L'activation du groupe doit être effectuée avant que le système n'y accède. Pendant le processus (activation), LVM lit les données de gestion dans les volumes physiques définis (dans le groupe de volumes). Ces données, qui comprennent une zone descripteur de groupe de volumes (VGDA) et une zone état de groupe de volumes (VGSA), sont enregistrées dans chaque volume physique du groupe de volumes.

Les données de la zone VGDA décrivent, pour chaque volume logique du groupe, le mappage des partitions physiques avec les partitions logiques, et d'autres données importantes, y compris l'horodateur. Celles de la zone VGSA indiquent les partitions physiques anciennes et les volumes physiques absents (non disponibles ou actifs) lors d'une tentative de mise en fonction (vary-on) d'un groupe de volumes.

Quand le processus vary-on ne donne pas accès à un ou à plusieurs volumes physiques définis dans le groupe de volumes, la commande affiche les noms et états de tous les volumes physiques définis pour ce groupe. Ceci vous permet de décider de poursuivre ou non le traitement avec ce groupe de volumes. Pour plus de détails sur la signification des états de volumes physiques affichés par la commande **varyonvg**, reportez-vous à la sous-routine **lvm_varyonvg**.

Quorum

Un quorum est un vote du nombre de zones VGDA/VGSA actives. En cas de défaillance du disque, le quorum garantit l'intégrité des données des zones VGDA/VGSA. Dans un groupe de volumes, chaque disque physique possède au moins une zone VGDA/VGSA. Quand un groupe de volumes est créé sur un seul disque, deux zones VGDA/VGSA résident sur ce disque. Si le groupe est constitué de deux disques, l'un deux a deux zones VGDA/VGSA et l'autre une seule. Si le groupe de volumes comporte trois disques ou plus, une seule zone VGDA/VGSA réside sur chaque disque.

Dans un groupe de volumes de deux disques, si le disque ne possédant qu'une zone VGDA/VGSA est perdu, le quorum est maintenu parce que deux des trois zones VGDA/VGSA sont accessibles. Le quorum est perdu quand un nombre de disques et leurs zones VGDA/VGSA sont inaccessibles, générant une majorité de 51 % des zones VGDA/VGSA disparues. Si, au contraire, l'autre disque est perdu, le quorum l'est aussi. Plus un groupe comporte de disques, plus ses chances sont grandes de conserver le quorum en cas de défaillance d'un disque.

Quand un quorum est perdu, la mise hors fonction du groupe est automatique et (LVM) n'a plus accès aux disques de ce groupe. Ainsi, aucune E/S disque n'étant possible sur ce groupe, les données ne peuvent être perdues ou supposées écrites en cas de problème de disque physique. En outre, la mise hors fonction permet de signaler à l'utilisateur, par le biais du journal des erreurs, le problème matériel et la nécessité de maintenance.

Vous pouvez opter pour la poursuite de l'exploitation d'un groupe de volumes qui a perdu le quorum. Vous pouvez le faire en désactivant la vérification du quorum correspondant. Auquel cas le groupe n'est pas soumis au quorum. Ceci est généralement le cas des groupes dont les volumes font l'objet d'écriture miroir. Si un disque est perdu et qu'une copie du volume logique réside sur un disque activé et accessible, les données ne sont pas perdues. Toutefois, le cas peut se produire où des données (y compris des copies) d'un groupe non soumis à quorum, avec ou sans écriture miroir, résident sur le ou les disques indisponibles ; même si le groupe reste en fonction, ces données risquent de ne pas être accessibles.

Mise en fonction forcée

Attention : Cette procédure doit faire exception et n'être appliquée qu'en dernier recours (par exemple, pour récupérer des données d'un disque défectueux). Elle ne doit être appliquée qu'une fois toutes les autres causes possibles du problème vérifiées (matériel, câbles, cartes, sources d'alimentation). Elle ne garantit pas l'intégrité des données résidant dans les copies sélectionnées des zones VGDA/VGSA.

Si vous décidez la mise en fonction forcée d'un groupe de volumes en ne le soumettant pas au quorum, sur tous les volumes physiques absents au cours du processus vary-on, PV STATE passe à l'état `removed` (supprimé). Ceci signifie que toutes les copies des zones VGDA/VGSA seront supprimées de ces volumes physiques. Ensuite, ces volumes ne feront plus partie de la vérification du quorum, ni des volumes actifs du groupe, tant qu'ils ne seront pas réintégrés à ce groupe.

Passer outre à la mise en fonction échouée pour accéder aux données des disques disponibles du groupe est possible si :

- les volumes physiques indisponibles semblent définitivement endommagés.
- vous êtes en mesure de confirmer que, parmi les volumes physiques accessibles, au moins un disque (contenant une copie valide des zones VGDA/VGSA) était en ligne à la dernière mise en fonction du groupe de volumes. La configuration des volumes physiques absents doit être annulée et les volumes mis hors tension jusqu'à leur diagnostic et réparation.

Cette procédure représente un moyen d'éviter de perdre le quorum lorsqu'un disque est absent ou défectueux, et requiert une réparation.

1. Supprimez temporairement le volume du groupe avec la commande **chpv –vr** . Il ne sera pas pris en compte dans le calcul du quorum. Toutefois, pour un groupe de deux disques, cette commande échoue si elle est appliquée à celui des disques qui possède les deux zones VGDA/VGSA. La commande empêchera de perdre le quorum.
2. Si vous ôtez un disque pour le réparer, mettez le système hors tension avant le retrait de ce disque. Lorsque le disque réintègre le système une fois réparé, exécutez la commande **chpv –v** pour qu'il soit pris en compte à la vérification du quorum du groupe de volumes concerné.

   Remarque : La commande **chpv** ne sert qu'en cas de modification de la vérification du quorum. Si le disque retiré ne réintègre pas le système, son contenu doit être transféré ou copié.

Groupes de volumes à l'état "nonquorum"

LVM désactive automatiquement le groupe de volumes n'atteignant pas le quorum de VGDA ou de VGSA. Toutefois, une option permet au groupe de rester en fonction tant qu'une zone VGDA/VGSA est intacte et ne soumet pas ce groupe au quorum. Cette option crée un *groupe à l'état "nonquorum".* LVM requiert l'accès à tous les disques des groupes de volumes non soumis au quorum avant d'autoriser la réactivation pour garantir des zones VGDA/VGSA à jour.

Cette procédure concerne de préférence les systèmes dont chaque volume logique possède au moins deux copies.

En cas de défaillance de disque, le groupe de volumes reste actif tant que réside sur un disque une copie intacte d'un volume logique.

Remarque : Cette procédure est applicable aux groupes de volumes utilisateur et au groupe rootvg, toutefois, avec des méthodes différentes quant à la configuration de la fonction et à la reprise après incident matériel. Veillez à employer la méthode adéquate.

Développement d’une stratégie relative aux groupes de volumes

L’incident matériel auquel le système de stockage est le plus souvent confronté est la défaillance du disque, suivi de la défaillance des cartes et du système d’alimentation. La configuration des volumes logiques joue un rôle important de prévention des incidents de disque. Reportez-vous à “Développement d’une stratégie relative aux volumes logiques”, page 6-13. En outre, comme expliqué plus loin, la taille du groupe de volumes a son importance.

Prévenir les défaillances d’alimentation et de cartes suppose une configuration matérielle particulière pour tout gruope de volumes spécifiques, avec deux cartes et au moins un disque par carte, avec utilisation de disques miroirs par le biais de cartes et une configuration de groupe de volumes “nonquorum”. L’investissement en jeu pour de telles configurations n’est pas à la portée de tous les sites ou systèmes. Il est surtout recommandé lorsque la haute disponibilité est l’exigence prioritaire du système. Selon la configuration, la haute disponibilité est apte à couvrir les incidents matériels se produisant entre la dernière sauvegarde et l’entrée des données en cours; Elle ne couvre pas les fichiers supprimés par accident.

Prérequis

Une bonne connaissance des informations fournies à la section “Stockage sur volume logique-généralités”, page 6-2.

Création de groupes de volumes distincts

Voici différentes raisons de structurer des volumes physiques en groupes de volumes indépendants de rootvg :

- Pour sécuriser et faciliter les opérations de maintenance.
  - Les mises à jour du système d’exploitation, les réinstallations et les reprises sur panne système sont plus fiables avec des systèmes de fichiers utilisateur séparés du système d’exploitation, préservant ainsi les fichiers utilisateur.
  - Les opérations de maintenance sont facilitées : vous pouvez actualiser le système d’exploitation ou le reinstaller sans restaurer les données utilisateur. Par exemple, avant de lancer une mise à jour, vous pouvez retirer du système un groupe de volumes utilisateur en démontant ses systèmes de fichiers, en le désactivant (avec la commande **varyoffvg**), puis en exportant le groupe (avec la commande **exportvg**). Ensuite, après la mise à jour, il suffit de réintégrer le groupe de volumes utilisateur (avec la commande **importvg**), puis de remonter ses systèmes de fichiers.
- Pour définir des tailles de partitions physiques hétérogènes. Tous les volumes physiques d’un même groupe de volumes doivent avoir la même taille de partition physique. Pour que des volumes physiques aient des tailles de partition physique différentes, placez chaque taille dans un groupe de volumes distinct.
- Pour définir des caractéristiques de quorum hétérogènes. Vous pouvez prévoir un groupe de volumes distinct pour un système de fichiers de type “nonquorum”, tous les autres systèmes de fichiers faisant partie du groupe de volumes soumis au quorum.
- Pour obtenir plusieurs journaux JFS ou des journaux JFS dédiés à un volume physique, l’objectif étant de réduire les goulets d’étranglement, tout particulièrement sur les serveurs.
- Pour la sécurité. Par exemple, vous pouvez être amené à retirer de nuit un groupe de volumes.
- Pour commuter des volumes physiques entre systèmes. La commutation de volumes physiques entre systèmes est possible en créant un groupe de volumes par système connecté à une carte accessible aux systèmes, ceci sans interrompre les opérations en cours (voir les commandes **varyoffvg**, **exportvg**, **importvg** et **varyonvg**).

- Pour supprimer des disques du système sans interrompre le fonctionnement normal du système. En créant un groupe de volumes distinct pour les disques amovibles, à condition qu'il ne s'agisse pas du groupe rootvg, vous pouvez désactiver ces disques et les retirer physiquement pendant que le système est en cours d'exploitation normale et sans affecter les autres groupes du volume.

Haute disponibilité face aux incidents de disque

Les principales mesures de prévention des incidents de disque comprennent les définitions de configuration de volume logique, telles que l'écriture miroir. Bien que secondaires, les informations ci-après ont des répercussions économiques significatives, impliquant le nombre de volumes physiques par groupe de volumes :

- Le quorum, configuré par défaut, maintient le groupe de volumes en fonction tant que le quorum des disques (51%) est atteint. Pour plus de détails, reportez–vous à "Processus vary–on" à la section "Stockage sur volume logique-généralités", page 6-7. Dans la plupart des cas, trois disques minimum par groupe de volumes sont nécessaires aux copies miroir, à titre de prévention des incidents de disque.
- Le configuration de type "nonquorum" maintient le groupe de volumes en fonction tant qu'une zone VGDA est disponible sur le disque (reportez–vous à "Passage d'un groupe de volumes à l'état nonquorum" dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*). Dans la plupart des cas, seuls deux disques par groupe de volumes sont nécessaires aux copies miroir, à titre de prévention des incidents de disque.

Au moment de déterminer vos besoins en disques dans chaque groupe de volumes, prévoyez l'espace disque nécessaire à l'écriture miroir des données. Sachez que les copies miroir et le transfert de données entre disques se font uniquement à l'intérieur du même groupe de volumes. Lorsque le site exploite des systèmes de fichiers volumineux, l'espace disque nécessaire à l'écriture miroir peut s'avérer important par la suite. En outre, prenez en compte l'impact de la disponibilité des définitions inter-disque pour les copies de volumes logiques et l'affectation intra–disque pour un volume logique.

Haute disponibilité face aux incidents de carte ou d'alimentation

Pour prévenir les incidents de carte ou d'alimentation, en fonction de la rigueur de vos besoins :

- Utilisez deux cartes, installées ou non dans la même armoire. Si les cartes ne sont pas situées dans la même armoire, elles seront toutes deux protégées en cas de problème d'alimentation dans une armoire.
- Utilisez deux cartes et connectez un ou plusieurs disques sur chacune : ceci pour pré–venir tout incident de carte (ou d'alimentation lorsque les cartes sont installées dans des armoires séparées) en maintenant le quorum dans le groupe de volumes, compte tenu de *l'écriture miroir croisée* (pour une partition logique, les copies ne pouvant partager le même volume physique) entre les volumes logiques du disque A (carte A) et les volumes logiques du disque B (carte B). Ceci signifie que vous copiez les volumes logiques des disques connectés à la carte A sur les disques connectés à la carte B, et inversement.
- Configurez tous les disques à partir des deux cartes dans le même groupe de volumes. Ainsi, au moins une copie de voume logique reste intact si une carte est défectueuse, ou, en cas d'armoires séparées, si l'alimentation est défaillante.
- Passez le groupe de volumes à l'état "nonquorum" Ainsi, le groupe reste actif tant qu'une zone VGDA est accessible sur un disque quelconque du groupe. Reportez-vous à "Passage d'un groupe de volumes à l'état nonquorum" dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.
- Si le groupe de volumes possède deux disques, configurez l'écriture miroir croisées entre les cartes. Si plusieurs disques sont connectés à chaque carte, configurez l'écriture miroir double en créant une copie miroir sur un disque utilisant la même carte et une autre sur un disque utilisant une autre carte.

Définition de la taille des partitions physiques

La taille de partition physique est définie au moment de la création du groupe de volumes. La taille par défaut est de 4 Mo. Elle est adaptée à la plupart des sites et des systèmes, mais peut être modifiée, le cas échéant. Vous pouvez la réduire à 1 Mo pour gagner en souplesse, auquel cas vous aurez besoin d'un plus grand nombre de partitions. En outre, dans ce cas, le temps système étant accru au niveau de LVM, la performance du système risque d'être affectée.

Si vous optez pour les partitions de plus de 4 Mo, vous perdez en souplesse et ne bénéficiez pas d'économie d'espace. Par exemple, avec des partitions de 20 Mo, la taille du journal JFS devra êter de 20 Mo également alors que 4 Mo sont suffisants. Toutefois, cette possibilité est envisageable pour un site ou un système qui requiert des partitions plus grandes.

Lors de la création et de l'extension des partitions physiques, pour définir/redéfinir la taille, vous devez l'incrémenter par sa valeur, par exemple, pour une partition de 20 Mo, en calculant sur la base d'un incrément de 20 Mo.

Développement d’une stratégie relative aux volumes logiques

Les règles ci-après vous aideront à élaborer une stratégie en matière d’exploitation des volumes logiques, orientée à la fois sur la disponibilité, les performances et le coût.

La *disponibilité* représente l’aptitude à restaurer les données perdues en raison de problèmes de disques, de cartes et d’autres problèmes matériels. Cette restauration est effectuée à partir de copies effectuées et actualisées sur des disques séparés (et cartes) pendant l’exploitation normale du système.

La *performance* s’évalue par la vitesse moyenne d’accès aux données. Les règles telles que celles relatives au contrôle de l’écriture et à l’écriture miroir améliorent la disponibilité mais augmentent la charge du traitement, au détriment des performances. L’écriture miroir double ou triple la taille du volume logique. Améliorer la disponibilité, affecte généralement les performances. La répartition de fichiers sur plusieurs disques améliore les performances. A partir du système AIX Version 4.3.3, l’écriture miroir est possible.

En contrôlant l’affectation des données des disques et entre disques, vous pouvez régler le système de stockage pour optimiser les performances. Pour plus de détails, reportez-vous aux sections relatives au contrôle et à la mise au point de la mémoire et des E/S disque dans *AIX - Guide d’optimisation*.

Les informations qui suivent vous permettront de trouver le meilleur compromis entre performance, disponibilité et coût. Ne perdez pas de vue que l’amélioration de la disponibilité réduit les performances, et vice versa. Le traitement miroir peut améliorer les performances si LVM choisit la copie sur le disque le moins chargé pour la lecture.

> **Remarque :** L’écriture miroir n’empêche pas la perte de fichiers supprimés accidentellement ou en raison de problèmes logiciels. Dans ce cas, les fichiers ne peuvent être restaurés qu’à partir de sauvegardes sur bande ou disquette.

Les points suivants sont traités dans cette section :

Prérequis

Une bonne connaissance des informations indiquées à la section ”Stockage sur volume logique - généralités”, page 6-2.

Analyse des besoins en performance et disponibilité

Déterminez si l’importance des données à stocker sur le volume logique justifie le traitement miroir et le coût correspondant en terme d’espace disque.

Les notions de performance et de traitement miroir ne sont pas toujours opposées. Si les instances (copies) des partitions logiques figurent sur des volumes physiques différents, de préférence connectés à des cartes distinctes, LVM peut améliorer les performances en ne lisant que la copie du disque qui a le plus d’espace libre. Le coût de l’écriture est presque toujours le même du fait de la mise à jour de chaque copie, excepté si les disques sont connectés à différentes cartes.

Si vous possédez un grand système de fichiers à accès séquentiel et sensible aux performances, vous pouvez envisager la répartition des fichiers sur plusieurs disques.

Normalement, lorsque des données sont mises à jour sur une partition logique, toutes les partitions physiques contenant cette partition sont automatiquement actualisées. Toutefois, la mise à jour des partitions physiques n'a pas toujours lieu, ceci en cas de dysfonctionnement du système ou de l'indisponibilité du volume physique au moment d'une mise à jour. LVM peut actualiser les partitions anciennes en recopiant dessus les partitions à jour. Ce processus est intitulé *synchronisation miroir*. Il peut avoir lieu au redémarrage du système, quand le volume physique est à nouveau disponible ou en exécutant la commande **syncvg**.

Le traitement miroir améliore la disponibilité du système de stockage, mais ne peut se substituer aux sauvegardes traditionnelles sur bande.

A partir du système AIX Version 4.3.3, le traitement miroir d'un volume logique d'amorçage est possible.

Vous devez exécuter **bosboot** après tout changement pouvant effecté la partition physique d'un volume logique d'amorçage. En d'autres termes, toute action telle que la modification du traitement miroir d'un volume logique d'amorçage nécessite l'exécution de **bosboot**.

Une tentative de cliché sur un volume logique avec copie miroir donne un cliché incohérent et doit donc être évitée. L'unité de cliché par défaut étant le volume de pagination primaire, vous devez créer un volume logique de cliché séparé pour permettre le traitement miroir des volumes logiques de pagination, et par conséquent pour permettre le traitement miroir de votre groupe de volumes également.

Règles de programmation des écritures miroir sur disque

Pour des données qui n'ont qu'une copie physique, LVDD traduit l'adresse des demandes de lecture ou d'écriture logique en adresse physique, et appelle le gestionnaire d'unités physiques approprié pour traiter la demande. Cette politique de copie unique ou d'écriture non miroir gère la réaffectation des blocs défectueux pour les demandes d'écriture et renvoie toutes les erreurs de lecture au processus appelant.

Si vous utilisez des volumes logiques en miroir, vous pouvez définir deux politiques de programmation différentes pour l'écriture sur disque dans le cadre d'un volume logique avec copies multiples : le traitement *séquentiel* et le traitement *parallèle*.

En traitement séquentiel, l'écriture sur plusieurs copies ou l'écriture miroir est traitée en séquence. Plusieurs partitions physiques représentant les copies miroir d'une seule partition logique sont qualifiées de primaires, secondaires et tertiaires. Les partitions physiques sont écrites séquentiellement (l'une après l'autre).

En traitement parallèle, l'écriture de toutes les partitions physiques d'une partition logique démarre en même temps. Elle prend fin après l'écriture de la partition physique la plus longue.

La lecture, en traitement séquentiel, est effectuée sur la copie miroir primaire. Si elle n'aboutit pas, elle est effectuée sur la copie suivante. A la seconde tentative de lecture sur la copie suivante, la copie primaire dont la lecture a échoué est corrigée par LVM, avec une réaffectation matérielle. Ainsi, le bloc défectueux empêchant la première lecture sera accessible ultérieurement.

Programmer le traitement parallèle peut améliorer les performances de lecture des entrées-sorties, les copies multiples permettant au système de traiter la lecture directement sur la copie accessible le plus rapidement.

(MWC (cohérence écrit–miroir) pour un volume logique

MWC identifie les partitions logiques susceptibles d'être incohérentes si le système ou le groupe de volumes n'est pas correctement fermé. Lorsque le groupe de volumes est à nouveau mis en fonction, ces informations sont utilisées pour rendre cohérentes les partitions logiques.

Si un volume logique utilise MWC, les requêtes portant sur ce volume logique sont retenues dans la couche de programmation jusqu'à ce que les blocs de la mémoire cache MWC puissent être mis à jour sur les volumes physiques cible. Une fois les blocs de mémoire cache MWC mis à jour, la requête passe aux opérations d'écriture des données physiques.

Lorsque MWC est utilisé, les performances système peuvent être diminuées. Ce fait est du à la surcharge de la journalisation et de la consignation par suite de l'activité de la requête d'écriture dans un LTG (Logical Track Group) (pages de 32 4 Ko ou 128 Ko). Cette surcharge n'existe que pour les écritures en miroir. Il est nécessaire d'assurer la cohérence des données entre les miroirs uniquement en cas de panne du système ou du groupe de volumes avant la fin de l'écriture vers tous les miroirs. Lorsque MWC n'est pas utilisé, les miroirs d'un volume logique en miroir peuvent être laissés dans un état incohérent au cas où le système ou le groupe de volumes tomberait en panne.

Après une panne, un volume logique en miroir avec MWC désactivé devrait effectuer une synchronisation forcée (**syncvg –f –l LV*nom***) avant l'utilisation des données du volume. Lorsque MWC est désactivé, les écritures restant à faire au moment de la panne peuvent laisser les miroirs dans un état incohérent lors de la prochaine mise en fonction du groupe de volumes. Il y a exception dans le cas de volumes logiques dont le contenu n'est valide que pendant l'ouverture du volume logique, par exemple les espaces de pagination.

Un volume logique en miroir n'est pas vraiment différent d'un volume logique non mis en miroir, en ce qui concerne l'opération d'écriture. A la fin de l'exécution de LVM avec une requête d'écriture, les données ont été écrites sur la ou les unités situées sous LVM. Le résultat de l'écriture est inconnu jusqu'à ce que LVM émette **iodone** sur une opération d'écriture. Tous les blocs subissant une écriture qui n'ont pas été terminés (**iodone**) au moment de l'arrêt d'une machine doivent être réécrits, qu'ils soient en miroir ou non, et quelle que soit la configuration de MWC.

MWC ne rend les miroirs cohérents que lorsque le groupe de volume est remis en fonction après une panne, en choisissant un miroir et en propageant ces données sur les autres miroirs. MWC ne conservant pas la trace des dernières données, (seul le suivi des LTG en cours d'écriture est effectué), il ne garantit pas la propagation des dernières données à tous les miroirs. C'est l'application au-dessus de LVM qui détermine la validité des données après une panne. D'après la perspective LVM, si l'application réemet toujours toutes les demandes d'écriture en suspens au moment de la panne, les miroirs éventuellement incohérents deviendront cohérents à la fin des opérations d'écriture (du moment que les blocs écrits sont les mêmes que ceux en suspens au moment de la panne).

Règles d'affectation inter-disque

Cette procédure permet de spécifier le nombre de disques sur lesquels sont situées les partitions physiques du volume logique. Celles-ci peuvent être placées sur un seul disque ou réparties sur l'ensemble des disques. Ce type d'affectation s'opère au moyen des deux options suivantes des commandes **mklv** et **chlv** :

- *Range* définit le nombre de disques utilisés pour une copie physique unique du volume logique.
- *Rigide* détermine si **mklv** peut aboutir lorsque plusieurs copies doivent occuper le même volume physique.
- Si les volumes logiques sont répartis sur plusieurs disques, les seules valeurs admises pour Range et **Rigide** sont **maximum** et yes.

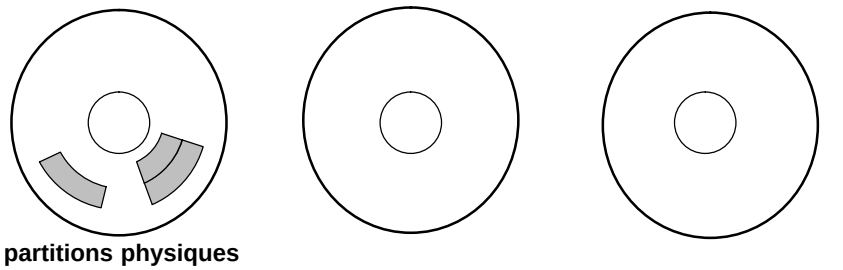
Copie unique du volume logique

Si vous optez pour l'affectation inter- (*Range* = **minimum**), les partitions physiques affectées au volume logique seront situées sur un seul disque, afin d'accroître la disponibilité. Avec l'affectation inter-disque optimale (*Range* = **maximum**), les partitions physiques sont situées sur plusieurs disque pour améliorer la performance. L'affectation de copies miroir aux partitions d'origine est décrite dans la section suivante.

Pour les volumes logiques de type "non miroir", prenez la valeur **minimum** qui confère au système une disponibilité optimale (quant à l'accès aux données en cas de d'incident matériel). Le paramètre **minimum** indique qu'un volume physique doit contenir, si possible, toutes les partitions physiques d'origine de ce volume. Si le programme d'affectation a besoin de plusieurs volumes physiques, il utilise le nombre minimal, tout en restant cohérent par rapport aux autres paramètres.

En utilisant le nombre minimal de volumes physiques, vous réduisez le risque de perdre des données en cas de défaillance d'un disque. Tout volume physique supplémentaire utilisé pour une copie physique unique accroît ce risque. En cas de défaillance d'un volume physique, le risque de perdre des données pour un volume logique de type "non miroir" réparti sur quatre volumes physiques est quadruplé par rapport à un volume logique logé dans un seul volume physique.

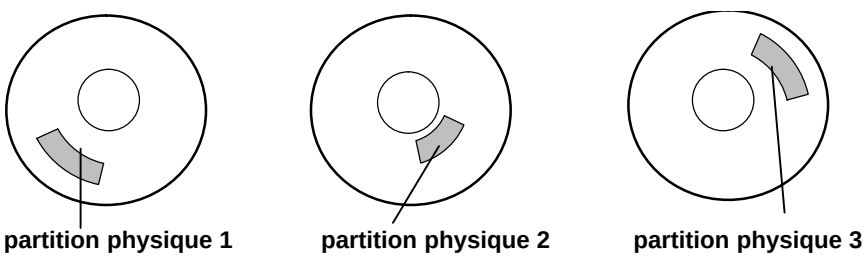
La figure ci-après illustre une affectation inter-disque minimale :



Affectation inter-disque minimale :
toutes les partitions physiques sont sur un seul disque.

Le paramètre **maximum**, compte tenu d'autres contraintes, répartit de façon aussi égale que possible, les partitions physiques du volume logique sur le plus grand nombre de volumes physiques possible. Cette option vise la performance, en tendant à réduire le temps d'accès du volume logique. Pour améliorer la disponibilité, elle ne doit être utilisée qu'avec des volumes logiques avec traitement miroir.

La figure ci–après illustre une affectation inter–disque optimale.



Affectation inter-disque optimale :
les partitions physiques sont réparties sur plusieurs disques.

Ces paramètres sont également applicables à l'extension et à la copie d'un volume logique existant. L'affectation de nouvelles partitions physiques est déterminée en fonction de la politique en cours, ce à l'emplacement où sont situées les partitions physiques existantes.

Copies multiples du volume logique

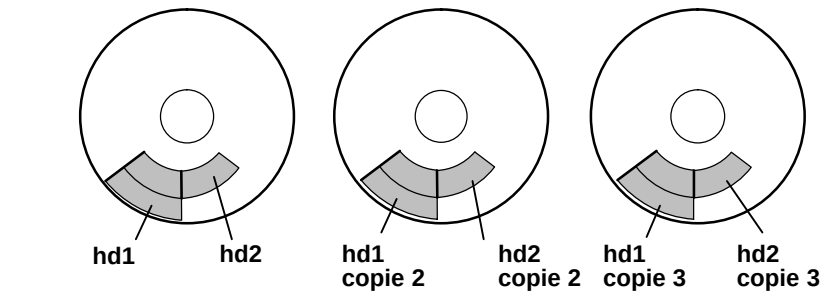
L'affectation d'une seule copie d'un volume logique est une procédure relativement simple. Toutefois, lors de la création de copies miroir, l'affectation résultante est quelque peu complexe. Les figures suivantes illustrent l'emploi des paramètres **minimum** et **maximum** *(Range)* pour la première instance d'un volume logique, et les définitions possibles de *rigide* pour les copies miroir de ce volume.

Par exemple, avec des copies miroir du volume logique, **minimum** permet d'affecter, si possible, les partitions physiques contenant la première instance du volume logique sur un seul volume physique. Puis, en fonction de la définition de *Rigide*, la ou les copies supplémentaires sont affectées sur le même volume physique ou sur des volumes distincts. Autrement dit, l'algorithme utilise le plus petit nombre possible de volumes physiques, en respectant les contraintes imposées par les autres paramètres tels que *Rigide*, pour maintenir toutes les partitions physiques.

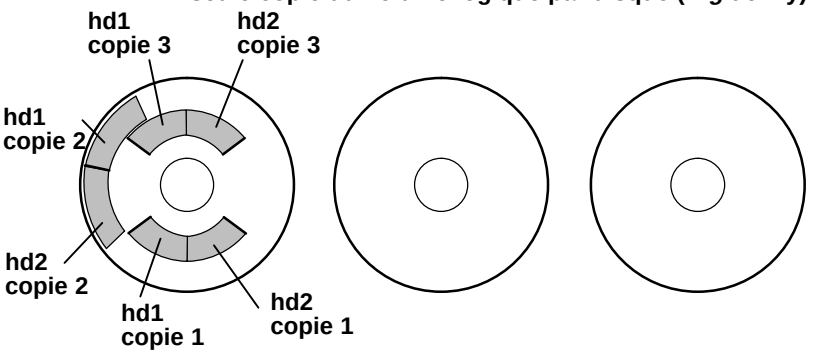
Rigide = **y** signifie que chaque copie de la partition logique est placée sur un volume physique différent. *Rigide* = **n** signifie que les copies ne sont pas tenues d'être situées sur des volumes distincts.

> **Remarque :** Lorsque le nombre de volumes physiques du groupe est inférieur au nombre de copies par partition logique sélectionnée, affectez la valeur **n** à *rigide*. Si *Rigide* = **y**, un message d'erreur est renvoyé quand vous tentez de créer le volume logique.

La figure Affectation inter–disque minimale/*Rigide* illustre une règle d'affectation inter–disque minimale avec différents paramètres *Rigide* :

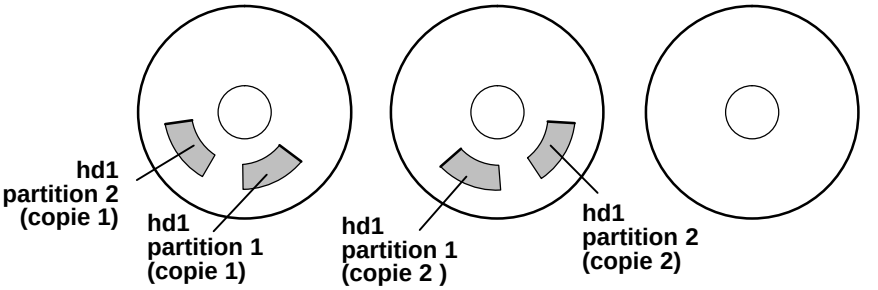


Affectation inter-disque minimale avec une seule copie du volume logique par disque (*Rigide* = y)

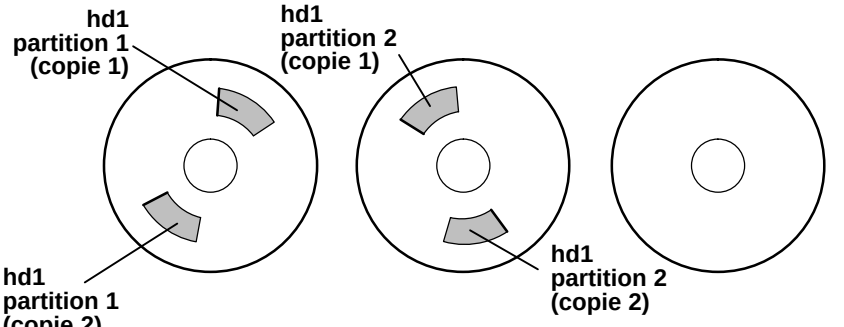


Affectation inter-disque minimale avec plusieurs copies du volume logique par disque (*Rigide* = n)

La figure Affectation inter-disque optimale/*Rigide* illustre une règle d'affectation inter-disque optimale avec différents paramètres *Rigide* :



Affectation inter-disque optimale avec une seule copie du volume logique par disque (*Rigide* = y)



Affectation inter-disque optimale avec plusieurs copies du volume logique par disque (*Rigide* = n)

Règles d’affectation intra-disque

Plus une partition physique est proche du centre d’un volume physique, plus le temps de recherche moyen est rapide, la distance de recherche moyenne étant la plus réduite par rapport aux autres zones du disque.

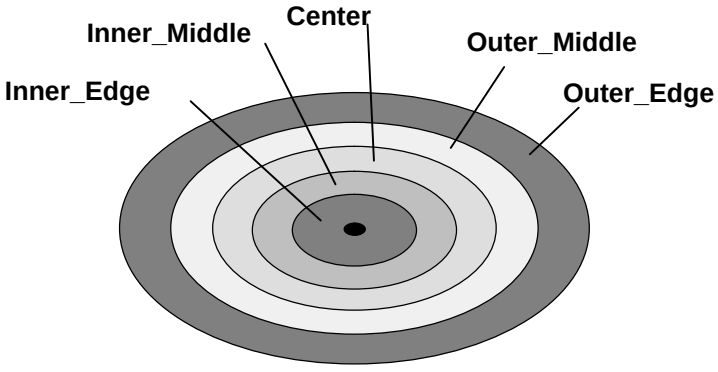
Le journal du système de fichiers est le candidat idéal à affecter au centre d’un volume physique, le système d’exploitation l’utilisant très souvent. A l’inverse, le volume logique d’amorçage, utilisé peu fréquemment, devrait être affecté au milieu ou à la limite extérieure du volume physique.

En règle générale, plus les E/S (en valeur absolue ou pendant l’exploitation d’une application importante) sont nombreuses, plus les partitions physiques du volume logique doivent être affectées près du centre des volumes physiques. Deux exceptions à cette règle :

1. Les volumes logiques sur disques de 200 Mo, de 540 Mo ou de 1 Go contenant des fichiers séquentiels volumineux doivent être placés au bord du volume physique, le mode séquentiel étant plus performant à cet emplacement (en raison du plus grand nombre de blocs par piste).
2. Les volumes logiques miroir configurés avec MWC (Cohérence-écrit-miroir) sur **ON** doivent également être placés au bord extérieur, le système inscrivant les données MWC à cet emplacement. Quand le traitement miroir n’est pas activé, MWC n’est pas applicable, et n’affecte pas les performances. Reportez-vous à la section relative aux performances des techniques de disque miroir dans les ”Règles d’installation associées aux performances” du manuel *AIX - Guide d’optimisation*.

La politique d’affectation intra-disque est fondée sur les cinq régions de disque où placer les partitions physiques. Ces cinq régions sont : *outer edge (bord extérieur)*, *inner edge (bord intérieur), outer middle (milieu extérieur)*, *inner middle (milieu intérieur)***,** et *center (centre)*. Au bord, le temps de recherche moyen est le plus lent, d’où des temps de réponse plus longs pour les applications. Les partitions du centre ont le temps de recherche moyen optimal, donc les meilleurs temps de réponse. Toutefois, le centre contient moins de partitions sur un volume physique que les autres régions du disque.

Les cinq régions où affecter les partitions physiques dans un volume physique sont illustrées ci-après.



Les cinq régions d’un disque

Affectations combinées

Le choix de politiques d'affectation inter-disque et intra-disque incompatibles peut générer des résultats imprévisibles. Le système donnera la priorité à un type d'affectation. Par exemple, si vous optez pour une affectation intra-disque de type "center" au centre et inter-disque de type minimum, l'affectation inter-disque sera prioritaire. Le système placera, si possible, toutes les partitions du volume logique sur un seul disque, même si elles ne tiennent pas toutes au centre du disque. Avant de combiner différentes politiques, assurez-vous d'en avoir assimilé les interactions.

Affectation affinée avec des fichiers mappe

Si les options par défaut des politiques d'affectation interdisque et intradisque ne sont pas adaptées à vos besoins, vous pouvez créer des fichiers mappe pour spécifier l'ordre et l'emplacement exacts des partitions physiques d'un volume logique.

Pour cela, vous pouvez faire appel à Web-based System Manager, SMIT ou à la commande **mklv –m**.

Remarque : L'option **–m** est incompatible avec la répartition de fichiers sur plusieurs disques.

Par exemple, pour créer un volume logique de dix partitions nommé lv06 dans rootvg en partitions 1 à 3, 41 à 45 et 50 à 60 sur le disque hdisk1, procédez comme suit :

1. Exécutez la commande :

   ```
   lspv –p hdisk1
   ```

   pour vérifier la disponibilité des partitions physiques que vous souhaitez affecter.

2. Créez un fichier, tel que `/tmp/mymap1` contenant :

   ```
   hdisk1:1–3
   hdisk1:41–45
   hdisk1:50–60
   ```

   La commande **mklv** affectera les partitions physiques dans l'ordre où elles figurent dans le fichier mappe. Assurez-vous que ce fichier contient un nombre suffisant de partitions à affecter à l'ensemble du volume logique spécifié par la commande **mklv**. (Au besoin, vous pouvez en afficher la liste.)

3. Exécutez la commande :

   ```
   mklv –t jfs –y lv06 –m /tmp/mymap1 rootvg 10
   ```

Développement d'une stratégie relative à la répartition du volume logique

La répartition des volumes logiques concerne les systèmes de fichiers séquentiels volumineux, sensibles aux performances, et à forte fréquence d'accès. Ce procédé vise à améliorer les performances.

Remarque :

1. Vous pouvez importer sur un système en version 4.1, un groupe de volumes créé en version 3.2 et inversement, si la répartition n'a pas été appliquée. Cette répartition une fois configurée, l'importation en version 3.2 n'est pas possible. **mksysb** ne peut pas restaurer un volume logique réparti dès lors que l'image **mksysb** est restaurée.
2. Un groupe de volumes réparti ne peut pas être importé dans une version antérieure à la version 4.3.3.
3. Un espace de cliché ou un volume logique d'amorçage ne doit pas être réparti.

Pour créer un volume logique de 12 partitions nommé `lv07` dans *NomVG* réparti sur hdisk1, hdisk2 et hdisk3, avec une taille de 16 Ko pour la répartition, entrez :

```
mklv –y lv07 –S 16K NomVG 12 hdisk1 hdisk2 hdisk3
```

Pour créer un volume logique de 12 partitions nommé `lv08` dans *NomVG* réparti sur trois disques quelconques, avec une taille de 8 Ko pour la répartition, entrez :

```
mklv –y lv08 –S 8K –u 3 NomVG 12
```

Pour plus de détails sur l'amélioration des performances avec cette stratégie, reportez-vous à *AIX - Guide d'optimisation*.

Règles de contrôle de l'écriture

L'option de contrôle de l'écriture lit en temps réel toutes les opérations d'écriture pour vérifier qu'elles sont lisibles. Un message d'erreur s'affiche lorsqu'elles sont incorrectes. Cette option améliore la disponibilité, ceci au détriment de la performance, en raison du temps de lecture nécessaire. Vous pouvez définir cette option sur un volume logique lors de sa création (**mklv**) ou, ultérieurement, en le modifiant (**chlv**).

Mise en œuvre des règles relatives aux groupes de volumes

1. Exécutez la commande **lspv** pour vérifier les volumes physiques affectés et ceux qui sont libres. Dans une configuration standard, avec quorum, plus le nombre de disques du groupe de volumes est important, plus le quorum a des chances d'être maintenu en cas de défaillance de disque. Dans un groupe de type ”nonquorum”, deux disques minimum sont requis.
2. Pour assurer le quorum, ajoutez un ou plusieurs volumes physiques (reportez–vous à Ajout d'un disque fixe sans données à un groupe de volumes existant dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*) ou à Ajout d'un disque fixe sans données à un nouveau groupe de volumes*AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*. Pour passer à l'état “nonquorum”, reportez–vous à la section Passage d'un volume de groupe utilisateurs à l'état “nonquorum”*AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités* .
3. La configuration standard comprend un seul groupe de volumes constitué de plusieurs volumes physiques connectés à la même carte disque et à d'autres éléments matériels nécessaires. Redéfinir la configuration matérielle représente un travail élaboré. Vous n'avez besoin de matériel externe que lorsque votre site requiert une haute disponibilité.

Mise en oeuvre des règles relatives aux groupes de volumes

1. Exécutez la commande **lspv** pour vérifier les volumes physiques affectés et ceux qui sont libres. Dans une configuration standard, avec quorum, plus le nombre de disques du groupe de volumes est important, plus le quorum a des chances d'être maintenu en cas de défaillance de disque.Dans un groupe de type "nonquorum", deux disques minimum sont requis.
2. La configuration standard comprend un seul groupe de volumes constitué de plusieurs volumes physiques connectés à la même carte disque et à d'autres éléments matériels nécessaires. Redéfinir la configuration matérielle représente un travail élaboré. Vous n'avez besoin de matériel externe que lorsque votre site requiert une haute disponibilité.

Limites de LVM - avertissements

- Tel qu'est conçu LVM, chaque partition logique est mappée sur une partition physique. De même, chaque partition physique est mappée sur un certain nombre de secteurs de disque. Le nombre de partitions physiques par disque contrôlé par LVM est limité à 1016. Dans la plupart des cas, un disque n'en utilise pas autant. En outre, pendant l'exécution d'une commande **mkvg**, la taille par défaut des partitions physiques est de 4 Mo, qui implique que des disques de 4 Go maximum peuvent faire partie d'un groupe de volumes.

  En conséquence, l'ajout d'un disque de plus de 4Go à un groupe de volumes (basé sur la valeur par défaut de 4 Mo de la partition physique) ne peut pas aboutir. Il génére alors l'avertissement suivant :

  ```
  The Physical Partition Size of <number A> requires the creation
  of <number B>:
  partitions for hdiskX.  The system limitation is <number C>
  physical partitions
  per disk at a factor value of <number D>. Specify a larger
  Physical Partition
  Size or a larger factor value in order create a volume group on
  this disk.
  ```

  Cette limite est renforcée dans les deux situations suivantes :

1. L'utilisateur tente de créer un groupe de volumes via la commande **mkvg**, avec un nombre de partitions physiques supérieur à 1016.

   Une nouvelle sélection de taille de partition physique parmi les valeurs ci-après (en Mo) :

   ```
   1, 2, (4), 8, 16, 32, 64, 128, 256, 512, 1024
   ```

   Mo, puis l'exécution de la commande **mkvg –s** OU

   Utilise un facteur (option mkvg –t) permettant de recevoir des multiples de 1016 partitions par disque.

2. Un disque passant outre la limite de 1016 tente de faire partie d'un groupe de volumes pré-existant avec la commande **extendvg**. L'utilisateur peut convertir le groupe de volumes existant pour qu'il contienne des multiples de 1016 partitions par disque à l'aide de l'option **–t** de la commande **chvg**. La valeur de l'option **–t** (facteur) avec la commande **chvg** peut être choisi de telle manière que le nouveau disque s'adapte à la nouvelle limite de (1016 * facteur). Toutefois, une fois que le volume est converti, il ne pas être réimporté dans la version AIX 4.3.0 ou inférieure. L'utilisateur peut aussi recréer le groupe de volumes avec une taille de partition supérieure permettant l'exploitation du nouveau disque, ou créer un groupe de volumes autonome avec une taille de disque physique plus importante.

   Si le code d'installation détecte une taille de l'unité **rootvg** supérieure à 4 Go, il modifie la valeur de **mkvg –s** jusqu'à pouvoir mapper la capacité totale du disque sur les 1016 pistes disponibles. En outre, cette modification va se répercuter sur la taille des partitions physiques de tout autre disque ajouté à **rootvg**, quelle que soit sa taille.

   Pour les systèmes RAID, le nom **/dev/hdiskX** utilisé dans AIX par LVM peut effectivement représenter un grand nombre de disques avec une capacité différente de 4 Go. Dans ce cas, la limite de 1016 existe toujours. LVM ne connait pas la taille de chaque disque constituant réellement **/dev/hdiskX**. La limite de 1016 est basée par LVM sur la taille de **/dev/hdiskX** telle reconnue par AIX et non sur les disques physiques qui constituent réellement **/dev/hdiskX**.

Limites :

Régulations VGSA 1016	VGSA 1016 permet de connaître l'ancienneté des miroirs. Cette ancienneté indique qu'une copie des données ne ressemble pas aux deux autres. Si vous ne respectez pas la régulation 1016, vous risquez d'obtenir un rapport faussé d'un ancien volume logique non mis en miroir, ou l'indication erronée qu'une des copies miroir est ancienne. En outre, la commande **migratepv** risque d'échouer car elle utilise la mise en miroir pour déplacer un volume logique d'un disque à l'autre. Si la partition logique cible est considérée à tort comme ancienne, **migratepv** ne peut pas supprimer la partition logique source et la commande échoue au cours de la migration. La commande **reorgvg** utilise également la mise en miroir temporaire.
Mise en miroir ou **migratepv**	Si vous n'utilisez pas la mise en miroir ni la commande **migratepv**, vos données sont justes jusqu'à la veille du jour où vous avez repéré les dépassements 1016. Elles peuvent être perdues si vous mettez un volume logique en miroir et que : – Toutes les copies sont erronées en même temps, – LVM n'en est pas averti car les copies erronées ont dépassé la limite de recherche 1016. Dans ce cas, et même si vous n'avez pas dépassé la limite 1016, vous perdrez surement les données. Cependant, si vous n'utiisez pas la mise en miroir ou la commande **migratepv**, la question ne se pose pas.
Déplacement d'un groupe de volumes	Un groupe de volumes peut être déplacé d'un système et d'une version AIX à l'autre. La limite 1016 n'est applicable que pendant l'exécution de **mkvg** et **extendvg**. Les données sont en sécurité sur toutes les versions d'AIX.
Modification de la taille limite de la partition physique	Aucune modification ne doit être apportée à la taille limite de la partition physique. Cependant, la possibilité de modifier cette valeur permet d'assurer que, quelle que soit la taille de l'unité de disque, vous pourrez amener l'unité en dessous de la limite 1016. Si une modification intervient dans cette limite, une incompatibilité se produira au niveau de LVM.
Reconstruction du groupe de volumes	Les versions ultérieures d'AIX citées dans ce document préviennent la création de disques dans des groupes de volumes qui dépassent la limite 1016. Ceux qui dépassent déjà cette limite doivent être recréés avec une partion physique plus importante.

- Dans certains cas, l'utilisateur aura des difficultés à ajouter un nouveau disque à un groupe de volumes existant ou à créer un groupe de volumes. LVM affichera l'avertissement suivant :

  ```
  L'espace zone descripteur restant dans ce groupe de volumes est
  insuffisant.
  Essayez d'ajouter un PV de plus petite taille ou utilisez un
  autre groupe de volumes.
  ```

  Dans un groupe de volumes, chaque disque possède une zone VGDA (descripteur de groupe de volumes). Cette zone permet l'importation du groupe de volumes dans un autre système AIX (avec la commande **importvg**). La zone VGDA contient les noms des disques constituant le groupe, leur taille physique, le mappage des partitions, les volumes logiques du groupe et d'autres informations utiles au traitement par LVM.

A la création d'un groupe de volumes, la commande **mkvg** applique la configuration par défaut pour permettre au nouveau groupe de disposer de 32 disques maximum. Toutefois, la tendance étant aux disques de grande capacité, cette limite n'est pas souvent atteinte, l'espace étant plus rapidement saturé dans la zone VGDA qui représente la capacité des plus grands disques. Sous LVM, la zone VGDA est dotée d'un espace maximal fixe, pour 32 disques. La gestion des disques de grande capacité requiert un plus grand espace de mappage dans la zone VGDA, qui réduit le nombre et la taille des disques disponibles à ajouter au groupe de volumes existant. Quand un disque est ajouté à un groupe de volumes, une copie à jour de la zone VGDA lui est intégrée, mais cette zone actualisée doit être acceptée par tous les autres disques du groupe de volumes.

rootvg constitue une exception, n'étant pas concerné par la description de la zone VGDA maximale. Pour offrir plus d'espace disque aux utilisateurs d'AIX, à la création de **rootvg**, la commande **mkvg** n'utilise pas la limite des 32 disques autorisés dans un groupe de volumes. Sous AIX 3.2, le nombre de disques choisi sur le menu d'installation sert de nombre de référence pour **mkvg –d** lors de la création de **rootvg**. Sous AIX4.1, ce nombre **d** est 7 pour un disque, plus un par disque supplémentaire. Par exemple, pour deux disques, ce nombre sera de 8 et pour trois disques, il sera de 9. Cette limite n'empêche pas l'utilisateur d'ajouter des disques supplémentaires dans rootvg après l'installation. Ainsi, l'espace libre d'une zone VGDA, et le nombre et la taille des disques ajoutés à un groupe de volumes dépendent de la taille et du nombre de disques déjà définis pour ce groupe.

Si une zone VGDA plus grande dans **rootvg** est nécessaire pour l'exploitation, reconstruisez et réorganisez **rootvg** avec les commandes **mksysb** et **migratepv** (le seul moyen de modifier la limite **–d** étant de recréer un groupe de volumes).

Remarque : Il est conseillé d'éviter de stocker les données utilisateur sur les disques **rootvg**. Cette séparation confère au système un degré d'intégrité supérieur.

