Changes in IUE data reduction

note and software prepared by
L.Chiappetti - IFCTR
Issue 1.2 - March 1992

"così per li gran savi si confessa
che la fenice muore e poi rinasce
quando al cinquecentesimo anno appressa"
(Inf. XXIV 106-108)

"rifatto sì come piante novelle
rinovellate di novella fronda,
puro e disposto a salire alle stelle"
(Purg. XXX 143-145)

Table of content

1. Introduction

This note describes the changes in IUE data reduction which are due to the phasing out of the HP 1000 based IHAP system. Most of the functionality supplied within IHAP (or related to it) will be replaced by software running on the Institute's Unix workstations, possibly within the framework of MIDAS or IRAF.

It is planned to discontinue the maintenance of the HP 1000 since January 1992. The machine will be left running. There is no maintenance for the Ramtek even now.

All software running on the IBM 3090 will remain available unchanged.

This note is intended as a **complement** to previous documentation, including:

[1] Belloni and Chiappetti, 1987, Sistema di analisi IUE. Manuale per l'uso.
[2] Chiappetti, 1989, GEX. An IHAP implementation of the Gaussian Extraction for IUE.

therefore extensive description of basic IUE concepts are not presented here, but just the changes and updates to previous procedures. The reader is referred to the documentation quoted above for more information, applying a minimum of judgment (*cum grano salis*) to evaluate what is still valid.

Document history

Issue	Date	Notes
1.0	Sep 91	Original issue : sections 1,2,3,4,5,6,7 and Appendix A
1.1	Nov 91	Reissued sections 1,2,5,6,7
1.2	Mar 92	Reissued section 2 (2.1 new plus renumbering of 2.2-3) Reissued pages 5-3 and 6-1 and added new section 6.7 New issue of Appendix B

2 Reading IUE tapes

It is now possible to access both IUE tapes as well as IHAP tapes written on the old HP without any need of using the HP any more.

2.1 Using IUEFITS to convert IUE tapes to FITS

There is now a replacement (running on Sun, and soon also under VMS) for the old HP procedure converting an IUE tape to an IHAP tape (ref. [1], sect. 3.1). The new procedure `IUEFITS` is described below, and has the following *differences* with respect to the HP original :

- ∑ `IUEFITS` produces its output as *FITS files on disk*, in the current directory, therefore there are no commands for the "output tape", tape positioning commands refer exclusively to the input tape.
  The naming convention used for output files is described below.

- ∑ `IUEFITS` produces a *named* log file (file `iuefits.log` in current directory). This can still be reset and printed internally with the `log` command, but is *not* printed by default at `stop`.

- ∑ The "nominal wavelengths" are never written to the log file, nor used.

- ∑ A full listing of tape labels can be obtained *only* by the `label` command, the `go` command produces only a short listing (also on terminal).

- ∑ It is *no longer necessary* to specify file type (e.g. 4 or 5 for 4th and 5th files) to the `go` command. The program determines the file type automatically and processes it accordingly.

- ∑ There is no `BATCH` command, however it is possible to supply a file of commands to the `IUEFITS` program using standard operating system facilities (input redirection in Unix, `.COM` files in VMS) as described in 2.1.2 and 2.1.3 respectively

Please note that the program has **not** been tested for third files and high dispersion fifth files !!!

The IUEFITS command can be invoked in fully interactive manner, or passing the following arguments on the runstring :

`IUEFITS [`*`tape`*`] [`*`prefix`*`] [`*`convention`*`]`

then all further commands (see below) are given interactively at the `IUEFITS>` prompt. Such commands can be abbreviated to at least 3 characters. Note that some commands accept a numeric argument.

`tape`is a valid tape device designation (e.g. `/dev/nrst9` on Sun, `MUA0:` on Vax, etc.)

`prefix`is used to build sequential filenames of the forms `prefix`*`nnnn`*`.mt` or `prefix`*`nnn`*`.fits` (note the different number of figures in the sequence number !) if the naming `convention` is set to, respectively, `Midas` or `Iraf`. If you want the file to be named according to the camera and image identification (so called default convention, e.g. `swp24010.fits`) enter the word `none` (or `NONE`) as `prefix`.

`convention`is one of `Midas`, `Iraf` or `Default` (can be abbreviated to one letter, upper or lower case). If you have specified `prefix=NONE` you won't be prompted for it, but the default convention is assumed. If you explicitly specify `convention=D`, any `prefix` value you gave is ignored.

After the preliminaries, a prompt `IUEFITS>` appears, and you can give one of the following commands (use at least 3 characters; the example shown is the output of the `help` command)

```
IUEFITS>help
Legal commands are (at least 3 char) :
REWind       rewind input tape
FSF [n]      forward  skip n [df=1] files
BSF [n]      backward skip n [df=1] files
FSR [n]      forward  skip n [df=1] blocks
BSR [n]      backward skip n [df=1] blocks
LISt [n]     list n [df=1] records
LABel [n]    list labels [df=all files on tape]
LOG          print and reset log file
HELp         this command
GO           process current file
STOp         terminate
```

A typical sequence to analyse a brand new observation tape will be as follows (for an archive tape you will not need to skip 1st and 2nd files as they are not present, for a known tape you can omit the label analysis, etc.):

```
IUEFITS> rew          position the tape at beginning
IUEFITS>label         produce label listing for all files on tape
IUEFITS>log           print the listing and reset log file
IUEFITS>rew           rewind tape to clear any error
IUEFITS>fsf 2         skip 1st and 2nd file of the first image
IUEFITS>go                  process  4th file (LBL)
IUEFITS>go                  process  5th file (extracted spectra) (or skip
IUEFITS>fsf 2         go on skipping 1st and 2nd and processing
...                   4th and 5th for all images on tape
IUEFITS>stop
```

2.1.1 File naming conventions

The output file of `IUEFITS` will be FITS files residing in the current directory, and will be named according to one of the following conventions (`M` for `Midas`, `I` for `Iraf`, `D` for default):

`M` files are named sequentially (in the sequence they are processed) as `prefix`*`nnnn`*`.mt` (this name is suitable for input using `INTAPE/FITS` in Midas)

`I` files are named sequentially as `prefix`*`nnn`*`.fits`

`D` files are named according to camera *`ccc`* (swp, lwp etc.), image number *`nnnnn`*, and file type as follows :

`cccnnnnn``_r.fits`	for first files (**r**aw images)
`cccnnnnn``_p.fits`	for second files (**p**hotometric corrected images)
`cccnnnnn``_q.fits`	for third files (now obsoleted)
`cccnnnnn`` .fits`	for fourth files (LBL=line-by-line spectra)
	for LO dispersion fifth files there are 4 spectra
`cccnnnnn``_g.fits`	**g**ross spectrum
`cccnnnnn``_b.fits`	**b**ackground spectrum
`cccnnnnn``_n.fits`	**n**et spectrum
`cccnnnnn``_a.fits`	**a**bnet spectrum
	for HI dispersion fifth files there are as many spectra as spectral order mm, but the user may select one of the following types for processing
`cccnnnnn``_g`*`mm`*`.fits`	**g**ross spectrum
`cccnnnnn``_i`*`mm`*`.fits`	**i**nterorder spectrum
`cccnnnnn``_n`*`mm`*`.fits`	**n**et spectrum
`cccnnnnn``_r`*`mm`*`.fits`	**r**ipple corrected spectrum

2.1.2 Using IUEFITS on Sun

In order to use IUEFITS to read an IUE tape for analysis on one of the Unix workstations, you proceed as follows :

a1) either log in on `helios`
a2) or from your favourite workstation open a window on `helios`
b) work in the `helios` window only
c) mount your tape on the Kennedy tape drive
d) cd to a temporary storage directory (e.g. `/data/`*`user`*`/fits`)
e) issue the iuefits command
f) eventually repeat c-e for other tapes
g) now you can start working on your favourite workstation
h) `ifcmidas` to start MIDAS
i1)`INTAPE/FITS` *`filelist prefix`* `/data/`*`user`*`/fits/`*`prefix`*

i2) or use @@ read (see below) to read named images

You may do i1 or i2 according to the fact you have used the Midas or the default naming conventions.

