Wide-field Infrared Survey Explorer (WISE)

Tempcal Subsystem Design Document

Version 1.0

28 July 2008

Prepared by: John W. Fowler & Frank. J. Masci

Infrared Processing and Analysis Center
California Institute of Technology

WSDC D-D011

Concurred By:

Roc Cutri, WISE Science Data Center Manager

Tim Conrow, WISE Science Data Center System Architect

John Fowler, WISE Science Data Center Tempcal Cognizant Programmer

Frank Masci, WISE Science Data Center Tempcal Cognizant Engineer

Revision History

Date	Version	Author	Description
28 July 2008	1.0	J. W. Fowler & F. J. Masci	Initial Draft

Table of Contents

1 Introduction

1.1 Subsystem Overview

This document presents the requirements, design, algorithms, and state of implementation of the Tempcal (Temporary Effects Calibration) subsystem of the WSDC data processing system. Tempcal runs offline on all or part of a single scan.

1.1.1 Requirements

Tempcal is required to compute a pixel 'sky-offset' image and to identify and flag *persistent* new "bad" pixels in a stack of *N* consecutive frames along a scan. This is the essence of dynamic pixel masking; tempcal updates FITS mask images by turning on bits in the pixel data to indicate conditions that it diagnoses. The Level 4 requirements supported by this processing are as follows.

> L4WSDC-012: Flux measurements in the WISE Source Catalog shall have a SNR of five or more for point sources with fluxes of 0.12, 0.16, 0.65 and 2.6 mJy at 3.3, 4.7, 12 and 23 micrometers, respectively, assuming 8 independent exposures and where the noise flux errors due to zodiacal foreground emission, instrumental effects, source photon statistics, and neighboring sources (*traceable to Level-1*).
>
> L4WSDC-013: The root mean square error in relative photometric accuracy in the WISE Source Catalog shall be better than 7% in each band for unsaturated point sources with SNR>100, where the noise flux errors due to zodiacal foreground emission, instrumental effects, source photon statistics, and neighboring sources. This requirement shall not apply to sources that superimposed on an identified artifact (*traceable to Level-1*).
>
> L4WSDC-024: The WSDC shall generate and maintain an archive of the calibrated, single epoch WISE images for the duration of the project for use by the Project Team. The purposes of this archive are quality assurance, transient analysis and moving object identification. Self-derived Demonstration Define duration of project.
>
> L4WSDC-037: The WSDC Pipelines subsystem shall convert raw WISE science and engineering data into calibrated images and extracted source lists from which the preliminary and final WISE data products will be derived.

L4WSDC-039: Within 3 days from receipt of a given data set at the WSDC all data shall be processed through the WSDS Scan/Frame pipeline which performs basic image calibration and source extraction from on images from individual orbits. The results of this processing step shall be Level 1 source extractions and image data, which are loaded into the WISE Level 1 extracted Source Working Database (L1WDB) and Image Archive allowing access by the WISE Science Team for external quality assessment.

L4WSDC-042: The WSDS Pipeline processing shall remove the instrumental signature from Level 0 image frames.

L4WSDC-062: The WSDC shall perform quality analysis of all WISE science data and make reports available on a regular basis.

The tempcal module is run offline on a list of preselected survey data images. Input frames will have already been precalibrated, e.g., dark subtracted, linearized and flat-fielded. The units of the pixel values are native WISE "*scaled*-slope" units as computed onboard for detector ramps. More specifically, they will be in units of *scaled* DN/SUR, where DN means Data Number and SUR means Sample Up the Ramp.

Short-term variations in the bias, dark (and possibly gain) structure over an array will not be captured by ground calibrations. The ground calibrations are designed to remove instrumental signatures that are essentially static in the long term. If the short-term systematic variations are not removed, they will persist as residuals and impact photometric accuracy. These can be corrected by computing a robust estimate of a zero-mean (or zero-median) background image from $N \sim 50$ - 100 frames within a moving block window along the WISE orbit, then subtracting this from all the frames in that window. The tempcal module will create the sky-offset image calibration product only, not apply it. It is estimated that at least 50 - 100 frames will be needed to filter out sources reliably across all bands. A window that's too big may miss the short-term instrumental variations sought for. An important assumption is that the transient bias/dark structure is approximately constant over this window span. A schematic of the concept is shown in Figure 1.

