SAM

SAM specific pages from the archive wiki.

- Index
- SAM Consumables
- SAM Data View
- SAM GATES Igor Tool
- SAM MAGE Igor Tool
- SAM Mailing List
- SAM PDL Setup
- SAM PDS Procedure
- SAM PUL Form
- SAM SAGE Igor Tool
- SAM Software Presentations
- SAM Software Suite Installation
- XINA SMS Tool

Index

- SAM PDL Setup
- XINA SMS Tool
- SAM PDS Procedure
- SAM Data View
- SAM Software Presentations
- SAM Mailing List

IGOR Software

- SAM SAGE Igor Tool
- SAM GATES Igor Tool
- ISAMS
- Tm2igor Quickstart

Deprecated Pages

- SAM Software Suite Installation --> See SAM PDL Setup
- Melissa Trainer
- Mike Wong
- 699util
- 699util User Scripts

SAM Consumables

A SAM consumable is defined as any piece of hardware in the SAM instrument that has a limited lifetime or a metric worth tracking. What exactly "lifetime" means is dependent on the specific piece of hardware we are interested in and a single consumable may have more than one way to define its lifetime. This page details the logic we use to extract the data for each consumable. Note that this is the master reference, which means the engineers and scientists should ensure that the logic detailed here is correct. All consumable data is available on XINA Online at https://ssed.gsfc.nasa.gov/xina/xo/tool/sam-timeline.

WRP

On Time

WRP on time is determined by calculating the total time that WRP HKIDs 334:WRP1 PWM, 336:WRP2 PWM break the threshold value of 50 for a TID.

Helium

He 1 Pressure

He 1 Tank Pressure is a reading of the current pressure; it is not a consumable where metrics are added between TIDs. The current He 1 Tank Pressure is calculated from PRES_1_He1 and T47_He_1 (HKIDs 89 and 43).

Because gas pressures are temperature sensitive, the pressures were temperature corrected to room temperature (22 degrees Celsius): Pressure = PRES_1_He1 / (T47_He_1 + 273) * (22 + 273).

He 2 Pressure

The He 2 Tank Pressure is a consumable recorded as a loss of pressure; it is a consumable starting at 2.3K psi and each TID records the amount of pressure lost. The He 2 Tank Pressure is calculated using regular expressions to measure Valve 33 and Valve 34 Open Times.

The loss in pressure is then calculated: Loss = (V33_on_time + V34_on_time) * .004 psi/sec.

O2 Gas

On Time

O2 Gas On Time was calculated using regular expressions to look for the quick opening and closing of Valve 35. Each time this event occurred, .02 seconds were added to O2 Gas On Time.

Cal Gas

On Time

Cal Gas On Time was calculated using regular expressions to look for the quick opening and closing of Valve 36. Each time this event occurred, .02 seconds were added to Cal Gas On Time.

TEC A & B

On Time

TEC A & B on times were determined using regular expressions to look for when the heaters corresponding to each TEC (A:61 | B:62) were open loop or off. The on time was determined by subtracting the ON time from the OFF time and summing all of these differences for a TID.

Cycles

TEC A & B on times were determined using regular expressions to look for when the heaters corresponding to each TEC (A:61 | B:62) were open loop or off. Each time a heater switched from ON to OFF, a cycle was added to the corresponding TEC.

SAM Data View

Download

- Mac OS X 10.6 (Snow Leopard) or later, 64-bit: SAMDataViewMacOSX10.6_2012-01-09.dmg (Version 1.03)
- Windows 32-bit: samdataview.msi

Version History

1.02

Release Date: 2011-12-19

- Start of version history.

1.03

Release Date: 2012-01-09

1.04

Release Date: 2012-03-13

SAM Deprecated

<small>**SAM-Specific**</small>

- 699util
- 699util User Scripts
- Software Presentations

<small>**Python Tools**</small>

- Python Tool Installation
- (DEPRECATED) Install Python 3 tools alongside old Python 2 tools

<small>**Other**</small>

- TunnelManager
- SVN (Subversion)

<small>**Misc. Installs (Required by many of our Python tools)**</small>

- gnuplot
- lxml
- LaTeX

SAM GATES Igor Tool

Latest News

NEW!!! Version 1.2 of GATES (for GCMS and TCD data analysis):

GATES_1.2.zip

SAM_Data_Analysis_1.2.zip

With now background and normalization calculations.

A first version of a graphical help: GC-SCI-GEN-UM-02-1-SAM_Soft_Igor_Pro_Manual_2.pdf

Datasets to download: Download it from the SAM FTP.

Example of TID 21568: 2010-04-28-09.38.46-21568-cpt6-exp1_sol3.zip

TID 50529 from TESTBED: 2012-09-27-11.40.45-50529-SS_ega_gc_tls.zip?

TID 50534 from TESTBED: 2012-10-04-10.39.42-50534-SS_ega-gc-tls-V140.zip?

Previous Versions

Version 0.7 of the SAM Analysis Monitor for Igor Pro is online: SAM_Data_Analysis_IGOR_0.7.zip

Version 0.6 of the SAM Analysis Monitor for Igor Pro is online: SAM_Data_Analysis_IGOR_0.6.zip

Version 0.5 of the SAM Analysis Monitor for Igor Pro is online: SAM_Data_Analysis_IGOR_0.5.zip

Data Preparation

A specific file hierarchy is necessary for the text files. Please download examples zip files to test with the tool. These data are obtained with a python script calling the other 699util python scripts and is automatically executed at FIMOC (CNES, Toulouse). This script could be uploaded into the 699util for all the users.

