Some Notes on the ONEDSPEC Package

G. Jacoby

National Optical Astronomy Observatories*
June 1985

ABSTRACT

The first phase of the ONEDSPEC prototype package is complete. Comments and some internal descriptions are presented for each task in the package. Also presented are some more global descriptions of strategies used in the package and considerations for future improvements.

Although this report accurately represents the ONEDSPEC package as of the above date, improvements and reorganization of the package are likely to have taken place before release of the IRAF system.

*Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

Some Notes on the ONEDSPEC Package

G. Jacoby

National Optical Astronomy Observatories*
June 1985

1. Why is ONEDSPEC Different?

This section describes some of the ways in which the ONEDSPEC package diverges from other IRAF package strategies. A few of these should someday be modified to more closely adhere to IRAF conventions, but in other cases, restrictions or limitations in the IRAF system are revealed.

Quantity

One of the major differences between a two dimensional image processing package and a one dimensional package is that spectra frequently congregate in groups of hundreds to thousands while two-dimensional images live in groups of tens to hundreds. What this means is that spectral processing must be somewhat more automated and streamlined - the software cannot rely on user input to provide assistance and it cannot afford excessive overhead; otherwise a large fraction of the processing time will be spent where it is least useful.

To process large volumes of spectra in a reasonably automated fashion, the software must be smart enough to know what to do with a variety of similar but different spectra. The way adopted here is to key off header parameters which define the type of spectrum and the processing required. In fact, most of the ONEDSPEC package will not work smoothly without some header parameter information.

It is also important that each task be self-reliant so that the overhead of task stop and restart is avoided. For many operations, the actual computation time is a fraction of a second, yet no operation in the ONEDSPEC package is faster than one second per spectrum due to task overhead. If task startup and stop were required for each spectrum, then the overhead would be much worse.

So the philosophy is one in which each task uses as much information as it can reasonably expect from the spectral image header. Usually this is not more than three or four elements. The strategy of using header information should not be limited to ONEDSPEC. Many image processing problems can be automated to a large degree if header information is used. The success of the KPNO CCD Mountain reduction system emphasizes this point. It would seem prudent that other IRAF applications make use of such information when possible. [See section 3 for a more detailed discussion of headers.]

Spectral Image Names

One implication of the quantity problem is that it must be easy for the user to specify the names of large numbers of spectra. The approach taken for ONEDSPEC was to assign a root name to a group of spectra and then append an index number of 4 or more digits starting with 0000. So spectra, by default, have the form root.0000, root.0001, ... To specify the spectra, the

*Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

user types only the root name and the range of indices such as "root" and "0-99,112-113,105-108". The range decoder accesses the spectral indices in the order given as opposed to access in ascending order, so that the spectrum root.0112 will be processed before root.0105 in the example specification above. Spectra having more general names may be specified using the standard IRAF filename expansion methods if the the range specification is given as null.

The specification of large numbers of images is an area where most IRAF applications are weak. Resorting to odd combinations of bracket and backslash characters in filename specifications is obscure to new users and still fails to meet the general need. The range specification adopted for ONEDSPEC comes closer but introduces a fixed image name format.

Apertures -- A way to group data

Many spectrographs generate multiple spectra simultaneously by placing more than one slit or aperture in the focal plane. Examples include the IIDS, IRS, and Cryogenic Camera in use at Kitt Peak. The Echelle may be considered a multi-aperture instrument for purposes of reductions by associating each order with an "aperture" number.

The concept of aperture can be generalized to indicate a set of spectral data having common group properties such as wavelength coverage. Most tasks in ONEDSPEC will key off an aperture number in the image header and treat those common aperture spectra uniformly. Defining data groups which are to be processed in this fashion is a technique not generally exploited by reduction programs. This is due in part to the problem of image header usage.

For programming convenience and to avoid an additional level of indirectness, in ONEDSPEC the aperture number is used directly as an index in many static arrays. The current implementation has a declaration for 50 apertures and due to the IIDS/IRS notation of apertures 0 and 1, the apertures are zero-indexed, contrary to standard IRAF nomenclature, from 0-49. It would certainly be better to map the aperture numbers to the allowable index range, but the added complexity of another level of indirectness seemed distracting. Actually the mapping can still be done by the header reader, "load_ids_hdr", and unmapped by the header writer, "store_keywords".

Static versus dynamic arrays

Although dynamic storage would alleviate some of the memory requirements in the package, the use of static arrays aids readability and accounts for only about 10 percent of the total task memory space. Many of the arrays are arrays of pointers. For example, in the task BSWITCH, there is an array (called "imnames") of pointers for the names of spectral images, several for each aperture. The actual space for the names is dynamically allocated, so first we allocate an array of pointers for each aperture:

```
call salloc (imnames[aperture], nr_names, TY_POINT)
```

Then, for each of these pointers, space must be allocated for the character arrays:

```
do i = 1, nr_names
    call salloc (Memp[imnames[aperture]+i-1], SZ_LINE, TY_CHAR)
```

Later to access the character strings, a name is specified as:

```
Memc[Memp[imnames[aperture]+nr_of_this_spectrum-1]]
```

If the "imnames" array was also dynamically allocated, the above access would be even less readable. If memory requirements become a serious problem, then these ONEDSPEC tasks should be modified.

Output image names

To retain the consistent usage of root names and ranges, output spectra also have the form root.nnnn. For user convenience, the current output root name and next suffix are maintained as package parameters onedspec.output and onedspec.next_rec. The latter parameter is automatically updated each time a new spectrum is written. This is done by the individual tasks which directly access this package parameter.

There is an interesting side effect when using indirect parameters (e.g.)onedspec.output) for input. In the local task parameter file, the mode of the parameter must be declared hidden. So when the user does an "lpar task", those parameters appear to be unnecessary (that is, they are enclosed in parenthesis). When run, prompts appear because the parameter is an automatic mode parameter in the package parameter file. If run as a background task, this is more annoying. Unfortunately, any other choice of parameter modes produces less desirable actions.

ONEDUTIL

As the number of tasks in ONEDSPEC started growing, the need for a subdivision of the package became clear. The first cut was made at the utility level, and a number of task names (not necessarily physical tasks) were moved out into the ONEDUTIL submenu. In the future, additional tasks will eventually require another subpackage.

Actually, many of the tasks in ONEDUTIL may be more at home in some other package, but a conscious effort was made to avoid contaminating other IRAF packages with tasks written for the ONEDSPEC project. If all the following tasks are relocated, then the need for ONEDUTIL is reduced.

Two of the entries in ONEDUTIL may be considered as more appropriate to DATAIO - RIDSMTN and WIDSTAPE. In fact RIDSMTN can replace the version currently in DATAIO. WIDSTAPE may replace the DATAIO task WIDSOUT if the usage of header parameters does not present a problem.

The task MKSPEC may be a candidate for the ARTDATA package. It should be enhanced to include optional noise generation. Also, it may be appropriate for SINTERP to replace INTERP in the UTILITY package.

I suppose one could argue that SPLOT belongs in the PLOT package. Certainly, the kludge script BPLOT should be replaced by a more general batch plot utility in PLOT. Also, the two task names, IDENTIFY and REIDENTIFY are present in the ONEDSPEC menu for user convenience, but the task declarations in ONEDSPEC.CL refer to tasks in the LONGSLIT package.

Because ONEDUTIL is a logical separation of the tasks, not a complete physical task breakup, there is no subdirectory for ONEDUTIL as there is in other packages. This is a bit messy and it may be best to completely disentangle the tasks in the subpackage into a true package having all the implications.