- Les 512 premiers octets d'un volume logique sont réservés au bloc LVCB. Cette zone contient des informations importantes, telles que la date de création du volume logique, des informations relatives aux copies miroir et les points de montage possibles dans JFS. Pour actualiser le bloc LVCB, certaines commandes LVM intégrées dans les algorithmes de LVM sont nécessaires. L'ancien bloc est lu et analysé pour vérifier sa validité. S'il est valide, il est mis à jour. Sinon, il n'est pas actualisé et l'utilisateur reçoit l'avertissement suivant :

  ```
  Attention : impossible d'écrire des données de bloc de contrôle
  de LV.
  ```

  Dans la plupart des cas, ceci est provoqué par des programmes de bases de données qui accèdent à des volumes logiques bruts (en passant outre JFS) comme support de stockage. Lorsque cela arrive, les informations de ces bases recouvrent littéralement le bloc LVCB. Cette situation peut sembler fatale, mais ce n'est pas le cas. Une fois que le LVCB est écrasé, l'utilisateur peut toujours exécuter les opérations suivantes :

  – extension d'un volume logique,

  – création de copies miroir du volume logique,

  – suppression du volume logique,

– création d'un JFS pour monter le volume logique. L'effacement des blocs LVCB a ses limites. L'importation vers d'autres systèmes AIX de volumes logiques dont les blocs LVCB sont effacés peut être incomplète. Pendant l'exécution de la commande **importvg**, la commande LVM recherche le contenu des blocs LVCB de tous les volumes logiques définis dans un groupe de volumes. Si un bloc est absent, le groupe de volumes importé définit tout de même le volume logique correspondant au système AIX cible et l'utilisateur a toujours accès au volume logique brut. Toutefois, les informations JFS sont perdues et le point de montage associé n'est pas importé. L'utilisateur doit créer d'autres points de montageet la disponibilité des données enregistrées dans le système de fichiers avant l'importation n'est pas garantie. En outre, certaines informations non JFS relatives au volume logique s'affichant normalement avec la commande **lslv** sont introuvables. Le système renseigne alors les informations ODM relatives au volume logique avec les valeurs par défaut. En conséquence, la sortie de **lslv** fournit des informations erronées. Si des copies du volume logique figurent encore sur les disques d'origine, la base de données ODM ne reflètera pas des informations correctes. L'utilisateur doit reconstruire les copies avec les commandes **rmlvcopy** et **mklvcopy**, puis procéder à une synchronisation de l'ODM.

Chapitre 7. Systèmes de fichiers

Ce chapitre, consacré aux systèmes de fichiers, contient des informations sur les répertoires, l'espace disque, le contrôle d'accès, les systèmes de fichiers et les répertoires montés ainsi que sur la restauration de ces systèmes. Les sujets traités sont les suivants :

Systèmes de fichiers - généralités

Un *système de fichiers* est une structure hiérarchique (arborescence de fichiers) de répertoires et de fichiers. Ce type de structure ressemble à un arbre inversé, dont les racines seraient à la cime et les branches au sol. Dans l'arborescence, des répertoires organisent les données et les programmes en groupes, permettant de gérer plusieurs répertoires et fichiers à la fois.

L'exécution de certaines tâches est plus souple au niveau d'un système de fichiers qu'au niveau de chaque répertoire du système. Par exemple, vous pouvez sauvegarder, déplacer ou protéger un système de fichiers complet en une seule opération.

Un système de fichiers réside sur un seul volume logique. La commande **mkfs** ou la commande **smit** correspondante crée un système de fichiers sur un volume logique. A l'intérieur d'un volume logique, tous les répertoires et les fichiers appartiennent à un système de fichiers.

Pour être accessible, un système de fichiers doit être monté sur un point de montage de répertoire. Si plusieurs systèmes de fichiers sont montés, une structure de répertoire est créée, présentant l'image d'un système de fichiers unique. Il s'agit d'une structure hiérarchique avec une racine unique. Cette structure comprend les systèmes de fichiers de base et ceux créés par l'utilisateur.

Vous avez accès aux systèmes de fichiers locaux et distants avec la commande **mount**. Avec cette commande de montage, les systèmes de fichiers sont accessibles en lecture et en écriture à partir de votre système. Généralement, vous devez être membre du groupe système pour monter et démonter les systèmes de fichiers. Le montage d'un système de fichiers peut être automatique, s'il est défini dans le fichier **/etc/filesystems**. Vous pouvez démonter un système de fichiers local ou distant au moyen de la commande **umount**, excepté quand il fait l'objet d'un accès par un utilisateur ou un processus.

Pour plus de détails sur la structure des systèmes de fichiers, reportez-vous à "Description de l'arborescence de fichiers", page 7-5.

Types de systèmes de fichiers

Plusieurs types de systèmes de fichiers sont pris en charge. Ces systèmes sont les suivants :

JFS

Il s'agit d'un type de système de fichiers natif appelé *système de fichiers journalisé* (JFS). JFS prend en charge le jeu complet de la sémantique du système de fichiers. Il fait appel aux techniques de journalisation des bases de données pour la maintenance de sa cohérence structurelle. Ces techniques empêchent l'altération du système de fichiers en cas d'arrêt anormal du système.

Chaque système JFS réside sur un seul volume logique distinct. Le système d'exploitation monte les systèmes JFS au moment de l'initialisation. Ce type de configuration de systèmes de fichiers multiples offre la possibilité d'isoler une partie de l'arborescence pour l'exploiter avec les fonctions de gestion système proposées, telles que la sauvegarde, la restauration et la maintenance.

NFS

Le système NFS (Network File System = *système de fichiers réseau*) est un système de fichiers distribué permettant d'accéder aux fichiers et aux répertoires situés sur des ordinateurs distants et de les exploiter comme des fichiers et aux répertoires locaux. Par exemple, l'utilisateur peut créer, supprimer, écrire, lire et définir des attributs sur des fichiers et des répertoires distants avec des commandes du système d'exploitation.

Système de fichiers CD-ROM

Le *système de fichiers CD-ROM* permet d'accéder au contenu d'un CD-ROM par le biais de l'interface de systèmes de fichiers standard. Il s'agit d'un système de fichiers accessible uniquement en lecture, installé localement sous la couche LFS (systèmes de fichiers logique) d'AIX qui est compatible avec les formats de structure de volumes et de fichiers suivants :

Norme ISO 9660:1988(E) :	Niveau d'échange 1 et niveau d'implémentation 3.
Spécification "High Sierra Group" :	Précède ISO 9660 et permet la compatibilité avec les anciens CD-ROM.
Protocole "Rock Ridge Group" :	Spécifie les extensions d'ISO 9660 en conformité totale avec la norme ISO 9660 et fournit toute la sémantique du système de fichiers POSIX sur la base des protocoles SUSP (System Use Sharing Protocol), et RRIP (Rock Ridge Interchange Protocol) qui permettent le montage et l'accès de CD-ROM comme avec tout autre système de fichiers UNIX.

Format de fichier XA CDROM (uniquement le format de secteur "Mode 2 Form 1")

Spécifie les extensions d'ISO 9660 utilisées dans les applications multimédia sur CD–ROM (par exemple, CD photo).

Ces formats sont uniquement applicables :

- aux volumes uniques
- aux fichiers non imbriqués.

Le système de fichiers CD-ROM dépend du gestionnaire d'unités CD-ROM sous-jacent pour fournir la transparence du format de secteur physique (CD-ROM Mode 1 et XA CD-ROM Mode 2 Form 1) et du format multisession des disques (par le mappage de l'ensemble descripteur de volume à partir de la zone de reconnaissance du volume de la dernière session).

Commandes

Voici les commandes exploitables sur les systèmes de fichiers, quel que soit leur type. Le fichier **/etc/filesystems** gère la liste des systèmes de fichiers auxquels les commandes suivantes sont applicables :

chfs	modifie les caractéristiques d'un système de fichiers.
crfs	ajoute un système de fichiers.
lsfs	affiche les caractéristiques d'un système de fichiers.
rmfs	supprime un système de fichiers.
mount	monte un système de fichiers pour le mettre à disposition de l'utilisateur.

Les quatre commandes ci-après sont dédiées aux systèmes de fichiers virtuels. Le fichier **/etc/vfs** contient les informations relatives aux types de systèmes de fichiers auxquels ces commandes sont applicables.

chvfs	modifie les caractéristiques d'un type de système de fichiers.
crvfs	ajoute un nouveau type de système de fichiers.
lsvfs	affiche les caractéristiques d'un type de système de fichiers.
rmvfs	supprime un type de système de fichiers.

Gestion des systèmes de fichiers

Un système de fichiers est une structure de répertoires complète, formée d'un répertoire racine comprenant tous les répertoires et les fichiers du système. Les systèmes de fichiers sont regroupés sur un seul volume logique. Les fonctions de gestion système essentielles applicables aux systèmes de fichiers sont les suivantes :

- affectation d'espace sur les volumes logiques,
- création de systèmes de fichiers,
- mise à la disposition des utilisateurs système d'espace disponible,
- gestion de l'exploitation de l'espace,
- sauvegarde pour prévenir la perte de données en cas de défaillance du système,
- maintien de la cohérence.

Voici une liste des commandes d'exploitation des systèmes de fichiers les plus couramment utilisées :

backup	sauvegarde complète ou incrémentée d'un système de fichiers.
dd	copie directe des données d'une unité dans une autre pour constituer des sauvegardes des systèmes de fichiers.
df	indication de l'espace occupé et disponible sur un système de fichiers.
fsck	contrôle des systèmes de fichiers et correction des incohérences.
mkfs	création d'un système de fichiers de la taille voulue sur un volume logique spécifié.
mount	montage d'un système de fichiers sur une structure d'appellation à l'échelle du système pour rendre les répertoires et les fichiers de ce système accessibles.
restore	restauration de fichiers à partir d'une sauvegarde.
umount	démontage d'un système de fichiers d'une structure d'appellation à l'échelle du système pour rendre les répertoires et fichiers de ce système inaccessibles.

Description de l’arborescence de fichiers

L’arborescence de fichiers AIX permet de regrouper dans un même répertoire des fichiers contenant des données similaires. Cette organisation simplifie le montage à distance des répertoires et des fichiers. Les administrateurs système peuvent exploiter ces répertoires pour construire une arborescence de fichiers unique à l’usage de tout client montant des répertoires à partir d’un ou de plusieurs serveurs. Comparé à l’accès aux données stockées localement, l’accès à distance offre les avantages suivants :

- préservation de l’espace disque local,
- gestion système simplifiée et centralisée,
- meilleure protection de l’environnement.

L’arborescence de fichiers AIX présente les caractéristiques suivantes :

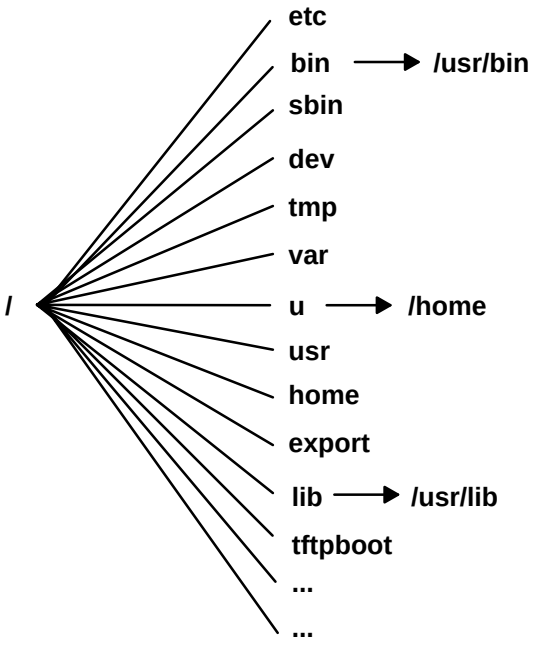
- Les fichiers partageables par des machines de même architecture matérielle résident dans le système de fichiers **/usr**.
- Les fichiers de variables client, tels que ”spool” et ”mail”, résident dans le système de fichiers **/var**.
- Les fichiers texte partageables, indépendants de l’architecture (texte du manuel, par exemple) figurent dans le répertoire **/usr/share**.
- Le système de fichiers racine (**/**) contient les répertoires et les fichiers essentiels à l’exploitation du système : notamment, un répertoire d’unités, les programmes d’amorçage du système, les points de montage des systèmes de fichiers à monter sur le système de fichiers racine.
- Le système de fichiers **/home** constitue le point de montage des répertoires utilisateur personnels.
- Il contient les fichiers de pagination, les systèmes de fichiers racine non partagés, les répertoires de cliché, personnels et **/usr/share** des clients sans disque, et les répertoires **/usr** exportés.

Pour des informations sur un système de fichiers ou un répertoire spécifique, reportez-vous à :

- Description du système de fichiers racine, page 7-6,
- Description du système de fichiers /usr, page 7-8,
- Description du répertoire /usr/share, page 7-10,
- Description du système de fichiers /var, page 7-11,
- Description du répertoire /export, page 7-12.

Description du système de fichiers racine

Le schéma du système de fichiers racine présente la plupart des sous-répertoires de ce système.



Système de fichiers racine

Le système de fichiers racine est situé au sommet de l'arborescence de fichiers. Il comprend les fichiers et les répertoires indispensables à l'exploitation du système, y compris le répertoire des unités et les programmes d'amorçage du système. Il contient également les points de montage dédiés à la connexion des systèmes de fichiers avec la hiérarchie du système de fichiers racine.

La liste ci-dessous indique le contenu et la description de quelques sous-répertoire du système de fichiers racine (/).

/etc contient des fichiers de configuration spécifiques à la machine. Par exemple :

- **/etc/hosts**
- **/etc/passwd**

Le répertoire **/etc** contient les fichiers dédiés à l'administration du système. La plupart des commandes qui résidaient dans le répertoire **/etc** ont été transférées dans le répertoire **/usr/sbin**. La compatibilité est assurée par des liens symboliques renvoyant aux emplacements de certains fichiers exécutables. Par exemple :

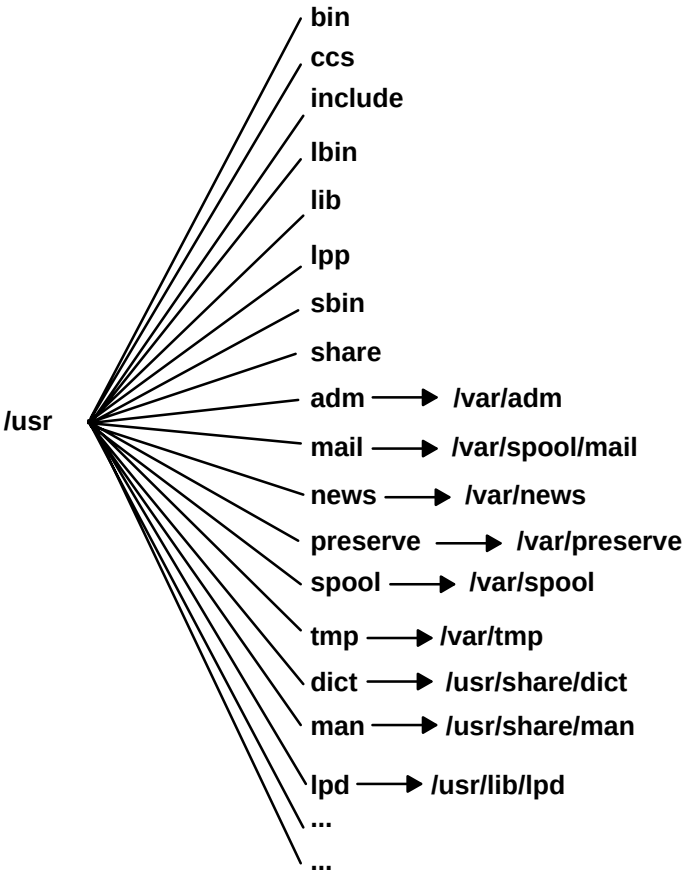
- **/etc/chown** est un lien symbolique associé à **/usr/bin/chown**.
- **/etc/exportvg** est un lien symbolique associé à **/usr/sbin/exportvg**.

/bin lien symbolique renvoyant au répertoire **/usr/bin**. Dans les anciens systèmes de fichiers UNIX, le répertoire **/bin** contenait des commandes utilisateur, qui figurent à présent dans le répertoire **/usr/bin**.

/sbin	contient les fichiers d'armorçage du système et de montage du système de fichiers **/usr**. La plupart des commandes d'amorçage sont issues du système de fichiers disque RAM de l'image d'amorçage ; il y a donc très peu de commandes dans le répertoire **/sbin**.
/dev	contient les noeuds des unités pour les fichiers spéciaux des unités locales. Le répertoire **/dev** contient les fichiers spéciaux des unités de bande, des imprimantes, des partitions de disque et des terminaux.
/tmp	sert de point de montage aux systèmes de fichiers contenant des fichiers temporaires générés par le système. Le système de fichiers **/tmp** est un répertoire vide.
/var	sert de point de montage aux fichiers variant d'une machine à l'autre. Le système de fichiers **/var** est considéré comme un système de fichiers car la taille des fichiers qu'il contient est généralement croissante. Reportez-vous à "Description du système de fichiers /var", page 7-11 pour plus d'informations.
/u	lien symbolique renvoyant au répertoire **/home**.
/usr	contient des fichiers fixes et partageables (fichiers exécutables et documentation ASCII). Les machines autonomes montent la racine d'un système de fichiers local sur le répertoire **/usr**. Les machines sans disque ou à faible capacité disque montent le répertoire d'un serveur à distance également sur le système de fichiers **/usr**. Reportez-vous à "Description du système de fichiers /usr", page 7-8 pour plus de détails sur l'arborescence de fichiers montée sur le répertoire **/usr**.
/home	sert de point de montage au système de fichiers où résident les répertoires utilisateur personnels. Le système de fichiers **/home** contient les répertoires et les fichiers utilisateur Sur une machine autonome, le répertoire **/home** réside dans un système de fichiers séparé dont la racine est montée sur le répertoire **/home** du système de fichiers racine. Dans le cas d'un réseau, un serveur peut contenir des fichiers utilisateur accessibles par plusieurs machines. Dans ce cas, la copie du répertoire **/home** du serveur est montée à distance sur un système de fichiers **/home** local.
/export	contient les répertoires et fichiers résidant sur un serveur et dédiés à des clients distants. Reportez-vous à "Description du répertoire /export", page 7-12 pour plus de détails sur l'arborescence de fichiers résidant sur **/export**.
/lib	lien symbolique renvoyant au répertoire **/usr/lib**. Reportez-vous à "Description du système de fichiers /usr", page 7-8 pour plus d'informations.
/tftpboot	contient des images et des données d'amorçage destinées aux clients sans disque.

Description du système de fichiers /usr

Le système de fichiers **/usr** contient des fichiers exécutables et partageables. Le diagramme suivant présente les principaux sous-répertoires de ce système de fichiers.



Système de fichiers /usr

Sur une machine autonome, le système de fichiers **/usr** est séparé (résidant sur le volume logique **/dev/hd2**). Sur une machine sans disque ou de faible capacité disque, le répertoire (autorisé uniquement en lecture) d'un serveur distant est monté sur le système de fichiers **/usr** local. **/usr** contient des données, des bibliothèques et des commandes accessibles en lecture seulement.

Les répertoires et les fichiers de **/usr**, à l'exception du répertoire **/usr/share**, peuvent être partagés par toutes les machines dépendant de la même architecture matérielle.

Le système de fichiers **/usr** comprend les répertoires suivants :

/usr/bin	contient les commandes ordinaires et les scripts shell. Par exemple, les commandes **ls**, **cat** et **mkdir**.
/usr/ccs	contient des modules de développement binaires facturés séparément.
/usr/include	contient des fichiers include ou d'en-tête.
/usr/lbin	contient des fichiers exécutables expéditeurs de commandes.
/usr/lib	contient des biliothèques indépendantes de l'architecture dont les noms sont au format **lib*.a**. le répertoire **/lib** situé sur la racine (**/**) est un lien symbolique qui renvoie au répertoire **/usr/lib** ; ce lien a permis le transfert de tous les fichiers du répertoire **/lib** dans le répertoire **/usr/lib**. En outre, ce répertoire contient des fichiers non bibliothèque assurant la compatibilité.

/usr/lpp	contient des modules installés en option.
/usr/sbin	contient des utilitaires dédiés à l'administration du système, y compris les commandes SMIT. La plupart des commandes qui résidaient dans le répertoire **/etc** ont été transférées dans le répertoire **/usr/sbin**.
/usr/share	contient des fichiers partageables entre machines d'architecture différente. Reportez-vous à "Description du système de fichiers /usr/share", page 7-10 pour plus d'informations.

Liens symboliques renvoyant au répertoire /var

/usr/adm	lien symbolique renvoyant au répertoire **/var/adm**.
/usr/mail	lien symbolique renvoyant au répertoire **/var/spool/mail**.
/usr/news	lien symbolique renvoyant au répertoire **/var/news**.
/usr/preserve	lien symbolique renvoyant au répertoire **/var/preserve**.
/usr/spool	lien symbolique renvoyant au répertoire **/var/spool**.
/usr/tmp	lien symbolique renvoyant au répertoire **/var/tmp**, **/usr** étant partageable entre plusieurs noeuds et accessible en lecture seulement.

Liens symboliques renvoyant aux répertoires /usr/shre et /usr/lib

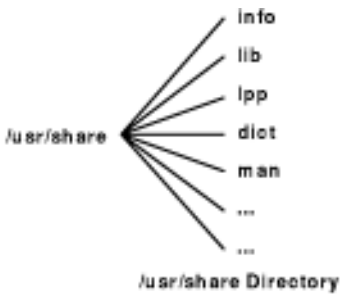
/usr/dict	lien symbolique renvoyant au répertoire **/usr/share/dict**.
/usr/man	lien symbolique renvoyant au répertoire **/usr/share/man**.
/usr/lpp	lien symbolique renvoyant au répertoire **/usr/lib/lpd**.

Description du répertoire /usr/share

Le répertoire **/sur/share** contient des fichiers texte partageables et indépendants de l'architecture, c'est-à-dire pouvant être partagés entre toutes les machines, quelle que soit l'architecture matérielle.

Dans un environnement d'architecture mixte, le client sans disque monte le répertoire d'un serveur sur sont propre répertoire **/usr**, puis un autre répertoire sur le répertoire **/usr/share**. Les fichiers de **/usr/share** résident sur un ou plusieurs modules installables séparément. Un noeud peut donc faire installer localement les autres parties du répertoire **/usr** dont il dépend, alors qu'il utilise un serveur pour fournir **/usr/share**.

Certains fichiers de **/usr/share** contiennent les répertoires et les fichiers présentés dans le diagramme du répertoire /usr/share suivant.



/usr/share Directory

Le répertoire **/usr/share** comprend :

/usr/share/man	contient le texte du manuel (s'il a été chargé).
/usr/share/dict	contient le dictionnaire orthographique et les index correspondants.
/usr/share/info	contient les fichiers de la base de données InfoExplorer.
/usr/share/lib	contient les fichiers de données indépendants de l'architecture, y compris **terminfo**, **learn**, **tmac**, **me** et **macros**.
/usr/share/lpp	contient des données et des informations relatives aux modules en option installables sur le système.

Description du système de fichiers /var

Avertissement : La taille du système de fichiers **/var** est croissante du fait que tous les sous-répertoires et les fichiers de données qui y résident sont exploités par des applications très sollicitées, telles que la comptabilité, le courrier et le spouleur d'impression. Si vos applications utilisent **/var** intensivement, développez ce système de fichiers en lui affectant une valeur supérieure à celle de sa taille par défaut (4 Mo).

Les fichiers **/var** assurant des contrôles périodiques sont **/var/adm/wtmp** et **/var/adm/ras/errlog**.

Le suivi est assuré par les fichiers **/var** suivants :

/var/adm/ras/trcfile	quand l'utilitaire de suivi est en fonction.
/var/tmp/snmpd.log	si la commande **snmpd** est lancée sur le système.

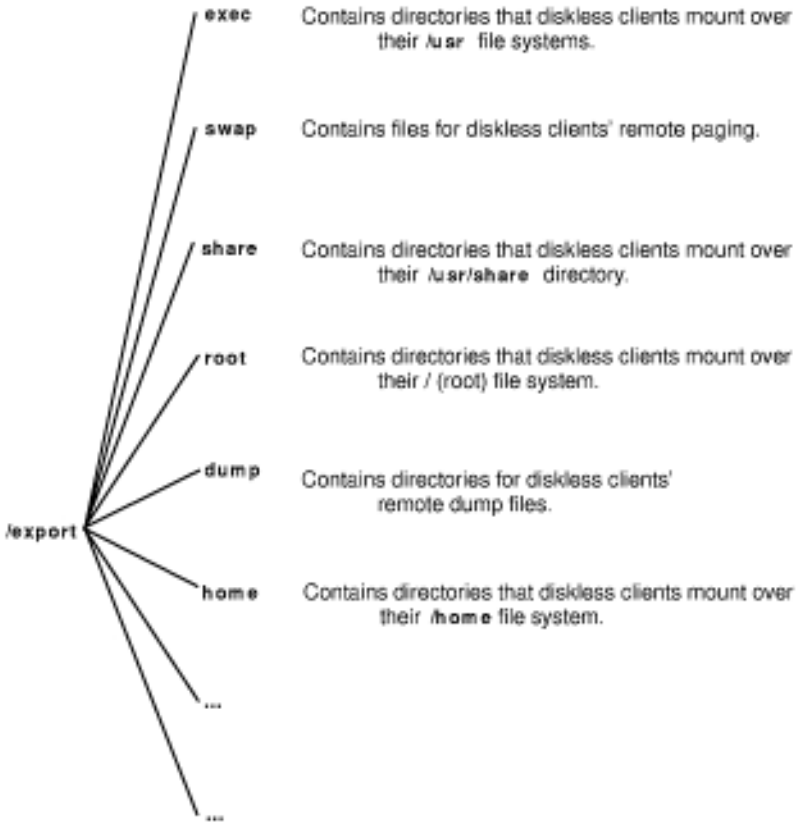
Le diagramme ci-dessous répertorie certains répertoires de **/var**.

/var Directory

/var/adm	contient les fichiers de journalisation et de comptabilité du système.
/var/news	contient le bulletin d'informations sur le système.
/var/preserve	contient des données préservées lors des interruptions de sessions d'édition ; correspond au répertoire **/usr/preserve** des versions précédentes.
/var/spool	contient les fichiers en cours de traitement (par le courrier électrique, par exemple) ; correspond au répertoire **/usr/spool** des versions précédentes.
/var/tmp	contient des fichiers temporaires : correspond au répertoire **/usr/tmp** des versions précédentes. Le répertoire **/usr/tmp** est le lien symbolique renvoyant à **/var/temp**.

Description du répertoire /export

Le répertoire **/export** contient des fichiers exportés du serveur vers les clients, c'est-à-dire sur des machines sans disque, sans données ou de faible capacité disque. Un serveur peut exporter différents types d'espace disque, y compris des modules de programmes exécutables, des espaces de pagination pour clients sans disque et des systèmes de fichiers racine pour clients sans disque ou ne disposant que d'une faible capacité disque. Dans l'arborescence de fichiers, cet espace disque réside généralement sous le répertoire **/export**. Le diagramme suivant présente quelques sous-répertoires du répertoire **/export** et leur contenu.



/export File System

/export est l'emplacement par défaut des ressources client sans disque (commandes).

/export représente simplement un emplacement réservé aux ressources client sur le serveur. Pour les clients, ces ressources sont situées à l'emplacement normal dans l'arborescence de fichiers, ceux-ci les montant sur leurs propres arborescences de fichiers. Voici les principaux sous-répertoires de **/export** et les points de montage correspondants sur une arborescence de fichiers client :

répertoire **/export/root**
: monté sur le système de fichiers racine (/) client. Par défaut, les répertoires racine du client sont situés dans **/export/root** et portent le nom hôte du client.

répertoire **/export/exec**, ou SPOT (Shared Product Object Tree)
: monté sur le système de fichiers **/usr** du client. Un SPOT est une version du système de fichiers **/usr** stockée dans **/export/exec** sous le nom correspondant au niveau de version. Par défaut, son nom est **RISCAIX**.

répertoire **/export/share** (ou Share)	monté sur le système de fichiers **/usr/share** du client. Ce répertoire contient des données partageables entre un grand nombre d'architectures. Par défaut, il est implanté dans **/export/share/AIX/usr/share**.
répertoire **/export/home**	monté sur le système de fichiers **/home** du client. Ce répertoire contient des répertoires utilisateur groupés par les noms hôte du client. Par défaut, les répertoires personnels client sont situés dans **/export/home**.
répertoire **/export/swap** (ou Paging)	pour un système autonome ou sans données, la pagination est fournie par un disque local et, pour un client sans disque, par un fichier résidant sur un serveur. Ce fichier porte le nom hôte du client et est situé, par défaut, dans **/export/swap**.
répertoire **/export/dump**	pour un système autonome, l'unité de cliché est fournie par un disque local et, pour un client sans disque, par un fichier résidant sur un serveur. Ce fichier porte le nom hôte du client et est situé, par défaut, dans **/export/swap**.
répertoire **microcode**	Ce répertoire contient le microcode réservé aux unités physiques. Par défaut, il réside dans **/export/exec/RISCAIX/usr/lib/microcode**.

Description de la compression de données

JFS prend en charge les systèmes de fichiers fragmentés et compressés. Ces deux types de systèmes permettent d'économiser l'espace disque, les blocs logiques étant stockés sur le disque en unités ou "fragments" plus petits que la taille habituelle de 4096 octets. Dans un système fragmenté, seul le dernier bloc logique de fichiers d'une capacité inférieure ou égale à 32 Ko est stocké de cette manière ; ainsi, seuls les systèmes de fichiers contenant un grand nombre de petits fichiers bénéficient des avantages de la fragmentation. Avec la compression de données, tous les blocs logiques, quelle que soit la taille des fichiers, peuvent être stockés en un ou plusieurs fragments contigus. La compression des données permet de diviser par deux l'occupation de l'espace disque.

L'exploitation des fragments et de la compression des données augmente donc les possibilités de fragmenter l'espace disque libre. Les fragments affectés à un bloc logique doivent être contigus sur le disque. Quand vous fragmentez l'espace libre d'un système de fichiers, il n'est pas toujours aisé de trouver suffisamment de fragments contigus pour l'affectation d'un bloc logique, même si vous avez besoin d'un nombre inférieur au nombre de fragments libres. JFS peut y remédier avec l'utilitaire **defragfs**, qui "défragmente" un système de fichiers en augmentant l'espace libre contigu. Vous pouvez exploiter cet utilitaire pour les systèmes de fichiers fragmentés et compressés. Les techniques de fragmentation et de compression de données permettent de libérer des espaces importants sur les disques ; quant à la fragmentation de l'espace libre, les difficultés à la mettre en œuvre sont tout à fait gérables.

Avec la version actuelle de JFS, la compression des données et les versions antérieures d'AIX sont compatibles. L'interface API (Application Programming Interface), formée de tous les appels système, est identique dans les deux versions de JFS.

Pour plus de détails (notamment sur le coût des performances), reportez-vous à "Description des fragments et du nombre variable d'i-nodes", page 7-17.

Configuration

Attention : Le système de fichiers racine (***/***) ne peut pas être compressé.

Attention : Il n'est pas conseillé de compresser le système de fichiers **/usr** car **installp** doit pouvoir calculer précisément les mises à jour et les nouvelles installations. Reportez-vous à la section Comportement implicite ci-après pour avoir plus d'informations sur la taille et les calculs.

Cet attribut est défini lors de la création du système de fichiers avec la commande **crfs** ou **mkfs**. La compression est uniquement applicable aux fichiers standard et aux longs liens symboliques longs figurant dans des systèmes de fichiers standard. Les fragments continuent d'être applicables aux répertoires et aux métadonnées non compressées. Tous les blocs logiques d'un fichier sont autocompressés avant d'être inscrits sur le disque. De cette façon, les recherches et les mises à jour aléatoires sont simplifiées et la perte en espace libéré est faible par rapport à celle générée par la compression en unités plus grandes.

Une fois compressé, un bloc logique requiert généralement un espace disque inférieur à 4096 octets. Il ne lui est affecté que le nombre de fragments contigus nécessaires à son stockage. S'il n'est pas compressé, il est inscrit sur le disque sous forme décompressée et 4096 octets de fragments contigus lui sont affectés.

Comportement implicite

Un programme qui écrit un fichier ne s'attendant pas à une condition ENOSPC (espace saturé) une fois l'écriture aboutie (ou le stockage, en ce qui concerne les fichiers mappés), il et indispensable de garantir la disponibilité de l'espace requis pour l'écriture des blocs logiques sur disque. C'est pourquoi 4096 octets sont affectés au bloc logique lorsqu'il est modifié pour la première fois : il disposera ainsi d'espace disque même s'il n'est pas compressé. Si cette quantité d'octets n'est pas disponible, le système renvoie une condition d'erreur ENOSPC ou EDQUOT. Si cette quantité d'octets n'est pas disponible, le système renvoie une condition d'erreur ENOSPC ou EDQUOT ; ceci peut se produire même si l'espace disque est suffisant pour loger le bloc logique compressé. L'espace disque saturé est signalé prématurément lorsque les limites de quota disque sont presque atteintes ou que le système de fichiers est presque saturé.

Voici certains aspects du comportement des systèmes de fichiers compressés :

- En raison des 4096 octets affectés au bloc logique, une erreur ENOSPC ou EDQUOT peut être retournée à certains appels système. Par exemple, un ancien fichier peut être mappé avec la commande d'appel système **mmap**, et le stockage à l'emplacement d'une opération d'écriture antérieure peut être à l'origine d'une erreur ENOSPC. L'appel système **ftruncate** peut également provoquer une erreur ENOSPC ou EDQUOT lorsque le bloc logique n'a pas été tronqué en raison d'une limite de bloc.
- Avec la compression de données, un bloc disque intégral reste affecté à un bloc modifié jusqu'à son écriture sur le disque. Si le bloc avait une affectation, validée antérieurement, d'une taille inférieure à celle du bloc intégral, la taille de l'espace disque liée au bloc est égale à la somme des deux, l'affectation antérieure n'étant pas libérée tant que le fichier (i-node) est engagé. Ceci est le cas des fragments standard. Le nombre de blocs logiques d'un fichier pouvant faire l'objet d'affectations validées antérieurement est de 1 maximum, mais peut être égal au nombre de blocs dans un fichier compressé.
- Aucune ressource validée antérieurement pour un bloc logique n'est libérée tant qu'un appel système **fsync** ou **sync** est lancé par le programme d'application.
- L'appel système **stat** signale le nombre de fragments affectés à un fichier. Ce nombre basé sur les 4096 octets affectés à des blocs modifiés, mais non écrits, et la taille compressée des blocs non modifiés. Les ressources validées antérieurement ne sont pas comptées par **stat**. **stat** indiquerait un nombre correct de fragments affectés après une validation d'i-node si aucun des blocs modifiés n'était compressé. De même, les quotas de disque sont affectés à l'emplacement en cours. Comme les blocs logiques d'un fichier sont inscrits sur le disque, le nombre de fragments qui leur sont affectés diminue s'ils sont compressés, ce qui modifie les quotas de disque et le résultat de **stat**.

Commandes de compression

Les commandes **crfs**, **mkfs** et **lsfs** ont été étendues pour la compression des données. Ces commandes, ainsi que l'outil SMIT, offrent des options permettant de définir et d'identifier la compression des données.

Identification de la compression

L'option **–q** de la commande **lsfs** affiche la valeur de compression en cours.

Compatibilité et migration

Les versions antérieures d'AIX et la version actuelle de JFS sont compatibles. La compatibilité des images disque avec ces versions antérieures est maintenue ; ainsi, le montage des systèmes de fichiers et leur accès est possible sans migration de disque ni impact sur les performances.

Sauvegarde/restauration

Les séquences de sauvegarde/restauration d'un système de fichiers compressé vers un système de fichiers non compressé ou entre systèmes de fichiers compressés de tailles de fragment différentes sont possibles, l'exploitation des disques étant améliorée par la compression des systèmes de fichiers. Toutefois, si l'espace disque est saturé, les opérations de restauration ne peuvent pas aboutir. Ce phénomène concerne la séquence de sauvegarde/restauration d'un système de fichiers intégral ; il est susceptible de se produire même si le système de fichiers cible est plus grand que le système de fichiers source.

Algorithme de compression

L'algorithme de compression est une version IBM de LZ. En général, ces algorithmes compressent les données en représentant la deuxième occurrence et les suivantes d'une chaîne données avec un pointeur identifiant la position de la première occurrence de la chaîne et sa longueur. Au début du processus, aucune chaîne n'a été identifiée, aussi le premier octet de données au moins doit-il être représenté comme un caractère "brut" qui requiert 9 bits (0,octet). Une fois qu'un certain nombre d'octets sont compressés (*N* octets), le compresseur recherche dans *N* octets la plus longue chaîne qui correspond à la chaîne commençant au niveau de l'octet non traité suivant. Si la longueur de la chaîne trouvée est 0 ou 1, l'octet suivant est codé comme un caractère brut. Sinon, la chaîne est représentée comme une paire (pointeur, longueur) et le nombre d'octets traités est incrémenté de la longueur. D'un point de vue architectural, LZ admet 512, 1024 ou 2048 comme valeur de *N* et définit le codage des paires et des caractères bruts. Le pointeur est une zone de taille fixe de log2 *N*, tandis que la longueur est codée comme une zone de taille variable.

Coût des performances

La compression des données découlant de la prise en charge des fragments, le coût de performance associé aux fragments, joue également sur la compression. Les systèmes de fichiers compressés affectent les performances, comme expliqué ci-après :

- La durée de compression/décompression étant assez longue, l'exploitation d'un système de fichiers compressé peut être limitée dans certains environnements utilisateur.
- La plupart des fichiers UNIX standard ne sont écrits qu'une fois, mais certains sont mis à jour. Pour les fichiers modifiés, la compression est grevée, en termes de coûts de performances, de l'obligation d'affecter 4096 octets d'espace disque à la première modification d'un bloc logique, puis de réaffecter l'espace disque une fois le bloc logique inscrit sur le disque. Cette affectation supplémentaire n'est pas nécessaire pour les fichiers standard d'un système de fichiers non compressé.
- La compression augmente le nombre de cycles processeur. Il faut en moyenne 50 cycles par octet pour la compression, et 10 par octet pour la décompression.

Description des fragments et du nombre variable d'i-nodes

Il est possible de diviser l'espace disque en unités d'affectation, ou fragments, d'une capacité inférieure à celle par défaut (4096 octets). Ce procédé de stockage dans des blocs logiques partiels permet une économie d'espace disque. Le comportement fonctionnel de la fonction de fragment par JFS est basé sur celui offert par Berkeley Software Distribution (BSD). Les deux concepts permettent aux utilisateurs de définir le nombre d'i-nodes d'un système de fichiers.

Exploitation du disque

De nombreux systèmes de fichiers UNIX n'affectent de l'espace disque contigu qu'en unités de taille égale aux blocs logiques utilisés pour la division logique des fichiers et des répertoires. Ces unités s'appellent des "blocs disque" ; un bloc disque unique sert exclusivement à stocker les données d'un bloc logique unique d'un fichier ou d'un répertoire.

Choisir une taille de bloc relativement grande (par exemple, 4096 octets) et affecter des blocs disque de taille égale à celle des blocs logiques offre l'avantage de réduire le nombre d'E/S disque à traiter, les données étant stockées sur disque dans un petit nombre de grands blocs disque plutôt que l'inverse. Par exemple, un bloc disque de 4096 octets est affecté à un fichier de 4096 octets (ou moins) si la taille du bloc logique est identique. Dans ce cas, une opération de lecture-écriture ne traite qu'une seule E/S disque pour l'accès aux données du disque. Avec une taille de bloc logique inférieure requérant plus d'une affectation pour la même quantité de données, plus d'une opération d'E/S peut être nécessaire pour l'accès aux données. Un bloc logique de grande taille et un bloc disque de même taille offrent l'avantage de simplifier l'opération d'affectation d'espace disque pour ajouter des données dans les fichiers et les répertoires, les blocs disque de grande taille contenant plus de données.

Toutefois, limiter l'unité d'affectation d'espace disque à la taille du bloc logique ne permet pas d'économiser l'espace disque dans un système de fichiers contenant de nombreux fichiers et répertoires de petite taille. L'espace disque est gaspillé quand un bloc logique représentant de l'espace disque est affecté à un bloc logique partiel de fichier ou de répertoire. La contenance des blocs logiques partiels étant toujours moindre que celle des blocs logiques contenant des données, un bloc logique partiel n'utilise pas totalement l'espace disque qui lui est affecté. L'autre portion reste inexploitée, aucun autre fichier ou répertoire ne pouvant écrire sur un espace disque déjà affecté. Pour un système de fichiers contenant un grand nombre de petits fichiers et répertoire, l'espace disque total inexploité peut s'avérer important. Un système de fichiers avec des unités d'affectation de 4096 octets peut ainsi avoir 45 % de son espace disque inexploité (source de ces statistiques : *UNIX System Manager's Manual*, Computer Systems Research Group, University of California - Berkeley, The Regents of the University of California and/or Bell Telephone Laboratories, 1988, SMM 14.)

Optimisation des disques

Dans JFS, l'unité d'affectation d'espace disque, ou *fragment*, peut être plus petite qu'un bloc logique de 4096 octets. Cela permet d'optimiser le stockage des données d'un bloc logique partiel en utilisant uniquement le nombre de fragments nécessaires. Par exemple, pour économiser l'espace disque, un fragment de 512 octets peut être affecté à un bloc logique partiel de 500 octets. Si un bloc logique partiel a besoin de plus d'espace de stockage, un ou plusieurs fragments supplémentaires peuvent lui être affectés.

Fragments

La taille de fragment est définie à la création du système de fichiers. Les tailles compatibles avec JFS sont 512, 1024, 2048 et 4096 octets. Pour la cohérence avec la version 3 d'AIX, la taille du fragment par défaut est 4096 octets. Les systèmes de fichiers peuvent avoir des tailles de fragment différentes, mais à l'intérieur d'un même système de fichiers, la taille de fragment doit être unique. Différentes tailles de fragment peuvent aussi coexister sur une même machine, ce qui permet de choisir la taille convenant le mieux à chaque système de fichiers.

La fonction de fragment de JFS donne une vue du système de fichiers sous la forme d'une série contigüe de fragments et non de blocs disque. Toutefois, pour le rendement des opérations de disque, l'espace disque est souvent affecté en unités de 4096 octets, de sorte que les blocs disque ou les unités d'affectation aient la même taille que les blocs logiques. Dans ce cas, l'affectation de bloc disque revient à une affectation de 4096 octets de fragments contigus.

Le temps système opérationnel (recherches supplémentaires sur le disque, transferts de données et opération d'affectation) et l'optimisation de l'espace disque augmente à mesure que la taille de fragment d'un système de fichiers diminue. Pour maintenir un équilibre optimal entre un temps système accru et l'augmentation de l'espace disque exploitable, les facteurs suivants s'appliquent à la fonction de fragment de JFS :

- Autant que possible, les affectations d'espace disque de 4096 octets de fragments sont maintenues pour les blocs logiques d'un fichier ou d'un répertoire.
- Une taille moindre est réservée uniquement aux blocs logiques partiels pour les fichiers ou les répertoires d'une capacité inférieure à 32 Ko.

Comme décrit précédemment (”Exploitation des disques”), avec le maintien de la taille de 4096 octets, le rendement des opérations de disque est meilleur.

Plus la taille des fichiers et répertoires d'un système de fichiers dépasse 32 Ko, plus le bénéfice du maintien des affectations d'espace inférieures à 4096 octets pour les blocs logiques partiels diminue ; l'économie d'espace disque, en terme de pourcentage de l'espace total du système de fichiers, diminue, alors que le coût des performances reste constant. Les affectations d'espace disque inférieures à 4096 octets offrant l'exploitation de l'espace disque la plus efficace avec des fichiers et répertoires de petite taille, 4096 octets de fragments sont toujours affectés aux blocs logiques de fichiers et de répertoires de taille supérieure ou égale à 32 Ko. De même, 4096 octets de fragments sont affectés à tout bloc logique partiel associé à un fichier ou à un répertoire de taille supérieure ou égale à 32 Ko.

Nombre variable d'i-nodes

La fonction de fragmentation optimisant l'exploitation de l'espace disque, elle permet d'augmenter la quantité des petits fichiers et répertoires à stocker dans un système de fichiers. Toutefois, l'espace disque n'est pas l'unique ressource requise par les fichiers et répertoires : chaque fichier ou répertoire a aussi besoin d'un i-node disque. Avec JFS, le nombre d'i-nodes disque créés dans un système de fichiers peut être spécifié, s'il est différent de la valeur par défaut. La quantité voulue est définie à la création du système de fichiers sous la forme d'un nombre d'octets par i-node (NBPI). Par exemple, la valeur 1024 génère la création d'un i-node disque par fraction de 1024 octets de l'espace disque du système de fichiers. Autrement dit, plus la valeur de NBPI est faible (512, par exemple), plus la quantité d'i-nodes est importante, tandis qu'une grande valeur de NBPI (16384, par exemple) génère un petit nombre d'i-nodes.

Les valeurs NBPI autorisées dépendent de la taille du groupe d'affectation (agsize). 4 Mo est la taille définie par défaut. Sous AIX version 4.1, agsize est fixée à 8 Mo. Les valeurs NBPI autorisées sont 512, 1024, 2048, 4096, 8192 et 16384 avec un agize de 8 Mo.

Sous AIX version 4.2 ou ultérieure, vous pouvez opter pour un agsize plus grand. Les valeurs admises sont 8, 16, 32 et 64. Les valeurs NBPI admissibles sont fonction de agsize. Si la taille du groupe d'affectation est doublée pour passer à 16, les valeurs de NBPI doublent également : 1024, 2048, 4096, 8193, 16384 et 32768.

Pour la cohérence avec les versions antérieures d'AIX version 3, la valeur de NBPI par défaut est 4096, et celle du groupe d'affectation (agsize) est 8. Les valeurs de NBPI et de agsize sont précisées au cours de la création du système de fichiers. Si la taille du système de fichiers augmente, alors les valeurs de NBPI et du agsize restent identiques à celles définies lors de la création du système de fichiers.

Définition de la taille de fragment et de la valeur de NBPI

La taille de fragment et la valeur de NBPI sont définies lors de la création du système de fichiers avec les commandes **crfs** et **mkfs**, ou avec SMIT. Le choix doit être opéré en fonction du nombre de fichiers prévu et de leur taille.

Identification de la taille de fragment et de la valeur de NBPI

Pour identifier la taille de fragment du système de fichiers et la valeur de NBPI, utilisez la commande **lsfs** ou SMIT. Pour les programmes d'application, la sous-routine **statfs** permet aussi d'identifier la taille de fragment.

Compatibilité et migration

Toutefois, soyez attentifs si vous effectuez la migration vers une version antérieure de systèmes de fichiers dont la taille de fragment ne correspond ni à la valeur par défaut, ni à la valeur de NBPI, ni à la taille du groupe d'affectation.

Images de système de fichiers

Les images de système de fichiers JFS créées sous une version antérieure d'AIX sont entièrement compatibles avec JFS. Ces images, ainsi que toute image de système de fichiers JFS créée avec les valeurs par défaut de la taille de fragment, de NBPI (4096 octets) et de la taille du groupe d'affectation peuvent être échangées avec les versions antérieures et actuelles d'AIX sans opération de migration particulière.

Les images de système de fichiers JFS créées avec la taille de fragment, la valeur de NBPI et celle de la taille du groupe d'affectation autre que celle par défaut risquent d'être incompatibles avec les versions antérieures d'AIX. Plus précisément, seules les images des systèmes de fichiers dont la taille est inférieure ou égale à 2 Go et créées avec les paramètres par défaut peuvent être échangées entre les versions 3.2, 4.1 et 4.2. Celles créées avec une taille de fragment égale à 512, 1024, 2048 ou 4096, un NBPI de 512, 1024, 2048, 4096, 8192 ou 16384, et une taille de groupe d'affectation de 8 Mo peuvent être échangées entre les versions 4.1 et 4.2 d'AIX. Finalement, créer un système de fichiers avec un NBPI supérieur à 16384 ou un groupe d'affectation supérieur à 8 Mo aboutit à un système de fichiers JFS pris en charge uniquement par AIX version 4.2.

Pour migrer des systèmes de fichiers d'une version AIX vers une autre version d'AIX incompatible, procédez comme suit :

1. Sauvegardez le système de fichiers sous le nom de fichier sur le système source.
2. Créez un système de fichiers avec une taille de fragment et une valeur de NBPI de 4096.
3. Restaurez les fichiers à partir de leur sauvegarde sur le système cible.

Sauvegarde/restauration

Les séquences de sauvegarde/restauration entre systèmes de fichiers de taille de fragment et de valeur NBPI différentes sont possibles, en raison de l'exploitation des disques améliorée et du grand nombre d'i-nodes. Toutefois, le manque de fragments libres ou d'i-nodes dû à une taille de fragment ou une valeur de NBPI du système de fichiers source inférieure à celle du système de fichiers cible, les opérations de restauration ne peuvent pas aboutir. Ce phénomène concerne la séquence de sauvegarde/restauration d'un système de fichiers intégral ; il est susceptible de se produire même si le système de fichiers cible est plus grand que le système de fichiers source.

Limites sur les pilotes d'unités

Un pilote d'unité doit fournir un adressage de blocs au niveau de l'unité identique à la taille de fragment du système de fichiers. Par exemple, si un système de fichiers JFS a été créé sur un pilote d'unité de disque RAM fourni par l'utilisateur, le pilote doit faire en sorte que les blocs de 512 octets puissent contenir un système de fichiers doté de fragments de 512 octets. Si le pilote n'autorisait qu'un adressage au niveau page, seul un JFS avec une taille de fragment de 4096 octets pourrait être exploité.

Remarque : N'importe quelle valeur NBPI admise peut être spécifiée pour une unité.

Coût des performances

Bien que les systèmes de fichiers utilisant des fragments inférieurs à 4096 octets comme unité d'affectation aient besoin de beaucoup moins d'espace que ceux faisant appel à l'unité d'affectation par défaut (de 4096 octets), les fragments de petite taille peuvent grever les coûts de performances.

Opérations d'affectation intensifiées

L'espace disque étant affecté en unités plus petites pour un système de fichiers associé à une taille de fragment différente de 4096 octets, les opérations d'affectations sont plus fréquentes dès lors que les fichiers et les répertoires sont agrandis de façon répétée. Par exemple, une écriture ajoutant 512 octets à un fichier de taille nulle revient à l'affectation d'un fragment de 512 octets au fichier. En cas de nouvelle extension de 512 octets, un fragment supplémentaire est affecté au fichier. Dans ce cas, aucune affectation supplémentaire n'est à effectuer lors de la deuxième opération d'écriture, l'affectation initiale étant suffisante pour l'ajout de données. Si cet exemple était appliqué à un système de fichiers avec des fragments de 4096 octets, l'affectation d'espace disque n'aurait lieu qu'une seule fois, au moment de la première opération d'écriture.
Les opérations d'affectation augmentent le temps système, en terme de performances, pour l'activité des systèmes de fichiers. Ce phénomène peut être réduit pour les systèmes de fichiers dont la taille de fragment est inférieure à 4096 octets, si les fichiers sont augmentés, dans la mesure du possible, par unités de 4096 octets.

Fragmentation de l'espace disponible

Les fragments inférieurs à 4096 octets génèrent une plus grande fragmentation de l'espace disque libre. Prenons pour exemple une zone de disque divisée en 8 fragments de 512 octets. Différents fichiers, requérant chacun 512 octets, ont effectué des écritures dans le premier, le quatrième, le cinquième et le septième fragments de cette zone ; le second, le troisième, le sixième et le huitième fragments sont donc libres et représentent un espace disque total de 2048 octets. Aucun bloc logique partiel requérant 4 fragments (ou 2048 octets) ne sera affecté aux 4 fragments libres, puisque dans une opération unique d'affectation, les fragments doivent être contigus.