Please note the following problems when using IUEFITS on Sun:

- the tape is implicitly rewound when opened (at start of iuefits)
- this implies you have to stop and reenter iuefits whenever you are changing tapes
- the bsf and bsr commands do not work (yet; this is due to a limitation in the 3f Fortran-callable Unix routines; it might be relieved doing a rewind and an fsf automatically, which is what you can do manually)
- errors may occur if an unexpected end-of-file is encountered (again this is a problem with the 3f routines). To clear it, do a rew or fsf.

2.1.3 Using IUEFITS on Vax

The possibility to run IUEFITS on the Vax will be provided as a backup in case of failures of the Kennedy tape drive attached to helios. In this case proceed as follows :

a) on the VAX (maybe in a vax window from an Unix workstation)
b) login under your username
c) cd (or SET DEF) to a temporary storage directory (e.g. [*user*.fits])
d) physically mount your tape on the MUA0: tape drive
e) **important !** do MOUNT/FOR/NOUNLOAD/BLOCK=2048 MUA0:
f) issue the IUEFITS command
g) when you have finished dismount the tape
h) you can then move back to your favourite workstation
i) your Vax directory can be accessed e.g. as /vax/*user*/fits.

FITS data can be read directly from Sun or DECstation without any conversion. They are of no use on the Vax since there is no Midas there.

Please note the following problems when using IUEFITS on Vax:

- the tape is rewound only at MOUNT, so you can position it from the outside using VMS commands and then enter IUEFITS.
- you **must** give the last colon of the tape drive name (MUA0 does not work, you **must** say MUA0: !)
- the skipping commands are implemented in a very inefficient manner in Fortran. fsr and fsf just read and discard the records, bsr uses Fortran BACKSPACE (and is quite slow), while bsf is not implemented.

2.2 Reading existing IHAP and FITS tapes

If one has already IUE data in IHAP format, he can use the following procedure (Midas reads both IHAP and FITS files alike):

a) mount the tape on the Kennedy unit attached to `helios`
b) logon on `helios` (also remotely)
c) `cd` to a temporary storage directory (e.g. `/data/`*`user`*`/fits`)
d) use `ihapread` or `fitread` (or `dd`) to transfer the files to disk (they need to have `.mt` extension to be readable from MIDAS). Note that `fitread` and `ihapread` accepts only a list of *relative* file positions on tape (e.g. `fitread 1 4 5 7` to read the first, fourth, fifth and seventh file from *current* position, wherever the tape is) and does not rewind the tape.
e) put the tape away and logoff (if not continuing on `helios`)
f) logon on your favourite workstation
g) `ifcmidas` to start MIDAS
h) `INTAPE/FITS` *`filelist prefix`* `/data/`*`user`*`/fits/fit`

(if one is working on a Sun only he may do f instead of b, and use `INTAPE/FITS` to read directly from tape, at the moment all Sun workstations can access the tape drive on `helios` transparently from within MIDAS). The final MIDAS files will be called something like *`prefix`*.bdf, *`prefix`*.bdf etc. preserving the original FITS tape numbering.

The procedure described above is general enough to work on any workstation. If data are kept on disk in FITS format they can be read from any Unix workstation (e.g. shared between Sun and DECstation), while if they were kept in MIDAS format they could be read only under the same operating system used to write them. Using `ihapread` instead of `fitread` in step d will read directly an IHAP tape (at step h remember to use `/data/`*`user`*`/fits/iha` instead of `/data/`*`user`*`/fits/fit`).

In the case one wants to rename an entire family of files coming from a FITS (or IHAP) tape (this is useful if one is reading in another tape, as `fitread` or `ihapread` always name files `fit`*`nnnn`* or `iha`*`nnnn`*`.mt`) the following trick is useful to rename them to a new name `new`*`nnnn`* (any prefix can be used instead of *`new`*) :

```
foreach i (*)
set a = `echo $i | cut -c4-10`
set b = new$a
mv $i $b
end
```

It is however quite inconvenient to handle such files within MIDAS using names like new0001 etc. which make no reference to the object and IUE image name, but only to the arbitrary position on the original tape. The following section suggests an easy convention to handle IUE data in a more natural way.

2.3 Easy naming convention for MIDAS IUE files

It is considered desirable to have a way to sort IUE files according to the object observed in a directory per object; it also considered desirable to have a file name which is given by the IUE camera id plus the image number (e.g. `swp24010`). If this has not been done by `IUEFITS` it can easily be handled by the MIDAS procedure `camera` described below (which also takes care of inserting information necessary for the further reduction as the exposure time).

There are some troubles however in the implementation of this approach (and some way of overcoming them). First of all, a MIDAS procedure operates on MIDAS (bdf) images only, not on FITS files. Second, there is only one MIDAS working directory, the one in which one enters MIDAS (in Milano this is `/data/`*`user`*`/midas` or `/poseidon/data/`*`user`*`/midas`), which cannot be changed at a later time. Third, any change to the name of a bdf file in MIDAS does not propagate to the FITS file when it is written out (due to the awkward way MIDAS writes files to disk via a catalogue), and conversely the name of a FITS image on disk cannot be used to read the data in MIDAS (again `INTAPE` expects a `pref`*`nnnn`*`.mt` file).

Let us preliminarily assume that one is able to have files with the proper names (e.g. `swp24010.mt`) in the proper directory-by-source (e.g. `/data/user/fits/pks2155`). The two utilities `read` and `write` described below provide a way to read the files into MIDAS with the wished name (e.g. `swp24010.bdf`)

One could then think to proceed as follows :

First `read` in the image from the proper directory
Then apply the `camera` command
Then `write` it back to the proper directory, e.g.

```
@@ read pks2155 swp24010
@@ camera swp24010
@@write pks2155 swp24010
```

Unfortunately the image has not the proper name before being read in. There is need for another procedure to read in the image, invoke camera, and write out the image in FITS to the appropriate directory. This is procedure `iuesetup`. The correct sequence is therefore :

Use `iuesetup` to change name and create a FITS file in the proper place (alternatively use `INTAPE`, `camera` and `write` manually)
At any later time use `read` to read it in
When finished use `write` to write it back

IMPORTANT NOTE:all procedures listed below are a *preliminary* release and are available in directory `/home/pian/midas`, from which they shall be copied to the user `midas` directory. A different arrangement will be organized for the future.

2.3.1 the camera *procedure*

It is invoked as : `@@ camera` *`image`*

where *`image`* is the default MIDAS .bdf filename (if the file was read in with `INTAPE` this will be of the form `prefix`*`nnnn`*). The file shall exist (i.e. have been read in within MIDAS), and be a fourth file (identifier *exactly* as specified in 3.1 below).

The procedure will check for the existence of the `EXPOTIME` descriptor and prompt for an exposure time in minutes (as the value retrieved from the VILSPA log) if this is not available. This overcomes the difficulty pointed out in 3.1 below.

The procedure will retrieve the *camera* and *image* number from the `IDENT` descriptor and rename the .bdf file to *`cameraimage`* (e.g. `swp24010`). MIDAS keyword `OUTPUTC` will contain the new name. This procedure will not write out any FITS file : one can go on working within MIDAS, but has to write the file out explicitly with `write`. Alternately use `iuesetup` to produce the FITS file in the proper place.

2.3.2 the iuesetup *procedure*

It is invoked as: `@@ iuesetup` *`number prefix source`*

This procedure will look for a file called *`prefix`*`nnnn` in the user FITS data directory (**WARNING!**this is currently encoded as `/poseidon/data /pian/fits`). *number* corresponds to `nnnn`, but is in free format (i.e. to access file `pinco0020` one can specify *`number`* just as `20`). This is where the files should be after they have been read from tape according to the procedure in 2.1 or 2.2 above. *`source`* is the name of the subdirectory (typically one per target) where the renamed FITS file will be put (this is a subdirectory of the original FITS directory). These subdirectories shall be created in advance.