Quickstart

1. If you don't have Igor Pro, install it from Wavemetrics

2. Download the "SAM Data Analysis" zip file (above), unzip it.
3. Installation of the tool:
    - Open Igor Pro, go to "Help" menu, then click on "Show Igor Pro User Files" in the menu.
    - You need to copy the SAM Data Analysis Folder inside this directory (at the same level as "Igor Extensions", "Igor Help Files", "Igor Procedures", "User Procedures" folders).
    - You then need to make 2 alias. 1 alias of the SAM Data Analysis Folder (right-click to have the option to make the alias), and move it into the "User Procedures" folder. 1 alias of the SAM_Main.ipf file inside the "SAM Data Analysis" Folder, and move it into the "Igor Procedures" folder.
    - You should have now, at the same level, the "SAM Data Analysis" folder with 7 .ipf files, the "User Procedures" with "SAM Data Analysis Alias", and the "Igor Procedures" with "SAM_Main.ipf alias" file.
    - Restart Igor Pro, you should have a new SAM menu.
    - The set up procedure is the same for Windows.
4. Update of the tool:
    - In general, just delete the old "SAM Data Analysis" folder, and put the new one at the same level, then restart Igor Pro. It should be enough.
    - If it does not work, just restart the whole install procedure with the new version.
5. Download some dataset (above). You can unzip it into any folder, but as Xina put tm.sam in a sam/gse/data/ directory, you could create a data/text/ directory for this use.
6. First basic steps:
    - Launch Igor Pro. You work automatically with a new experiment.
    - Launch the "SAM Data Visu" from the "SAM" menu (or just use the "command/ctrl-1" shortcut)
    - Click on "Browse", to load data.
    - Choose a dataset folder (just click one time on the folder, and "choose" it, you should not open the folder)
    - The initial objective was to plot the TCD signal, so it is lightly GC oriented, but basically you can choose to plot the "Sg1" TCD signal (the first gain), and to view QMS signal with it.
    - So You can add bands (between 1 and 22, like 1,2,4 or 4-7, without space), masses (same way, between 2 and 537), and range of masses (start of the first mass,interval, as 10,10, to plot masses 10 to 19 and move every 10 masses).
    - You can add the column temperature as well, by checking the "Temp. Col." box inside the CDH Parameters.
    - Then click on "Plot GCMS" to create (or recall) the graph.
    - Basically, with the plot graph, the best way to look at different peak on TCD is to use the zooming functionality above axes: put the mouse cursor on the horizontal axe, and you can use the wheel (or the 2-fingers zooming on mac trackpad) to quickly zooming in to a peak. It is very efficient after some training.
    - Now, the background calculation works. You can choose a marquee zone, right click on it, and create a background.

Tool Tips

- First important thing: sometimes, it goes wrong... The easiest way could be to restart a new experiment.
- The data are loaded into a special Igor Pro structure: the data folders. You can access it with the "data browser" in the "Data" menu, and navigate into the data folders, by TID number. You can load as many dataset as you want. One of the next functionality will be the possibility to plot the same parameter or the same signal for all the TID experiments loaded.
- Several functionalities are working in a development version, and will be very soon available for a new stable version. This tool is used to make the GC QuickLook product, with automatic peak detection and calcul of amplitudes, elution time... but it is a little more tricky, and will be explained in the manual.
- See other tips on the SAM Fitting tool for IGOR.

Features for next version (hope every week)

- Enhanced visualization when the TCD signal is not present
- Automatic elution time on plots
- Pressure plots with TCD and MS signals
- New function for masses, bands sums
- New panel with metadata keywords
- Refinements of the GC (TCD) QuickLook procedure
- Valves and Markers on plots
- Add science related HK
- HK plots through different TIDs

SAM MAGE Igor Tool

Latest

Use the #igorpro channel of the SAM-MSL Slack for help requests and discussion.

Software: MAGE1.0.1.pxp

Overview

Derived from SAGE, the intent of MAGE is to work more generally for multiple instruments -- mainly lab instruments. MAGE doesn't deal with importing data. It expects waves and data directories to be in a pre-defined structure. The proposed organization is as follows:

Every data directory under the root directory is a separate dataset. In the SAM model, this corresponds to a TID. For a lab instrument, it would be a single run. The name of the data directory may contain alphanumeric characters and underscores, however, must not begin with a number.

The main goal is to quickly plot signal versus temperature using the familiar listbox interface of SAGE. Indicate which waves are associated with which mass by naming them with a suffix of '_m#' where '#' is any integer value. For example:

- _m44
- _m66

Temperature, time and generic x waves are acceptable to plot these values against. The prefix of the wave name indicates that it may be used as the x-axis quantities. The following are the only valid x-axis prefixes:

- time
- temp
- x

Anything named with a different prefix is assumed to be a y-axis value. For example, if you want the interface to be able to plot the counts per second of m/z 44 over temperature, the following waves must exist:

- cps_m44
- temp_m44

Other quantities can be plotted as "overlays" with independently labeled y-axes. They must have a corresponding x-axis wave with the same prefix naming convention (i.e. time, temp or x). For example, suppose you have some differential scanning calorimetry (DSC) data. Waves with the following names will allow the DSC data to be overlayed:

- dsc
- time_dsc

History

For background and historical discussion, see the #igorpro channel archive: igorpro_archive.docx

SAM Mailing List

You can sign up for the SAM Software Mailing list at https://lists.nasa.gov/mailman/listinfo/sam-software-release. Doing so will keep you notified of any software updates for XINA, SAM Data View, and Python Scripts!

