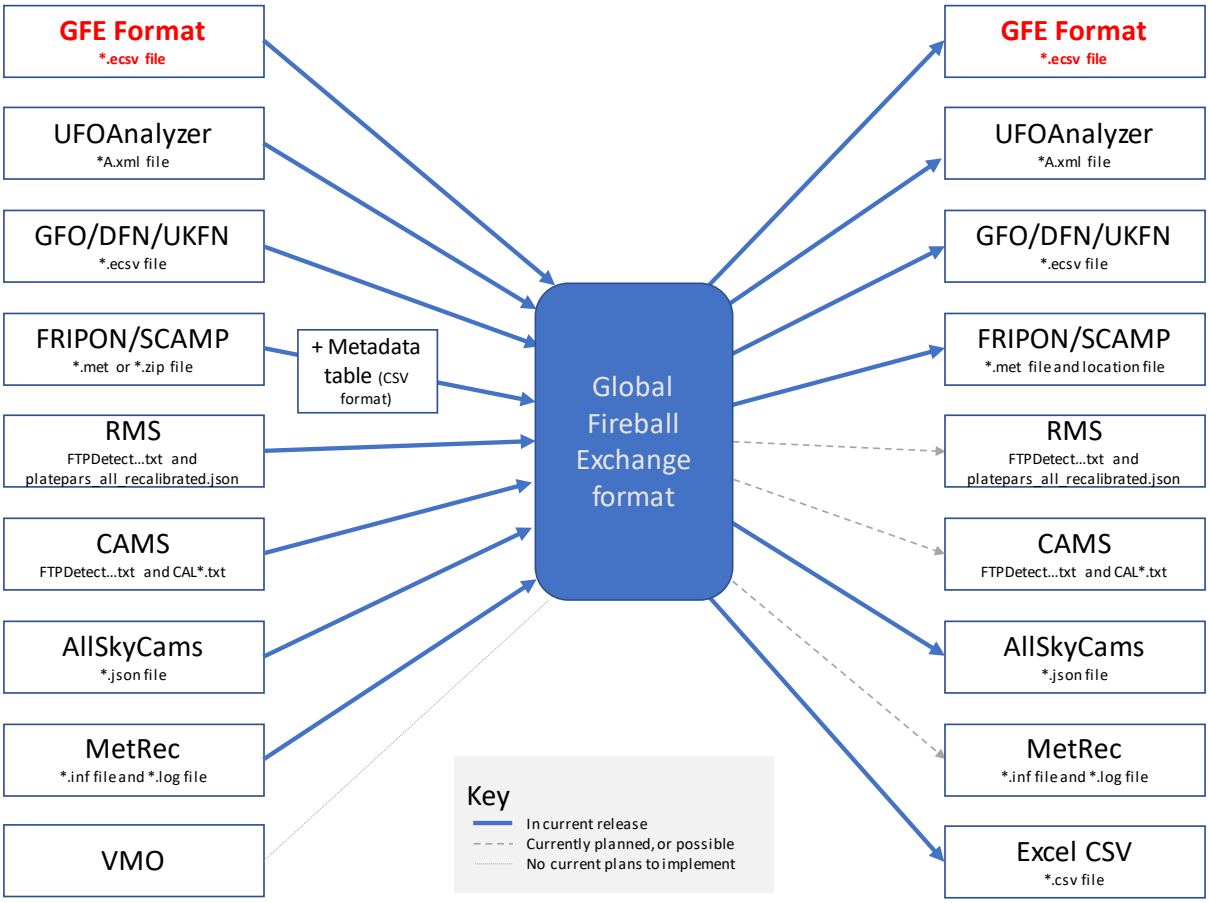
Fireball Data converter script – User guide

Version 4.35, 1st September 2021

Overview

This script converts fireball observations from several meteor camera systems into the Global Fireball Exchange (GFE) format, enabling sharing of data between networks, as below. In the diagram, formats that the script can read are on the left, and those that it can write are on the right:



Input will always consist of a single file of meteor data and (sometimes) an additional file of camera or location data. The script's output may consist of multiple files. If there is more than one output file, they are zipped together and served to the user as a single zip file.

1. How to use the script

The script is written in Python. First install Jupyter (available from the internet), then use Jupyter to open the script. From the Jupyter menu, chose "Kernel/Restart and Run All".

The script will display a file dialogue box, asking for the name of a file containing meteor data to convert. To narrow down the choice of files so that it's easier to find the one that you want, use the dropdown menu in the bottom right corner of the file dialogue box. If you are converting from a FRIPON *.met file or a FRIPON *.zip "Processed Multiple Event" file, you will need an active internet connection so that the FRIPON observatory location data can also be downloaded.