2. Task Information

There are currently about 30 tasks in the ONEDSPEC package. These are summarized in the menu listing below and a brief description of some less obvious aspects of each follows.

ONEDSPEC

addsets - Add subsets of strings of spectra
batchred - Batch processing of IIDS/IRS spectra
bswitch - Beam-switch strings of spectra to make obj-sky pairs
calibrate - Apply sensitivity correction to spectra
coincor - Correct spectra for photon coincidence
dispcor - Dispersion correct spectra
extinct - Correct data for atmospheric extinction
flatfit - Sum and normalize flat field spectra
flatdiv - Divide spectra by flat field
identify - Identify features in spectrum for dispersion solution
iids - Set reduction parameters for IIDS
irs - Set reduction parameters for IRS
onedutil - Enter ONEDSPEC Utility package
process - A task generated by BATCHRED
reidentify- Automatically identify features in spectra
sensfunc - Create sensitivity function
slist - List spectral header elements
splot - Preliminary spectral plot/analysis
standard - Identify standard stars to be used in sensitivity calc
subsets - Subtract pairs in strings of spectra

ONEDUTIL

bplot - Batch plots of spectra
coefs - Extract mtn reduced coefficients from henear scans
combine - Combine spectra having different wavelength ranges
lcalib - List calibration file data
mkspec - Generate an artificial spectrum
names - Generate a list of image names from a string
rebin - Rebin spectra to new dispersion parameters
ridsmtn - Read IIDS/IRS mountain format tapes
sinterp - Interpolate a table of x,y pairs to create a spectrum
widstape - Write Cyber format IDSOUT tapes

ADDSETS

Spectra for a given object may have been observed through more than one instrument aperture. For the IIDS and IRS, this is the most common mode of operation. Both apertures are used to alternately observe the program objects.

Each instrument aperture may be considered an independent instrument having unique calibration properties, and the observations may then be processed completely independently until fully calibrated. At that point the data may be combined to improve signal-to-noise and reduce systematic errors associated with the alternating observing technique. Because the data are obtained in pairs for IIDS and IRS (but may be obtained in groups of larger sizes from other instruments), ADDSETS provides a way to combine the pairs of observations.

Each pair in the input string is added to produce a single output spectrum. Although the word "pair" is used here, the parameter "subset" defines the number of elements in a "pair" (default=2). The input string is broken down into groups where each group consists of the pair of spectra defined in order of the input list of image names.

"Add" in ADDSETS means:

1. Average the pairs if the data are calibrated to flux (CA_FLAG=0) optionally weighted by the integration time.
2. Add the pairs if uncalibrated (CA_FLAG=-1).

BATCHRED

This is a script task which allows spectra from dual aperture instruments to be processed completely in a batch mode after the initial wavelength calibration and correction has been performed. The processes which may be applied and the tasks referenced are:

1. Declaring observations as standard stars for flux calibration (STANDARD).
2. Solving for the sensitivity function based on the standard stars (SENSFUNC).
3. Generating object minus sky differences and summing individual observations if several were made (BSWITCH).
4. Correcting for atmospheric extinction (BSWITCH).
5. Applying the system sensitivity function to generate flux calibrated data (CALIBRATE).
6. Adding pairs of spectra obtained through the dual apertures (ADDSETS).

Any or all of these operations may be selected through the task parameters.

BATCHRED generates a secondary script task called PROCESS.CL which is a text file containing constructed commands to the ONEDSPEC package. This file may be edited by the user if an entry to BATCHRED is incorrect. It may also be saved, or appended by further executions of BATCHRED.

BATCHRED also generates a log file of the output generated by the ONEDSPEC tasks it calls.

BSWITCH

This task combines multiple observations of a single object or multiple objects taken through a multiaperture instrument. Object minus sky differences are generated as pairs of spectra are accumulated, then optionally corrected for atmospheric extinction, and the differences added together with optional weighting using counting statistics. Each instrument aperture is considered an independent device.

Despite the apparently simple goal of this task, it is probably the most complicated in the ONEDSPEC package due to the bookkeeping load associated with automated handling of large data sets having a number of properties associated with each spectrum (e.g object or sky, aperture number, exposure times).

There are several modes in which BSWITCH can operate. The mode appropriate to the IIDS and IRS assumes that the spectra are input in an order such that after 2N (N=number of instrument apertures) spectra have been accumulated, an equal number of object and sky spectra have been encountered in each aperture. When in this mode, a check is made after 2N spectra have been processed, and the optional extinction correction is applied to the differences of the object minus sky, and then (optionally weighted and) added into an accumulator for the aperture.

If the IIDS mode is switched off, then no guarantee can be made that sky and object spectra pair off. If extinction correction is required, it is performed on each spectrum as it arrives, including sky spectra if any. The spectra are then added into separate accumulators for object and sky for each aperture after optional weighting is applied.

If after all spectra have been processed, there are no sky spectra, the object spectrum is written out. If there is no object spectrum, the sky spectrum is written out after multiplying by -1. (This allows adding an object later on with addsets, but the -1 multiply is probably a

mistake.) If at least one of each, object and sky spectra were encountered, then the difference is computed and written out. Since all accumulations are performed in count rates and later converted back to counts, the object and sky spectra may have different exposure times (non IIDS mode only).

A statistics file is maintained to provide an indication of the quality of the individual spectra going into the sum. The statistics information is maintained internally and only written out after the sums have been generated. The basic data in the file is the count rate of the spectrum having the largest count rate, and the ratios of the count rates from all other spectra to that one.

If weighting is selected, the weights are taken as proportional to the count rate (prior to extinction correction) over a wavelength delimited region of the spectrum. (Perhaps the weight should be proportional to counts, not count rate.) The default wavelength region is the entire spectrum. If the total count rate is negative, the weight is assigned a value of 0.0 and will be disregarded in the sum. (The counts may be negative if the object minus sky difference approaches zero on a bright and cloudy night.)

If extinction is selected, an extinction table is read from the package calibration file. An optional additive term may be applied as computed by the system sensitivity task SENSFUNC which is placed in the parameter sensfunc.add_const. A revision to the standard extinction table (delta extinction as a function of wavelength) may be read from a text file whose name is specified by the parameter sensfunc.rev_ext_file. The file format is that of a text file having pairs of (wavelength, delta extinction) on each line. [The option to solve for this function in SENSFUNC has not yet been implemented, but BSWITCH can read the file that would be generated. Thus, one can experiment with revisions, although this has never been tested.] BSWITCH will interpolate the values given in the file so that a course estimate of the revision may be entered, say if the deltas at U, B, V, R, and I are known.

BEWARE that the extinction correction is performed assuming the header parameters used for airmass refer to a "mean" airmass value for the exposure. In general the header value is wrong! It usually refers to the beginning, middle, or end of the exposure. I have never seen a header airmass value which was an equivalent airmass for the duration of the exposure. This is partly because there is no way to compute a single effective airmass; it is a function of wavelength, telescope position as a function of time, and the extinction function. Fortunately, for most observations this is not very significant. But anyone taking a one hour exposure near 3500 Angstroms at airmass values greater than 2, should not complain when the fluxes look a bit odd.

CALIBRATE

Having a system sensitivity function allows the data to be placed on an absolute flux scale. CALIBRATE performs this correction using the output sensitivity function from SENSFUNC. Operations are keyed to the instrument aperture, and a system sensitivity function is required for each observing aperture, although the user may override this requirement.