This procedure will read in the file in MIDAS with a temporary name, invoke the `camera` procedure on it, and write it out using a temporary catalogue to the destination (*`source`*) directory renaming it with the correct name (e.g. swp24010). The original *`prefix`*`nnnn` file and the MIDAS temporary file will be deleted.

Limitations: the procedure uses intermediate files, hence a space 3 times one file is needed. Also the commands used for moving the FITS file to the final directory are *Unix dependent* (VMS users shall change them).

2.3.3 the read *procedure*

It is invoked as : `@@ read source file`

This procedure will read in MIDAS a FITS *file* having the canonical IUE name (e.g. `swp24010`) from the user FITS subdirectory source (**WARNING!**this is currently encoded as `/poseidon/data/pian/fits/source`).

This procedure can be applied on any file (not only fourth files produced by `iuesetup`, but any processed IUE file - or even any other file - written out by `write` with their proper name).

Limitations: the procedure uses two *Unix dependent* commands for renaming the FITS files (in fact, to avoid the proliferation of temporary files, it uses *Unix soft links* which allow a file to have more than one name). VMS users shall seek a different arrangement.

2.3.4 the write *procedure*

It is invoked as : `@@ write source file`

This procedure will write out a *file* having the canonical IUE name (e.g. `swp24010`) to the user FITS subdirectory source (**WARNING!**this is currently encoded as `/poseidon/data/pian/fits/source`).

WARNING! This command is not yet available.

The standard extraction (replacement for the IHAP batches IUEZZZ, IUELOO, IUE4TH and DISK4) is currently implemented as a MIDAS procedure. At the moment only 4-th files are processed.

3The standard extraction

3.1 Prerequisites

The procedure assume the files have an identifier of the same form used by the old IHAP extraction (Ref. [1], sect. 3.5.1) The convention is that the first character of the identifier be a `4` followed by a *blank* followed by the *camera* and image number (e.g. `4 SWP22553`). As in the case of the previous procedure, only 55 or 110 line spectra are processed. The user should take care that the `IDENT` descriptor is set appropriately.

This will be the case if you use images prepared on the HP and read in as described in 2 above.

It is also assumed to have the exposure time in seconds stored in a descriptor `EXPOTIME`. Unfortunately the IHAP `WFITS` command does not preserve the information (originally present in IHAP's global `G37`). The user can repristinate it manually, or wait to be prompted by the procedure.

The exposure time is correctly inserted by the IUE tape reader program `IUEFITS` (see 2.1 above).

3.2 The procedure

The procedure is called `IUEX` and currently processes a single 4-th file stored in MIDAS format (read in as described in 2 above). It is invoked as:

`@@ /poseidon/lucio/iue/iuex` *`name`* `[D] [N]`[1]

where

`name` is the name of the image

the optional argument `D` is used if one wants to follow the procedure step by step on the display. In this case it is recommended to create two image displays (size e.g. 1024∞110) and one graphics display (size e.g. 1024∞400) in advance, as the procedure does not take care of it.

the optional argument `N` is used to inhibit deletion of temporary files, which are otherwise automatically deleted (they are listed below).

The procedure is the exact analogue of the IHAP `DISK4` batch (see [1] 3.5.3.4). **Note however that presently no filtering is performed yet and the original image is used unchanged**. A minor difference is represented by the fact that the calibration curves are kept in MIDAS tables (and not read in from ASCII tables all the times). The latest calibration curves used in IHAP (i.e. the

1 You might invoke it as `IUEX` *`filename`*`[D] [N]` in the case you have previously issued (e.g. in your `login.prg`) the MIDAS command:

CREATE/COMMAND IUEX @@ /poseidon/lucio/iue/iuex

Cassatella, Lloyd & Gonzalez 1989 for camera LWP) are used. There is some minor difference in the way the spline-interpolation of the calibration curve is done, in particular in the wavelength range outside that of the table the curve is set to zero, instead of extrapolating as IHAP did. Anyhow the difference is in general less than 2%.

The procedure produces two MIDAS files, a 4-th file calibrated in flux and an extracted 1-d spectrum. They are labelled accordingly using the same identifier of the input file followed by the word `calibrated` or `extracted`. The extracted file has the date of extraction in the `HISTORY` descriptor. The files are named appending a suffix to the original *`name`*, namely:

`name``_calib` for the calibrated 4-th file
`name``_iuex` for the extracted spectrum

In the case the extraction is repeated several times, and files with the same name exist already, a further 4-digit sequence number is appended to the file name, e.g. *`name`*`_calib_0001`.

The *`name`*`_calib` and *`name`*`_iuex` files correspond respectively to the files labelled `CALIB` and `EXTRA` on IHAP (ref. [1], 3.5.3).

All other files produced by the procedure are temporary files (accessible within MIDAS as `&A`,`&B`...`&H` : they are known on disk as `middumma.bdf` etc., see MIDAS manual). They can be deleted with `DELETE/TEMP`.

The temporary files are, in order of production :

`&A` filtered 4th file (currently copy of input file)
`&B` background in lower strip
`&C` background in upper strip
`&D` average background
`&E` smoothed background
`&F` 2-d smoothed background
`&G` interpolated calibration curve
`&H` 2-d calibration curve
`&I` calibrated 4th file, later renamed to *`name`*`_calib`
`&J` extracted spectrum, later renamed to *`name`*`_iuex`

In the case you enable the Display option, you will be shown the following (in time sequence):

In image 0:	the original file with standard cuts
In image 1:	the filtered file (&A) with the same cuts
In graphics :	the background (upper, lower, average, smoothed) upper (&C) and lower (&B) in cyan and magenta, average (&D) in black, smoothed (&E) in red : note that the upper and lower background refers to 8-16 scan lines and the average is reported to 1 line)
In graphics :	the calibration points with the interpolation (&G)
In image 1:	the calibrated 4-th file (`_calib`) with cuts 1-10
In graphics :	the extracted spectrum (`_iuex`)

4 The Gaussian extraction

4.1 Prerequisites

They are the same reported in section 3.1 above

4.2 The procedure

The procedure is called GEX and currently processes a single 4-th file stored in MIDAS format (read in as described in 2 above). It is invoked as:

```
@@ /poseidon/lucio/iue/gex name step1 step2 [D] [N]
```
[1]

where

name is the name of the image and *step1, step2* are the binning factors for the two GEX passes (see [2])

the optional argument D is used if one wants to follow the procedure step by step on the display and to enable verbose mode.

the optional argument N is used to inhibit deletion of temporary files, which are otherwise automatically deleted (they are listed below).

The procedure is the exact analogue of the IHAP GEX batch (see [2] and reference therein for the details of the algorithm.).. A minor difference is represented by the fact that the calibration curves are kept in MIDAS tables and in the way the spline-interpolation of the calibration curve is done (for this refer to 3.2 above since the same calibration code is used). Also the binning procedure in MIDAS is aligned to the original bins, while in IHAP it is offset by half new bin.

The procedure takes (for a 55-line spectrum on a DECStation 5000) 0.2 s for the first pass and about 20 s for the second pass, plus the time needed to MIDAS for binning and calibration.

The procedure produces two MIDAS files, the raw result of GEX 2nd pass (which is a 1-d file scaled 0 to 1) and an extracted 1-d spectrum. They are labelled accordingly using the same identifier of the input file followed by the wording `GEX 2nd pass output 0-1` or `Gauss s1:`*`step1`* `s2:`*`step2`*. The extracted file has the date of extraction in the HISTORY descriptor. The files are named appending a suffix to the original *name*, namely:

---

1 You might invoke it as `GEX` *filename*`[D] [N]` in the case you have previously issued (e.g. in your `login.prg`) the MIDAS command:

```
CREATE/COMMAND GEX @@ /poseidon/lucio/iue/gex
```

*name*_gex for the raw result of GEX 2nd pass
*name*_gauss for the extracted spectrum

In the case the extraction is repeated several times, and files with the same name exist already, a further 4-digit sequence number is appended to the file name, e.g. *name*_gauss_0001.

