



Intelligence

StorIQ Archiver v. 0.8.2a

Installation et configuration

Auteur: Emmanuel FLORAC

Réf. ADM-STORIQARCH

Version: 1.0

Mise à jour: 13/01/2004

Publique

Contacts :

Emmanuel FLORAC, (eflorac@Intelligence.com)

tel: 01 46 97 25 46

www.Intelligence.com - info@Intelligence.com

Table des matières**1- Description**

- 1.1 - concept**
- 1.2 - fonctionnalités**
- 1.3 - installation**

2- StorIQ Archiver logiciel serveur

- 2.1 généralités**
- 2.2 principe de fonctionnement**
- 2.3 configuration**
- 2.4 paramètres**

3- StorIQ Archiver interface utilisateur

- 3.1 généralités**
- 3.2 configuration**
- 3.3 architecture du code**

4- structures de données

- 4.1 base de données : généralités**
- 4.2 base de données : description**
- 4.3 organisation des supports bandes**
- 4.4 organisation des supports disques**

Appendice A : liste des fichiers**Appendice B : liste des messages d'erreur**

1 – Description

1.1 Concept

StorIQ Archiver est un logiciel de sauvegarde et d'archivage centralisé. Il est multi-utilisateurs, multi-périphériques. Il peut sauvegarder tout espace de stockage accessible depuis le serveur applicatif, soit les ressources locales (disques...) ou réseau (partage NFS, SMB...). Les périphériques cibles peuvent être soit des systèmes à bande (DDS, DLT, AIT, Mammoth...) soit un espace disque dédié.

Des rapports sont générés d'une part dans un fichier journal, d'autre part par courriel de manière paramétrable soit à l'utilisateur, soit à l'administrateur défini dans le fichier de configuration principal, soit aux deux.

Evolutions prévues :

- pouvoir utiliser des périphériques cibles accessibles via le réseau.
- pouvoir utiliser des disques optiques, CD-R comme supports de stockage.

StorIQ Archiver est organisé en deux modules principaux : un logiciel serveur (démon) qui effectue les tâches demandées par les utilisateurs; une interface utilisateur de type web servant à définir les tâches. Les deux communiquent via une base SQL (mySQL).

Evolutions prévues :

- permettre l'utilisation d'autres moteurs de base de données (postgreSQL...)
- créer une interface client native.

1.2 Fonctionnalités

StorIQ Archiver peut sauvegarder tout fichier accessible directement en lecture depuis la machine hôte sur un périphérique à bande connecté à cet hôte. Ceci inclut tous les disques locaux et réseaux.

De même, StorIQ Archiver peut utiliser un espace disque dédié comme support de sauvegarde .

StorIQ Archiver peut également faire des sauvegardes différentielles ou incrémentales.

Une sauvegarde différentielle est la sauvegarde des seuls fichiers modifiés d'après une sauvegarde précédente. Une sauvegarde incrémentale est une série de sauvegardes différentielles répétées à intervalle régulier.

La restauration peut s'effectuer en place ou non. Dans le cas d'une restauration en place, il est implicite que les anciens fichiers seront écrasés.

Reportez-vous au manuel de l'utilisateur final pour plus d'informations.

1.3 Installation

1.3.1 Programmes externes

Certaines fonctions de StorIQ Archiver sont exécutées via des programmes externes. Assurez-vous que les programmes suivants sont bien présents :

-mt (gnu mt)

Permet de lire des paramètres et status des lecteurs de bandes (position sur la bande,etc).

-df

Permet de connaître l'espace disponible sur un espace de stockage donné (avant de faire une restauration par exemple).

-du

Permet de connaître l'espace occupé par un ou plusieurs fichiers avant leur sauvegarde.

-perl (5.6.1 ou supérieur)

l'interpréteur du programme.

-md5sum (V 2.0.11)

Permet de générer des sommes de contrôle sur des fichiers simples.

-GNU tar (1.13.19 ou supérieur)

Utilisé pour enregistrer les données sur bande ou sur disque.

Par ailleurs, procurez-vous également les logiciels suivants :

-mtx, tapeinfo (v 1.2.13 ou supérieur)

<http://mtx.sourceforge.net>

Permet de contrôler les robotiques et de lire leurs paramètres.

-MySQL (3.23 ou supérieur)

<http://www.mysql.com>

SGBD. Le programme n'utilise pas de fonctionnalités spécifiques, il devrait donc

fonctionner avec d'autres moteurs

-freeveracity ou veracity (OS non libres) (version 3.2).

<http://www.veracity.com/>

<http://www.freeveracity.org/>

programme de génération de sommes de contrôle pour une arborescence entière.

Par ailleurs StorIQ Archiver utilise les modules perl suivants :

-POSIX (format de données standard)

-DBI (interface générique de base de données)

-DBD:mysql (pilote MySQL pour DBI)

-IPC (gestion de processus et de signaux)

-Libnet::SMTP (envoi de courriels)

L'installation des modules perl est extrêmement simple, dès l'instant où on a un accès facile à Internet avec un débit raisonnable. Le cas échéant, il est préférable de déplacer le serveur devant recevoir les mises à jour.