A valid exposure time is required (a value of 1.0 should probably be assumed if not present) to compute the observed count rate. Input counts are transformed to units of ergs/cm2/sec/Angstrom (or optionally ergs/cm2/sec/Hz). CALIBRATE will calibrate two dimensional images as well, applying the sensitivity function to all image lines.

The operation is performed on a pixel-by-pixel basis so that the defined sensitivity function should overlap precisely with data in terms of wavelength.

COINCOR

This task applies a statistical correction to each pixel to account for undetected photoevents as a result of coincidental arrival of photons. This is a detector specific correction, although the photoelectric detector model provides a reasonable correction for many detectors when a judicious value for the deadtime parameter is chosen. This model assumes that the correction follows the typical procedures applied to photoelectric photometer data:

$$Ic = Io * \exp [Io * dt / T]$$

where Ic is the corrected count rate in a pixel, Io is the observed count rate in that pixel, dt is the detector deadtime, and T is the observation integration time.

In addition to the photoelectric model, a more accurate model is available for the IIDS and is included in COINCOR. This model is taken from Goad (1979, SPIE Vol 172, 86.) and the correction is applied as:

$$Ic = \ln [1 - Io * t] / t$$

where t is sweep time between pixel samples (t=1.424 msec). The IIDS differs from a photomultiplier detector, in that there is a fixed rate at which each pixel is sampled due to time required for the dissector to sweep across the image tube phosphor whether a photoevent has occurred in a pixel or not. The photomultiplier plus discriminator system assumes that once a photoevent has been recorded, the detector is dead until a fixed interval has elapsed.

DISPCOR

If a relation is known linking pixel coordinate to user coordinate (i.e. wavelength as a function of pixel number), then any non-linearities can be removed by remapping the pixels to a linear wavelength coordinate. This procedure, dispersion correction, is complicated by the lack of a wavelength-pixel solution which is derived from data simultaneously obtained with the object data. Any drifts in the detector then require an interpolation among solutions for the solution appropriate to the object observations. Depending on the detector, this interpolation may be a function of the time of observation, temperature, or some telescope parameter such as airmass. When multiple solutions are available, DISPCOR will linearly interpolate the solution in any available header parameter known to ONEDSPEC (see section 3).

Each solution is read from the database file created by the IDENTIFY task (in TWODSPEC$LONGSLIT), and the image name leading to that solution is also read from the database file. The image is opened to extract the header parameter to be used in the above interpolation. A null name for the interpolation parameter indicates that none is to be used. In this case, one of the options on the "guide" parameter should be set to indicate what solution should be used. The guide may be "precede", "follow", or "nearest" to select the most appropriate choice for each spectrum.

If an explicit wavelength solution is to be used, the parameter "reference" may be used to specify the image name of a comparison spectrum to be used as the reference for the wavelength solution. In this case all spectra will be corrected using a single solution - no flexure correction will be applied.

If the parameter to be used for interpolation is a "time-like" variable, such as RA, UT, ST, then the variable is discontinuous at 24|0 hours. If UT is the chosen parameter (as has been the case for IIDS and IRS spectra), the discontinuity occurs at 5 PM local Kitt Peak time. A comparison spectrum taken at 4:59PM (=23:59h UT, =just before dinner), will be treated as an "end of the night" observation rather than a beginning of the night observation. To circumvent this error, the parameter, "time_wrap", can be specified to a time at which a true zero should be

assigned. For UT at Kitt Peak, a choice like 17h UT (=10AM local, =asleep), is an unlikely hour for nighttime observations to be made. Then for a given night's observations, 17h UT becomes the new zero point in time.

Each solution in the database may be any of the forms legal to IDENTIFY: legendre, chebyshev, spline3, or spline1 - the form is encoded in the database and will automatically be recalled. The interpolation in the solution is performed by locating the pixel location for each required wavelength for the two solutions bounding each observation and linearly interpolating for the appropriate pixel location. One cannot simply interpolate across the coefficients of the solutions to derive a new single solution because the solutions may have different forms or orders, so that the coefficients may have quite different meanings.

Dispersion correction requires that there be equal intervals of wavelength between pixels. The wavelength solution is of a form describing the wavelength for a given pixel location, not a pixel location for a given wavelength. So the solution must be inverted.

The inversion to pixel location for wavelength is done in the following way: The pixel coordinate in the solution is incremented until the desired wavelength is bounded. The pixel value for the desired wavelength is obtained by linearly interpolating across these two bounding pixel locations. A linear approximation appears to be very good for typical solutions, providing proper pixel locations to better than 0.01 pixels. An improvement may be obtained by increasing the order of the interpolation, but the improvement is generally not warranted because the wavelength solutions are rarely known to this accuracy. [Note that the use of real and not double precision limits the precision of this technique! For spectra longer than 50,000 pixels, the errors due to the precision of reals can be serious.]

Note that a transformation to a wavelength coordinate which is linear in the logarithm of wavelength only requires that the inversion occur at wavelengths selected by equal increments in the logarithm of wavelength.

During the actual remapping, 5 possible techniques are available. Actually there are only two techniques: re-interpolation in 4 flavors, and rebinning by partial pixel summation. The re-interpolation may be performed with polynomials of order 1 (=linear), 3, or 5, or by a cubic spline. The 3rd and 5th order polynomials may introduce some ringing in the wings of strong, sharp, features, but the 5th order is good at preserving the high frequency component of the data. The linear and spline interpolators introduce significant smoothing. The rebinning algorithm offers conservation of flux but also smooths the data. In fact, rebinning to a course grid offers a good smoothing algorithm.

At some future date, it would be a good idea to include a "synch" function interpolator in the image interpolator package. This would be a little slower to process, but results in very good frequency response.

Other options in DISPCOR include "ids_mode" which forces spectra from all apertures to a single output mapping (starting wavelength and pixel-to-pixel increment), and "cols_out" forces the output spectra to a specified length, zero-filling if necessary.

DISPCOR will correct two-dimensional data by applying the remapping to all lines in the image. If the input two-dimensional spectrum has only one line, the output spectrum will be written as a one-dimensional spectrum.

EXTINCT

Extinction is currently only available as a script file which drives BSWITCH. This is possible by suppressing all options: weighting, ids_mode, statistics file, and setting the subset pair size to the number of instrument apertures.

FLATDIV

This task divides the specified spectra by their flat field spectra. This is not much more than an "en mass" spectrum divider, with the exceptions that the header elements are used to key on the aperture number so that the appropriate flat field spectrum is used, and that the header processing flags are checked to prevent double divisions and subsequently set after the division. Also, division by zero is guarded by setting any zeroes in the flat field spectrum to 1.0 prior to the division.

FLATFIT

Pixel-to-pixel variations in the detector response can be removed by dividing all observations by a flat field spectrum. Flat field spectra are generally obtained by observing a source having a continuous energy distribution, such as a tungsten filament lamp. This is sometimes called a "quartz" lamp when the enclosing glass bulb is made with quartz rather than silicon. The quartz enclosure transmits ultraviolet light much better than glass.

If the color temperature of the source is very low (or very high, though this is extremely unlikely), then a color term would be introduced into the data when the flat is divided into the data. Large scale variations in the system sensitivity also introduce a color term into the flat - the same variations that are introduced into any spectrum taken with the system. [Large scale variations are evaluated by STANDARD and SENSFUNC, and removed by CALIBRATE.] This is not of any particular importance except that counting statistics are destroyed by the division.