The *name*_gex and *name*_gauss files correspond respectively to the files labelled GEX and GAUSS on IHAP (ref. [2], pag. 4)

All other files produced by the procedure are temporary files (accessible within MIDAS as &A,&B...&H : they are known on disk as middumma.bdf etc., see MIDAS manual). They can be deleted with DELETE/TEMP.

The temporary files are, in order of production :

&A the input file scaled 0 to 1
&B &A rebinned according to coarse *step1*
&C &A rebinned according to fine *step2* (only if not equal 1)
&D result of GEX pass 2, later renamed to *name*_gex
&E &D rescaled back to flux numbers
&F integration of &E perpendicular to dispersion
&G interpolated calibration curve
&I extracted spectrum, later renamed to *name*_gauss

In the case you enable the Display option, **you will so far not be shown any plot** (it is planned to add such option later). This option only enables verbose mode in pass 2 (in this case for each wavelength been processed, the resulting flux is shown in a one-line status area on the screen). Pass 1 is always semi-verbose, showing a status line with the current wavelength, and messages about wavelength bins ignored because of insufficient signal, as well as the result of the final parabolic fit to the Gaussian width.

5 Hints for reduction and analysis

This section will be filled with a brief reminder of the most useful MIDAS and IRAF commands.

51 Tape analysis commands

The only equivalent to the commands described in [1], sect. 3.2 is the IRAF command `mtexamine`. Refer to [1], sect. 3.3 for the typical format of IUE, IHAP and FITS tapes.

Bewarethat the new FITS standard allows blocking factors and data formats (IEEE floating points) incompatible with IHAP. Check for the default formats used by MIDAS and IRAF (they are different) when planning to export data, and eventually change them as wished (it should always be possible to revert explicitly to the old format if needed).

5.1 Reduction commands

This section shall describe replacements for commands listed in [1], sect 3.6, 3.7 etc. (to be written)

5.2.1 rebinning

5.2.2 sum, rescaling and algebra on images/spectra

See help for `COMPUTE/IMA`

5.2.3 joining images/spectra

5.2.4 smoothing

5.2.5 rigid shift and change of coordinates

5.2.6 extraction of a region, truncation, extension of an image

5.2.7 maximum and minimum

See help for `STATISTICS/IMA` and `FIND/MINMAX`

MIDAS files (images and tables) can be copied, renamed and deleted with the variants of the `COPY`, `RENAME` and `DELETE` commands (see the MIDAS help for use of qualifiers).

External files can be manipulated from within MIDAS using the `$` command followed by a system command. Unix users shall note that this gives access only to `sh` commands. To access `csh` commands (including the *Uniq interface*) one can define a `SYS` command as follows (typically do this in `login.prg`) :

```
CREATE/COMMAND SYS $ csh -c
```

and then issue any csh command prefixing it with `SYS`.

There is no obvious facility to handle a directory of images (as done in IHAP with DLIST) unless perhaps one uses MIDAS catalogues. One could note the following commands:

`SHOW/DESC` *`file`*	to list all descriptors of a file (verbose)
`READ/DESC` *`file descriptor`*	to look at the value of one descriptor
`READ/DESC` *`file`* `IDENT`	to look at the image identifier

This section shall describe replacements for commands listed in [1], sect [5.1 and 5.2 etc. (to be written : see also graphics in section 7 below)

5.5.1 computation of fluxes

The following MIDAS procedure (**WARNING!**this temporarily resides in /home/pian/midas) can be useful to compute the flux in an extracted spectrum (it assumes the usual IUE convention that units for data are 10^{-14} erg/cm^2/s/Å) :

```
@@ flux file start end
```

This will use `STATISTICS/IMA` to compute the mean flux (in erg/cm^2/s/Å) of *`file`* between wavelength *`start`* and *`end`* (in Å). It will also compute the wide band flux (in erg/cm^2/s) multiplying by the band width.

The error shown is the *standard deviation* computed by MIDAS, which has been verified to be

$$R(ISU(1,n,F((x_i - XTO(x))^2,n-1)))$$

Former IHAP users shall note that, while this standard deviation is numerically the same as the RMS returned by IHAP `SAMP` command, there may be an *inconsistency in usage* if one was accustomed to divide the IHAP RMS by the number of points `n` to obtain the error (this was done systematically by several of our procedures). The current convention uses the raw standard deviation without dividing by n.

The following trick is useful to record the results of repeated invocations of flux into MIDAS

logfile and to print it :

`log/on` to enable logging
issue all relevant commands
`log/off` to disable logging
with a system command ($) rename or print file `$MID_WORK/ FORGR00.LOG` (capital letters !!) then delete it (otherwise new stuff will be appended to the previous log file)
`log/on` to re-enable logging

5.5.2 generation of wide-band spectra (SIGMA files)

Since the `makesigma` MIDAS procedure is mainly used as interface to the fitting programs, it is described in section 6 below.

At the moment the spectral fitting programs run unchanged on the IBM (see [1], sect. 4.3, and 6.7 below). Here are given only hints on the way to generate input files for them and to import results back to MIDAS. It might be possible that on a later date the fitting programs are ported to Unix.

One should note that the transfer of files between the workstation and the IBM is greatly enhanced by the use of ftp, and that it is possible to keep a window open on the IBM while on the workstation. For more details see *The Link Guide* (Chiappetti, 1992, Issue 2.0), the red book in the terminal room.

WARNING!All procedures listed below are not publicly released and reside temporarily in /home/pian/midas from where shall be copied. It is planned to have a different arrangement in the future.

6 Spectral fitting

6.1 Generation of SIGMA files from MIDAS spectra

A simple MIDAS procedure replaces the awkward IHAP procedure described in [1] 6.5. This procedure is invoked as :

```
@@ makesigma file (start end step) mode
```

where `file.bdf` shall be an extracted IUE spectrum. `mode` can be `AUTO` or `NOAUTO` (default) The remaining parameters are wavelengths in Å and are optional. The default is to use as `start` and `end` the start and end of the file, and as `step` a value of 50 Å. `mode`=`AUTO` disables interactive prompting.

As default operation the procedure will produce a file with fluxes and errors computed in bands `step` Å wide, covering with continuity the interval from `start` to `end`. This is however easily changed (typically to reject regions with noise or lines, *previously* identified e.g. in a graphical way) interactively : in fact the procedure (in `NOAUTO` mode) will prompt the user for the start and end wavelength of each bin as follows :

The start wavelength of the *first* bin will be suggested to be `start`. The user can however change it (and generally *must* change it : it is obvious that there is no sense in analysing a SWP spectrum from less than 1230 Å, or a LW spectrum from less than 2100 or 2400 Å, but the raw files will start at 1000 or 1700 Å respectively).

The start wavelength of *each next* bin will be the end of the previous bin. The user shall change it to introduce gaps (reject regions).