L'installation des modules perl est alors complètement automatique. Il suffit d'utiliser la commande suivante (en tant que root):

```
perl -MCPAN -e shell
```

puis

```
install <NomDuModule>
```

au prompt *cpan>* puis de suivre les instructions. Si le système propose de mettre à jour perl lui-même, il faut bien sûr après l'installation quitter MCPAN, et s'assurer que le perl par défaut correspond bien à la nouvelle version et relancer avant de poursuivre l'installation des modules.

L'interface graphique nécessite :

-Apache

-php 4 (avec le module mySQL).

Il convient d'installer préalablement l'ensemble des programmes ci-dessus avant l'installation de StorIQ Archiver.

1.3.2 Installation de StorIQ Archiver

Il faut tout d'abord décompacter le tarball à l'emplacement choisi pour l'installation, par exemple /usr/local/storIQ.

Après s'être assurée que tous les programmes listés en 1.3.1 sont correctement installés et configurés, en particulier le démon MySQL (ou autre SGBD), créez un utilisateur de base de données pour l'usage de StorIQ Archiver (par exemple

'storiq'), donnez lui tous les droits sauf éventuellement 'alter' et 'grant' qui ne sont pas nécessaires, par sécurité; puis créez une base de données destinée à storiq (par exemple 'storiq_db').

Il n'y a plus qu'à exécuter le script SQL tablestructure.sql avec la commande suivante :

```
mysql -u root -p <nom base de données> < tablestructure.sql
```

afin de créer les tables nécessaires au bon fonctionnement de StorIQ Archiver.

Reportez -vous ensuite à la configuration des parties serveur et client.

2 - StorIQ Archiver logiciel serveur

2.1 généralités

Les dossiers liés au serveur sont :

Le programme principal est un démon que l'on peut démarrer manuellement ou au démarrage du système. Il génère un fichier de log contenant les rapports d'activité et d'erreur. Il prend ses paramètres de fonctionnement de différents fichiers de configuration. L'ensemble des fichiers et dossiers du serveur doivent être placés dans un dossier unique, par exemple /usr/local/storiq que nous appellerons ~.

~/bin : dossier contenant les exécutables, au minimum storiqarch.pl.

~/logs : dossier contenant les fichiers logs, peut être remplacé par un lien symbolique, par exemple vers /var/log.

~/etc : dossier contenant les fichiers de configuration.

Les dossiers ~/html, ~/temp sont uniquement liés à la partie client et peuvent donc être déplacés sans inconvénient.

Le programme exécute ses tâches via différentes applications externes lancées via des forks. Pour pouvoir accéder à tous les fichiers afin de les sauvegarder, il doit normalement être exécuté avec les droits administrateur (root ou wheel selon le cas).

2.2 Principe de fonctionnement

Le programme principal storiqarch.pl est composé de diverses parties. Bien que pour l'instant monolithique, il est prévu de le découper plus tard en plusieurs modules.

Pré-initialisation

Variables globales et paramètres de ligne de commande.

Initialisation

Ouverture du journal et des fichiers de configuration.
Initialisation du matériel et actualisation de la base de données.

Principal

Démarrage d'un processus indépendant pour chaque périphérique défini.
Récupération des paramètres de chaque périphérique.
Exécution d'une tâche pour chaque périphérique.
Attente de la prochaine tâche et reprise.

Principal (père)

Interception des signaux.
Contrôle des processus fils et fermeture du programme.

2.3 Configuration

La configuration de StorIQ Archiver s'effectue via un ensemble de fichiers de configuration.

le dossier `~/etc` doit contenir deux fichiers : le fichier de configuration principal `k_stor.conf`, le fichier de configuration des périphériques de sauvegarde `drives.conf`, et un dossier `~/etc/drivelist/` contenant les fichiers de configuration spécifiques à chaque type de périphérique de sauvegarde.

2.3.1 Format du fichier de configuration principal

Le fichier `k_stor.conf` contient trois types d'informations : les paramètres du courriel de l'administrateur, les paramètres de base de données, et les chemins et paramètres des applications utilisées par le programme (voir 1.3.1, « programmes externes »). Il est possible d'y insérer des commentaires en les précédant du signe `#`. L'ordre des paramètres est sans importance. Tous les paramètres sont sous la forme (les apostrophes sont indispensables):

`NomParametre='valeur'`

Le nom des paramètres est sensible à la casse.

NB: le fichier contenant un mot de passe d'accès à la base de données, il est important que ce fichier appartienne à root et ne soit lisible que par lui.

Les espaces de part et d'autre du signe « = », avant le nom du paramètre et tous les signes en dehors de l'apostrophe à droite sont ignorés.