To preserve the statistics, many find it desirable to divide by a flat field spectrum which has been filtered to remove any large scale variations but in which the pixel-to-pixel variations have been retained. A filtered flat can be obtained by fitting a low order polynomial through the spectrum and dividing the spectrum by the polynomial. The result is a spectrum normalized to 1.0 and having high frequency variations only. If one does not care to preserve the statistics, then this procedure is not required. In fact, for certain instruments (the IRS), the fitting and normalizing procedure is not recommended because some intermediate order curvature can be introduced.

The purpose of FLATFIT is to find the combination of parameters which produces a well flattened flat with a minimum of wiggles. The usual curve fitting package is used to fit a function (chebyshev, legendre, spline3, spline1) to the flats. Pixel rejection is user selectable by a choice of cutoff sigmas, both above and below the mean, and an optional growing region [A growing region is the number of pixels on either side of one rejected which will also be rejected - Growing regions are not recommended for most spectral applications]. Any number of iterations may be used to further reject discrepant pixels. The fitting may be performed interactively and controlled by cursor keystrokes to select the fitting order, and other fit parameters.

Prior to the fit, the specified spectra are read, optionally corrected for coincidence losses, and added to accumulators appropriate to their instrument apertures. Each aperture is treated independently, except that, the interactive fitting mode may be selected to operate on the first aperture only, and then apply the same fitting parameters to all other aperture accumulations. Or the interactive procedure may be selected to operate on all apertures or none.

After the fit has been done, the fit is divided into the accumulation and written as a new spectrum having a specified root name and a trailing index indicating the aperture.

IDENTIFY

This task (written by Frank Valdes) is used to identify features in the comparison arcs to be used in the solution for a wavelength calibration. The solution is performed interactively for at least one spectrum and then optionally in a batch mode using REIDENTIFY. IDENTIFY writes to a database file which will contain the solutions generated from each input comparison

spectrum. The database is later used by DISPCOR to correct spectra according to the solution.

IIDS

This script file initializes several hidden parameters in a variety of tasks to values appropriate to the IIDS instrument. There is also a script for the IRS. There should probably be a script for resetting the parameters to a default instrument. These parameters are:

1. onedspec.calib_file - the package parameter indicating which file should be used for standard star calibration data and the atmospheric extinction table (=onedspec$iids.cl.)
2. addsets.subset - the number of instrument apertures (=2).
3. bswitch.ids_mode - assume and check for data taken in beam-switched quadruple mode (=yes).
4. coincor.ccmode - coincidence correction model (=iids).
5. coincor.deadtime - detector deadtime (=1.424e-3 seconds)
6. dispcor.flex_par - the name of the parameter to be used as the guide to removing flexure during the observations (=ut).
7. dispcor.time_wrap - the zero point to be adopted for the flexure parameter if it is a time-like variable having a discontinuity at 0/24 hours (=17).
8. dispcor.idsmode - should data from all instrument apertures be dispersion corrected to a uniform wavelength scale? (=yes).
9. dispcor.cols_out - the number of columns (row length of the spectrum) to which the output corrected spectrum should be forced during mapping (=1024).
10. extinct.nr_aps - the number of instrument apertures (=2).
11. flatfit.order - the order of the fit to be used when fitting to the flat field spectra (=6).
12. flatfit.coincor - apply coincidence correction to the flat field spectra during accumulations (=yes).
13. flatdiv.coincor - apply coincidence correction to all spectra during the flat field division process (=yes).
14. identify.function - the fitting function to be used during the wavelength solution process (=chebyshev).
15. identify.order - the order of the fit to be used during the wavelength solution process (=6).

IRS

This script file initializes several hidden parameters in a variety of tasks to values appropriate to the IRS instrument. These parameters are:

1. onedspec.calib_file - the package parameter indicating which file should be used for standard star calibration data and the atmospheric extinction table (=onedspec$irs.cl.)
2. addsets.subset - the number of instrument apertures (=2).
3. bswitch.ids_mode - assume and check for data taken in beam-switched quadruple mode (=yes).
4. coincor.ccmode - coincidence correction model (=iids).
5. coincor.deadtime - detector deadtime (=1.424e-3 seconds)

6. dispcor.flex_par - the name of the parameter to be used as the guide to removing flexure during the observations (=ut).
7. dispcor.time_wrap - the zero point to be adopted for the flexure parameter if it is a time-like variable having a discontinuity at 0/24 hours (=17).
8. dispcor.idsmode - should data from all instrument apertures be dispersion corrected to a uniform wavelength scale? (=yes).
9. dispcor.cols_out - the number of columns (row length of the spectrum) to which the output corrected spectrum should be forced during mapping (=1024).
10. extinct.nr_aps - the number of instrument apertures (=2).
11. flatfit.order - the order of the fit to be used when fitting to the flat field spectra. IRS users have frequently found that any curvature in the fit introduces wiggles in the resulting calibrations and a straight divide by the flat normalized to the mean works best (=1).
12. flatfit.coincor - apply coincidence correction to the flat field spectra during accumulations (=no).
13. flatdiv.coincor - apply coincidence correction to all spectra during the flat field division process (=no).
14. identify.function - the fitting function to be used during the wavelength solution process (=chebyshev).
15. identify.order - the order of the fit to be used during the wavelength solution process. The IRS has strong deviations from linearity in the dispersion and a fairly high order is required to correct the dispersion solution (=8).

ONEDUTIL

This is a group of utility operators for the ONEDSPEC package. They are documented separately after the ONEDSPEC operators. ONEDUTIL is a "pseudo-package" - it acts like a package under ONEDSPEC, but many of its logical tasks are physically a part of ONEDSPEC. This is done to minimize disk storage requirements, and to logically separate some of the functions from the main ONEDSPEC menu which was getting too large to visually handle.

PROCESS

This task generally does not exist until the user executes the script task BATCHRED which creates PROCESS.CL, a secondary script file containing a CL command stream to batch process spectra. The task is defined so that the CL is aware of its potential existence. It is not declared as a hidden task so that the user is also aware of its existence and may execute PROCESS in the foreground or background.

REIDENTIFY

This task (written b Frank Valdes) is intended to be used after IDENTIFY has been executed. Once a wavelength solution has been found for one comparison spectrum, it may be used as a starting point for subsequent spectra having similar wavelength characteristics. REIDENTIFY provides a batch-like means of performing wavelength solutions for many spectra. The output solution is directed to a database text file used by DISPCOR.

SENSFUNC

This task solves for the system sensitivity function across the wavelength region of the spectra by comparison of observations of standard stars to their (assumed) known energy distribution. Each instrument aperture is treated completely independently with one exception discussed later. SENSFUNC is probably the largest task in the ONEDSPEC package due to heavy use of interactive graphics which represents more than half of the actual coding.

Input to SENFUNC is the "std" text file produced by STANDARD containing the ratio of the count rate adjusted for atmospheric extinction to the flux of the star in ergs/cm2/s/Angstrom. Both the count rates and fluxes are the average values in the pre-defined bandpasses tabulated in the calibration file (indicated by the parameter onedspec.calib_file).

Each entry is the "std" file may have an independent set of wavelength sampling points. After all entries have been loaded, a table containing all sampled wavelengths is built (a "composite" wavelength table) and all sensitivity values are reinterpolated onto this sampling grid. This allows the inclusion of standards in which the observational samples are not uniform.

