MAVEN IDL

Overview

The system of scripts for running the MAVEN IDL Toolkit to generate MATLAB-compatible HDF5 and csv files resides on ITF2 under the ~/tplot_auto directory, which contains the following subdirectories:

- prod - Production folder, where generated HDF5 and csv files are committed into SVN
- working - Working folder, where scripts operate. It contains cache files written by IDL/tplot and other intermediate files
- bin - Python scripts that generate and execute IDL .pro scripts
- idl - IDL modules and configuration files

Most instrument file generators have the same basic process:

- run_INST.py - Kicked off through launchd at specific weekday and time
    - Generates INST.pro file to run covering specified TID range
    - Run IDL with generated script
    - Ready to commit to SVN

Automated execution of scripts is handled through launchd, using an app called LaunchControl. The app generates XML-based configuration files saved in ~/Library/LaunchAgents

IDL is licensed through a license server maintained by SED.

For some background details on the IDL code used to generate these files, see NGIMS_IDL_Requests.docx

Software Updates

Development on the MAVEN IDL Toolkit has mostly settled down, so updates are not as frequent. However, it would be prudent to occasionally check for updates. Unfortunately, this is a manual process. Although the toolkit has an automatic update facility, it does not seem to be triggered when run noninteractively by the automated scripts.

To perform a manual update of the MAVEN Toolkit:

- Start idl (command line or GUI)
- Run the command: mvn_kp_check_version

On occasion I have noticed an update on the SDC Software page, which is not installed by this command. In that case, the recommended manual installation is:

- Uncompress zip file by double-clicking and move new directory into ~/idl (e.g. Toolkit_V2019-09-25_Team)
- cd ~/tplot_auto/idl

- rm Toolkit_Team
- Soft link newly created directory as Toolkit_Team (e.g. ln -s ~/idl/Toolkit_V2019-09-25_Team Toolkit_Team)

If you'd rather not follow this recommendation, the new IDL code must included in the IDL_PATH in the idlstart.pro file of the ~/tplot_auto/idl directory.

The publicly distributed MAVEN Toolkit only covers access to kp (or key parameters) data. There is an add-on used exclusively by the MAVEN Science Team, which is used to generate L2 files, etc. This is updated weekly by downloading the latest maven_sw.zip from the Berkeley site.

- update_maven_sw.py
  - Runs Mondays at 8am

This script unzips the latest IDL code into ~/tplot_auto/idl/maven_sw and keeps a copy of all the past zip files in ~/tplot_auto/idl/history

TIDs

A TID is a unique sequential identifier assigned to NGIMS data. Each orbit that MAVEN makes around Mars is divided into two TIDs. The even-numbered TIDs kick off at the beginning of inbound segment, go through periapsis and end at the start of the outbound segment. The odd-numbered TIDs cover the rest of the orbit. Primary science occurs in the first type of TID where the spacecraft dips to a lower altitude, while the odd TIDs are mostly composed of instrument idle time. Sometimes functional tests and calibrations occur in those TIDs.

The division into TIDs is exclusively a feature of NGIMS data, while the other instruments on MAVEN are generally on continuously and divide their data into day-length CDF files. Since NGIMS is idle for a large portion of the orbit, it's more convenient to compare data with other instruments by selecting the portion of the orbit where we are all taking science measurements. The mapping between TID number and SCLK time is recorded in a file generated by the NGIMS PDL tool and committed to SVN in the mavendata repository:

- spacecraft_data/L0/maven_pdl_TID_table.txt

A special copy of the directory containing this file is checked out under:

- ~/tplot_auto/prod/L0

A script is executed to convert SCLK times into unix time and save them in ~/tplot_auto/working/tidtimes.csv for use by instrument-specific scripts:

- update_tidtimes.py
  - Run daily at noon

STATIC

Most of the instrument data generated by this system is simply for the convenience of the NGIMS science team. However, the HDF5 files containing STATIC data are a necessary step in the NGIMS data processing pipeline, as they contain spacecraft potential data required to calibrate NGIMS science data.

This dependence on STATIC adds some timing constraints. Spacecraft potential does not appear in STATIC data until there are as-run SPICE kernels with spacecraft location. Updated kernels are delivered about a week after new instrument data arrives. STATIC data is then reprocessed after data is 2 weeks, 1 month and 2 months old, possibly refining the data.

Given that we don't want to continually reprocess NGIMS data, we have chosen to focus on two events: when spacecraft potential first becomes available and when we need to make a PDS delivery.

The python script that generates the S/C potential data at the earliest opportunity is the incremental script:

- run_static_incr.py
  - Runs Sundays at 12:30pm and Wednesdays at 8:30am
  - Looks for most recent sta#####.h5 file where ##### is the TID. To account for partials, the script goes back 25 TIDs and starts processing from there. This is a somewhat arbitrary number that seems to work.
  - Updated and newly generated files are automatically committed to SVN.