Voici un fichier d'exemple :

```
#####
#
#       General configuration for K_Stor
#
#####

#####
## general info
#####
#administrator email
adm_mail='eflorac@Intelligence.com'

#secondary administrator email
otherAdm_mail='eflorac@Intelligence.com'

# SMTP server
smtp_host='ares.Intelligence.com'

#####
## database configuration
#####

# Database type ( only mysql supported right now )
DB_type='mysql'

# Database host and port
DB_host='localhost'
DB_port='3306'

# Database to use
DB_name='storiqarchDB'

#Database username and password
DB_user='storiqarch'
DB_password='grantme'

#####
## helpers programs
#####
# archive tool ( path to GNU tar )
TAR_prg='/bin/tar'

# tape control tool ( path to mt )
MT_prg='/bin/mt'

# loader control tool (path to mtx )
MTX_prg='/usr/local/bin/mtx'

# disk usage control
DU_prg='/usr/bin/du'

DF_prg='/bin/df'

# directory checksum generator (freveracity, or any
program/script with same behaviour)
dirchk_prg='/usr/local/bin/freeveracity'
```

```
# single file checksum generator (md5sum or similar program)
filechk_prg='/usr/bin/md5sum'

#dir. checksum filenames (separate with |)
dirchk_files='0veracty.vsf|0veracty.vrf|0veracty.vbf'
```

2.3.2 Format du fichier de configuration des périphériques de sauvegarde

le fichier drives.conf contient les paramètres uniques de tous les périphériques de sauvegarde que l'on souhaite utiliser avec StorIQ Archiver.

Chaque périphérique est défini par les paramètres suivants :

dr_name : nom unique du périphérique

dr_fam : type de périphérique parmi : *streamer*, *fixeddisk*, *removabledisk*, *WORMdisk*.

A l'heure actuelle, seuls les familles *streamer* et *fixeddisk* sont utilisables.

dr_type : nom du fichier de configuration de périphérique générique correspondant. Ceci doit correspondre à nom de fichier dans le dossier ~/etc/drivelist.

dr_dev : pour un périphérique à bande, le fichier de périphérique « sans rembobinage automatique », par exemple sous Linux /dev/nst0, /dev/nst1; sous Irix /dev/nrtape, etc. Pour un périphérique à accès direct (disque dur), le point de montage.

dr_scdev : périphérique SCSI (ou assimilé) d'accès direct au périphérique pour un périphérique à bande, par exemple sous Linux /dev/sg0, /dev/sg1; sous Irix /dev/scsi/scld110. Pour un périphérique à accès direct (disque), le périphérique à monter, par exemple /dev/scd0 sous Linux.

chgr_scdev : le seul paramètre optionnel, définit le périphérique SCSI (ou assimilé) à utiliser pour contrôler une éventuelle robotique de pilotage.

Il est possible d'insérer des commentaires en les précédant du signe #. Tous les paramètres entrés après le champ *dr_name* s'appliquent au périphérique considéré, jusqu'à la prochaine entrée *dr_name*. L'ordre des paramètres est sans importance après le paramètre *dr_name*. Tous les paramètres sont sous la forme (les apostrophes sont indispensables):

NomParametre='valeur'

Le nom des paramètres est sensible à la casse.

NB: Les espaces de part et d'autre du signe « = », avant le nom du paramètre et tous les signes en dehors de l'apostrophe à droite sont ignorés.

Voici un fichier d'exemple, avec un périphérique à bande doté d'une robotique, un

dérouleur de bandes simple et un disque de stockage :

```
dr_name='ExabyteId'  
dr_fam='streamer'  
dr_type='Exabyte_215M'  
dr_dev='/dev/nst1'  
dr_scdev='/dev/sg5'  
chgr_scdev='/dev/sg4'  
  
dr_name='DAT'  
dr_fam='streamer'  
dr_type='genericDDSdrive'  
dr_dev='/dev/nst0'  
dr_scdev='/dev/sg2'  
  
dr_name='Corbeille'  
dr_fam='fixeddisk'  
dr_type='genericdisk4GB'  
dr_dev='/mnt/corbeille'  
dr_scdev='/dev/sg0'
```

2.3.3 Format des fichiers de définitions de type de périphériques

Les fichiers de définitions contiennent les informations suivantes : les caractéristiques physiques du périphérique (nombre de dérouleurs, de bras, de portes, de slots...), le type de cartouche ou de bande utilisé et ses capacités, et les performances moyennes du périphérique.

Au fur et à mesure des tests et des validations, StorIQ Archiver sera fourni avec des fichiers de définitions pour un nombre de plus en plus important de périphériques. Il est généralement très simple d'adapter un fichier existant. Le seul paramètre qu'il peut être utile d'éditer est *tp_blk*

qui définit la taille de blocs logiques à utiliser pour le périphérique concerné. Il est recommandé d'utiliser la même taille de bloc pour tous les périphériques de même type (paramètre *tp_type*) afin que les supports (bandes, disques...) de même type soit lisibles sur tous les périphériques compatibles.

Pour l'instant seules les valeurs 0 et 1 sont supportées pour les paramètres *tp_drv*, *tp_changers*, *tp_door* et *tp_outp*.

Il est possible d'insérer des commentaires en les précédant du signe #. L'ordre des paramètres est sans importance. Tous les paramètres sont sous la forme (les apostrophes sont indispensables):

NomParametre='valeur'

il est préférable de définir les paramètres inutilisés à 0 ou " (chaîne vide).

NB: Les espaces de part et d'autre du signe « = », avant le nom du paramètre et tous les signes en dehors de l'apostrophe à droite sont ignorés.

