The SMARTS[1] 1.5m RC Spectrograph Reduction Software Manual

Frederick M. Walter
SUNY Stony Brook
Version 1.0
January 2009

Table of Contents

---

[1]SMARTS, the Small and Medium Aperture Research Telescope System, is a consortium of about a dozen institutions and individual users who operate the 4 small telescopes at CTIO, under contract to AURA.

1 Introduction

This documentation describes a set of IDL procedures which can be used to reduce spectra obtained using the RC spectrograph on the SMARTS 1.5m telescope at Cerro Tololo.

This code has been under development since February 2003. It is run routinely most mornings to download and process data obtained the previous night. In most cases it runs smoothly. There are some known bugs and work-arounds that will be described later.

The software requires only minimal user oversight.

2 The Software Distribution

This software is freely distributed. It may be copied or modified for legitimate scientific purposes. An acknowledgement would be fitting if you actually use the software for anything useful.

Everything you will need, including some test data, are in a tar file, s15reduct.tar.gz. This tar file contains:

- s15caldata.tar.gz

- s15reduct.pro.tar.gz

- s15testdata.tar.gz

- two database definition files, s15p_2008.dbd and s15_2008.dbd

Download this file and untar it.

2.1 Environment Variables

You will need to define two environment variables in your .cshrc or .bashrc file:

S15DATA points to the main data directory. e.g.,
  setenv S15DATA /raid1/DATA/SMARTS/1.5m

S15CALDATA points to the directory containing the wavecal line lists, the fiducial wavecal spectra, and the spectrophotometric standard star data. e.g.,
  setenv S15CALDATA /raid1/DATA/SMARTS/1.5m/CALDATA/

2.2 Contents of the Distribution

s15reduct.pro.tar.gz contains the IDL procedures you will need (along with a lot of other procedures you won't need). Put these in your IDL path.

s15caldata.tar.gz contains the contents of the **S15CALDATA** directory. Put these in the appropriate directory.

The .dbd files are needed only if you would like to make an IDL database. These should be in a distinct directory.

The test data should be in an appropriate directory under **S15DATA**.

You will also need the Astronomy Users Library (http://idlastro.gsfc.nasa.gov) and the MPFIT procedures (http://cow.physics.wisc.edu/ craigm/idl/idl.html) in your IDL path.

3   Running the Software

This code has been run under IDL version 6 on LINUX (Fedora Core) and Mac OS-10.4

3.1   Data Preparation

The software acts either on data that have been previously downloaded, or it can download data into the appropriate directory for you.

You must have write access to your directory. The raw data are in subdirectories YYMMDD in your main directory. The reduced data will be written to the directory REDUCED/YYMMDD under the main directory. You must create the REDUCED directory manually; subdirectories are created automatically.

File names must be in the format *s15myymmdd.xxzz.fits*, where yymmdd is the 6 digit date, xx is the two digit target number, and zz is the 2 digit observation number. Target numbers are 00 for biases (zero images), 01 for flat images, and 10−99 for all other targets. Observation numbers start at 1 for each target, and generally run 1−4 or 1−5. The files may be gzipped.

3.2   The Main Reduction Program

The main routine is **s15_start**. Invoke it by typing
>**s15_start**,*date*
where *date* is the date of the observation in YYMMDD format. If the directory YYM-MDD does not exist it will be created. If there are no raw data present, the program will attempt to scp data files from the v17 account on ctio60.ctio.noao.edu. You will need to enter the ctio password and, of course, will need scp access to ctio60.

Any gzipped files are gunzipped at this point.

The program will use **s15targets** to display the target list. By default, the program pauses at this point so that the user can edit the fits header if required. Sometimes the service observers mistype target names or program IDs. For the software to work properly, program IDs must be correct. Check the grating name and tilt and the filter: these are used to determine the observing mode and the initial wavelength fit.

Use the procedure **updatefitshead** to edit the fits headers. If the header information is correct, or when done with editing the headers, type **.continue**.

The header information is checked at this point. If any vital information needed for the data reduction is missing, the program will stop and ask you to update the entries. The observing log is generated after this stage.

3.2.1 Data Reduction

The data processing occurs in **improc**. Data are stored in directory REDUCED/YYMMDD (the subdirectory YYMMDD will be created if it does not exist). At no point are the raw data files altered or overwritten, unless the header information is specifically edited.