SAM PDL Setup

Last Updated February 2, 2026

Prerequisites

- Mac - We recommend that team members filling the PDL role use a Mac to perform their PDL duties, as that platform is the most thoroughly tested.
- JPL RSA Token
- GSFC RSA Token or PIV Card
- Account on samioc, repos699

Terminal Setup

- Default shell in macOS 10.15 (Catalina) or later is zsh, however, python tools require sh or bash
- Recommended solution:
  - Create .zshrc file in home directory with the following lines:
  - `emulate bash`
  - `source ~/.bash_profile`
- Run the command `touch ~/.bash_profile` to insure that the file exists

Terminal alternative

- as of March 2023, the zsh shell on an M2 Mac was unstable with the SAM/MSL tools and would hang if more than one tab was opened. If the above solution works for you, great. If not:
- install bash (using brew)
- open preferences in terminal to open the bash shell you just installed. It will use the same .bash_profile script you created above.

SVN Access Notes

In order to access the SVN repository for SAM data, you need to be on the GSFC VPN and tunneled to the repository. The repository can be accessed using an RSA token or a PIV Card.

The easiest access method is a PIV card added to the ssh-agent. We suggest you create an alias for adding PIV card in your .bash_profile as shown:

```
alias pivmac='ssh-add -e /usr/lib/ssh-keychain.dylib; ssh-add -s /usr/lib/ssh-keychain.dylib'
```

You will be prompted to enter your PIN after running pivmac. The -e option removes any existing credentials before adding with -s. If you get a 'Card added' message, the process has worked. Once the card is added, commands like stunnel and rtunnel will pick up your card credentials and use them, automatically logging into the SVN host. The added card will remain in effect until you reboot. If the tunnel operation is not automatically logging

in, try rerunning the pivmac command.

If you are still using an RSA Token, you must enter your token's code every time you tunnel.

Installation

General Tools

- Install Homebrew
  - This will install Xcode command line tools as side effect: `xcode-select --install`
  - macOS 10.15 and above requires svn to be installed manually: `brew install svn`
- Install BBEdit, although any text editor will do
- Install X11
- Install Mattermost and follow the operations channel
  - `brew install mattermost`

Python Tools

- Install Python 3 from python.org
- From Terminal:
  - `pip3 install construct_legacy`
  - If you receive a permission error and recommendation to use --user, then:
  - `pip3 install --user construct_legacy`
  - Apparently, the prior method of installing 'construct<2.8' is not working on some systems. Look at the error messages when you try to do this, as I recall it directs you how to correctly install construct.
  - IF you use tmplot2.py (and probably some other newer scripts), install matplotlib:
  - `pip3 install matplotlib`
- If you have access to the code repository (repos699), tunnel into it and checkout python tools into your home directory
  - `ssh username@repos699.gsfc.nasa.gov -L 6990:localhost:3690`
    - Note: The above is equivalent to rtunnel, but that command is a python tool, so it will not be installed, yet
  - Then from a new terminal:
  - `svn co svn://localhost:6990/labcode/699util/branches/samops py699`
- Otherwise, download py699_Rev2818.tgz
  - Double-click to decompress
  - Move new "py699" directory to home directory (Move aside existing "py699" if one already exists)

NOTE: the repos699 method is recommended, as it will make python tool updates much easier

- Edit your ~/.bash_profile to include the following lines:
  `source ~/py699/shell/pyconfig.sh`
  `source ~/py699/shell/tids.sh`
- **Prior Bugs:** If you encounter any of the following messages, upgrade py699 to the latest.
  - "AttributeError: module 'collections' has no attribute 'Mapping'"
  - Missing windll on Macs

Limit Check/Plotref

- `brew install gnuplot`

    - The original color scheme from GNUPlot 4 is available by adding a line to your gnuplotrc file. Find gnuplotrc file by running gnuplot in a terminal window then typing “show loadpath”. Exit out of gnuplot. Navigate to the loadpath listed previously, then
    - open the gnuplotrc file – add this line to get classic line colors
    - `set colorsequence classic`
- `brew install ghostscript`
- `brew install enscript`
- Download and install MacTex from http://tug.org/mactex (*this might no longer be necessary, I didn't install it and PDL tools have worked fine for me. DA - Jan 2024)

Download Data

- Connect to GSFC VPN
- From a new Terminal window: `stunnel`
- From another Terminal window:
    - `cd`
    - `svn co svn://localhost:6991/samgse gse`
    - `svn co svn://localhost:6991/samdata`
    - `cd gse`
    - `ln -s ../samdata/fmdata data`
- With BBEdit create `.699config.INI` file in your home directory. Copy the text below, then search and replace 'myname' with your username.