Voici un fichier d'exemple :

```
#
#      Tape drive information
#      for Exabyte 215M Mammoth-2 autoloader (single drive)
#
#####
# general hardware
#####

# number of tape drives (streamers)
tp_drv='1'

# number of changers
tp_changers='1'

# number of autoloader slots
tp_slots='15'

# how many in/out slots
tp_outp='0'

# how many doors
tp_door='1'

# SNMP capable
snmp_capable='Y'

#####
# general tape info
#####

# acceptable tapes
tp_type='Mammoth, Mammoth-2'

# minimum and maximum tape size, no compression, in MegaBytes
tp_minsize='2500'
tp_maxsize='60000'

# typical hardware compression ratio
tp_hwcompress='2.5'

#####
# performance related infos (edit to fit your needs)
#####

# drive thruput in KBytes/s
tp_speed='4500'

# preferred block size in 512 bits/blocks
tp_blk='16384'

# Time for an unload/load operation cycle in seconds
tp_cycle='60'
```

2.4 Paramètres du programme

Le programme reçoit des paramètres de trois façons différentes : via les options de ligne de commande, via les fichiers de configuration (voir 2.3), via les signaux.

2.4.1 Les signaux

Actuellement seul les signaux TERM et CHILD sont gérés. En pratique l'envoi d'un signal TERM (ou signal 15) sur tout processus du programme (via les commandes *kill* ou *killall*) est intercepté et commande l'arrêt de tous les processus dès la fin des tâches en cours.

2.4.2 Options de ligne de commande

les options de ligne de commande sont destinées au débogage et à l'aide à la configuration.

Les options peuvent être combinées dans n'importe quel ordre.

Options:

-d : mode « debug » (débogage). Toutes les commandes externes, toutes les requêtes de base de données et certaines valeurs de variables sont systématiquement loggées.

-v : mode « verbose » (bavard). Les opérations principales sont loggées, à savoir : l'ouverture des fichiers de configuration, le contrôle des périphériques, la connexion à la base de données, le démarrage et l'arrêt des tâches, etc.

-i : mode « init only » (initialisation uniquement). Le programme lit ses fichiers de configuration, contrôle les périphériques et sort. Permet de simplement vérifier que les fichiers de configuration sont corrects sans risque.

-o : mode « run once » (une seule exécution). Le programme exécute une seule tâche par périphérique, puis sort. Permet de déboguer le déroulement des tâches.

-e : mode « exit when done » (sort si plus de tâches). Le programme s'exécute normalement mais s'arrête s'il n'y a plus de tâches à exécuter. Peut être utilisé pour le débogage du déroulement des tâches, ou pour contrôler le programme manuellement.

3- StorIQ Archiver interface utilisateur

3.1 généralités

Le logiciel client StorIQ Archiver est une interface web écrite en php, qui permet une utilisation à distance aisée et est compatible avec tous les navigateurs supportant les cookies (Netscape 3 et supérieur, Internet Explorer 4 et supérieur...).

Les fichiers constitutifs de l'interface client sont rassemblés dans le dossier ~/html; les sous-dossiers en sont ~/html/images qui contient les images de l'interface, ~/html/css qui contient le fichier css (Cascading Style Sheet) correspondant, et ~/html/fonts qui contient des polices utilisées pour générer des étiquettes pour les supports d'archivage. Au même niveau doit se trouver également un dossier ~/temp où sont créés les fichiers temporaires comme les dumps utilisateur de la base de données.

3.2 Configuration

Bien que les fichiers de l'interface utilisateur se trouve dans un sous-dossier de l'arborescence StorIQ Archiver, il n'y a pas d'inconvénients à les déplacer pour peu que l'on respecte les conditions suivantes :

- Tous les fichiers html et php doivent se trouver dans le même dossier;
 - les dossiers css/, images/ et fonts/ doivent se trouver dans ce même dossier;
- un dossier accessible au serveur http nommé temp/ doit se trouver au niveau supérieur.

Il est actuellement nécessaire que la directive REGISTER_GLOBALS soit à ON au niveau de la configuration de PHP; elle est par défaut à OFF sur les versions de PHP supérieures à 4.2.

La configuration de l'interface s'effectue au niveau d'un unique fichier php, dbuser.inc.php. Les paramètres à modifier sont les suivants :

- Le nom d'utilisateur et le mot de passe à utiliser pour l'accès à la base de données (ce peut être le même que pour le serveur, les paramètres sont en tout cas identiques);
- le nom de la base à utiliser;
- éventuellement le nombre de lignes à afficher dans les pages à résultats multiples (réglage de la pagination).

Dans la mesure où ce fichier est un code actif, n'y faites pas de modification autre que les valeurs entre apostrophes ' sus-citées...

Voici un exemple de fichier de configuration:

```
<?
// ici définir le user/pass pour l'accès base de données

// définition :
$dbuser='storiqarch';
$dbpass='grantme';
$dbdatabase='storiqarchDB';

// longueur maximale des listes de résultat:

$listlength=10;

// ***** ne modifiez rien au-delà *****
// *****
// connexion et permissions user

if ($mysqllink=mysql_connect( 'localhost', $dbuser, $dbpass))
{
    $mydb=mysql_select_db($dbdatabase, $mysqllink);
    $myquery="SELECT KS_SAVE, KS_BACKUP, KS_REST, KS_LIST,
KS_ADMIN, KS_JOBED FROM storiqusers WHERE KSLOGIN='$username'";
    $myresult=mysql_query($myquery, $mysqllink);
    $permissions=mysql_fetch_array($myresult, $mysqllink);
}
else
{
    readfile ("./index.html");
    exit;
}

?>
```

3.3 Architecture du code

Le logiciel client est constitué d'un nombre de scripts php nommés par ensemble de fonctions, suivi d'un ordre dans la séquence d'action, suivi d'un nom :

-les scripts destinés à l'administration:

admin-1-main.php, admin-2-archlist.php, admin-2-dbadmin.php, admin-2-logfile.php, admin-2-tpedit.php, admin-2-tplist.php, admin-2-usercreate.php, admin-2-useredit.php, admin-3-archedit.php, admin-3-label.php.