The end wavelength of *each* bin will be `step` Å more than its start. The user shall normally not change it, unless the bin is affected by a line (in which case one would make it *smaller*) or the next bin is affected by a line (in which case one would *extend* the current bin to include the line-free portion of the next one)

The user is prompted for bins a number of times determined in advance. This is 5 more than the number of bins `step` Å wide necessary to cover the range from `start` to `end`, to take into account the possibility of shorter bins specified by the user. The user can however terminate the input at any time by:

specifying a negative start wavelength
specifying a start wavelength shorter than the end of the previous bin
specifying an end wavelength beyond the file end

The following tracing of a typical run illustrates the way `makesigma` operates.
Boldface indicates user input. <CR> indicates a plain carriage return.

```
Midas 003> @@ makesigma swp24010_iuex
Table prova will be created with max 0024 fluxpoints
Enter a NEGATIVE or out-of-range value to terminate
Point 0001 From lambda (df 1.00000E+03) 1250                X(1)
Point 0001 To   lambda (df 1.30000E+03) <CR>
Mean flux is 3.66367E-01 +/- 2.38836E-01 *E-14 erg/cm2/s/A
Point 0002 From lambda (df 1.30000E+03) <CR>
Point 0002 To   lambda (df 1.35000E+03) <CR>
Mean flux is 1.99881E-01 +/- 1.90026E-01 *E-14 erg/cm2/s/A
Point 0003 From lambda (df 1.35000E+03) 1500                X(2)
Point 0003 To   lambda (df 1.55000E+03) 1525
Mean flux is 3.67966E-01 +/- 2.46752E-01 *E-14 erg/cm2/s/A
Point 0004 From lambda (df 1.52500E+03) 1575                X(1)
Point 0004 To   lambda (df 1.62500E+03) <CR>
Mean flux is 2.12177E-01 +/- 2.53506E-01 *E-14 erg/cm2/s/A
Point 0005 From lambda (df 1.62500E+03) -1
Only 0004 points entered
Midas 004> read/tab swp24010_iuex
 Sequence LAMBDA       FLUX         ERROR        DELTA        NPTS
 -------- ------------ ------------ ------------ ------------ ------------
        1  1.27500e+03  3.66367e-01  2.38836e-01  2.50000e+01  4.30000e+01
        2  1.32500e+03  1.99881e-01  1.90026e-01  2.50000e+01  4.40000e+01
        3  1.51250e+03  3.67966e-01  2.46752e-01  1.25000e+01  2.20000e+01
        4  1.60000e+03  2.12177e-01  2.53506e-01  2.50000e+01  4.40000e+01
 -------- ------------ ------------ ------------ ------------ ------------
Midas 005>
```

X(1) : change the start wavelength of the bin
X(2) : change both the start and end wavelength of the bin

The procedure computes the flux *and error* as described in 5.5.1 above.

The procedure will produce two files (both in the current MIDAS data directory, the same where the input *file*.bdf resides) :

file.sigmais a SIGMA file in the old format (the choice of the old format is to make easier plotting it with programs like SMONGO : this choice means the file cannot be plotted with HP PABLO), that is:

5 header lines (inclusive of the image identification)
n records with three columns (wavelength, flux, error respectively)

file.tblis a MIDAS table. The only use of this is to plot the data within MIDAS using the PLOT/TAB or OVER/TAB commands. The table has 5 columns (the *first three* are the same as in the SIGMA file), that is:

#1 (LAMBDA) :	the central wavelength of each bin
#2 (FLUX):	the flux
#3 (ERROR):	the associated error
#4 (DELTA):	the half width of the bin in Å
#5 (NPTS):	the number of data points used in the bin

the purpose of the additional information is to allow, *should it prove necessary*, generation of a SIGMA file in the new format (including DELTA), or recomputing the error according to the old convention (dividing by NPTS) just using standard MIDAS table manipulation.

Limitations:the command used to transfer the table data outside MIDAS is PRINT/TABLE. This implies resetting the current ASSIGN/PRINT. Since the way the assignment is coded is not clearly documented, it is impossible to restore it. Therefore a default ASSIGN/PRINT is made at the end : in the case this *does not repristinate* the previous assignment a *warning* is issued and the user shall take care of it manually.

A further limitationis that the way used to transform the output of PRINT/TAB into the wished format (particularly appending the header) is strongly *dependent on Unix* commands.

6.2 Generation of FLUX files from MIDAS spectra

A simple MIDAS procedure replaces the IHAP procedure described in [1] 6.4. This procedure is invoked as :

```
@@ makeflux file
```

where *file*.bdf shall be an extracted IUE spectrum.

The procedure will produce *file*.flux (in the current MIDAS data directory, the same where the input *file*.bdf resides), in the standard FLUX format, that is :

5 header lines (inclusive of the image identification)
n records with two columns (wavelength, flux respectively)

This procedure is essentially a wrapper around the COPY/IT command to convert a MIDAS .bdf image into an intermediate MIDAS table. The table is then written out to ASCII using PRINT/TAB in a way similar to makesigma. Any intermediate files are deleted.

Limitations:the same described in 6.1 above.

6.3 Conversion of FLUX files into MIDAS spectra

A simple MIDAS procedure replaces the IHAP procedure described in [1] 6.3. This procedure is invoked as :

```
@@ fromflux file reffile (KEEP)
```

where *file*.flux shall be a standard FLUX file, and *reffile*.dbf an existing spectrum in MIDAS .bdf format used as reference to determine the wavelength range and the step in wavelength necessary to create *file*.bdf.

The procedure works reading in the ASCII FLUX file into an intermediate table (using CREATE/TABLE) and converting this into an image (with CONVERT/TABLE). This second step requires the reference image. Normally one should use as a reference image a spectrum from which the parent of *file*.flux was generated (therefore with the same step), however the MIDAS command is capable of interpolating the FLUX file on any other binning (but cannot extrapolate outside the wavelength range of the FLUX file and will set to zero the flux there).

There is generally no sense in keeping and using the intermediate table (which can be quite large), but if wished this is accomplished specifying the optional keyword KEEP. In this latter case the table will be named *file*.tbl.

Limitations: there are two *Unix dependent* commands used to chop off the header of the FLUX file and dispose of a temporary file.

6.4 Conversion of SIGMA files into MIDAS tables

This will be needed very seldom, but can be done straight away with the MIDAS CREATE/TABLE command. It is recommended to chop off the 5 header lines from the SIGMA file before.

6.5 Recommended procedure

This is the recommended procedure to run a fitting session using MIDAS to produce the SIGMA files and to plot the results, and the IBM for fitting.

a extract the IUE spectra according to the procedures in sections 3 or 4, the file selected for processing be e.g. good.bdf

b plot good.bdf and note down the regions to be rejected

c use makesigma (see 6.1) to produce good.sigma

d use makeflux (see 6.2) to produce good.flux

e transfer good.sigma and good.flux to the IBM (see 6.6) : note they will be named GOOD SIGMA and GOOD FLUX

f on the IBM do not forget to use SETMODE (see [1] 2.1) and make sure that GOOD SIGMA and GOOD FLUX have RECFM F (see 6.6)

g run the fitting program UVFIT (see [1] 4.3) on GOOD SIGMA and print the final results in FIT PRINT; eventually rename such file if you want to keep it

h in order to have a visual impression of the fit (optional), you might (it could be not necessary to save the intermediate files):

h1 transfer FIT FLUX from IBM to the workstation with MIDAS

h2 use fromflux (see 6.3) with option KEEP to create a table fit.tbl

h3 use OVER/TAB to overplot both the input and the output of the fitting procedure, that is

`good.tbl` and `fit.tbl` over `good.bdf`

i on the IBM run `UVGRID` or multiple runs of `UVFIT` to find the confidence range of the parameters of the best fit, and/or to refine the best fit

j once you have determined the final best fit parameters, run again `UVFIT` on `GOOD FLUX` in dummy mode, that is do not fit the initial guesses (reply `NO` to all questions)

k throw away `FIT PRINT` but rename `FIT FLUX` to some name reminding you of the original (e.g. to `GOODFIT FLUX`)

l transfer `GOODFIT FLUX` from IBM to the workstation with MIDAS

m use `fromflux` in default mode to create `goodfit.bdf`

n use `OVER/IMA` to overplot the fitted curve from `goodfit.bdf` over the original spectrum `good.bdf` (or equivalent, e.g. it might be convenient to use the result of a Gaussian extraction for generating the SIGMA files, but the result of the default extraction, without gaps, for plotting)

So far there are no facilities to plot the results of `UVGRID` within MIDAS.

6.6 Utilities for file transfer to and from the IBM

It is suggested to have an IBM window open on the same workstation running MIDAS and to `get` the SIGMA and FLUX file via ftp invoking the ftp session from the IBM (this is easier than making a `put` from Unix which requires the IBM disk passwords). Similarly the results of the fit can be `put` to the workstation opoening the ftp session from the IBM.