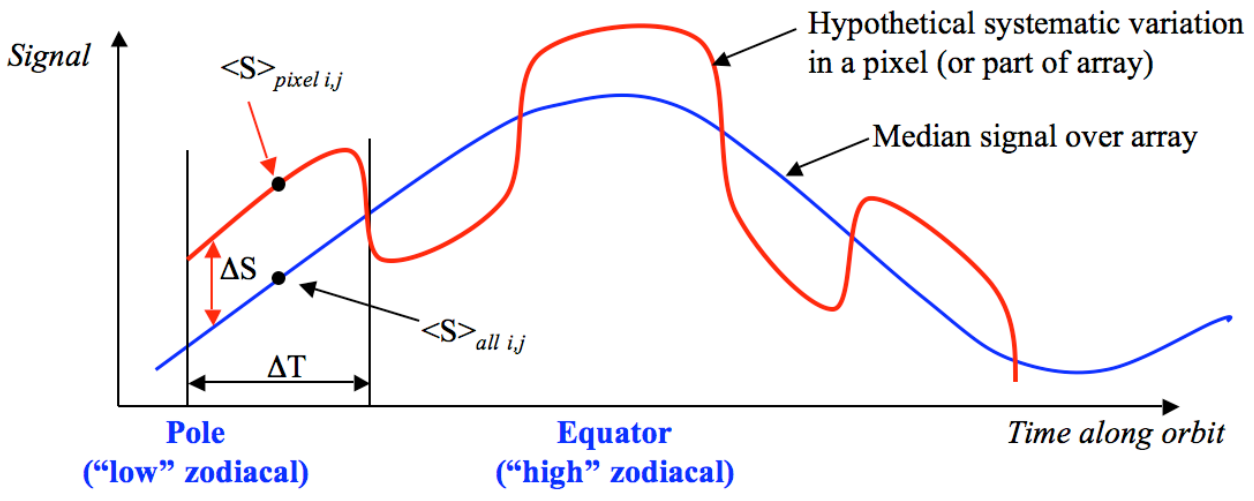


Figure 1. Sky-Offset Schematic

The labels in Figure 1 are defined as follows:

> ΔT = timescale of possible systematic variation
> $<S>_{pixel\ i,j}$ = median or mean sky signal in single pixel *i*, *j* over stack of *N* frames in ΔT
> $<S>_{all\ i,j}$ = median or mean sky signal over all pixels and *N* frames in ΔT
> Sky-offset correction : $\Delta S_{i,j,\Delta T} = <S>_{pixel\ i,j} - <S>_{all\ i,j}$

The number of frames must satisfy : $N_{\mathrm{min}} \leq N \leq N_{\Delta\mathrm{T}}$, where N_{min} is the minimum needed to filter out stars. If $N_{\Delta\mathrm{T}} < N_{\mathrm{min}}$ (fast instrumental variations), then this method can't be used. The plan is to determine optimal frame windows N_{min} and $N_{\Delta\mathrm{T}}$ in IOC.

Note: the angled brackets can represent either a *trimmed mean*, *median* or whatever robust estimator is used to estimate the sky background seen by the pixel in the *N*-frame stack.

This same module will also perform dynamic pixel masking. The rationale behind this is that it is the only place in the infrastructure (so far) that gathers all frames pertaining to the same time interval, ΔT, along a scan. As discussed above, this time-interval may contain 50 - 100 frames, or $\lesssim$40% of a scan (North-to-South ecliptic pole). If a pixel suddenly becomes hot, it may persist in this state for the entire interval ΔT, or just part of it for a duration δ*t*. Bad pixels (where the criteria for 'bad' are defined below) occur systematically at the same location in pixel space, whereas astronomical sources do not. One can therefore envisage a method where if a pixel is detected as an outlier with respect to its neighbors in the *same* frame, and it persists in this state for a time δ*t* in subsequent frames of the stack, then it can be identified as a transient bad pixel. Pixels identified as permanently bad *a priori* (e.g., on the ground) are omitted before performing the 'dynamic' bad-pixel search.