-les scripts destinés à l'archivage:

archive-1-selfiles.php, archive-2-archinfo.php, archive-2-backupinfo.php, archive-2-fileinfo.php, archive-3-createjob.php, archive-3-createprogjob.php.

-les scripts destinés à la restauration:

restore-1-selarch.php, restore-2-selfiles.php, restore-3-createjob.php.

-les scripts destinés à la gestion des tâches (jobs):

jobs-1-list.php, jobs-2-edit.php, jobs-2-files.php, jobs-2-message.php, jobs-2-progredit.php, jobs-2-proglist.php.

-les scripts destinés à la recherche:

search-1-display.php, search-2-selfiles.

Les scripts liés à l'accès au système ne suivant pas cette charte de nommage:

login.php, showlogin.php.

Et enfin le script tapelabel.php qui devrait être renommé admin-3-tapelabel.php.

4 structures de données

4.1 base de données : Généralités

Huit tables StorIQ Archiver sont nécessaire au fonctionnement minimal du système. Le script SQL permettant de générer ces tables est fourni et se nomme tablestructure.sql.

Les tables alljobs et progjobs sont emplies par l'interface utilisateur et déterminent les tâches à effectuer. Les autres tables (excepté la table des utilisateurs storiquers) sont remplies par le logiciel serveur.

4.2 base de données : description

Voici la description commentée des tables :

kstorusers	
KSLOGIN	identifiant unique d'utilisateur.
KSPASSWD	Mot de passe utilisateur.
KSHOMEDIR	Répertoire de travail de base de l'utilisateur.
KSFULLNM	Nom complet de l'utilisateur
KSEMAIL	Adresse courriel de l'utilisateur.
KS_SAVE	Drapeau d'autorisation d'archivage.
KS_BACKUP	Drapeau d'autorisation de sauvegarde.
KS_REST	Drapeau d'autorisation de restauration.
KS_LIST	Drapeau d'autorisation de visualisation des contenus et des bases de données.
KS_ADMIN	Drapeau de status administrateur.
KS_JOBED	Drapeau d'autorisation d'édition de tâches.

alldrives	
DRIVEID	Identifiant unique de périphérique.
DRNAME	Nom du périphérique (libre mais unique)
DRTYPE	Type de périphérique (nom du fichier descripteur)
TPFORMAT	Format de support (tel que défini dans le fichier descripteur)
DRDEV	Périphérique d'accès direct
DRSCDEV	Périphérique d'accès indirect
CHGRDEV	Périphérique d'accès à la robotique
TPDRVENDOR	Constructeur du périphérique d'entrée-sortie
TPDRMODEL	Modèle du périphérique d'entrée-sortie
CHGRVENDOR	Constructeur de la robotique
CHGRMODEL	Modèle de la robotique
DRPOSONTP	Position courante de la bande chargée
DRSPEEDK	Débit du périphérique d'E/S en Ko/s
DRCYCLE	Temps moyen d'un cycle de chargement de support
DRBLKSIZEK	Taille de bloc logique à utiliser sur le périphérique
DRPROCESS	PID du processus chargé de ce périphérique

allslots	
DRIVEID	Identifiant unique de périphérique.
SLOTNUM	Identifiant unique d'emplacement de support.
SLOTTYPE	Type d'emplacement de support parmi DRIVE (périphérique d'entrée/sortie), SLOT(conteneur), INOUT (passe-plat)
TAPEIN	drapeau de présence de support
TAPEID	Identifiant unique de support.
TPLABEL	Code barre unique du support.
XCHANGE	drapeau indiquant que l'emplacement est accessible manuellement.
STATUS	État du support présent parmi UNKNOWN (état inconnu ou pas de support), UNUSED (support jamais utilisé), INUSE (support déjà utilisé), FULL (support plein), ERROR (support inutilisable ou endommagé)
FROMSLOT	Emplacement précédent du support.

alltapes	
TAPEID	Identifiant unique de support.
TPLABEL	Code barre unique du support.
TPNAME	Nom du support (libre)
TPDATETIME	Date et heure de première utilisation
TPTYPE	Type d'usage du support, ARCHIVE (destiné à la conservation longue) ou BACKUP (destiné à un effacement régulier)
EXPIRES	Date et heure de péremption du support
NOACCESS	Drapeau d'interdiction d'usage du support
TPSPACE	Espace disponible calculé sur le support
TPFORMAT	Format de support (tel que défini dans le fichier descripteur)