NOTE: The first time you commit data to the repository, you will need to login to the `Authentication Realm: <svn://localhost:6991> SAMIOC`

- If you don't know your repository username and password, contact Kiran and Micah.

```
[SAM]
gse = /Users/myname/gse
tm_definitions = /Users/myname/gse/TMDef
tm_database = /Users/myname/gse/TMDef/SAM_TM_Database.txt
sclk_table = /Users/myname/gse/TMDef/SAM_SCLK.txt
heater_table = /Users/myname/gse/fswTables/heater.txt
data_root = /Users/myname/samdata/fmdata
repository_hostname = samioc.gsfc.nasa.gov
repository_port = 6991
repository_url = svn://localhost:6991/samdata

[SAM_TESTBED]
gse = /Users/myname/gse
tm_definitions = /Users/myname/gse/TMDef
tm_database = /Users/myname/gse/TMDef/STB_TM_Database.txt
sclk_table = /Users/myname/gse/TMDef/STB_SCLK.txt
heater_table = /Users/myname/gse/fswTables/heater.txt
data_root = /Users/myname/samdata/tbdata
repository_hostname = samioc.gsfc.nasa.gov
```

```
repository_port = 6991
repository_url = svn://localhost:6991/samdata
```

FEIGET

- Install Java
- Go to FEI GUI Page
- Click 'Launch the FEI5 GUI App for MSL'
- Control-click the resulting jnlp file and select 'Open'
- Change the pop-up menu that reads 'MSL' to 'MSLOPS'
- Click the button to the right of the menu with a green arrow inside a blue circle
- Login using your JPL RSA token
- Enter the 'msl_misc' directory by double-clicking it in the right pane
- Navigate the left pane, labeled 'Local Filesystem,' to your home folder
- Select the highest version of fei*msl for unix in the right pane and click the left pointing arrow
- From Terminal:
  - `cd /usr/local`
  - `sudo mkdir fei5`
  - `cd fei5`
  - `sudo tar xvf ~/fei*msl_unix.tar`
- Edit ~/.bash_profile to add the following line:
  - `cd /usr/local/fei5/latest/; source use_FEI5.sh; cd -;`
- From a new Terminal:
  - `fei5kinit`
  - Enter MSLOPS for Server group
  - Enter your username
  - Use JPL RSA Token to login
- `cd ~/gse`
- `ln -s ~/samdata/fmdata data`
- `cd data`
- `mkdir edrhub`
- `cd edrhub`
- `date "+%Y-%m-%dT%H:%M:%S" > lastfei.txt`
- `initcsvreports`

Ops

- Install JPL VPN
- Install Adium

PDL Bookmarks

- MSL Reports
- RAMPAGE
- Data Management Tool
- Password Support

Optional

- Install samdataview
  - Go to home directory and double-click "gse" then "Apps"
  - Open samdataview_MAC.dmg

- Drag "samdataview" (the Yosemite Sam icon) to "Applications"

SAM PDS Procedure

This procedure outlines the process for creating a SAM Reduced Data Record archive. Make sure to follow all the steps outlined here. Edit this page if anything changes.

Important People

- **Heather Franz:** Heather wrote the SAM RDR SIS (Software Interface Specification), a user's guide for the RDR archive. Heather generally attends the MSL DAWG meetings and will keep you informed if anything important happens there (you usually don't need to attend). Heather also generates the high-level QMS products.
- **Jean-Yves Bonnet:** Jean-Yves generates the high-level GC products.
- **Greg Flesch:** Greg generates all TLS products.
- **Susan Slavney:** Susie is the "Geosciences" PDS Node. She is our main point-of-contact between the SAM and PDS teams. If you have questions about PDS validation, etc., contact her.
- **Joy Crisp:** Joy is the chair of the MSL DAWG, but usually does not get directly involved in SAM's RDRs. I believe she announces the delivery schedule.

Before the Delivery

The MSL DAWG discusses a release schedule, usually shortly after the last release. You will usually have at least two months to get the products together, but since assembling the archive requires inputs from other parties, it's important to contact the other parties as soon as possible.

When you get the email from Susie labeled "MSL PDS Release 13 schedule", there will be an attachment that indicates when certain things are due. The first thing asked is "PDS asks the data providers for their archive readiness reports" which will be an email that will come out on the date (about one month before delivery is due). You want to plan to email Heather, Greg, and Jean-Yves a couple days before this date basically saying the following:

> Hi all,
>
> Release XX covers sols yy-
>
> zz, (START_DATE – END_DATE). Heather can you please release a list of the TIDs that we need for this delivery.
>
> Delivery is due DELIVERY_DATE so I will need all materials by 2_WEEKS_BEFORE_DELIVERY_DATE.

All the information will be in the attachments from Susie's email.

When you get the email asking for readiness report, just indicate delivery will be made on time and there are no changes unless Heather/Jean-Yves indicate something will be new or if you do not believe you will have delivery made in time.

Now just wait until you get all the materials from the team members and if someone has not delivered them on the date make sure you message them (if they forget they can usually get it to you in less than 24 hours).

Preparing the Inputs

SVN

Make sure you have all the current SAM data checked out fresh from SVN. Make sure that your working copy does not contain any unversioned, modified, conflicted files.

EDRs

Generating an RDR requires EDRs as input. EDRs should not be kept in SVN, and so you will probably need to download them separately. The process for getting EDRs is the same as it is for SAM Ops. Create an edrhub folder in the fmdata branch, and inside it run `fei5kinit` (to log in), `feiget.py` (to download the RDRs), followed by `movefei.py` to move them to the appropriate telemetry directories.

An alternative approach is to get someone on the SAM team to generate the EDRs and submit the TIDs with them included. I usually contact Benito Prats who is part of the SAM team, he can usually get them updated in a couple of days so just let him know ahead of time.