Les fragments affectés pour les blocs logiques d'un fichier ou d'un répertoire devant être contigus, la fragmentation de l'espace libre peut faire échouer une demande d'espace disque supplémentaire pour le système de fichiers, même si l'espace libre est suffisant. Par exemple, une écriture destinée à ajouter un bloc logique à un fichier de taille nulle requiert l'affectation de 4096 octets d'espace disque contigu. Si l'espace libre du système de fichiers est constitué de 32 fragments non contigus de 512 octets ou d'un espace disque libre total de 16 Ko, l'opération d'écriture ne peut pas aboutir, puisque 8 fragments contigus ou 4096 octets d'espace disque contigu sont nécessaires.

Si l'espace libre d'un système de fichiers fragmenté n'est pas gérable, il peut être défragmenté au moyen de la commande **defragfs**. L'exécution de cette commande a un impact sur les performances.

Taille de la mappe d'affectation de fragments

Contenir les mappes d'affectation de fragments pour des systèmes de fichiers avec une taille de fragment inférieure à 4096 octets peut requérir plus de mémoire virtuelle et plus d'espace disque. Les fragments servent d'unité de base pour l'affectation d'espace disque ; en outre, l'état d'affectation de chaque fragment d'un système de fichiers est enregistré dans sa mappe d'affectation de fragments.

Description des limites de taille de JFS

La taille maximale de JFS est définie lors de la création du système de fichiers. A la création d'un système JFS, vous devez prendre en compte :

- Nombre d'i-nodes
- Taille du groupe d'affectation
- Capacité d'adressage du fragment du système de fichiers
- Taille du journal JFS
- Taille maximale de JFS

Nombre d'i-nodes

Le nombre total d'i-nodes d'un système de fichiers limite le nombre total de fichiers et la taille globale du système de fichiers. JFS fournit le paramètre nbpi (nombre d'octets par i-node) qui affecte le nombre d'i-nodes d'un système de fichiers. JFS prend en charge les valeurs suivantes de nbpi : `512`, `1024`, `2048`, `4096`, `8192`, `16384`, `32768`, `65536` et `131072`. Les valeurs `32768`, `65536` et `131072` ne concernent qu'AIX version 4.2 (et ultérieures).

Il existe un i-node par octet de nbpi de l'espace du groupe d'affectation affecté au système de fichiers. Un groupe d'affectation peut être partiellement affecté, bien que la totalité des i–nodes par groupe d'affectation soit toujours affectée. NBPI est inversement proportionnel au nombre total d'i-nodes d'un système de fichiers.

JFS limite les systèmes de fichiers à des i-nodes de 16 Mo (2^{24}).

Taille du groupe d'affectation

AIX version 4.2 (ou ultérieures) prend en charge diverses tailles de groupe d'affectation. JFS sépare l'espace du système de fichiers en groupes d'i-nodes et en blocs disques pour les données utilisateur. Ces groupes sont appelés groupes d'affectation. Leur taille peut être précisée à la création du système de fichiers. Les tailles des groupes d'affectation sont de 8 Mo, 16 Mo, 32 Mo et 64 Mo. A chaque taille correspondent des valeurs npbi. Ces valeurs sont définies ci-après :

```
Groupe d'affectation
Taille en Mo du groupeValeurs  NBPI d'affectation autorisées
8                  512, 1024, 2048, 4096, 8192, 16384
16                   1024, 2048, 4096, 8192, 16384, 32768
32                       2048, 4096, 8192, 16384, 32768, 65536
64                          4096, 8192, 16384, 32768, 65536, 131072
```

Capacité d'adressage d'un fragment de système de fichiers

JFS prend en charge quatre tailles de fragment (taille d'unité de `512`, `1024`, `2048` et `4096` octets d'espace disque contigu. JFS conserve les adresses de fragment en i-nodes et en blocs indirects de 28-bits. Chaque fragment doit être adressable par une valeur comprise entre 0 et (2^{28}).

Taille du journal JFS

La taille du journal JFS est un autre problème lié à la taille. Dans la plupart des cas, les JFS multiples utilisent un journal commun de 4 Mo. Par exemple, après l'installation initiale, tous les systèmes de fichiers du groupe de volumes racine utilisent le volume logique hd8 comme journal JFS commun. La partition du volume logique par défaut est de 4 Mo, et la taille du journal par défaut est d'une partition ; ainsi, le groupe de volume racine intègre normalement un journal JFS de 4 Mo. Lorsqu'un système de fichiers dépasse 2 Go ou que l'espace global du système de fichiers utilisant un seul journal dépasse 2 Go, la taille par défaut du journal peut s'avérer insuffisante. Dans les deux cas, la taille du journal doit être augmentée conjointement à celle du système de fichiers. La taille du journal est limitée à 256 Mo.

Taille maximale de JFS

La taille maximale de JFS est définie à la création du système de fichiers. Par exemple, sélectionner un ratio nbpi de 512 limite la taille du système de fichiers à 8 Go ($512 * 2^{28} = 8$ Go). A la création d'un système de fichiers JFS, les facteurs décrits plus haut (nbpi, taille du fragment et taille du groupe d'affectation) doivent être considérés soigneusement. La limite de taille du système de fichiers est le minimum de $NPBI * 2^{24}$ ou $Tailles\ Fragment * 2^{28}$.

Fichiers volumineux

Cette section traite de la création de fichiers de grande taille et de l'affectation de fichier. Les grands systèmes de fichiers ne concernent qu'AIX version 4.2 ou ultérieures.

Création de systèmes de fichiers pour fichiers volumineux

Les systèmes de fichiers adaptés aux fichiers volumineux peuvent être créés via **crfs** et **mkfs**. Ces deux commandes disposent d'une nouvelle option (`bf=true`) pour spécifier les systèmes de fichiers pour fichiers de grande taille. Les menus SMIT JFS permettent également la création de ce type de systèmes de fichiers.

Géométrie des grands fichiers

Dans les systèmes pour fichiers volumineux, les données stockées avant le déplacement du fichier de 4 Mo sont affectées en blocs de 4096 octets. Celles stockées après sont affectées en blocs disques de 128 Ko. Les blocs des disques volumineux sont constitués de 32 blocs contigus de 4096 octets. Par exemple, un fichier de 132 Mo intégré à un système pour fichiers volumineux dispose de 1024 blocs disque de 4 Ko et de 1024 blocs disque de 128 Ko. Dans un système de fichiers classique, le fichier de 132 Mo nécessiterait 33 blocs uniques indirects (chacun comprenant 1024 adresses disque de 4 Ko). Cependant, la géométrie des fichiers volumineux requiert seulement 2 blocs indirects uniques pour le fichier de 132 Mo.

Affectation de fichier fractionné

Les fichiers dont tous les blocs disque ne sont pas affectés aux blocs logiques sont dits fractionnés. Ils sont créés par la recherche de deux déplacements de fichiers différents et l'écriture de données. Si les déplacements sont supérieurs à 4 Mo, un bloc disque de 128 Ko est affecté. Les applications utilisant des fichiers fractionnés de plus de 4 Mo peuvent requérir plus de blocs disque dans un système pour fichiers volumineux que dans un système de fichiers classique.

Fragmentation de l'espace disponible

Les blocs disque importants requièrent 32 blocs contigus de 4 Ko. Si vous créez des fichiers volumineux après le fichier de 4 Mo, le déplacement de fichier échouera avec ENOSPC si le système de fichiers ne comprend pas 32 blocs contigus et inutilisés de 4 Ko.

> **Remarque :** Le système de fichiers peut contenir plusieurs milliers de blocs disponibles, mais si 32 d'entre eux ne sont pas contigus, l'affectation échouera.

La commande **defragfs** permet de réorganiser les blocs disque en de plus grandes zones de blocs contigus disponibles.

Compatibilité disque image

Les systèmes pour fichiers volumineux ne peuvent pas migrer vers les versions 3 ou 4.1 d'AIX. Ces versions ne reconnaissent pas le numéro de version du superbloc et peuvent générer le message suivant : `unknown file system type`. Ces fichiers ne sont pas les seuls à être soumis à ce comportement. Reportez-vous à "Description des fragments et du nombre variable d'i-nodes", page 7-17 pour plus d'informations sur la migration des systèmes de fichiers.

> **Remarque :** Les systèmes de fichiers des versions 3 et 4.1 d'AIX peuvent migrer vers des versions ultérieures.

Mise à zéro de kproc pour affectation de fichiers volumineux

JFS est requis pour initialiser toutes les nouvelles affectations de disques. JFS lance la procédure de noyau kproc utilisée pour mettre à zéro les affectations initiales de fichiers lors du montage du premier système pour fichiers très volumineux. La procédure kproc subsiste après la réussite du montage du système pour fichiers très volumineux.

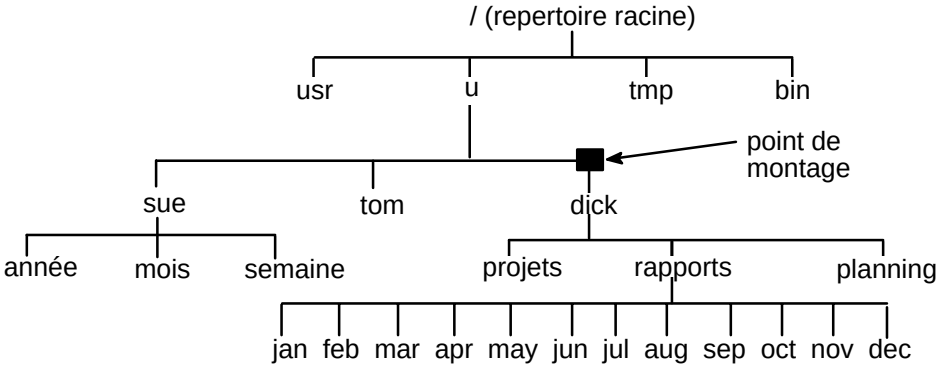
Montage : généralités

Le *montage* met les systèmes de fichiers, les fichiers, les répertoires, les unités et les fichiers spéciaux à disposition de l'utilisateur à un emplacement spécifique. Il représente l'unique moyen de donner l'accès à un système de fichiers. Par le biais de la commande **mount**, le système d'exploitation reçoit l'instruction d'associer un système de fichiers au répertoire spécifié.

Pour monter un fichier ou un répertoire, vous devez posséder les droits d'accès à ce fichier ou à ce répertoire, et avoir l'autorisation d'écriture au niveau du point de montage. Les membres du groupe système également peuvent procéder à des montages d'unités (dans lesquels les systèmes de fichiers ou les unités sont montés sur les répertoires), ainsi qu'aux montages décrits dans le fichier **/etc/filesystems** les montages décrits plus loin. L'utilisateur racine peut aussi monter un système de fichiers arbitrairement, en nommant l'unité et le répertoire sur la ligne de commande. Dans **/etc/filesystems**, il est possible de définir les montages qui seront automatiquement effectués à l'initialisation du système. La commande **mount** sert au montage une fois le système démarré.

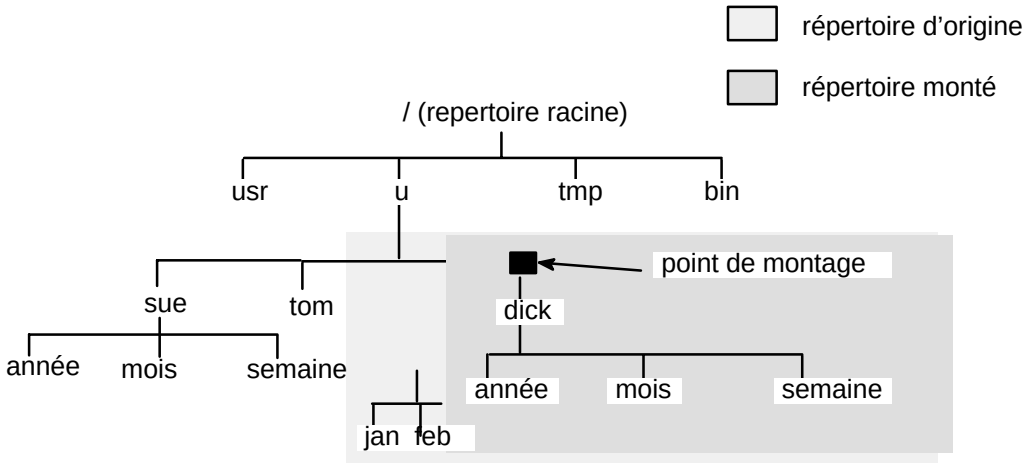
Description des points de montage

Un *point de montage* est un répertoire ou un fichier au niveau duquel un nouveau système de fichiers, un répertoire ou un fichier est accessible. Le point de montage d'un fichier est obligatoirement un fichier, et celui d'un répertoire ou d'un système de fichiers, obligatoirement un répertoire. Le point de montage d'un système de fichiers est illustré ci-après.



Arborescence de fichiers avant montage

Un système de fichiers, un répertoire ou un fichier est généralement monté au niveau d'un point de montage vide, mais ceci n'est pas obligatoire. Si le répertoire ou le fichier servant de point de montage contient des données, celles-ci ne sont plus accessibles. En effet, elles sont recouvertes par les données du répertoire ou du fichier monté qui se trouvait préalablement dans ce répertoire. Elles redeviennent accessibles après le démontage du répertoire ou du fichier monté. La figure ci-après illustre le montage d'un système de fichiers.



Arborescence de fichiers après montage

Quand un système de fichiers est monté sur un répertoire, les droits associés au répertoire racine du système de fichier monté ont priorité sur ceux du point de montage. Une exception, toutefois, concernant le répertoire parent .. (point point) du répertoire de montage. Ses informations doivent être disponibles pour permettre au système d'exploitation d'accéder au nouveau système de fichiers.

Par exemple, si le répertoire courant est `/home/frank`, la commande **cd ..** passe au répertoire `/home`. Si `/home/frank` est le répertoire racine d'un système de fichiers monté, le système d'exploitation doit avoir accès aux informations du répertoire parent dans `/home/frank` pour faire aboutir la commande **cd ..**.

Pour toute commande dont l'exécution requiert des informations du répertoire parent, l'utilisateur doit être autorisé à rechercher ces informations dans le répertoire de montage. En cas d'échec du répertoire de montage à accorder cette autorisation, le résultat est imprévisible, d'autant que ce type d'autorisation n'est pas visible. L'échec de la commande **pwd** est courant. En l'absence d'autorisation, **pwd** renvoie le message :

```
pwd: Permission denied
```

Affecter la valeur 111 aux autorisations du répertoire de montage permet d'éviter ce problème.

Montage des systèmes de fichiers, des répertoires et des fichiers

Il existe deux types de montages : local et à distance. Le *montage à distance* s'effectue sur un système distant par transfert des données sur une ligne de télécommunication. Les systèmes de fichiers distants, tels que NFS, doivent être exportés avant d'être montés. Le *montage local* est effectué sur le système local.

Chaque système de fichiers est associé à une unité distincte (volume logique). Un système de fichiers, pour être exploité, doit être connecté au préalable à la structure de répertoire existante (soit le système de fichiers racine soit un autre système de fichiers déjà connecté). C'est la commande **mount** qui se charge de cette connexion.

Plusieurs chemins permettent d'accéder au même système de fichiers, répertoire ou fichier. Par exemple, pour l'accès multi-utilisateur à une base de données, il est préférable d'avoir plusieurs points de montage. Chaque montage doit avoir ses propres noms et mot de passe, pour des raisons de suivi et de séparation des travaux. Ainsi, le même système de fichiers peut être monté sur différents points de montage. Par exemple, à partir de `/home/server/database`, vous pouvez monter au niveau du point de montage `/home/user1`, `/home/user2` et `/home/user3` :

```
/home/server/database        /home/user1
/home/server/database        /home/user2
/home/server/database        /home/user3
```

Un système de fichiers, un répertoire ou un fichier peut être mis à disposition de l'utilisateur par le biais de liens symboliques. Ces liens symboliques sont créés par le biais de la commande **ln –s**. Les liens entre plusieurs utilisateurs et un fichier central permettent de refléter les modifications du fichier à chaque accès utilisateur.

Contrôle des montages automatiques

Les montages peuvent être configurés pour s'effectuer automatiquement à l'initialisation du système. Il existe deux types de montages automatiques : les montages requis pour amorcer et exploiter le système, et les montages utilisateur. En ce qui concerne les premiers, les systèmes de fichiers sont automatiquement montés par le processus d'amorçage. Leurs strophes, dans le fichier **/etc/filesystems**, sont assorties de **mount = automatic**. Le second type de montage est contrôlé par l'utilisateur. Les systèmes de fichiers sont montés automatiquement par le script **/etc/rc** à l'exécution de la commande **mount all**. Les strophes de ces systèmes sont assorties de **mount = true** dans **/etc/filesystems**.

C'est le fichier **/etc/filesystems** qui contrôle les montages automatiques, effectués un à un, selon la hiérarchie. L'ordre spécifié dans ce fichier peut être modifié et restructuré.

/etc/filesystems est structuré en strophes, une par montage. La strophe décrit les attributs du système de fichiers correspondant et le procédé de montage. Les systèmes de fichiers sont montés dans l'ordre où ils figurent dans **/etc/filesystems**. Voici un extrait de **/etc/filesystems** avec un exemple de strophes :

```
/:
 dev=/dev/hd4
 vol="root"
 mount=automatic
 check=false
 free=true
 vfs=jfs
 log=/dev/hd8
 type–bootfs

/home:
 dev=/dev/hd1
 vfs=jfs
 log=/dev/hd8
 mount=true
 check=true
 vol="/home"
 free=false

/usr:
 /dev=/dev/hd2
 vfs=jfs
 log=/dev/hd8
 mount=automatic
 check=false
 type=bootfs
 vol="/usr"
 free=true
```

Pour vérifier l'ordre de montage, vous pouvez afficher le fichier **/etc/filesystems**. Quand un montage n'aboutit pas, le processus de montage continue. Par exemple, si le montage du système de fichiers **/home** échoue, le système de fichier suivant, **/usr**, est monté. Un montage peut échouer en raison d'une erreur de typographie, d'une dépendance ou d'un incident système.

Description du montage sécurisé sur les clients sans disque

Une station de travail sans disque doit être capable de créer des fichiers unité spéciaux et d'y accéder sur des machines distantes, pour monter des répertoires **/dev** à partir d'un serveur. Le serveur ne peut distinguer si ces fichiers lui sont dédiés ou s'ils sont dédiés à un client ; l'utilisateur du serveur peut ainsi avoir accès aux unités physiques du serveur, par le biais des fichiers unité spéciaux du client.

Par exemple, la propriété d'un **tty** est automatiquement définie pour l'utilisateur exploitant ce **tty**. Si les ID utilisateur ne sont pas les mêmes sur le client et sur le serveur, un utilisateur qui ne détient pas de privilège utilisateur sur le serveur peut avoir accès à un **tty** exploité par un autre utilisateur sur le serveur.

Un utilisateur doté de privilèges sur un client peut créer des fichiers unité spéciaux qui vont correspondre aux unités physiques du serveur. Ainsi, ces unités ne requièrent pas de privilège d'accès et l'utilisateur peut exploiter un compte non privilégié sur le serveur pour avoir accès à des unités qui sont normalement protégées, ceci par le biais des fichiers unité spéciaux qu'il a créés.

Le même phénomène se produit avec l'utilisation des programmes **setuid** et **setgid** sur le client et sur le serveur. Pour l'exploitation normale du système, les clients sans disque doivent avoir la possibilité de créer et d'exécuter les programmes **setuid** et **setgid** sur le serveur. Là encore, le serveur ne distingue pas si ces programmes sont dédiés au client ou à lui-même.

En outre, les ID utilisateur et les ID de groupe ne correspondent pas nécessairement entre le serveur et le client, permettant ainsi aux utilisateurs sur le serveur de lancer des programmes avec des fonctions qui ne leur étaient pas destinées.

En principe, les programmes **setuid** et **setgid** et les fichiers unité spéciaux ne devraient être exploitables que sur la machine qui les a créés.

La solution consiste à utiliser les options de sécurité de la commande **mount** (décrites ci-après) pour limiter l'exploitation de ces programmes. Ces options figurent également dans les strophes du fichier **/etc/filesystems**.

L'option **nosuid** empêche l'exécution des programmes **setuid** et **setgid**, accessibles via le système de fichiers monté. Utilisez-la pour tout système de fichiers monté sur un hôte particulier pour être exploité uniquement par un autre hôte (par exemple, un système de fichiers exporté pour des clients sans disque).

L'option **nodev** empêche l'ouverture des unités qui utilisent des fichiers unité spéciaux accessibles via le système de fichiers monté. Utilisez-la également pour tout système de fichiers monté sur un hôte particulier pour être exploité uniquement par un autre hôte (par exemple, un système de fichiers exporté pour des clients sans disque).

Montage sur clients sans disque

Bien que les systèmes de fichiers d'une station de travail sans disque soient montés à partir du répertoire **/exports** du serveur, la machine sans disque ne fait aucune de différence entre ces systèmes de fichiers et ceux d'une machine autonome.

Montage sur clients sans disque :

Exportations serveur	Importations sans disque
/export/root/ *NomHôte*	/ (répertoire racine)
/export/exec/ *NomSPOT*	**/usr**
/export/home/ *NomHôte*	**/home**
/export/share	/usr/share
/export/dump	espace de cliché des clients sans disque.
/export/swap	espace de pagination à distance des clients sans disque.

Pour plus de détails, reportez-vous à "Description du répertoire /export", page 7-12.

Sécurité des montages

Généralement, les utilisateurs n'ont aucun droit d'accès au répertoire **/export**.

Exportation de /export/root

Le répertoire **/export/root** n'est exportable qu'avec des autorisations d'accès en lecture-écriture, et l'utilisateur racine du serveur doit y avoir accès. Pour le montage de ce répertoire, les deux options suivantes de la commande **mount** sont à votre disposition :

nosuid	empêche un utilisateur du serveur d'exécuter les programmes client **setuid**.
nodev	empêche un utilisateur d'accéder aux unités du serveur avec un fichier unité spécial (fichier client).

Au lieu de monter le répertoire **/export/root** avec ces options, vous pouvez ne donner aucun droit d'accès à ce répertoire aux utilisateurs exploitant le serveur.

Exportation de /export/exec

Le répertoire **/export/exec** n'est exportable qu'avec des autorisations d'accès en lecture, et l'utilisateur racine doit y avoir accès. Pour le montage de ce répertoire, les deux options suivantes de la commande **mount** sont à votre disposition :

nosuid	empêche un utilisateur du serveur d'exécuter les programmes client **setuid**. Vous ne pouvez pas utiliser cette option si vous exportez le répertoire **/usr** du serveur.
nodev	empêche un utilisateur d'accéder aux unités du serveur avec un fichier unité spécial (fichier client).

Exportation de /export/share

Le répertoire **/export/share** n'est exportable qu'avec des autorisations d'accès en lecture, et l'utilisateur racine doit y avoir accès. Pour le montage de ce répertoire, qui contient généralement uniquement des données (et non des fichiers exécutables, ni des fichiers d'unités), vous n'avez pas besoin des options de sécurité.

Exportation de /export/home

Vous avez le choix entre trois méthodes :

- Vous pouvez monter le répertoire **/export/home/***NomHôteClient* sur le répertoire **/home** du client. Dans ce cas, le client doit avoir l'autorisation d'accès en lecture-écriture et l'utilisateur racine doit avoir accès au répertoire. Pour garantir la sécurité du système, montez **/export/home** avec la commande **mount** assortie des options suivantes :

nosuid	empêche un utilisateur du serveur d'exécuter les programmes client **setuid**.
nodev	empêche un utilisateur d'accéder aux unités du serveur avec un fichier unité spécial (fichier client).

- Vous pouvez monter le répertoire **/home** du serveur sur le répertoire **/home** du client. Dans ce cas, **/home** doit être exporté avec des autorisations en lecture-écriture et sans accès racine. Pour garantir la sécurité du système, montez **/home** sur le serveur et sur le client avec la commande **mount** assortie de **nosuid** et **nodev**.
- Vous pouvez monter chaque répertoire **/home/***NomUtilisateur* du serveur sur le répertoire **/home/***NomUtilisateur* du client. Ceci permet aux utilisateurs de se connecter sur plusieurs machines et de conserver l'accès à leurs répertoires personnels. Dans ce cas, les répertoires **/home/***NomUtilisateur* du serveur et des clients doivent être montés avec la commande **mount** assortie des options **nousid** et **nodev**.

Exportation de /export/dump

Le répertoire **/export/dump/***NomHôteClient* n'est exportable qu'avec des autorisations d'accès en lecture-écriture. En outre, seul l'utilisateur racine du serveur et aucun autre, doit y avoir accès.

Exportation de /export/swap

Le répertoire **/export/swap/***NomHôteClient* n'est exportable qu'avec des autorisations d'accès en lecture-écriture. Aucune mesure de sécurité n'est nécessaire. En outre, seul l'utilisateur racine du serveur et aucun autre, doit y avoir accès.

Chapitre 8. Espace de pagination et mémoire virtuelle

Ce chapitre décrit les différents types d'espace de pagination, ainsi que les règles et procédures d'affectation de ces espaces. Pour l'impact sur les performances, reportez-vous à correspondante dans *AIX - Guide d'optimisation*.

Les sujets traités sont les suivants :

Espace de pagination - généralités

Un *espace de pagination* est une zone fixe de stockage sur disque, dédiée à des données, qui réside en mémoire virtuelle et à laquelle l'accès n'est pas courant. Cet espace, également désigné par espace de permutation (swap), est un volume logique dont l'attribut type est défini à *paging*. Ce type de volume logique est appelé volume logique d'espage de pagination ou simplement pagination. Si le système ne dispose que d'une faible quantité de mémoire réelle, les programmes et les données qui n'ont pas été récemment exploitées sont transférés dans l'espace de pagination pour libérer de la mémoire réelle.

Les points suivants traitent des espaces de pagination :

- Gestion des espaces de pagination, page 8-6,
- Règles d'affectation, page 8-3.

Voici les différentes procédures permettant de gérer les espaces de pagination :

- Ajout/activation d'un espace de pagination,
- Modification/suppression d'un espace de pagination,
- Réduction/déplacement de l'espace de pagination hd6.

Voici les différentes procédures dans *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités* qui permettent de gérer les espaces de pagination :

La taille par défaut de l'espace de pagination est définie lors de la personnalisation du système à l'installation d'AIX, en tenant compte des éléments suivants :

- L'espace minimal de pagination est de 16 Mo, sauf pour hd6 pour lequel il est de 32 Mo sous AIX version 4.2.1 et ultérieures.
- L'espace de pagination ne peut occuper plus de 20 % de l'espace disque total.
- Sur un système avec une mémoire réelle inférieure à 32 Mo, la taille de l'espace de pagination est le double de celle de la mémoire réelle.
- Sur un système avec une mémoire réelle supérieure à 32 Mo, la taille de l'espace de pagination est égale à celle de la mémoire réelle plus 16 Mo.

Il existe un autre type d'espace de pagination, accessible par le biais d'une unité qui fait appel à un serveur NFS pour stocker l'espace de pagination. L'accès du client NFS à cet espace suppose la création et l'exportation d'un fichier depuis le serveur NFS vers ce client. Pour le client, la taille de ce fichier est celle de l'espace de pagination.

Pour définir l'espace de pagination, créez un volume logique dédié ou agrandissez les volumes logiques d'espace de pagination existant. Pour agrandir un espace de pagination NFS, le fichier résidant sur le serveur doit être agrandi sur le serveur.

L'espace total à disposition du système pour la pagination est la somme des tailles des volumes logiques d'espaces de pagination actifs.

Règles d’affectation

Deux modes sont à la disposition du système d’exploitation pour affecter des espaces de pagination : les modes ”late” (mode par défaut) et ”early” définis dans la variable d’environnement **PSALLOC**. Le mode par défaut est l’affectation d’espace de pagination “late”. Pour changer de mode, il suffit de donner à **PSALLOC** la valeur `early`.

Observations sur l’espace de pagination

La taille de l’espace nécessaire dépend du type des activités du système. Un espace sous-évalué peut provoquer la perte de processus et s’il se sature, le système peut ”perdre le contrôle”. Quand une condition d’espace insuffisant est détectée, un espace de pagination complémentaire doit être défini.

Le système contrôle le nombre de blocs disponibles dans l’espace de pagination. Quand le seuil ou le niveau de pré-alerte est atteint, le système informe les processus (excepté **kprocs**) par le biais du signal **SIGDANGER**. Si le nombre de blocs disponibles continue de diminuer, atteignant un second seuil d’alerte (kill), le système transmet un signal **SIGKILL** aux processus exploitant le plus d’espace de pagination et aux processus dépourvus de gestionnaire de signal pour **SIGDANGER** (par défaut, ce signal est ignoré). Il continue d’émettre des signaux **SIGKILL** tant que le nombre de blocs disponibles justifie l’alerte (kill).

Les processus affectant dynamiquement de la mémoire peuvent garantir suffisamment d’espace de pagination en contrôlant les seuils avec la sous-routine **psdanger** ou par le biais de routines d’affectation spécifiques. Vous pouvez utiliser la sous-routine **disclaim** pour empêcher l’arrêt de processus quand le seuil d’alerte (kill) est atteint : définissez un gestionnaire de signal dédié au signal **SIGDANGER** et libérez des ressources mémoire et espace de pagination affectées dans leurs zones de données et de pile, et dans des segments de mémoire partagée.

Comparaison entre l’affectation Late et Early

Le système d’exploitation se sert de la variable d’environnement **PSALLOC** pour définir le mécanisme d’affectation de mémoire et d’espace de pagination. Si cette variable n’est pas définie, a la valeur zéro ou une valeur différente de `early`, le système retient par défaut l’algorithme d’affectation late.

Cet algorithme contribue à une exploitation rentable des ressources disque ; il est compatible avec les applications utilisateur qui souhaitent mettre à profit un algorithme d’affectation fractionnée pour la gestion des ressources. Il ne réserve pas d’espace de pagination quand une demande de mémoire est effectuée mais valide la demande pour affecter l’espace quand les pages sont touchées. Certains programmes affectent de grandes quantités de mémoire virtuelle qui ne sont que partiellement utilisées. Ce sont par exemple des applications techniques utilisant des matrices ou des vecteurs fractionnés comme structures de données. En outre, l’algorithme d’affectation late est plus rentable pour un noyau paginé à la demande et en temps réel tel que le noyau du système d’exploitation.

Pour la version AIX 4.3.2 et les versions ultérieures, l’algorithme d’affectation late est modifié pour retarder l’affectation de l’espace de pagination. Comme susmentionné, dans les versions antérieures à la version AIX 4.3.2, l’espace de pagination était affecté lorsqu’une page était touchée. Cependant, cet espace peut ne jamais être utilisé, en particulier sur les systèmes ayant une grande quantité de mémoire réelle pour lesquels la pagination est rare. Par conséquent, l’affectation de l’espace de pagination est retardé jusqu’à ce qu’il soit nécessaire d’évacuer une page ; ainsi, aucune affectation d’espace de pagination n’est perdue mais cela entraîne en revanche une surévaluation supplémentaire de l’espace de pagination. Sur un système où une quantité suffisante de mémoire virtuelle permet une pagination, la quantité d’espace de pagination requise peut être la même que celle requise dans les versions précédentes.

Cet algorithme peut surévaluer l'attribution des ressources. Auquel cas, un processus obtenant la ressource avant un autre provoque un incident. Le système d'exploitation s'efforce d'éviter une panne système en tuant les processus affectés par cette surévaluation. Le signal **SIGDANGER** est transmis pour informer certains processus que l'espace de pagination disponible est faible. Si la situation devient encore plus critique, les processus sélectionnés à qui le signal **SIGDANGER** n'avait pas été adressé reçoivent un signal **SIGKILL**.

L'utilisateur peut passer à l'algorithme d'affectation early avec la variable d'environnement **PSALLOC**. Cet algorithme affecte de l'espace de pagination au processus en cours lors de la demande de mémoire. Si l'espace de pagination est insuffisant au moment de cette demande, l'affectation de mémoire ne peut aboutir.

Quand la variable **PSALLOC** a la valeur `early`, tout programme qui a démarré dans l'environnement concerné, processus en cours exclus, est exécuté dans l'environnement d'affectation early. Les interfaces telles que les sous-routines **malloc** et **brk** ne peuvent pas aboutir si suffisamment d'espace de pagination ne peut pas être réservé au moment de la demande.

Les processus exécutés dans cet environnement ne reçoivent pas de signal **SIGKILL** en cas d'insuffisance d'espace de pagination.

Les sous-routines d'interface d'affectation de mémoire répertoriées ci-après sont affectées par le passage à un environnement d'affectation early :

- **malloc**
- **free**
- **calloc**
- **realloc**
- **brk**
- **sbrk**
- **shmget**
- **shmctl**

Définition de PSALLOC pour le mode early

Voici quelques exemples illustrant les différentes méthodes pour donner à la variable d'environnement **PSALLOC** la valeur `early`. Ces exemples expliquent également les résultats obtenus.

1. La commande suivante entrée sur la ligne de commande shell :

   ```
   PSALLOC=early;export PSALLOC
   ```

   Exécutez en mode early toutes les commandes ultérieures de la session shell.

2. La commande ci-après insérée dans un fichier **.shrc** ou **.kshrc** :

   ```
   PSALLOC=early;export PSALLOC
   ```

   Exécutez en mode early tous les processus de la session de connexin utilisateur, excepté le shell de connexion; Ne déclenchez pas le mécanisme du signal **SIGKILL**.

3. La commande suivante est insérée dans le fichier **/etc/environment** :

   ```
   PSALLOC=early
   ```

   Exécutez en mode early tous les processus du système, excepté le processus **init** (ID 1), et ne déclenchez pas le mécanisme du signal **SIGKILL**.

4. Pour donner à la variable **PSALLOC** la valeur `early` à partir d'un programme, utilisez la sous-routine **putenv**. Cette nouvelle valeur sera appliquée à l'appel suivant de la sous-routine **exec**.

Observations sur le mode early

L'algorithme d'affectation early garantit l'espace de pagination requis par une demande d'affectation mémoire. Une affectation adéquate de l'espace de pagination est en effet importante pour exploiter efficacement le système. Quand l'espace disponible diminue en deçà d'un certain seuil, les nouveaux processus ne peuvent démarrer et les processus en cours n'ont pas l'assurance d'obtenir de la mémoire supplémentaire. Tout processus en cours en mode d'affectation late (mode par défaut) devient très vulnérable par rapport au mécanisme du signal **SIGKILL**. En outre, le noyau du système d'exploitation requérant parfois une affectation de mémoire, la saturation de l'espace de pagination est susceptible de provoquer une panne système.

Avant d'étendre le mode d'affectation early à l'ensemble du système, il est très important de définir une quantité d'espace de pagination adéquate. Pour évaluer l'espace nécessaire, il faut tenir compte de la façon dont est exploité le système et des programmes utilisés. Le plus souvent, ce mode requiert plus d'espace que le mode par défaut. Au départ, un espace de pagination quatre fois plus important que la quantité de mémoire physique du système est un bon compromis.

Certaines applications, en mode d'affectation early, utilisent énormément d'espace de pagination. Dans ce mode, le serveur AIXwindows requiert couramment plus de 250 Mo d'espace de pagination. Pour une application, l'espace de pagination nécessaire dépend de la façon dont l'application est programmée et exploitée.

Toutes les commandes et sous-routines montrant un espace de pagination et traitant l'exploitation de la mémoire ont un espace de pagination qui leur est attribué en mode early. La commande **lsps** assortie de l'indicateur **–s** affiche le total de l'affectation d'espace de pagination, y compris l'espace attribué en mode early.

Interface de programmation

L'interface de programmation qui contrôle le mode d'affectation de l'espace de pagination utilise la variable d'environnement **PSALLOC**. Pour vérifier si l'application tourne toujours sous le mode voulu (avec ou sans affectation early) :

1. Utilisez la sous-routine **getenv** pour examiner l'état courant de la variable **PSALLOC**.
2. Si la valeur de **PSALLOC** n'est pas celle requise par l'application, utilisez la sous-routine **setenv** pour la modifier. La sous-routine **execve** étant l'unique sous-routine capable d'examiner l'état de **PSALLOC**, appelez-la avec l'environnement et le même ensemble de paramètres que l'application. Lors de ce nouvel examen, une fois la valeur correcte trouvée, l'application se poursuit normalement.
3. Si **getenv** révèle que l'état de **PSALLOC** est correct, vous n'avez rien à modifier. L'application se poursuit normalement.

Gestion des espaces de pagination

Les commandes suivantes sont dédiées à la gestion des espaces de pagination :

chps	modifie les attributs d'un espace de pagination.
lsps	affiche les caractéristiques de l'espace de pagination.
mkps	ajoute un espace de pagination.
rmps	supprime un espace de pagination inactif.
swapon	active un espace de pagination.

Pour créer un volume logique d'espace de pagination, **mkps** se sert de la commande **mklv** assortie d'un ensemble d'options. Pour créer un espace de pagination NFS, elle fait appel à la commande **mkdev** assortie d'un autre ensemble d'options spécifiques. Certaines des caractéristiques suivantes sont requises pour tous les types d'espace de pagination :

- type de pagination,
- pas de réaffectation de blocs défectueux,
- pas de traitement miroir.

Les options ci-après visent à optimiser les performances :

- affectation au milieu du disque (pour réduire la course du bras du disque),
- espaces de pagination multiples (affectés sur de volumes physiques distincts).

Pour les espaces de pagination NFS, **mkps** a besoin du nom d'hôte du serveur NFS et du chemin d'accès au fichier exporté du serveur.

La commande **swapon** sert en début d'initialisation du système pour activer la première unité d'espace de pagination. Dans une phase ultérieure, quand les autres unités sont disponibles, **swapon** active les autres espaces de pagination pour répartir l'activité de pagination sur plusieurs unités.

Un espace de pagination actif ne peut pas être supprimé. Il doit d'abord être désactivé. Pour cefaire, utilisez la commande **chps** : ainsi, au redémarrage du système, il sera inactif et pourra ensuite être supprimé avec la commande **rmps**.

Le fichier **/etc/swapspaces** indique les unités d'espace de pagination activées par la commande **swapon –a**. Dans ce fichier, est ajouté tout espace de pagination créé par la commande **mkps –a**, est ôté tout espace de pagination supprimé par la commande **rmps**, et est ajouté ou supprimé tout espace de pagination directement par la commande **chps –a**.

VMM - généralités

Le gestionnaire de mémoire virtuelle VMM (Virtual Memory Manager) fournit les fonctions de mémoire virtuelle exploitées par les autres composants du système pour mettre en oeuvre les éléments suivants :

- espace d'adresses virtuelles des process,
- partage des exécutables
- segments de mémoire partagée,
- fichiers mappés.

VMM implante la mémoire virtuelle, permettant la création de segments plus grands que la mémoire physique disponible sur le système. Ces segments sont divisés en unités de taille fixe appelées *pages*. Dans un segment, chaque page réside en mémoire physique ou est stockée sur disque tant qu'elle n'est pas exploitée. Lorsqu'un process accède à une page absente en mémoire physique, VMM lit la page dans la mémoire ; cette opération s'appelle *PageIn*. En l'absence de mémoire physique disponible, VMM écrit les pages sur disque, opération appelée *PageOut* ou *PageSteal*.

Voici quelques types de segments :

Mémoire de travail	segments dédiés à l'implantation de zones de données pour les process et les segments de mémoire partagée. Les pages sont stockées dans les espaces de pagination configurés sur le système.
Mémoire permanente	segments servant à manipuler les fichiers et les répertoires. Lors de l'accès à ces segments, les pages sont lues et écrites dans leur système de fichiers.
Mémoire client	segments dédiés à l'implantation de certains systèmes de fichiers virtuels, tels que NFS et le système de fichiers CD–ROM. Les pages de ces segments client sont stockées sur une machine locale ou distante.

Chapitre 9. Sauvegarde et restauration

Ce chapitre donne des informations relatives aux méthodes de sauvegarde et de restauration des données.

Les sujets traités sont les suivants :

Sauvegarde - généralités

Une fois votre système opérationnel, l'impératif suivant est la sauvegarde des systèmes de fichiers, des répertoires et des fichiers. Répertoires et fichiers peuvent représenter un investissement considérable en termes d'efforts et de temps de travail. En outre, sur un ordinateur, il est très facile d'effacer (intentionnellement ou accidentellement) des fichiers. Avec une approche rigoureuse et méthodique de la sauvegarde des systèmes de fichiers, vous serez toujours en mesure d'en restaurer les versions récentes sans grand effort. Quand un disque est en panne, les données qu'il contient sont détruites ; le seul moyen de récupérer ces données est de les restaurer à partir de la copie de sauvegarde.

Méthodes de sauvegarde

Il existe différentes méthodes de sauvegarde. Une des plus courantes est la sauvegarde par nom, également appelée archivage par nom de fichier. Avec cette méthode, l'indicateur **i** est spécifié pour effectuer une copie de sauvegarde des fichiers et répertoires. Les utilisateurs emploient couramment cette méthode pour sauvegarder leurs comptes.

La sauvegarde par système de fichiers, aussi appelée sauvegarde par i-node ou archivage par système de fichiers est également fréquemment employée. Sans faire appel à l'indicateur **i**, elle permet de sauvegarder la totalité d'un système de fichiers. Les administrateurs utilisent couramment cette méthode pour les groupes de fichiers volumineux, tels que les comptes utilisateur dans **/home**. Avec cette méthode, la sauvegarde incrémentale de systèmes de fichiers est une opération simple. Cette sauvegarde incrémentale permet de sauvegarder tous les fichiers modifiés depuis la dernière sauvegarde.

Avec les commandes **compress** et **pack**, vous pouvez compresser ou condenser les fichiers pour les stocker, les commandes **uncompress** et **unpack** permettant ensuite de les décompresser ou de les décondenser une fois restaurés. Ces différents processus demandent du temps mais les données compressées mobilisent moins de place sur le support de sauvegarde.

Il existe plusieurs commandes qui créent des sauvegardes et des archives. C'est la raison pour laquelle les données sauvegardées doivent être étiquetées afin d'identifier la commande et la méthode (par nom ou par système de fichiers) employées. **backup** est la commande la plus utilisée. Voici la description de ces différentes commandes :

backup	sauvegarde les fichiers par nom ou par système de fichiers.
mksysb	crée une image installable du groupe de volumes rootvg.
cpio	copie les fichiers vers et à partir de l'archivage. Peut généralement lire les données archivées sur une autre plate-forme à condition qu'elles soient au format cpio.
dd	convertit et copie un fichier. Cette commande est couramment employée pour convertir et copier des données vers et à partir de systèmes non-AIX, par exemple, des gros systèmes. **dd** ne regroupe pas plusieurs fichiers en une seule archive ; elle sert au transfert et à la manipulation de données.
tar	manipule les archives au format tar.
rdump	commande réseau qui sauvegarde les fichiers par système de fichiers sur l'unité spécifiée d'une machine distante.
pax	utilitaire d'archivage compatible avec POSIX qui lit et écrit des archives au format **tar** et **cpio**.

Choix d’une politique de sauvegarde

Une politique de sauvegarde ne peut à elle seule répondre aux besoins de tous. Une politique adaptée à un système monoutilisateur, par exemple, peut être inadéquate pour un système servant cinq ou dix utilisateurs. De même, une politique développée pour un système dont nombre de fichiers sont modifiés chaque jour ne convient pas à un système avec des données rarement modifiées. Quelle que soit la stratégie de sauvegarde sur votre site, l’important est qu’elle existe et qu’elle soit appliquée fréquemment et régulièrement. A défaut de stratégie efficace et opérationnelle, il peut s’avérer difficile de restaurer des données perdues.

C’est vous qui ferez le choix de la politique à adopter. Voici toutefois quelques règles qui vous guideront :

- Prévenir les pertes importantes de données

  La poursuite de l’activité du système est-elle possible après la panne d’un disque dur, quel qu’il soit ? La reprise du système est-elle possible si tous les disques fixes sont en panne ? La reprise du système est-elle possible si vous égarez vos disquettes ou bandes de sauvegarde ? Pour récupérer des données en cas de perte, pouvez-vous mesurer les difficultés rencontrées ? Elaborez ensuite une politique de sauvegarde qui prenne toutes ces questions en compte.

- Vérifier régulièrement les sauvegardes

  Les supports et unités matérielles de sauvegarde ne sont pas toujours fiables. Une bibliothèque volumineuse sur bandes ou disquettes de sauvegarde n’a de valeur que si les données sont restituables et lisibles sur disque. Pour vérifier que vos sauvegardes sont exploitables, affichez régulièrement la table des matières à partir de la bande de sauvegarde (avec la commande **restore –T** ou **tar –t** pour les bandes d’archives). Pour les sauvegardes sur disquettes, lisez régulièrement les disquettes, si possible à partir d’une unité de disquette différente de celle utilisée pour la création des sauvegardes. Pour plus de sécurité, vous pouvez doubler les sauvegardes de niveau 0 sur un deuxième support. Sur les bandes en continu contenant des sauvegardes, appliquez la commande **tapechk** qui vérifie sommairement la cohérence.

- Conserver les anciennes sauvegardes

  Prévoyez des recyclages réguliers des supports de sauvegarde, sans toutefois les réutiliser tous. L’absence ou l’endommagement d’un fichier important n’est pas toujours détectée en temps réel. Il est donc conseillé de conserver quelques sauvegardes anciennes. Les recyclages suivants des bandes ou disquettes de sauvegarde sont indiqués à titre d’exemple :

  – Une fois par semaine, recyclez les disquettes quotidiennes, excepté celle du vendredi.

  – Une fois par mois, recyclez toutes les disquettes hebdomadaires, excepté la plus récente ; les sauvegardes des quatre derniers vendredis sont ainsi toujours disponibles.

  – Tous les trimestres, recyclez toutes les disquettes mensuelles, excepté la dernière. Conservez cette dernière disquette, de préférence dans un autre bâtiment.

- Vérifiez les systèmes de fichiers avant la sauvegarde

  La sauvegarde d’un système de fichiers endommagés risque d’être inexploitable. Avant de faire des sauvegardes, il est bon de vérifier l’intégrité du système de fichiers avec la commande **fsck**.

- S’assurer que les fichiers ne sont pas en cours d’utilisation pendant la sauvegarde

  Pendant les sauvegardes, le système ne doit pas être en cours d’utilisation. Sinon, les fichiers sont susceptibles d’être modifiés auquel cas les sauvegardes seraient inexactes.

- Sauvegarder le système avant toute modification majeure

La sauvegarde complète du système est toujours conseillée avant tout test ou réparation matérielle, avant l'installation d'une unité, d'un programme ou d'autres fonctions système.

Remarque : la sauvegarde des tubes désignés (fichiers spéciaux FIFO) fonctionne, que ceux–ci soient fermés ou ouverts. Toutefois, la restauration est impossible lorsque la sauvegarde est faite sur des tubes ouverts. Lors de la restauration d'un fichier spécial FIFO, son inode est le seul élément indispensable pour le recréer car il contient toutes ses caractéristiques. Le contenu d'un tube désigné n'est pas nécessaire à la restauration. C'est pourquoi, la taille du fichier lors de la sauvegarde doit être zéro (tous les FIFO fermés) avant de lancer cette procédure.

Attention : La sauvegarde et la restauration d'un système doivent être effectuées sur le même type de plate-forme. Les cartes principales CPU et d'E/S notamment doivent être de même type. Les procédures de sauvegarde et restauration étant décrites d'après les tests effectués sur plate-forme IBM ESCALA, elles ne fonctionnent pas nécessairement de la même façon sur d'autres plates-formes.

Description du support de sauvegarde

Il existe plusieurs types de supports de sauvegarde. Ils sont compatibles avec la configuration de votre système en fonction du logiciel et du matériel utilisés. Les bandes 8 mm, les bandes 9 pistes et les disquettes 3 1/2 pouces sont les plus utilisées.

Pour la sauvegarde de fichiers et de systèmes de fichiers individuels, les disquettes sont le support standard. Sauf spécification autre, la commande **backup –f**, par défaut, sauvegarde automatiquement sur **/dev/rfd0** (unité de disquette). Pour sauvegarder sur l'unité de bande par défaut, sélectionnez **/dev/rmt0**.

Attention : La commande **backup** détruit, le cas échéant, les données stockées sur le support de sauvegarde sélectionné.

Restauration des données

Il existe plusieurs méthodes. Choisissez-en une compatible avec celle utilisée pour la sauvegarde.

Vous devez connaître la méthode adoptée pour la sauvegarde ou l'archivage effectué. Chaque procédure de sauvegarde fournit des informations sur la restauration des données. Par exemple, si vous utilisez la commande **backup**, vous pouvez spécifier une sauvegarde par système de fichiers ou par nom. Une sauvegarde effectuée par système de fichiers ou par nom doit être restaurée de la même manière.

Voici les différentes commandes relatives à la restauration des données :

restore	copie les fichiers créés avec la commande **backup**.
rrestore	commande réseau qui copie sur la machine locale les systèmes de fichiers sauvegardés sur une machine distante.
cpio	copie les fichiers vers et à partir de l'archivage.
tar	Manipule les archives. Commande réservée aux répertoires.

Développement d’une stratégie de sauvegarde

Il existe deux méthodes de sauvegarde de grandes quantités de données :

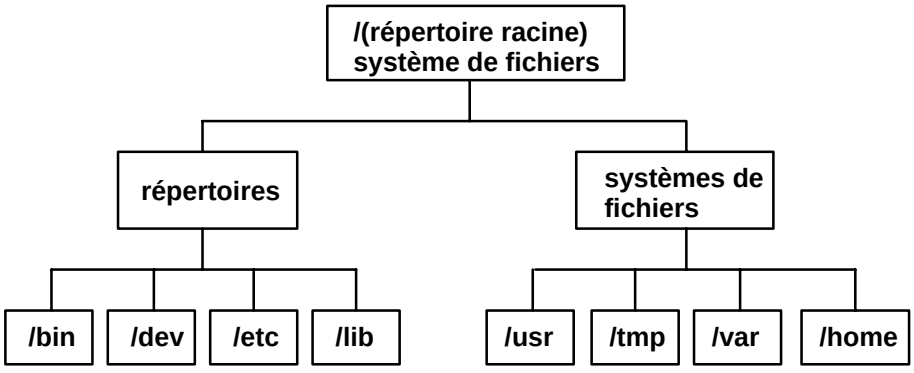
- la sauvegarde intégrale du système,
- la sauvegarde incrémentale.

Pour choisir la méthode la plus adaptée à votre site ou à votre système, il est important de bien appréhender la structure du système de fichiers et le placement des données. La stratégie de placement des données doit être déterminée avant de développer une stratégie de sauvegarde. Reportez–vous à la section ”Planification des sauvegardes” dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d’exploitation et unités* pour un exemple de stratégie combinant la sauvegarde intégrale, hebdomadaire et la sauvegarde incrémentale quotidienne.