See `HELP FTP` and `HELP FTP MENU` for details (e.g. for the use of `mget` to get multiple files). In general you shall open an interactive ftp session as listed below (this is typically done if you want to get more files in a session) or issue a special command to invoke one of the automatic unsupported procedures also listed below.

```
ftp poseidon                                                    X(1)
VM TCP/IP FTP R1.2.1
Connecting to POSEIDON 192.65.131.49, port 21
220 poseidon FTP server (Version 4.1 Sun Mar 25 22:59:11 EST 1990) ready.
USER (identify yourself to the host):
lucio                                                           X(2)
>>>USER lucio
331 Password required for lucio.
Password:
typeyourpasswordhere                                            X(2)
>>>PASS ********
230 User lucio logged in.
Command:
cd /poseidon/data/pian/midas                                    X(3)
>>>CWD /poseidon/data/pian/midas
250 CWD command successful.
Command:
get swp36820_iuex.flux                                          X(4)
>>>PORT 192,65,131,1,112,53
200 PORT command successful.
>>>RETR swp36820_iuex.flux
150 Opening data connection for swp36820_iuex.flux (192.65.131.1,28725) (25962
bytes).
226 Transfer complete.
26802 bytes transferred. Transfer rate 5.46 Kbytes/sec.
Command:
quit
>>>QUIT
221 Goodbye.
Ready;
reform swp36820 flux a f                                        X(5)
COPY SWP36820 FLUX A = = = ( RECFM F REPLACE
Ready;

X(1): select your workstation (helios, poseidon, etc.)
X(2): type your username and password
X(3): cd to the MIDAS data directory (/data/user/midas typically)
X(4): get the file : beware the name might be truncated to 8 characters
X(5): important : change the file format  manually !!!!
```

Note that ftp will truncate any filename to 8 character. In the example above the IBM file will be called `SWP36820 FLUX`. If a file with the same name exists it will not be replaced by default. You shall (in an interactive ftp session) either issue a `get` *`file.type`* (`REPLACE` command or specify a new name on the IBM issuing `get` *`file.type newfile.typ`*.

IMPORTANT!ftp transfers files with `RECFM V`, while `UVFIT` wants them with `RECFM F`. To change the record format use (if command available) :

```
REFORM filename filetype * F              or otherwise do
COPY filename filetype * = = = (RECFM F REPLACE
```

where *`filetype`* is either `FLUX` or `SIGMA`. Do not forget to change record format or the fitting program will fail.

There will be a set of unsupported utilities to automatically arrange for copies. These could be

invoked as :

FROMSUN	name type	or	TOSUN name type
FROMDEC	name type	or	TODEC name type
FROMVAX	name type	or	TOVAX name type

They will copy file `name.type` from respectively `helios`, `poseidon` or `IFCTR` to the IBM as `name type`, or viceversa copy `name type` from IBM to the specified workstation as `name.type`. The arrangement will be such that one will be prompted for the username, password and directory on the remote machine only once at the beginning. One still have to make a `REFORM`.

```
fromdec swp36820_iuex flux
Enter username
lucio
Enter password
youwontexpectIlltypemypasswordheredontyou
Enter remote directory
/poseidon/data/pian/midas
VM TCP/IP FTP R1.2.1
OPEN (name of foreign host):
Connecting to poseidon 192.65.131.49, port 21
220 poseidon FTP server (Version 4.1 Sun Mar 25 22:59:11 EST 1990) ready.
USER (identify yourself to the host):
>>>USER lucio
331 Password required for lucio.
>>>PASS ********
230 User lucio logged in.
Command:
Usage of SITE command with PUT is OFF
Command:
>>>CWD /poseidon/data/pian/midas
250 CWD command successful.
Command:
200 PORT command successful.
>>>RETR swp36820_iuex.flux
150 Opening data connection for swp36820_iuex.flux (192.65.131.1,29996) (25962
bytes).
226 Transfer complete.
Ready; T=0.10/0.18 09:28:52
reform swp36820 flux a f
COPY SWP36820 FLUX A = = = ( RECFM F REPLACE
Ready; T=0.03/0.05 09:29:24
```

6.7 Unidimensional chi-square grids

A further (IBM) program has been added to `UVFIT` and `UVGRID` to allow to search for confidence intervals on a single interesting parameter in an easier and automatized way. This program is named `UVGRID1D` and is used exactly with the same calling sequence as `UVGRID` (see [1], 4.3.2). The only differences are :

Only oneparameter (typically this is spectral index) can be stepped. The other parameters can be fitted (typically the normalization) or fixed (typically A_V). There are 100 steps in the 1-d grid (so the range is from the *value* given for the interesting parameter to *value*+99**step*

The program performs an auto search of the best fit (minimum of the chi-square profile vs interesting parameter), which is often more accurate than the one performed by `UVFIT` (which tends to get stuck in local minima) and of the 90% confidence limit (using a $\Delta\chi^2$ of 2.71 appropriate for one interesting parameter). The results are written to the terminal and to the output file `GRID PRINT`. *Note that if the confidence interval is larger than the stepped interval, suitable warnings are given.*

There is no 2-d chi-square grid image file. Instead one has a file `GRID1D TABLE` (2 header lines, 2 columns), which can be used to plot the chi-square (column 2) versus the single interesting parameter (column 1).

An additional file `GRID1D RESULT` contains a single line with the image id, the best fit value and the 90% interval extrema for the interesting parameter. Its purpose is to be appended (manually) to a "database" summary file, in case of fitting of multiple images.

One can still read the values of the uninteresting fitted parameters in the file `GRID PRINT`.

Programmer's note: in order to implement this program quickly with minimum disruptions the source file contains dedicated variants of routines `READP` and `SETP` which replace the official library version

It is presently not planned to support officially any graphics program (although it might be possible that some of the old HP programs are converted). Users are referred to the plotting facilities available within MIDAS and IRAF, to the other plotting utilities available (e.g. Supermongo - it shall be noted that this is considered *unsupported*), or to their PC plotting facilities.

In particular for what concerns MIDAS it is recommended to create one or more image displays and one graphic display. The following is the arrangement assumed e.g. by the IUEX batch for displaying its results : two image windows of appropriate size (of which display 0 is used as default) and one graphic window.

```
create/display 0 1024,110,0,1000
create/display 1 1024,110,0,800
create/graphic 0 1024,400,0,600
load/lut rainbow
assign/display d,0
```

7 Graphics

7.1 Vector graphics

7.1.1 Graphics control

Graphics command to look at spectra include PLOT/IMA and OVER/IMA (and the corresponding PLOT/TAB and OVER/TAB). The layout of the plot can be controlled by the SET/PLOT command : particularly useful are the following parameters :

XAXIS(set to AUTO or use it to control the wavelength range)
YAXIS(set to AUTO or use it to control the cuts in flux for plot)
BIN=ON(to have histogram-like, not spiky plots)
PMODE(use 2 as default, use 1 for hardcopies without long legenda)
COLOUR(useful for display on screen, 1=black 2=red 3=green etc.)
LTYPE(use 1=solid 2=dotted instead of colour for hardcopies)

7.1.2 Graphics hardcopy

To produce an hardcopy see section 6.1.6 of volume A of the MIDAS Users Guide: SEND/PLOT looks preferred to ASSIGN/PLOT to dispose of the plot. It appears that, selecting LASERA as hardcopy device, a file pscrplot.0 is produced in the MIDAS data directory : however the PostScript previewer has some difficulty with it.

7.1.3 Interactive graphics

Interactive graphics can be used to read coordinates out of a plot. The standard MIDAS command to do this is GET/GCURS. However it has a problem, that is it accepts only filenames not longer than 12 characters (limit which is generally violated by the naming convention used by the IUEX and Gaussian extraction) A replacement procedure (**WARNING**! temporarily this resides in /home/pian/midas) is provided and can be invoked as :

```
@@ gcur
```

This procedure retrieves the full name of the image currently plotted from keyword MID$PLOT and temporarily renames it to the first 12 characters of the name (if longer) while invoking GET/GCURS.