- The zero image is created. The individual images are median filtered. The data are trimmed, and the overscan is fit and subtracted. The output is called *zero.fits*. The image is plotted to the terminal. If the median value of the zero image exceeds 10 counts, zero subtraction is disabled. Zero subtraction can also be disabled by setting the **/nozero** keyword in the call to **s15_start**.

- A flat file is created for each spectrograph setup. If there is a single setup, this file is called *flat.fits*, otherwise it is called *flatx.fits*, where x runs from 0 through $1 -$ the number of setups. The flat file is the median of the flat images in the given setup, bias-trimmed and overscan and zero-subtracted (if appropriate). The images are plotted to the terminal.

- All object and wavecal images are bias-trimmed, overscan and zero-subtracted (if appropriate), and flattened by dividing by the normalized flat (the median is set to unity). The suffix *_tzf* is appended to the target number (before the .fits extension). The "t" indicates the image is trimmed (to an 1199 x 260 array) and that the overscan is subtracted; "z" indicates that the image is zero-subtracted - it is omitted if the zero image is not subtracted; "f" indicates that the image is flattened.

- The first wavecal spectrum acquired for each setup is fit as the master wavecal. The fit will be displayed for acceptance (see Section 4).

A number of other reduction options are available in **improc**, but these are generally not useful for normal data reductions.

3.2.2 Spectral Extraction

Following data reduction, the spectra are extracted.

Use the **prog** keyword to select (by program ID) the spectra which are to be extracted. **prog** is the name of a particular program or institution matching the program ID. Program-matching is case-insensitive, and matches only the first n characters of the program ID, where n is the length of the **prog** keyword string. By default, all SUNY, STANDARD, and CAL program data are extracted. Other users should change this by editing **s15_start** to set

```
suny=0
prog='myprogram'
```
where 'myprogram' is the appropriate program ID, such as 'STSI'. You can also set these as keywords.

By default all cal targets (sky flats and spectrophotometric standards) are reduced; set the **/nocal** keyword to skip this.

The first step in the extraction is to generate the median image. If more than one image is obtained (in general, 3 are recommended), the images are median-filtered to reduce cosmic-ray contamination (if 2 are taken, they are averaged). I employ no sophisticated cosmic-ray filtering. These combined images are named *Txx.fits*, where xx is the target number.

Data extraction is done in four ways:

- The data are extracted from the median image *Txx.fits*. The output is called *XC_xx.fits*

- The data are extracted from each of the **_tzf.fits* files. These are then summed, and the output is writted to *X_xx.fits*

- Within each of the above, the data are extracted both by a simple boxcar extraction and with a Gaussian fitting of the spatial profile. The latter does a better job preserving flux; the latter does better with cosmic ray rejection and generally gives superior S/N. Gaussian extraction does not work well when there is another source nearby in the slit, since the model is a single source + background.

Prior to extraction, the spectrum is traced. Sky images use the adjacent spectrum for trace extraction, unless the **skytrace** keyword is set to a target number, in which case that target defined the trace to be used for sky extraction.

If the target is very weak, or not the strongest near the center of the spectral slit, the trace may not be found, or the wrong target may be extracted. In that case, you will have to do the extraction manually using **dos15extr** (see §6).

The width of the extraction boxcar is ±2.5 sigma from a Gaussian fit to the spatial profile near the center of the array.

Background is extract above and below the spectrum. In the case of the Gaussian extraction, the background is the baseline linear fit.

The center of the trace, the width of the extraction box, and the Y positions of the background, are all written to the fits header (keywords *TRCEN, TRWIDTH1, TRWIDTHx, and BACKGNDx*, where x=1-4).

These values are a slowly-varying function of X position.

The formats of the output data files are described in Section 9

4   Fitting the Wavelength Scale

Each target generally is observed along with a wavecal arc lamp. If not, no wavelength solution is done.

5

The initial wavelength solution is estimated from the grating and filter settings in the fits headers. If these are incorrect the solution will likely fail. Sometimes the grating tilt is set incorrectly. The extracted lamp spectrum is cross-calibrated against a fiducial spectrum to determine any offsets. So long as the shift is not too extreme ($<$ about 20% of the full spectral range), this seems to recover the solution. Fiducial spectra are defined for modes 26/Ia, 47/Ib, and 47/II. The spectral shift determined by this cross-calibration is flashed to the screen in a line that says:

<div align="center">MK_REFWAV: Bin shift = ...</div>

By default the wavelength solution for each target is displayed for examination (set **wcheck=0** to skip this). The upper panel shows the fit less the quadratic terms so you can see how much non-linearity exists. The lower panel shows the residuals in Angstroms between the tabulated line wavelengths and the fit wavelengths as a function of wavelength. The order of the initial fit, which is hard-wired, is shown in the plot title.