RDR Configuration Files

For each of the directories in the release, you will need to create a file named "rdr.config". This file contains information that cannot be obtained automatically by the RDR generator, or at least could not at the time the software was written. It is a newline-delimited CSV file, with the first column as keys and the second as values. I recommend you copy an existing one into each of the new TID directories, and then edit each one to fill out the information. All of the following lines need to be present:

```
SOURCE,F
MSL:SAM_GC_COLUMN_NUMBER,5
EXPERIMENT_TYPE,SPYR
VERSION,1
RELEASE,XX
PYRO_OVEN_NUMBER,1
```

- The **SOURCE** field is a one-character description of where the data came from: calibration data ("C"), ATLO data ("A"), testbed data ("T"), or flight data ("F"). I do not know if we will ever archive a data set from any one of these alternate sources.
- The **MSL:SAM_GC_COLUMN_NUMBER** is the number of the GC column. You can determine it by searching the message log with "tmmsg.py column".
- The **EXPERIMENT_TYPE** is a four-character description of the type of experiment. The values are defined by the SIS, and in the code they are defined by the SAM_EXPERIMENT_TYPES dictionary. They can take the following values. If you are unsure of the classification for a given experiment, ask Heather. Note that some of these will never apply to flight data (but could in theory apply to testbed or calibration data).
  - SPYR: Solid sample pyrolysis with GCMS
  - SDER: Solid sample derivatization
  - CSOL: Solid sample calibration
  - ADIR: Direct atmospheric measurement
  - AENR: Atmospheric enrichment
  - AMET: Atmospheric methane enrichment
  - ANGE: Atmospheric noble gas enrichment

    - SCMB: Solid sample combustion
    - CGAS: Gas calibration
- The **VERSION** field is the release version of the data set. It should start at 1, but if you ever re-release a data set for some reason (usually because there was an error in a previous delivery), you increment it.
- For **RELEASE**, put the current release number of the delivery. (Delivery due 11/4/16 is release 13).
- The **PYRO_OVEN_NUMBER** field is more difficult to obtain. I find it much easier to set it to 1 at first and debug it as I do tm2rdr.py. I will explain how this is done below during the processing part and will include what needs to be done when pyro oven is set to 2.

Once you create these configuration files, commit them to SVN.

High-level GC Products

Jean-Yves' inputs are the easiest to include, because he delivers them directly to SVN. For each TID he was assigned, he creates a directory called rdr_gc. The directory should have files named "notes.txt", "noise.csv", "species.csv", and "species.jpg". To my knowledge, he creates all four of these files for each TID. All you have to do is make sure that the files are there and named correctly.

High-level QMS Products

Heather does not have SVN access, and her deliveries are small enough for email. She generally delivers her products as ZIP files, but names them .piz so they don't get block from the email server, so just rename them back to .zip and they should extract easily. For each TID she delivers, create an rdr_qms directory in that TID, and unzip the contents of her ZIP file into that directory. Her files should only have the following names: "NOTES.TXT", "ATMCOMP.CSV", "ISOTOPE.CSV", "EGA.CSV", and "EGACOMP.CSV." If you are unsure what a file is supposed to be called, ask Heather.

High-level TLS Products

Greg does not have SVN access, either, but his deliveries are so large he cannot email them. He usually uploads the deliveries on dropbox and will send you the link via email to the products. For each TID, make an "rdr_tls" directory in that TID directory, copy the contents there, and add them all to SVN. Make sure the three high-level products are named "notes.txt", "abundance.csv", and "ratios.csv", respectively. (He will prefaces them with the TID and names the notes file TID.txt.)

Creating the RDRs

Once the inputs are created, the RDR can be created with a single command.

Change directories to the `fmdata` branch of the working copy. individually go into each of the TIDs and run `tm2rdr.py` and fix any errors that come up.

While running tm2rdr.py and you have pyro oven set to 1 you may see the following error:

```
INFO Processing QMS science data
Traceback (most recent call last)
```

```
File "/Users/briancorrigan/labcode/699util/scripts-pds/tm2rdr.py", line 2017, in <module>
  exit(main())
File "/Users/briancorrigan/labcode/699util/scripts-pds/tm2rdr.py", line 2003, in main
  process_qms_data(tmfile, rdr)
File "/Users/briancorrigan/labcode/699util/scripts-pds/tm2rdr.py", line 1371, in process_qms_data
  pyro_time, pyro_temps = zip(*pyro_temps)
ValueError: need more than 0 values to unpack
```

The usual fix to this is that the pyro oven is actually 2 so do the following:

- `tmfields.py --sclk 86 223 > tmfields.txt`
- Go into tmfields.txt and delete the first three lines that begin with # so it is just the 3 columns of data.
- change PYRO_OVEN_NUMBER in rdr.config to 2
- re-run tm2rdr.py

If this still does not fix it, just set PYRO_OVEN_NUMBER to 0 and it will just skip over them (I only had to do this once).

If you want to run all the TIDs at once you can try by using runall.py but I have had errors with this in the past and it is easier to debug issues that come up with pyro ovens by doing them individually.

You should have already created the pds_MMM-NNN.txt file. You can now use it as an input to a `runall.py`.

Execute `runall.py -d pds_MMM-NNN.txt tm2rdr.py`. Running this command will take a long time. The program will generate RDRs for each TID, one at a time. You should go get lunch or do something else. If anything goes wrong, all progress will cease, and you will probably have to fix something in `tm2rdr.py`.

If an error comes up while processing QMS data, this is probably because you indicated pyro oven 1 was on when really it was pyro oven 2, so modify the rdr.config file and change pyro oven to 2.

Notes for Debugging

To create an RDR for a single TID, just run `tm2rdr.py` just like you would any other python script.