7.2 Image display

7.2.1 Display control

The typical command to look at an image is LOAD/IMA. Useful manipulation of the appearance of the image is obtained controlling the CUTS or via the lookup table (LOAD/LUT, DISP/LUT and MODI/LUT; also CLEAR/LUT). An useful command to be associated with a

shortcut synonym is CLEAR/CHANNEL OVERLAY.

7.2.2 Display hardcopy

There are two commands which can be used to produce a gray scale hardcopy of an image (PLOT/GRAY and PLOT/PERSPECTIVE are not considered as they work only on a limited image size) : LOAD/IMA having done an ASSIGN/DISP to LASERA, and COPY/DISP from the current image display. It is however unclear and undocumented how to easily control the appearance of the hardcopy (e.g. black-on-white, white-on-black, position on the page etc.).

7.2.3 Interactive display

Interactive display can be used to read coordinates out of an image. The standard MIDAS command to do this is GET/CURS. However it has a problem, that is it accepts only filenames not longer than 12 characters (this limit is however not considered seriously, since normally one will use it on raw fourth files, whose names are 8 character long).

The best results are obtained using GET/CURS in a zoom window. Unfortunately the syntax is complex and also it is not possible to resize the zoom window once created. The following command string is a reasonable approach, and could be aliased to a shortcut synonym or a private procedure :

```
get/cursor ? ? NN ? W,8 512,220,0,400
```

Appendix A.
Instructions for installation of MIDAS procedures

A.1 Introduction

The MIDAS procedures IUEX and GEX (described in 3.2 and 4.2) are potentially available for distribution to external sites. They are distributed "as is" with no guarantee express or implied of fitness to any particular purpose. It is requested that the author of the procedure be duly acknowledged in each paper using results produced by these extraction codes, with a sentence like:

> The routines for the IUE spectral extraction are a local implementation within ESO MIDAS written by L.Chiappetti.

We give here instruction for installation of the procedures at a generic site running MIDAS (portable MIDAS version 91MAY or later), together with some programming information.

A.2 Material to be retrieved

The material to be retrieved for an installation is given in the table in the following page (for each the complete path name on the `poseidon` DECstation is given; see below for information about network retrieval).

A.3 Instructions for retrieval

If you have an Internet connection, you can retrieve all the above via `ftp` to `poseidon.mi.cnr.it` (192.65.131.49), with username `none` password `none`. Please do not retrieve the system dependent (optional) parts if you are not on a compatible system. The mandatory parts shall be retrieved in `ascii` mode, while the optional parts are to be retrieved in `binary` mode.

If you have only Decnet connections, you should be able to retrieve the material using IFCTR (= 39610) as gateway as in the following example :

```
COPY IFCTR::SUN"none none"::"/poseidon/lucio/iue/midas/gex.prg" *.*
```

There is of course no sense in retrieving the optional material on a VMS system.

Mandatory material	**File name**
ASCII source for calibration files	/poseidon/lucio/iue/calibration/twsp.ascii /poseidon/lucio/iue/calibration/tlwp.ascii /poseidon/lucio/iue/calibration/tlwr.ascii
Main IUEX procedure	/poseidon/lucio/iue/midas/iuex.prg
Main GEX procedure	/poseidon/lucio/iue/midas/gex.prg
Auxiliary procedures (for both)	/poseidon/lucio/iue/midas/iueaux_smo.prg /poseidon/lucio/iue/midas/iuerename.prg
Fortran sources for GEX	/poseidon/lucio/iue/source/gex1.for /poseidon/lucio/iue/source/gex2.for /poseidon/lucio/iue/source/gexco.for
Optional compilation csh script	/poseidon/lucio/iue/source/mcomp
Optional material	
MIDAS calibration tables (Ultrix)[1]	/poseidon/lucio/iue/midas/tswp.tbl /poseidon/lucio/iue/midas/tlwp.tbl /poseidon/lucio/iue/midas/tlwr.tbl
MIDAS calibration tables (Sun)[2]	/poseidon/lucio/iue/midas/sun/tswp.tbl /poseidon/lucio/iue/midas/sun/tlwp.tbl /poseidon/lucio/iue/midas/sun/tlwr.tbl
Executables for GEX (DEC Ultrix)[3]	/poseidon/lucio/iue/midas/gex1.exe /poseidon/lucio/iue/midas/gex2.exe
Executables for GEX (Sun)[4]	/poseidon/lucio/iue/midas/sun/gex1.exe /poseidon/lucio/iue/midas/sun/gex2.exe

A.4 Common procedure for installation

A.4.1 Main and auxiliary procedures

a) Select a directory where you want to install everything (this is here `/poseidon/lucio/iue/midas`) and go there

b) copy there all *.`prg` files (`iuex.prg`, `gex.prg` and the auxiliary files)

1 to be retrieved only from another Ultrix system

2 to be retrieved only from another Sun system

3 to be retrieved only from another DECstation running Ultrix (RISC)

4 to be retrieved only from another Sun system with `sun4` architecture

c) edit the `iuex.prg` file, read the header, then go the `define/local` statement defining the `disk` variable and replace it with the name of your directory (Unix or VMS syntax). Note you may have to change the length of the `disk` variable too if it is longer than the current value. You should also disable/delete the following statements which reset `disk` if an environment variable `DECSTATION` is not set to 1 (this is used locally for multiple architecture support on Decstation and Sun).

d) edit the `gex.prg` file and apply the same changes done to `iuex.`

e) do not forget to do a `CREATE/COMMAND` pointing to the new correct location of iuex and gex if you want to call them by name.

The calibration tables are supplied as ASCII files, since MIDAS table files are not portable among different systems. You should reinstall them by reading the ASCII file into a MIDAS table within MIDAS. The ASCII tables may reside anywhere, and can be deleted after the conversion (even if it is suggested to store them in some safe place out of the way). The MIDAS tables shall reside in the same directory where the .prg files are.

a) go into MIDAS
b) `create/tab` *`name`* `2` *`n`* `/poseidon/lucio/calibration/`*`name`*`.ascii`
c) assign names to columns (the units and formats are optional)
```
name/col name #1 :LAMBDA "Angstrom" G12.4
name/col name #2 :CALIB "S-1 FN ?" G12.4
```
d) make sure to repeat b-c for all three tables (*`name`* is `tswp`,`tlwp`, `tlwr` and *`n`* is `34`,`61`,`31` respectively)

A.5 IUEX specific notes

IUEX is a pure MIDAS procedure. Therefore it shall be available immediately if the above instructions have been followed correctly.

A.6 GEX specific notes and installation

GEX, similarly to his IHAP predecessor, calls two external Fortran programs GEX1 and GEX2. Unless you have a compatible DECstation or Sun system, you shall not install the corresponding `.exe` files but recompile them from the sources.

Both programs make reference to a common include file `gexco.for`, which shall be present on the same disk where `gex1.for` and `gex2.for` are.

You may copy the program sources to any directory you wish (possibily different from the one where the `.prg` files reside) and remove them once you have generated the `.exe` files and moved them to the same directory where you have the `.prg` and `.tbl` files.

Note that the source files shall have a `.for` extension even on an Unix system. The corresponding `.f` files will be created by the `esoext` preprocessor. The source files are compliant to the rules in the *MIDAS Environment* document (ESO 1991) with the exception that terminal output is handled via Fortran `WRITE`s.

To generate the `.exe` files on an Unix system and move them to the appropriate place, you may use the provided `mcomp` script.

```
mcomp gex1 destination
mcomp gex2 destination
```

where *`destination`* is the final directory where the `.exe` files are to be stored. Note that this csh script explicitly specifies to `esoext` to look for include files also in the current directory.