Structure du système de fichiers

Notez bien la différence entre un système de fichiers et un répertoire. Un système de fichiers est une section du disque affectée aux données. L’accès à cette section se fait par montage du système de fichiers sur un répertoire. Du point de vue utilisateur, ce système de fichiers, une fois monté, ressemble à un répertoire. Toutefois, en raison des différences structurelles entre les systèmes de fichiers et les répertoires, les données à l’intérieur de ces deux entités peuvent être gérées séparément.

Lorsque le système d’exploitation est installé pour la première fois, il est chargé dans une structure de répertoire, comme indiqué dans l’illustration de l’arborescence du système de fichiers **/root**.



Les répertoires de droite (**/usr**, **/tmp**, **/var** et **/home**) sont tous des systèmes de fichiers, affectés d’une section du disque. Ces systèmes de fichiers sont montés automatiquement à l’amorçage du système ; c’est pourquoi l’utilisateur ne les différencie pas des répertoires de gauche (**/bin**, **/dev**, **/etc** et **/lib**).

Données système et données utilisateur

Les données (programmes ou texte) sont réparties ici en deux catégories :

- Les données système, qui établissent la relation au système d’exploitation et à ses extensions. Elles doivent toujours figurer dans les systèmes de fichiers système, **/** (racine), **/usr**, **/tmp**, **/var**, etc.
- Les données utilisateur, qui sont exploitées localement par les utilisateurs pour effectuer des tâches spécifiques. Elles doivent être stockées dans le système de fichiers **/home** ou dans des systèmes de fichiers créés à cet effet.

Les applications utilisateur et le texte ne doivent en aucun cas être placés dans des systèmes de fichiers dédiés aux données système.

Elles doivent placées plutôt, par exemple, dans un système de fichiers créé par l'administrateur et monté sur un répertoire nommé **/local**.

Sauvegarde

Les sauvegardes de données système et utilisateur sont conservées en cas de destruction accidentelle ou de défaillance d'un disque. Il est plus facile de gérer des sauvegardes distinctes pour les données système et les données utilisateur. Les raisons sont les suivantes :

- Les données utilisateur sont beaucoup plus souvent modifiées que les données du système d'exploitation. En outre, les images de sauvegarde sont beaucoup plus petites quand les deux types de données ne sont pas sur la même image. Par ailleurs, le nombre d'utilisateurs affecte le support et la fréquence du stockage nécessaires.
- La restauration des données utilisateur est plus facile et plus rapide quand la sauvegarde est séparée des données système. La restauration du système d'exploitation en même temps que des données utilisateur prend plus de temps et demande plus d'efforts. En effet, pour restaurer les données système, il faut amorcer le système à partir d'un support amovible (bande ou CD-ROM) puis installer la sauvegarde système.

Pour sauvegarder les données système, démontez tous les systèmes de fichiers utilisateur, y compris **/home** avec la commande **umount**. Si ces systèmes de fichiers sont en cours d'exploitation, vous ne pourrez pas les démonter. Prévoyez donc vos sauvegardes en dehors des heures d'exploitation ; si les systèmes de fichiers utilisateur ne sont pas démontés, ils sont sauvegardés avec les données du système d'exploitation. Ensuite, pour sauvegarder uniquement les données du système d'exploitation, entrez la commande :

```
mount,
```

Seuls les systèmes de fichiers **/**, **/usr**, **/var** et **/tmp** seront sauvegardés. A l'issue de la commande **mount**, un écran semblable à celui-ci s'affiche :

```
  node  mounted  mounted over vfs       date         options

        /dev/hd4     /         jfs   Jun 11 10:36  rw,log=/dev/hd8

        /dev/hd2     /usr      jfs   Jun 11 10:36  rw,log=/dev/hd8

        /dev/hd9var  /var      jfs   Jun 11 10:36  rw,log=/dev/hd8

        /dev/hd      /tmp      jfs   Jun 11 10:36  rw,log=/dev/hd8
```

Une fois tous les systèmes de fichiers utilisateur démontés, reportezvous à "Sauvegarde du système", pour plus de détails sur la sauvegarde des données du système d'exploitation.

Quand la sauvegarde du système d'exploitation est terminée, montez le système de fichiers utilisateur avec la commande **smit mount**. Vous pouvez ensuite sauvegarder fichiers, systèmes de fichiers ou autres groupes de volumes, en fonction de vos besoins. Les procédures correspondantes sont décrites plus loin dans ce chapitre.

Reproduction d'un système (clonage)

Le clonage permet de sauvegarder les données de configuration avec les données utilisateur ou les données système. La reproduction d'un système ou d'un groupe de volumes est parfois appelée clonage. L'image obtenue est installable sur un autre système et donc exploitable comme sur le premier système. La commande **mksysb** sert au clonage du groupe de volumes rootvg, qui contient le système d'exploitation tandis que la commande **savevg** sert au clonage des autres groupes de volumes. Les procédures de sauvegarde du système et des groupes de volumes utilisateur sont décrites plus loin dans ce chapitre.

Sauvegarde des systèmes de fichiers et fichiers utilisateur

Il existe trois procédures de sauvegarde des fichiers et systèmes de fichiers utilisateur : le raccourci Web-based System Manager **wsm fs**, les raccourcis SMIT **smit backfile** ou **smit backfilesys**, et la commande **backup**.

L'interface SMIT est adaptée à la sauvegarde par nom de fichiers et de systèmes de fichiers de petite taille, tels que **/home** sur le système local. SMIT peut créer des archives aux formats fournis par la commande **backup**. Dans SMIT, tous les indicateurs de la commande **backup** ne sont pas disponibles, pour éviter des dialogues SMIT trop confus. SMIT s'arrête quand plusieurs bandes ou disques sont nécessaires en cours de sauvegarde (reportez-vous à "Sauvegarde par nom" dans la commande **backup**).

Utilisez la commande **backup** pour sauvegarder plusieurs grands systèmes de fichiers. Vous pouvez spécifier un numéro de niveau pour contrôler la quantité de données à sauvegarder (sauvegarde intégrale, 0 ou incrémentale, 1 à 9). **backup** est la seule commande permettant d'indiquer un niveau.

Cette commande crée des copies de sauvegarde dans un des deux formats suivants :

- fichiers spécifiques sauvegardés par nom avec **backup** assortie de l'indicateur **i**.
- systèmes de fichiers sauvegardés intégralement par i-node avec les paramètres *–Niveau* et *SystèmeFichiers*. L'avantage est que la sauvegarde est défragmentée lors de sa restauration.

  Attention : La sauvegarde par i-node n'est pas compatible avec des fichiers dont l'UID (ID utilisateur) ou le GID (ID groupe) est supérieur à 65535. Ces ID étant tronqués, leurs attributs sont incorrects à la restauration. C'est la sauvegarde par nom qui est adaptée à ces fichiers.

Sauvegarde de l’image système et des groupes de volumes définis par l’utilisateur

L’image de sauvegarde a une double fonction : d’une part, elle restaure un système endommagé à partir de sa propre image. D’autre part, elle transfère le logiciel installé et configuré d’un système sur un autre système. Vous pouvez sauvegarder le système ou les groupes de volumes à l’aide de Web-based System Manager, SMIT ou des procédures de commandes.

Le *groupe de volumes rootvg* est un disque ou un groupe de disques contenant les fichiers de démarrage du système (BOS), les données de configuration et tout autre produit logiciel en option. Un *groupe de volumes utilisateur* (ou *groupe de volumes non rootvg*) contient les fichiers de données et les logiciels d’application.

Les procédures SMIT et Web-based System Manager font appel à la commande **mksysb** pour créer une image de sauvegarde, stockée sur bande ou dans un fichier. Si vous optez pour la bande, le programme de sauvegarde écrit une *image d’amorçage* sur la bande et l’image obtenue pourra donc servir à l’installation.

Remarque :

1. Les bandes amorçables ne peuvent être ni créées, ni exploitées sur un PowerPC.
2. Si vous optez pour SMIT, installez d’abord l’ensemble de fichiers **sysbr** dans le progiciel **bos.sysmgt**. Reportez-vous à ”Installation de logiciels en option et de mises à jour de service” dans *AIX Installation Guide*.

Configuration du système source

Configurez le système source avant de créer son image de sauvegarde. Faites-le, excepté si cette image est destinée à l’installation d’autres systèmes (cible) dont les configurations prévues sont différentes.

Le système *source* est celui à partir duquel vous créez la copie de sauvegarde. Le système *cible* est celui sur lequel vous installez cette copie.

Le programme d’installation n’installe que le support logiciel d’unité correspondant à la configuration matérielle de la machine. Aussi, si vous prévoyez d’utiliser une copie du système pour installer d’autres machines, vous aurez probablement d’autres unités à installer sur le système source avant d’en faire l’image de sauvegarde.

Servez-vous du raccourci Web-based System Manager, des **unités wsm**, du raccourci SMIT, de **smit devinst**, pour installer la prise en charge d’une unité supplémentaire sur le système source.

- Si les système source et cible disposent de suffisamment d’espace disque, installez l’ensemble du support logiciel d’unité.
- Si l’espace disque est limité, n’installez que les supports indispensables.

Pour en savoir plus, reportez-vous au chapitre ”Installation de logiciels en option et de mises à jour de service” dans *AIX Installation Guide*.

Sont transférées par la sauvegarde, du système source vers le système cible, les données relatives :

- à l’espace de pagination,
- aux volumes logiques,
- au groupe de volumes rootvg.
- à la position des partitions logiques (si l’option SMIT Web-based System Manager ou Création de fichiers MAPPES a la valeur **oui**).

Reportez-vous au chapitre ”Personnalisation du programme d’installation de BOS” dans*AIX Installation Guide* pour en savoir plus sur les paramètres d’installation et le moyen de passer outre certains menus pour installer la machine cible à partir d’une sauvegarde système.

Montage et démontage des systèmes de fichiers

La procédure ”Sauvegarde du système” dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d’exploitation et unités* est dédiée exclusivement à la sauvegarde des systèmes de fichiers montés dans le groupe de volumes rootvg. De ce fait, vous devez monter avant la sauvegarde tous les systèmes de fichiers que vous voulez inclure dans la sauvegarde. A l’inverse, vous devez démonter tous ceux que vous ne souhaitez *pas* sauvegarder.

La procédure effectue une double sauvegarde des fichiers figurant dans un répertoire local monté sur un autre répertoire local du même système de fichiers. Par exemple, si **/tmp** est monté sur **/usr/tmp**, les fichiers de **/tmp** sont sauvegardés deux fois. Cette duplication peut provoquer le dépassement du nombre de fichiers admis et, par conséquent, l’échec de la future installation de l’image de sauvegarde.

Remarques sur la sécurité

Si vous installez une image de sauvegarde sur d’autres systèmes, pour des raisons de sécurité, les mots de passe et les adresses de réseau ne doivent pas être copiés sur les systèmes cible. Et ce d’autant que des adresses en double peuvent provoquer l’interruption des communications sur le réseau.

Restauration d’une image de sauvegarde

Pendant l’installation de l’image de sauvegarde, le système vérifie si l’espace disque du système cible est suffisant pour créer tous les volumes logiques stockés sur la sauvegarde. Si cet espace est suffisant, la restauration complète de l’image est possible. Sinon, la procédure s’arrête et le système vous invite à sélectionner des disques supplémentaires.

Les systèmes de fichiers créés sur le système cible ont la même taille que sur le système source, sauf si la variable **SHRINK** était définie à **oui** dans le fichier **image.data** avant l’exécution de la sauvegarde. Une exception cependant : le répertoire **/tmp**, qui peut être agrandi pour réserver suffisamment d’espace à la commande **bosboot**. Pour en savoir plus sur les variables, reportez-vous à la section traitant du fichier **image.data** dans le manuel *AIX Files Reference*.

Une fois l’image de sauvegarde installée, le programme d’installation reconfigure le gestionnaire ODM (Object Data Manager) sur le système cible. Si les deux systèmes n’ont pas exactement la même configuration matérielle, le programme peut modifier les attributs de certaines unités sur le système cible :

- dans tous les fichiers **/etc/objrepos** commençant par Cu,
- dans tous les fichiers du répertoire **/dev**.

Pour en savoir plus sur l’installation et la restauration d’une image de sauvegarde, reportez-vous à Installation de BOS à partir d’une sauvegarde système dans *AIX Installation Guide*.

Chapitre 10. Environnement système

A la base, l'environnement système est l'ensemble de variables qui définissent ou contrôlent certains aspects de l'exécution des processus. Ces variables sont définies ou redéfinies à chaque démarrage d'un shell. Du point de vue de l'administrateur système, il est important de garantir des valeurs correctes pour la connexion de l'utilisateur. La plupart de ces variables sont définies lors de l'initialisation du système, soit par défaut, soit en fonction des valeurs lues dans le fichier **/etc/profile**.

Les sujets abordés sont les suivants :

Profils - généralités

Lors de la connexion au système d'exploitation, le shell utilise deux types de fichiers de profils. Il analyse les commandes figurant dans ces fichiers puis les exécute pour définir votre environnement système. Ces fichiers ont des fonctions similaires à ceci près que **/etc/profile** contrôle les variables de profil concernant l'ensemble des utilisateurs du système tandis que **.profile** permet de personnaliser votre propre environnement.

Ce chapitre traite des points suivants :

- Fichier **/etc/profile**,
- Fichier **.profile**,
- Modification de la date/heure système dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*,
- Modification du message du jour dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*,
- Services de manipulation des données sur l'heure, page 10-3.

Fichier /etc/profile

/etc/profile est le premier fichier qu'utilise le système d'exploitation au moment de la connexion. Il contrôle les variables par défaut à l'échelle du système, telles que :

- les variables d'exportation,
- le masque de création de fichier (umask),
- les types de terminaux,
- les messages signalant l'arrivée du courrier.

L'administrateur système configure le fichier **profile** pour tous les utilisateurs du système. Il est le seul à pouvoir modifier ce fichier.

fichier .profile

.profile est le deuxième fichier qu'utilise le système d'exploitation au moment de la connexion. Ce fichier est présent dans votre répertoire personnel (**$HOME**) ; il vous permet de personnaliser votre environnement de travail. Les commandes de **.profile** ont la priorité sur celles de **/etc/profile**. **.profile** étant un fichier caché, vous avez besoin, pour l'afficher, de la commande **li -a**. Le fichier **.profile** contrôle par défaut :

- les shells à ouvrir,
- l'apparence de l'invite,
- les variables d'environnement (par exemple, les variables de chemin d'accès),
- le son du clavier.

L'exemple suivant illustre un fichier **.profil** courant :

```
PATH=/usr/bin:/etc:/home/bin1:/usr/lpp/tps4.0/user:/home/gsc/bin::
epath=/home/gsc/e3:
export PATH epath
csh
```

Dans cet exemple, deux chemins d'accès ont été définis (`PATH` et `epath`) puis exportés et un shell C a été ouvert (`csh`).

Le fichier **.profile** (ou, à défaut, le fichier **profile**) sert aussi à déterminer les variables shell de connexion. En outre, vous pouvez personnaliser les autres environnements shell. Par exemple, utilisez les fichiers **.chsrc** et **.kshrc** pour personnaliser un shell C et un shell Korn au démarrage de ces deux shells.

Services de manipulation des données sur l’heure

Les fonctions de date/heure servent à accéder à la date et l’heure courantes du système et à modifier leur format. Aucun indicateur n’est à spécifier au compilateur pour exploiter ces fonctions.

Ajoutez le fichier d’en–tête de ces fonctions dans le programme. Pour inclure un fichier d’en–tête, procédez comme suit :

```
#include <time.h>
```

Voici la liste des services de date/heure :

adjtime

ajuste l’heure pour synchroniser l’horloge système.

ctime, **localtime**, **gmtime**, **mktime**, **difftime**, **asctime**, **tzset**

convertit la date et l’heure selon la représentation de la chaîne.

getinterval, **incinterval**, **absinterval**, **resinc**, **resabs**, **alarm**, **ualarm**, **getitimer**, **setitimer**

gère l’heure d’expiration de plusieurs horloges.

gettimer, **settimer**, **restimer**, **stime**, **time**

recherche ou définit la valeur courante de l’horloge spécifiée à l’échelle du système.

gettimerid

affecte une horloge par processus.

gettimeofday, **settimeofday**, **ftime**

recherche et définit la date et l’heure.

nsleep, **usleep**, **sleep**

met un processus en veille.

reltimerid

libère une horloge affectée.

Support AIX pour la spécification X/Open UNIX95

Le système d'exploitation est conçu pour prendre en charge la spécification X/Open UNIX95 pour la portabilité des systèmes d'exploitation basés sur UNIX. Un certain nombre d'interfaces, dont certaines courantes, ont été ajoutées ou améliorées pour répondre à cette spécification. AIX est encore plus ouvert et portable pour les applications.

En outre, la compatibilité avec les versions antérieurs d'AIX a été maintenue, grâce à la création d'une nouvelle variable d'environnement, qui peut être utilisée pour définir l'environnement système par système, utilisateur ou processus.

L'environnement AIX par défaut préserve la compatibilité avec les versions antérieures d'AIX. Pour que l'environnent soit conforme à la spécification UNIX95, la valeur **ON** doit être affectée à la variable **XPG_SUS_ENV**. Si **XPG_SUS_ENV** est définie à une auter valeur, ou n'est pas définie, l'environnement par défaut d'AIX sera utilisé.

Lorsque **XPG_SUS_ENV** est définie, chaque programme de l'environnement fonctionne dans l'environnement du système d'exploitation spécifié par UNIX95. Il est possible que certaines applications compilées pour l'environnment AIX (peut-être pour une version antérieure d'AIX) ne fonctionnent pas correctement lorsque **XPG_SUS_ENV** est définie.

Mise hors service dynamique d’un processeur

A partir des types de serveurs 7044, Modèle 270, le hardware de tous les systèmes ayant plus de deux processeurs sera capable de détecter les erreurs corrigibles, rassemblées par les microprogrammes. Ces erreurs ne sont pas fatales et, tant qu’elles ne se produisent que rarement, celles–ci peuvent être ignorées sans risque. Toutefois, si une suite d’erreurs semble se développer sur processeur spécifique, cette combinaison signale éventuellement pour ce composant, un risque d’erreur fatale dans un futur proche. Le microprogramme effectue cette prévision à partir de l’analyse des seuils et des taux d’erreur.

AIX, sur ces systèmes, met en oeuvre une surveillance continue du matériel et interroge régulièrement le microprogramme sur la présence éventuelle d’erreurs matérielles. Lorsque le nombre d’erreurs de processeur atteint un seuil et que le microprogramme estime que ce composant de système présente un réel risque de défaillance, le microprogramme envoie un état d’erreur à AIX. Dans tous les cas, AIX consigne l’erreur dans le journal des erreurs. Qui plus est, dans les systèmes multiprocesseur, selon le type d’erreurs, AIX essaie de ne plus utiliser le processeur non fiable et de le mettre hors service. Cette fonction s’appelle *Mise hors service dynamique d’un processeur*.

A ce point, le microprogramme signale également que le processeur restera hors service pour tout réamorçage ultérieur, jusqu’à ce que le personnel de maintenance l’ait remplacé.

Impact éventuel sur les applications

Cette mise hors service du processeur est transparente pour la plupart des applications, dont les pilotes et extensions de noyau. Vous pouvez toutefois utiliser AIX pour savoir si une application ou une extension de noyau est exécutée sur un serveur multiprocesseur, pour connaître le nombre de processeurs et pour associer des threads à des processeurs spécifiques.

L’interface servant à associer les processus ou threads aux processeurs utilise des numéros de CPU. Ces numéros sont inclus dans la plage [0..*N*–1] où *N* correspond au nombre total de CPU. Afin de ne pas désactiver d’applications ou d’extensions de noyau ne supportant pas de “trou” dans la numérotation des CPU, AIX vérifie toujours que le CPU qui doit être mis hors service, apparaît comme le dernier de la liste (avec le numéro le plus élevé) aux applications. Par exemple, sur un SMP 8 processeurs, les numéros logiques de CPU seront [0..7]. Si un processeur est mis hors service, le nombre total de CPU disponible deviendra 7, et ils seront numérotés de [0..6]. De l’extérieur, le CPU 7 semble avoir disparu, quel que soit le processeur défaillant. Dans la suite de cette description, le terme “CPU” sera utilisé pour l’entité logique, et “processeur” pour l’entité physique.

Les applications ou extensions de noyau utilisant des liaisons de processus/threads, pourraient se retrouver désactivées si AIX mettait silencieusement fin à leurs threads associés ou les transférait autoritairement sur un autre CPU, dans le cas où un des processeurs serait mis hors service. AIX fournit des interfaces de programmation afin que ces applications et extensions de noyau puissent être prévenues en cas de mise hors service imminente d’un processeur. Lorsqu’elles recevront cet avertissement, elles devront faire en sorte que leur threads et ressources associés (tels que les blocs d’appel d’horloge) ne soient plus alloués au dernier CPU logique et s’adapter à la configuration du nouveau CPU.

Si, après avertissement des applications et extensions de noyau, certains des threads se trouvent toujours associés au dernier CPU logique, la mise hors service sera interrompue. Dans ce cas,AIX consignera dans le journal, le fait que la mise hors service a été interrompue et continuera d’utiliser le processeur défectueux. Lorsque le processeur cessera finalement de fonctionner, il se produira un blocage total du système. Il est donc essentiel que les applications ou extensions de noyau avec des liaisons aux CPU soient dûment averties de la mise hors service imminente d’un processeur et qu’elles réagissent en conséquence.

Dans les rares cas où une mise hors service ne peut aller jusqu'à son terme, AIX lancera malgré tout la procédure de préalerte aux administrateurs système. En consignant l'erreur dans le journal, ceux–ci pourront ainsi planifier une opération de maintenance sur le système afin de remplacer le composant défectueux avant que ne se produise un blocage total du système.

Mise hors service d'un processeur

Pour mettre un processeur hors service, procédez de la façon suivante :

1. Le microprogramme a détecté qu'un des processeurs a atteint le seuil d'une erreur récupérable.
2. AIX consigne l'état d'erreur du microprogramme dans le journal des erreurs du système, et – alors qu'il tourne sur une machine acceptant la mise hors service d'un processeur – lance le processus de mise hors service.
3. AIX signale les processus et threads hors noyau associés au dernier CPU logique.
4. AIX attend que tous les threads alloués soient dissociés du dernier CPU logique. Si les threads restent associés, AIX met fin au délai d'attente (après 10 minutes) et interrompt la mise hors service.
5. Sinon, AIX appelle les gestionnaires d'événements haute disponibilité précédemment consignés (HAEH). Un HAEH peut éventuellement envoyer une erreur qui mettra fin à la mise hors service.
6. Autre option : AIX poursuit le processus de mise hors service et arrête ultérieurement le processeur défectueux.

En cas d'échec à n'importe quel point de la mise hors service, AIX consigne l'erreur ainsi que la cause de l'interruption de la procédure. L'administrateur système peut consulter le journal des erreurs, prendre l'action corrective qui s'impose (si possible) et relancer la mise hors service. Par exemple, si celle–ci a été interrompue parce qu'au moins une application n'a pas dissocié ses threads alloués, l'administrateur système peut arrêter le(s) application(s), relancer le processus de mise hors service (qui devrait aller jusqu'au bout, cette fois–ci) et redémarrer l'application.

Administration de système

Activation/désactivation de la fonction de mise hors service du processeur

La fonction de mise hors service dynamique d'un processeur peut être activée ou désactivée en changeant la valeur de l'attribut **cpuguard** de l'objet ODM ***sys0***. Les valeurs acceptées pour l'attribut sont **enable** et **disable**.

L'option par défaut, dans cette version d'AIXest que la mise hors service dynamique d'un processeur est désactivée (l'attribut **cpuguard** a la valeur **disable**). Les administrateurs système qui veulent profiter de cette fonction doivent l'activer en utilisant, soit les menus du gestionnaire système basé sur le Web-based System Manager soit les menus d'environnement système SMIT, la commande **chdev** .

Remarque : Si la fonction de mise hors service d'un processeur est désactivée, AIX continuera de signaler les erreurs dans le journal des erreurs. Ainsi, vous verrez l'erreur indiquant qu'AIX a été informé de l'existence d'un problème sur un CPU (CPU_FAILURE_PREDICTED, voir format ci–après).

Relancement de la fonction de mise hors service d'un processeur

Il peut arriver que la mise hors service d'un processeur échoue parce que, par exemple, une application n'a pas dissocié ses threads alloués du dernier CPU logique. Une fois ce problème réglé, soit en dissociant (si c'est possible), soit en arrêtant l'application, l'administrateur système peut relancer la procédure de mise hors service du processus à l'aide de la commande **ha_star**.

La syntaxe de cette commande est la suivante :

```
ha_star –C
```

où –C correspond à la survenue d'une erreur probable de CPU.

Présentation de l'état d'un processeur

Les processeurs physiques sont représentés dans la base de données ODM par des objets appelés **proc*n***, où *n* est le numéro du processeur physique (*n* est un nombre décimal). Comme tout autre "dispositif" représenté dans la base de données ODM, les objets processeur sont dotés d'un état (Défini/Disponible) et d'attributs. L'état d'un objet **proc** est toujours "Disponible" tant que le processeur correspondant est présent, qu'il soit utilisable ou non par AIX. L'attribut **state** d'un objet **proc** indique si le processeur est utilisé par AIX et, si la réponse est négative, pour quelle raison. Cet attribut peut avoir trois valeurs :

enable

Le processeur est utilisé par AIX.

disable

Le processeur a été mis hors service dynamiquement par AIX.

faulty

Le processeur a été déclaré défectueux par le microprogramme lors de l'amorçage.

Dans le cas d'erreurs de CPU, si un processeur pour lequel le microprogramme a signalé une défaillance probable, a pu être correctement mis hors service par AIX, son état passera de "enable" à "disable". Indépendamment d'AIX, ce processeur est également signalé comme défectueux par le microprogramme. Au réamorçage, AIX ne pourra y accéder et son état sera déclaré défectueux. Toutefois, l'objet **proc** d'ODM affiche toujours la valeur "Available". La valeur de l'objet **proc** pourrait passer à "Defined" si, et uniquement si, le CPU défectueux avait été physiquement retiré de la carte système ou de la carte CPU (dans le cas où le retrait serait possible).

Exemples :

Le processeur **proc4** fonctionne correctement et il est utilisé par AIX :

```
# lsattr –EH –l proc4
attribute value description     user_settable

state   enable                  Processor state False
type    PowerPC_RS64–III        Processor type  False
#
```

Le processeur **proc4** est signalé comme potentiellement défectueux et il est mis hors service par AIX :

```
# lsattr –EH –l proc4
 attribute value   description  user_settable

 state  disable                 Processor state False
 type   PowerPC_RS64–III        Processor type  False
  #
```

Au réamorçage suivant, le processeur **proc4** est déclaré défectueux par le microprogramme et non disponible pour AIX :

```
# lsattr –EH –l proc4
 attribute value   description  user_settable

state   faulty                  Processor state False
type    PowerPC_RS64–III        Processor type  False
#
```

Or, dans ces trois cas, l'état du processeur **proc4** a conservé la valeur "Available" :

```
# lsdev –CH –l proc4
name    status          location        description

proc4   Available       00–04           Processor
#
```

Entrées du journal des erreurs

Voici des exemples accompagnés de descriptions d'entrées dans le journal des erreurs :

format court errpt – résumé

Trois messages différents du journal des erreurs sont associés à la mise hors service d'un CPU. Voici un exemple des entrées affichées par la commande **errpt** (sans options) :

```
# errpt
IDENTIFIER          TIMESTAMP          T      C
RESOURCE_NAME        DESCRIPTION
804E987A            1008161399         I      O      proc4
      CPU DEALLOCATED
8470267F            1008161299         T      S      proc4
      CPU DEALLOCATION ABORTED
1B963892            1008160299         P      H      proc4
      CPU FAILURE PREDICTED
#
```

– Si la mise hors service du processeur est activée, un message `CPU FAILURE PREDICTED` sera toujours suivi d'un message `CPU DEALLOCATED` ou d'un message`CPU DEALLOCATION ABORTED`.

– Si la mise hors service du processeur n'est pas activée, seul le message `CPU FAILURE PREDICTED` sera consigné. L'activation de la mise hors service du processeur faisant suite à la consignation d'un ou plusieurs messages `CPU FAILURE PREDICTED`, lance le processus de mise hors service et se traduit par l'enregistrement d'un message de succès ou d'échec dans le journal des erreurs, tel que décrit ci–dessus, pour chaque processeur déclaré défectueux.

format long errpt – description détaillée

Voici le type de résultat obtenu avec **errpt –a**:

– **CPU_FAIL_PREDICTED**

Description de l'erreur : défaillance probable d'un processeur

Cette erreur indique que le matériel a détecté qu'un processeur présente un risque élevé de défaillance dans un futur proche. Cette erreur est toujours consignée, que la mise hors service soit activée ou non.

DETAIL DATA: numèro du processeur physique, adresse

Exemple : entrée du journal des erreurs – format long

```
LABEL:          CPU_FAIL_PREDICTED
IDENTIFIER:     1655419A

Date/Time:      Thu Sep 30 13:42:11
Sequence Number:53
Machine Id:     00002F0E4C00
Node Id:        auntbea
Class:          H
Type:           PEND
Resource Name:  proc25
Resource Class: processor
Resource Type:  proc_rspc
Location:       00–25

Description
CPU FAILURE PREDICTED

Probable Causes
CPU FAILURE

Failure Causes
CPU FAILURE

Recommended Actions
ENSURE CPU GARD MODE IS ENABLED
RUN SYSTEM DIAGNOSTICS.

Detail Data
PROBLEM DATA
0144    1000    0000    003A    8E00    9100    1842
1100    1999    0930    4019
0000    0000    0000    0000    0000
0000    0000    0000    0000    0000    0000    0000
0000    4942    4D00    5531
2E31    2D50    312D    4332    0000
0002    0000    0000    0000    0000    0000    0000
0000    0000    0000    0000
0000    0000    0000    00000000
0000    0000    0000    0000    0000    0000    0000
0000    0000    0000    0000
0000    0000    0000    0000    0000
...     ...     ...     ...     ...
```

– CPU_DEALLOC_SUCCESS

Description de l’erreur : Un processeur a été mis hors service après détection d’une probable défaillance.

Ce message est consigné par AIX lorsque le mise hors service est activée et appliquée avec succès au CPU potentiellement défectueux.

DETAIL DATA: numéro logique du processeur mis hors service.

Exemple : entrée du journal des erreurs – format long

```
LABEL:          CPU_DEALLOC_SUCCESS
IDENTIFIER:     804E987A

Date/Time:      Thu Sep 30 13:44:13
Sequence Number:63
Machine Id:     00002F0E4C00
Node Id:        auntbea
Class:          O
Type:           INFO
Resource Name:  proc24

Description
CPU DEALLOCATED


        Recommended Actions
        MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE

Detail Data
LOGICAL DEALLOCATED CPU NUMBER

        0
```

L'exemple précédent montre que **proc24** a été mis hors service et qu'il s'agissait du CPU logique **0** lorsque la défaillance s'est produite.

– CPU_DEALLOC_FAIL

Description de l'erreur : Echec de la mise hors service d'un processeur après détection d'un risque de la défaillance de celui–ci.

Ce message est consigné par AIX lorsque la mise hors service est activée et appliquée sans succès au CPU potentiellement défectueux.

DETAIL DATA: Code de la cause d'échec, numéro logique de CPU, informations supplémentaires selon le type de défaillance.

Le code de la cause d'échec est une valeur hexadécimale numérique. Voici la liste des codes possibles :

2	Un ou plusieurs processus/threads restent associés au dernier CPU logique. Dans ce cas, les données détaillées donnent les PID des processus en cause.
3	Un pilote enregistré ou une extension de noyau a envoyé un message d'erreur lorsqu'il (elle) en a été averti(e). Dans ce cas, le champ de données détaillées contient le nom du pilote en cause ou de l'extension de noyau (en code ASCII).

4 La mise hors service d'un processeur signifierait que le système disposerait de moins de deux CPU. AIX ne met jamais plus de *N*–2 processeurs hors service sur un système avec *N* processeurs pour éviter d'induire en erreur des applications ou extensions de noyau utilisant le nombre total de processeurs disponibles, afin de déterminer s'ils tournent sur un système uniprocesseur (UP) – où il est recommandé de ne pas utiliser de verrous multiprocesseur – ou sur un SMP (Symmetric Multi Processor).

200 (0xC8) La mise hors service d'un processeur est désactivée (l'attribut ODM **cpuguard** a la valeur **disable**). Cette erreur ne devrait normalement pas se produire à moins d'avoir lancé **ha_star** manuellement.

Exemples : entrées de journal des erreurs – format long

Exemple 1 :

```
LABEL:          CPU_DEALLOC_ABORTED
IDENTIFIER:     8470267F
Date/Time:      Thu Sep 30 13:41:10
Sequence Number:50
Machine Id:     00002F0E4C00
Node Id:        auntbea
Class:          S
Type:           TEMP
Resource Name:  proc26

Description
CPU DEALLOCATION ABORTED

Probable Causes
SOFTWARE PROGRAM

Failure Causes
SOFTWARE PROGRAM

Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE
SEE USER DOCUMENTATION FOR CPU GARD

Detail Data
DEALLOCATION ABORTED CAUSE
0000 0003
DEALLOCATION ABORTED DATA
6676 6861 6568 3200
```

L'exemple précédent montre l'échec de la mise hors service de **proc26**. Le code de la cause d'échec **3** signifie que l'extension de noyau a renvoyé une erreur à la routine de notification du noyau. Ci–dessus, le code spécifié sous `DEALLOCATION ABORTED DATA` correspond à **fvhaeh2**, qui est le nom de l'extension utilisée lors de l'enregistrement avec le noyau.

Exemple 2 :

```
LABEL:          CPU_DEALLOC_ABORTED
IDENTIFIER:     8470267F
Date/Time:      Thu Sep 30 14:00:22
Sequence Number:71
Machine Id:     00002F0E4C00
Node Id:        auntbea
Class:          S
Type:           TEMP
Resource Name:  proc19

Description
CPU DEALLOCATION ABORTED

Probable Causes
SOFTWARE PROGRAM

Failure Causes
SOFTWARE PROGRAM

Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE;
SEE USER DOCUMENTATION FOR CPU GARD

Detail Data
DEALLOCATION ABORTED CAUSE
0000 0002
DEALLOCATION ABORTED DATA
0000 0000 0000 4F4A
```

L'exemple précédent montre l'échec de la mise hors service de **proc19**. Le code de la cause d'échec **2** signifie que des threads étaient associés au dernier processeur logique et qu'ils le sont restés à la réception du signal `SIGCPUFAIL`. Le message `DEALLOCATION ABORTED DATA` montre que ces threads appartenaient au processus **0x4F4A**.

Les options de la commande **ps** (`-o THREAD, -o BND`) permettent le listage de l'ensemble des threads ou processus avec le numéro du CPU auquel ils sont associés lorsque cela s'avère pertinent.

Exemple 3 :

```
LABEL:          CPU_DEALLOC_ABORTED
IDENTIFIER:     8470267F

Date/Time:      Thu Sep 30 14:37:34
Sequence Number:106
Machine Id:     00002F0E4C00
Node Id:        auntbea
Class:          S
Type:           TEMP
Resource Name:  proc2

Description
CPU DEALLOCATION ABORTED

Probable Causes
SOFTWARE PROGRAM

Failure Causes
SOFTWARE PROGRAM

Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE
SEE USER DOCUMENTATION FOR CPU GARD

Detail Data
DEALLOCATION ABORTED CAUSE
0000 0004
DEALLOCATION ABORTED DATA
0000 0000 0000 0000
```

L'exemple précédent montre que la mise hors service de **proc2** a échoué parce qu'il n'y avait que deux processeurs activés (ou même moins) au moment de la défaillance (code de la cause d'échec **4**).

Chapitre 11. NLS

Un grand nombre de variables définissent l’environnement de la langue du système. Elles sont regroupées avec les commandes, fichiers et autres outils les prenant en charge dans un programme appelé National Language Support (NLS).

Les sujets abordés sont les suivants :

NLS - généralités

NLS fournit des commandes et des sous-routines de la bibliothèque C standard prenant en charge une base système unique à l'échelon mondial. Aucune hypothèse ou dépendance n'est intégrée à un système internationalisé quant aux conventions spécifiques d'une langue ou d'une culture telles que :

- les jeux de codes,
- la classification des caractères,
- les règles de comparaison des caractères,
- l'interclassement des caractères,
- les formats numériques et monétaires,
- les formats des dates et heures,
- la langue des textes et messages.

C'est lors de l'exécution du processus que le système a accès à l'ensemble des données relatives aux conventions culturelles et aux langues.

NLS fournit les fonctions ci-dessous pour la prise en charge du système en environnement international :

- localisation des données,
- séparation entre messages et programmes,
- conversion entre jeux de codes.

Localisation des données

Un système internationalisé est en mesure de traiter correctement des données pour différentes sources géographiques. Par exemple, une date au format 9/6/1995 est interprétée aux Etats-Unis comme le 6 septembre 1995. Au Royaume-Uni la même date sera interprétée comme le 9 juin 1995. Le format numérique et monétaire est également spécifique de chaque pays par exemple, le dollar US et la livre britannique. Pour traiter l'information, les conventions spécifiques des langues et des cultures définissent un environnement local.

Les programmes doivent avoir accès à l'environnement local au moment de l'exécution pour pouvoir traiter et afficher les données conformément aux conventions culturelles et à la langue voulue. Ce processus s'appelle localisation. Il consiste à développer une base de données où figurent les règles spécifiques de l'environnement local régissant les formats des données et une interface d'accès à ces règles. Pour en savoir plus sur la localisation, reportez-vous à "Environnement local - généralités", page 11-4.

Séparation entre messages et programmes

Pour faciliter la traduction multilingue des messages et permettre l'accès du programme aux messages traduits sur la base d'un environnement local utilisateur, il est nécessaire de séparer les messages des programmes et de les mettre à disposition de ces programmes au moment de l'exécution, sous la forme de catalogues de messages. C'est le composant Message Facility de NLS qui fournit les commandes et sous-routines dédiées à cette tâche. Pour en savoir plus, reportez-vous à "Message Facility - généralités", dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*

Conversion entre jeux de codes

Un *caractère* est un symbole servant à structurer, contrôler ou représenter une donnée. Ces symboles, regroupés pour décrire une langue donnée, forment un *jeu de caractères*. Un jeu de codes contient des valeurs de codage d'un jeu de caractères. Ce sont ces valeurs qui fournissent l'interface entre le système et ses unités d'entrée et de sortie.

Par le passé, les efforts ont été concentrés sur le codage de l'alphabet anglais. On utilisait une méthode sur 7 bits, suffisante du fait du petit nombre de caractères formant cet alphabet. Pour prendre en charge des alphabets plus grands, des langues asiatiques par exemple (le chinois, le japonais et le coréen), d'autres jeux de caractères ont été développés intégrant des codages multi-octets.

Les jeux de codes pris en charge sont répertoriés ci-après.

- La famille de jeux de codes ISO8859 fournit une variété de jeux de codes mono-octet conformes aux normes de l'industrie : Latin-1, Latin-2, Arabic, Cyrillic, Hebrew, Greek, et Turkish. Le jeu de codes IBM-eucJP, également conforme aux normes de l'industrie, prend le japonais en charge.
- Sont aussi pris en charge les jeux de codes sur plate-forme PC (ordinateurs personnels) IBM-850, et IBM-943 (et IBM-932). IBM-850 (mono-octet) prend en charge les langues du groupe de pays Latin-1 (Etats-Unis, Canada et Europe de l'Ouest). IBM-943 et IBM-932 (multi-octets) prennent en charge le japonais
- Les environnement Unicode (TM) fonctionnnant avec le jeu de codes UTF–8 sont pris en charge pour toutes les langues et tous les pays pris en charge. UTF–8 permet la prise en charge des caractères de la plupart des principales langues du monde et peut être utilisé dans des environnement dans lesquels plusieurs langues sont traitées en même temps.

Plus la prise en charge des jeux de codes est étendue, plus il est important de ne pas encombrer les programmes avec les connaissances relatives à un jeu de codes spécifique, pour que ces programmes n'en soient pas dépendants. C'est ce qu'on appelle *l'indépendance par rapport au jeu de codes*. Pour conférer au système l'indépendance par rapport aux jeux de codes, NLS fournit des convertisseurs qui traduisent les valeurs de codage des caractères trouvés dans les différents jeux de codes. Ces convertisseurs permettent au système de traiter avec exactitude les données générées dans des environnements aux jeux de codes multiples. Pour en savoir plus, reportez-vous à "Convertisseurs - généralités", page 11-16.

Environnement local - généralités

Aucune hypothèse ou dépendance relative aux jeux de codes, à la classification des caractères, aux règles de comparaison des caractères, à l'interclassement des caractères, aux formats monétaires, à la ponctuation numérique, aux formats des dates et heures, ou aux textes des messages n'est intégrée à un système internationalisé. Un *environnement local* est défini par des conventions culturelles et des langues. C'est à l'exécution du processus que le système a accès à l'ensemble des données relatives à ces conventions.

Les programmes doivent avoir accès à l'environnement local au moment de l'exécution pour pouvoir traiter et afficher les données conformément aux conventions spécifiques de votre langue et de votre culture. NLS fournit cet environnement local. NLS fournit une base de données contenant les règles spécifiques de l'environnement local régissant le formatage des données et une interface d'accès à ces règles.

Description de l’environnement local

L’environnement local est formé de la combinaison de la langue, du pays et du jeu de codes identifiant un jeu de conventions sur la langue concernée. Ces conventions sont relatives à l’interclassement, la conversion en majuscules/minuscules, la classification des caractères, la langue des catalogues de messages, la représentation de la date et de l’heure, le symbole monétaire et la représentation numérique.

Les données figurant dans les fichiers source de définition de l’environnement local doivent être tout d’abord converties en une base de données d’environnement local avec la commande **localedef**. Ensuite, la sous-routine **setlocale** peut avoir accès à ces données et définir l’environnement local pour les applications. Les données d’environnement local sont divisées en six catégories. Chaque catégorie est spécialisée dans un aspect particulier de l’environnement. Par le biais des variables d’environnement **LC_*** et **LANG**, vous pouvez spécifier l’environnement local voulu.

Conventions d’appellation

L’environnement local porte le nom de son fichier source de définition. Ce dernier est désigné par la combinaison de la langue, du pays et du jeu de codes qu’il décrit. Le format utilisé est le suivant :

```
language[_territory][.codeset][@modifier]
```

Par exemple, l’environnement local pour la langue danoise parlée au Danemark à l’aide du jeu de codes ISO8859–1 est `da_DK.ISO8859-1`. `da` correspond à danois, et `DK` à Danemark. Pour désigner cet environnement local, la forme abrégée `da_DK` est suffisante. En effet, l’unique autre environnement local de même langue et même pays, mais utilisant un autre jeu de code s’appelle `Da_DK.IBM-850` ou, en abrégé, `Da_DK`.

Les fichiers de définition des environnements locaux définis par le système fournissent le format des catégories d’environnement et leurs mots-clés. Ils sont situés dans le répertoire **/usr/lib/nls/loc**. L’environnement local C ou POSIX définit l’environnement local standard ANSI en C hérité par tous les processus au moment du démarrage. Voici les autres fichiers source de définition des environnements locaux définis par le système :

Environnement local	Langue	Pays	Jeu de codes
Ar_AA	arabe	Pays arabes	IBM–1046
ar_AA	arabe	Pays arabes	IS08859–6
be_BY	biélorusse	Biélorussie	ISO8859–5
bg_BG	bulgare	Bulgarie	IS08859–5
ca_ES	catalan	Espagne	IS08859–15
cs_CZ	tchèque	République tchèque	IS08859–2
Da_DK	danois	Danemark	IBM–850
da_DK	danois	Danemark	ISO8859–1
da_DK	danois	Danemark	ISO8859–15
De_CH	allemand	Suisse	IBM–850
de_CH	allemand	Suisse	ISO8859–1
de_CH	allemand	Suisse	ISO8859–15
De_DE	allemand	Allemagne	IBM–850
de_DE	allemand	Allemagne	ISO8859–1
de_DE	allemand	Allemagne	ISO8859–15

Environnement local	Langue	Pays	Jeu de codes
el_GR	grec	Grèce	ISO8859–7
En_AU	anglais	Australie	ISO8859–15
En_BE	anglais	Belgique	ISO8859–15
En_GB	anglais	Grande–Bretagne	IBM–850
en_GB	anglais	Grande–Bretagne	ISO8859–1
en_GB	anglais	Grande–Bretagne	ISO8859–15
En_US	anglais	Etats–Unis	IBM–850
en_US	anglais	Etats–Unis	ISO8859–1
en_US	anglais	Etats–Unis	ISO8859–15
en_ZA	anglais	Afrique du sud	ISO8859–15
Es_ES	espagnol	Espagne	IBM–850
es_ES	espagnol	Espagne	ISO8859–1
es_ES	espagnol	Espagne	ISO8859–15
Et_EE	estonien	Estonie	IBM–922
ET_EE	estonien	Estonie	UTF–8
Fi_FI	finnois	Finlande	IBM–850
fi_FI	finnois	Finlande	ISO8859–1
fi_FI	finnois	Finlande	ISO8859–15
Fr_BE	français	Belgique	IBM–850
fr_BE	français	Belgique	ISO8859–1
fr_BE	français	Belgique	ISO8859–15
Fr_CA	français	Canada	IBM–850
fr_CA	français	Canada	ISO8859–1
fr_CA	français	Canada	ISO8859–15
Fr_FR	français	France	IBM–850
fr_FR	français	France	ISO8859–1
fr_FR	français	France	ISO8859–15
Fr_CH	français	Suisse	IBM–850
fr_CH	français	Suisse	ISO8859–1
fr_CH	français	Suisse	ISO8859–15
hr_HR	croate	Croatie	ISO8859–2
hu_HU	hongrois	Hongrie	ISO8859–2
Is_IS	islandais	Islande	IBM–850
is_IS	islandais	Islande	ISO8859–1
It_CH	italien	Suisse	ISO8859–15
It_IT	italien	Italie	IBM–850
it_IT	italien	Italie	ISO8859–1
it_IT	italien	Italie	ISO8859–15
Iw_IL	hébreu	Israël	IBM–856

Environnement local	Langue	Pays	Jeu de codes
iw_IL	hébreu	Israël	ISO8859–8
Ja_JP	japonais	Japon	IBM–943
ja_JP	japonais	Japon	IBM-eucJP
ko_KR	coréen	Corée	IBM–eucKR
Lt_LT	lituanien	Lituanie	IBM–921
LT_LT	lituanien	Lituanie	UTF–8
Lv_LV	letton	Lettonie	IBM–921
LV_LV	letton	Lettonie	UTF–8
mk_MK	macédonien	Ex–république yougoslave de Macédoine	ISO–8859–5
Nl_BE	flamand	Belgique	IBM–850
nl_BE	flamand	Belgique	ISO8859–1
nl_BE	flamand	Belgique	ISO8859–15
Nl_NL	flamand	Pays–Bas	IBM–850
nl_NL	flamand	Pays–Bas	ISO8859–1
nl_NL	flamand	Pays–Bas	ISO8859–15
No_NO	norvégien	Norvège	IBM–850
no_NO	norvégien	Norvège	ISO8859–1
no_NO	norvégien	Norvège	ISO8859–15
pl_PL	polonais	Pologne	ISO8859–2
pt_BR	portugais	Brésil	ISO8859–1
pt_BR	portugais	Brésil	ISO8859–15
Pt_PT	portugais	Portugal	IBM–850
pt_PT	portugais	Portugal	ISO8859–1
pt_PT	portugais	Portugal	ISO8859–15
ro_RO	roumain	Roumanie	ISO8859–2
ru_HU	russe	Russie	ISO8859–5
sh_SP	serbe (latin)	Yougoslavie	ISO8859–2
sl_SI	slovène	Slovénie	ISO8859–2
sk_SK	slovaque	Slovaquie	ISO8859–2
sq_AL	albanais	Albanie	ISO8859–1
sq_AL	albanais	Albanie	ISO8859–15
sr_SP	serbe (cyrillique)	Yougoslavie	ISO8859–5
Sv_SE	suédois	Suède	IBM–850
sv_SE	suédois	Suède	ISO8859–1
sv_SE	suédois	Suède	ISO8859–15
th_TH	thaïlandais	Thaïlande	TIS–620
TH_TH	thaïlandais	Thaïlande	UTF–8
tr_TR	turc	Turquie	ISO8859–9

Environnement local	Langue	Pays	Jeu de codes
Uk_UA	ukrainien	Ukraine	IBM–1124
Vi_VN	vietnamien	Vietnam	IBM–1129
VI_VN	vietnamien	Vietnam	UTF–8
Zh_CN	chinois simplifié	République Populaire de Chine	GBK
zh_CN	chinois simplifié	République Populaire de Chine	IBM–eucCN
ZH_CN	chinois	République Populaire de Chine	UTF–8
zh_TW	chinois (trad)	République de Chine	IBM–eucTW
Zh_TW	chinois (trad)	République de Chine	big5

Environnement local par défaut à l'installation

Il s'agit de l'environnement local sélectionné à l'installation. Par exemple, à l'invite, pendant le processus d'installation, vous pouvez spécifier la langue française parlée au Canada. Le jeu de codes passe automatiquement à la valeur par défaut ISO8859-1. Avec cette information, le système définit la valeur de l'environnement local par défaut, spécifiée par la variable **LANG**, à `fr_CA` (`fr` pour la langue française qui correspond au jeu de codes ISO8859-1 et `CA` pour Canada). Tous les processus utilisent l'environnement local par défaut tant que les variables **LC_*** ou **LANG** ne sont pas modifiées. Vous pouvez changer l'environnement local par défaut avec le menu SMITManage Language Environment.

Description des catégories d'environnement local

Une *catégorie* d'environnement local est un regroupement particulier de données spécifiques de la langue et des conventions culturelles. Par exemple, le format de la date et de l'heure, les noms des mois et des jours de la semaine, et d'autres informations relatives à l'heure, les noms des mois et des jours de la semaine, et d'autres informations relatives à l'heure, sont regroupées dans la catégorie **LC_TIME**. Chaque catégorie utilise un jeu de mots-clé décrivant les particularités de ce sous-ensemble de l'environnement local.

Les catégories standard suivantes peuvent être définies dans un fichier source de définition d'environnement local :

LC_COLLATE	Définit l'interclassement de caractères ou de chaînes.
LC_CTYPE	Définit la classification des caractères, la conversion des majuscules/minuscules et d'autres attributs de caractères.
LC_MESSAGES	Définit le format des réponses affirmatives et négatives.
LC_MONETARY	Définit les règles et symboles relatifs au format monétaire numérique.
LC_NUMERIC	Définit les règles et symboles relatifs au format numérique non monétaire.
LC_TIME	Définit une liste de règles et symboles relatifs au format de la date et de l'heure.

Remarque : Les catégories d'environnement local ne peuvent être modifiées qu'en éditant le fichier source de définition d'environnement local. Il ne faut pas les confondre avec les variables d'environnement de même nom qui peuvent être définies à partir de la ligne de commande.

