Data Recovery and Backup Solution

for the RHEA SHR Adapter module

|  |  |  |
| --- | --- | --- |
| Date | Author | Version |
| 0.1 | Suranga | Initial draft |
| 0.2 | Suranga | Working on Wayne’s comments |

# Data backup procedures

Considering the importance + volume of data stored on the SHR database, backups should be made at the end of each working day.

Since we do not plan to run any extra-large queries on the SHR server, it is unnecessary to follow the AMPATH process of maintaining two servers, one to run complex queries and the other for data entry.

However, in order to ensure minimum down time, it is preferred to maintain a second backup server which can be swapped in if the first server crashes. For this to work, the two servers must be identical. The ideal way to do this is to use mysql data replication to ensure that both servers are always up to date.

Mysql data replication is currently used at AMPATH for this same purpose.

# Backing up / restoration of data stored in the file system

Data files stored in the server’s file system also needs to be protected.

**Backing up the App Data folder**

Since we do not currently store any flat data files in the App Data folder, it is unnecessary to backup this folder as frequently as the SHR database.

Ideally, the App Data folder should be backed up only whenever an administrator wishes to upload a new version of a module into the system.

**Backing up the OpenMRS war file**

The OpenMRS war file needs to be backed up only if the system is being upgraded to a later version.

# How to perform a database backup

The backup can be executed via a simple mysqldump command.

Alternatively, we may use the OpenMRS database backup module (<https://wiki.openmrs.org/display/docs/Database+Backup+Module>).

This module provides a ui interface for the backup procedure. As an added benefit, we may set the database backup process to be executed as a scheduled task via this module. Furthermore, admins are allowed to restrict which tables would be included in the backup.

Alternatively, we may use a script file to perform the database + file system backup as a scheduled cron job.

A sample script file (provided by James Arbaugh) is as follows-

# MySQL Backup Script Created by James Arbaugh

# Updated: Wednesday, January 4, 2012

# Dump the data from OpenMRS to a file

mysqldump -u root --password=mysqlpassword openmrs > /home/openmrs/.OpenMRS/openmrs\_backup.sql

# Connect to the windows server

export PASSWD=myserverpassword

mount -t cifs -o username=administrator //myservername/departments /media/network

# Copy it over to the Windows Server

cp -r /home/openmrs/.OpenMRS /media/network/IS/Backup\_Archive/

cp -r /srv/www/htdocs /media/network/IS/Backup\_Archive/.OpenMRS/

# Disconnect from the windows server

umount /media/network

# Backing up and transferring system metadata

System metadata refer to concepts, html forms, locations, roles, programs, etc) to be exchanged between different OpenMRS installations. Important metadata in terms of the SHR Adapter module are concepts and concept mappings.

For the SHR Adapter module to function well, it must have exactly the same concepts and concept mappings that exist in the Point of Care implementations which interact with it. In the event that changes are made to one remote implementation, then these changes must also be applied to the SHR Adapter database.

The most convenient method to do this would be via the OpenMRS metadata sharing module.

(<https://wiki.openmrs.org/display/docs/Metadata+Sharing+Module> )

Using this module, admins can select which Meta data should be backed up for transferring across implementations. This module eradicates the need to re-create meta data manually.

# Remote vs. local backup

Considering the importance of the data, backups should be physically moved to remote locations at least once a week.

# Alternate solution

The MySQL enterprise server backup system may be used as an alternative to the aforementioned tools in order to manage the SHR database.

Benefits of the MySQL enterprise server are –

* It keeps the database available during backup operations.
* It allows admins to backup only selected databases or tables
* It allows users to back up only the changes made since a previous backup, thus reducing the size of the backup file, and saving on memory