If you need to re-run the RDR generation script over again, you should run `rdrclean.sh` for each TID before re-running. You will notice the second time you run, the script will run must faster. Each major step in the RDR generation process will create a `.pickle` file, which is basically a large stored Python object. If you need to change sections of the code that create the data that goes into these files (e.g., housekeeping extraction), you should delete the associated pickle file.

Assembling and Delivering

Creating an Archive Directory

Download the mslsam_1xxx directory from the current archive:

```
wget -r -nH --cut-dirs 2 -l 2 ftp://pds-geosciences.wustl.edu/msl/msl-m-sam-2-rdr-l0-v1/mslsam_1xxx/
```

Make sure the data folder is empty. Move all the files out of the index directory to a temporary place. The tools will generate the delta rows from this delivery and you will have to manually add them to these index files later.

Summarize Your Changes

Edit the file mslsam_1xxx/errata.txt. In "SECTION A", copy and paste an entry from a previous release and put it at the top of the section. Fill out the information with the current release date (consult Joy's email) and the changes in this delivery. If you did not update/change any old files, you can write "N/A" under "REASON FOR UPDATES". Save the changes.

Install Software

The first time you do a delivery from a computer, you will need to install `VTool`. This can be downloaded from the PDS website here: http://pds.nasa.gov/tools/label-validation-tool.shtml. To install it, unzip the TAR archive and put it somewhere on your filesystem. I put mine under /usr/local. Then, add the archive's "bin" directory to your path.

You will also need to download and install md5deep. You can get the source code here: http://md5deep.sourceforge.net/. It's been a long time since I installed it, but I imagine it's a normal `./configure`, `make`, `[sudo] make install` installation.

Link the new deliveries

Navigate to mslsam_1xxx/data. Run the following command with each TID in the delivery as an argument:

`rdrlink TID1 TID2 TID3 TID4 ...`

This will create symbolic links to the RDR products you just made.

Update Index Files

This step requires that you have the index files you downloaded from the archive and moved aside.

Start by going into mslsam_1xxx/ folder. Run `rdrindex.py` which should add 8 files to your index folder (4 lbl files and 4 tables for each level). Next, open a new folder window and go to the previous delivery's index folder. You are just going to concatenate the old folders index tables to the new ones.

- open PREVIOUS_RELEASE/mslsam_1xxx/index/l0_index.tab
- open NEW_RELEASE/mslsam_1xxx/index/10_index.tab
- copy every line in previous release file
- paste it on the first line of the new release file
- repeat for remaining tables

Now all that needs to be done is to update the label files to match the table files.

- open NEW_RELEASE/mslsam_1xxx/index/l0_index.lbl
- open NEW_RELEASE/mslsam_1xxx/index/10_index.tab
- indicate how many lines are in the index table
- change FILE_RECORDS in the label file to the number of lines that were in the index table

- change ROWS in the label file to the number of lines that were in the index table

Archive is now ready to be packaged and released.

Package the Archives

Navigate to the directory that contains mslsam_1xxx. Run `rdrpackage` which should create a "mslsam_1xxx.tar", "mslsam_1xxx_manifest", and "mslsam_1xxx_checksum". On Mac, you should be able to control + click the .tar file and hit "compress" which will make a much smaller .tar.zip.

Deliver the Archive

Now all you need to do is use an SFTP client to connect to wuftp.wustl.edu and copy the .zip as well as both the checksum and manifest over. Email Susie and let her know you have made the delivery and fix any errors that she has.

BELOW ARE OLD INSTRUCTIONS FOR PACKAGING I NO LONGER DO.

Navigate to the directory containing mslsam_1xxx. Create a directory called "rdrstage". Now, run the following command (with each TID in the delivery specified): `rdrsemipackage TID1 TID2 TID3 ...`. This will create a tar.gz archive in the rdrstage folder. If you want to include older TIDs in the delivery (perhaps because you reprocessed them), include them in this command as well.

The rdrsemipackage not only archives the files you want to deliver (and excludes the other stuff), but it also creates a "manifest" file and a "checksum" file, both of which Susie needs to validate we got her everything correctly. Look at the script to see how these files are created. There is also an rdrpackage script, which I used for the first delivery. I have not needed to use it since.

SAM PUL Form

The SAM PUL Form application generates JSON files for the XINA Import application.

Download

sam_pul_form.jar

Usage

`java -jar sam_pul_form.jar [ *path/to/tid/folder* ]`

History

1.0.1

- fixed issue not recognizing file name
- uses current working directory by default

1.0.0

- initial release

SAM SAGE Igor Tool

SAM SAGE - Software for Analysis of GCMS and EGA - was developed to aid in the analysis of SAM flight model data. SAGE will be used to model undersampled GCMS and complicated EGA data, calculate mass spectra based on modeled GCMS data, and determine peak onset temperature, peak maximum temperature, and peak area of modeled EGA data.

Latest

Use the #igorpro channel of the SAM-MSL Slack for help requests and discussion.

Igor 8 requires SAGE 1.8.4 or higher.

Software: SAGE1.8.4.pxp

Documentation/History

Manual: SAGEManualv1.3.pdf

SAGE new features.pptx in 1.7

Selected new features in 1.2, presented at the GCMS focus group meeting: SAGE_update_09_20_12.pptx

SAGE Release History

Data Preparation

The data that SAGE requires can be checked out from XINA and takes the form of a file named IGORdata.zip. Use 'Checkout All' or 'Checkout Secondary' to retrieve it. Unzip the archive inside the TID folder, and then load the TID folder from SAGE using the "Macros/Load TID Folder" menu item.