Once you've selected a file, you will be prompted for the file type that you want to produce. Choices given are "1=Global Fireball Exchange (GFE), 2=UFO, 3=DFN/UKFN, 4=FRIPON, 5=AllSkyCams, 9=ExcelCSV".

After that, you will be prompted for an output file name. If there is just one meteor in the input file and the output type is Global Fireball Exchange format, DFN or AllSkyCams, then there is just a single output file. In this case, the script will prompt you for a suitable name for that file. In all other cases, there will be multiple output files zipped together, so the script will prompt you for the name of a .zip file to be written.

Once you choose the output file/zip name, the script will write the file then display "finished".

2. What the script produces

The data files produced by the script depend on the answer to the prompt "1=Global Fireball Exchange (GFE), 2=UFO, 3=DFN/UKFN, 4=FRIPON, 5=AllSkyCams, 9=ExcelCSV" as below. They also depend on the number of meteors in the input file. Let's assume there are "N" meteors in the data that has been read, where N >= 1:

1. **Global Fireball Exchange format** – an *.ECSV file for each of the N meteors. Each output file is a single-meteor file in an Astropy extended CSV table format. This is very similar to "DFN" format (see 3 below), but omits DFN-specific information and contains additional orientation information which is relevant to non-all-sky cameras;
2. **UFOAnalyzer's** *A.XML file - a single-meteor file in XML format for each of the N meteors, plus a summary CSV file in R91 format;
3. **Desert Fireball Network (DFN/UKFN)** – an *.ECSV file, which is a single-meteor file in an Astropy extended CSV table format, for each of the N meteors. Similar to GFE format (see 1 above) but contains placeholders for DFN-specific information;
4. **FRIPON/SCAMP's** *.met file, for each of the N meteors, a single-meteor *.met file in SExtractor extended CSV table format, plus a small text file containing observatory location;
5. **AllSkyCams** – a *.json file, which is a single-meteor file in JSON format, for each of the N meteors; or
9. **Excel CSV** – for each meteor a *.csv file suitable for reading in Excel, with descriptive labelling of metadata and date/time in Excel number format as well as in a Y,M,D,H,M,S format.

3. How the script works

The script reads Global Fireball Exchange format, DFN and FRIPON files as Astropy tables and reads FRIPON "Processed Multiple Event" .zip files as a list of tables. The script reads UFOAnalyzer files as XML strings and reads AllSkyCams files as JSON strings. MetRec files are read using library routines published by ESA and imported into the script. RMS and CAMS files need to be read line-by-line as no library routine is suitable.

Once the file is read, about two dozen key pieces of metadata are picked out of the input and stored as an Astropy table. The point observation data for each frame or point in the observed image is also read and converted to a standardised form in the Astropy table. These data items are listed overleaf. The data for multiple meteors are stored as an array of Astropy tables. If the input for each data point only contains RA and DEC, it is assumed to be J2000 data and so precession is calculated and a conversion to AZ ALT coordinates is done. Similarly, if only AZ and ALT are given, this is converted to RA and DEC in the J2000 reference frame.

Once the output type is chosen, the array of tables is converted to the specific output type chosen and is written by an Astropy table writing function (in the case of GFE or DFN output), as a standard JSON file in the case of AllSkyCams data or line-by-line for UFO, FRIPON and Excel CSV output.

4. Thanks and credits

Thanks to Hadrien Devillepoix of Desert Fireball Network for providing the DFN Read/write code which is incorporated (in altered form) into this notebook.

Substantial development work was carried out in June and July 2020 by Nicholas Pochinkov of Dunsink Observatory, Dublin and Trinity College, Dublin. This includes the MetRec, CAMS and RMS data-reading code and the multi-meteor and zip file capabilities.

RA/DEC Alt/Az conversion and precession code is taken from RMS on the Croatian Meteor Network's GitHub site, copyright (c) 2016 Denis Vida.

The script is maintained by Jim Rowe of the UK Fireball Alliance, www.ukfall.org.uk, jim.rowe@ukfall.org.uk

5. Bugs, and future developments

There may be ways to crash the script. If you find problems, please let me know. Even better, please fix the problems and make a pull request on the UKFAll GitHub page at https://github.com/UKFAll/standard

Appendix – Variable names in "GFE" form, and their equivalents in the other formats.

Variable names in Global Fireball Exchange format are the same as in DFN format except where highlighted in orange, below.