1.1.2 Liens

- Low priority: implement an even more robust frame "offset" (and uncertainty) estimation algorithm
- Implement latent tagging using a prior model ("model" may be a mathematical model such as exponential decay, or it may be a behavioral model such as simply monotonic decrease from the first transient sample over a specified number of time-consecutive samples in the transient interval, or any other acceptable approximation to latent behavior)
- Avoid tagging the first sample if transient pixel is a latent
- Add the "FRMIDSEQ" keyword (see section 4.1) to output sky-offset FITS file headers after frame ID is included in input FITS headers by the ingest subsystem

1.2 Applicable Documents

This subsystem conforms to the specifications in the following project documents:

- WISE Science Data Center Functional Requirements Document, WSDC D-R001
- WISE Science Data System Functional Design, WSDC D-D001
- Software Management Plan, WSDC D-M003
- Instrumental Calibration Plans and Considerations: http://web.ipac.caltech.edu/staff/fmasci/home/wise/SingleOrbit_Cal.html
- Infrastructure and Instrumental Calibration Scan Pipeline: http://web.ipac.caltech.edu/staff/fmasci/home/wise/ScanPL_instrumental_cal.pdf
- Software Interface Specification (ICL01) Frame Processing Status Mask: http://web.ipac.caltech.edu/staff/fmasci/home/wise/InstruCal01.txt
- Software Specifications for the *tempcal* module: http://web.ipac.caltech.edu/staff/fmasci/home/wise/tempcal_specs.pdf

1.3 Acronyms

2MASS	Two-Micron All-Sky Survey
DN	Data Number
FITS	Flexible Image Transport System
FRD	Functional Requirements Document
IOC	In-Orbit Checkout
NAXIS1	Number of columns in an image
NAXIS2	Number of rows in an image
Nframes	Number of science images to be processed
SDS	Subsystem Design Specification
SIS	Software Interface Specification
SUR	Sample Up the Ramp
W1	WISE wavelength channel 1, 3.3 microns
W2	WISE wavelength channel 2, 4.7 microns
W3	WISE wavelength channel 3, 12 microns
W4	WISE wavelength channel 4, 24 microns
WISE	Wide-field Infrared Survey Explorer
WSDC	WISE Science Data Center
WSDS	WISE Science Data System

2 Input

2.1 Control Input

Tempcal reads control input in the form of Fortran Namelist files and command-line parameters. For control parameters included in both command line and namelist inputs, the command line inputs override.