Description des variables d'environnement local

NLS utilise plusieurs variables qui influencent la sélection des environnements locaux. Vous pouvez définir les valeurs de ces variables pour modifier les chemins d'accès aux données d'environnement local :

LANG	Spécifie l'environnement local par défaut à l'installation. **Remarque :** La valeur de **LANG** est établie à l'installation. (Elle correspond à l'environnement local de chaque processus, excepté si les variables d'environnement **LC_*** sont définies). **LANG** peut être modifiée avec SMIT. C et POSIX sont les environnements locaux offrant les meilleures performances.
LC_ALL	Remplace la valeur de **LANG** et les valeurs de toute autre variable **LC_***.
LC_COLLATE	Spécifie l'environnement local à utiliser pour les données de la catégorie **LC_COLLATE**. Cette catégorie détermine les règles d'interclassement des caractères et des chaînes régissant le comportement des plages, des classes d'équivalence et des éléments d'interclassement multi-caractère.
LC_CTYPE	Spécifie l'environnement local à utiliser pour les données de la catégorie **LC_CTYPE**. Cette catégorie détermine les règles de manipulation des caractères régissant l'interprétation de séquences d'octets de caractères de texte (c'est-à-dire mono-octet contre multi-octets), la classification des caractères (par exemple, alphabétiques, numériques, etc.) et le comportement des classes de caractères.
LC__FASTMSG	Spécifie que les messages par défaut servent aux environnements locaux C et POSIX et que **NLSPATH** n'est pas pris en compte quand **LC__FASTMSG** est défini à `true` ; La valeur par défaut sera `LC__FASTMSG=true` dans **/etc/environment**.
LC_MESSAGES	Spécifie l'environnement local à utiliser pour les données de la catégorie **LC_MESSAGES**. Cette catégorie détermine les règles régissant les réponses affirmatives et négatives et la langue des messages et des menus.
LC_MONETARY	Spécifie l'environnement local à utiliser pour les données de la catégorie **LC_MONETARY**. Cette catégorie détermine les règles régissant le format monétaire.
LC_NUMERIC	Spécifie l'environnement local à utiliser pour les données de la catégorie **LC_NUMERIC**. Cette catégorie détermine les règles régissant le format numérique non monétaire.
LC_TIME	Spécifie l'environnement local à utiliser pour les données de la catégorie **LC_TIME**. Cette catégorie détermine les règles régissant le format de la date et de l'heure.
LOCPATH	Spécifie le chemin d'accès aux données localisées, y compris les fichiers d'environnement local binaire, les méthodes d'entrée et les convertisseurs de jeu de codes. **Remarque :** Tous les programmes **setuid** et **setgid** ignorent la variable **LOCPATH**.
NLSPATH	Spécifie le chemin d'accès aux fichiers de catalogues de messages. Cette variable sert au composant Message Facility du sous-système NLS. Reportez-vous à la routine **catopen** pour plus d'informations sur le format de la variable **NLSPATH**.

Les variables affectant la sélection de l'environnement local peuvent être classées en trois niveaux de priorité, comme suit :

Hiérarchie des variables d'environnement local	
Niveau de priorité	**Variables d'environnement**
Elevé	**LC_ALL**
	LC_COLLATE
	LC_CTYPE
Moyen	**LC_MESSAGES**
	LC_MONETARY
	LC_NUMERIC
	LC_TIME
Faible	**LANG**

Les variables d'environnement affectent le comportement d'un programme internationalisé de la façon suivante :

- Quand la variable **LC_ALL** est définie, sa valeur est utilisée par toutes les catégories. Par exemple, avec une valeur de **LC_ALL** égale à `en_US` et de **LANG** égale à `fr_FR`, l'environnement local est défini à `en_US`.
- Quand la variable **LC_ALL** n'est pas définie, les valeurs prises en compte sont celles des variables de niveau de priorité moyen. Par exemple, avec une valeur de **LANG** égale à `en_US` et de **LC_TIME** égale à `fr_FR`, la catégorie qui sera chargée de la base de données de l'environnement local `fr_FR` est **LC_TIME**. **LC_TIME** n'affecte pas le comportement des autres catégories.
- En l'absence de définition de variables **LC_*** individuelles, la valeur de **LANG** désigne l'environnement local de toutes les autres catégories.
- Quand la variable **LANG** n'est pas définie, c'est l'environnement local par défaut C qui est pris en compte pour toutes les autres catégories.

Description du fichier source de définition d’environnement local

A la différence des variables d’environnement, que vous pouvez définir depuis la ligne de commande, vous ne pouvez modifier les environnements locaux qu’en éditant et compilant le fichier source de définition d’environnement local correspondant.

Quand un environnement local voulu ne fait pas partie de la bibliothèque, il est possible d’en compiler une version binaire avec la commande **localedef**. Le comportement de l’environnement local des programmes n’est pas affecté par un fichier source de définition d’environnement local, excepté quand le fichier est converti, en un premier temps, avec la commande **localedef** et que l’objet environnement local est mis à disposition du programme. La commande **localedef** convertit dans un format d’exécution les fichiers source contenant des définitions d’environnements locaux et copie la version d’exécution dans le fichier désigné sur la ligne de commande, qui est généralement un nom d’environnement local. Les commandes et sous-routines internationalisées ont ensuite accès aux données de l’environnement local. Pour en savoir plus sur la préparation des fichiers source à convertir avec la commande **localedef**, reportez-vous à la section relative à leur format dans *AIX Files Reference*.

Description du fichier source charmap

Avec le fichier source charmap, qui est un fichier de description du jeu de caractères, vous pouvez affecter des noms symboliques aux codages des caractères.

Les personnes qui élaborent les fichiers source charmap sont libres de choisir leurs propres noms symboliques, à condition que ces noms ne soient pas en conflit avec les noms symboliques standard décrivant le jeu de caractères portable.

Le fichier charmap résoud les problèmes de portabilité des sources, et particulièrement les sources de définition d'environnements locaux. Le jeu de caractères portable standard est constant dans tous les environnements locaux. Le fichier charmap permet d'élaborer une définition d'environnement local commune à plusieurs jeux de codes. Ainsi, la même source de définition d'environnement local sert à des jeux de codes pour lesquels il existe plusieurs codages des mêmes caractères étendus.

Un fichier charmap définit un jeu de symboles servant de référence pour les codages de caractères au fichier source de définition d'environnement local. Les caractères du jeu portable peuvent être ajoutés au fichier charmap, à condition que leurs codages et leurs codages par défaut soient identiques.

Les fichiers charmap résident dans le répertoire **/usr/lib/nls/charmap**.

Modification de l’environnement local

Modification de l’environnement NLS

Dans le cadre de la modification de l’environnement NLS, vous pouvez utiliser l’application Web-based System Manager Users ou l’interface SMIT pour :

- modifier l’environnement de la langue par défaut,
- modifier la mappe du clavier au redémarrage suivant du système,
- gérer les polices,
- convertir le jeu de codes des catalogues de messages,
- convertir le jeu de codes des fichiers texte à plat (sans structure hiérarchique),

En outre, avec la commande **setmaps**, vous pouvez définir la mappe du jeu de codes d’un terminal.

Modification de l’environnement de la langue par défaut

La définition de la variable **LANG** (chaîne **”LANG = <nom>”** dans le fichier **/etc/environment**) désigne l’environnement local par défaut (combinaison langue/pays/jeu de codes). Cet environnement fournit les formats par défaut d’interclassement, de classification des caractères, de conversion des majuscules et minuscules, des données monétaires et numériques, de la date et de l’heure, et des réponses affirmatives ou négatives. En outre, l’environnement local par défaut fait référence au jeu de codes.

Modification des mappes de clavier par défaut au redémarrage suivant du système

Si plusieurs jeux de codes sont pris en charge pour une combinaison donnée langue/pays, il existe plusieurs mappes de clavier LFT. La mappe de clavier sélectionnée doit correspondre au jeu de codes de l’environnement de la langue sélectionnée.

Gestion des polices

L’utilisateur peut sélectionner la police active ou la police à charger au prochain redémarrage du système. La police sélectionnée doit prendre en charge le même jeu de codes que l’environnement de la langue sélectionnée et que la mappe de clavier LFT.

Conversion du jeu de codes des catalogues de messages

Les catalogues de messages sont livrés en un jeu de codes par combinaison de langue/pays traduite. Ce jeu doit correspondre à celui de l’environnement local.

Conversion du jeu de codes des fichiers texte à plat

Un jeu de codes d’un fichier à plat défini par l’utilisateur peut être converti au besoin en un autre jeu de codes (IBM-850 en ISO8859-1, par exemple).

Scénarios utilisateur

Voici plusieurs scénarios associés à NLS que l’utilisateur peut rencontrer sur le système. Ces scénarios sont assortis des interventions suggérées.

- L’utilisateur conserve le jeu de codes par défaut

  Le jeu de codes par défaut vous convient, en ce qui concerne la combinaison langue/pays, même si plusieurs versions de cette combinaison sont prises en charge. Dans la mesure où l’environnement utilisateur en cours exploite ce jeu de codes, vous pouvez le conserver.

  La combinaison langue/pays sélectionnée lors de l’installation du système sera celle par défaut dans l’environnement local approprié, sur la base du jeu de codes par défaut. Les mappes de clavier par défaut, les polices par défaut et les catalogues de messages dépendent tous du jeu de codes par défaut. Dans ce scénario, vous n’avez pas à intervenir.

- L’utilisateur change le jeu de codes par défaut

  Vous êtes dans un environnement local Latin-1 (ou Japanese) et souhaitez migrer vos données et l’environnement NLS vers un jeu de codes autre que celui par défaut. Procédez comme suit :

  – Avec des données existantes à convertir

    La conversion des fichiers texte à plat à convertir vers le jeu de codes souhaité se fait avec l’utilitaire **iconv**, avec l’application Web-based System Manager Users ou avec le menu SMIT Manage the Language Environment. Les fichiers structurés définis par l’utisateur doivent être convertis avec des outils de conversion développés par l’utilisateur, dotés des fonctions bibliothèque **iconv** (pour la conversion des zones de texte voulues en fichiers structurés).

    – Pour passer à un autre jeu de codes

      Si, pour une combinaison langue/pays donnée, plusieurs jeux de codes sont pris en charge, vous pouvez passer à un environnement local autre que celui par défaut.

      – Application Web-based System Manager Users
      – Menu SMIT Manage Language Environment
      – Commandes **chlang**, **chkbd** et **chfont**

Convertisseurs - généralités

NLS fournit une base pour l'internationalisation des données qui peuvent ainsi passer d'un jeu de codes à l'autre. Vous pouvez avoir besoin de convertir des fichiers texte ou des catalogues de messages. A cet effet, il est fait appel à un certain nombre de convertisseurs standard.

Quand un programme envoie des données à un autre programme résidant sur un hôte distant dont le jeu de codes est différent, ces données doivent être converties. Par exemple, pour la communication avec un système IBM VM, le système convertit les données ISO8859-1 en données EBCDIC. Les jeux de codes définissent les caractères et contrôlent les affectations de fonction de contrôle et de caractères en points de code. Ces caractères codés doivent être convertis dès lors qu'un programme reçoit des données dans un jeu de codes et les affiche dans un autre jeu de codes.

Il existe deux interfaces de conversion :

- iconv (commande)
  permettant de demander une conversion en indiquant le nom du jeu de codes à convertir et celui du jeu de codes cible.
- libiconv functions
  permettant aux applications de demander un convertisseur en indiquant son nom.

Le système fournit une bibliothèque de convertisseurs prête à l'emploi. Il suffit d'indiquer le nom du convertisseur voulu. Ces bibliothèques résident dans les répertoires suivants : **/usr/lib/nls/loc/iconv/*** et **/usr/lib/nls/loc/iconvTable/***.

En plus des convertisseurs de jeux de codes, la bibliothèque fournit un ensemble de convertisseurs d'échange de réseau. Dans un environnement de réseau, la méthode de conversion dépend des jeux de codes des systèmes de communication et des protocoles utilisés.

Les convertisseurs d'échange servent à convertir des données transmises d'un système à un autre. Les conversions effectuées entre jeux de codes internes font appel aux convertisseurs de jeux de codes. Qu'une conversion des données soit à effectuer entre le jeu de codes de l'expéditeur et celui du destinataire ou du format 8 bits en format 7 bits, une interface uniforme est indispensable. Les sous-routines **iconv** fournissent cette interface.

Convertisseurs standard

Les convertisseurs standard s'exploitent avec la commande et les sous-routines **iconv**. La liste suivante décrit les différents types de convertisseurs.

Pour la liste des convertisseurs, reportez-vous au document *AIX General Programming Concepts : Writing and Debugging Programs*.

Types de convertisseur

- Convertisseur de table
  Convertit les jeux de codes mono-octets sans état. Traduit la table d'un octet en un autre octet.
- Convertisseur multi-octets
  Fournit les conversions en jeux de codes multi-octets, par exemple, entre le japonais sur plate-forme PC (IBM-943 et IBM-932) ou le japonais sous AIX (IBM-eucJP) et le japonais sur hôte IBM (IBM-930 et IBM-939).

Types de convertisseurs d'échange

- convertisseur 7 bits
  Fait la conversion entre les jeux de codes internes et les formats d'échange standard (7 bits).
- convertisseur 8 bits
  Fait la conversion entre les jeux de codes internes et les formats d'échange standard (8 bits).
- Convertisseur de texte composé
  Convertit le texte composé en jeux de codes internes.
- convertisseur d'uucodeFournit les mêmes mappes que les commandes **uuencode** et **uudecode**.
- Convertisseurs divers
  Utilisés par certains convertisseurs cités plus haut.

Description des bibliothèques iconv

L'utilitaire iconv consiste en un ensemble de fonctions intégrant les données et la partie logique à convertir entre différents jeux de codes. Il intègre également la commande **iconv** de conversion de données. Un seul système peut posséder plusieurs convertisseurs. La variable d'environnement **LOCPATH** définit le convertisseur qu'exploitent les sous-routines **iconv**.

Remarque : Tous les programmes **setuid** et **setgid** ignorent la variable **LOCPATH**.

Convertisseur universel UCS

UCS-2 est un codage universel 16 bits (se reporter aux généralités sur les jeux de codes dans le document *AIX General Programming Concepts : Writing and Debugging Programs*) exploitable comme support d'échange pour fournir des fonctions de conversion compatibles avec presque tous les jeux de codes. Le convertisseur universel UCS peut se charger de la conversion entre tout jeu de codes XXX et YYY comme suit :

```
XXX >–> UCS-2 <–> YYY
```

Il faut que les conversions de XXX et YYY figurent dans la liste supportée des convertisseurs d'échange UCS-2 et soient installées sur le système.

Le convertisseur universel est installé comme le fichier **/usr/lib/nls/loc/iconv/Universal_UCS_Conv**. De nouvelles conversions peuvent être prises en charge en créant de nouveaux liens avec des noms appropriés dans le répertoire **/usr/lib/nls/loc/iconv**. Par exemple, pour prendre en charge de nouveaux convertisseurs d'IBM-850 en IBM-437, exécutez les commandes :

```
ln –s /usr/lib/nls/loc/iconv/Universal_UCS_Conv
/usr/lib/nls/loc/iconv/IBM–850_IBM–437

ln –s /usr/lib/nls/loc/iconv/Universal_UCS_Conv
/usr/lib/nls/loc/iconv/IBM–437_IBM–850
```

Attention : Un lien de convertisseur créé pour des jeux de codes incompatibles (par exemple, ISO8859-1 et IBM-eucJP), avec des données source comportant des caractères inexistants dans le jeu de codes cibles peut provoquer des pertes de données importantes.

La conversion entre les caractères multi–octets et le code de caractères dépend de la configuration de l'environnement local. Ne permutez pas de codes de caractères entre deux processus, à moins que vous ne sachiez que chaque environnement local susceptible d'être utilisé traite les codes de caractères de façon cohérente. La plupart des environnements locaux AIX utilisent le code Unicode, sauf pour les environnements locaux basés sur les jeux de codes IBM–850 et IBM–eucTW.

Chapitre 12. Gestion des processus

Le processus est l'entité dont se sert le système d'exploitation pour contrôler l'utilisation des ressources système. AIX version 4 a introduit l'utilisation de *routines* pour contrôler la consommation du temps processeur ; toutefois, pour se référer au processus dans lequel une routine est en cours, la plupart des outils de gestion font appel à l'administrateur (et non à la routine).

Reportez-vous à *AIX 4.3 – Guide de l'utilisateur : système d'exploitation et unités* pour des informations de base sur la gestion de vos propres processus, par exemple sur l'arrêt/redémarrage d'un processus en cours ou sur la programmation d'un processus en différé. Ce manuel définit en outre les termes décrivant les processus, tels que le démon et le zombie.

Ce chapitre décrit, à l'attention de l'administrateur système, les processus et les outils fournis par le système d'exploitation et dédiés à la gestion de processus.

Pour les procédures relatives aux tâches, reportez–vous à la section Gestion des processus dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

AIX contient des outils permettant de :

- Surveiller la création, l'annulation, l'identité des processus, et leur consommation de ressources par le biais de :
  - **ps** qui rend compte des ID processus, des utilisateurs, de la consommation du temps CPU et d'autres attributs.
  - **who –u** qui rend compte de l'ID processus shell des utilisateurs connectés.
  - **svmon** qui rend compte de la consommation par le processus de la mémoire réelle. Pour en savoir plus sur cette commande, reportez-vous à *Performance Toolbox 1.2 and 2.1 for AIX: User's Guide*.
  - **acct** (mécanisme) qui écrit des articles en fin de processus résumant l'utilisation des ressources par le processus. Pour la mise en oeuvre d'un système de comptabilité, reportez-vous à "Comptabilité - généralités", page 15-2.
- Contrôler le niveau de priorité auquel prétend le processus pour le CPU par le biais de :
  - **nice** qui lance une commande spécifiant la priorité d'un processus. Reportez-vous à *AIX 4.3 Guide de l'utilisateur : système d'exploitation et unités.*
  - **renice** qui change la priorité d'un processus donné.
- Mettre fin aux processus incontrôlables par le biais de :
  - **kill** qui adresse un signal au(x) processus concerné(s).
- Régler les mécanismes de gestion de processus du système d'exploitation par le biais de :
  - **schedtune** qui permet de modifier les paramètres du programmateur de processus. Pour en savoir plus sur cette commande, reportez-vous à *AIX - Guide d'optimisation*.

Chapitre 13. Workload Management

AIX Workload Management (WLM) permet à l'administrateur de mieux contrôler le mode d'affectation des ressources aux différents processus par le biais du programmateur et du gestionnaire de mémoire virtuelle (VMM). Le WLM peut être utilisé pour éviter toute interférence entre les différentes classes de tâches et affecter les ressources en fonction des besoins des différents groupes d'utilisateurs.

WLM est essentiellement destiné aux gros systèmes. Ces systèmes sont souvent utilisés pour la consolidation des serveurs regroupant les charges de travail de plusieurs systèmes serveurs différents (tels que des systèmes d'imprimantes, de bases de données, d'utilisateurs généraux et de traitement des transactions) afin de réduire le coût de maintenance du système. Ces charges de travail interfèrent souvent entre elles. Elles ont des objectifs et des protocoles d'accord différents.

WLM permet également de compartimenter les communautés d'utilisateurs présentant des comportements très différents en termes de système. Cette fonction permet d'éviter tout déséquilibre entre des tâches interactives ou à faible sollicitation du CPU et des tâches de traitement par lots ou à forte sollicitation du CPU.

WLM vous permet de créer différentes classes de service pour des tâches et de définir leurs attributs. Ces attributs déterminent la quantité minimum et maximum de CPU et les ressources de mémoire physique à affecter à une classe. Vous pouvez ensuite classifier automatiquement les tâches en classes à l'aide des règles d'affectation correspondantes. Ces règles sont basées sur le nom d'utilisateur ou de groupe du processus ou le nom du chemin des applications.

Gestion des ressources à l’aide du WLM

Le WLM contrôle et régule l’utilisation du CPU et de la consommation de mémoire physique des routines et processus actifs du système. Vous pouvez définir des limites minimum ou maximum par classe pour chaque ressource gérée par le WLM. En outre, il est possible d’affecter une valeur cible par classe et par ressource. Cette valeur représente le quota de ressources optimal pour les tâches de cette classe. Les processus sont automatiquement affectés à une classe par le WLM au moment de son exécution, à l’aide d’un ensemble de règles d’affectation fourni par l’administrateur système.

Cette nouvelle version de WLM permet à WLM de démarrer en mode **active** – mode où WLM surveille et gère le CPU et la mémoire (mode de fonctionnement *normal*), ou en mode **passive** – mode où WLM ne fait que classer les processus et surveiller l’utilisation des ressources sans intervenir sur les algorithmes d’attribution de ressources standard &Symbol.AIX;. Ce mode est particulièrement intéressant lors de la mise en oeuvre d’une nouvelle configuration WLM :

- pour tester les règles de classification et d’affectation en vérifiant (avec **ps**) que les différentes applications sont correctement affectées à la classe à laquelle elles étaient destinées sans modifier le comportement du système.
- pour obtenir une ligne de base de l’utilisation des ressources de l’application “sans WLM,” à l’aide de la commande **wlmstat**. Ceci devrait offrir une référence aux administrateurs système et les aider à partager et limiter les ressources (si nécessaire) pour favoriser les applications critiques et/ou restreindre l’utilisation des ressources par les tâches de moindre importance, de façon à ce qu’ils puissent atteindre leurs objectifs de rentabilité.

Limites de ressources minimum et maximum

Les différentes ressources peuvent être limitées par les valeurs suivantes :

- Le pourcentage minimum de ressources mises à disposition si nécessaire. Valeurs possibles : nombres entiers de 0 à 100. En l’absence de spécification, la valeur par défaut est 0.
- Le pourcentage maximum de ressources mises à disposition si nécessaire, même en l’absence de limitation des ressources. Valeurs possibles : nombres entiers de 1 à 100. En l’absence de spécification, la valeur par défaut est 100.

Les valeurs de limite des ressources sont définies dans le fichier du même nom, et ce par type de ressources au sein de strophes pour chaque classe. Les limites sont définies avec un minimum et un maximum, séparé par un tiret. Les espaces blancs ne sont pas pris en compte. Chaque valeur limite est suivie du signe de pourcentage (%).

WLM n’impose pas de restrictions notables aux limites de ressources. Les seules restrictions sont les suivantes :

- La plage minimum doit être inférieure ou égale à la plage maximum.
- La somme des minima de toutes les classes au sein d’un niveau ne peut pas dépasser 100.

WLM applique la plage maximum pour limiter le volume de ressources affecté à une classe ou à un processus au sein d’une classe. Il est important de signaler qu’en cas de limitation de la mémoire, les performances des processus au sein de la classe concernée peuvent s’avérer très insuffisantes. Les valeurs minima de mémoire pour d’autres classes doivent être utilisées en priorité avant de définir des valeurs maxima de mémoire pour une classe.

Une contrainte de valeur minimum appliquée à une classe signifie qu’un minimum de ressource est toujours affecté aux processus au sein de cette classe. WLM ne peut pas garantir que les processus atteignent réellement la limite minimum. Cela dépend du mode d’utilisation des ressources par les processus ainsi que d’autres limites en vigueur. A titre d’exemple, une classe risque de ne pas pouvoir atteindre son quota minimum de CPU en raison d’une insuffisance de mémoire.

Partages cible

Les partages précisent la cible d'utilisation des différents types de ressources. Ces partages sont définis comme les quotas relatifs d'utilisation entre différentes classes. En l'absence de définition, la valeur par défaut est 1. Les partages peuvent être comparés à des pourcentages auto–variables.

A titre d'exemple, un système a 3 classes définies : A, B et C, dont les cibles sont respectivement 50, 30 et 20.

- Si ces 3 sont toutes actives, le nombre total de partages pour les classes actives est de 100. Leurs cibles exprimées en pourcentage sont 50%, 30% et 20% .
- Si A n'est pas active, le nombre total de partages est 50 (chaque partage représente donc 2%). Les pourcentages cible de B et C sont 60% et 40%.
- Si une seule classe est active, sa cible est 100%.

Dans cet exemple, la somme des partages des 3 classes était de 100 pour simplifier les calculs. Une cible peut être un nombre entre 1 et 65535.

La cible représente un pourcentage de ressources qui peut varier considérablement en fonction du nombre de classes actives à un moment donné. Toutefois, le WLM vérifie que la valeur dynamique de la cible reste compatible avec les valeurs minimum et maximum de la classe. Si le pourcentage calculé est inférieur au minimum, le WLM l'utilise comme cible. Si le pourcentage calculé est supérieur au maximum, le WLM l'utilise comme cible. Si ce pourcentage se situe entre le minimum et le maximum, le WLM utilise la valeur calculée.

Valeur de rang

La valeur de rang d'une classe est l'importance que cette classe revêt par rapport aux autres. La valeur de rang 0 est la plus importante, la valeur 9 la moins importante.

Exemples de classification et de limites

Il existe plusieurs méthodes de classification d'un processus qui fonctionnent parallèlement. Un algorithme de stricte concordance descendante est utilisé pour une souplesse de configuration maximale. Vous pouvez organiser les groupements de processus par utilisateur avec des cas particuliers pour les programmes portant certains noms, par nom de chemin avec des cas particuliers pour certains utilisateurs ou opter pour un tout autre système.

Exemple de limites de CPU

Cet exemple traite de l'affectation du CPU en partant du fait que chaque classe peut consommer tout le CPU qui lui a été affecté.

Deux classes, A et B, occupent le même rang. Les limites de CPU pour A sont [30% – 100%]. Les limites de CPU pour B sont [20% – 100%]. Lorsque ces deux classes fonctionnent et utilisent suffisamment de CPU, le WLM s'assure tout d'abord qu'elles disposent de leurs pourcentages minimum pour chaque seconde (moyenne calculée sur plusieurs secondes). Le WLM répartit ensuite les cycles de CPU restants en fonction des valeurs de partage cible du CPU.

Si les valeurs pour A et B sont respectivement de 60% et 40%, l'utilisation du CPU pour ces deux classes se stabilise de même à 60% et 40%.

On ajoute une troisième classe C. Cette classe est un groupe de tâches liées au CPU qui doivent s'exécuter avec environ la moitié (ou plus) du CPU disponible. La classe C a des limites de [20% – 100%] et des partages cible de 100%. Supposons que C occupe le même rang que A et B. Lorsque C est lancé, A et B voient alors leur quota d'affectation de CPU décroître fortement pour se stabiliser respectivement à 30%, 20% et 50%. Leurs cibles dans ce cas sont également le minimum pour A et B.

Il se peut qu'un administrateur système veuille éviter qu'une tâche en traitement par lots consomme jusqu'à 50% du CPU alors que d'autres tâches, pouvant présenter un niveau de priorité supérieur, s'exécutent également. Dans une situation comme l'exemple ci–dessus, C occupe un rang de priorité inférieur. C reçoit alors le volume de CPU laissé par A et B. Dans l'exemple ci–dessus, C ne recevrait rien, car A et B ont absorbé chacun 100% du CPU. Dans la plupart des cas, toutefois, A et B occupant un rang de priorité élevé, comportent des tâches interactives ou orientées–transactions, qui n'utilisent pas en permanence la totalité du CPU. C reçoit alors une partie du CPU qu'il partage avec d'autres classes occupant le même rang ou des rangs inférieurs.

Exemple de limites de mémoire

Cet exemple traite de l'affectation de mémoire à des groupes de processus assortis de cibles de mémoire variables. Trois groupes de processus doivent s'exécuter : un groupe de processus interactifs fonctionnant si nécessaire (PEOPLE), une tâche de traitement par lots s'exécutant toujours en arrière–plan (BATCH1) et une deuxième tâche de traitement par lots plus importante qui est lancée chaque nuit (BATCH0).

PEOPLE a un minimum de mémoire de 20%, une cible mémoire de 50 partages et une valeur de rang de 1. Cette limite minimum de 20% permet la reprise relativement rapide des applications du bureau lorsque les utilisateurs touchent leurs claviers.

BATCH1 a un minimum de mémoire de 50%, une cible mémoire de 50 partages et une valeur de rang de 3.

BATCH0 a un minimum de mémoire de 80%, une cible mémoire de 50 partages et une valeur de rang de 2.

Les classes PEOPLE et BATCH1 ont au total une limite minimum de mémoire de 70. Dans des conditions de fonctionnement normales (lorsque BATCH0 n'est pas exécuté), ces deux classes peuvent utiliser toute la mémoire réservée. Elles partagent le reste de la mémoire de la machine à concurrence de moitié, bien qu'elles occupent des rangs différents. A minuit, au moment du lancement de BATCH0, le minimum de mémoire atteint 150. Le WLM ignore les minima des rangs inférieurs tant que les processus de rang supérieur ne sont pas terminés. BATCH0 prend de la mémoire dans la réserve de BATCH1 (50%), mais pas dans la réserve de PEOPLE (20%). A la fin du BATCH0, les réserves de mémoire des processus de rang 3 sont à nouveau disponibles et le système retrouve son équilibre.

Configuration du WLM

Le WLM offre un niveau de contrôle précis en matière d'affectation des ressources. Toutefois, il est facile de configurer des valeurs conflictuelles pour les divers paramètres et obtenir ainsi des comportements système indésirables. Les conseils suivants vous permettent d'éviter cet écueil :

- Gardez à l'esprit votre base d'utilisateurs ainsi que ses principaux besoins lorsque vous définissez des classes et les règles d'affectation correspondantes.
- Pour connaître les besoins en ressources des principales applications. Une façon de déterminer leur utilisation des ressources pourrait être de lancer WLM en mode "passive" dès que vous avez défini vos classes et les règles de classification associées. **wlmstat** vous fournira alors un instantané de l'utilisation des ressources, par classe, lorsque l'attribution des ressources n'est pas contrôlée par WLM.
- Optez de préférence pour des cibles au–dessus des limites minimum et maximum. Les cibles offrent une plus grande flexibilité au système que les limites rigides. En outre, ces cibles permettent d'éviter toute insuffisance de ressources au niveau des applications.
- Essayez d'équilibrer la charge en n'utilisant que des cibles et contrôlez le système avec la commande **wlmstat**. Appliquez des limites minimum aux classes qui ne reçoivent pas suffisamment de partages.
- Définissez des tâches prioritaires par l'affectation de rangs.
- Utilisez la valeur maximum en dernier recours uniquement pour limiter les applications qui consomment de grandes quantités de partages. Le maximum peut également être utilisé pour limiter de manière stricte la consommation en ressources des utilisateurs (pour les besoins de la comptabilité, par exemple).

Définition des propriétés du WLM

Vous pouvez définir les propriétés du sous–système WLM en utilisant une interface utilisateur graphique Web-based System Manager, le SMIT, une interface utilisateur orientée ASCII ou en créant des fichiers ASCII à plat. Les interfaces Web-based System Manager et SMIT enregistrent les informations dans les mêmes fichiers ASCII à plat. Il s'agit de fichiers de propriétés WLM. Ils portent le nom de **classes**, **description**, **rules**, **limits** et **shares**. Les fichiers de propriétés WLM ne peuvent être chargés que par l'utilisateur racine.

Vous pouvez définir plusieurs ensembles de fichiers de propriétés assortis de différentes configurations de gestion de la charge de travail. Ces configurations sont habituellement situées dans des sous–répertoires de **/etc/wlm**. Un lien symbolique **/etc/wlm/current** pointe vers le répertoire contenant les fichiers de configuration courants. Ce lien est mis à jour par la commande **wlmcntrl** lorsque WLM commence par un ensemble spécifique de fichiers de configuration.

Chapitre 14. SRC et sous-systèmes

Ce chapitre présente le contrôleur de ressources système (SRC) et les différents sous–systèmes qu'il gère. Pour les procédure relatives aux tâches, reportez–vous à la section SRC et sous–systèmes dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

SRC - généralités

Le contrôleur SRC (System Resource Controller) fournit un ensemble de commandes et de sous-routines rendant le travail de création et de contrôle des sous-systèmes plus facile pour l'administrateur et le programmeur. Un *sous-système* est un programme, un processus, un ensemble de programmes ou un ensemble de processus capables d'opérer indépendamment ou avec un système de contrôle. Le sous-système est conçu comme une unité remplissant une fonction donnée.

Le SRC a été conçu pour minimiser l'intervention de l'opérateur. Il fournit un mécanisme de contrôle des processus du sous-système opérant à partir d'une ligne de commande et avec l'interface C Ce mécanisme comprend :

- Une interface utilisateur cohérente pour lancer, arrêter et générer des états.
- La journalisation des arrêts anormaux des sous-systèmes.
- Un programme de notification appelé lors de l'arrêt système anormal des processus associés.
- Le suivi d'un sous-système, d'un groupe de sous-systèmes ou d'un sous-serveur.
- La prise en charge du contrôle des opérations sur un système distant.
- Le rafraîchissement d'un sous-système (par exemple, après en avoir modifié la configuration).

SRC représente un moyen simple de lancer et arrêter la collecte d'informations sur l'état des processus.

Composants du sous-système

Un sous-système peut être doté des propriétés suivantes :

- Il possède un nom connu du système.
- Il requiert un environnement d'exécution plus complexe qu'un sous-routine ou un programme non privilégié.
- Il intègre des programmes d'application, des bibliothèques et un code sous-système.
- Il contrôle des ressources qu'il peut démarrer et arrêter par leur nom.
- Il a besoin d'être notifié de l'échec d'un processus associé pour effectuer un nettoyage ou restaurer des ressources.
- Il requiert un contrôle opérationnel plus important qu'un simple processus démon.
- Il a besoin d'être contrôlé à distance par un opérateur.
- Il met en oeuvre des sous-serveurs pour gérer des ressources spécifiques.
- Il ne se place pas lui-même en arrière-plan.

Voici quelques sous-systèmes : ypserv, ntsd, qdaemon, inetd, syslogd et sendmail.

Remarque : Pour en savoir plus sur les fonctions SRC d'un sous-système spécifique, reportez-vous à ce sous-système.

Pour la liste des sous-systèmes actifs et inactifs de votre système, exécutez la commande **lssrc –a**.

Groupe de sous-systèmes

Un *groupe de sous-systèmes* est un ensemble de sous-systèmes spécifiés. Regrouper des sous-systèmes permet de les contrôler ensemble et en une seule opération. Voici quelques exemples de groupes de sous-systèmes : TCP/IP, SNA Services, NIS (Network Information System) et NFS (Network File Systems).

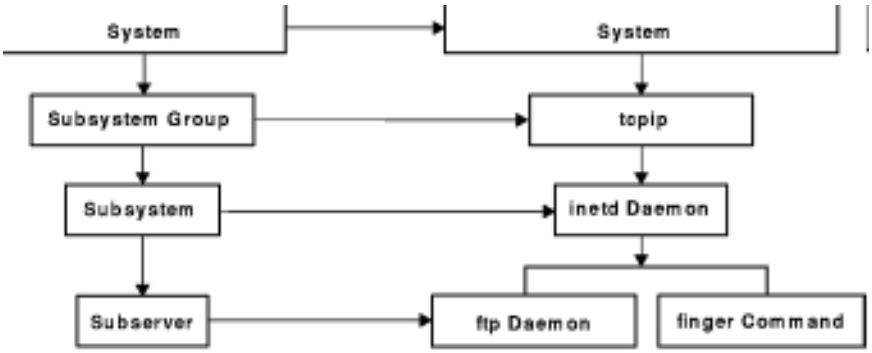
Sous-serveur

Un *sous-serveur* est un programme ou un processus appartenant à un sous-système. Un sous-système peut posséder plusieurs sous-serveurs, auquel cas il est responsable du démarrage et de l'arrêt des sous-serveurs ; il doit en outre fournir leurs états. Les sous-serveurs ne peuvent être définis que pour un sous-système communiquant par sockets et files de message IPC. Les sous-systèmes communiquant par signaux sont incompatibles avec les sous-serveurs.

Les sous-serveurs sont automatiquement démarrés en même temps que leurs sous-systèmes parent. Si vous tentez de démarrer un sous-serveur dont le parent est inactif, la commande **startsrc** démarre également le sous-système.

Structure hiérarchique de SRC

SRC possède une structure hiérarchique (voir figure). En début de structure, se trouve le système d'exploitation, suivi d'un groupe de sous-systèmes (tel que **tcpip**) lui-même contenant un sous-système (tel que le démon **inetd**), qui, à son tour, peut posséder plusieurs sous-serveurs (tels que le démon **ftp** et la commande **finger**.



SRC Hierarchical Structure

Commandes d'aministration SRC

démon **srcmstr**

Démarre SRC

commande **startsrc**

Démarre un sous-système, un groupe de sous-systèmes ou un sous-serveur

commande **stopsrc**

Arrête un sous-système, un groupe de sous-systèmes ou un sous-serveur

commande **refresh**

Rafraîchit un sous-système

commande **traceson**

Active le suivi d'un sous-système, d'un groupe de sous-systèmes ou d'un sous-serveur.

commande **tracesoff**

Désactive le suivi d'un sous-système, d'un groupe de sous-systèmes ou d'un sous-serveur.

commande **lssrc**

Recherche l'état d'un sous-système.

Chapitre 15. Comptabilité système

L’utilitaire de comptabilité système permet de collecter des informations et de générer des rapports sur l’exploitation (de groupe ou individuelle) des différentes ressources système.

Comptabilité - généralités

Ces informations peuvent servir à facturer l'exploitation des ressources aux utilisateurs et à contrôler certains aspects de l'exploitation du système. Pour faciliter la facturation, le système de comptabilité fournit le cumul par membre du groupe adm de l'utilisation des ressources et, avec la commande **chargefee** (le cas échéant), les éléments de facturation.

En outre, le système de comptabilité permet d'évaluer l'adéquation des affectations de ressources en cours, de définir des limites et des quotas de ressources, de planifier les besoins futurs et de commander des fournitures pour les imprimantes et autres unités.

Pour la mise en œuvre de l'utilitaire de comptabilité sur votre système, reportez-vous à :

Collecte et rapport de données système

Pour la collecte automatique de données, un membre du groupe adm doit suivre les procédures décrites à "Mise en œuvre d'un système de comptabilité" dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*. Ces procédures permettent au démon **cron** d'exécuter des commandes générant des données sur :

- le temps de connexion de chaque utilisateur sur le système,
- l'exploitation du processeur, de la mémoire et des ressources d'entrée-sortie,
- l'espace disque occupé par chaque fichier utilisateur,
- l'exploitation des imprimantes et des traceurs,
- l'exploitation des commandes.

Chaque session et chaque processus terminé fait l'objet d'un enregistrement écrit par le système. Ces enregistrements sont convertis en enregistrements comptables cumulés (**tacct**), regroupés par utilisateur puis fusionnés dans un rapport quotidien.
Un rapprochement des rapports quotidiens est effectué régulièrement pour générer les cumuls concernant l'exercice fiscal défini. Les méthodes de collecte des données et de rapport sur celles-ci, ainsi que les différents fichiers et commandes comptables sont décrits ci-après.

Toutefois, pour obtenir des informations spécifiques, un membre du groupe adm peut entrer certaines commandes (au clavier). Ces commandes sont décrites à "Commandes clavier", page 15-7.

Collecte de données comptables

Il existe différents types de données comptables : données sur les durées de connexion, les processus, l'utilisation disque, l'utilisation de l'imprimante et la taxation. Chacun de ces types est décrit dans les paragraphes suivants.

Durée de connexion

Ce sont les commandes **init** et **login** qui collectent ce type de données. Lors de la connexion, le programme **login** écrit un enregistrement dans le fichier **/etc/utmp**. Dans cet enregistrement figurent le nom de l'utilisateur, la date, l'heure et le port de connexion. Les commandes, telles que **who**, utilisent ce fichier pour rechercher les utilisateurs connectés sur les différentes stations. Si le fichier comptable sur l'heure de connexion **/var/adm/wtmp** existe, la commande **login** y intègre une copie de l'enregistrement de connexion.

A la fin du programme de connexion (c'est-à-dire généralement lorsque vous vous déconnectez), la commande **init** enregistre la fin de session dans un autre enregistrement du fichier **/var/adm/wtmp**. A la différence des enregistrements de connexion, ces enregistrements ne reprennent pas le nom de l'utilisateur. Le format de ces deux types d'enregistrements est décrit dans le fichier **utmp.h**.

La commande **acctwtmp** écrit également des entrées spécifiques dans le fichier **/var/adm/wtmp**, relatives aux démarrages et fermetures du système.

Pour en savoir plus, reportez-vous à "Rapports de données comptables", page 15-4.

Processus

Le système collecte des données sur l'utilisation par chaque processus en cours des différentes ressources. Ces données comprennent :

- les numéros des utilisateurs et groupes exécutant des processus,
- les huit premiers caractères du nom de la commande,
- le temps écoulé et le temps processus utilisé par le processus,
- la mémoire utilisée,
- le nombre de caractères transférés,
- le nombre de blocs disque écrits ou lus pour l'exécution du processus.

La commande **accton** enregistre ces données dans un fichier spécial, généralement dans le fichier **/var/adm/pacct**.

Les commandes associées sont **startup**, **shutacct**, **dodisk**, **ckpacct** et **turnacct**.

Pour en savoir plus, reportez-vous à "Rapport de données comptables", page 15-4.

Utilisation disque

Un grand nombre de données comptables sont collectées pendant l'exploitation des ressources. La commande **dodisk**, exécutée comme défini par le démon **cron**, écrit périodiquement dans le fichier **/var/adm/acct/nite/dacct** un enregistrement par utilisateur sur l'utilisation des disques. Pour ce faire, **dodisk** appelle d'autres commandes. Selon que la recherche comptable est pointue ou non, c'est la commande **diskusg** ou **acctdusg** qui collecte les données. La commande **acctdisk** écrit l'enregistrement comptable cumulé. Cet enregistrement est exploité ensuite par la commande **acctmerg** pour préparer le rapport comptable quotidien.

dodisk facture à l'utilisateur les liens aux fichiers trouvés dans son répertoire de connexion et répartit équitablement le coût de chaque fichier entre ces liens. Ainsi, le coût d'utilisation d'un fichier est partagé également entre tous ceux qui l'utilisent et ne grève pas les utilisateurs qui renoncent à l'accès au fichier concerné.

Pour en savoir plus, reportez-vous à "Utilisation disque", page 15-3.

Utilisation de l’imprimante

La collecte des données sur l’utilisation de l’imprimante résulte de la coopération de la commande **enq** et du démon de mise en file d’attente. **enq** met en file d’attente le nom de l’utilisateur, le numéro du travail d’impression et le nom du fichier à imprimer. Après l’impression du fichier, la commande **qdaemon** écrit un enregistrement ASCII dans un fichier, généralement le fichier **/var/adm/qacct**, contenant le nom et numéro de l’utilisateur et le nombre de pages imprimées. Vous pouvez trier ces enregistrements et les convertir en un enregistrement comptable cumulé.

Pour en savoir plus, reportez-vous à ”Utilisation disque”, page 15-3.

Taxation

Avec la commande **chargefee**, vous pouvez générer dans le fichier **/var/adm/fee** un enregistrement comptable cumulé en ASCII. Ce fichier peut être intégré aux rapports quotidiens avec la commande **acctmerg**.

Pour en savoir plus, reportez-vous à ”Comptabilité - généralités”, page 15-3.

Rapport de données comptables

Une fois les différents types de données comptables collectées, les enregistrements sont traités et convertis en rapports.

Les commandes comptables convertissent automatiquement les enregistrements en notation scientifique dès lors que les nombres sont élevés. Les nombres sont représentés en notation scientifique au format suivant :

*Base***e+***Exp*

OU

*Base***e–***Exp*

Selon cette formule, le nombre égal à *Base* est multiplié par 10 à la puissance **+***Exp* (exposant positif) ou **–***Exp* (exposant négatif). Par exemple, la notation scientifique 1.345e+9 est égale à 1.345x109, soit 1 345 000 000 et 1.345e–9 égale à 1.345x10–9, soit 0,000 000 001 345.

Durée de connexion

La commande **runacct** appelle deux commandes, **acctcon1** et **acctcon2**, pour traiter les enregistrements de connexion, de déconnexion et de fermeture du système collectés dans le fichier **/var/adm/wtmp**. La commande **acctcon1** convertit ces enregistrements en enregistrements de session et les inscrit dans le fichier **/var/adm/acct/nite/lineuse**. Ensuite la commande **acctcon2** convertit les enregistrements de session en un enregistrement comptable cumulé, **/var/adm/logacct**, intégré par la commande **acctmerg** aux rapports quotidiens.

Si vous exécutez **acctcon1** depuis la ligne de commande, vous devez l’assortir de l’indicateur **–l** pour générer le rapport correspondant, **/var/adm/acct/nite/lineuse**. Pour produire un rapport global sur la session pour la période comptable, **/var/adm/acct/nite/reboots**, utilisez la commande **acctcon1** avec l’indicateur **-o**.

La commande **lastlogin** produit un rapport donnant la dernière date à laquelle chaque utilisateur s’est connecté.

Processus

Deux commandes traitent les données de facturation collectées dans le fichier **/var/adm/pacct** ou dans un autre fichier spécifié. La commande **acctprc1** traduit l’ID utilisateur en un nom d’utilisateur et écrit des enregistrements ASCII contenant les éléments facturables : La commande **acctprc2** convertit ces enregistrements en enregistrements comptables cumulés intégrés aux rapports quotidiens par la commande **acctmerg**.

Les données comptables sur les processus fournissent des informations exploitables pour contrôler l'utilisation des ressources système. La commande **acctcms** en résume l'utilisation par nom de commande. Ce résumé indique combien de fois la commande a été exécutée, le temps processeur et la mémoire utilisés, et l'intensité d'exploitation de ces ressources (appelé *hog factor* = facteur de monopolisation). La commande **acctcms** génère des statistiques à long terme sur l'utilisation du système, dont l'utilisation totale et la fréquence d'utilisation des commandes.

La commande **acctcom** gère les mêmes données que **acctcms**, mais fournit des informations détaillées sur chaque processus. Vous pouvez afficher tous les enregistrements comptables de processus ou sélectionner ceux qui vous intéressent. Les critères de sélection comprennent la charge imposée par le processus, l'heure de fin de processus, le nom de la commande, l'utilisateur ou le groupe qui a appelé le processus et le port sur lequel il a été exécuté. A la différence des autres commandes, **acctcom** peut être exécutée par tous les utilisateurs.

Utilisation disque

Les enregistrements d'utilisation disque collectés dans le fichier **/var/adm/acct/nite/dacct** sont fusionnés dans les rapports comptables quotidiens par la commande **acctmerg**.

Utilisation de l'imprimante

L'enregistrement ASCII dans le fichier **/var/adm/qacct** peut être converti en un enregistrement comptable cumulé à intégrer au rapport quotidien avec la commande **acctmerg**.

Taxation

Si vous avez utilisé la commande **chargefee** pour facturer aux utilisateurs des services tels que les restaurations et consultations de fichiers, ou des matériels, un enregistrement comptable cumulé en ASCII est inscrit au fichier **/var/adm/fee**. Ce dernier est intégré aux rapports quotidiens par la commande **acctmerg**.

Rapport quotidien

Les données comptables brutes sur les durées de connexion et des processus, l'utilisation disque et imprimante, et la taxation sont fusionnées en rapports quotidiens par la commande **acctmerg**. Appelée par la commande **runacct** dans le cadre des routines quotidiennes, **acctmerg** génère les rapports suivants :

/var/adm/acct/nite/dacct	rapport intermédiaire produit dès qu'un des fichiers d'entrée est saturé.
/var/adm/acct/sum/tacct	rapport cumulé au format **tacct**. La commande **monacct** s'en sert pour produire le rapport ASCII mensuel.

La commande **acctmerg** peut faire des conversions entre les formats ASCII et binaire et fusionner des enregistrements de sources différentes en un enregistrement par utilisateur.

Rapport mensuel

Appelé par le démon **cron**, la commande **monacct** génère le rapport suivant :

/var/adm/acct/fiscal	produit par le rapport **/var/adm/acct/sum/tacct** avec la commande **monacct**. Selon la définition de cette commande, elle est exécutée tous les mois ou en fin d'exercice fiscal.

Commandes comptables

Les commandes comptables ne fonctionnent pas toutes de la même façon. Certaines :

- collectent des données ou produisent des rapports pour un type de comptabilité spécifique : durées des connexions ou des processus, utilisation disque ou imprimante, ou utilisation des commandes.
- appellent d'autres commandes. Par exemple, la commande **runacct**, généralement exécutée automatiquement par le démon **cron**, appelle nombre des commandes qui collectent et traitent les données comptables et préparent les rapports. Pour le traitement automatique de la comptabilité, vous devez tout d'abord configurer le démon **cron** pour exécuter **runacct**. Reportez-vous à la commande **crontab** pour plus de détails sur la configuration du démon **cron** en vue de soumettre des commandes à intervalles réguliers.
- exécutent des fonctions de maintenance et garantissent l'intégrité des fichiers de données actifs.
- permettent aux membres du groupe adm d'exécuter des tâches occasionnelles, telles que afficher des enregistrements spécifiques au moyen d'une commande entrée au clavier.
- permettent à l'utilisateur d'afficher des informations spécifiques. **acctcom** est l'unique commande utilisateur ; elle affiche la synthèse comptable des processus.

Commandes exécutées automatiquement

Plusieurs commandes, généralement exécutées par le démon **cron**, collectent automatiquement des données comptables.

runacct	gère la principale procédure comptable quotidienne. Généralement initiée par le démon **cron** en dehors des heures d'exploitation, **runacct** appelle plusieurs autres commandes comptables pour traiter les fichiers de données actifs et produire des synthèses sur l'utilisation des commandes et des ressources, triées par nom d'utilisateur. En outre, elle appelle la commande **acctmerg** pour générer une synthèse quotidienne et la commande **ckpacct** pour maintenir l'intégrité des fichiers de données actifs.
ckpacct	gère la taille du fichier **pacct**. Si vous devez recommencer la procédure **runacct** après le traitement échoué de ces enregistrements, mieux vaut avoir plusieurs petits fichiers **pacct**. **ckpacct** vérifie la taille du fichier de données actif **/var/adm/pacct** ; s'il dépasse 500 blocs, elle appelle la commande **turnacct switch** pour désactiver temporairement la comptabilité de processus. Les données sont transférées dans un nouveau fichier **pacct**, **/var/adm/pacct** *x*. (*x* est un entier incrémenté à chaque création de fichier **pacct**. Si le nombre de blocs disque libres chute en-deça de 500, **ckpacct** appelle la commande **turnacct off** pour désactiver la comptabilité de processus.
dodisk	appelle la commande **acctdisk** et la commande **diskusg** ou **acctdusg** pour inscrire les enregistrements d'utilisation disque au fichier **/var/adm/acct/nite/dacct**. Ces données sont ensuite fusionnées en rapports quotidiens.
monacct	produit une synthèse périodique à partir des rapports quotidiens.
sa1	collecte et enregistre des données binaires dans le fichier **/var/adm/sa/sa***dd*, où *dd* représente le jour (du mois).
sa2	écrit un rapport quotidien dans le fichier **/var/adm/sa/sa***dd*, où *dd* représente le jour (du mois). La commande supprime les rapports de plus d'une semaine du fichier **/var/adm/sa/sa***dd*.

