Database Patch Files

A database patch file is a data interpretation method that was conceived during MOMA project development to address the need to correctly handle differences in telemetry data structure and/or interpretation. The telemetry data differences may be produced by either inherent differences in the various models or updates to the hardware/software that break backwards compatibility.

Terminology

- Housekeeping row or row : a row in the database file
- Metadata field : a column in the database file
- Metadata value : a metadata field's value
- HKID (housekeeping ID) and Data ID are used interchangeably
- Unique identifier : the metadata field that should be used to determine a row's uniqueness. For patch files, the HKID is the unique identifier.
- Main database : the master database file that is always used first

What exactly is it?

A patch file is a tab delimited file that is nearly identical to the main database file. A patch file is characterized as follows:

- It should have the exact same format and metadata fields as the main database file.
- It is maintained in an Excel sheet and exported to a tab delimited text file when ready to be used by our software.
- It is stored in the TMDef directory just like the main database file.
- Unlike the main database file, a patch file may have negative HKIDs and does not require all metadata fields to have a value, i.e. blank fields are acceptable and encouraged to reduce repetition of information
- A patch file should only have the housekeeping rows the patch file is needed to "patch". It *should not* duplicate housekeeping rows that the patch file does not modify

A patch file may do the following:

- Replace a housekeeping row's metadata values
- Remove a housekeeping row
- Add a new housekeeping row

Why was it conceived?

When changes are made to the instrument that affect either the interpretation of the data or the actual structure of the data, you are left with two general choices:

- Update your data interpretation to support the new changes and declare that interpretation of data prior to the changes is no longer supported
- Come up with some scheme to support both the old data and the new data

We have opted for the second choice because we predict there will be a need to analyze old data. In previous missions this was semi-achieved by having multiple main database files. I say semi-achieved because they really only used it to support differences in the various models. Changes to the same model meant the old data was no longer supported unless a completely new main database file was created. Changes to a single housekeeping row would have to be made to all relevant database files. Because this process was cumbersome, error prone, and inefficient, the concept of a patch file was born.

Pros and Cons of patch files

Pros:

- Allows us to easily support interpretation of data from the various models and also old data
- Adheres to the DRY (don't repeat yourself) principle. We should only ever have to make a change in a single location. This should greatly reduce the potential for user error.
- It is a fairly simple convention but provides great power and flexibility
- Since patch files contain incremental changes, any order or combination of patch files may be applied to achieve the desired result

Cons:

- Changes to code that was tried-and-true, which may result in bugs
- Slightly more difficult to implement
- It is a new methodology, which means there may be unforeseen consequences
- The patch file requires the HKID to be the unique identifier rather than the name/tag. The ramifications of such a decision are not fully known.
- A telemetry file may have many file extensions. There is nothing inherently bad about this, but I imagine it may reduce readability.

How patch files should be handled by software

v1.0

In order to provide uniformity and consistency across our different applications, all code should conform to the following specification:

- A telemetry file's required patch files should be determined by its file extensions. Any extensions other than the mission extension (i.e. .sam for SAM, .mom for MOMA, etc.) should be interpreted as a patch file extension.
- For each file extension, the corresponding patch file will have the same name but with the .txt file extension. e.g. tm.mom.m1 should apply the "M1.txt" patch file.
- Handling of both the file extension and the patch file should be completely case insensitive, although establishing an accepted convention is encouraged for consistency. The currently accepted convention is for file extensions to be lower case, and the corresponding patch file to be in upper case.
- A telemetry file may have 0 or more patch file extensions.
- Patch files should be applied in the same order as the file extensions (left to right). This allows new patch files to replace older patch file's values.
- If a corresponding patch file can not be found, then an appropriate message should be presented to the user. What action the application should take after this (i.e. immediately exit or continue on) is domain dependent and is left up to the developer's discretion.
- The only unique identifier in a patch file should be the Data ID (HKID) column. Any replacing, removing, or adding of rows or values should be determined using the row's Data ID.
- For each housekeeping row in each patch file, the following actions should be taken as necessary:
    - If the Data ID currently exists, then each column in the patch file with a value should replace the existing value. If the column does not have a value, then no action should be taken for that column.

- If the Data ID does not exist, then a new entry should be added
- If the Data ID is negative, and the absolute value of said Data ID currently exists, then the existing housekeeping row should be completely removed. If the Data ID does not exist, then no action should be taken.
- If the Data ID field is empty, then no action should be taken.

Revision #2
Created 22 March 2023 17:45:40 by Nick Dobson
Updated 22 March 2023 17:53:02 by Nick Dobson