You can edit the line list, and refit the data. Options are:

- 0: restart the fit with the default parameters and all lines.

- C: clip the most deviant point.

- E: examine the profile of the line closest to the cursor. Move the cursor first, then type "E".

- P: make a hard copy plot of the fits and exit.

- Q: exit without saving the fit. The program could do strange things if you do this.

- R: refit the data. You will be asked to enter the order of the polynomial.

- S: restore all deleted points.

- X: delete the point closest to the cursor.

- Z: stop. You can muck around with the variables if you wish.

Any other key exits the editor without saving the plot, and lets the processing proceed.

You may remove as many lines as you want (but you must leave at least one more than the order of the polynomial fit). Sometime high order fits produce unacceptable ringing. Use the lowest order that produces an acceptable fit.

What constitutes a good fit? You don't want any single points that are away from the zero-line by much more than the average scatter. You certainly don't want any syatematic trends in the residuals. You would like the residuals to be less than about 0.1 pixels.

The spectrum will be linearized, and the solution (the intial wavelength and the reciprocal dispersion) are written to the fits header (keywords *WAVE_0* and *DELTA_W*.

Fits keywords *WAVE_0, WAVE_1, and W_COEFS* are the coefficients of the polynomial fit; the order of the fit is given by *NW_COEFS*.

This software has no provision for interpolating between wavelength solutions.

5 Flux Calibration

After all data are extracted, the calibration targets are examined to find any that are spectrophotometric standards. The standards currently defined are listed in Table 5.

Spectrophotometric Standards

Standard	RA	Dec	V	SpT
Feige 110	23 19 58.39	-5 09 55.8	11.83	DOp
Hiltner 600	6 45 13.33	+2 08 14.1	10.44	B1
LTT 4364	11 45 37.70	-64 50 25.1	11.50	C2
LTT 6248	15 39 00.02	-28 38 33.1	11.80	A
LTT 7987	20 10 57.38	-30 13 01.2	12.23	DA
HD 49798	6 48 04.64	-44 18 59.3	8.30	O6
μ Col	5 45 59.92	-32 18 23.4	5.17	O9V

Conversion from counts to flux is computed for each defined bandpass, and this is interpolated as a polynomial function of wavelength. This vector is stored in the extracted fits file.

You can do this flux calibration manually. Use **set_photcal** to determine the counts-to-flux conversion. You can check how well the final fit works by overplotting the fluxes on the "calibrated" data by using the **/verify** keyword. You can change orders, and fit the inverse (**/inv** fits counts vs. flux as opposed to flux vs. counts), or fit a fitted continuum rather than the data (**/fit**) until you find an acceptable solution. You will need to run **set_photcal** twice, once with the **/com** keyword set, to generate conversion for both the X_* and XC_* files.

setphotcal applies the nominal wavelength-dependent CTIO extinction correction based on the air mass before fitting the counts-to-flux conversion. When two or more standards are observed on the same night, the solution obtained closest in time is applied. The only way to skip a standard, if you decide not to use it, is to move or delete the corresponding X_* and XC_* files.

Note that these calibrations are generally not very trustworthy. There are variable slit losses depending on seeing variations and tracking, as well as changes in transparency through the night. In addition, there may be slopes induced when the standard or the target are observed at large zenith distances due to differential refraction. We do not change the slit orientation from E-W.

In addition, some very low resolution spectra, particularly with gratings 9 and 13, are generally taken without order-sorting filters. Fluxing red objects with blue

photometric standards may leave artifacts due to second order light in the standard star's spectrum.

6 Manual Data Extraction

The automatic extractions may fail for a number of reasons. The code assumes that the brightest object in the slit (excluding the extreme 50 pixels) is the target. This is not always the case. If the object is very faint, it may not be traced accurately. If the object is close to another star, boxcar extraction may include both stars, and the Gaussian extraction may give unpredictable results.