allarchives	
ARCHID	Identifiant unique d'archive.
TAPEID	Identifiant unique de support.
POSONTP	Position de l'archive sur la bande en nombre d'enregistrements.
CREATOR	Utilisateur créateur de l'archive
ARDTTIME	Date et heure de création de l'archive
ARENDDTTIME	Date et heure de fin d'enregistrement de l'archive
ARCHSET	Liste d'identifiants d'archives appartenant au même groupe logique
ARCHNAME	Nom de l'archive (libre)
DESCR	Champ libre
ARSIZEK	Taille de l'archive en kilo-octets
WHAT	Liste des fichiers à traiter

allfiles	
FIDX	identifiant unique de fichier.
FILENAME	Nom du fichier
FILEPATH	Chemin d'accès original du fichier
FILESIZE	Taille du fichier en octets
FILEOWNER	Propriétaire système du fichier
FILEDATE	Date et heure système de modification du fichier
RECDATETIME	Date et heure d'archivage du fichier
TAPEID	Identifiant unique du support contenant le fichier
ARCHID	Identifiant unique de l'archive contenant le fichier
MD5SUM	Somme de contrôle du fichier (sauf pour les répertoires)

alljobs	
JOBID	Identifiant unique de tâche.
PRIORITY	Priorité de la tâche de 20 (à l'occasion) à -19 (urgence rouge)
JOBTYPE	Type de tâche parmi : RESTORE (restauration de données), SAVE (sauvegarde de données), LIST(lecture du contenu d'une archive), DIFFSAVE (sauvegarde différentielle d'une archive existante), COMPARE (comparaison d'une archive et de fichiers), VERIFY (non implémenté, contrôle d'intégrité de la bande), TEST(teste le positionnement de bande et le contenu d'une archive au hasard).
FROMDIR	Dossier dans lequel restaurer (RESTORE uniquement)
CMDOPT	Options de ligne de commande pour la commande d'archivage (tar)
WHAT	Liste des fichiers à traiter
TAPEID	Identifiant unique de support.
DRIVEID	Identifiant unique de périphérique.
REPORTTO	Liste d'adresses courriel vers lesquelles émettre un rapport.
STATUS	État de la tâche parmi TODO(à faire), RUNNING (en cours), DONE (fait), DELAYED (remis à plus tard parce que support indisponible), ERROR (tâche terminée sur erreur)
MESSAGE	Rapport d'exécution.
ISARCHIVE	Drapeau indiquant si la tâche porte sur une archive entière.
CREATOR	utilisateur créateur de la tâche
ARCHID	Identifiant unique d'archive.
ARCHNAME	Nom de l'archive (libre)
DESCR	Champ libre
JSTTIME	Date et heure effectif de début d'exécution de tâche.
JENDTIME	Date et heure de fin d'exécution de la tâche.

Progjobs	
PJOBID	Identifiant unique de tâche programmée.
JOBTYPE	Type de tâche parmi : RESTORE (restauration de données), SAVE (sauvegarde de données), LIST(lecture du contenu d'une archive), DIFFSAVE (sauvegarde différentielle d'une archive existante), COMPARE (comparaison d'une archive et de fichiers), VERIFY (non implémenté, contrôle d'intégrité de la bande), TEST(teste le positionnement de bande et le contenu d'une archive au hasard).
FROMDIR	Dossier dans lequel restaurer (RESTORE uniquement)
CMDOPT	Options de ligne de commande pour la commande d'archivage (tar)
WHAT	Liste des fichiers à traiter
TAPEID	Identifiant unique de support.
DRIVEID	Identifiant unique de périphérique.
REPORTTO	Liste d'adresses courriel vers lesquelles émettre un rapport.
STATUS	État de la tâche parmi TODO(à faire), RUNNING (en cours), DONE (fait), DELAYED (remis à plus tard parce que support indisponible), ERROR (tâche terminée sur erreur)
MESSAGE	Rapport d'exécution.
ISARCHIVE	Drapeau indiquant si la tâche porte sur une archive entière.
CREATOR	utilisateur créateur de la tâche
ARCHID	Identifiant unique d'archive.
ARCHNAME	Nom de l'archive (libre)
DESCR	Champ libre
JSTTIME	Date et heure programmé de début d'exécution de tâche.
JENDTIME	Inutilisé...
REPEATMODE	Mode de répétition de la tâche parmi None (pas de répétition), Day (quotidienne), Week (hebdomadaire), Month (mensuelle).
REPEATCOUNT	Nombre de répétitions.
GENJOB	identifiant unique de tâche générée.