When multiple measurements are available, one of two corrections may be applied to the data to account for either clouds or an additive extinction term. The effect of clouds is assumed to be grey. Each contributing observation is compared to the one producing the highest count rate ratio at each wavelength sample. The deviation averaged over all wavelengths for a given observation is derived and added back to each wavelength sample for that observation. This produces a shift (in magnitudes) which, on the average across the spectrum, accounts for an extinction due to clouds. This process is called "fudge" primarily for historical reasons (from the IPPS, R.I.P.) and also because there is questionable justification to apply this correction. One reason is so that one can better assess the errors in the data after a zero-point correction has been made. Another is that the sensitivity function is that closest to a cloud-free sky so that calibrations may approach a true flux system if one standard was observed during relatively clear conditions. Alsom there are claims that the "color solution" is improved by "fudging", but I admit that I don’t fully understand this argument.

[Perhaps it goes as follows: Although a grey scale correction is applied to each observation, a color term is introduced in the overall solution. Consider the case where 5 magnitudes of cloud extinction obscure one standard relative to another. This star generates a sensitivity curve which is a factor of 100 smaller. When averaged with the other curve, any variations are lost, and the net curve will be very similar to the first curve divided by 2. Now apply a "fudge" of 5 magnitudes to the second curve. On the average, both curves have similar amplitudes, so variations in the second now influence the average. The net curve then has color dependent variations not in the "un-fudged" net curve. If we assume that the variations in the individual observations are not systematic, then "fudge" will improve the net color solution. Amazing, isn’t it? End of hypothesis.]

The second form of correction is much more justifiable. In ONEDSPEC it is referred to as a "grey shift" and accounts for possible changes in the standard atmospheric extinction model due to a constant offset. SENSFUNC will optionally solve for this constant provided the observations sample a range of airmass values. The constant is computed in terms of magnitudes per airmass, so if the airmass range is small, then a large error is likely. To solve for this value, a list of pairs of delta magnitude (from the observation having the greatest sensitivity) as a function of delta airmass (relative to the same observation) is generated for all observations. The list is fit using a least squares solution of the form:

delta_mag = delta_airmass * grey_shift

Note that this is a restricted least-squares in the sense that there is no zero-point term. The standard curve fit package in IRAF does not support this option and the code to perform this is included in SENSFUNC.

Because the atmosphere is likely to be the same one for observations with each instrument aperture, it is not appropriate to limit the least-squares solution to the individual apertures, but rather to combine all the data to improve the solution. This would mean that the user could not view the effects of applying the grey term until all apertures had been analyzed. So, although each aperture is solved independently to derive a preliminary value, a final value is computed at the end when all data have been reviewed. This is the one exception to the independent aperture equals independent instrument philosophy.

When "fudging" is applied, the sensitivity function that is generated is altered to account for the shifts to the observations. But when the "grey shift" is computed, it cannot be directly applied to the sensitivity function because it must be modified by the observing airmass for each individual object. So the grey shift constant is written into the image headers of the generated sensitivity functions (which are IRAF images), and also placed into the task parameter "add_const" to be used later by BSWITCH.

SENSFUNC can be run in an interactive mode to allow editing of the sensitivity data. There are two phases of interaction: (1) a review of the individual observations in which every data element can be considered and edited, and (2) a review of the composite sensitivity table and the calculated fit to the table. In the interactive mode, both phases are executed for every instrument aperture.

At both phases of the interactive modes there will be a plot of the error in the input values for each wavelength. This is an RMS error. [The IPPS plotted standard error which is always a smaller number and represents the error in the mean; the RMS represents the error in the sample. I'm not sure which is better to use, but RMS is easier to understand. RMS is the same as the standard deviation.] During phase one, the rms is computed as the standard deviation of the sensitivity in magnitudes; but during phase two, it is computed as the standard deviation in raw numbers and then converted to a magnitude equivalent. The latter is more correct but both converge for small errors.

There is one option in SENSFUNC which has never been tried and it won't work - the option to enter a predefined table of sensitivities as a function of wavelength as a simple text file. This option may be useful a some time and should probably be fixed. I think the only problem with it is a lack of consistency in the units.

An additional option has been requested but it is not clear that it is a high priority item - the ability to compute the extinction function. There may be instances when the mean extinction table is not appropriate, or is not known. If sufficient data are available (many observations of high precision over a range of airmasses during a photometric night), then the extinction function is calculable. Presently SENSFUNC can only compute a constant offset to the extinction function, but the same algorithm used may be applied at each wavelength for which observations are made to compute a correction to an adopted extinction function (which may be zero), and the correction can then be written out to the revised extinction table file. This file will then be read by BSWITCH during the extinction correction process. So at each wavelength, pairs of delta magnitude as a function of delta airmass are tabulated and fit as above:

delta_mag[lambda] = delta_airmass * delta_extinction[lambda]

Because the data have been heavily subdivided into wavelength bins, there are only a few measurements available for solving this least-squares problem and the uncertainties are large unless many observations have been taken. Experience has shown that at least 7-8 measurements are needed to come close, and 15 measurements are about the minimum to get a good solution. Unless the data are of high quality, the uncertainty in the solution is comparable to the error in assuming a constant offset to the mean extinction function. Nevertheless, the option should be installed at some time since some observers do obtain the necessary data.

SLIST

The spectrum specific header elements are listed in either a short or long form. See the discussion on headers (section 3) for an explanation of the terms. Values for airmass are printed if present in the header; otherwise, the value is given as the string "?????" to indicate no value present (even if one can be calculated from the telescope pointing information elsewhere in the header).

The short form header lists only the image name, whether it is an object or sky observation, the spectrum length, and the title.

SPLOT

This is probably the second largest task in the ONEDSPEC package. It continues to grow as users provide suggestions for enhancement, although the growth rate appears to be slowing. SPLOT is an interactive plot program with spectroscopy in mind, although it can be used to plot two dimensional images as well.

SPLOT should still be considered a prototype - many of the algorithms used in the analysis functions are crude, provided as interim software to get results from the data until a more elaborate package is written. It would probably be best to create an analysis specific package - SPLOT is reasonably general, and to enhance it further would complicate the keystroke sequences.

Ideally it should be possible to do anything to a spectrum with a single keystroke. In reality, several keystrokes are required. And after 15 or 20 functions have been installed, the keystroke nomenclature becomes obscure - all the best keys are used up, and you have to resort to things like ’(’ which is rather less mnemonic than a letter. So some of the functionality in SPLOT has been assigned to the "function" submenu invoked by ’f’ and exited by ’q’ keystrokes. These include the arithmetic operators: add, multiply by a constant, add, subtract, multiply, divide by a spectrum, and logarithms, square root, inverse, and absolute value of a spectrum.

Some of the analysis functions include: equivalent width, line centers, flux integration under a line, smoothing, spectrum flattening, and deblending of lines.

The deblender has serious limitations but handles about half the cases that IIDS/IRS users are interested in. It fits only Gaussian models to the blends, and only a single width parameter. The fit is a non-linear least-squares problem, so starting values present some difficulties. All starting values are initialized to 1.0 - this includes the width, relative strengths of the lines, and deviation from initial marked centers. The iterative solution usually converges for high signal-to-noise data, but may go astray, resulting in a numerical abort for noisy data. If this occurs, it is often possible to find a solution by fitting to a single strong line to force a better approximation to the starting values, and then refit the blend of interest.