The procedure **dos15extr** gives the user some manual control over the extraction. The basic call (the one used in the automatic reduction) is **dos15extr,list**, where list is the list of target numbers to be extracted. The full call includes the following optional keywords:

cfile: The name of the wavecal file to use, if not observed with the target.

cutx: X location to automatically search for the spectrum, By default, $X = 600 \pm 20$ pixels. An emission line source may not have any flux here. Use this in conjunction with the **mark** keyword.

mark: set to 1 to interactively identify the spectrum. Set to 2 to do so on all spectra (useful if the spectra shift along the slit). In both cases, 20 columns (centered on the column specified by **cutx**) are coadded and plotted. Mark the peak with the cursor. A new window shows the automatically defined extraction region and background region. You can reset these if needed - follow the directions on the screen.

Set **mark** to 3 to mark the position of the source on the image (right mouse button click). If the plot is too faint, or saturated, with the default scaling, change the scale by setting keyword **med**. The default is 20 (counts above the median).

trace: Number of the target to be used as the trace. By default a self-trace is attempted. A trace file must be defined for an extended object (i.e., the sky). The trace will be adjusted in Y to match the data unless **/fixtrace** is set.

wcheck: Set to 1 to do a manual check of the wavelength solution (recommended).

7 Customizing the Reduction

There are a number of options that can be used with **s15_start**.

- set the **/a2** keyword to download from ctioa2 rather than ctio60 (useful if there is a problem with ctio60).

- By default, the program does not download data if any raw data exist in the YYMMDD directory. Set the **/force** keyword to re-download the raw data. Similarly, if a zero.fits or flat*.fits images exist in REDUCED/YYMMDD, these files will not be regenerated. Setting the **/force** keyword will overwrite this and force the zero and flat images to be regenerated.
  To reprocess the zero and flat images without re-downloading the data, set the **/force** and **/nodownload** keywords.

- If you merely want to download the data and generate a log, set the **/noprocess** keyword.

- To process data, but skip the extractions, set the **/noextract** keyword.

- To skip the interactive wavelength checking, set **wcheck=0**

- If you are having trouble with the automatic wavelength solutions, you can:

  – Override the observing mode by setting **mode=**_mode_, where _mode_ is a known observing mode (e.g., 26/Ia).

  – If the spectrum is shifted by a known number of pixels, setting **binshift=**n may help. However, if the shift is too large you will get an idl error for an out-of-bounds array element.

- Set the keyword **/skytrace** to the target number of a bright point source if the default sky trace (the first source of the night) is likely to be too weak to provide a good trace.

8 Known Bugs and Work-arounds

Sometimes there is a single file with a discrepant grating tilt or filter name. This will cause the program to think there are more grating setups than there really are. The program may get confused if it cannot find any observations for a setup. I have built in a number of traps to catch such cases, but there may be new failure modes. The only way to find these is to check the header parameters (e.g., using s15id), and then correcting erroneous entries.

If ARCON fails, the link between the data writing computer and the TCS may fail. You may have to enter these by hand when the program checks the headers. If incorrect but not absurd values are entered, there may be subtle errors in the airmass correction and the conversion from counts to flux.

If a trace fails, you may have to do that extraction and all subsequent reductions manually. But first, try returning to see if it will eventually get to the next target.

If there is no arc lamp spectrum associated with a target, there will be no wavelength solution. You can run **dos15extr** with keyword **cfile** set to the name of a wavecal spectrum to get a wavelength solution.

There are known bad columns. Some old ones are fixed in the software; more recent ones may not be. These can come and go.

9   Data Products

The three output files are *Txx.fits, X_xx.fits*, and *XC_xx.fits*. *xx* is the target number.

Txx.fits is the median image (it is not created in there is only one spectral image).

The extracted spectra are saved in two fits-format files. One file, named X_xx.fits (where xx is an integer), is the data extracted from each of the individual observations, then coadded. The XC_xx.fits file is the extraction from the median-filtered summed image.

The data are extracted in two ways.

- a boxcar extraction, with the slit width held constant. The background is the median of the background measured in boxes above and below the target. This can recover all the source counts, at the expense of some noise where the source is low relative to the background.

- a Gaussian extraction. The net counts are the integral of the fit Gaussian. The width is a free parameter. The background is the interpolation of a quadratic background to the center of the target trace. This extraction does a better job at cosmic ray rejection, but it tends to underestimated the true counts, because the point source prpfile has broader wings than a Gaussian.