2.1.1 Tempcal Command-Line Parameters

The command-line parameters for Tempcal are given by its tutorial display:

```
tempcal version: 1.25 A80725 - execution begun on 28-07-08 at 13:13:39

Usage: tempcal

 -f1 <inp_img_list_fname>  (Required; list of pre-calibrated frames in
                            FITS format)

 -f2 <inp_mask_list_fname> (Optional; list of bad-pixel masks in 32-bit INT
                            FITS format; only values 0 -> 2^31 are used)

 -f3 <inp_unc_list_fname>  (Optional; list of uncertainty images in FITS
                            format)

 -lt <lower_threshold>     (Optional; lower-tail threshold for in-frame
                            outlier [candidate bad-pixel] detection;
                            Default = 5 sigma)

 -ut <upper_threshold>     (Optional; upper-tail threshold for in-frame
                            outlier [candidate bad-pixel] detection;
                            Default = 5 sigma)

 -pn <frame_persist_num>   (Optional; minimum number of consecutive frames
                            in time-ordered sequence of stack for which an
                            'outlier pixel' must persist to be declared bad;
                            Default = number of frames in input list)

 -m <inp_mask_bits>        (Optional; mask template [decimal] specifying
                            bits to flag/omit from processing; Default = 0)

 -p <out_maskdy_bits>      (Required if <-f2> specified; mask template
                            [decimal] specifying bit to set in _specific_
                            input masks for dynamic bad-pixel masking)

 -tf 0                     (Optional; turn off transient flagging)

 -s <out_maskso_bits>      (Required if <-f2> specified; mask template
                            [decimal] specifying bit to set in _all_ input
                            masks for unreliable/erroneous sky-offset)

 -o1 <out_skyoff_img>      (Required; output sky-offset image FITS filename)

 -su <out_maskso_bits>     (Required if <-f2> specified; mask template
                            [decimal] specifying bit to set in _all_ input
                            masks for unreliable uncertainty in sky-offset)

 -pl <out_maskso_bits>     (Required if <-f2> specified; mask template
                            [decimal] specifying bit to set in affected
                            masks for probable latent contamination)
```

```
-o2 <out_skyoff_unc_img>  (Required; output sky-offset uncertainty image
                           FITS filename)

-o3 <out_chi-square_img>  (Required; output sky-offset uncertainty image
                           FITS filename)

-o4 <out_Nused_img>       (Optional; output image of #pixels used in stack;
                           FITS filename)

-n  <namelist>            (Optional; namelist file name)

-d                        (Optional; switch to print debug statements
                           to stdout, ancillary QA to ascii files)

-v                        (Optional; switch to increase verbosity to stdout)
```

2.1.2 Tempcal Namelist Parametes

The Tempcal module optionally reads a NAMELIST file. The name of this file must be given on the command line via the "-n" option. The name of the NAMELIST is tmpcalin. The parameters defined in the NAMELIST are as follows.

Name	Description	Dim	Type	Units	Default
ChiSqMax	Maximum reduced chi-square of sky offset (if prior uncertainties given) or ratio of dynamic range to sky-offset uncertainty (if no prior uncertainties) *not* to flag sky-offset uncertainty as potentially unreliable	1	R*4	-	3.0
ITimSlic1	Lower column number for time-slice image output (see section 4.6)	1	I*4	-	0
ItimSlic2	Upper column number for time-slice image output (see section 4.6)	1	I*4	-	0
JTimSlic1	Lower row number for time-slice image output (see section 4.6)	1	I*4	-	0
JtimSlic2	Upper row number for time-slice image output (see section 4.6)	1	I*4	-	0
Mask	Mask template [decimal] specifying bits to omit from processing	1	I*4	-	0
MinPersist	Minimum number of conse-cutive frames in time-ordered sequence of stack for which an 'outlier pixel' must persist to be declared bad	1	I*4	-	Nframes
MinPix	Minimum number of pixels In a stack for the stack to be usable for robust estimation	1	I*4	-	5
NamWrt	If T, namelist will be written to stdout	1	L*4	-	F

Name	Description	Dim	Type	Units	Default
SkpST	If T, sky-offset computation will be skipped (only transient flagging will be done)	1	L*4	-	F
SubOff	If T, each value for a given pixel will have the corresponding frame offset subtracted before the pixel’s sky offset is computed; if F, this is not done, and instead the global frame offset is subtracted from the pixel’s absolute offset to obtain the pixel’s sky offset	1	L*4	-	F
ThrshHi	Positive offset in sigma units from frame offset limiting non-outlier pixel range	1	R*4	-	5.0
ThrshLo	Negative offset in sigma units from frame offset limiting non-outlier pixel range	1	R*4	-	5.0

2.2 ASCII Input

Tempcal reads one to three ASCII (text) file lists of FITS file names corresponding to FITS images. One is required, and its name is specified on the command line via the “-f1” flag; this is the list of names of science images. If the “-f2” flag was used, then a list of mask images is also read, and if the “-f3” flag was used, then a list of uncertainty images is read. These last two lists must be in one-to-one correspondence with the first list, i.e., the n^{th} mask image and the n^{th} uncertainty image must correspond to the the n^{th} science image. Each list is read, the number of lines is counted, and all must have the same number of lines; this number is used for memory allocation. Then the lists are rewound, each line is read, and the corresponding file is read into memory. The file name on each line must be left-justified and is case-sensitive. Path names may be included and are required if the FITS files are not in the working directory.