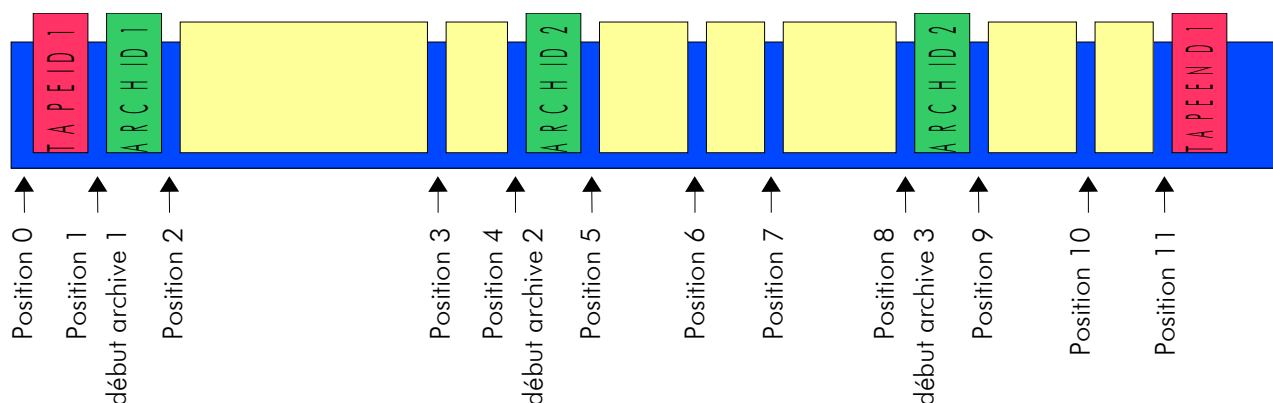
4.3 Organisation des supports bandes

Bien qu'il soit possible sur la plupart des lecteurs de bandes modernes d'utiliser le partitionnement de bande pour construire des index, StorIQ Archiver n'utilise pas cette possibilité pour ne pas limiter sa compatibilité. Le positionnement de la bande est donc repéré via un compteur interne d'enregistrements (bien que la commande `mt` inclue son propre compteur, celui-ci n'est pas non plus utilisé pour l'instant car cette fonction n'est pas présente sur tous les systèmes), un enregistrement correspondant à une unique opération d'écriture sur la bande indépendamment du nombre de fichier écrits lors de l'opération.

Chaque bande comporte en premier enregistrement un fichier nommé `TAPEID<tapeid>` permettant de la reconnaître facilement; en dernier

enregistrement, on doit toujours trouver son pendant TAPEEND<tapeid>. Entre ces deux enregistrements se trouvent les archives que l'on identifie via un marqueur ARCHID<archid>. Tous les enregistrements depuis le marqueur ARCHID<xxx> jusqu'au marqueur suivant ARCHID<zzz> appartiennent donc à l'archive ARCHID<xxx>.

Nous obtenons donc la structure suivante :



4.3.1 Enregistrement

Une opération d'écriture s'effectue toujours en fin de bande; il faut donc 1) aller jusqu'à la fin de la zone enregistrée 2) revenir en arrière d'un enregistrement pour se positionner sur le marqueur TAPEEND<tapeid> 3) récupérer la position courante sur la bande depuis le marqueur 4) se repositionner sur le marqueur afin de l'écraser lors de la prochaine écriture.

4.3.2 Lecture

Une opération de lecture (ou de restauration) s'effectue toujours relativement à une archive. On utilise donc 1) la position de l'archive telle que stockée dans la base de données pour rechercher le marqueur ARCHID<archid> correspondant; 2) lire tous les enregistrements jusqu'au prochain ARCHID<xxx> ou TAPEEND<xxx>. 3) En cas de restauration de plusieurs fichiers, il peut être nécessaire de revenir à la position initiale et recommencer la lecture autant de fois qu'il y a de fichiers selon l'ordre des fichiers sur la bande.

4.4 Organisation des supports disques

L'organisation des sauvegardes sur disque, pour demeurer cohérente avec la sauvegarde sur bandes, utilise également la notion d'archives, mais non celle de support (« tape »). Les données sauvegardées sur un disque sont donc copiées dans un dossier nommé ARCHID<archid>, mimant ainsi le fonctionnement de la sauvegarde sur bande. Ainsi il sera possible aisément (bien que cette fonction ne soit pas implémentée telle quelle) de migrer une archive disque sur bande, en sauvegardant sur bande le contenu d'un dossier archive.

NB: l'archivage sur disque préserve néanmoins les attributs des fichiers comme les dates de création.

Appendice A : liste des fichiers

Voici la liste actuelle (version 0.8.2a) des fichiers constitutifs du programme StorIQ Archiver tel que livrable dans une archive standard (tgz):

index.html	pour empêcher l'accès au listing de ce dossier au cas où...
README	instructions de base en anglais.
tablesstruct.sql	structure de base de données.
VERSION	version et copyrights.

Dossier ~bin/ :

storiq.pl	le programme lui-même.
Start.sh	script de lancement de StorIQ Archiver.
Stop.sh	script d'arrêt de StorIQ Archiver.
index.html	comme ci-dessus.

Dossier ~etc/ :

drives.conf	fichier de configuration de périphériques de stockage.
k_stor.conf	fichier de configuration générale.
index.html	Idem.

Dossier ~etc/drivelist :

Exabyte_215M	fichier de configuration pour robot Mammoth
Exabyte 215M.	
genericDDSdrive	fichier de configuration pour lecteur DAT (DDS) générique.
genericdisk4GB	fichier de configuration pour disque dur 4Go
générique.	
HPSureStore_1.9	fichier de configuration pour robot DLT HP SureStore 1.9.
index.html	Idem.