The fits file encodes a single n x m array, where n is the number of points in the spectrum (generally 1199), and m is the number of extracted vectors. m is between 4 and 9.

The data include the following records, in this order:

- The net boxcar-extracted spectrum counts

- The boxcar-extracted background counts (if background is extracted)

- The uncertainty on the net boxcar-extracted spectrum counts

- The uncertainty on the boxcar-extracted background counts (if background is extracted)

- The net Gaussian-extracted spectrum counts

- The Gaussian-extracted background counts

- The uncertainty on the net Gaussian-extracted spectrum counts

- The flux conversion factor for the boxcar-extracted spectrum counts

- The flux conversion factor for the Gaussian-extracted spectrum counts

Multiply the net extracted counts spectrum by the flux conversion factor for a flux calibrated spectrum. The absolute scaling depends on the weather, but the shape of the spectrum should be accurate.

The wavelength scale is linear, and is stored in the header words wave_0, the initial wavelength in Angstroms, and delta_w, the reciprocal dispersion in Angstroms per pixel.

10   Stand-alone Procedures

A number of the procedures can be used stand-alone for simple data examination.

ps15: *Procedure to plot and or extract the spectra.*

```
* PS15 - plot extracted 1.5m RC spectra
* calling sequence: PS15,file,w,sp
*    FILE: name of file or target number
*    W, SP: output wavelength and spectrum vectors (if W and SP are
*           specified a plot is produced only if \plt is specified
*
* KEYWORDS:
*    COMB:   set to extract from combined image (default)
*    COUNTS: set to plot counts spectra
*    FCOR:   set to apply PHOTCOR if in header
*    FLUX:   set to plot photometrically-corrected spectra (def)
*    GAUSS:  set to plot Gaussian extraction
*    HEADER: equate to named variable to return fits header
*    LAMBDA: wavelength range to be plotted
*    OPBG:   set to overplot background
*    OPEB:   set to overplot error bars
*    PIX:    set to plot pixels on the X axis, def = wavelength
*    SCAN:   loop though all spectra in directory (set if FILE is unspecified)
*    SM:     smoothing box size
*    SUMMED: set to extract from individually summed spectra
*    YLOG:   set to plot logarithmic Y axis
*    YRANGE: set to truncate spectrum (sp>yrange(0))<yrange(1)
```

ps15im: *Procedure to plot the 2-dimensional spectral image.*

```
* PS15IM - plot 1.5m spectral images
* calling sequence: PS15IM,file,w,sp
*    FILE: name of file or target number
*    D, H: output image array and header
*
```

```
*  KEYWORDS:
*     COMB:   set to extract from combined image
*     HEIGHT: optional height for plot
*     LAM:    optional wavelength range
*     LINEAR: set for linear plot scaling, def=logarithmic
*     MARK:   set to use TVP to check plot pixel values
*     MAXVAL: maximum array value for plot scaling
```

The wavelength scale is approximate. The data in the image are not resampled to a linear scale - that happens during extraction. The polynomial fit for the wavelength scale is stored in the fits header./

s15id: *Generate a brief listing of all the observations of a target or an individual observation (by observation number or filename).*

```
* S15ID - listing of all the observations of a target or obervation
* Calling sequence: S15ID,INP,Z
*    INP: target number, file name or fits header vector
*    Z:   output string ID
*
* KEYWORDS:
*    DO:  set to set root to default data directory
*    DIR: name of directory, default = current
*    RAW: set to search raw data; def searches REDUCED directory
```

s15targets: *List the targets in the directory.*

```
* S15TARGETS - list targets in directory
* calling sequence: S15TARGETS,list,targets
*    LIST: list of target numbers (output)
*    TARGETS: list of target names (output)
*
* KEYWORDS
*    BRIEF: set for brief listing
*    DIREC: directory to search, def=current directory
*    ENG:   set to list some engineering data
*    FULL:  set to list more information (default)
*    NOCAL: set to skip calibration targets
*    OUT:   name for output file, def=yymmdd.targets
*    PROC:  set to list processed files only
*    PROG:  program ID to list, def=all
*    SUNY:  set to list only SUNY and CAL observations
```

updatefitshead: *Use to edit fits headers.*

```
*  UPDATEFITSHEAD - change values in FITS headers
*  calling sequence: UPDATEFITSHEAD,files
*     FILES: list of files or target numbers
*
*  KEYWORDS:
*     FIELD:  fits field to be updated
*     OBJECT: equivalent to FIELD="object",VALUE="xxx"
*     MODE:   set to update grating, tilt, and filter for defined modes
*     PROPID: equivalent to FIELD="propid",VALUE="xxx"
*     REMOVE: set to delete keyword from header
*     VALUE:  new value for field
```

11   Making a Database

The IDL Astronomy User's Library has a useful set of databasing procedures. If you
have lots of data, organizing it into a database can simplify searched and data retrieval.