All SAM runs with QMS data should have an IGORdata.zip file generated and available alongside the tm.sam file. If there is one missing let the PDL team know.

Quickstart

1. If you don't have IGOR Pro, install it from Wavemetrics
2. Double-click the .pxp file of the latest version of SAGE
3. Save the .pxp file with a new name

4. Under the Macro menu select "Load TID Folder" and navigate to the TID folder you want to look at
5. Try out SAGE's data processing capabilities
    - Check the TM Msgs to determine which EM was used, in order to select the correct coefficients for deadtime correction (v1.2 only)
    - Optionally display TM markers and other overlay data
    - Drag the vertical lines to set a background region
    - Display different data corrections with the dropdown menu at the top left
    - Use the Plot tab to add/remove bands and masses from the graph
    - Select peak regions in the Peak tab
    - Open Peak Graph window with the Magnify button
        - In Time tab, adjust time range of peak region, if necessary
        - In Mass tab, select set of masses and bands to use in fit
        - In Fit tab, select fit function, choose initial peak parameters, and click Fit
        - Display mass spectra of fitted peaks, with options to output to NIST 11 format text file
        - In Review tab (> tab, v1.2 only), quickly compare the "raw" data to the fitted data of a single mass (or a single band) at a time
    - Export all fitted peaks to a chromatogram (in a .jdx format text file, which is compatible with ACD), with gaps between fitted peaks modeled with either a line fit or a moving average or connect data points.
6. Send any feedback to Micah Johnson and Anna Brunner

IGOR Tips

- Graph Interface
    - Click and drag marquee on plot, click inside marquee for Expand/Shrink options
    - Hover cursor over axis, scroll wheel to zoom in an out (side scroll slides data)
    - Option-Click to slide data with inertia, click to stop if it's still moving
    - Command-I shows cursors (drag cursors onto a wave in the graph to see info about the data point)
    - Command-T shows graph tools such as line or shape drawing capabilities
- KBColorizer Interface
    - If you close the 'Make Traces Different' window, newly added traces will be black.
    - Bring up the window again by selecting the last menu item in the Graph menu.
    - Nudging any of the three scroll bars in the Color Wheel section of that window will differentiate trace colors.
- Global Fit
    - If the fit does not converge, adjust the cursors if they seem off, or vary the value of the EMG modification factor. The following Igor-generated error messages are common and do not necessarily mean there is anything wrong with the data set or the program - they usually just mean the initial guesses weren't quite right. If you see these errors do not panic, just tweak the parameters. If they continue to occur, however, there may be something wrong besides the parameters.
        - "While executing FuncFit the following error occurred: singular matrix or other numeric error"
        - "While executing FuncFit the following error occurred: The fitting function returned NaN for at least one X value."
    - Global Fit does not do well if the peak you are trying to fit is not lined up precisely amongst the masses/bands - if there are slight time shifts, the fits do not look as good

Release History

Preview 1

- Import QMS count rates and marker data into waves
- Perform simple dead time correction
- Plot band data
- Selection of time range for background calculation

Preview 2

- Band and mass selection interface in Plot tab
  - Add/remove masses or bands by checking/unchecking the boxes
  - Highlighting a band automatically scrolls to corresponding mass range
  - Press up/down buttons to scroll quickly through masses and bands
  - Press spacebar when a mass or band is highlighted to quickly add or remove it

Preview 3

- In the Plot tab:
  - Quickly navigate between selected bands/masses by pressing j and k
  - In the mass list, masses corresponding to the highlighted band are color coded
- Peak selection interface in Peak tab:
  - Press "New" to select the time range of a peak using cursors
  - Rename peaks in peak list
  - Check box next to a peak name to highlight that range on the graph
  - Press "Magnify" to make a new Peak Graph window with a zoomed-in plot of the selected time range
- Peak Graph window:
  - In the Time tab, adjust time range of peak with cursors or text fields
  - In the Mass tab:
    - Set the sigma multiplier to specify a detection level (standard deviations above background)
    - Select/deselect masses and bands until the graph contains only the data with peaks at that time
  - In the Fit tab:
    - Select the peak center with the green square cursor, and the boundaries of the full-width half-maximum with the purple circle cursors
    - Press Reset to get cursors back if one gets accidentally dragged off the plot
    - Enter a modification (skew) factor if fitting to an Exponentially Modified Gaussian
    - Press Calc. Coeffs to generate guesses for the fit coefficients
    - Use calculated coefficients to perform a Global Fit

Version 1.0

- New peak fit functions: Lorentzian, exponentially-modified Gaussian
- Interface for entering initial guesses for multi-peak fit functions
- Simplified interface for calling Global Fit
- Method for quickly culling mass data that can't be fit
- Interpolation of GCMS data using fit coefficients
- Export file in JCAMP DX format, compatible with ACDLabs software
- Quick mass spectrum plot from fitted peak

Version 1.1

- Added peak fit check tab for looking at individual fit compared to mass data

Version 1.2

- Dead time correction interface and support for EM1/2 and custom coeffs
- Selective plotting of telemetry markers
- Implemented export tab
  - Set time range of export with cursor or time field
  - Snap cursor to visible marker
  - Fill-in data with line-fit or averaging option
  - Re-add background back to exported data
  - Show offset time as top axis of plot
- Remove bad fits from peak fit check tab
- Add prefix name to fit output waves so multiple fits can be saved