The non-linear least-squares routine is one obtained from an industrial source. The code is very poorly written and in FORTRAN. No one should attempt to understand it. The basic algorithm is an unconstrained simplex minization search combined with a parabolic linear least-squares approximation when in the region of a local minimum. A test was made comparing this to the algorithm in Bevington, and the Bevington algorithm appeared less likely to converge on noisy data. Only one test case was used, so this is hardly a fair benchmark.

The problem with non-convergence is that a floating point error is almost surely to arise. This is usually a floating point over/under flow while computing an exponential (as required for a Gaussian). In UNIX, there is apparently no easy way to discriminate from FORTRAN which floating point exception has occurred, and so there is no easy way to execute a fix up and continue. This is most unfortunate because the nature of these non-linear techniques is that given a chance, they will often recover from searching down the wrong alley. A VMS version of the same routines seems to survive the worst data because the error recovery is handled somewhat better. [The VMS version also seems to run much faster, presumably because the floating point

library support is better optimized.]

The net result of all this, is that a weird undocumented subroutine is used which provides no error estimate. The Bevington routines do provide an error estimate which is why I wanted to use them. [In fact, there is no way to exactly compute the errors in the fit of a non-linear least-squares fit. One can however apply an approximation theory which assumes the hypersurface can be treated locally as a linear function.]

There are several methods for computing equivalent widths in SPLOT. The first method for measuring equivalent width is simply to integrate the flux above/under a user defined continuum level. Partial pixels are considered at the marked endpoints. A correction for the pixel size, in Angstroms, is applied because the units of equivalent width are Angstroms. You will probably get a different answer when doing equivalent width measurements in channel mode ('$' keystroke) as compared to wavelength mode ('p').

Centering is performed as a weighted first moment of the region:

```
int1 = integral [ (I-Ic) * sqrt (I-Ic) * w]
int2 = integral [ (I-Ic) * sqrt (I-Ic)    ]
xc = int1 / int2
```

where I is the intensity at the pixel at wavelength w, and Ic is the estimated continuum intensity. The square root term provides the weighting assuming photon statistics [sigma = sqrt(I)], and xc is the derived center of the region.

An alternative method for equivalent widths was supplied by Caty Pilachowski and is described in some detail in the help file for SPLOT. This method is fast and insensitive to cursor settings, so the user can really zip through a spectrum quickly.

Smoothing is performed using a simple boxcar smooth of user specified size (in pixels). To handle edge effects, the boxcar size is dynamically reduced as the edge is approached, thereby reducing the smoothing size in those regions.

The flattening operator is a preliminary one, written before the curve fitting package was available in IRAF. This operator should probably be re-written to include the interactive style used in FLATFIT. Currently the flattening is done using classic polynomial least-squares with pixel rejection chosen to preferentially reject absorption lines and strong emission lines. The rejection process is repeated through a number of iterations specifiable as a hidden parameter to SPLOT. This is poorly done - the order of the fit and the number of iterations should be controllable while in SPLOT. However, experimentation has shown that for a given series of spectra, the combination of rejection criteria, order, and iteration count which works well on one spectrum will generally work well on the other spectra. Note that the flatten operator attempts to find a continuum level and normalize to that continuum, not to the average value of the spectrum.

There are also the usual host of support operators - expansion, overplotting, and so forth. There is also a pixel modifier mode which connects two cursor positions. This forces a replot of the entire spectrum after each pair of points has been entered. This should probably be changed to inhibit auto-replot.

Some users have requested that all two cursor operators allow an option to escape from the second setting in case the wrong key was typed. I think this is a good idea, and might be implemented using the "esc" key (although I could not seem to get this keystroke through the GIO interface).

Another user request is the option to overplot many spectra with autoscaling operational on the entire range. This is also a good idea. Yet another improvement could be made by allowing the user to specify the x and y range of the plot, rather than autoscaling.

There is one serious problem with respect to plotting spectra corrected to a logarithmic

wavelength scale. It would be nice to plot these spectra using the logarithmic axis option, but this option in GIO requires that at least one entire decade of x axis be plotted. So for optical data, the x axis runs from 1000 Angstroms to 10,000 Angstroms. Imagine a high dispersion plot having only 100 Angstroms of coverage - the plot will look like a delta function! The current version of SPLOT uses a linear axis but plots in the log10 of wavelength. Not very good, is it.

STANDARD

This task computes the sensitivity factor of the instrument at each wavelength for which an a priori measured flux value is known and within the wavelength range of the observations. Sensitivity is defined as [average counts/sec/Angstrom]/[average ergs/cm2/sec/Angstrom] over the specified bandpass for which the star has been measured. Both numerator and denominator refer to quantities above the Earth's atmosphere and so the count rates must be corrected for extinction. The wavelengths of known measurements, the bandpasses, the fluxes (in magnitudes), and the mean extinction table are read from a calibration file whose name is specified by the calib_file parameter (see LCALIB for a description of this file). If a magnitude is exactly 0.0, it is assumed that no magnitude is known for this star at the wavelength having a 0.0 magnitude. This allows entries having incomplete information.

As each observation is read, it is added into an accumulator for its aperture. Or subtracted if it is a sky measurement. After a pair of object and sky observations have been added, the difference is corrected for extinction (as in BSWITCH), converted to counts per second, and integrations performed over the bandpasses for which flux measures are known. The bandpasses must be completely contained within the spectrum - partial coverage of a bandpass disqualifies it from consideration. The integrations are compared with the known flux values and the ratio is written to a text file (the "std" file) along with the wavelength of the measurement and the total counts in the bandpass. The total counts value may be used by SENSFUNC for weighting the measurements during averaging.

Many users are surprised by the order of the spectral names printed out as STANDARD executes since the order is not necessarily ascending through the spectrum list. This is because the name printed is the name of the object spectrum most recently associated with an object-sky pair. So if a sky pair is several spectra down the list, an intervening object-sky pair taken through a different instrument aperture may be processed in the meantime. For example, say spectra 1-8 are taken so that object spectra numbers 1 and 7 and sky spectra 3 and 5 are taken through aperture 0, object spectra 4 and 6 and sky spectra 2 and 8 are taken through aperture 1. [This is a very common pattern for IIDS/IRS users.] Then spectrum 1 and 3 will pair up and be processed first (spectrum name 1 will be printed). Then 4 and 2 (name 4 printed), then 7 and 5 (name 7 printed), and then 6 and 8 (name 6 printed). So the order of names printed will be 1,4,7,6. Simple, isn't it?

If the input spectra are not taken in a beam-switched mode then the parameter "beam_switch" should be set to no. Then no sky subtraction will be attempted.

The user may enter sensitivity values directly into a file and use it as the "std" file for a correction. See the help file for STANDARD for a description of the entries in the file, and see a typical file.

STANDARD offers a limited interactive mode. The first sky subtracted spectrum is displayed and the bandpasses at which sensitivity measurements are made will be shown as boxes. This provides a means to see where the measurements are falling on the observational data and to assess whether a bandpass may be including some absorption edge which may be affecting the measurement. While it is true that the wavelengths of the reference measurements should fall in the same place, the effects of instrument resolution and inaccuracies in the wavelength calibration may shift the positions of the apparent bandpasses. The samples may then be biased.