Metadata

Status	*Explanation*	*Data type*	*Standard*	*Desert Fireball Net*	*UFOAnalyzer*	*FRIPON*	*RMS*	*CAMS*	*AllSkyCams*	*MetRec*
Essential	Observatory latitude	Float	obs_latitude	obs_latitude	lat	Latitude	lat	Latitude +north (d	lat	Latitude
Essential	Observatory longitude	Float	obs_longitude	obs_longitude	lng	Longitude	lon	Longitude +west (-	lng	Longitude
Essential	Observatory MSL	Float	obs_elevation	obs_elevation	alt	Altitude	elev	Height above WGS	alt	Altitude
Recc.	Network name	Text	origin	origin	['UFOAnalyzer']	['FRIPON']	['RMS']	['CAMS']	['All Sky Systems']	['MetRec']
Recc.	Name of location of obs	Text	location	location	lid	City	station_code	Camera number	station	Site code
Recc.	Name of the station	Text	telescope	telescope	lid_sid	Stations	Cam#	Camera number	device	CameraName
Recc.	Coded name of station (	Text	camera_id	dfn_camera_coder	sid (two letters)	Stations	Cam#	Camera number	station	CameraName
Recc.	Person or other name	Text	observer	observer	observer	[calculate]	'RMS'	Camera number	[location+telescope]	
Recc.	Camera make and mode	Text	instrument	instrument	cam	Camera	['Unknown']	Camera descriptio	[unknown]	
Recc.	Horizontal pixel count	Int	cx	NAXIS1	cx	[calculate]	X_res	Cal center col (colc	['calib']['img_dim']	TimeStampXPositi
Recc.	Vertical pixel count	Int	cy	NAXIS2	cy	[calculate]	Y_res	Cal center row (ro	['calib']['img_dim']	TimeStampYPositi
Recc.	File as obtained from ca	Text	image_file	image_file	clip_name	['Unknown']	[at top of capture]	file_name	org_hd_vid	inf.path
Recc.	Start datetime of clip	ISO datetime	isodate_start_obs	isodate_start_obs	[calculate]	[calculate]	[calculate]	[calculate]	[calculate]	
Recc.	Time of middle of clip	ISO datetime	isodate_mid_obs	isodate_mid_obs	[calculate]	[calculate]	[calculate]	[calculate]	[calculate]	
Recc.	Total clip length in seco	Float	exposure_time	exposure_time	[calculate]	[calculate]	[calculate]	[calculate]	[calculate]	
Recc.	Number of stars identif	Int	astrometry_numb	astrometry_numb	rstar	[0]	len(star_list)	[0]	[calculate]	[0]
Optional	Reference magnitude o	Float	photometric_zero	photometric_zero	mimMag	[0.0]	mag_lev	[0]	[0]	
Optional	Uncertainty in mag	Float	photometric_zero	photometric_zero	[0.0]	[0.0]	mag_lev_stddev	[0]	[0]	
Optional	Lens make and model	Text	lens	lens	lens	['Unknown']	['Unknown']	Lens description	[unknown]	[unknown]
Optional	Azimuth of camera cent	Float	obs_az	[0.0]	az	[0.0]	az_centre	Cal center Azim (d	['center']['az']	Center of plate Az
Optional	Elevation of camera cer	Float	obs_ev	[0.0]	ev	[0.0]	alt_centre	Cal center Elev (de	['center']['el']	Center of plate Alt
Optional	Rotation of camera	Float	obs_rot	[0.0]	rot	[0.0]	rotation_from_hor	Cam tilt wrt Horiz (	[0]	[0]
Optional	FOV horiz - degrees	Float	fov_horiz	[0.0]	vx	[0.0]	fov_h	FOV width (deg)	[0]	cfg_fov[1]
Optional	FOV vertical - degrees	Float	fov_vert	[0.0]	[0.0]	[0.0]	fov_v	FOV height (deg)	[0]	cfg_fov[2]

Point observation data

Status	*Explanation*	*Data type*	*Standard*	*Desert Fireball Net*	*UFOAnalyzer*	*FRIPON*	*RMS*	*CAMS*	*AllSkyCams*	*MetRec*
Essential	Time of point data	ISO datetime	datetime	datetime, datatype: string}		TIME	Decimal frame no.	timestamp	dt	timestamp
Essential	Azimuth, N=0, E=+90	Float	azimuth	azimuth, datatype	az	[calculate]	Azim	azim	az	[calculate]
Essential	Elevation, zenith = +90	Float	altitude	altitude, datatype:	ev	[calculate]	Elev	elev	el	[calculate]
Essential	Right ascension	Float	ra	[calculate]	ra	ALPHAWIN_J2000	RA	ra	ra	inf['alpha']
Essential	Declination	Float	dec	[calculate]	dec	DELTAWIN_J2000	Dec	dec	dec	inf['delta']
Recc.	Astronomical magnitud	Float	mag	[0.0]	mag	[0.0]	Mag	mag	[0]	inf['bright']
Optional	Location - pixel count	Float	x_image	x_image	[0.0]	XWIN_IMAGE	Col	col	x	inf['x']
Optional	Location - pixel count	Float	y_image	y_image	[0.0]	YWIN_IMAGE	Row	row	y	inf['y']