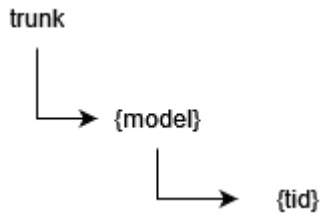
XINA Data Mining

Overview

Telemetry archives (also called "TID's") are imported into XINA by the process described here. "TID" is short-hand for "Telemetry Identifier", a text string that uniquely describes the experiment contained in the telemetry archive.

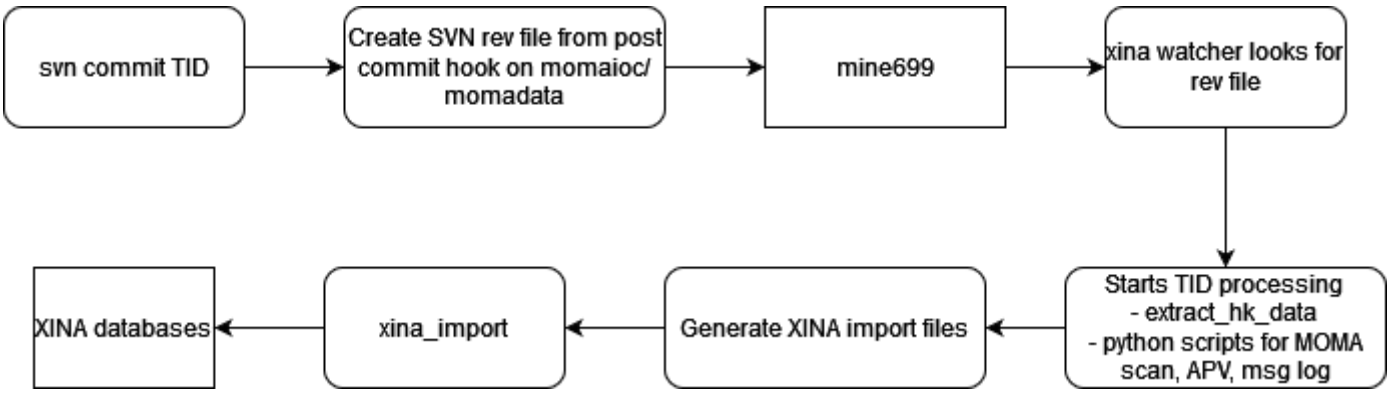
The SVN repository is typically laid out using the following heirarchy:



where {model} matches the XINA database name and {tid} is a unique description of the telemetry archive.

The XINA Data Mining process uses a shared network drive from the mining server (mine699.gsfc.nasa.gov) to the IOC server (e.g., MOMAIOC/DRAMSIOC/etc). Each mission uses a mission-specific folder on this shared drive to perform the data mining processes described below (e.g., /mine699/missions/moma/).

This flow chart shows the major steps taken during the data mining process:



Importing files to XINA

1. The user commits a TID to Subversion (SVN).
2. On MOMAIOC/DRAMSIOC/etc):

- An SVN post-commit hook calls the "svnlook" utility to generate a list of all revision changes in standard format
- "svnlook" output is saved to `/mine699/missions/*/rev/{rev#}.rev`.

3. The **XINA Commit Watch** utility running on MINE699 detects new rev file. Each mission has a separate running instance of this utility.

- Runs command(s) as specified in configuration file: `/mine699/missions/*/config.watch.json`

- Standard setup runs two commands: *XINA Commit* and *XINA Import*
- Once completed successfully, the rev file is moved to the rev archive: `/mine699/missions/*/rev/archive/`

4. The **XINA Commit** utility is run by *XINA Commit Watch* (above)

- Runs command(s) as specified in configuration file: `/mine699/missions/*/config.json`
- Reads in rev file generated by svnlook
- For each file listed in the rev file, the file is checked out from that revision into a local directory: `/mine699/missions/*/local/{file}`
- An additional directory is provided for temporary storage for mining processes if required: `/mine699/missions/*/temp`
- Files can only be processed individually
- After single file mining operations are finished, the local and temp directories are emptied

5. The **XINA Import** utility is run by *XINA Commit Watch* (above)

- Requires the **XINA Tunnel** utility to be running in order to connect to the XINA server (this is always kept running in the background on mine699)
- Reads and imports files in alphanumeric order from `/mine699/missions/*/import/{timestamp}_{rev#}`
- Moves the files as they are completed to the current revision archive directory: `/mine699/mission/*/archive/`
- Because the import process relies on operations happening sequentially, any errors shut down the import process until they are repaired

Handling Import Errors

XINA Commit Watch will not move on to the next rev file until the current file has been completed successfully

- If any operation in *XINA Commit* or *XINA Import* fails, the process returns an error code
- When *XINA Commit Watch* reads an error code from any sub-process it writes an empty file into the mission revision directory: `/mine699/mission/*/rev/rev.lock`
- As long as a **rev.lock** file is present *XINA Commit Watch* will not attempt to process any rev files

Debugging

Log output from *XINA Commit Watch*, *XINA Commit*, and *XINA Import* is stored in ```/mine699/mission/*/log```. - Errors generated during the "commit phase" are generally easier to fix, and my indicate a problem with a configuration file or script. After a commit phase bug is fixed, the rev.lock file can simply be deleted. *XINA Commit Watch* will perform a clean attempt to re-process the entire revision. - Errors generated during the "import phase" can be more complicated to fix because some operations may have already been imported before the failure, and running them again may cause new errors. It is necessary to check the archive directory for completed operations before attempting to re-process the revision. It may be preferable to perform corrections on the mined files depending on the error (for example, a typo in a field name), import the files manually, transfer the rev file to the rev archive manually, and then remove the rev.lock file.

Preload

For mining previously committed data, each mission has directory ```/mine699/mission/*/preload```

- Full checkouts of each mission are kept in `/mine699/mission/*/data`. This checkout is not related to the *XINA Commit Watch* process.
- Scripts are provided in each preload directory to create new task directories: `setup.sh task_name`
- Generated directories require a **config.json** file
- Provide scripts to process and import the backlog:
- **commit.sh, commit_bg.sh** - run across all files (in foreground or background)
- **commit_from.sh, commit_from_bg.sh** - run starting at a TID (in foreground or background)
- **import.sh, import_bg.sh** - import all generated files (in foreground or background)

Revision #11
Created 9 May 2024 16:47:44 by Michael Burkhart
Updated 13 May 2024 13:47:32 by Michael Burkhart