Dossier ~html/ :

admin-1-main.php	administration : première page.
admin-2-archlist.php	administration : liste des archives.
admin-2-dbadmin.php	administration : gestion de la base de données.
admin-2-logfile.php	administration : visualisation du journal.
admin-2-tpedit.php	administration : édition des paramètres de support.
admin-2-tplist.php	administration : liste des supports.
admin-2-usercreate.php	administration : création d'utilisateurs.

admin-2-useredit.php	administration : modification d'utilisateur.
admin-3-archedit.php	administration : modification d'archive.
admin-3-label.php	administration : édition d'étiquettes.
tapelabel.php	administration : affichage d'étiquettes.
archive-1-selfiles.php	sauvegarde : sélection de fichiers.
archive-2-archinfo.php	sauvegarde : entrée de paramètres d'archive.
archive-2-backupinfo.php	sauvegarde : entrée de paramètres de backup.
archive-2-fileinfo.php	sauvegarde : paramètres de fichier.
archive-3-createjob.php	sauvegarde : génération de tâche.
archive-3-createprogjob.php	sauvegarde : génération de tâche programmée.
dbuser.inc.php	général : paramètres de base.
forbidden.html	général : accès refusé.
index.html	général : retour au login.
login.php	général : authentification et menu principal.
showlogin.php	général : affichage de l'invite d'identification.
treemenu.inc.php	général : fichier « include » de gestion d'arborescence.
jobs-1-list.php	tâches : liste des tâches.
jobs-2-edit.php	tâches : modification de tâche.
jobs-2-files.php	tâches : affichage des fichiers traités.
jobs-2-message.php	tâches : affichage des rapports.
jobs-2-progedit.php	tâches : modification de tâche programmée.
jobs-2-proglist.php	tâches : liste des tâches programmées.
restore-1-selarch.php	restauration : sélection d'archive.
restore-2-selfiles.php	restauration : sélection de fichiers.
restore-3-createjob.php	restauration : génération de tâche.
search-1-display.php	recherche : affichage.
search-2-selfiles.php	recherche : sélection de fichiers.
Dossier ~html/css/ :	
style1.css	fichier de style XHTML.
Dossier ~html/fonts/ :	
FREE3OF9.TTF	fonte de code à barre.
FRE3OF9X.TTF	fonte de code à barre avec caractères étendus.

Dossier ~html/images/ :

bg.gif
browse.gif
continue.gif
database.gif
Dayrunner.gif
deselect.gif
DLT.png
Edit.gif
Edit_sm.gif
extract.gif
Folders.gif
in.gif
journal.gif
kanuma.gif
KnumaBondi.gif
Move.gif
out.gif
search.gif
select.gif
tape.gif
tapes.gif
titlebg.gif
titleleft.gif
titleright.gif
tools.gif
trash.gif
tree_collapse.gif
tree_end.gif
tree_expand.gif
tree_leaf.gif
tree_space.gif
tree_split.gif
tree_vertline.gif
user.gif
user-menu.gif
watch.gif

Dossier ~/logs/ :

K_Stor.log	uniquement si le programme a été exécuté.
index.html	Idem.

Dossier ~/temp/ :

index.html	Idem.
------------	-------

Appendice B : liste des messages d'erreur

Les messages se classent de la façon suivante par ordre de gravité :

JOB DELAYED : la tâche n'a pu être exécutée car le support prévu n'est pas disponible. Elle est remise à plus tard.

WARNING : une erreur s'est produite mais la tâche continue à se dérouler normalement.

ERROR : une erreur s'est produite et la tâche s'arrête, mais a été partiellement exécutée.

FATAL : une erreur grave s'est produite, la tâche n'est pas effectuée.

JOB DELAYED : TAPE \$rows{TAPEID} not available!

WARNING : drive description file \$thedrive not found!!

WARNING : error accessing drive, \$err Won't update hardware information!

WARNING : error accessing robot, \$err Won't update hardware information!

WARNING : Can't create tape status for tape \$Slots{TAPEID}', labeled \$Slots {TAPELABEL}', request \$request!

WARNING : Can't update changer status for tape \$rows{TAPEID}'!\n \$request failed.

WARNING : Following error occurred while saving files : \$err

WARNING : Can't insert file info in DB! request : \$request

WARNING : Can't update arch. \$archid end time to '\$SQLtime' and size to '\$FileSize' !

WARNING : can't get available space for tape \$tplabel!

WARNING : Can't update tape size for tape \$tplabel! \n Size should be '\$tpsize'!

WARNING : Not enough room on target filesystem : \$posrows[1] KB required, \$avspace KB available.

WARNING : Couldn't get available space on target Filesystem.

WARNING : the following error occurred while restoring : \$err

ERROR : couldn't run command : \$command !

ERROR : Can't update job \$rows{JOBID} status to DONE! !

FATAL : can't update DB job status ! \n request : \$request

FATAL : can't unload tape from drive \$i! Got : \$err

FATAL : you just encountered a missing feature!

FATAL : can't load tape into drive \$i! Got : \$err

FATAL : can't create new tape \$rows{TAPEID}

FATAL : Unrecoverable ERROR : tape \$rows{TAPEID}, label \$tlabel !!!

FATAL : Can't create archive \$archid ! request : \$request

FATAL : Can't create archive header \$archname! !

FATAL : Can't save archive header \$archname! error \$err !

FATAL : No archive defined! job failed

FATAL : '\$rows{FROMDIR}' is not a valid directory

FATAL : Can't get Archive position from DB!: \$request failed

FATAL : Wrong Label on Tape : waiting for ARCHID\$rows{ARCHNAME}, got \$listed
[0]

FATAL : Problem on tape : waiting for TAPEEND, got \$listed[0]

FATAL : tape drive response timeout!!!