The procedures *make_s15db* and *make_s15pdb* can be used to create data bases from
the fits headers. I organize these by years. *make_s15db* is used to make a database
for the raw (unprocessed) data, *make_s15pdb* does the same for the processed and
extracted data.

The .dbd files you will need are in the distribution.

You can use the procedure *find_targets* to query the database.

```
* FIND_TARGETS - find all targets in s15p_* db matching specifications
* calling sequence: FIND_TARGETS,file,object
*    FILE:   output file names
*    OBJECT: name of target.
*       if only one parameter is passed, it is assumed to be OBJECT.
*
* KEYWORDS:
*    NAME, PROPID, MODE: specifiers
*    OBJNAME:  object name (output)
*    SEMESTER, YEAR: set to specify database, def= 2009
*    DATE:     set to print UT date rather than file name
*    XFILE, TFILE: set to return X_ or T file, def = XC_ file
*    HC:           print hard copy to file dbprint.prt
```

12   Procedures to Facilitate Data Access

The output data are in fits format. While fits is good, it is not perfect, and there are
cases when different output formats may be desirable. For example, many IRAF users
cannot read these fits files because they are not in the format expected by splot.

Use the procedure **ascii_spectrum** to generate a formatted ascii table.

```
* ASCII_SPECTRUM - write out 1.5m RC spectrum in ASCII format
* calling sequence: ASCII_SPECTRUM,id
*    ID: ID number in directory or file name
*
* KEYWORDS:
*    FILE:  file name, if not in s15 format
*    GAUSS: set to extract the Gaussian-extracted spectrum
*    OUT:   name of output file, def=target name.ascii
*    DO:    set if not using current directory
*    DIREC: Directory name
```

The output looks like

```
# CSS0810070305                              ;target name
# 2008-11-06T03:48:19.8                      ;UT at start
#
#      Wavelength        Flux    Uncertainty        Counts   Uncertainty
#
        3653.7575  8.01059e-16  2.67040e-16       26.0444      8.68210
        3655.2346  5.88581e-16  2.62005e-16       19.2235      8.55727
        3656.7117  1.07509e-15  2.83688e-16       35.2737      9.30784
        3658.1888  2.68070e-16  2.37662e-16        8.83576     7.83349
        3659.6659  9.60199e-17  2.48685e-16        3.17942     8.23448
        3661.1430  1.25024e-15  2.83085e-16       41.5887      9.41672
                .            .            .             .            .
                .            .            .             .            .
                .            .            .             .            .
```

If you use my **ICUR** spectral analysis software, you can use **make_icd** to translate fits files to .icd format. **make_icd** works with **find_targets**, so you can search by target name (including widl cards), program ID, or observing mode. You can exclude files with specific strings in the target name.

```
* MAKE_ICD - make ICD file from SMARTS spectra
* Calling sequence: MAKE_ICD
*
* KEYWORDS:
*    COMB:     set to 0 for coadded spectra; def=median image.
*    COUNTS:   set to extract count spectrum, def=flux
*    DIREC:    if set, create this directory and output to it
*    EXCLUDE:  string array of names to exclude
*    GAUSS:    set to extract Gaussian spectrum, def=boxcar
*    ICD:      name of output .icd file
*    MODE:     if set, find all targets with this mode
*    NAME:     if set, find all targets with this name
```

```
*    NEWFILE:  set to overwrite existing icd file
*    NOJDSORT: if set, files are not sorted by observation time
*    OBJECT:   if set, find all targets with this name (same as NAME)
*    PROPID:   if set, find all targets with this propid
*    RECS:     records to store (max of 480)
*    XFILE:    set to return  Xfile, def=XCfile
*    Y2003, S4M, S2003A,S2003B, etc: database to search, def=all
```