There is no error notification when the script has an issue. The maintainer of the script must be on the SVN notification list and look for the commit message expected on Sunday afternoon and Wednesday morning. If no such message is received, the most likely reason is that the SVN tunnel has gone down. From the ITF2 machine, log into mavenioc and restore the tunnel. The script has performed all the necessary 'svn add' operations, and you should only need to run 'svn commit' in the tplothdf/static directory. Also, there is a crude check for an automated process run amuck. The script will not automatically commit STATIC HDF5 files if there are more than 100 new files generated. In this case, an operator should review the output files and manually 'svn commit' if everything is in order.

When a PDS delivery is imminent, a manual run of the run_static.py script is necessary.

1. Determine the date range of the PDS delivery
2. Before you run the script, it's important that the end of this range is at least two months ago. Otherwise, the last scheduled STATIC reprocessing has not finished. Verify that STATIC L2 files in archive have a recent timestamp.
3. Determine the TID range analogous to this date range
4. Edit run_static.py to modify the value assigned to starttid
5. In the LaunchControl app, unload the run_static_incr.py script to avoid conflict with the weekly run by selecting it and from the menu Job/Unload
6. Use the Job menu to load the local.tplot.run_static job
7. Use the Job menu to start the local.tplot.run_static job, which executes the run_static.py script
8. Monitor the progress of the run (e.g. `tail -f sta.log` from tplothdf/static directory). A month of data takes about a day to process.
9. When complete, unload the run_static.py script and load the run_static_incr.py script to reenable the weekly run
10. SVN add and commit operations must be performed manually

In addition to the spacecraft potential, STATIC HDF5 files also include O+, O2+ and H+ densities.

MAG

MAG data generation is one of the more complicated processes. The final product is one csv file per TID, containing measured magnetic field and modeled crustal field at 1-sec resolution. There are two scripts involved, run_mag.py and split_mag.py. The first creates two intermediate files for each day, one with MAG data and the other with crustal field model data. The split_mag script merges the data and then splits them into TID-based files.

- run_mag.py

    - Runs daily at 8am
    - Looks for most recent mag########model.csv, identified by year, month and day. To account for partials, the script goes back 2 TIDs and starts processing from there. The processing ends on the last day where crustal field model data exists in the archive. Measured magnetic field data is output based on the days determined by the model availability, which is generally two weeks old.
- split_mag.py
    - Runs Mondays at 4:30pm

Generated files are not automatically added or committed to SVN. Manual svn add and commit is carried out upon request by the NGIMS science team.

LPW

LPW HDF5 data generation follows the standard pattern described in the Overview section.

- run_lpw.py
    - Runs Fridays at 10am
    - This script calls 'svn add' but does not automatically commit the new files

Manual commit is carried out upon request by the NGIMS science team. The LPW L2 data lags by at least a month. This process generates empty HDF5 files for the missing data, which are easily identifiable as they have a consistent size of 800 bytes. To avoid committing these empty files, the script only adds files over 1000 bytes to svn.

SWIA

SWIA data generation is similar to MAG. The final product is one csv file per TID, containing the following tplot data structures flattened into their component columns:

- mvn_swim_velocity_mso = Vx,Vy,Vz
- -VxB = Ex,Ey,Ez
- mvn_swim_density = n
- mvn_swim_pressure = p
- mvn_swim_temperature_mso = Tx,Ty,Tz

There are two scripts involved, run_swia.py and split_swia.py. The first creates a SWIA file for each day. The split_swia script merges the data and then splits them into TID-based files.

- run_swia.py
    - Runs daily at 3pm - DISABLED as data did not seem to be useful to team
    - Looks for most recent swi########.csv, identified by year, month and day. To account for partials, the script goes back 2 TIDs and starts processing from there. The processing ends on the last day where SWIA data exists in the archive.
- split_swia.py
    - Runs on Wednesdays at 4:30pm - DISABLED as data did not seem to be useful to team

Generated files are not automatically added or committed to SVN. Manual svn add and commit is carried out upon request

SWEA

SWEA HDF5 data generation follows the standard pattern described in the Overview section.

- run_swea.py
  - Runs Tuesdays at 10:30am - DISABLED as data did not seem to be useful to team
  - Script does not automatically svn commit or add

Manual commit is carried out upon request by the NGIMS science team.

Maintenance Notes

The act of running the MAVEN IDL Toolkit downloads and caches needed files into ~/tplot_auto/working/idl from the Berkeley site. It is populated with day-length files for those that have been loaded and converted to HDF5 or csv format. SPICE kernels are also saved in here. The bulk of the disk space used by this system in the form of the prod (SVN) and working directories resides on the external disk drive. The expectation is that there is plenty of space on ITF2_External and no special cleanup is necessary.

Some scripts have a heuristic for determining where to start processing, but many require periodic updates to the starting TID to avoid unnecessary reprocessing. SWEA, LPW and the tidtimes.csv generator rely on a manual edit of the starttid variable.

Revision #6
Created 22 March 2023 19:56:22 by Nick Dobson
Updated 14 July 2025 14:29:52 by Micah Johnson