# Sync Module Overview

## session description

- session was lead by Dave Thomas (dthomas@pih.org)
- the sync module is in use by PIH Rwanda
- more documentation can be found here: Sync Module

## setup / overview

- sync operates on a spoke and hub model (parent server with children)
    - each child needs to initiated with a copy of the parent server's DB
        - this was done through database dumping
        - an alternate option to consider is to copy the mysql files instead of importing/exporting the DB, in order to save import/export time
    - *any way to not start out with clones and have some kind of merge done?*
        - short answer is no since the records would need to be reconciled
        - the key to doing that successfully is to be able to match up UUIDs, which is a non-trivial task
    - registration is done between the child to parent and parent to children
        - parent server and child servers have a static IP
            - it may be possible to use a dynamic DNS (DDNS) service (e.g. no-ip.org, dyndns.com) to get around this
    - sync process always runs on the child; configured to contact the parent server on a regular basis
        - *can we use the parent server as the operational server?*
            - yes, this is how it is currently used in PIH Rwanda
            - *is there a performance impact?*
                - none observed so far
- configurable settings
    - the amount of time when the child contacts the parent
    - timeout for when the child should stop contacting the parent
    - maximum number of tries for the child to reach the parent
    - whether sync'ed data is compressed or not (which is run through the openmrs configuration which can optionally be gzip)
    - the class of data to be sync'ed
    - at the technical level, it is sync'ing at the level of db tables
        - not able to sync subsets of patients, but this is a feature we'd like to implement in the future
- prerequisites
    - *did we need to beef up our servers for the sync?*
        - no, we have our servers running on off the shelf laptops with 3GB of RAM
    - had issues with the servers not being on proper UPS (we used laptops)
        - mysql databases can get corrupted if power is lost in the middle of a transaction
    - *what about network firewall considerations?*
        - this is done via HTTP post, so either port 80 (HTTP) or 443 (HTTPS)
    - bandwidth
        - our sites are on VSAT and performance has been fine even with all of the outages that we have. sometimes the bandwidth is 1kbps or less.
            - *how long do we need to have something up and running in order for the sync'ing to work?*
                - dependent on bandwidth, but from our experience, sending 200 records at a time, it takes about 5 minutes to sync
            - this can even be sync'ed via USB key
                - and there won't be duplicates even if it is manually sync'ed by USB and afterwards, the network is restored
            - we will be piloting this on GPRS modems; the issue is not bandwidth, but rather having fixed IP addresses on the children (that are using the modem)
                - however, this may be overcome by using a DDNS service (see above)

## functional questions

- still able to do updates on both the parent and the child?_
    - yes
- what is the format of what is being sent?
    - xml (serialized java objects)
        - *can the XML be consumed by other services?*
            - probably possible
- *is it possible to have grandparents? i.e. multiple layers of syncing between children and the parent?*
    - theoretically it could be possible
    - sync was built to be a child to parent relationship, where each node will communicate to a parent (if it exists) and its children (if they exist); children don't know about each other.
    - *is there any thought about doing a peer-to-peer relationship rather than a hierarchical relationship?*
        - this might be something to consider in the future
        - (a concern was raised by an audience member): not sure if this should be perceived as a solution for a national reporting system built off of (child) districts and (grandchild) clinics because of data access/privacy concerns; DIHS should be used instead
        - filtering might help to alleviate this challenge
- complex obs files able to be sync'ed?
    - may not be possible to do at the current time since it would require so much bandwidth to do

- 2 vs. 1 way sync
    - it is possible to use the sync module as an info path alternative (in a one directional way)

# similar/related efforts

- run a sync on an android device
    - the android device has a full DB copy
    - it is not made to handle conflicts
    - questions to think about:
        - *is there any opportunity to combine the efforts through refactoring?*
        - *because there has been trouble to get hibernate to run on android, is there any way to remove that dependency from the existing sync module?*