The second purpose of the interactive mode is to allow the user to artificially create new

bandpasses on the fly. By placing the cursor to bound a new wavelength region, STANDARD will interpolate in the magnitude table of the reference star to estimate the magnitude of the star at the bounded wavelength. The sensitivity will be calculated at that wavelength just as if the bandpass had come from the calibration file. This option should be exercised with care. Obviously, points should not be generated between reference wavelengths falling on strong absorption lines, or on a line either. This option is most useful when at a high dispersion and few samples happen to fall in the limited wavelength region. Sufficient space is allocated for 10 artificial samples to be inserted. Once the artificial bandpasses have been designated, they are applied to the entire sequence of spectra for the current invocation of STANDARD. Once STANDARD completes, the added bandpasses are forgotten. This prevents an accidental usage of newly created bandpasses on stars of different spectral types where a bandpass may fall in a region of continuum for one star, but on an absorption line in another.

SUBSETS

This is a simple task to subtract the second spectrum from the first in a series of spectra. So if spectra 1-10 are input, 5 new spectra will be created from 1 minus 2, 3 minus 4, and so on. This is a straight subtraction, pixel for pixel, with no compensation for exposure time differences. The header from the first spectrum of the pair is applied to the output spectrum.

The ONEDUTIL tasks

These utility tasks are logically separated from the ONEDSPEC package.

COEFS

This task reads the header parameters contained in comparison arc spectra describing the wavelength solution generated by the mountain reduction program and re-writes the solution parameters into a database text file for use by DISPCOR. Otherwise those solutions would be lost. COEFS assumes that the coefficients represent a Legendre polynomial which is what the mountain reduction programs use.

COMBINE

When an object has been observed over a wide range of wavelength coverage by using more than one instrumental setup (such as a blue and a red setting) or with different instruments (such as IUE and the IRS), it is often desirable to combine the spectra into a single spectrum. COMBINE will rebin a group of spectra to new spectra having a single dispersion and average the new spectra to create a single long spectrum. If there are gaps in the composite spectrum, zeroes are used as fillers. Ideally those pixels which have no known value should be considered blank pixels. IRAF does not currently support blank pixels, so zeroes are used for now. [One might suggest using INDEF, but then all other routines will have to check for this value.] A side effect of choosing 0.0 is that during the averaging of overlapping spectra, a true 0.0 will be ignored by COMBINE. The basic rebinning algorithms used in DISPCOR are used in COMBINE (and also REBIN).

The averaging can be weighted by exposure time, or by user assigned weights. It would be better if each spectrum had an associated vector of weights (one weight at each wavelength) so that the weighted averaging could be done on a pixel basis. This is very expensive in terms of both storage and file access overhead since each spectrum would require twice the storage and number of files. [Actually weights could be small 4 bit integers and take up very little space.]

A less ideal alternative would be to place a small number (about 16) of weight parameters in the header file which represent the approximate weights of that many regions of the spectrum, and then one could interpolate in these parameters for a weight appropriate to the pixel of

interest.

A third solution (and even less ideal) is to place a single parameter in the header which represents an average weight of the entire spectrum. For the latter two cases, the header weights could be derived from the average counts per wavelength region - the region being the entire spectrum in the last case. The weights must be entered into the header during the BSWITCH operation since that is the last time that true counts are seen. [An implicit assumption is that counts are proportional to photons. If data from two different instruments are to be averaged, then the weights should be expressed in photons because the ratio of counts to photons is highly instrument dependent.]

COMBINE suffers from a partial pixel problem at the end points. Interpolation at the ends can lead to an underestimate of the flux in the last pixels because the final pixel is not filled. When averaging in data from another spectrum or instrument, these pixels show up as sharp drops in the spectrum. The problem appears due to the rebinning algorithm and should be corrected someday (also in DISPCOR and REBIN).

LCALIB

This utility provides a means of checking the calibration files containing the standard star fluxes and extinction table. Any of the entries in the file may be listed out - the bandpasses, extinction, standard star names, standard star fluxes in either magnitudes, flambda, or fnu. For a description of the calibration file format, see the help documentation for LCALIB.

The primary uses for LCALIB are to verify that new entries in the tables are correct, to generate a list of standard star names in a calibration file, and to produce a table of fluxes for a given standard star. The table may then be used to generate a spectrum over a specified wavelength region using SINTERP and overplotted with observational data to check the accuracy of the reductions.

MKSPEC

MKSPEC provides a way to generate a limited set of artificial spectra. Noise generation is not available. The current options are to generate a spectrum which is either a constant, a ramp, or a black body. The spectrum may be two dimensional, but all image lines will be the same.

NAMES

This is the simplest task in the ONEDSPEC package. It generates the image file names which are implied by a root name and record string. The primary use for this task is to generate a list of image names to be used as input for some other program such as WFITS. The output from NAMES can be redirected to file and that file used with the "@file" notation for image name input. An optional parameter allows an additional string to be appended to the generated file name to allow a subraster specification.

REBIN

Spectra are rebinned to the wavelength parameters specified by either matching to a reference spectrum or by user input. The algorithms are those used by DISPCOR and the same options for the interpolation method are available. REBIN is useful when data are obtained with different instruments or setups producing roughly comparable wavelength ranges and possibly different dispersions, and the data are to be compared. REBIN may also be used as a shift operator by specifying a new starting wavelength. Or it may be used as a smoothing operator by specifying a course dispersion. It may also be used to convert between the two formats - linear in wavelength and linear in the logarithm of wavelength. This latter option has not been

thoroughly exercised - proceed with caution.

RIDSMTN

This task was stolen from the DATAIO package to make the following modification: IIDS and IRS data are both written as 1024 pixel spectra at the mountain. But the detectors do not produce a full 1024 pixels of acceptable data. In fact the IRS only has 936 pixels. The data are written this way to conform to the IIDS ideal spectrum which does have 1024 pixels, but the first few (about 6) are not usable. To signal the good pixels, the IIDS/IRS header words NP1 and NP2 are set to the beginning and ending good pixels. Actually NP1 points to the first good pixel minus one. [Really actually NP1 and NP2 may be reversed, but one is big and the other small so you can tell them apart.]

The version of RIDSMTN in ONEDUTIL keys off these parameters and writes images containing only good pixels which means that the images will be smaller than 1024 pixels. The user has the option of overriding the header values with the task parameters "np1" and "np2". These may be specified as 1 and 1024 to capture the entire set of pixels written to tape or any other subset. Beware that np1 and np2 as task parameters refer to the starting pixel and ending pixel respectively. None of this nonsense about possible role reversals or "first good minus one" is perpetuated.

SINTERP

I think this is a handy little program. It provides a way to make an IRAF spectral image from a table of values in a text file. The table is interpolated out to any length and at any sampling rate. A user can create a table of corrections to be applied to a set of spectra, for example, use SINTERP to build a spectrum, and run CALIBRATE to multiply a group of spectra by the correction.

The original raison d'etre for SINTERP was to create spectra of standard stars from the listing of fluxes generated by LCALIB. Using SPLOT the created spectrum can be overplotted with calibrated observations to compare the true tabulated fluxes with the observed fluxes.

SINTERP grew out of the task INTERP in the UTILITIES package and works pretty much the same way. One major change is that the table containing the x-y pairs is now stored in a dynamically allocated array and can be as large as the user requests. The default size is 1024 pairs, but the parameter tbl_size can be set to a larger value. This then allows one to create a spectrum from its tabulated values of wavelength and flux even if the the table is several thousand elements long. Note that the option to route the output from INTERP to STDOUT has been retained if a new table is to be generated rather than an IRAF image.