Les commandes suivantes sont exécutées automatiquement par des procédures autres que le démon **cron** :

startup	ajoutée au fichier**/etc/rc**, **startup** initie les procédures de démarrage pour le système de comptabilité.
shutacct	enregistre l'heure de désactivation de la comptabilité en appelant la commande **acctwtmp** pour inscrire une ligne au fichier **/var/adm/wtmp**. Elle appelle ensuite la commande **turnacct off** pour désactiver la comptabilité de processus.

Commandes clavier

Les membres du groupe adm peuvent entrer ces commandes au clavier :

ac	imprime les enregistrements des durées de connexion. Cette commande est fournie pour des raisons de compatibilité avec les systèmes Berkeley Software Distribution (BSD).
acctcom	affiche la synthèse comptable des processus. Cette commande est aussi une commande utilisateur.
acctcon1	affiche la synthèse des durées de connexion. Cette commande doit être assortie de l'indicateur **–l** ou **–o**.
accton	active/désactive la comptabilité des processus.
chargefee	facture à l'utilisateur des frais prédéterminés pour les unités de travail effectuées. Ces frais sont intégrés aux rapports quotidiens par la commande **acctmerg**.
fwtmp	fait des conversions entre fichiers binaires et ASCII.
last	affiche des informations sur les connexions précédentes. Cette commande est fournie pour des raisons de compatibilité avec les systèmes Berkeley Software Distribution (BSD).
lastcomm	affiche des informations sur les dernières commandes exécutées. Cette commande est fournie pour des raisons de compatibilité avec les systèmes Berkeley Software Distribution (BSD).
lastlogin	affiche l'heure de la dernière connexion de chaque utilisateur.
pac	Prépare les enregistrements comptables de l'imprimante/du traceur. Cette commande est fournie pour des raisons de compatibilité avec les systèmes Berkeley Software Distribution (BSD).
prctmp	affiche un enregistrement de session.
prtacct	affiche les fichiers comptables cumulés.
sa	résume les informations comptables brutes pour aider à gérer de grands volumes de données comptables. Cette commande est fournie pour des raisons de compatibilité avec les systèmes Berkeley Software Distribution (BSD).
sadc	établit des rapports sur différentes opérations de systèmes locaux, telles que l'utilisation des tampons, l'activité d'E/S des disques et bandes, les compteurs d'unités TTY et ceux d'accès au fichiers.
time	imprime le temps réel, utilisateur et système requis pour exécuter une commande.
timex	indique le temps écoulé, utilisateur et d'exécution (en secondes).
sar	écrit en sortie standard le contenu des compteurs spécifiés des activités cumulées locales du système d'exploitation. La commande **sar** rapporte uniquement les activités locales.

Fichiers comptables

Il existe deux répertoires principaux de comptabilité : **/usr/sbin/acct** où sont stockés tous les programmes en C et les procédures shell indispensables pour lancer la comptabilité système et **/var/adm** contenant les fichiers de données, de rapports et de synthèse.

Ce sont les membres du groupe adm qui possèdent ces fichiers de données comptables ; tous les fichiers de données actifs (tels que **wtmp** et **pacct**) résident dans le répertoire personnel adm **/var/adm**.

Fichiers de données

Les fichiers du répertoire **/var/adm** sont les suivants :

/var/adm/diskdiag	sortie des diagnostics pendant l'exécution de programmes comptables sur disque.
/var/adm/dtmp	sortie de la commande **acctdusg**.
/var/adm/fee	sortie de la commande **chargefee**, sous forme d'enregistrements **tacct** en ASCII.
/var/adm/acct/	fichier comptable de processus actif.
/var/adm/wtmp	fichier comptable de processus actif.
/var/adm/Spacct *.mmdd*	fichier comptable de processus actif.

Fichiers de rapport et de synthèse

Les fichiers de rapports et de synthèse résident dans un sous-répertoire **/var/adm/acct**. Avant d'activer le système de comptabilité, vous devez créer les sous-répertoires suivants. Pour en savoir plus, reportez-vous à "Mise en œuvre d'un système de comptabilité".

/var/adm/acct/nite	contient les fichiers réutilisés chaque jour par la commande **runacct**.
/var/adm/acct/sum	contient les fichiers de synthèse avec cumuls quotidiennement mis à jour par la commande **runacct**.
/var/adm/acct/fiscal	contient les fichiers de synthèse mensuelle créés par la commande **monacct**.

Fichiers de commande runacct

Les fichiers de rapport et de synthèse suivants, générés par la commande **runacct**, présentent un intérêt tout particulier :

/var/adm/acct/ nite/lineuse	contient des statistiques sur l'utilisation de chaque ligne de terminal du système. Ce rapport est particulièrement utile pour détecter les lignes défectueuses. Une différence de plus de 3 à 1, le cas échéant, entre le nombre de déconnexions et de connexions signifie vraisemblablement qu'une ligne est défectueuse.
/var/adm/acct/ nite/daytacct	contient le fichier comptable cumulé du jour précédent.
/var/adm/acct/ sum/tacct	contient le cumul de chaque fichier **nite/daytacct** quotidien et peut servir à la facturation. La commande **monacct** réinitialise ce fichier chaque mois ou au moment de l'exercice fiscal.
/var/adm/acct/ sum/cms	contient le cumul des synthèses quotidiennes de commandes. La commande **monacct** lit cette version binaire du fichier et la purge. Une version ASCII figure dans **nite/cms**.
/var/adm/acct/ sum/daycms	contient la synthèse quotidienne de commandes dont une version ASCII est stockée dans **nite/daycms**.

/var/adm/acct/sum/loginlog	contient un enregistrement de la dernière date/heure à laquelle chaque ID utilisateur a été employé.
/var/adm/acct/sum/rprt *mmdd*	contient une copie du rapport quotidien sauvegardé par la commande **runacct**.

Fichiers du répertoire /var/adm/acct/nite

active	utilisé par la commande **runacct** pour enregistrer la progression et imprimer les messages d'erreur et les avertissements. Le fichier **active.** *mmjj* est une copie du fichier **active** effectuée par le programme **runacct** après une détection d'erreur.
cms	synthèse cumulée en ASCII sur les commandes utilisées par la commande **prdaily**.
ctacct.*mmjj*	enregistrements comptables cumulés sur les connexions.
ctmp	enregistrements sur les sessions de connexion.
daycms	synthèse quotidienne en ASCII sur les commandes, utilisée par la commande **prdaily**.
daytacct	enregistrements comptables cumulés sur un jour.
dacct	enregistrements comptables cumulés sur les disques, créés par la commande **dodisk**.
accterr	sortie des diagnostics générée pendant l'exécution de la commande **runacct**.
lastdate	dernière exécution de **runacct**, sous la forme **date +%m%d**.
lock1	contrôle l'exploitation en série de la commande **runacct**.
lineuse	rapport sur l'utilisation de la ligne tty, exploité par la commande **prdaily**.
log	sortie de la commande **acctconl**.
log*mmjj*	semblable à **log** après détection d'une erreur par la commande **runacct**.
reboots	contient les dates de début et fin de **wtmp,** et un listing des redémarrages du système.
statefile	enregistre l'état courant pendant l'exécution d'une commande **runacct**.
tmpwtmp	fichier **wtmp** corrigé par la commande **wtmpfix**.
wtmperror	Contient les messages d'erreur **wtmpfix**.
wtmperr*mmjj*	semblable à **wtmperror** après détection d'une erreur par la commande **runacct**.
wtmp.*mmjj*	Fichier **wtmp** du jour précédent.

Fichiers du répertoire /var/adm/acct/sum

cms	fichier de synthèse de toutes les commandes pour l'exercice fiscal en cours, en format binaire.
cmsprev	fichier de synthèse des commandes sans la dernière mise à jour.
daycms	fichier de synthèse des commandes pour le jour précédent, en format binaire.
lastlogin	fichier créé par la commande **lastlogin**.
pacct.*mmjj*	version concaténée de tous les fichiers **pacct** pour *mmjj.* Ce fichier est supprimé par la commande **remove** après le démarrage du système.
rprt*mmjj*	sortie sauvegardée de la commande **prdaily**.
tacct	fichier comptable de cumuls pour l'exercice fiscal en cours.

tacctprev	semblable à **tacct** sans la dernière mise à jour.
tacct*mmjj*	fichier comptable cumulé pour *mmjj*.
wtmp.*mmjj*	copie sauvegardée du fichier **wtmp** pour *mmjj*. Ce fichier est supprimé par la commande **remove** après le démarrage du système.

Fichiers du répertoire /var/adm/acct/fiscal

cms*?*	fichier de synthèse de toutes les commandes pour l'exercice fiscal, spécifié par *?*, en format binaire.
fiscrpt*?*	rapport semblable à celui de la commande **prdaily** pour l'exercice fiscal, spécifié par *?*, en format binaire.
tacct*?*	fichier comptable cumulé pour l'exercice fiscal, spécifié par *?*, en format binaire.

Formats des fichiers comptables

Les sorties générées par les fichiers comptables et les formats correspondants sont décrits ci-après :

wtmp	fichier comptable de processus actifs, au format défini dans le fichier **utmp.h**.
ctmp	enregistrements sur les sessions de connexion. Le format est décrit dans le fichier **ctmp.h**.
pacct*	enregistrements comptables de processus actifs, au format défini dans le fichier **/usr/include/sys/acct.h**.
Spacct*	fichier comptable de processus actif. Le format de ces fichiers est défini dans le fichier **sys/acct.h**.
daytacct	enregistrements comptables cumulés sur un jour. Le format du fichier est défini dans le fichier **tacct**.
sum/tacct	fichier binaire cumulant les synthèses quotidiennes des commandes. Le format de ce fichier est défini dans le fichier d'en-tête **/usr/include/sys/acct.h**.
ptacct	versions concaténées des fichiers **pacct**. Le format de ces fichiers est défini dans le fichier **tacct**.
ctacct	enregistrements comptables cumulés sur les connexions. Le format de ce fichier est défini dans le fichier **tacct**.
cms	synthèse comptable cumulée des commandes exploitée chaque jour par la commande **prdaily** au format binaire. Une version ASCII figure dans **nite/cms**.
daycms	synthèse quotidienne des commandes exploitée par la commande **prdaily** au format binaire. Une version ASCII figure dans **nite/daycms**.

Chapitre 16. Web-based System Manager

Web-based System Manager est une interface utilisateur graphique (GUI) permettant d'exécuter les tâches d'administration système suivantes : visualisation des utilisateurs et des groupes, des logiciels installés ainsi que des imprimantes et des périphériques, gestion des volumes logiques, des utilisateurs, des groupes et des ressources, montage et démontage des systèmes de fichiers, configuration réseau ainsi que bien d'autres tâches. Vous pouvez gérer ces systèmes à partir d'un écran en local ou à distance à partir d'un autre système AIX ou d'un ordinateur personnel équipé d'un navigateur web.

La GUI de Web-based System Manager offre un contrôle des objets par pointage et cliquage, qui offre une alternative à l'apprentissage et à l'utilisation des commandes AIX ou de SMIT.

Pour plus de détails sur les procédures d'utilisation du Web-based System Manager, reportez–vous à la section Exécution de Web-based System Manager dans *AIX 4.3 Quick Beginnings*. Pour les procédures d'installation et de configuration du Web-based System Manager, reportez–vous à Définition et exécution de Web-based System Manager dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Chapitre 17. SMIT

SMIT (System Management Interface Tool) - généralités

Bien que Web-based System Manager soit l'interface principale de gestion du système AIX, SMIT offre une interface en langage simple, orientée tâche. SMIT vous guide par l'intermédiaire de menus, de sélecteurs et de dialogues, vous libérant de tout souci de syntaxe, d'orthographe des commandes complexes, de validité des valeurs des paramètres. L'outil SMIT est exploitable avec deux interfaces, ASCII (non graphique) ou AIXwindows (graphique).

Remarque : En environnement de réseau, vous pouvez utiliser SMIT distribué. Reportez-vous à *Distributed SMIT 2.2 for AIX: Guide and Reference*.

SMIT est exploitable avec une interface ASCII (non graphique) et AIXwindows (graphique).

Quand vous appelez SMIT, un menu principal s'affiche, renvoyant à des sous–menus, ce qui permet de cibler de mieux en mieux la tâche souhaitée. Pour passer outre le menu principal et accéder directement à un sous-menu ou à un dialogue, utilisez la commande **smit** avec le paramètre *Fast Path*. Pour en savoir plus sur SMIT :

- Lancez SMIT, puis sélectionnez **Utilisation de SMIT (Informations seulement)** dans le menu principal SMIT.
- Dans les boîtes de dialogue SMIT, sélectionnez **Contextuelle (Ctrl+F1)** à partir du menu Aide, puis positionnez le curseur sur l'option de menu ou le champ sur lequel vous souhaitez obtenir des informations.

Le tableau suivant répertorie les tâches SMIT de base :

Tâches SMIT de base			
Tâche	*Raccourci SMIT*	*Sélection (ASCII)*	*Sélection (AIXwindows)*
Entrer SMIT	**smit**		
Quitter SMIT		F12	F12 ou option Exit SMIT du menu Exit
Commande Show		F6	F6 ou option Command du menu Show
Show fast path		F8	F6 ou option FastPath du menu Show

Chapitre 18. Le CDE Desktop

Avec CDE Desktop, vous avez accès aux unités et aux outils en réseau sans vous soucier de leur emplacement. Vous pouvez échanger des données entre applications tout simplement en faisant glisser et en déplaçant des objets.

Les administrateurs système trouveront que nombre de commandes complexes sont à présent aisément exécutables et sont en outre identiques d'une plate-forme à l'autre. Vous pouvez optimiser les coûts de vos investissements en matériel et logiciel avec une configuration centralisée et des applications distribuées aux utilisateurs. Il est aussi possible de centraliser la gestion de la sécurité, de la disponibilité et des échanges d'informations entre applications.

> **Remarque :** environnement CDE (Common Desktop Environment) 1.0 d'AIX. Dans l'aide et dans la documentation, cet environnement peut être appelé CDE Desktop, Desktop AIXwindows, Desktop CDE, AIX CDE 1.0 ou tout simplement Desktop.

Pour plus d'informations sur les procédures d'utilisation du CDE, reportez–vous au chapitre Gestion du CDE Desktop dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Chapitre 19. Service de recherche documentaire

Le Documentation Library Service vous permet de lire et d'effectuer des recherches dans des documents HTML en ligne. Il fournit une bibliothèque documentaire au sein de votre navigateur web. Au sein de cette application, vous pouvez cliquer sur des liens pour ouvrir les documents et les consulter. Vous pouvez également entrer des mots dans le formulaire de recherche de cette application. Les mots tapés sont recherchés, puis une page de résultats s'affiche, avec les liens conduisant aux documents cible.

Pour lancer cette application, tapez la commande **docsearch** ou sélectionnez l'icône d'aide CDE, cliquez sur l'icône d'aide de l'écran principal, puis sur l'icône Recherche documentaire.

Ce service de recherche documentaire vous permet d'accéder à des documents qui ont été enregistrés et indexés sur le serveur de documentation. Vous ne pouvez pas faire une recherche sur l'internet ou sur tous les documents de votre ordinateur. L'indexation génère un exemplaire spécialement compressé du document ou d'un ensemble de documents. La recherche porte sur cet index et non sur les documents originaux. Cette technique permet de gagner beaucoup en performance. Lorsqu'une phrase recherchée est trouvée dans l'index, le service de recherche documentaire affiche une page de résultats contenant des liens permettant de sélectionner et d'ouvrir le document dans lequel se trouve la phrase objet de la recherche.

Vous pouvez enregistrer les documents HTML de votre société dans cette bibliothèque. Ce faisant, tous les utilisateurs peuvent accéder et effectuer une recherche à l'aide de l'application en question. Avant de commencer une recherche, vous devez créer des index de documents. Pour plus d'informations sur l'ajout de vos propres documents dans la bibliothèque, reportez–vous à la section Documents et index.

A l'exception du moteur de recherche de la bibliothèque, ses composants sont installés avec le système d'exploitation de base. Pour fonctionner, le service de recherche documentaire doit être configuré. Vous pouvez configurer un ordinateur comme serveur de documentation et installer les documents sur cet ordinateur ou comme système client de documentation qui récupère tous ses documents sur un serveur de documentation. Si l'ordinateur est un serveur de documentation, le moteur de recherche et la documentation doivent également être installés manuellement.

Il est vivement recommandé de configurer complètement le service de recherche documentaire car il renferme les manuels du système d'exploitation et la documentation du Web–based System Manager. Même si vous n'avez pas besoin de ces manuels, configurez le service de recherche documentaire car il se peut que d'autres applications l'utilisent pour leur propre documentation en ligne. Pour toute instruction sur le mode d'installation et de configuration du service de recherche documentaire, reportez–vous aux sections Installation et configuration du Documentation Library Service et Installation de la documentation AIX dans le manuel *AIX 4.3 Installation Guide*.

Pour plus d'informations, reportez–vous aux rubriques suivantes du manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*:

- Modification de la configuration
- Documents et index
- Rubriques avancées
- Identification des problèmes

Chapitre 20. Power Management

Power Management est une technique d'optimisation de la consommation électrique du matériel et du logiciel. Elle est tout particulièrement rentable pour les produits Desktop et les produits alimentés par une batterie.

Les tâches de Power Management peuvent être exécutées grâce à :

- SMIT
- commandes
- application Power Management

Pour une liste des tâches et des procédures de Power Management que vous pouvez utiliser, reportez–vous à la section Utilisation de Power Management dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

La section Limites de Power Management - avertissements, qui suit contient des informations importantes destinées à tous les utilisateurs de Power management.

Limites de Power Management - avertissements

Les utilisateurs de Power Management doivent tenir compte des limites suivantes.

Modification de la configuration du système à l'état suspendu/hibernation

Modifier la configuration d'un système à l'état suspendu ou hibernation (par exemple, la taille mémoire, la taille des unités) peut produire des résultats imprévisibles. Ces résultats peuvent être la perte de données, l'endommagement d'un système de fichiers, une panne système ou l'échec de la reprise sur l'état suspendu ou hibernation.

Power Management non déclaré aux gestionnaires d'unités.

Quand Power Management n'est pas déclaré à un gestionnaire d'unités installé, la reprise sur l'état suspendu ou hibernation peut produire des résultats imprévisibles. Dans ce cas, n'utilisez jamais ces deux états. En tant qu'utilisateur racine, vous pouvez les désactiver avec une commande (voir ci-après) dont le résultat est pris en compte à l'amorçage suivant du système.

La commande suivante désactive l'hibernation du volume logique et n'autorise plus la sélection ultérieure de l'état suspendu ou hibernation :

```
/usr/lib/boot/disable_hibernation
```

Pour réactiver ces fonctions à compter de l'amorçage suivant du système et à condition qu'elles soient compatibles avec la plate-forme matérielle, utilisez la commande :

```
/usr/lib/boot/enable_hibernation
```

Amorçage depuis un CD–ROM ou autre support après hibernation

L'accès à **rootvg** depuis le mode maintenance tel que par un amorçage à partir du CD–ROM quand une image d'hibernation valide existe peut produire une perte de données et l'endommagement du système de fichiers.

N'utilisez le mode maintenance qu'après un arrêt ou une mise hors tension normale du système et non après une mise hors tension à l'état hibernation.

Connexion réseau d'un état suspendu/hibernation

Les connexions réseau d'un système à l'état suspendu/hibernation sont déconnectées et devront par conséquent être rétablies lors de la reprise. Elles devront par conséquent être rétablies lors de la reprise. Dans ce cas, les données placées localement en mémoire cache ne sont pas accessibles aux autres nœuds du réseau et le nœud local ne peut pas contrôler l'activité du réseau. Il est donc recommandé de ne pas utiliser les états suspendu et hibernation avec des interfaces réseau actives telles que TCP/IP, NFS, AFS, DCE, SNA, OSI, NetWare, NetBIOS, etc.

La commande suivante désactive l'hibernation du volume logique et n'autorise plus la sélection ultérieure de l'état suspendu ou hibernation :

```
/usr/lib/boot/disable_hibernation
```

Pour réactiver ces fonctions à compter de l'amorçage suivant du système et à condition qu'elles soient compatibles avec la plate-forme matérielle, utilisez la commande :

```
/usr/lib/boot/enable_hibernation
```

Fonctionnement du bouton de tension

Lorsque Power Management est actif, le bouton de tension est contrôlé par le logiciel. En cas de problème système quelconque, le logiciel requis pour effectuer la transition d'état Power Manager nécessaire via l'interrupteur de tension n'est peutêtre pas exécutable. Dans cette situation, ou lorsque nécessaire, il doit toujours être possible de couper l'alimentation immédiatement en appuyant trois fois de suite sur le bouton de tension (dans un délai de deux secondes). Ce qui permet de passer outre la transition d'état sélectionnée pour l'interrupteur, et requiert un réamorçage complet.

En outre, si le démon Power Management(/usr/bin/pmd) n'est jamais lancé (par une entrée dans le /etc/inittab par défaut), l'interrupteur fonctionne comme s'il n'y avait pas de gestion de l'alimentation : appuyer une fois sur le bouton met le sytème hors tension. Si /usr/bin/pmd est lancé puis tué, les deux premiers appuis sur le bouton sont sans effet, le troisième met le système hors tension. Cet état peut subsister aussi longtemps que /usr/bin/pmd n'est pas relancé.

Chapitre 21. Unités

Les unités comprennent les composants matériels tels que imprimantes, lecteurs, cartes, bus, boîtiers ainsi que les pseudo–unités, telles que le fichier spécial d'erreurs et le fichier spécial nul. Ce chapitre offre une vue d'ensemble des méthodes utilisées par le système d'exploitation pour gérer ces unités :

- Nœuds d'unités
- Codes d'emplacement
- Gestion des unités PCI hot plug

Nœuds d'unité

Les unités sont structurées en grappes appelées *nœuds*. Chaque nœud représente un sous-système logique d'unités, les unités de niveau inférieur étant dépendantes de celles de niveau supérieur dans la hiérarchie (enfant-parent). Par exemple, le nœud système est en haut de la hiérarchie et comprend l'ensemble des unités physiques du système. Le noeud système est le noeud le plus élevé dans la hiérarchie, suivi du bus et des cartes qui en dépendent. En bas de la hiérarchie sont situées les unités auxquelles ne sont pas connectées d'autres unités. Ces unités sont dépendantes de toutes les autres dans la hiérarchie.

Lors de l'amorçage, les dépendances parent-enfant servent à configurer toutes les unités formant un nœud. La configuration commence par le nœud supérieur et continue vers le bas. Les unités qui dépendent d'une unité d'un niveau supérieur ne peuvent être configurées qu'après celle-ci.

Classes d'unité

La gestion d'unité suppose la compréhension par le système d'exploitation des connexions d'unité autorisées. Le système d'exploitation classe les unités en trois groupes :

- classes fonctionnelles,
- sous-classes fonctionnelles,
- types d'unité.

Une classe fonctionnelle représente des unités exécutant la même fonction. Par exemple, les imprimantes constituent une classe fonctionnelle. Les classes fonctionnelles sont réparties en sous-classes, tenant compte de certaines similitudes des unités. Par exemple, les imprimantes peuvent être divisées en deux sous-classes : les imprimantes série et les imprimantes parallèles. Les types d'unités sont classés en fonction des modèles et du fabricant.

Les classes d'unité définissent des connexions parent-enfant pour le système d'exploitation. La hiérarchie définit les sous-classes pouvant être connectées pour chaque emplacement potentiel de connexion enfant. Par exemple, le terme carte 8 ports RS-232 indique que seules les unités de la sous-classe RS-232 peuvent être connectées à un des ports de la carte.

Les classes d'unité et leurs dépendances hiérarchiques sont actualisées dans une base de configuration d'unités ODM (Object Data Manager).

Base de configuration d'unités

Les données relatives aux unités figurent dans une base prédéfinie ou une base personnalisée constituant la base de configuration d'unités.

La base prédéfinie regroupe les données de configuration de toutes les unités possibles prises en charge par le système. Les données sur la hiérarchie de la classe d'unité font partie de cette base de données.

La base personnalisée contient des données de configuration concernant chaque unité définie et configurée du système. Un enregistrement de chaque unité connectée au système est conservé.

Le gestionnaire de configuration est un programme qui configure automatiquement les unités pendant l'amorçage du système et le temps d'exécution. Pendant ce processus, il fait appel aux données de la base prédéfinie et de la base personnalisée et actualise ensuite cette dernière.

Etats des unités

Quatre états peuvent caractériser les unités connectées au système :

Undefined	Le système ne connaît pas l'unité (état non défini).
Defined	Les données spécifiques de l'unité sont enregistrées dans la base personnalisée sans toutefois être disponibles (état défini).
Available	Une unité définie est associée au système d'exploitation ou l'unité définie est configurée (état disponible).
Stopped	L'unité n'est pas disponible mais est toutefois connue de son gestionnaire (état arrêté).

Quand une unité tty et une imprimante utilisent alternativement le même connecteur tty, dans la base de configuration d'unités, l'unité tty et l'imprimante sont définies sur le même parent et le même port. Une seule unité peut être configurée à la fois. Pendant la configuration du connecteur tty, les données d'installation de l'imprimante sont retenues en attendant d'être à nouveau configurées. L'unité n'est pas supprimée mais est à l'état défini. Maintenir une unité à l'état défini retient les données de la base personnalisée pour une unité qui n'est pas couramment utilisée, avant sa première mise à disposition ou pendant sa suppression temporaire du système.

C'est le gestionnaire d'unité, s'il existe, qui rend l'unité disponible.

Certaines unités, en particulier les pseudo-unités TCP/IP, ont recours à l'état arrêté.

Gestion des unités

Pour gérer les unités (ajout, suppression, etc.), vous pouvez utiliser les commandes du système d'exploitation, SMIT ou l'application Web-based System Manager Devices.

Codes d’emplacement

Le *code d’emplacement* représente le chemin d’accès de l’unité centrale ou du tiroir CPU à la carte, aux câbles de signaux et, le cas échéant, au multiplexeur asynchrone de l’unité ou de la station. Ce code représente un autre moyen d’identifier les unités physiques.

Le code d’emplacement est formé de une à quatre zones de données de deux caractères, selon le type d’unité. Ces zones représentent le tiroir, l’emplacement de carte, le connecteur et le port. Chacune de ces zones comporte deux caractères.

Le code d’emplacement du tiroir est un code de deux caractères situé dans la zone correspondante. Celui de la carte figure dans la zone du tiroir et celle de l’emplacement de carte au format AA–BB, AA correspondant au code d’emplacement du tiroir et BB au code du bus et de l’emplacement de la carte. Les autres unités ont des codes d’emplacement au format AA–BB–CC ou AA–BB–CC–DD, où AA–BB est le code d’emplacement de la carte à laquelle l’unité est connectée, CC le code du connecteur de la carte et DD un numéro de port ou une adresse d’unité SCSI.

Pour trouver les étiquettes mentionnant les codes d’emplacement sur le matériel, reportez-vous au guide de l’opérateur.

Carte

Le code d’emplacement d’une carte est un code au format AA–BB, où AA représente le code d’emplacement du tiroir logeant la carte et BB le bus d’E/S et l’emplacement de la carte.

La valeur 00 dans la zone AA signifie que la carte est située dans le tiroir CPU ou l’unité centrale, selon le type du système. Dans cette zone, toute autre valeur indique que la carte loge dans une unité d’extension d’E/S. Dans ce cas, la valeur de AA identifie le bus d’E/S et le numéro d’emplacement du tiroir CPU où se trouve la carte d’extension asynchrone. Pour le premier chiffre, la valeur 0 correspond au bus d’E/S standard et la valeur 1 au bus d’E/S en option. Le second chiffre identifie le numéro d’emplacement sur le bus d’E/S indiqué.

Le premier chiffre de la zone BB identifie la carte d’E/S contenant la carte d’adaptation. Si elle loge dans le tiroir CPU ou l’unité centrale, la valeur de ce chiffre est 0 pour le bus d’E/S standard et 1 pour celui en option. Si la carte est située dans un tiroir d’extension d’E/S, ce chiffre est 0. Le second chiffre identifie le numéro d’emplacement sur le bus d’E/S indiqué (ou dans le tiroir d’extension d’E/S).

Le code d’emplacement 00–00 identifie la carte d’E/S standard.

Exemples :

00–05	carte à l’emplacement 5 de la carte d’E/S standard, située dans le tiroir CPU ou l’unité centrale, selon le type du système.
00–12	carte à l’emplacement 2 du bus d’E/S en option, située dans le tiroir CPU.
18–05	carte à l’emplacement 5 d’un tiroir d’extension d’E/S. Ce tiroir est connecté à la carte d’extension asynchrone située à l’emplacement 8 du bus d’E/S en option dans le tiroir CPU.

Imprimante/traceur

Les codes d'emplacement 00-00-S1-00 ou 00-00-S2-00 représentent l'imprimante/traceur connecté au port série s1 ou s2 de la carte d'E/S standard. Le code 00-00-0P-00 représente l'imprimante parallèle connectée au port parallèle de la carte d'E/S standard.

Tout autre code d'emplacement indique que l'imprimante ou le traceur n'est pas connecté à la carte d'E/S standard mais à une autre carte. Le code a le format AA-BB-CC-DD, où AA-BB indique le code d'emplacement de la carte pilotant l'unité.

AA	La valeur 00 dans la zone AA signifie que la carte est située dans le tiroir CPU ou l'unité centrale, selon le type du système. Toute autre valeur indique que la carte loge dans un tiroir d'extension d'E/S ; dans ce cas, le premier chiffre identifie le bus d'E/S et le second le numéro d'emplacement sur le bus dans le tiroir CPU contenant la carte d'extension asynchrone à laquelle le tiroir d'extension d'E/S est connecté.
BB	Le premier chiffre de cette zone identifie le bus d'E/S contenant la carte. Si la carte loge dans le tiroir CPU ou l'unité centrale, la valeur 0 représente le bus d'E/S standard et 1 celui en option. Si la carte se trouve dans un tiroir d'extension d'E/S, ce chiffre est 0. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S (ou dans le tiroir d'extension d'E/S) contenant la carte.
CC	Cette zone identifie le connecteur de la carte à laquelle le multiplexeur asynchrone est connecté. Les valeurs possibles de cette zone sont 01, 02, 03 et 04.
DD	Cette zone identifie le numéro de port sur le multiplexeur asynchrone auquel l'imprimante ou le traceur est connecté.

Unité tty

Les codes d'emplacement 00-00-S1-00 ou 00-00-S2-00 indiquent que l'unité tty est connectée aux ports série d'E/S standard s1 ou s2.

Tout autre code indique que l'unité tty n'est pas connecté à la carte d'E/S standard mais à une autre carte. Le code a le format AA-BB-CC-DD, où AA-BB indique le code d'emplacement de la carte pilotant l'unité.

AA	La valeur 00 dans la zone AA signifie que la carte est située dans le tiroir CPU ou l'unité centrale, selon le type du système. Toute autre valeur indique que la carte loge dans une unité d'extension d'E/S. Dans ce cas, le premier chiffre identifie le bus d'E/S et le second le numéro d'emplacement sur le bus dans le tiroir CPU contenant la carte d'extension asynchrone à laquelle le tiroir d'extension d'E/S est connecté.
BB	Le premier chiffre de la zone BB identifie le bus d'E/S contenant la carte d'adaptation. Si elle loge dans le tiroir CPU ou l'unité centrale, la valeur de ce chiffre est 0 pour le bus d'E/S standard et 1 pour celui en option. Si la carte est située dans un tiroir d'extension d'E/S, ce chiffre est 0. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S indiqué (ou dans le tiroir d'extension d'E/S).
CC	Cette zone identifie le connecteur de la carte à laquelle le multiplexeur asynchrone est connecté. Les valeurs possibles de cette zone sont 01, 02, 03 et 04.
DD	Cette zone identifie le numéro de port sur le multiplexeur asynchrone auquel l'unité tty est connectée.

Unité SCSI

Toute unité SCSI, y compris :

- CD-ROM,
- disques,
- unités Initiator,
- unités optiques de lecture-écriture,
- bandes,
- mode Target.

Le format du code d'emplacement est AA–BB–CC–S,L. AA–BB identifie le code d'emplacement de la carte SCSI pilotant l'unité SCSI.

AA	La valeur 00 dans la zone AA signifie que la carte pilotant l'unité est située dans le tiroir CPU ou l'unité centrale, selon le type du système.
BB	Cette zone identifie le bus d'E/S et l'emplacement contenant la carte. Le premier chiffre identifie représente le bus d'E/S standard. La valeur 00 représente le bus d'E/S standard et 1 celui en option. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S contenant la carte. La valeur 00 dans cette zone indique le contrôleur SCSI standard.
CC	Cette zone identifie le bus SCSI de la carte à laquelle l'unité est connectée. Pour une carte ne fournissant qu'un seul bus SCSI, la valeur de cette zone est 00. Sinon, la valeur 00 indique une unité reliée au bus SCSI interne de la carte et la valeur 01 une unité reliée au bus SCSI externe de la carte.
S,L	Cette zone identifie l'ID SCSI et le numéro d'unité logique (LUN) de l'unité SCSI. La valeur S identifie l'ID SCSI et L le numéro d'unité logique (LUN).

Unité DBA

Pour une unité DBA (Direct-Attached Disk), le format du code d'emplacement est AA–BB. La valeur de la zone AA est 00, indiquant que le disque réside sur l'unité centrale. La zone BB indique le bus d'E/S et le numéro d'emplacement auquel le disque est relié. Le premier chiffre est toujours 0, indiquant que le disque est relié au bus d'E/S standard. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S standard auquel le disque est relié.

Disque série

Le code d'emplacement des unités de disque en série a le format AA–BB–CC–DD, où AA–BB indique le code d'emplacement de la carte pilotant l'unité.

Les différentes zones s'interprètent comme suit :

AA	La valeur 00 dans la zone AA signifie que la carte pilotant l'unité est située dans le tiroir CPU ou l'unité centrale, selon le type du système.
BB	Cette zone identifie le bus d'E/S et l'emplacement contenant la carte. Le premier chiffre identifie le bus d'E/S standard, 0 pour le bus d'E/S standard et 1 celui en option. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S contenant la carte.
CC	Cette zone identifie le connecteur de la carte à laquelle le tiroir pilotant l'unité est connecté. Les valeurs possibles de cette zone sont 00, 01, 02 et 03.
DD	Cette zone identifie le numéro d'unité logique (LUN) du disque. Il correspond à l'emplacement du tiroir logeant le disque.

Unité de disquette

Les codes d'emplacement des unités de disquette sont `00-00-0D-01` ou `00-00-0D-02`, indiquant qu'elles sont reliées au port `0` ou `1` de la carte principale d'E/S standard.

Rotateur/clavier LPFK

Pour une carte d'entrée graphique reliée à une unité rotateur/clavier LPFK, le code d'emplacement à le format `AA-BB-CC.`

Les différentes zones s'interprètent comme suit :

`AA`	La valeur `00` dans la zone `AA` signifie que la carte pilotant l'unité est située dans le tiroir CPU ou l'unité centrale, selon le type du système.
`BB`	Cette zone identifie le bus d'E/S et l'emplacement contenant la carte. Le premier chiffre identifie le bus d'E/S standard, `0` pour le bus d'E/S standard et `1` celui en option. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S contenant la carte.
`CC`	Cette zone indique le connecteur de carte auquel l'unité est connectée. La valeur est `01` ou `02`, selon que l'unité est reliée au port `1` ou `2` sur la carte.

Remarque : En série, ces unités n'indiquent pas de codes d'emplacement. Elles sont supposées reliées à une unité tty. Cette dernière est spécifiée par l'utilisateur lors de la définition des rotateurs/claviers LPFK.

Port multiprotocole

Le code d'emplacement d'un port multiprotocole a le format `AA-BB-CC-DD`, où `AA-BB` indique le code d'emplacement de la carte multiprotocole.

Les différentes zones s'interprètent comme suit :

`AA`	La valeur `00` dans la zone `AA` signifie que la carte est située dans le tiroir CPU ou l'unité centrale, selon le type du système.
`BB`	Cette zone identifie le bus d'E/S et l'emplacement contenant la carte. Le premier chiffre identifie le bus d'E/S standard, `0` pour le bus d'E/S standard et `1` celui en option. Le second chiffre identifie le numéro d'emplacement sur le bus d'E/S contenant la carte.
`CC`	Cette zone identifie le connecteur de la carte à laquelle le multiplexeur multiprotocole est connecté. La valeur est toujours `01`.
`DD`	Cette zone identifie le numéro de port physique sur le multiplexeur multiprotocole. Les valeurs possibles de cette zone sont `00`, `01`, `02` et `03`.

Gestion des unités PCI hot plug

Cette section offre une vue d'ensemble de la gestion des unités hot plug, notamment du support PCI hot plug pour carte PCI. Si vous souhaitez plus d'informations sur l'utilisation des fonctions et des procédures PCI hot plug pour déconfigurer, ajouter, supprimer et remplacer des cartes, reportez–vous à la section Gestion des connecteurs hot plug dans le manuel *AIX 4.3 Guide de gestion du système – Système d'exploitation et unités* .

Si vous souhaitez en savoir plus sur les commandes permettant d'afficher des informations sur les emplacements PCI hot plug et pour ajouter, remplacer et supprimer des cartes PCI hot plug, reportez–vous à :

- La commande lsslot, dans le manuel *AIX Commands Reference, Volume 3*. Cette commande affiche une liste de tous les emplacements hot plug accompagnés de leurs caractéristiques.
- La commande drslot, dans le manuel *AIX Commands Reference, Volume 2*. Cette commande prépare un connecteur hot plug pour l'ajout ou le retrait d'une carte hot plug.

Présentation

La gestion des unités PCI hot plug repose sur des interfaces utilisateur permettant de gérer des connecteurs hot plug, également appelés connecteurs de *reconfiguration dynamique* ou emplacements. Un connecteur définit le type d'emplacement, par exemple, PCI. Un emplacement est un identifiant unique. On appelle reconfiguration dynamique la capacité du système à s'adapter aux changements de configuration matérielle ou micrologicielle alors qu'il est en cours de fonctionnement.

Le support PCI hot plug pour cartes PCI est un sous–ensemble spécifique de la fonction de reconfiguration dynamique qui offre la possibilité d'ajouter, de retirer et de remplacer des cartes PCI pendant que le système hôte fonctionne, sans interrompre l'utilisation d'autres cartes sur le système. Vous pouvez aussi afficher des informations sur les emplacements PCI hot plug.

Des types de connecteurs hot plug différents exigent des opérations différentes pour effectuer diverses fonctions de gestion hot plug. A titre d'exemple, l'ajout d'une unité SCSI implique des opérations différentes de celles que vous effectuez pour ajouter une carte PCI.

> **Remarque :** Bien que la gestion des unités PCI hot plug permette d'ajouter, de supprimer et de remplacer des cartes PCI sans mise hors tension du système ou réamorçage du système d'exploitation, certaines unités logées dans les emplacements hot plug ne peuvent pas être gérées de cette façon. A titre d'exemple, le disque dur qui constitue le groupe de volumes rootvg ou le contrôleur d'E/S auquel il est relié ne peut pas être supprimé ou remplacé sans mise hors tension du système puisqu'il est nécessaire à l'exécution du système d'exploitation.

Certaines cartes ne peuvent pas être connectées à chaud et ne devraient pas être retirées pendant que le système est sous tension. Pour déterminer si un adaptateur peut être connecté à chaud, reportez–vous à la liste des cartes PCI prises en charge dans le document *PCI Adapter Placement Reference*, livré avec les unités centrales qui gèrent les unités PCI hot plug.

Ajout d'une carte PCI hot plug

Vous pouvez insérer une nouvelle carte PCI dans un emplacement PCI disponible pendant que le système d'exploitation est en cours de fonctionnement. Il peut s'agir d'une autre carte du même type que celle qui est actuellement installée ou d'un type de carte PCI différent. De nouvelles ressources sont mises à la disposition du système d'exploitation et des applications, sans qu'un réamorçage du système ne soit nécessaire. Une carte peut être ajoutée pour plusieurs raisons :

- Extension des fonctionnalités ou de la capacité de votre matériel ou de votre micrologiciel.

- Migration de cartes PCI à partir d'un système ne nécessitant plus la fonctionnalité fournie par ces cartes.
- Installation d'un nouveau système pour lequel les cartes deviennent disponibles après la configuration initiale des sous–systèmes matériels optionnels, y compris les cartes PCI, et installation et amorçage du système d'exploitation.

Pour connaître la procédure d'ajout d'une carte PCI hot plug, reportez–vous à la section Ajout d'une carte PCI hot plug dans le manuel *AIX 4.3 Guide de gestion du système – Système d'exploitation et unités*

Retrait d'une carte PCI hot plug

Vous pouvez retirer une carte PCI hot plug de son tiroir d'E/S ou de son boîtier sans arrêter le système d'exploitation ni couper l'alimentation du système. Lorsqu'une carte est retirée, les ressources fournies par cette dernière ne sont plus accessibles au système d'exploitation et aux applications. Avant de retirer la carte, vous devez vous assurer qu'aucune des ressources faisant appel à la carte n'est utilisée. Une carte peut être retirée pour plusieurs raisons :

- Retrait de sous–systèmes d'E/S existants.
- Retrait d'une carte devenue inutile ou défectueuse sans qu'une carte de rechange ne soit disponible.
- Migration d'une carte vers un autre système lorsque la fonction n'est plus nécessaire sur le système dont elle est retirée.

Pour connaître la procédure de retrait d'une carte PCI hot plug, reportez–vous à la section Retrait ou remplacement d'une carte PCI hot plug dans le manuel *AIX 4.3 Guide de gestion du système – Système d'exploitation et unités* .

Remplacement d'une carte PCI hot plug

Vous pouvez échanger une carte PCI hot plug défectueuse ou défaillante par une autre carte du même type, sans arrêter le système d'exploitation ni couper l'alimentation du système. La carte étant de même type, le gestionnaire d'unité existant peut prendre en charge la carte de rechange. La fonction de remplacement conserve les informations de configuration de la carte échangée et les compare à celles de la carte de rechange. La configuration des unités et les informations de configuration concernant les unités sous la carte sont utilisées pour la configuration de l'unité de rechange.

Une carte peut être remplacée pour plusieurs raisons :

- Remplacement temporaire de la carte pour faciliter la détermination d'un problème ou pour isoler un FRU défaillant.
- Remplacement d'une carte défectueuse, ou défaillante par intermittence par une carte en état de marche.
- Remplacement d'une carte redondante défaillante dans une configuration HACMP ou d'accès multiples au stockage.

Pour connaître la procédure de remplacement d'une carte PCI hot plug, reportez–vous à la section Retrait ou remplacement d'une carte PCI hot plug dans le manuel *AIX 4.3 Guide de gestion du système – Système d'exploitation et unités* .

Utilisation des ressources

Avant de retirer ou de remplacer une unité hot plug, vous devez déconfigurer cette dernière. Le gestionnaire d'unité associé doit libérer toutes les ressources système qu'il a allouées à l'unité. Il lui faut donc désallouer et libérer la mémoire, annuler la définition des gestionnaires d'interruption et EPOW, libérer les ressources DMA et d'horloge, et effectuer toute autre étape obligatoire. Le gestionnaire doit également s'assurer que les interruptions, la mémoire et les E/S du bus sont désactivées sur l'unité.

L'administrateur système doit normalement effectuer les tâches suivantes avant et après la procédure de retrait :

- Arrêter et restaurer les applications, démons ou processus qui utilisent l'unité.
- Démonter et remonter les systèmes de fichiers.
- Supprimer et recréer les définitions d'unité et effectuer d'autres opérations nécessaires à la libération d'une unité utilisée.
- Placer le système dans un état sûr pour les opérations de maintenance.
- Obtenir et installer les gestionnaires d'unités nécessaires.

Remarque: Si vous ajoutez une carte selon la méthode d'ajout ou de remplacement de carte PCI, cette carte et ses unités enfants risquent de ne pas pouvoir être désignées comme unités d'amorçage avec la commande boolist. Vous aurez peut–être à réinitialiser la machine pour que toutes les unités d'amorçage potentielles soient connues du système d'exploitation.

Dans certains cas, l'administrateur système peut également effectuer les tâches suivantes :

- Préparer la carte PCI hot plug à insérer, retirer ou remplacer.
- Identifier les emplacement ou les cartes PCI impliqués dans l'opération hot plug.
- Retirer ou insérer les adaptateurs PCI hot plug.

Attention : Avant toute tentative de retrait ou d'insertion de cartes PCI hot plug, reportez–vous au document de référence relatif à l'emplacement des cartes PCI (PCI Adapter Placement Reference), fourni avec les unités centrales qui gèrent les unités hot plug, pour déterminer si votre carte peut être connectée à chaud. Reportez–vous aux instructions d'installation et de retrait des cartes dans la documentation de votre unité centrale.

Déconfiguration d'une unité à partir du système

Les opérations de retrait et de remplacement échouent si l'unité connectée à l'emplacement identifié n'a pas été déconfigurée et n'est pas à l'état défini. Vous pouvez effectuer cette opération avec la commande rmdev. Avant de placer la carte à l'état défini, fermez toutes les applications qui l'utilisent, sinon la commande échouera.

Déconfiguration des cartes de communication

Cette section offre un aperçu du processus de déconfiguration des cartes de communication PCI. Il s'agit notamment des cartes Ethernet, FDDI, ATM et de réseau en anneau à jeton. Pour connaître les étapes de la procédure, reportez–vous à la section Déconfiguration des cartes de communication dans le manuel AIX 4.3 Guide de Gestion du Système : Système d'exploitation et unités.

Si votre application utilise le protocole TCP/IP, vous devez supprimer l'interface TCP/IP pour la carte dans la liste d'interfaces réseau avant de placer la carte à l'état défini. Utilisez la commande netstat pour déterminer si votre carte est configurée pour TCP/IP et vérifier les interfaces réseaux actives sur votre carte.

Une carte Ethernet peut avoir deux interfaces : Ethernet standard (enX) ou IEEE 802.3 (etX). X correspond au nombre contenu dans le nom de carte entX. Le protocole TCP/IP ne peut être utilisé que par l'une de ces interfaces à la fois. A titre d'exemple, la carte Ethernet ent0 peut avoir les interfaces en0 et et0.

Une carte de réseau en anneau à jeton ne peut avoir qu'une seule interface. Token–ring (trX). X correspond au nombre contenu dans le nom de carte tokX. A titre d'exemple, la carte de réseau en anneau à jeton tok0 a une interface tr0.

Une carte ATM ne peut avoir qu'une interface atm : ATM (atX). X correspond au nombre contenu dans le nom de carte atmX. A titre d'exemple, la carte ATM atm0 a une interface at0. Toutefois, pour ce qui est des cartes ATM, plusieurs clients peuvent être émulés par l'intermédiaire d'une seule et même carte.

La commande ifconfig supprime une interface sur le réseau. La commande rmdev déconfigure l'unité PCI tout en conservant sa définition d'unité dans la classe d'objets des unités personnalisées. Une fois que la carte est à l'état défini, vous pouvez utiliser la commande drslot pour la supprimer.

Chapitre 22. Unités de bande

Les sujets abordés sont les suivants :

Les tâches de base des unités de bande sont répertoriées sous la rubrique Unités de bande dans le manuel *AIX 4.3 Guide de Gestion du Système: Système d'exploitation et unités*.

Attributs des unités de bande

Cette section décrit les attributs modifiables des unités de bande. Vous pouvez les afficher ou les modifier à l'aide de l'application Web-based System Manager Devices, de SMIT ou de commandes (notamment **lsattr** et **chdev**).

Chaque type d'unité de bande n'utilise qu'une partie des attributs.

Présentation générale

Taille de bloc

L'attribut taille de bloc indique la taille de bloc à utiliser pour la lecture ou l'écriture d'une bande. Les données sont inscrites sous forme de blocs de données délimités par des espaces interblocs. Sur les bandes non formatées, il est préférable d'utiliser des blocs de grande taille pour réduire le nombre d'espaces interblocs et disposer ainsi de davantage d'espace pour l'inscription des données. La valeur **0** indique une taille de bloc variable. Les valeurs par défaut et les valeurs admises varient en fonction de l'unité de bande.

Mémoires tampon

Lorsque vous positionnez l'attribut mémoires tampon sur **yes** (avec **chdev**, l'attribut **mode**), les applications reçoivent un message de confirmation d'écriture dès le transfert des données en mémoire tampon sur l'unité de bande, même si l'écriture de bande n'est pas encore réalisée. Avec la valeur **no**, l'écriture n'est notifiée qu'une fois les données inscrites sur la bande. La valeur **no** n'est pas compatible avec la lecture et l'écriture sur bande en mode continu. La valeur par défaut est **yes**.

Lorsque cet attribut est positionné sur **no**, l'unité de bande est moins rapide mais elle garantit une meilleure intégrité des données en cas de coupure de courant ou de défaillance du système et facilite le traitement des fins de support.

Marques de fichier étendues

Lorsque cet attribut est positionné sur **no** (avec **chdev**, l'attribut **extfm**), une marque de fichier standard est inscrite sur la bande chaque fois que nécessaire. La valeur **yes** provoque l'inscription d'une marque de fichier étendue. Pour les unités de bande, cet attribut peut être activé. La valeur par défaut est **no**. Par exemple, les marques de fichiers étendues sur unités de bande 8 mm mobilisent 2,2 Mo et nécessitent pour leur inscription jusqu'à 8,5 secondes. Les marques de fichiers standard utilisent 184 Ko et environ 1,5 secondes.

Lorsque vous utilisez des bandes 8 mm en mode adjonction, il est préférable d'utiliser les marques de fichier étendues pour un meilleur positionnement après des opérations inverses sur marques de fichier. Ceci permet de réduire les risques d'erreur.

Tension

La valeur **yes** (avec **chdev**, l'attribut **ret**) qu'après chaque insertion ou réinitialisation d'une bande, la bande est automatiquement retendue. Cela signifie que la bande est déroulée jusqu'à la fin puis entièrement rembobinée. Cette opération, qui demande plusieurs minutes, diminue le risque d'erreurs. Avec la valeur **no**, l'unité de bande ne retend pas automatiquement la bande. La valeur par défaut est **yes**.

Densité

L'attribut Densité égale à #1 (avec **chdev**, l'attribut **density_set_1**) définit la densité appliquée par l'unité de bande pour l'utilisation de fichiers spéciaux **/dev/rmt*, /dev/rmt*.1**, **/dev/rmt*.2** et **/dev/rmt*.3**. L'attribut Densité égale à #2 (avec **chdev**, l'attribut **density_set_2**) définit la densité appliquée par l'unité de bande pour l'utilisation de fichiers spéciaux **/dev/rmt*.4**, **/dev/rmt*.5**, **/dev/rmt*.6** et **/dev/rmt*.7**. Pour plus de détails, reportez–vous à "Fichiers spéciaux pour unités de bande", page 22-14.