Version 1.3

- Support for IGORdata.zip archive
- Simple pyrolysis ramp temperature plotting from pyro_temp.log file
- EMG peak improvements
- Added "connect-the-dots" export capability for data that is not fit
- Command window readout for multiplier usage to assist with dead-time configuration
- Log/Linear plot toggle checkbox

Version 1.4

- Added 'Show Overlays' feature
  - Pyrolysis oven and sample temp plotting
  - GC temperatures and TCD signal plotting
- Plot mass vs temperature for EGA analysis from Pyro overlay tab

Version 1.5

- New overlays added
  - Low speed mux (HK1-154) data
  - Heater temperatures
  - MOT housekeeping data
  - Valve operations
- Support for dt.prms file
- Peak area calculations
- Support of band table update (1.5.2)

Version 1.6.1

- TLS housekeeping data overlay added

Version 1.7

- New macros: Smooth Data and Compare TIDs
- Additional option for custom_pyro_temp.txt file to define sample temperature

Version 1.8.1

- Igor 7 compatibility
- Pyro polyfit equations updated for TB and FM

Version 1.8.3

- Support for special files in TID
  - dt.prms - Dead time parameters
  - band_template.txt - Define masses that are contained in bands
  - fracmass.txt - Define special fractional masses that should appear in list checkbox interface
  - maptounit.txt - Defined fractional masses that should be mapped to unit masses

Version 1.8.4

- Igor 8 compatibility

SAM Software Presentations

- SAM_Data_Analysis_Overview.pptx
- SAM_Data_Processing.pdf
- SAM_PDL-PUL_Software_Training_Introduction.pptx
- SAM_PDL-PUL_Software_Training_Sam_Data_View.pptx
- SAM_PDL-PUL_Software_Training_Python_PUL.pptx
- SAM_PDL-PUL_Software_Training_Python_PDL.pptx

SAM Software Suite Installation

These instructions will install SAMDataview, Python 2.7, and SAM python utilities:

Mac OS X

Download and install the following files ***in order***:

1. Download the Mac OS X Python Installer from the Python.org homepage. Open the disk image and run the enclosed installer package. You will (probably) never have to update Python again, as far as SAM software is concerned. You can ignore this step when updating in the future.
2. Download the source code archives of the most current distribution of 699util. Instructions for installing Python modules from source code archives can be found on that page, as well.
3. Download the Mac Installer from the SAM Data View page. Open the disk image and run the enclosed installer package.
4. Head to the XINA Client Install Guide page and follow the instructions to install the XINA client.

System Requirements

SAM Data View for Mac was built for Mac OS X 10.6.8, 64-bit. In our limited experience with OS X 10.7 ("Lion"), SAM Data View installs and runs with no problems, though we are not yet certain that it will work on all systems with 10.7. We do not expect a high demand for a 32-bit Mac Version of SAM Data View, but if there is, we will build a 32-bit installer. The Python tools and XINA will work on any OS (provided Python and Java, respectively, are installed).

Windows

Download and install the following files ***in order***:

1. Download the appropriate installer from the Python.org download page: Windows 32-bit or Windows 64-bit.
2. Download the Windows installer from the most current distribution of 699util. Windows 7 and Vista users will need to right-click on the flie after it has downloaded and choose "Install as Administrator..."
3. Head to the XINA Client Install Guide page and follow the instructions to install the XINA client.
4. Download the current Windows Installer for SAM Data View (note: requires reboot after installation).

XINA SMS Tool

The XINA SMS Tool displays the state of samples within the SMS carousel on a cup-by-cup basis.

In the default view, a cup drawn in green represents a sample that has been placed and has not yet been pyrolized. A cup drawn in brown represents a used up sample. White cups have never had any sample placed in them. Yellow cups have had a history where a small nickel cup has been placed in them, used for an experiment on Earth (e.g. during TVAC for FM or anytime on the testbed) and then later the cup was removed.

When data is committed to the SVN repository, the smscuplog.py script is automatically executed to determine all the SMS motions that have been conducted during that run. Possible events include:

Event	Description
CupInOven	Cup is sealed in the oven (e.g., for preconditioning or after sample drop)
CupOutOven	Cup is removed from oven (e.g. after an EGA)
CupToSSIT	Cup is raised to the SSIT funnels presumably to receive sample
CupToShPunc	Cup is raised to shallow puncture station
CupToDePunc	Cup is raised to deep puncture station (e.g. prior to derivatization)

All of these event types result in an entry in the XINA database under Cup History, along with the TID and date. The CupInOven type also has an automatically populated Load value in lbf (pounds-force).

The database also includes fields which can be manually entered:

Field Name	Description
Sample ID	A two-letter designation, usually followed by an incremented digit for each (e.g. CB7)
Aliquots	Number of portions (typically 3 for early dropoffs and 4 during the FEST era)
Sample Used	Left blank until manually entering 'true' for the CupOutOven event after pyrolysis
Sample Desc	Any text may be entered here, but at minimum it should include the full name of the sample

Note: for testbed the aliquot value of 1 is typically used, since that is a term with meaning only in the FM Mars operations context

For those manually entering values, a CupToSSIT event where sample is received should be given a value for Sample ID, Aliquots and Sample Desc. A CupOutOven event should have the 'Sample Used' field set to 'true' if it was preceded by a pyrolysis.

In testbed operations, there is a special value for 'Sample ID' when the nickel cup-in-cup is removed. For this case, enter 'empty'. Then, the XINA app will display the cup in yellow indicating it can be used for future loads.