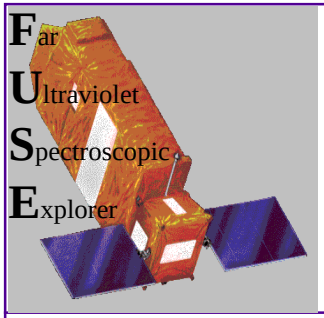


Using and Manipulating FUSE File Formats in IRAF

★ Histogram (hist)

These are formatted as image extensions and can be displayed as is.

★ Photon address mode = time-tag (ttag)

These are FITS binary tables.

- View them using `tprint`:

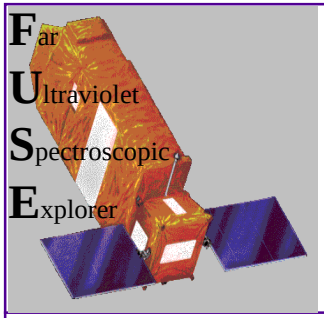
```
cl> tprint X01701010071attagfraw.fits | more
```

- Convert them to IRAF QPOE files. Display as images, select data using the selector description format, or use `xray.xtiming`:

```
cl> fuse2qp X01701010071attagfraw.fits x.qp
cl> display x.qp 1
cl> qpcopy 'x.qp[time=100:900,pha=2:9]' new.qp
```

★ Extracted, calibrated 1D spectra (cal)

- These are also FITS binary tables. Manipulate with `ttools` tasks.



Viewing FUSE Images in IRAF

- Set up a special `ximtool` window with a large image area:

```
ximtool -geometry 1024x560+244+0 -imtoolrc /home/gak/iraf/dev/imtoolrc &
```

- Make sure that your IRAF setup has its `dev$imtoolrc` and `dev$graphcap` files with a definition to handle this. Or, you can set up your own versions in your personal filespace

  imtoolrc: make an entry that looks like this:

```
66 1 16384 1024 #imt66|imtfuse
```

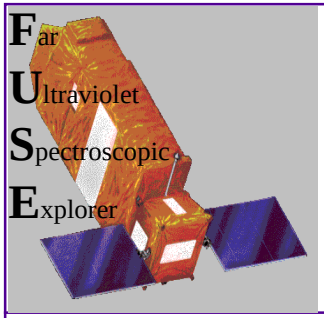
  graphcap: make an entry that looks like this:

```
imt66|imtfuse:cn#66:xr#16384:yr#1024:\
 :LC:BS@:z0#1:zr#200:dd=node!imtool,,16384,1024:tc=ism70:
```

  Then,

```
cl> set stdimage=imtfuse
cl> display X01701010071attagfraw.qp 1
```

- Make measurements on the image with standard tools such as `imexamine.`



Rudimentary Timing Analysis with Raw Images

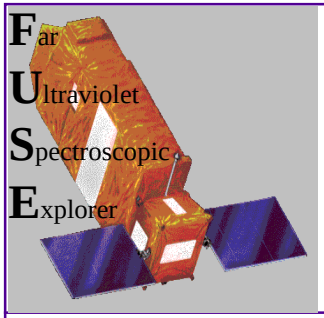
- Data converted to QPOE format with `fuse2qp` is compatible with many functions in the PROS `xray` package.

- Generate light curves using `xtiming.ltcurv` and QPOE selectors. This example traces the intensity of the Lβ airglow line:

```
xt> ltcurv 'X01701010071attagfraw.qp[x=(6650:6850),y=(540:610)]'
```

View the result using

```
xt> sgraph 'x_ltc.tab time ctrt'
```



Viewing 1D Calibrated Spectra in IRAF

★ The contents can be listed using `tprint`:

```
cl> tprint X0170101tot1alif4ttagfcal.fits | more
```

or, output specific columns to an ASCII file:

```
cl> tprint X0170101tot1alif4ttagfcal.fits
     col='WAVE FLUX ERROR' > flx.asc
```

★ Make rudimentary plots using `sgraph` (and `=gcur` for cursors):

```
cl> sgraph 'X0170101tot1alif4ttagfcal.fits WAVE FLUX ERROR'
   wl=1050 wr=1075 wb=0. wt=3e-13
cl> =gcur
```

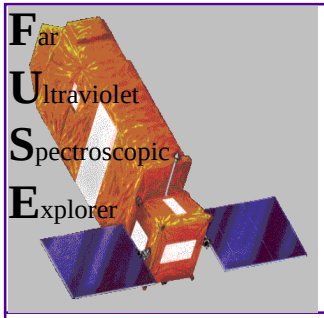
★ Convert the FITS binary table format to an image format:

- Extract just the flux column to an image file:

```
cl> tabim X0170101tot1alif4ttagfcal.fits flux.imh FLUX
```

- Get fluxes and wavelengths in `onedspec` image format for use with `splot`:

```
cl> fuse2img X0170101tot1alif4ttagfcal.fits flx.imh
cl> splot flx.imh
```



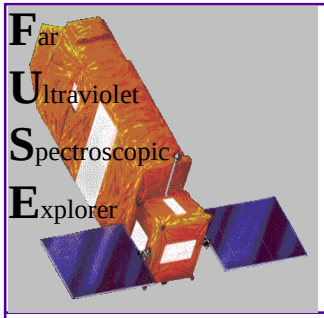
Combining FUSE 1D Spectra

- The task `fusecombine` will take a file that lists the files to be combined and an output file name. The output is in FUSE cal file format.

```
cl> fusecombine tons180.in lif2atot.fits
cl> fuse2img lif2atot.fits lif2atot.imh
cl> splot lif2atot.imh
```

- Since FUSE spectra are oversampled for many scientific applications, it is useful to bin adjacent pixels for increased signal to noise, especially on faint sources. Use the task `fusebin` to bin up a FUSE cal-file format spectrum:

```
cl> fusebin lif2atot.fits lif2atotb5.fits 5
```



Making Measurements with FUSE Spectra

- ★ Rudimentary measurements of features in FUSE spectra can be made by converting them to IRAF image format and using `splot`. *These should be viewed with caution— the wavelength scale has been manipulated to get the data into the* `onedspec` *format.*
- ★ Complex functions with a variety of continuum shapes, emission line shapes, and absorption line shapes can be fit to FUSE spectra using the task `specfit`, which can be found in the package `stsdas.contrib.spfitpkg`.
- ★ The preferred input file format for `specfit` is an ASCII file with a 2-line header followed by three columns giving wavelengths, fluxes, and errors. This file can be made from FUSE cal files using

  ```
  cl> listfuse lif2atot.fits lif2atot.spec
  ```

- ★ A detailed help file for `specfit` is available via

  `cl> help specfit [page- | lpr` for a printed copy`]`

- ★ Also see http://www.pha.jhu.edu/~gak/specfit.html