Les attributs de densité sont représentés par des nombres décimaux compris entre **0** et **255**. La valeur **0** demande l'application de la densité par défaut pour l'unité de bande, généralement la densité maximale. Les valeurs admises et leur signification varient en fonction du type d'unité de bande. Ces attributs n'ont aucune répercussion sur la capacité de lecture de l'unité pour des bandes écrites dans des densités admises par l'unité. Habituellement, l'attribut densité égale à #1 est positionné à la valeur maximale possible pour l'unité de bande et l'attribut densité égale à #2, à la seconde valeur maximale possible pour l'unité de bande.

Réservation

Pour les unités de bande qui acceptent cet attribut (avec **chdev**, l'attribut **res_support**), la valeur **yes** réserve l'unité de bande sur le bus SCSI à son ouverture. Lorsque plusieurs cartes SCSI partagent l'unité de bande, l'activation de cet attribut permet de limiter l'accès à une seule carte lorsque l'unité est ouverte. Certaines unités SCSI ne prennent pas en charge cette fonction. D'autres ont une valeur prédéfinie pour cette fonction et la prennent toujours en charge.

Taille de bloc de longueur variable

Cet attribut (avec **chdev**, l'attribut **var_block_size**) spécifie la taille de bloc requise par l'unité de bande lors de l'écriture d'articles de longueur variable. Sur certaines unités de bande SCSI, une taille de bloc non nulle doit être spécifiée (dans les données Mode Select) lors de l'écriture d'articles de longueur variable. La taille de bloc est positionnée à **0** pour indiquer des blocs de longueur variable. Reportez–vous aux informations spécifiques de l'unité de bande SCSI pour déterminer le positionnement requis.

Compression de données

La valeur **yes** de cet attribut (avec **chdev**, l'attribut **compress**) passe l'unité de bande en mode compression, si l'unité offre cette fonction. Dans ce cas, elle inscrit les données sur la bande dans un format compressé pour stocker plus d'informations. La valeur **no** force l'unité de bande à écrire les données en mode natif (non compressé). Cet attribut est sans incidence sur les opérations de lecture. La valeur par défaut est **yes**.

Autochargement

La valeur **yes** de cet attribut (avec **chdev**, l'attribut **autoload**) active la fonction d'autochargement, si l'unité offre cette fonction. Dans ce cas, si la fin de la bande est atteinte lors d'une opération de lecture et d'écriture, la bande suivante est automatiquement chargée pour poursuivre l'opération. Cette fonction est sans incidence sur les commandes applicables uniquement à une seule bande en cartouche. La valeur par défaut est **yes**.

Délai entre deux tentatives

Cet attribut définit le délai d'attente en secondes au-delà duquel le système relance une commande qui n'a pas abouti. Le système peut effectuer quatre tentatives maximum. Cet attribut ne s'applique qu'aux unités de bande de type ost. La valeur par défaut est **45**.

Délai de lecture/écriture

Cet attribut définit le délai maximal (en secondes) accordé au système pour exécuter aevc succès une commande de lecture (READ) ou d'écriture (WRITE). Cet attribut ne s'applique qu'aux unités de bande de type ost. La valeur par défaut est **144**.

Renvoyer erreur sur changement de bande

Lorsque l'attribut Renvoyer erreur sur changement de bande ou réinitialisation est sélectionné, une erreur est renvoyée à l'ouverture lorsque l'unité de bande a été réinitialisée ou que la bande a été changée. Une opération ayant laissé la bande au milieu de la bande à la fermeture doit avoir eu lieu. L'erreur renvoyée est un **–1** et **errno** a la valeur **EIO**. Une fois présentée à l'application, la situation d'erreur est annulée. De même, la reconfiguration de l'unité de bande annule la situation d'erreur.

Attributs pour unités de bande 4 mm 2 Go (type 4mm2gb)

Taille de bloc

La valeur par défaut est **1024**.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité. Les valeurs de densité sont prédéfinies car l'unité de bande écrit toujours en mode 2 Go.

Attributs pour unités de bande 4 mm 4 Go (type 4mm4gb)

Taille de bloc

La valeur par défaut est **1024**.

Mémoires tampon

Reportez–vous aux informations générales fournies pour cet attribut.

Densité

L'utilisateur ne peut pas modifier la densité appliquée par cette unité. L'unité module automatiquement la densité utilisée en fonction du type de support DDS (Digital Data Storage) installé :

Type de support	Configuration de l'unité
DDS	Lecture seulement
DDS ||||	Lecture/écriture en mode 2 Go uniquement
DDS2	Lecture dans l'une ou l'autre des densités, écriture en mode 4 Go seulement
non–DDS	Non pris en charge ; cartouche éjectée

Compression de données

Reportez–vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité.

Attributs pour unités de bande 8 mm 2,3 Go (type 8mm)

Taille de bloc

La valeur par défaut est **1024**. Une valeur inférieure réduit le volume de données stockées sur bande.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité. Les valeurs de densité sont prédéfinies car l'unité de bande écrit toujours en mode 2,3 Go.

Attributs pour unités de bande 8 mm 5 Go (type 8mm5gb)

Taille de bloc

La valeur par défaut est **1024**. Pour une bande inscrite en mode 2,3 Go, une valeur inférieure réduit la quantité de données stockées.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
140	Mode 5 Go (compression possible)
21	Mode 5 Go (compression impossible)
20	Mode 2,3 Go
0	Valeur par défaut (mode 5 Go)

Les valeurs par défaut sont **140** pour l'attribut densité égale à #1 et **20** pour l'attribut densité égale à #2. La valeur **21** associée à l'un de ces attributs autorise la lecture ou l'écriture en mode 5 Go non compressé.

Compression de données

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité.

Attributs pour unités de bande 8 mm 20000 Mo (autoconfiguration)

Taille de bloc

La valeur par défaut est **1024**.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

L'unité peut lire et écrire sur des cartouches de format 20 Go. Pendant la lecture, l'unité détermine automatiquement le format des données inscrites sur la bande. Pendant l'écriture, la valeur de la densité détermine le format des données inscrites sur la bande.

Valeurs possibles :

Valeur	Signification
39	Mode 20 Go (compression possible)
0	Valeur par défaut (mode 20 Go)

La valeur par défaut est **39** pour les attributs densité égale à #1 et densité égale à #2.

Compression de données

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de tension, de réservation, de taille de bloc variable et de densité.

Attributs pour unités de bande 35 Go (type 35gb)

Taille de bloc

La capacité de traitement de l'IBM 7205 Modèle 311 est affectée par la taille de bloc. Pour cette unité, la taille de bloc minimale recommandée est de 32 Ko. Toute valeur inférieure réduit le débit des données (temps de sauvegarde/restauration). Le tableau ci-après répertorie les tailles de bloc recommandées par les commandes AIX :

Commande AIX	Taille de bloc par défaut (octets)	RECOMMENDATION
BACKUP	32 Ko ou 51,2 Ko (par défaut)	32 Ko ou 51,2 Ko, selon que la commande Backup est par nom ou pas. Aucune modification de l'utilisateur n'est requise.
TAR	10 Ko	Il y a erreur dans le manuel qui indique une taille de bloc de 512 Ko. Définissez le paramètre de taille de bloc à **–N64**.
MKSYSB	Voir BACKUP	MKSYSB utilise la commande BACKCUP. Aucune modification de l'utilisateur n'est requise.
DD	n/a	Définissez le paramètre de taille de bloc à **bs=32K**.
CPIO	n/a	Définissez le paramètre de taille de bloc à **–C64**.

Remarque : Vous devez connaître la puissance et la capacité de traitement lorsque vous sélectionnez une taille de bloc. Les tailles de bloc réduites affectent les performances, mais pas la puissance de traitement. Les puissances de traitement des formats 2,6 Go (densité) et 6 Go (densité) sont affectées si vous utilisez une taille de bloc inférieure à la taille recommandée. Par exemple : la sauvegarde de 32 Go dure

environ 22 heures avec une taille de bloc de 1024 octets. La même sauvegarde dure environ 2 heures avec une taille de bloc de 32 Ko.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Le tableau ci–après présente le type de cartouche et les valeurs de densité (décimal et hexadécimal) pris en charge par l'unité de bande IBM 7205–311. Lors d'une opération de restauration (lecture), l'unité règle automatiquement la densité sur celle de l'écriture. Lors d'une opération de sauvegarde (écriture), vous devez régler la densité sur celle de la cartouche de données que vous utilisez.

Cartouches de données prises en charge	Capacité native	Capacité des données compressées	Web-based System Manager ouSMIT	Valeur de densité hexadécimale
DLTtape III	2,6 Go	2,6 Go (sans compression)	23	17h
	6,0 Go	6,0 Go (sans compression)	24	18h
	10,0 Go	20,0 Go (par défaut pour l'unité)	25	19h
DLTtapeIIIxt	15,0 Go	30,6 Go (par défaut pour l'unité)	25	19h
DLTtapeIV	20,0 Go	40,0 Go	26	1Ah
	35,0 Go	70,0 Go (par défaut pour l'unité)	27	1Bh

Remarque : Si vous demandez une capacité native non prise en charge pour la cartouche de données, l'unité utilise la puissance de traitement maximale prise en charge pour la cartouche chargée dans l'unité.

Compression de données

La compression réelle dépend du type de données écrites. (voir le tableau ci-dessus) Un rapport de compression de 2/1 est adopté pour cette capacité des données compressées.

Attributs à valeur fixe

Reportez-vous aux informations générales fournies pour cet attribut.

Attributs pour unités de bande 1/4 pouce 150 Mo (type 150mb)

Taille de bloc

La taille de bloc par défaut est **512**. Pour les blocs de longueur variable, la seule taille de bloc possible est **0**.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

L'écriture sur une bande 1/4 pouce ne peut être effectuée qu'en début de bande (BOT) ou sur bande vierge. Si la bande contient des données, vous ne pouvez écraser les données qu'à partir du début de la bande. Pour ajouter des données sur une bande non vide et rembobinée, vous devez la faire dérouler jusqu'à la marque de fichier suivante (signalée par le système lorsqu'elle est détectée par un message d'erreur). Vous pouvez alors reprendre les opérations d'écriture.

Tension

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
16	QIC–150
15	QIC–120
0	Valeur par défaut (QIC–150) ou dernière valeur de densité utilisée par le système.

Les valeurs par défaut sont **16** pour l'attribut densité égale à #1 et **15** pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de réservation, de taille de bloc variable et de compression.

Attributs pour unités de bande 1/4 pouce 525 Mo (type 525mb)

Taille de bloc

La taille de bloc par défaut est **512**. Les autres valeurs possibles sont **0** pour des blocs de longueur variable et **1024**.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

L'écriture sur une bande 1/4 pouce ne peut être effectuée qu'en début de bande (BOT) ou sur bande vierge. Si la bande contient des données, vous ne pouvez écraser les données qu'à partir du début de la bande. Pour ajouter des données sur une bande non vide et rembobinée, vous devez la faire dérouler jusqu'à la marque de fichier suivante (signalée par le système lorsqu'elle est détectée par un message d'erreur). Vous pouvez alors reprendre les opérations d'écriture.

Tension

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
17	QIC–525*
16	QIC–150
15	QIC–120
0	Valeur par défaut (QIC–525) ou dernière valeur de densité utilisée par le système.

* QIC–525 est le seul mode qui accepte une taille de bloc de 1024.

Les valeurs par défaut sont **17** pour l'attribut densité égale à #1 et **16** pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de réservation, de taille de bloc variable et de compression.

Attributs pour unités de bande 1/4 pouce 1200 Mo (type 1200mb–c)

Taille de bloc

La taille de bloc par défaut est **512**. Les autres valeurs possibles sont **0** pour des blocs de longueur variable et **1024**.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

L'écriture sur une bande 1/4 pouce ne peut être effectuée qu'en début de bande (BOT) ou sur bande vierge. Si la bande contient des données, vous ne pouvez écraser les données qu'à partir du début de la bande. Pour ajouter des données sur une bande non vide et rembobinée, vous devez la faire dérouler jusqu'à la marque de fichier suivante (signalée par le système lorsqu'elle est détectée par un message d'erreur). Vous pouvez alors reprendre les opérations d'écriture.

Tension

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
21	QIC–1000*
17	QIC–525*
16	QIC–150
15	QIC–120
0	Valeur par défaut (QIC–1000) ou dernière valeur de densité utilisée par le système.

* QIC–525 et QIC–1000 sont les seuls modes qui acceptent une taille de bloc de 1024.

Les valeurs par défaut sont **21** pour l'attribut densité égale à #1 et **17** pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de réservation, de taille de bloc variable et de compression.

Attributs pour unités de bande 4 mm 12 000 Mo (autoconfiguration)

Taille de bloc

La capacité de traitement de l'IBM 12 000 Mo 4 mm est affectée par la taille de bloc. Pour cette unité, la taille de bloc minimale recommandée est de 32 Ko. Toute valeur inférieure réduit le débit des données (temps de sauvegarde/restauration). Le tableau ci–après répertorie les tailles de bloc recommandées par les commandes AIX :

Commande AIX	Taille de bloc par défaut (octets)	RECOMMENDATION
BACKUP	32 Ko ou 51,2 Ko (par défaut)	32 Ko ou 51,2 Ko, selon que la commande Backup est par nom ou pas. Aucune modification de l'utilisateur n'est requise.
TAR	10 Ko	Il y a erreur dans le manuel qui indique une taille de bloc de 512 Ko. Définissez le paramètre de taille de bloc à **–N64**.
MKSYSB	Voir BACKUP	MKSYSB utilise la commande BACKCUP. Aucune modification de l'utilisateur n'est requise.
DD	n/a	Définissez le paramètre de taille de bloc à **bs=32K**
CPIO	n/a	Définissez le paramètre de taille de bloc à **–C64**.

Remarque : Vous devez connaître la puissance et la capacité de traitement lorsque vous sélectionnez une taille de bloc. Les tailles de bloc réduites affectent les performances, mais pas la puissance de traitement.

Mémoires tampon

Reportez–vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez–vous aux informations générales fournies pour cet attribut.

Densité

Le tableau ci–après présente le type de cartouche et les valeurs de densité (décimal et hexadécimal) pris en charge par l'unité de bande IBM 12 000 Mo 4 mm. Lors d'une opération de restauration (lecture), l'unité règle automatiquement la densité sur celle de l'écriture. Lors d'une opération de sauvegarde (écriture), vous devez régler la densité sur celle de la cartouche de données que vous utilisez.

Cartouches de données prises en charge	Capacité native	Capacité des données compressées	Valeur de densité Web-based System Manager ou SMIT	Valeur de densité hexadécimale
DDS III	2,0 Go	4,0 Go	19	13h
DDS2	4,0 Go	8,0 Go	36	24h
DDS3	12,0 Go	24,0 Go	37	25h

Remarque : Si vous demandez une capacité native non prise en charge pour la cartouche de données, l'unité utilise la puissance de traitement maximale prise en charge pour la cartouche chargée dans l'unité.

Compression de données

La compression réelle dépend du type de données écrites. (voir le tableau ci–dessus) Un rapport de compression de 2/1 est adopté pour cette capacité des données compressées.

Attributs à valeur fixe

Reportez–vous aux informations générales fournies pour cet attribut.

Attributs pour unités de bande 1/4 pouce 13 000 Mo (autoconfiguration)

Taille de bloc

La taille de bloc par défaut est **512**. Les autres valeurs possibles sont **0** pour des blocs de longueur variable et **1024**.

Mémoires tampon

Reportez–vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

L'écriture sur une bande 1/4 pouce ne peut être effectuée qu'en début de bande (BOT) ou sur bande vierge. Si la bande contient des données, vous ne pouvez écraser les données qu'à partir du début de la bande. Pour ajouter des données sur une bande non vide et rembobinée, vous devez la faire dérouler jusqu'à la marque de fichier suivante (signalée par le système lorsqu'elle est détectée par un message d'erreur). Vous pouvez alors reprendre les opérations d'écriture.

Tension

Reportez–vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
33	QIC–5010–DC*
34	QIC–2GB*
21	QIC–1000*
17	QIC–525*
16	QIC–150
15	QIC–120
0	Valeur par défaut (QIC–5010–DC)*

* QIC–525, QIC–1000, QIC–5010–DC et QIC–2GB sont les seuls modes qui acceptent une taille de bloc de 1024.

Les valeurs par défaut sont **33** pour l'attribut densité égale à #1 et **34** pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de réservation et de taille de bloc variable.

Attributs pour unités de bande 9 pistes 1/2 pouce (type 9trk)

Taille de bloc

La valeur par défaut est **1024**.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

Valeurs possibles :

Valeur	Signification
3	6 250 bits par pouce (bpp)
2	1 600 bpp
0	Densité précédemment utilisée

Les valeurs par défaut sont **3** pour l'attribut densité égale à #1 et **2** pour l'attribut densité égale à #2.

Attributs à valeur fixe

Pour les unités de bande déclarées de ce type, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de tension, de réservation, de taille de bloc variable et de compression.

Attributs pour cartouche 1/2 pouce 3490e (type 3490e)

Taille de bloc

La valeur par défaut est **1024**. Cette unité offre un débit de transfert de données élevé et la taille de bloc peut se révéler critique pour certaines opérations. La vitesse d'exploitation peut être sensiblement améliorée avec des blocs de grande taille. De façon générale, il est conseillé d'opter pour la plus grande taille de bloc possible.

> **Remarque :** Augmenter la taille de bloc peut entraîner des incompatibilités avec d'autres programmes installés sur le système. Dans ce cas, vous en êtes averti lors de l'exécution des programmes concernés par le message :
>
> ```
> Un appel système a reçu un paramètre incorrect.
> ```

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Compression

Reportez-vous aux informations générales fournies pour cet attribut.

Autochargement

Cette unité est équipée d'un séquenceur de bande, fonction d'autochargement qui charge et éjecte séquentiellement les cartouches de bande d'une série à partir d'un chargeur de

cartouche Pour cette opération, le commutateur situé sur le panneau avant de l'unité doit être positionné sur AUTO et l'attribut d'autochargement sur **yes**.

Attributs pour autres bandes SCSI (type ost)

Taille de bloc

La valeur par défaut est **512**, mais elle peut être ajustée à la taille de bloc par défaut de votre unité de bande. Les valeurs les plus courantes sont **512** et **1024**. Les unités de bande 8 et 4 mm utilisent généralement une taille de bloc de **1024**. L'espace sur bande est mal exploité si l'attribut taille de bloc est laissé à **512**. La valeur **0** indique une taille de bloc variable sur certaines unités.

Mémoires tampon

Reportez-vous aux informations générales fournies pour cet attribut.

Marques de fichier étendues

Reportez-vous aux informations générales fournies pour cet attribut.

Densité

La valeur par défaut est **0** pour les deux densités. Les valeurs possibles et leur signification varient en fonction du type d'unité de bande.

Réservation

La valeur par défaut est **no**. Elle peut être basculée sur **yes** si l'unité accepte la fonction de réservation. En cas de doute, conservez la valeur **no**.

Taille de bloc de longueur variable

La valeur par défaut est **0**. Les valeurs non nulles sont utilisées sur des unités QIC (Quarter Inch Cartridge). Pour plus de précisions, reportez-vous aux informations relatives à votre unité de bande.

Délai entre deux tentatives

Cet attribut ne s'applique qu'aux unités de bande de type ost.

Délai de lecture/écriture

Cet attribut ne s'applique qu'aux unités de bande de type ost.

Attributs à valeur fixe

Pour les unités de bande déclarées de type ost, des valeurs prédéfinies non modifiables sont affectées aux attributs de marques de fichier étendues, de tension et de compression.

Fichiers spéciaux pour unités de bande

L'écriture et la lecture de fichiers sur bande se fait à l'aide de fichiers spéciaux **rmt**. Plusieurs types de fichiers spéciaux sont associés à chaque unité de bande connue du système d'exploitation. Ces fichiers sont **/dev/rmt***, **/dev/rmt*.1**, **/dev/rmt*.2**, ... **/dev/rmt*.7** où **rmt*** représente le nom logique d'une unité de bande, par exemple **rmt0** ou **rmt1**.

Sélectionner l'un de ces fichiers spéciaux revient à choisir le mode d'exécution des opérations d'E/S sur l'unité de bande.

Densité Vous pouvez opter pour une densité égale à #1 ou à #2. Ces densités sont définies dans les attributs de l'unité de bande. La densité maximale possible est généralement attribuée à la densité #1 et la valeur maximale suivante possible à la densité #2. C'est pourquoi, par abus de langage, les fichiers spéciaux utilisant la densité égale à #1 sont parfois assortis du qualificatif Haute densité et les fichiers spéciaux utilisant la densité égale à #2 du qualificatif Faible densité. Lors de la lecture de la bande, le paramètre de densité est ignoré.

Rembobinage à la fermeture Vous pouvez demander le rembobinage automatique complet de la bande à la fermeture du fichier spécial relatif à l'unité de bande. Dans ce cas, le positionnement en début de bande est intégré au processus de fermeture du fichier.

Tension à l'ouverture Vous pouvez demander que la bande soit retendue à l'ouverture du fichier, c'est-à-dire déroulée jusqu'à la fin, puis entièrement rembobinée. Cette précaution réduit le risque d'erreurs. Dans ce cas, le positionnement en début de bande est intégré au processus d'ouverture du fichier.

Le tableau ci-dessous donne la liste des fichiers spéciaux **rmt** et de leurs caractéristiques.

Fichier spécial	Rembobinage à la fermeture	Tension à l'ouverture	Densité
/dev/rmt*	oui	non	#1
/dev/rmt*.1	non	non	#1
/dev/rmt*.2	oui	oui	#1
/dev/rmt*.3	non	oui	#1
/dev/rmt*.4	oui	non	#2
/dev/rmt*.5	non	non	#2
/dev/rmt*.6	oui	oui	#2
/dev/rmt*.7	non	oui	#2

Si, par exemple, vous souhaitez écrire trois fichiers sur bande dans l'unité de bande **rmt2**, le premier en début de bande et les deux autres à la suite, avec la densité égale à #1 pour l'unité de bande, vous pouvez utiliser, dans l'ordre, les fichiers spéciaux suivants :

1. `/dev/rmt2.3`
2. `/dev/rmt2.1`
3. `/dev/rmt2`

Explication :

- Le fichier `/dev/rmt2.3` est choisi comme premier fichier car il est doté de l'option de rembobinage à l'ouverture qui garantit l'écriture du premier fichier en début de bande. L'option de rembobinage à la fermeture n'est pas retenue car l'opération d'E/S suivante doit commencer à la fin de ce fichier. Si la bande est déjà positionnée au début, l'utilisation du fichier `/dev/rmt2.1` comme premier fichier se révèle plus rapide, la phase de retension de la bande étant omise.
- Le fichier `/dev/rmt2.1` est choisi comme deuxième fichier car il ne comporte ni l'option de retension à l'ouverture, ni l'option de rembobinage à la fermeture. Or, le repositionnement en début de bande à l'ouverture ou à la fermeture du fichier est inutile.
- Le fichier `/dev/rmt2` est choisi comme troisième et dernier fichier car l'option de retension à l'ouverture n'est pas souhaitée, ce fichier étant précédé du deuxième fichier. En revanche, l'option de rembobinage à la fermeture est sélectionnée car aucune opération d'écriture n'est prévue à la suite du troisième fichier. La prochaine utilisation de la bande commencera au début de la bande.

Le choix du fichier spécial **rmt** n'est pas le seul moyen de contrôle des opérations sur bande ; vous disposez également de la commande **tctl**.

Annexe A. AIX pour administrateurs système BSD

Cette annexe s'adresse aux administrateurs familiers des systèmes d'exploitation Unix BSD 4.3 ou System V. Elle traite des différences et des ressemblances entre ces systèmes et AIX.

Les sujets abordés sont les suivants :

AIX pour administrateurs système BSD - généralités

Les sections suivantes sont d'ordre général :

Les sections suivantes entrent dans le détail des tâches d'administration système :

Introduction à AIX pour administrateurs système BSD

Voici quelques conseils qui vous aideront à démarrer l'administration du système :

- Commencez par vous connecter, en tant qu'utilisateur racine, sur la console graphique.
- Exécutez les tâches de gestion à partir de la console système tant que vous n'êtes pas complètement à l'aise avec le système : il est plus simple de travailler depuis cette console qu'à partir d'un terminal distant. Une fois qu'AIX n'aura plus de secret pour vous, vous pourrez sans problème travailler à distance depuis un terminal xterm ou ASCII.
- Plusieurs utilitaires AIX sont proposés pour la gestion du système. Ces utilitaires sont les suivants :
  - SMIT (System Management Interface Tool) : fournit une interface entre l'administrateur et les commandes de configuration et de gestion. SMIT facilite l'exécution de nombreuses tâches d'administration. Pour en savoir plus, reportez-vous à "SMIT (System Management Interface Tool) - généralités", page 17-2.
  - Le gestionnaire ODM (Object Data Manager) : fournit des routines d'accès aux objets des bases de données ODM. Ces bases contiennent des informations sur la configuration des unités. Pour en savoir plus, reportez-vous à "Unités - généralités", page 21-1.
  - Le contrôleur SRC (System Resource Controller) : donne accès et contrôle les démons et aux autres ressources système, et permet le contrôle, via une interface unique. Pour en savoir plus, reportez-vous à "Contrôleur SRC – généralités", page 14-2.

Principales différences entre BSC 4.3 et AIX

Cet article récapitule les principales différences entre les systèmes AIX et BSD 4.3. Pour en savoir plus, reportez-vous à la liste des articles dans ”AIX pour administrateurs systèmes BSD - généralités”, page A-2.

Stockage des données de configuration

BSD 4.3 stocke généralement les données de configuration dans des fichiers ASCII. Les informations apparentées se trouvent sur une même ligne et le traitement des enregistrements (tri et recherche) peut être effectué sur le fichier ASCII lui-même. Ces enregistrements, de longueur variable, sont terminés par un saut de ligne. BSD 4.3 offre des outils permettant de convertir les fichiers ASCII volumineux en format base de données (dbm). Les fonctions de bibliothèque correspondantes explore la paire de fichiers dbm s’ils existent, ou, dans le cas contraire, le fichier ASCII d’origine.

Certaines données de configuration AIX sont stockées dans des fichiers ASCII, mais le plus souvent sous forme de *strophes*. Une strophe est un ensemble d’éléments d’information apparentés, stockés dans un groupe de lignes. Chaque élément est doté d’une étiquette, simplifiant l’appréhension du contenu du fichier.

AIX prend également en charge les versions dbm des mots de passe et des informations utilisateur. Les fichiers **/etc/passwd**, **/etc/group** et **/etc/inittab** sont en outre des exemples de fichiers AIX où les informations sont stockées sous forme traditionnelle et non sous forme de strophes.

Les autres données de configuration AIX sont stockées dans des fichiers maintenus par le gestionnaire d’objet ODM (Object Data Manager). Web-based System Manager ou SMIT (System Management Interface Tool) peut manipuler et afficher les informations des fichiers ODM. Vous pouvez également faire appel directement aux commandes ODM pour visualiser ces fichiers. Pour interroger les fichiers ODM, vous disposez des commandes :

- **odmget,**
- **odmshow.**

Pour modifier ces fichiers, des commandes :

- **odmadd,**
- **odmcreate,**
- **odmdrop,**
- **odmchange,**
- **odmdelete.**

Attention : Modifier les fichiers ODM de manière incorrecte peut provoquer l’arrêt du système, avec impossibilité de le relancer. Ne faites pas appel aux commandes ODM que si des commandes spécifiques (telles que celles générées par SMIT ou Web-based System Manager) échouent.

Gestion de la configuration

Au démarrage d’un système AIX, un ensemble de commandes de configuration sont appelées par le gestionnaire de configuration. Ces commandes sont appelées *méthodes*. Elles identifient les unités du système et mettent à jour les fichiers ODM appropriés dans le répertoire **/etc/objrepos**.

Les fichiers unité spéciaux du répertoire **/dev** ne sont pas préinstallés. Certains fichiers spéciaux (fichiers disque, par exemple) sont créés automatiquement au cours du processus de configuration du démarrage. D’autres fichiers spéciaux (ceux des terminaux ASCII, par exemple) doivent être créés par l’administrateur système par le biais du menu Unités de SMIT ou de l’application Web-based System Manager Devices. Ces informations sont conservées dans ODM en vue d’un usage ultérieur.

Gestion de disque

Sous AIX, les unités de disque sont des *volumes physiques*. Les partitions forment des *volumes logiques*. Comme dans BSD 4.3, un volume physique peut être associé à plusieurs volumes logiques. Mais, contrairement à BSD 4.3, un seul volume AIX peut s'étendre sur plusieurs volumes physiques. Pour ce faire, il faut regrouper les volumes physiques dans un *groupe de volumes* et créer les volumes logiques sur ce groupe.

AIX Voici quelques-unes des commandes relatives aux systèmes de fichiers et à la gestion des volumes :

- **crfs**
- **varyonvg**
- **varyoffvg**
- **lsvg**
- **importvg**
- **exportvg**

Les commandes BSD 4.3 suivantes sont également disponibles :

- **mkfs**
- **fsck**
- **fsdb**
- **mount**
- **umount**

Les différences entre la version BSD 4.3 et la version AIX de ces commandes sont explicitées à la section "Systèmes de fichiers pour administrateurs système BSD 4.3", page A-19.

BSD 4.3 maintient la liste des systèmes de fichiers dans le fichiers **/etc/fstab**. AIX maintient une strophe pour chaque système de fichiers dans le fichier **/etc/filesystems**.

Le système de fichiers BSD 4.3 lit généralement par grands blocs de 8 ko, mais peut enregistrer plusieurs petits fichiers dans des *fragments* généralement de 1 ko. Le système de fichiers AIX ne prend pas en charge les fragments et chaque fichier consomme au moins un bloc. La taille de bloc est généralement de 4 ko.

Nouvelles commandes

Pour prendre en charge les nouveaux systèmes de configuration et de gestion de disque, AIX propose maintenant une cinquantaine de commandes nouvelles (nouvelles aussi pour les administrateurs BSC 4.3). Pour en savoir plus, reportez-vous à "Commandes d'administration de AIX pour administrateurs système BSD 4.3", page A-11.

Amorçage et lancement

AIX prend en charge l'identification et la configuration automatiques des unités. En conséquence, le processus d'amorçage et de lancement diffère sensiblement du processus BSD 4.3. Outre le noyau, une image d'un système de fichiers d'amorçage, ainsi que les données de configuration (des unités) antérieures, sont chargés sur un disque RAM. Au cours de la première phase du lancement, un nombre de données de configuration suffisant pour permettre l'accès aux volumes logiques est chargé et vérifié. L'unité d'espace de pagination est identifiée auprès du noyau et le système de fichiers racine du disque est vérifié. AIX déplace alors le système de fichiers racine du disque RAM vers le disque dur, et achève la procédure de lancement, en configurant notamment les autres unités.

Autorisation utilisateur

BSD 4.3, et les systèmes UNIX version AT&T antérieures à SVR4, stockent toutes les données d'authentification utilisateur (mots de passe chiffrés compris) dans le fichier **/etc/passwd**. Normalement, ce fichier est lisible par tous.

Sur les systèmes SVR4, les mots de passe chiffrés ne se trouvent plus dans le fichier **/etc/passwd**, mais dans le fichier **/etc/shadow**. Ce fichier est accessible aux seuls utilisateurs racine et aux programmes sécurisés (**/bin/login** par exemple).

AIX enregistre les mots de passe chiffrés dans le fichier **/etc/security/passwd**. Le répertoire **/etc/security** contient deux autres fichiers,**user** et **limits**. Ces trois fichiers déterminent les droits d'accès d'un utilisateur au système (accès aux commandes **rlogin** ou **telnet**, par exemple) et les limites des ressources utilisateur (taille de fichier, espace d'adressage, etc.).

Impression

La plupart des commandes d'impression BSD 4.3 sont acceptées. Parmi les différences, minimes, notez que **/etc/qconfig** est le fichier de configuration AIX.

Les systèmes d'impression ligne AIX et BSC 4.3 peuvent coopérer : il est possible de soumettre des travaux à des systèmes BSD 4.3 et d'imprimer des travaux soumis par des systèmes BSC 4.3.

Shells

AIX prend en charge les shells Bourne, C et Korn. Le chemin d'accès complet au shell Bourne est **/bin/bsh**. Le fichier **/bin/sh** est un lien fixe au fichier **/bin/ksh**. Ce fichier est modifiable par l'administrateur.

Remarques :

1. AIX ne dispose pas de scripts shell basés sur **/bin/sh**. Mais de nombreux scripts d'autres systèmes sont basés sur **/bin/sh** (le shell Bourne).
2. Malgré la similarité des shells Bourne et Korn, le shell Korn n'est pas exactement un surensemble du shell Bourne.

Comptabilité pour administrateurs système BSD 4.3

Les fichiers de comptabilité AIX du répertoire **/usr/lib/acct** et les outils de suivi de l'activité système du répertoire **/usr/lib/sa** sont identiques à ceux de AT&T System V Release 4 (SVR4) combinés avec les utilitaires de comptabilité BSC 4.3.

La plupart des commandes de comptabilité se trouvent dans le répertoire **/usr/lib/acct**. Pour démarrer le système de comptabilité, exécutez la commande **/usr/lib/acct/startup**. Sinon, aucune commande de comptabilité (**lastcomm** (1), par exemple) ne renverra d'informations.

AIX fournit les fonctions de compatibilité BSC 4.3 suivantes :

last(1)	Indique les dernières connexions utilisateur et terminale.
lastcomm(1)	Affiche, dans l'ordre inverse, les dernières commandes exécutées.
acct(3)	Active/désactive la comptabilité des processus.
ac(8)	Comptabilité de connexion.
accton(8)	Active/désactive la comptabilité système.
sa(8)	Maintient les fichiers de comptabilité système.

AIX fournit également les commandes de comptabilité SVID (System V Interface Definition) Issue II et les fonctions de bibliothèque suivantes :

acctcms(1)	Génère un rappel de la syntaxe des commandes pour les enregistrements de comptabilité.
acctcms(1)	Affiche les récapitulatifs sélectionnés des enregistrements de comptabilité des processus.
acctcon1(1)	Convertit les enregistrements de connexion/déconnexion en enregistrements de session.
acctcon2(1)	Convertit les enregistrements de connexion/déconnexion en enregistrements de cumul.
acctdisk(1)	Génère des enregistrements de cumul à partir des résultats de la commande **diskusg** (1).
acctmerg(1)	Fusionne des fichiers de cumul dans un fichier intermédiaire.
accton(1)	Active le système de comptabilité.
acctprc1(1)	Traite les données comptables issues de la commande **acct** (3).
acctprc2(1)	Traite les résultats de la commande **acctprc1** (1) dans des enregistrements de cumul.
acctwtmp(1)	Manipule les enregistrements de durée de connexion.
chargefee(1)	Impute au nom de connexion.
ckpacct(1)	Contrôle la taille du fichier **/usr/adm/pacct**.
diskusg(1)	Génère des données comptables relatives au disque.
dodisk(1)	Effectue des opérations comptables sur le disque.
fwtmp(1)	Convertit des enregistrements binaires (fichier **wtmp**) en enregistrements ASCII formatés. **Remarque :** Le fichier **wtmp** se trouve dans le répertoire **/var/adm**.
lastlogin(1)	Met à jour les dates de dernière connexion de chaque utilisateur.

monacct(1)	Crée des fichiers récapitulatifs mensuels.
prctmp(1)	Imprime le fichier d'enregistrement de session issu de la commande **acctcon1**(1).
prdaily(1)	Formate l'état comptable de la veille.
prtacct(1)	Formate et imprime un fichier de cumul comptable.
runacct(1)	Exécute la comptabilité quotidienne.
shutacct(1)	Appelée par la commande d'arrêt du système (shutdown) pour interrompre la comptabilité et en consigner la cause.
startup(1)	Appelée par l'initialisation du système pour démarrer la comptabilité.
turnacct(1)	Active/désactive la comptabilité des processus.
wtmpfix(1)	Corrige l'horodate dans un fichier avec le format **wtmp**.

Sauvegarde pour administrateurs système BSD 4.3

Les commandes **tar** et **cpio** peuvent transférer des données d'un système à un autre. La commande **tar** d'AIX n'est pas totalement compatible avec la commande **tar** de BSD 4.3. La commande **tar** d'AIX requiert l'option **–B** (entrée bloquée) si la lecture est effectuée à partir d'un tube. La version AT&T de la commande **cpio** est compatible avec la version AIX.

AIX peut lire et écrire le format des commandes **dump** et **restore**. Par exemple, la commande AIX **backup** avec la syntaxe :

```
backup –0uf Unité SystèmeFichiers
```

équivaut à la commande BSC 4.3 **dump** avec la syntaxe :

```
dump 0uf Unité SystèmeFichiers
```

De même, la commande AIX **restore** avec la syntaxe :

```
restore –mivf Unité
```

équivaut à la commande BSD 4.3 **restore** avec la syntaxe :

```
restore ivf Unité
```

AIX propose également les commandes BSD 4.3 **rdump** et **rrestore**. La seule différence est que, sous AIX, chaque argument doit être précédé d'un **–** (tiret). Par exemple, la commande :

```
rdump –0 –f orca:/dev/rmt0 /dev/hd2
```

équivaut à la commande BSD 4.3 :

```
rdump 0f orca:/dev/rmt0 /dev/hd2
```

La commande AIX **backup** avec la syntaxe :

```
backup –0f /dev/rmt0 /dev/hd2
```

équivaut à la commande BSD 4.3 **dump** avec la syntaxe :

```
dump 0f /dev/rmt0 /dev/hd2
```

Support de bande SCSI non IBM

AIX ne prend pas directement en charge les dérouleurs de bande SCSI non IBM. Mais vous pouvez y ajouter vos propres en-tête et interface utilisant le dérouleur SCSI IBM. Pour en savoir plus, reportez-vous à Adding an unsupported device to the system dansdans *AIX Kernel Extensions and Device Support Programming Concepts*.

Pour en savoir plus, reportez-vous à "Sauvegarde - généralités", page 9-2.

Amorçage et lancement pour administrateurs système BSD 4.3

Sur les systèmes BSD 4.3, le programme **init** est la dernière étape de la procédure d'amorçage. Le rôle essentiel de ce programme est de créer des processus pour chaque port de terminal disponible. Les ports de terminal disponibles sont repérables en lisant le fichier **/etc/ttys**.

Sur un système System V, le programme **init** est démarré à l'initialisation du système. Le processus **init** lance les processus en fonction des entrées du fichier **/etc/inittab**.

AIX adopte la procédure d'initialisation de System V. Vous pouvez éditer le fichier AIX **/etc/inittab**, directement via la commande **telinit** ou par le biais d'une des commandes AIX suivantes :

chitab(1)	modification d'article(s) dans le fichier **/etc/inittab**.
lsitab(1)	Affiche la liste des enregistrements du fichier **/etc/inittab**.
mkitab(1)	Crée des enregistrements dans le fichier **/etc/inittab**.
rmitab(1)	Supprime des enregistrements du fichier **/etc/inittab**.

Les modifications apportées au fichier **/etc/inittab** prennent effet au réamorçage suivant du système ou après exécution de la commande **telinit q**.

Commandes d’administration d’AIX pour administrateurs système BSD 4.3

Voici la liste des commandes propres à l’administration de l’environnement AIX.

bosboot(1)	Initialise une unité d’amorçage.
bootlist(1)	Modifie la liste des unités d’amorçage (ou leur ordre dans cette liste) disponibles pour le système.
cfgmgr(1)	Configure des unités en exécutant les programmes du répertoire **/etc/methods**.
chcons(1)	Réachemine la console système vers une unité ou un fichier – effectif au démarrage suivant.
chdev(1)	Modifie les caractéristiques d’une unité.
chdisp(1)	Change l’écran utilisé par le sous-système LFT (low-function terminal).
checkcw(1)	Prépare le texte en espacement fixe pour la commande **troff**.
checkeq(1)	Vérifie les documents formatés par les macros de type memorandum.
checkmm(1)	Vérifie les documents formatés par les macros de type memorandum.
checknr(1)	Contrôle les fichiers **nroff** et **troff**.
chfont(1)	Change la police par défaut sélectionnée au moment de l’amorçage.
chfs(1)	Modifie les attributs d’un système de fichiers.
chgroup(1)	Modifie les attributs des groupes.
chgrpmem(1)	Modifie les administrateurs ou les membres d’un groupe.
chhwkbd(1)	Modifie les attributs du clavier LFT (low-function terminal) enregistrés dans la base de données ODM (Object Data Manager).
chitab(1)	Modifie des articles dans le fichier **/etc/inittab**.
chkbd(1)	Modifie la mappe clavier par défaut utilisée par le LFT (low-function terminal) au moment de l’amorçage.
chkey(1)	Modifie votre clé de chiffrement.
chlang	Définit la variable d’environnement **LANG** dans le fichier **/etc/environment** pour la connexion suivante.
chlicense(1)	Il existe deux types de licence utilisateur: Les licences fixes sont toujours activées, leur nombre pouvant être modifié via l’option **–u**. Les licences flottantes sont activées ou désactivées via l’option **–f**.
chlv(1)	Modifie les caractéristiques d’un volume logique.
chnamsv(1)	Modifie la configuration d’un service de noms TCP/IP sur un hôte.
chprtsv(1)	Modifie la configuration d’un service d’impression sur une machine client ou serveur.
chps(1)	Modifie les attributs d’un espace de pagination.
chpv(1)	Modifie les caractéristiques d’un volume physique dans un groupe de volumes.
chque(1)	Modifie le nom de la file d’attente.
chquedev(1)	Change le nom d’unité de l’imprimante ou du traceur.
chssys(1)	Modifie la définition d’un sous–système dans la classe d’objets sous-système.
chtcb(1)	Modifie ou interroge l’attribut TCB (Trusted Computing Base) d’un fichier.

chtz	Modifie les informations relatives au temps système.
chuser(1)	Modifie les attributs d'un utilisateur.
chvfs(1)	Modifie des articles dans le fichier **/etc/vfs**.
chvg(1)	Définit les caractéristiques d'un groupe de volumes.
chvirprt(1)	Modifie la valeur des attributs d'une imprimante virtuelle.
crfs(1)	Ajoute un système de fichiers.
crvfs(1)	Crée des entrées dans le fichier **/etc/vfs**.
exportvg(1)	Exporte la définition d'un groupe de volumes à partir d'un ensemble de volumes physiques.
extendvg(1)	Ajoute des volumes physiques à un groupe de volumes.
grpck(1)	Vérifie une définition de groupe.
importvg(1)	Importe une nouvelle définition de groupe de volumes à partir d'un ensemble de volumes physiques.
lsallq(1)	Affiche la liste de toutes les files d'attente configurées.
lsallqdev(1)	Affiche la liste de toutes les files d'attente d'imprimante et de traceur configurées dans une file d'attente donnée.
lsdisp(1)	Affiche la liste des écrans disponibles sur le système.
lsfont(1)	Affiche la liste des polices disponibles sur l'écran.
lsfs(1)	Affiche les caractéristiques de systèmes de fichiers.
lsgroup(1)	Affiche les attributs de groupes.
lsitab(1)	Affiche la liste des enregistrements du fichier **/etc/inittab**.
lskbd(1)	Affiche la liste des mappes clavier disponibles pour le sous-système LFT (low-function terminal).
lslicense(1)	Affiche le nombre de licences fixes et l'état des licences flottantes.
lslpp(1)	Affiche la liste des logiciels en option.
lsnamsv(1)	Affiche les informations sur le service de noms, enregistrées dans la base de données.
lsprtsv(1)	Affiche les informations sur le service d'impression, enregistrées dans la base de données.
lsps	Affiche l'espace de pagination et ses attributs.
lsque(1)	Affiche le nom de la strophe de file d'attente.
lsquedev(1)	Affiche le nom de la strophe d'unité.
lssrc(1)	Récupère l'état d'un sous-système, d'un groupe de sous-systèmes ou d'un sous-serveur.
lsuser(1)	Affiche les attributs des comptes utilisateur.
lsvfs(1)	Affiche la liste des entrées dans le fichier **/etc/vfs**.
mkcatdefs(1)	Prétraite un fichier source de messages.
runcat(1)	Etablit un tube du résultat de la commande **mkcatdefs** pour la commande **gencat**.
mkdev(1)	Ajoute une unité au système.
mkfont(1)	Ajoute au système le code de police associé à un écran.
mkfontdir(1)	Crée un fichier **fonts.dir** à partir d'un répertoire de fichiers de police.
mkgroup(1)	Crée un groupe.
mkitab(1)	Crée des enregistrements dans le fichier **/etc/inittab**.
mklv(1)	Crée un volume logique.

mklvcopy(1)	Ajoute des copies à un volume logique.
mknamsv(1)	Configure un service de noms TCP/IP sur un hôte pour un client.
mknotify(1)	Ajoute une définition de méthode de notification à la classe d'objets de notification.
mkprtsv(1)	Modifie la configuration d'un service de noms TCP/IP sur un hôte.
mkps(1)	Ajoute un espace de pagination au système.
mkque(1)	Ajoute une file d'attente d'impression au système.
mkquedev(1)	Ajoute une unité de file d'attente d'impression au système.
mkserver(1)	Ajoute une définition de sous-serveur à la classe d'objets sous-serveur.
mkssys(1)	Ajoute une définition de sous-système à la classe d'objets sous-système.
mksysb	Sauvegarde des systèmes de fichiers montés dans le groupe de volumes **rootvg** pour les réinstallations ultérieures.
mkszfile	Enregistre la taille des systèmes de fichiers montés dans le groupe de volumes **rootvg** pour les réinstallations ultérieures.
mktcpip(1)	Définit les valeurs requises pour démarrer TCP/IP sur un hôte.
mkuser(1)	Crée un compte utilisateur.
mkuser.sys(1)	Personnalise un nouveau compte utilisateur.
mkvg(1)	Crée un groupe de volumes.
mkvirprt(1)	Crée une imprimante virtuelle.
odmadd(1)	Ajoute des objets aux classes d'objets créées.
odmchange(1)	Modifie le contenu d'un objet sélectionné dans la classe spécifiée.
odmcreate(1)	Génère les fichiers **.c** (source) et **.h** (en-tête) requis pour le développement d'application ODM et crée des classes d'objets vides.
odmdelete(1)	Supprime les objets sélectionnés de la classe d'objets spécifiée.
odmdrop(1)	Supprime une classe d'objets.
odmget(1)	Extrait des objets des classes spécifiées et les place dans le fichier d'entrée **odmadd**.
odmshow(1)	Affiche la définition d'une classe d'objets.
pwdck(1)	Vérifie les informations d'authentification locale.
redefinevg	Redéfinit l'ensemble de volumes physique d'un groupe de volumes dans la base de données de configuration des unités.
reducevg(1)	Supprime des volumes physiques d'un groupe de volumes. Si tous les volumes physiques d'un groupe sont supprimés, le groupe lui-même l'est également.
reorgvg(1)	Réorganise l'affectation de la partition physique d'un groupe de volume.
restbase(1)	Restaure les informations personnalisées de l'image d'amorçage.
rmdel(1)	Supprime un delta d'un fichier SCCS (Source Code Control System).
rmdev(1)	Supprime une unité du système.
rmf(1)	Supprime des dossiers et les messages qu'ils contiennent.
rmfs(1)	Supprime un système de fichiers.
rmgroup(1)	Supprime un groupe.
rmitab(1)	Supprime des enregistrements du fichier **/etc/inittab**.
rmlv(1)	Supprime des volumes logiques d'un groupe de volumes.
rmlvcopy(1)	Supprime des copies d'un volume logique.

rmm(1)	Supprime des messages.
rmnamsv(1)	Modifie la configuration d'un service de noms TCP/IP sur un hôte.
rmnotify(1)	Supprime une définition de méthode de notification de la classe d'objets de notification.
rmprtsv(1)	Supprime de la configuration un service d'impression sur une machine serveur ou client.
rmps(1)	Supprime un espace de pagination du système.
rmque(1)	Supprime une file d'attente d'impression du système.
rmquedev(1)	Supprime du système une unité de file d'attente imprimante ou traceur.
rmserver(1)	Supprime une définition de sous-serveur de la classe d'objets sous-serveur.
rmssys(1)	Supprime une définition de sous-système de la classe d'objets sous-système.
rmuser(1)	Supprime un compte utilisateur.
rmvfs(1)	Supprime des entrées du fichier **/etc/vfs**.
rmvirprt(1)	Supprime une imprimante virtuelle.
savebase(1)	Sauvegarde les données d'unité personnalisées ODM sur l'unité d'amorçage.
syncvg(1)	Synchronise les copies non courantes de volumes logiques.
usrck(1)	Vérifie une définition utilisateur.
varyoffvg(1)	Désactive un groupe de volumes.
varyonvg(1)	Active un groupe de volumes.

Cron pour administrateurs système BSD 4.3

Le démon AIX **cron** est semblable à celui du System V Release 2. Une entrée du fichier **/etc/inittab** active le démon **cron**.

Unités pour administrateurs systèmes BSD 4.3

Une unité d'un système BSD 4.3 n'est accessible à une application que si :

- l'unité est physiquement installée et qu'elle fonctionne,
- le pilote de l'unité se trouve dans le noyau,
- les fichiers unité spéciaux correspondants se trouvent dans le répertoire **/dev**,

Une unité d'un système AIX n'est accessible à une application que si :

- l'unité est physiquement installée et qu'elle fonctionne,
- le pilote de l'unité se trouve dans le noyau ou dans une extension chargée,
- les fichiers unité spéciaux correspondants se trouvent dans le répertoire **/dev**,
- la base de données objet du répertoire **/etc/objrepos** contient des entrées pour l'unité qui correspondent à la configuration physique.

Les programmes propres aux unités, appelés *méthodes*, qui se trouvent dans le répertoire **/etc/methods**, maintiennent la base de données objet. Les méthodes sont appelées par le gestionnaire de configuration (accessible via la commande **cfgmgr**) et d'autres commandes.

Si une application ne peut plus accéder à une unité, ce peut être le matériel qui est en cause, ou encore la base de données du répertoire **/etc/objrepos** qui est endommagée.

Les processus de la commande **cfgmgr** de la base de données de configuration (du répertoire **/etc/objrepos**) sont traités au moment du lancement par la commande **cfgmgr** (le gestionnaire de configuration).

Le pseudo-code ci-dessous illustre la logique du gestionnaire de configuration :

```
/* Main */
While there are rules in the Config_Rules database
        {
        Get the next rule and execute it
        Capture stdout from the last execution
        Parse_Output(stdout)
        }
/* Parse Output Routine */
/* stdout will contain a list of devices found */
Parse_OutPut(stdout)
        {
        While there are devices left in the list
                {
                Lookup the device in the database
                if (!defined)
                        Get define method from database and
execute
                if (! configured)
                        {
                        Get config method from database and
execute
                        Parse_Output(stdout)
                         }
                }
        }
```

Tableau de comparaison de fichiers BSD 4.3, SVR4 et AIX

Le tableau suivant compare noms et fonctions des fichiers dans les trois systèmes BSD 4.3, SVR4 et AIX.