Another major change from INTERP is the use of the IRAF curve fitting routines as an option. These were not originally available. The choices now include linear or curvey interpolators, Legendre or Chebyshev polynomial fits, and cubic or linear splines.

WIDSTAPE

This task has vague origins in the DATAIO task WIDSOUT which writes a tape having the format of the IDSOUT package which ran on the CYBER (R.I.P.). For convenience to users this format has been maintained for spectra having lengths up to 1024 pixels. The version in DATAIO requires that the user enter all the header parameters as task parameters. For several hundred spectra, this approach is unwieldy. Because the ONEDSPEC package uses the header parameters heavily, it is able to read them directly and write the values to the tape file without user intervention.

The output tape (or diskfile) may be in either ASCII or EBCDIC format. Spectra shorter than 1024 are zero filled. Each invocation of the task write a new tape file followed by a tape

mark (EOF).

3. Image Header Parameters

The ONEDSPEC package uses the extended image header to extract information required to direct processing of spectra. If the header information were to be ignored, the user would need to enter observing parameters to the program at the risk of typographical errors, and with the burden of supplying the data. For more than a few spectra this is a tedious job, and the image header information provides the means to eliminate almost all the effort and streamline the processing.

However, this requires that the header information be present, correct, and in a recognizable format. To meet the goal of providing a functional package in May 1985, the first iteration of the header format was to simply adopt the IIDS/IRS headers. This allowed for processing of the data which would be first used heavily on the system, but would need to be augmented at a later date. The header elements may be present in any order, but must be in a FITS-like format and have the following names and formats for the value fields:

Parameter	Value Type	Definition
HA	SX	Hour angle (+ for west, - for east)
RA	SX	Right Ascension
DEC	SX	Declination
UT	SX	Universal time
ST	SX	Sidereal time
AIRMASS	R	Observing airmass (effective)
W0	R	Wavelength at center of pixel 1
WPC	R	Pixel-to-pixel wavelength difference
NP1	I	Index to first pixel containing good data (actually first-1)
NP2	I	Index to last pixel containing good data (last really)
EXPOSURE	I	Exposure time in seconds (ITIME is an accepted alias)
BEAM-NUM	I	Instrument aperture used for this data (0-49)
SMODE	I	Number of apertures in instrument minus one (IIDS only)
OFLAG	I	Object or sky flag (0=sky, 1=object)
DF-FLAG	I	Dispersion fit made on this spectrum (I=nr coefs in fit)
SM-FLAG	I	Smoothing operation performed on this spectrum (I=box size)
QF-FLAG	I	Flat field fit performed on this spectrum (0=yes)
DC-FLAG	I	Spectrum has been dispersion corrected (0=linear, 1=logarithmic)
QD-FLAG	I	Spectrum has been flat fielded (0=yes)
EX-FLAG	I	Spectrum has been extinction corrected (0=yes)
BS-FLAG	I	Spectrum is derived from a beam-switch operation (0=yes)
CA-FLAG	I	Spectrum has been calibrated to a flux scale (0=yes)
CO-FLAG	I	Spectrum has been coincidence corrected (0=yes)
DF1	I	If DF-FLAG is set, then coefficients DF1-DFn (n <= 25) exist

The values for the parameters follow the guidelines adopted for FITS format tapes. All keywords occupy 8 columns and contain trailing blanks. Column 9 is an "=" followed by a space. The value field begins in column 11. Comments to the parameter may follow a "/" after the value field. The value type code is as follows:

SX This is a sexagesimal string of the form ’12:34:56 ’ where the first quote appears in column 11 and the last in column 30.

R This is a floating point ("real") value beginning in column 11 and extending to column 30 with leading blanks.

I This is an integer value beginning in column 11 and extending to column 30 with leading blanks.

The parameters having FLAG designations all default to -1 to indicate that an operation has not been performed. The ONEDSPEC subroutines "load_ids_hdr" and "store_keywords" follow these rules when reading and writing spectral header fields. If not present in a header, load_ids_hdr will assume a value of zero except that all flags are set to -1, and the object flag parameter defaults to object.

When writing an image, only the above parameters are stored by store_keywords. Other header information is lost. This needs to be improved.

Not all programs need all the header elements. The following table indicates who needs what. Tasks not listed generally do not require any header information. Header elements not listed are not used. The task SLIST requires all the elements listed above. The task WIDTAPE requires almost all (except NP1 and NP2). The headings are abbreviated task names as follows:

ADD	addsets	COI	coincor	FIT	flatfit
BSW	bswitch	COM	combine	REB	rebin
CAL	calibrate	DIS	dispcor	SPL	splot
COE	coefs	FDV	flatdiv	STA	standard

Key	ADD	BSW	CAL	COE	COI	COM	DIS	FDV	FIT	REB	SPL	STA
HA		X										X
RA		X										X
DEC		X										X
ST		X										X
UT		X										X
AIRMASS		X										X
W0		X	X			X				X	X	X
WPC		X	X			X				X	X	X
NP1											X	
NP2											X	
EXPOSURE	X	X			X	X					X	
BEAM-NUM		X	X				X	X	X		X	X
OFLAG		X										X
DF-FLAG				X								
DC-FLAG		X				X				X	X	X
QD-FLAG								X				
EX-FLAG		X										
BS-FLAG		X										
CA-FLAG	X		X								X	
CO-FLAG					X							
DFn				X								

Headers From Other Instruments

The header elements listed above are currently created only when reading IIDS and IRS data from one of the specific readers: RIDSMTN and RIDSFILE. The time-like parameters, (RA, DEC, UT, ST, HA), are created in a compatible fashion by RCAMERA and RFITS (when the FITS tape is written by the KPNO CCD systems).

For any other header information, the ONEDSPEC package is at a loss unless the necessary information is edited into the headers with an editing task such as HEDIT. This is not an acceptable long term mode of operation, and the following suggestion is one approach to the header problem.

A translation table can be created as a text file which outlines the mapping of existing header elements to those required by the ONEDSPEC package. A mapping line is needed for each parameter and may take the form:

```
1D_param   default   hdr_param   key_start   value_start   type   conversion
```

where the elements of an entry have the following definitions:

1D_param	The name of the parameter expected by the ONEDSPEC package, such as EXPOSURE, OFLAG, BEAM-NUM.
default	A value to be used if no entry is found for this parameter or if no mapping exists.
hdr_param	The string actually present in the existing image header to be associated with the ONEDSPEC parameter.
key_start	The starting column number at which the string starts in the header.
value_start	The starting column number at which the string describing the value of the parameter starts in the header.
type	The format type of the parameter: integer, real, string, boolean, sexigesimal.
conversion	If the format type is string, a further conversion may optionally be made to one of the formats listed under type. The conversion may requires some expression evaluation.

Consider the example where the starting wavelength of a spectrum is contained in a FITS-like comment and the object- sky flag in a similar fashion:

```
COMMENT = START-WAVE  4102.345   / Starting wavelength
COMMENT = OBJECT/SKY  'SKY       '/ Object or Sky observation
```

The translation file entries for this would be:

```
W0  0.0  START-WAVE  12  24  R
OFLAG 0  OBJECT/SKY  12  25  S  SKY=0;OBJECT=1
```

The first entry is fairly simple. The second requires an expression evaluation and second conversion.

A translation file can be built for each instrument and its special header format, and the file name can be associated with a ONEDSPEC package parameter. The two subroutines in ONEDSPEC dealing directly with the headers (load_ids_hdr and store_keywords) can be modified or replaced to access this file and translate the header elements.