On VMS systems you compile and link as described on pag. 5-1 of the *MIDAS Environment* document, then manually move the files to their destination. Note that this has not been tested. If compilation fails, look for the `INCLUDE` statements referring to `:GEXCO.FOR` (which are in the unusual MIDAS syntax) and try removing the initial colon. Such statements are located in the main program and in `MID_GEX` routine. You might also have to comment out the calls and the code of the `CPUCLK` subroutine (or replace with one written by you).

Some notes for programmers. GEX1 and GEX2 retrieve information from the calling `gex.prg` via MIDAS keywords. They read and write MIDAS images using the STI interface routines. For the rest (apart the syntax of the INCLUDE statement) they are virtually unchanged w.r.t. the original HP sources. All debug statements are commented out (`esoext` does not like them)

Of course the main computation has been moved into a subroutine, since this is required by the STI pseudomapping mechanism.

Due to an unexplained deficiency in the STI routines on the DECstation (it appears not possible to have two images open at same time), the GEX2 works on a copy of the input image, opened in read/write mode (`F_IO_MODE`).

It was also necessary to protect the exponential for underflows, as the handling of floating point exceptions on DECstations is less robust than on HP.

At the moment GEX1 and GEX2 still communicate via a Fortran- written direct access file called `gexfin` (in current MIDAS data directory). It is planned to replace this with a MIDAS table and a set of keyword (for the parabolic fit coefficients). At the moment all debug plotting is disabled, but is planned to insert some plotting (either directly using plotlib or in the calling .prg files using intermediate tables).

Appendix B.
Instructions for installation of IUEFITS

B.1 Introduction

The program IUEFITS (described in 2.1) is potentially available for distribution to external sites. It is distributed "as is" with no guarantee express or implied of fitness to any particular purpose. It is desirable that the author of the program be duly acknowledged in each paper using results produced by this program, with a sentence like:

> The routines for reading IUE tapes into FITS are a local implementation written by L.Chiappetti. The use of W.Pence's FITSIO library is gratefully acknowledged.

We give here instruction for installation of the program at a generic site, together with some programming information.

B.2 Material to be retrieved

The material to be retrieved for an installation is given in the table in the following page (for Unix files the complete path name on the `poseidon` DECstation is given, for VMS the path on disk `IFCTR::DUA0:` is given; see below for information about network retrieval).

Mandatory material	**File name**
Sun executable	/poseidon/lucio/iue/midas/sun/iuefits
VMS executable	[LUCIO.IUE]IUEFITS.EXE
Optional material for relink	
Main program sources (just one copy needed, they are *identical* for Unix and VMS)	/poseidon/lucio/iue/source/iuefits.f /poseidon/lucio/iue/source/iuefits.inc or [LUCIO.IUE.SOURCE]IUEFITS.FOR [LUCIO.IUE.SOURCE]IUEFITS.INC
Sun relocatable libraries	/poseidon/lucio/lib/libluciolib.a /poseidon/lucio/xas/lib/libfitsio.a /poseidon/lucio/xas/lib/libvos.a
VMS relocatable libraries	[LUCIO.LIB]LUCIOLIB.OLB [LUCIO.XAS.LIB]FITSIO.OLB [LUCIO.XAS.LIB]VOS.OLB

Continued in next page.

Optional material for full recompile

Sun sources for luciolib library	all files in in /poseidon/lucio/fortran/luciolib in /poseidon/lucio/xas/libsource/vos
Sun sources for fitsio library	all files in /poseidon/lucio/xas/libsource/fitsio
Sun source for vos library	from /poseidon/lucio/xas/libsource/vos : z_op_sys.f z_tape_open.f z_print_file.f
VMS sources for luciolib library	all files in [LUCIO.FORTRAN.LUCIOLIB]
VMS sources for fitsio library	all files in [LUCIO.XAS.LIBSOURCE.FITSIO]
VMS source for vos library	from [LUCIO.XAS.LIBSOURCE.VOS] : Z_OP_SYS.FOR Z_TAPE_OPEN.FOR Z_PRINT_FILE.FOR

B.3 Instructions for retrieval

If you have an Internet connection, you can retrieve all the above via `ftp` to `poseidon.mi.cnr.it` (192.65.131.49), with username `none` password `none`. Please retrieve only what you really need. Source files shall be retrieved in `ascii` mode, while executables and libraries are to be retrieved in `binary` mode.

If you have only Decnet connections, you should be able to retrieve the material using IFCTR (= 39610) as gateway as in the following example :

```
COPY IFCTR::DEC"none none"::"/poseidon/lucio/iue/pinco.f" *.*
```

B.4 Common procedure installation

B.4.1 load and go installation

This is the simplest case. You just retrieve the executables (mandatory material) and run them. You **do not need to recompile and relink**. Do this if you are not curious of what is inside.

Just select a directory *yourpath* where you want to install everything and copy there the Sun or VMS executable (**only** the one relevant for your system !!). Then define a command for it as :

```
alias iuefits 'yourpath/iuefits \!*'      on Unix)
IUEFITS == "$yourpath:IUEFITS"                (on VMS)
```

B.4.2 installation with relink

If you want to have a look at the source of `IUEFITS`, you can retrieve the source files (listed under optional material). The source code is identical for Unix and VMS, so you need just to retrieve them once.

If you want to modify the program, or for any other reason you need to relink it, you need also to retrieve the relocatable libraries already compiled (also listed as optional material). You can store all the libraries in the same directory, even if the arrangement in Milano is different. The simplest procedure for a generic site is :

a) select a directory `yourpath` where you put everything
b) retrieve source code *and* relocatable libraries
c) compile `iuefits.f` (or `IUEFITS.FOR`)
d) link it with the three libraries
e) delete the object files

B.4.3 full reinstallation

Only in exceptional cases you might need to retrieve all the sources. In general an installation with relink (B.4.2) *should suffice* to most sites. Since each site has local conventions about using libraries, and since the libraries used by `IUEFITS` are used at IFCTR for other projects, only *generic* guidelines can be given :

a) select a directory `yourpath` where you put everything
b) retrieve source code *and* the source for all library routines (you do not need to put library routines in different subdirectories, but you might find this helpful)
c) compile `iuefits.f` (or `IUEFITS.FOR`)
d) compile each library routine separately
e) put each object into a relocatable library (you can put all of them in a single library, or preserve the arrangement in three libraries as you wish) with a command like `ar` and `ranlib` (Unix) or `LIBRARY` (VMS).
f) link the main program with the library (or libraries)
g) delete all object files

B.4.4 IFCTR specific instructions

The following procedure is used to recompile `iuefits` at IFCTR. Please note that the source of the main program, and of the associated include file reside physically on `IFCTR::DUA0:[LUCIO.IUE.SOURCE]` while the files in `/`

`poseidon/lucio/iue/source` are just links. Therefore all edits have to be done on the Vax.

Library routines have to be compiled separately on each system, using *the same* Real Programmer Tool commands. The example refers to the `upcase.f` routine in library `luciolib`. The paths for `libsourcedir` and `libdir` in a generic case can be derived from the table above :

```
libsourcedir /poseidon/lucio/fortran
libdir /poseidon/lucio/lib
complib lucliolib upcase
```

On Sun proceed as follows for the relink of the main program :

login to `helios` **only** !

```
cd /poseidon/lucio/iue/source
configure
comlink iuefits iuefits [term]
```

Note that the file `iuefits.loader` contains indications on where to find the correct libraries. The executable is stored in `/poseidon/lucio/iue /midas/sun` together with all Sun-specific executables, even if it has nothing to do with MIDAS directly.

On Vax proceed as follows for the relink of the main program :

```
cd [lucio.iue.source]
@configure
comlink iuefits iuefits [term]
```

Here too the file `IUEFITS.LOADER` is used to locate the correct libraries. The executable is stored in `[LUCIO.IUE]` in lack of a better place.

Unofficial update 4 May 1993
Made separate configure and loader files for Sun and DEC
Changed iuefits.f source to handle also DEC (as Vax)
DEC executable in /poseidon/lucio/iue/midas