Tableau de comparaison de fichiers				
Fichier BSD 4.3	**Fichier SVR4**	**Fichier AIX**	**Base de données**	**Type (odm/dbm)**
L–Devices	Devices	Devices	non	
L–dialcodes	Dialcodes	Dialcodes	non	
L.cmds	Permissions	Permissions	non	
L.sys	Systems	Systen	non	
USERFILE	Permissions	Permissions	non	
aliases	mail/namefiles	aliases	alias DB/DB	dbm
fstab	vfstab	filesystems	non	
ftpusers	ftpusers	ftpusers	non	
gettytab		N/A		
group	group	group	non	
hosts	hosts	hosts	non	
hosts.equiv	hosts.equiv	hosts.equiv	non	
inetd.conf	inetd.conf	inetd.conf	non	
map3270	N/A	map3270	non	
motd	motd	motd	non	
mtab	mnttab	N/A	non	
named.boot	named.boot	named.boot	non	
named.ca		named.ca	non	
named.hosts		named.data (voir remarque)	non	
named.local		named.local	non	

named.pid	named.pid	named.pid	non	
named.rev		named.rev	non	
networks	networks	networks	non	
passwd	passwd	passwd	non	
printcap	qconfig	qconfig		
protocols		protocols	non	
remote	remote	remote	non	
resolv.conf	resolv.conf	resolv.conf	non	
sendmail.cf	sendmail.cf	sendmail.cf	sendmail.cfDB	aucun
services		services	non	
shells	shells	N/A		
stab		N/A		
syslog.conf		syslog.conf	non	
syslog.pid		syslog.pid	non	
termcap	terminfo	terminfo		
ttys	ttys	N/A	yes	odm
types		N/A		
utmp	utmp	utmp		
vfont		N/A		
vgrindefs		vgrindefs		
wtmp	wtmp	wtmp		

Remarque : Les noms de fichiers **named.ca**, **named.hosts**, **named.local** et **named.rev** peuvent être définis par l'utilisateur dans le fichier **named.boot**. Les noms indiqués ici sont ceux qui ont été retenus dans la documentation AIX.

Systèmes de fichiers pour administrateurs système BSD 4.3

Cette section propose une comparaison sommaire entre les systèmes de fichiers AIX et ceux d'autres systèmes et indique les types de systèmes de fichiers pris en charge sur les systèmes AIX.

AIX passe par le fichier **/etc/filesystem** pour obtenir les informations sur les unités des systèmes de fichiers et propose des commandes similaires pour le montage et le démontage des systèmes de fichiers.

Fichiers /etc/filesystems et /etc/fstab

Les systèmes BSD 4.3 stockent les listes d'unités par bloc et de points de montage dans le fichier **/etc/fstab**.

Les systèmes SVR4 stockent les donées sur les unités par bloc et sur les points de montage dans le fichier **/etc/vfstab**.

AIX stocke les données sur les unités sur bloc et sur les points de montage dans le fichier **/etc/filesystems**. Le commandes **crfs**, **chfs** et **rmfs** mettent à jour le fichier **/etc/filesystems**.

Les administrateurs BSD 4.3 seront sans doute intéressés par la variable **check** du fichier **/etc/filesystems**. Vous pouvez affecter à cette variable la valeur True, False ou une valeur numérique. Par exemple, vous pouvez spécifier `check=2` dans le fichier **/etc/filesystems**. Le nombre précise le passage de la commande **fsck** qui effectuera la vérification du système de fichiers concerné. Le paramètre **check** correspond au cinquième champ d'un enregistrement du fichier **/etc/fstab**.

Aucun paramètre relatif à la fréquence de cliché ne se trouve dans le fichier **/etc/filesystems**.

Support des systèmes de fichiers sur AIX

AIX prend en charge les quotas disque.

AIX ne permet pas le montage de disquettes comme systèmes de fichiers.

La syntaxe AIX des commandes **mount** et **umount** diffère de celle des versions BSD 4.3 et SVR4 de ces commandes. Le tableau récapitule les différentes syntaxes.

Commandes mount et unmount			
Fonction	**AIX Syntaxe**	**Syntaxe BSD 4.3**	**Syntaxe SVR4**
monte tous les systèmes de fichiers	**mount all**	**mount –a**	**mountall**
démonte tous les systèmes de fichiers	**umount all**	**umount –a**	**umountall**

Pour en savoir plus, reportez-vous à "Systèmes de fichiers - généralités", page 7-2.

Recherche et examen de fichiers pour administrateurs système BSD 4.3

AIX accepte les commandes BSD 4.3 suivantes :

- **which,**
- **whereis,**
- **what,**
- **file.**

AIX n'accepte pas la syntaxe BSD 4.3 **fast find** de la commande **find**. Il n'existe pour le moment pas de fonction de remplacement. Le script **ffind** suivant peut simuler la fonction :

```
#!/bin/bsh
PATH=/bin
for dir in /bin /etc /lib /usr
do
find $dir –print | egrep $1
done
```

La syntaxe du script **ffind** est la suivante :

```
ffind NomFichie
```

Espace de pagination pour administrateurs système BSD 4.3

Les commandes AIX suivantes aident à gérer l'espace de pagination (également appelé espace de permutation) :

chps(1)	Modifie les attributs d'un espace de pagination.
lsps(1)	Affiche la liste des attributs d'un espace de pagination.
mkps(1)	Ajoute un espace de pagination au système.
rmps(1)	Supprime un espace de pagination du système.
swapon(1)	Spécifie d'autres unités de pagination et de permutation.

Si vous avez besoin d'un grand espace de pagination, placez un volume logique de pagination sur chaque disque : vous pourrez ainsi planifier la pagination sur plusieurs unités de disque.

Réseau pour administrateurs système BSD 4.3

Cet article traite de l'utilisation de la configuration de réseau BSD 4.3 ASCII sur un système AIX, des commandes et des options AIX complémentaires, de la résolution de noms et d'adresses sur les systèmes AIX et des différences entre la gestion d'un réseau BSD 4.3 et celle d'un réseau AIX.

Configuration BSD 4.3 : modification du lancement par défaut

Vous pouvez administrer les interfaces réseau AIX via SMIT et les fichiers ODM, ou encore via les fichiers de configuration BSD 4.3 ASCII.

Pour administrer des interfaces réseau via les fichiers de configuration BSD 4.3 ASCII, annulez, dans le fichier **/etc/rc.net,** la mise en commentaire des commandes sous l'en-tête :

```
# Part II – Traditional Configuration
```

Puis, si vous souhaitez que soient pris en charge la configuration des fichiers ordinaires et le support SRC, éditez le fichier **/etc/rc.net** et annulez la mise en commentaire des commandes **hostname**, **ifconfig** et **route** avec les paramètres appropriés.

Pour la configuration des fichiers ordinaires sans support SRC, lancez la commande **smit setbootup_option** pour passer à une configuration **rc** de style BSD. Cette option configure le système pour qu'il utilise le fichier **/etc/rc.bsdnet** au moment du lancement. Vous devez également éditer le fichier **/etc/rc.bsdnet** et annuler la mise en commentaire des commandes **hostname**, **ifconfig** et **route** avec les paramètres appropriés.

Autres options pour les commandes ifconfig et netstat

La commande AIX **ifconfig** peut être assortie des options complémentaires suivantes :

mtu	Unité de transfert maximum (MTU) utilisée sur le réseau local (et les sous-réseaux) et MTU utilisée sur les réseaux distants. Pour optimiser la compatibilité avec Ethernet et les autres réseaux, donnez par défaut à la variable **mtu** la valeur 1 500 pour les réseaux en anneau à jeton et pour les réseaux Ethernet.
allcast	Définit la stratégie de diffusion de l'anneau à jeton. Définir l'indicateur **allcast** optimise la connectivité à travers les ponts en anneau à jeton. Annuler l'indicateur **allcast** (en spécifiant **–allcast**) diminue la densité du trafic sur l'anneau.

La commande AIX **netstat** peut être assortie de l'indicateur **–v**. La commande **netstat –v** imprime des statistiques sur le pilote de l'imprimante (décompte des octets transmis, décompte des erreurs de transmission, décompte des octets reçus, décompte des erreurs reçues, etc.).

Autres commandes de gestion de réseau

Voici les commandes complémentaires prises en charge par AIX :

securetcpip	Script shell qui active le mode accès contrôlé, lequel procure un surcroît de sécurité. Il interdit l'exécution de plusieurs programmes TCP/IP non sécurisés tels que **tftp**, **rcp**, **rlogin** et **rsh,** et restreint l'usage du fichier .netrc. Il restreint également l'usage du fichier **.netrc**.
gated	Fournit un support MIB pour SNMP.

no	Définit des options réseau :	
	dogticks	Définit la granularité de l'horloge pour les routines **ifwatchdog**.
	subnetsarelocal	Détermine si l'adresse paquet est sur le réseau local.
	ipsendredirects	Spécifie si le noyau doit envoyer les signaux de réacheminement.
	ipforwarding	Spécifie si le noyau doit faire suivre les paquets.
	tcp_ttl	Durée de vie des paquets TCP (Transmission Control Protocol).
	udp_ttl	Durée de vie des paquets UDP (User Datagram Protocol).
	maxttl	Durée de vie des paquets RIP (Routing Information Protocol).
	ipfragttl	Durée de vie des fragments IP (Internet Protocol).
	lowclust	Spécifie un niveau bas pour le pool de grappe **mbuf**.
	lowmbuf	Spécifie un niveau bas pour le pool **mbuf**.
	thewall	Quantité maximale de mémoire susceptible d'être affectée aux pools **mbuf** et aux pools de grappe **mbuf**.
	arpt_killc	Délai (en minutes) au bout duquel une entrée ARP (Address Resolution Protocol) inactive est supprimée.
iptrace	Fournit un suivi de paquet au niveau interface pour les protocoles Internet.	
ipreport	Formate le suivi pour le rendre lisible. Voici un exemple :	

```
iptrace –i en0 /tmp/iptrace.log
# kill iptrace daemon
kill ‘ps ax | grep iptrace | awk ’{ print $1 }’‘
ipreport /tmp/iptrace.log | more
```

Résolution de noms et d'adresses

Les sous-routines **gethostbyname** et **gethostbyaddr** de la bibliothèque **libc** fournissent le support des services DNS (Domain Name Service), NIS (Network Information Services, ex–Yellow Pages), et la base de données **/etc/hosts**. Si le fichier **/etc/resolv.conf** existe, le serveur de noms est le premier exploré. Si le nom n'est pas résolu et que NIS est actif, NIS est exploré. Si NIS n'est pas actif, c'est le fichier **/etc/hosts** qui est exploré.

Différences entre AIX et BSD 4.3

Sur les systèmes AIX, les démons réseau sont lancés depuis le fichier **/etc/rc.tcpip**, et non depuis le fichier **/etc/rc.local**. Le script shell **/etc/rc.tcpip** est appelé depuis le fichier **/etc/inittab**, et non depuis le fichier **/etc/rc**.

Si le contrôleur SRC (System Resource Controller) est actif, les démons TCP/IP sont exécutés sous son contrôle. Si vous ne souhaitez pas qu'ils le soient, lancez la commande **smit setbootup_option** pour passer à une configuration **rc** de style BSD.

Les fonctions de gestion de réseau BSD 4.3 acceptées par AIX sont les suivantes :

- fonctions de journalisation SYSLOG au niveau noyau,
- support des systèmes XNS (Xerox Network Systems),
- droits d'accès aux sockets de domaine UNIX.

Commande tn3270

La commande **tn3270** est un lien avec la commande **telnet**, mais qui utilise le fichier **/etc/map3270** et la variable d'environnement courante **TERM** pour générer les mappages clavier 3270. Ainsi, la commande **tn3270** opère exactement comme sa version BSD.

Si vous souhaitez modifier les séquences d'échappement par défaut utilisées par les commandes **tn3270**, **telnet** et **tn**, définissez la variable d'environnement **TNESC** avant de lancer ces commandes.

Documentation en ligne et commande man pour administrateurs système BSD 4.3

AIXaccepte les commandes **man –k**, **apropos** et **whatis**, mais la base de données utilisée par ces commandes soit être créée au préalable via la commande **catman –w.**

La commande AIX **man** recherche d'abord les pages de texte plat dans les fichiers **/usr/man/cat?**. Puis, elle recherche les pages formatées **nroff** dans le fichier **/usr/man/man?**. Les nouvelles pages de manuel peuvent être ajoutées en texte plat ou au format **nroff.**

Remarques :

1. Les pages de texte de la commande **man** ne sont pas fournies avec le système. La base de données correspondante doit être créée via la commande **catman**. Ces pages peuvent être soit du texte plat stocké dans les fichiers **/usr/man/cat**?, soit des pages formatées **nroff** stockées dans les fichiers **/usr/man/man**?.
2. Le programme sous licence de formatage de texte doit être installé pour que la commande **nroff** soit à disposition de la commande **man** pour la lecture des pages formatées **nroff**.

NFS et NIS (ex”Yellow Pages”) pour administrateurs système BSD 4.3

Les démons NFS (Network File System) et NIS (Network Information Services)sont lancés à partir du fichier **/etc/rc.nfs**. Ils supposent l’activation préalable du démon **portmap** dans le fichier **/etc/rc.tcpip**. Par défaut, le fichier **/etc/rc.nfs** n’est pas appelé par le fichier **/etc/inittab**. Si vous ajoutez une ligne dans le fichier **/etc/inittab** pour appeler le script **/etc/rc.nfs**, il doit être appelé après le script **/etc/rc.tcpip.**

Si NIS est actif, vous devez intégrer une entrée racine avant l’entrée **+**:: (signe plus, deux-points, deux-points) dans le fichier **/etc/passwd** et une entrée système avant l’entrée **+**:: dans le fichier **/etc/group**. Un administrateur système peut ainsi se connecter comme utilisateur racine et effectuer les modifications requises si le système ne parvient pas à communiquer avec le serveur NIS.

NFS peut être configuré via le raccourci Web-based System Manager, **wsm network** ou SMIT, **smit nfs**. Les menus Web-based System Manager et SMIT font référence à NIS (exYellow Pages) sous la forme NIS. Nombre des commandes NFS et NIS se trouvent dans les répertoires **/etc** et **/usr/etc.**

Certains environnements NFS utilisent une commande **arch** pour identifier les familles et les types de machines.A titre d’exemple, si vous utilisez le ESCALA, nous vous suggérons de définir l’identifiant **power** pour la famille (CPU).

Mots de passe pour administrateurs système BSD 4.3

Voici quelques précisions sur les différences de gestion des mots de passe sur les systèmes AIX et BSD 4.3.

Définition d'un mot de passe utilisateur

Lorsque vous exécutez la commande AIX **/bin/passwd** comme utilisateur racine, vous êtes invité à fournir le mot de passe racine. Voici un exemple :

```
# passwd cslater
Changing password for ″cslater″
Enter root's Password or
cslater's Old password:
cslater's New password:
Re–enter cslater's
new password:
#
```

La version BSD 4.3 ne vous invite pas à entrer le mot de passe racine. En voici un exemple :

```
# passwd cslater
New password:
Retype new password:
#
```

Importation d'un fichier de mots de passe BSD 4.3

Pour importer un fichier de mots de passe BSD 4.3, copiez-le dans le fichier **/etc/passwd,** puis entrez :

```
pwdck –y ALL
```

Le fichier **/etc/security/limits** doit être ensuite mis à jour avec une strophe nulle pour tout nouvel utilisateur. La commande **usrck** le fait, mais elle peut poser problème sauf si le fichier **/etc/group** est importé avec le fichier **/etc/passwd**.

> **Remarque :** Si le fichier **/etc/security/limits** est modifié, la pile ne doit pas dépasser 65 536 octets. Si elle dépasse cette limite, l'exécution de la commande **usrck** risque de poser problème : ramenez la taille de la pile à 65536 et relancez la commande **usrck**.

Exécutez également les commandes **grpck** et **usrck** pour vérifier les attributs groupe et utilisateur.

Edition du fichier de mots de passe (Password)

Sous AIX, les commandes **lsuser**, **mkuser**, **chuser** et **rmuser** permettent de gérer les mots de passe. Toutes ces commandes peuvent être exécutées via SMIT ou Web-based System Manager. Toutefois elles ne traitent qu'un utilisateur à la fois.

> **Remarque :** Passer par un éditeur pour modifier plusieurs noms utilisateur requiert d'éditer simultanément plusieurs fichiers, car les mots de passe sont stockés dans le fichier **/etc/security/passwd**, les informations sur les droits d'accès, dans le fichier **/etc/security/user** et les autres données utilisateur, dans le fichier **/etc/passwd**.

AIXrejette la commande **vipwn,** mais accepte la commande **mkpasswd**. Mais vous pouvez toujours administrer les mots de passe sur un système AIX comme vous le feriez sur un système BSD 4.3. Procédez comme suit :

1. Placez un fichier de mots de passe BSD 4.3 dans le fichier **/etc/shadow**.
2. Modifiez les droits d'accès au fichier :

   ```
   chmod 000 /etc/shadow
   ```

3. Placez le script shell **vipw** suivant dans le répertoire **/etc** :

```
#!/bin/bsh
#
# vipw for AIX V3. Uses pwdck for now. May use usrck someday
#
PATH=/bin:/usr/bin:/etc:/usr/ucb # Add to this if your editor is
                                 # some place else
if [ –f /etc/ptmp ] ; then
                 echo ”/etc/ptmp exists. Is someone else using
vipw?”
        exit 1
fi
if [ ! –f /‘which ”$EDITOR” | awk ’{ print $1 }’‘ ] ; then
        EDITOR=vi
fi
cp /etc/shadow /etc/ptmp
if (cmp /etc/shadow /etc/ptmp) ; then
        $EDITOR /etc/ptmp
else
        echo cannot copy shadow to ptmp
        exit 1
fi
if (egrep ”^root:” /etc/ptmp >/dev/null) ; then
        cp /etc/ptmp /etc/shadow ; cp /etc/ptmp /etc/passwd
        chmod 000 /etc/passwd /etc/shadow
        pwdck –y ALL 2>1 >/dev/null # return code 114 may change
                rc=$?
        if [ $rc –eq 114 ]; then
                chmod 644 /etc/passwd
                        rm –f /etc/passwd.dir /etc/passwd.pag
                mkpasswd /etc/passwd
                        # update /etc/security/limits, or ftp
                        # will fail
                else
                         pwdck –y ALL
                fi
else
        echo bad entry for root in ptmp
fi
rm /etc/ptmp
```

4. Si vous utilisez le script shell **vipw** ou la commande **mkpasswd**, n’oubliez pas que Web-based System Manager, SMIT et les commandes **mkuser**, **chuser** et **rmuser** n’utilisent pas la commande **mkpasswd**. Vous devez lancer :

   `pour mettre à jour les fichiers` **`/etc/passwd.dir`** `et` **`/etc/passwd.pag.`**

   Attention : L’initialisation de la variable **IFS** et des instructions **trap** permettent de se prémunir contre certaines méthodes exploitant les failles au niveau de la sécurité, inhérentes à la fonction **setuid**. Les scripts shell **vipw** et **passwd** sont toutefois conçus pour des environnements relativement ouverts, où la compatibilité est un élément-clé. Si vous souhaitez un environnement plus sûr, utilisez exclusivement les commandes AIX standard.

5. Placez le script shell **passwd** suivant dans le répertoire **/usr/ucb** :

```
#!/bin/ksh
#
# matches changes to /etc/security/passwd file with changes to
#/etc/shadow
#
IFS=" "
PATH=/bin
trap "exit 2" 1 2 3 4 5 6 7 8 10 12 13 14 15 16 17 18 21 22 \
        23 24 25 27 28 29 30 31 32 33 34 35 36 60 61 62
if [ -n "$1" ]; then
        USERNAME=$1
else
        USERNAME=$LOGNAME
fi
if [ -f /etc/ptmp ] ; then
        echo password file busy
        exit 1
fi
        trap "rm /etc/ptmp; exit 3" 1 2 3 4 5 6 7 8 10 12 13 \
                14 15 16 17 18 21 22 23 24 25 27 28 29 30 31 \
                32 33 34 35 36 60 61 62
if (cp /etc/security/passwd /etc/ptmp) ; then
        chmod 000 /etc/ptmp else
        rm -f /etc/ptmp exit 1
fi
if ( /bin/passwd $USERNAME ) ; then
        PW=‘ awk ’ BEGIN { RS = "" }
                $1 == user { print $4 } ’ user="$USERNAME:" \
/etc/security/passwd ‘
else
        rm -f /etc/ptmp
        exit 1
fi
rm -f /etc/ptmp
awk -F: ’$1 == user { print $1":"pw":"$3 ":"$4":"$5":"$6":"$7 }
        $1 != user { print $0 }’ user="$USERNAME" pw="$PW" \
                /etc/shadow > /etc/ptmp
chmod 000 /etc/ptmp
mv -f /etc/ptmp /etc/shadow
```

6. Modifiez les droits d’accès au script **passwd** :

```
chmod 4711 /usr/ucb/passwd
```

7. Vérifiez que la variable d’environnement **PATH** de chaque utilisateur spécifie d’explorer le répertoire **/usr/ucb** avant le répertoire **/bin.**

Mesure et affinement des performances pour administrateurs système BSD 4.3

Sous AIX, toutes les unités sont dotées d'attributs. Pour les visualiser, entrez :

```
lsattr –E –l NomUnité
```

Tout attribut ayant pour valeur True peut être modifié via la commande :

```
chdev –l NomUnité –a attr=valeur
```

Attention : Modifier incorrectement les paramètres d'unité peut endommager le système.

Par défaut, le nombre maximal de processus par utilisateur est de 40. Cette valeur peut se révéler insuffisante pour des utilisateurs ayant ouvert simultanément plusieurs fenêtres. Pour modifier la valeur sur tout le système, entrez :

```
hdev –l sys0 –a maxuproc=100
```

Le maximum est ici porté à 100 (effectif dès le réamorçage du système).

Pour afficher la valeur courante, ainsi que d'autres attributs, entrez :

```
lsattr –E –l sys0
```

L'attribut **maxmbuf** n'est pour le moment pas accepté par les services **mbuf**.

AIXaccepte les commandes **vmstat** et **iostat**, mais non la commande **systat,** ni les moyennes de charge.

Imprimantes pour administrateurs système BSD 4.3

L'impression AIX est gérée par des programmes et des configurations du répertoire **/usr/lpd**. La conception, la configuration, le mécanisme de mise en file d'attente et le processus démon des sous-systèmes d'impression de BSD 4.3 et d'AIX sont différents. Les deux systèmes utilisent néanmoins le protocole **lpd** pour les impressions à distance. Ils utilisent également le fichier **/etc/hosts.lpd**, s'il existe, ou sinon **/etc/host.equiv**. Le sous-système d'impression AIX offre une passerelle vers les sous-systèmesBSD 4.3 ; les systèmes AIX peuvent donc soumettre des travaux aux systèmes BSD 4.3 et accepter des travaux soumis par ces systèmes.

Le fichier **/etc/printcap** de BSD 4.3 n'existe pas dans AIX. Ce fichier combine des informations de configuration du spouleur et de la base de données des capacités d'imprimante. Les utilisateurs doivent bien connaître le format et les mots-clés du fichier **printcap** pour configurer correctement une imprimante.

Le fichier **/etc/qconfig** d'AIX ne contient que des informations sur la configuration du spouleur. Les capacités de l'imprimante sont définies dans la base de données prédéfinie/pré-personnalisée ODM. Vous disposez de la commande **mkvirprt** pour définir les capacités d'une imprimante donnée sur le système.

Pour rendre l'imprimante **lp0** disponible pour imprimer sur l'hôte distant `viking`, insérez, dans un fichier système BSD 4.3 **/etc/printcap** :

```
lp0|Print on remote printer attached to
viking:Z
:lp=:rm=viking:rp=lp:st=/usr/spool/lp0d
```

Pour faire de même sur un système AIX, insérez les lignes suivantes dans le fichier **/etc/qconfig** :

```
lp0:
        device = dlp0
        host = viking
        rq = lp
dlp0:
        backend = /usr/lib/lpd/rembak
```

Pour en savoir plus sur le sous-système d'impression, reportez-vous à la section relative aux imprimantes (gestion système).

AIX accepte les commandes d'impression et les fonctions de bibliothèque suivantes :

cancel(1)	Annule les demandes à une imprimante ligne.
chquedev(1)	Change le nom d'unité de l'imprimante ou du traceur.
chvirprt(1)	Modifie la valeur des attributs d'une imprimante virtuelle.
disable(1)	Désactive une file d'attente d'imprimante.
enable(1)	Active une file d'attente d'imprimante.
hplj(1)	Posttraite la sortie **troff** pour HP LaserJetII avec cartouche K.
ibm3812(1)	Posttraite la sortie **troff** pour IBM 3812 Mod 2 Pageprinter.
ibm3816(1)	Posttraite la sortie **troff** pour IBM 3816 Pageprinter.
ibm5587G(1)	Posttraite la sortie **troff** pour IBM 5587G avec cartouche 32x32/24x24.
lp(1)	Envoie des demandes à une imprimante ligne.
lpr(1)	Met des travaux d'impression en file d'attente.
lprm(1)	Supprime des travaux de la file de spoulage d'une imprimante ligne.

lpstat(1)	Affiche des informations sur l'état d'une imprimante ligne.
lptest(1)	Génère la configuration d'impression d'une imprimante ligne.
lsallqdev(1)	Affiche la liste de toutes les unités de file d'attente configurées d'une file d'attente.
lsvirprt(1)	Affiche les attributs d'une imprimante virtuelle.
mkque(1)	Ajoute une file d'attente d'impression au système.
mkquedev(1)	Ajoute une unité de file d'attente d'impression au système.
mkvirprt(1)	Crée une imprimante virtuelle.
pac(1)	Prépare les enregistrements comptables de l'imprimante/du traceur.
piobe(1)	Imprime le gestionnaire des travaux d'impression pour le programme expéditeur de l'imprimante.
pioburst(1)	Génère les pages d'en-tête et de fin pour les sorties.
piocmdout(3)	Sous-routine qui génère une chaîne d'attribut pour un formateur d'impression.
piodigest(1)	Prétraite les valeurs des attributs de définition d'une imprimante virtuelle et les enregistre.
pioexit(3)	Sous-routine existant à partir d'un formateur d'impression.
pioformat(1)	Pilote un formateur d'impression.
piofquote(1)	Convertit certains caractères de contrôle destinés aux imprimantes PostScript.
piogetstr(3)	Sous-routine qui extrait une chaîne d'attribut pour un formateur d'impression.
piogetvals(3)	Sous-routine qui initialise les variables base de données des attributs d'impression pour un formateur d'impression.
piomsgout(3)	Sous-routine qui envoie un message à partir d'un formateur d'impression.
pioout(1)	Programme pilote d'unité du programme expéditeur de l'imprimante.
piopredef(1)	Crée une prédéfinition du flot de données d'impression.
proff(1)	Formate le texte pour les imprimantes avec flots de données personnelles.
prtty(1)	Imprime vers le port d'imprimante du terminal.
qadm(1)	Administre le système de spoulage de l'imprimante.
qconfig(4)	Configure un système de files d'attente d'impression.
qstatus(1)	Fournit l'état de l'imprimante au système de files d'attente d'impression.
restore(3)	Restaure l'imprimante à son état par défaut.
rmque(1)	Supprime une file d'attente d'impression du système.
rmquedev(1)	Supprime du système une unité de file d'attente imprimante ou traceur.
rmvirprt(1)	Supprime une imprimante virtuelle.
splp(1)	Affiche ou modifie les paramètres du pilote d'impression.
xpr(1)	Formate un fichier de cliché de fenêtre pour une sortie imprimante.

Terminaux pour administrateurs système BSD 4.3

Traditionnellement, pour activer/désactiver un port, les administrateurs BSD 4.3 modifient le fichier **/etc/ttys** et envoient un signal **HUP** au programme **init.**

AIXstocke les informations sur le port du terminal dans le gestionnaire ODM et lance les terminaux lorsque le programme **init** lit le fichier **/etc/inittab**. Dans AIX, vous devriez utiliser l'application Web-based System Manager Devices ou SMIT pour configurer les ports du terminal.

Il n'existe pas de mappage fixe entre le port et le nom de fichier unité spécial dans le répertoire **/dev**. Cela peut engendrer des doutes sur le port à configurer, pour les administrateurs abordant AIX. Dans les menus SMIT, le port série de la première carte (libellé s1) est référencé par l'emplacement `00-00-S1`, la carte `sa0` et le port `s1` dans le menu SMIT. Le port série de la seconde carte (libellé s2) est référencé par l'emplacement `00-00-S2`, la carte `sa1` et le port `s2`.

Pour activer/désactiver un port, vous disposez des commandes **penable** et **pdisable**.

termcap et terminfo

Comme System V, AIX se sert des entrées **terminfo** du fichier **/usr/lib/terminfo/?/***. Les utilisateurs BSD 4.3 trouveront sans doute utiles les commandes suivantes :

captoinfo(1)	Convertit un fichier **termcap** en fichier **terminfo**
tic(1)	Traduit les fichiers **terminfo** source en format compilé.

AIXinclut la source de nombreuses entrées **terminfo**. Certaines doivent être compilées via la commande **tic**. Le fichier **termcap** se trouve dans le fichier **/lib/libtermcap/termcap.src**.

Dave Regan a fait don de son programme **untic** au domaine public. Ce programme "décompile" les entrées **terminfo**, de sorte que la forme source puisse être modifiée et recompilée avec **tic**. Il est disponible sur les sites archivant **comp.sources.unix**.

UUCP pour administrateurs système BSD 4.3

AIX fournit les utilitaires BNU (Basic Networking Utilities) System V (souvent appelés HDB UUCP).

Dialers(4)	Affiche la liste des modems utilisés par les liaisons BNU à distance.
Maxuuxqts(4)	Limite le nombre d'instances de démons BNU **uuxqt** exécutables simultanément.
Permissions(4)	Spécifie les droits des systèmes distants sur les commandes BNU.
Poll(4)	Spécifie le moment où BNU doit interroger les systèmes distants.
Systems(4)	Affiche la liste des ordinateurs distants avec lesquels peut communiquer le système local.
rmail(1)	Gère le courrier reçu à distance via BNU.
uucheck(1)	Vérifie les fichiers et les répertoires requis par BNU.
uuclean(1)	Supprime les fichiers du répertoire de spoulage BNU.
uucleanup(1)	Supprime les fichiers sélectionnés du répertoire de spoulage BNU.
uucpadm(1)	Entre les informations de configuration BNU de base.
uudemon.admin(1)	Donne régulièrement des informations sur l'état des transferts de fichiers BNU.
uudemon.cleanu(1)	Nettoie les répertoires de spoulage et les fichiers journaux BNU.
uudemon.hour(1)	Lance les appels de transport de fichier vers les systèmes distants via BNU.
uudemon.poll(1)	Interroge les systèmes spécifiés dans le fichier d'interrogation BNU.
uulog(1)	Donne des informations sur les activités de transfert de fichiers BNU sur un système.
uupoll(1)	Force l'interrogation d'un système BNU distant.
uuq(1)	Affiche la file d'attente des travaux BNU et supprime de cette file les travaux spécifiés.
uusnap(1)	Affiche l'état des contacts BNU avec les systèmes distants.
uustat(1)	Consigne l'état et propose un contrôle limité des opérations BNU.

AIX propose également les commandes BSD 4.3 **uuencode** et **uudecode**. La commande HDB **uugetty** n'est pas acceptée.

Pour en savoir plus, reportez-vous à Fichiers BNU, formats de fichiers et répertoires dans *AIX 4.3 Guide de l'utilisateur : communications et réseaux*.

Annexe B. InfoExplorer

Dans les versions précédentes d'AIX, la documentation était livrée sous forme de base de données InfoExplorer. Avec la version actuelle, la documentation est fournie dans un format HTML en vue de son utilisation avec un navigateur web. Les bases de données InfoExplorer **ne sont plus** fournies. Pour naviguer dans les bases de données InfoExplorer, vous devez avoir acheté la fonction InfoExplorer avec AIX. Ce chapitre n'est utile que pour les clients ayant fait l'acquisition d'InfoExplorer.

Personnalisation d'InfoExplorer

InfoExplorer peut être installé sur CD-ROM ou disque fixe. L'utilisation d'un disque améliore les performances, mais requiert davantage d'espace disque en raison de la taille des bases de données.

Pour personnaliser InfoExplorer, vous pouvez définir des notes publiques (voir page B-3) à l'attention des utilisateurs pour leur fournir des informations sur l'installation. Vous pouvez aussi définir des listes de signets et des fichiers historiques puis les transmettre aux utilisateurs pour leur fournir une aide ou des informations sous forme de listes structurées.

Bases de données InfoExplorer

Le code et la bibliothèque InfoExplorer se trouvent dans le répertoire **/usr/lpp/info**. Ce répertoire regroupe différents sous-répertoires contenant les programmes exécutables, les bases de données, les polices et les notes publiques. Voici la liste et le contenu de ces sous-répertoires :

bin	programmes exécutables pour les outils ASCII et les outils graphiques InfoExplorer. Contient également la commande **mergenote**, qui fusionne des groupes de fichiers de notes dans un fichier unique.
data	fichier **ispaths** décrivant les bases de données installées ainsi que certains fichiers de définition du système et fichier de stockage des notes publiques créées. Reportez-vous à "Création des notes publiques InfoExplorer", page B-9. Contient également des fichiers de définitions utilisés par les exécutables InfoExplorer.
data/JP	fichiers de définition pour l'environnement de la langue japonaise MBCS.
X11fonts/JP	polices japonaises servant à l'interface graphique InfoExplorer pour l'environnement de la langue japonaise MBCS.
notes	notes système, le cas échéant.
lib/*Langue*	bases de données installées pour la langue indiquée par le nom du répertoire *Langue*. Ce nom est dérivé de la langue configurée pour le système. Par exemple, sur un système canadien–français, le nom du répertoire sera **fr_CF**. Le nom par défaut est **en_US** (système américain en langue anglaise).
lib/Language/ *bibliothèque*	sous-répertoires de bibliothèques supplémentaires dans un répertoire de bibliothèques.

Sur un système installé en plusieurs langues, plusieurs répertoires *Langue* peuvent coexister dans le répertoire **/usr/lpp/info/lib**. Par exemple, sur un système qui exploite l'allemand et le français, la base de données allemande peut être installée dans le sous-répertoire */usr/lpp/info/lib/de_DE*, et la base française dans le sous-répertoire */usr/lpp/info/lib/fr_FR*. Les utilisateurs peuvent définir une langue en modifiant la valeur de la variable **LANG**, **INFOLANG** ou **INFOLOCALE**. Pour en savoir plus, reportez-vous à "Modification de la langue dans InfoExplorer", page B-8.

Il existe deux autres fichiers qui ne résident pas dans **/usr/lpp/info** :

/usr/bin/info	script shell qui détermine l'appel de la version ASCII ou graphique d'InfoExplorer.
/usr/lib/x11/app–defaults/Info_gr	fichier d'application par défaut contenant les définitions des ressources système.

Notes publiques InfoExplorer

Les notes publiques sont accessibles à tout utilisateur, en hypertexte. Par défaut, les notes d'InfoExplorer sont privées et seul l'utilisateur qui les a créées y a accès.
Ces notes privées sont sauvegardées dans les répertoires utilisateur **$HOME/info** et **$HOME/info/<library>/notes**.

Tout utilisateur ayant accès en écriture au répertoire **/usr/lpp/info/data** peut créer des notes publiques et les stocker dans ce répertoire, en convertissant en notes publiques ses fichiers de notes privées avec la commande **mergenote**.

Accès à InfoExplorer à partir du CD-ROM

Lors du premier accès à InfoExplorer à partir du CD-ROM, vous devez :

- créer un système de fichiers CD-ROM.
- monter le système de fichiers CD-ROM.
- exécuter le script **linkinfocd**.

Remarque : Vous pouvez aussi installer les bases de données à partir du CD-ROM. Certaines bases du CD-ROM peuvent être déjà installées avec le système d'exploitation ou d'autres produits sous licence. Lancez la commande **lslpp** ou faites appel à SMIT pour répertorier les bases déjà installées sur le système.

L'application d'installation que vous utilisez (SMIT ou l'une des applications VSM) crée un point de montage temporaire pour le CD-ROM.

Prérequis

1. Vous devez être utilisateur racine ou membre du groupe système pour créer et monter le système de fichiers CD-ROM et lancer le script **linkinfocd**.

Création d'un système de fichiers CD-ROM

1. Si vous utilisez une unité CD-ROM externe, mettez-la sous tension.
2. Retirez le CD-ROM de son boîtier et placez-le dans le chargeur ou sur le plateau.
3. Insérez-le dans le chargeur ou sur le plateau.
4. Entrez le raccourci **smit crcdrfs** pour créer le système de fichiers CD-ROM. Le menu Add a CDROM File System s'affiche.
5. Appuyez sur F4 pour afficher la liste des unités disponibles. La zone NOM DE L'UNITÉ est affichée sur le menu précédent.
6. Indiquez l'unité CD-ROM voulue.
7. Passez à la zone POINT DE MONTAGE.
8. Tapez ce qui suit et n'appuyez pas sur Entrée avant l'étape 10.

   ```
   /infocd
   ```

9. Passez à la zone Montage AUTOMATIQUE lors de l'InitSystème? et sélectionnez l'une des options :
   a. Mount InfoExplorer every time the system starts : appuyez sur Tab pour basculer sur **yes.**
   b. Mount InfoExplorer manually : conservez la valeur par défaut, **no**.
10. Appuyez sur Entrée.
11. Appuyez sur F10 pour quitter SMIT.

Montage du système de fichiers CD-ROM

Procédez comme suit :

1. A l'invite système, entrez :

   ```
   smit mountfs
   ```

   Le menu Mount a File System s'affiche.
2. Passez à la zone FILE SYSTEM name.

Remarque : Le système monte toujours le CD-ROM comme un système de fichiers accessible en lecture seule. Avec la touche Tab, sélectionnez **yes** ou **no** dans la zone Mount as READ ONLY file system.

3. Appuyez sur F4 pour afficher la liste des noms de systèmes de fichiers. La zone FILE SYSTEM name est affichée sur le menu précédent.
4. Sélectionnez une ligne semblable à :

   ```
   /dev/cdx /infocd cdrfs
   ```

   où *x* représente le numéro de votre CD-ROM.
5. Appuyez sur Entrée.
6. Sélectionnez **Do**.
7. Appuyez sur F10 pour quitter SMIT dès que la zone Command: status indique **OK**.

A ce stade, les bases de données InfoExplorer sont montées et accessibles à partir du CD-ROM.

Remarques :

1. Si le CD-ROM est éjecté de son unité alors qu'il est monté, la connexion du montage est interrompue et vous ne pouvez plus accéder à InfoExplorer. Avant de retirer le CD-ROM de l'unité, démontez le système de fichiers avec la commande **unmount**. Pour y accéder de nouveau, remontez le CD-ROM avec la commande **mount** ou **smit**.
2. Au cas où vous ne pourriez plus accéder au CD-ROM, vous avez la possibilité de conserver des copies des bases de données sur votre disque. Pour en savoir plus, reportez-vous à "Suppression des bases de données InfoExplorer", page B-7.

Exécution du script linkinfocd

Le script **linkinfocd** associe les sous-répertoires de bases de données du système de fichiers CD-ROM **/infocd** au répertoire **/usr/lpp/info/lib/en_US/aix41**. Chaque sous-répertoire est associé séparément, chaque base de données pouvant être installée séparément. Ceci vous permet de disposer des bases de données installées sur votre disque, des bases de données liées à partir d'un CD-ROM monté ou d'une combinaison des deux. Le script associe en outre le fichier **ispaths** du système de fichiers **/infocd** CD-ROM au répertoire **/usr/lpp/info/data**.

Le script **linkinfocd** vérifie l'existence :

- d'InfoExplorer (**/usr/lpp/info**) sur le système ; s'il n'existe pas, le script quitte.
- du répertoire **/usr/lpp/info/lib/en_US/aix41** et le crée s'il ne le trouve pas,
- des sous-répertoires de bases de données ; si le nom de sous-répertoire de bases de données trouvé dans **/usr/lpp/info/lib/en_US/aix41** est :
  - un lien provenant du CD-ROM, le script signale par un message que la base de données est déjà associée à partir du CD-ROM,
  - un autre lien, le script remplace de force ce lien par celui provenant du CD-ROM,
  - un répertoire, le script vous demande (par le biais d'un message) de supprimer l'installation de cette base de données si vous souhaitez la lier à partir du CD-ROM monté.

  Si le sous-répertoire de bases de données est introuvable dans **/usr/lpp/info/lib/en_US/aix41**, le script associe ce sous-répertoire à partir de **/infocd/usr/lpp/info/lib/en_US/aix41** avec **/usr/lpp/info/lib/en_US/aix41**.

- du fichier **/usr/lpp/info/data/ispaths**. Si le nom de fichier **ispaths** correspond à :
  - un lien provenant du CD-ROM, le script signale par un message que le fichier **ispaths** est déjà associé à partir du CD-ROM.
  - un autre lien, le script copie le fichier lié **ispaths** dans **ispaths.linked** et associe le fichier **ispaths** à partir du CD-ROM à **/usr/lpp/info/data**.
  - un fichier, le script copie le fichier **ispaths** existant dans **ispaths.orig** et associe le fichier **ispaths** à partir du CD-ROM à **/usr/lpp/info/data**.

  En l'absence du fichier **/usr/lpp/info/data/ispaths**, le script associe le fichier **ispaths** à partir du CD-ROM à **/usr/lpp/info/data**.

Pour lancer le script **linkinfocd**, entrez :

```
/infocd/linkinfocd
```

Suppression des bases de données InfoExplorer

La méthode est variable, selon qu'InfoExplorer est installé sur disque dur à partir d'un support d'installation ou associé à partir d'un CD-ROM hypertexte monté.

Prérequis

Vous devez avoir accès en écriture au répertoire **/usr/lpp/info/lib/***$LANG*. La variable d'environnement *$LANG* indique la langue utilisée pour InfoExplorer.

Suppression des bases de données installées sur disque fixe

Pour afficher les bases installées, lancez la commande **lslpp** ou passez par SMIT. Pour supprimer une base de données installée sur disque fixe, vous devez supprimer l'option logicielle correspondante. Reportez-vous au chapitre "Maintenance des logiciels en option" dans le *Guide d'Installation d'AIX*.

Suppression des bases de données associées à partir du CD-ROM

Pour déterminer les bases associées à partir du CD-ROM hypertexte, passez au répertoire **/infocd/usr/lpp/info/lib/en_US/aix41** (commande **cd**) puis exécutez la commande **ls –l** : les répertoires de bases de données associés à partir du CD-ROM sont repérés dans la liste par les liens symboliques à **/infocd**.

Supprimez les liens symboliques des bases inutiles avec la commande **rm**. Procédez comme suit :

```
rm –f /usr/lpp/info/lib/$LANG/NomBaseDonnées
```

Par exemple, pour supprimer le lien symbolique de la base **files** (*AIX Files Reference*) à partir d'un système exploité en anglais-américain, entrez :

```
rm -f /usr/lpp/info/lib/en_US/aix41/files
```

Remarque : Il est impossible de supprimer les bases de données du CD-ROM. Pour améliorer les performances, vous pouvez installer sur disque fixe les bases de données les plus utilisées. Reportez-vous au chapitre "Installation de logiciels en option et de mises à jour de service" dans le *Guide d'Installation d'AIX*.

Modification de la langue dans InfoExplorer

Procédure

Sur un système multilingue, chaque version de la base de données est installée dans un sous-répertoire de **/usr/lpp/info/lib**. Par exemple, la version allemande est installée dans **/usr/lpp/info/lib/de_DE** et la version anglaise-américaine dans **/usr/lpp/info/lib/en_US**.

InfoExplorer détermine la langue de la base de données indépendamment de celle employée pour les messages (par exemple, les options de menu, les noms de bouton). La langue des messages est déterminée par la variable d'environnement **LANG** ou **LC_MESSAGES**. Si la variable **LC_MESSAGES** est définie, sa valeur est prise en compte ; sinon, **LANG** détermine quels messages utiliser.

InfoExplorer Pour déterminer la langue des bases de données, fait appel à plusieurs méthodes. L'ordre de priorité est le suivant :

1. Si la variable **INFOLANG** est définie, InfoExplorer essaie de lire les bases de données dans le répertoire **/usr/lpp/info/lib/<$INFOLANG>**.
2. Sinon ou si aucune bibliothèque n'est trouvée dans ce répertoire, InfoExplorer utilise la variable d'environnement **INFOLOCALE**. Vous pouvez indiquer une liste d'environnements locaux dans **INFOLOCALE** en les séparant par deux points (:). InfoExplorer tente de lire le premier environnement local de la liste puis les suivants jusqu'à trouver celui qui correspond.
3. Si aucune bibliothèque n'est trouvée avec **INFOLOCALE**, la variable **LC_MESSAGES** est alors utilisée.
4. Si aucune bibliothèque n'est trouvée avec **LC_MESSAGES**, la variable **LANG** est alors utilisée.
5. Si aucune bibliothèque n'est trouvée, InfoExplorer utilise par défaut celles installées dans **/usr/lpp/info/lib/en_US**.

Création de notes publiques InfoExplorer

Les notes publiques sont accessibles à tout utilisateur, en hypertexte. Elles sont créées par fusion puis déplacement des fichiers de notes privées.

Prérequis

1. Vous devez avoir accès en écriture au répertoire **/usr/lpp/info/data**.
2. Vous devez créer puis sauvegarder les notes privées dans un fichier de l'interface (interface InfoExplorer ou ASCII).

Procédure

Optez pour une des méthodes suivantes :

- Pour la bibliothèque InfoExplorer par défaut, utilisez la commande **mergenote** pour fusionner les fichiers de notes privées en un seul fichier. Indiquez le répertoire autres **/usr/lpp/info/data** pour y placer la nouvelle liste de notes publiques.
- Pour les bibliothèques publiques, les notes privées sont stockées dans le répertoire **$HOME/info/***NomBibliothèque*. Utilisez la commande **mergenote** pour fusionner les fichiers de notes privées dans un seul fichier. Indiquez le répertoire **/usr/lpp/info/data/***NomBibliothèque* pour y placer la nouvelle liste de notes publiques.

  Remarques :

  a. Les notes privées et les listes de notes sont sauvegardées dans les répertoires utilisateur **$HOME/info** et **$HOME/info/***NomBibliothèque***/notes**.

  b. Les notes publiques de la bibliothèque par défaut InfoExplorer doivent être placées dans le répertoire **/usr/lpp/info/data**.

Les signets créés par un utilisateur peuvent être copiés pour les mettre à disposition d'autres utilisateurs, en hypertexte. Les fichiers de signets sont sauvegardés dans les répertoires utilisateur **$HOME/info** ou **$HOME/info/***NomBibliothèque* avec l'extension **.bmk**.

Prérequis

Vous devez avoir accès en lecture-écriture aux répertoires utilisateur **$HOME/info**.

Procédure

1. Utilisez la commande **cp** pour copier un fichier de signets d'un répertoire utilisateur à un autre.

   Par exemple, pour copier le fichier de signets `review.bmk` du répertoire utilisateur **$HOME** `sharon`vers le répertoire utilisateur **$HOME** `donna`, entrez :

   ```
   cd /home/sharon/info
   cp review.bmk /home/donna/info
   ```

2. Dans InfoExplorer, redéfinissez le fichier de signets par défaut du nouvel utilisateur via la fenêtre Defaults Editor.

Transfert des signets InfoExplorer entre utilisateurs

Les signets créés par un utilisateur peuvent être copiés pour les mettre à disposition d'autres utilisateurs, en hypertexte. Les fichiers de signets sont sauvegardés dans les répertoires utilisateur **$HOME/info** ou **$HOME/info/***NomBibliothèque* avec l'extension **.bmk**.

Prérequis

Vous devez avoir accès en lecture-écriture aux répertoires utilisateur **$HOME/info**.

Procédure

1. Utilisez la commande **cp** pour copier un fichier de signets d'un répertoire utilisateur à un autre.

   Par exemple, pour copier le fichier de signets `review.bmk` du répertoire utilisateur **$HOME** `sharon`vers le répertoire utilisateur **$HOME** `donna`, entrez :

   ```
   cd /home/sharon/info
   cp review.bmk /home/donna/info
   ```

2. Dans InfoExplorer, redéfinissez le fichier de signets par défaut du nouvel utilisateur via la fenêtre Defaults Editor.

Index

Symboles

A

B

C

D

E

F

G

I

J

L

M

N

P

Q

R

S

T

U

V

W

X

Y

Z

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull Concepts de Gestion du Système AIX : – Système d'exploitation et unités

Nº Référence / Reference Nº : 86 F2 21KX 02

Date / Dated : Mai l 2000

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement.
Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : _____ Date : _____

SOCIETE / COMPANY : _____

ADRESSE / ADDRESS : _____

_____

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

Technical Publications Ordering Form

Bon de Commande de Documents Techniques

To order additional publications, please fill up a copy of this form and send it via mail to:
Pour commander des documents techniques, remplissez une copie de ce formulaire et envoyez-la à :

BULL ELECTRONICS ANGERS
CEDOC
ATTN / MME DUMOULIN
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

Managers / Gestionnaires :
Mrs. / Mme : **C. DUMOULIN** +33 (0) 2 41 73 76 65
Mr. / M : **L. CHERUBIN** +33 (0) 2 41 73 63 96

FAX : +33 (0) 2 41 73 60 19
E–Mail / Courrier Electronique : srv.Cedoc@franp.bull.fr

Or visit our web site at: / Ou visitez notre site web à:
`http://www-frec.bull.com` (PUBLICATIONS, Technical Literature, Ordering Form)

CEDOC Reference # N° Référence CEDOC	**Qty** Qté	**CEDOC Reference #** N° Référence CEDOC	**Qty** Qté	**CEDOC Reference #** N° Référence CEDOC	**Qty** Qté
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
[__] : **no revision number means latest revision** / pas de numéro de révision signifie révision la plus récente					

NOM / NAME : _____ Date : _____

SOCIETE / COMPANY : _____

ADRESSE / ADDRESS : _____

_____

PHONE / TELEPHONE : _____ FAX : _____

E–MAIL : _____

For Bull Subsidiaries / Pour les Filiales Bull :
Identification: _____

For Bull Affiliated Customers / Pour les Clients Affiliés Bull :
Customer Code / Code Client : _____

For Bull Internal Customers / Pour les Clients Internes Bull :
Budgetary Section / Section Budgétaire : _____

For Others / Pour les Autres :
Please ask your Bull representative. / Merci de demander à votre contact Bull.

PLACE BAR CODE IN LOWER LEFT CORNER

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

REFERENCE
86 F2 21KX 02

Bull

Utiliser les marques de découpe pour obtenir les étiquettes.
Use the cut marks to get the labels.

Concepts de Gestion
du Système AIX
Système
d'exploitation
et unités

86 F2 21KX 02

Concepts de Gestion
du Système AIX
Système
d'exploitation
et unités

86 F2 21KX 02

Concepts de Gestion
du Système AIX
Système
d'exploitation
et unités

86 F2 21KX 02