2.3 FITS Input

For each line in each ASCII file described in section 2.2 above, tempcal reads the FITS file whose name is given. All FITS images must have the same values for the following header parameters: NAXIS (must be 2), NAXIS1, NAXIS2, and BAND. Each frame must have its own parameter UTCS_OBS, which is used to force time order in the frame stack.

3 Processing

3.1 Initialization

The tempcal module initializes itself by:

A.) reading and processing its control inputs;

B.) verifying that all required inputs were given;

C.) reading the lists of FITS files and allocating memory;

D.) reading in all specified FITS images;

E.) sorting an index array of UTCS_OBS values.

3.2 Sky-Offset Computation

The tempcal module performs sky-offset computation using all unmasked pixels unless this function was deselected (see `SkpST` in section 2.1.2). If no mask files were specified, then all pixels are treated as unmasked.

3.2.1 Frame Offset Computation

The first step is to compute a "standard offset" for each science image; by "standard offset", we mean a robust estimate of the representative sky background, the median signal that would be observed if there were no point-like objects or radiation hits. This will be called simply the "offset" for brevity, and this term will be applied to entire frames (spatial distribution of signal) and also to pixel stacks containing all unmasked samples of a given array pixel (temporal distribution of signal seen by the pixel). We use the term "offset" to avoid the more specific terms such as "median", "trimmed average", etc., which imply specific algorithms that may not be in use. The tempcal code is modularized to facilitate installing different robust estimation algorithms for the "offset" of frames and pixel stacks; this is designed to allow different methods to be tried, and any new accepted variations will have a corresponding version of this SDS.

During initialization, all frames to be processed were loaded into a three-dimensional data cube in memory. The amount of memory required for this is `4*NAXIS1*NAXIS2*Nframes` for the science images, and the same each for optional uncertainty images and optional mask images. For W1, W2, and W3, `NAXIS1` = `NAXIS2` = 1016, so for these science images, the amount of memory required in bytes is 4.12904e6*`Nframes`. For W4, `NAXIS1` = `NAXIS2` = 508, so for W4 science images, the amount of memory required in bytes is 1.032256e6*`Nframes`.

An example of a minimal processing run would be 50 W4 frames with no masks or uncertainties. The total input-image memory for this case is 51.6 MB. An example of a typical expected case would be 100 W1 frames including science, uncertainty, and mask images. The total input-image memory for this case is 1.24 GB.

A loop over all frames is executed, and in this loop, the following processing is performed on each frame:

- A one-dimensional stack is loaded with all pixel values that are neither masked nor NaN.
- If the number of values in the stack is less than `MinPix` (see section 2.1.2), an offset value of NaN is returned; otherwise processing continues.
- The stack is sorted in ascending value, and the median value is found.
- The standard deviation of the lower 50%-tile about the median is computed, σ_{50}.
- All values in the stack that are either less than median–`ThrshLo`*σ_{50} or greater than median+`ThrshHi`*σ_{50} are compressed out of the stack (see section 2.1.2).
- The remaining stack is re-sorted into ascending order, and the new median is found; this is the frame offset.
- The standard deviation about the offset is computed for the values in the compressed stack, σ.
- Frame outlier limits are computed and returned for storage and later use; ; if `SubOff` is T (see section 2.1.2), then these are the lower limit FramLo = -`ThrshLo`*σ and FramHi = `ThrshHi`*σ, otherwise they are FramLo = offset - `ThrshLo`*σ and FramHi = offset + `ThrshHi`*σ.

After all frame offsets have been computed, all values that are not NaN are put into a stack, sorted, and the median is computed; this is the global frame offset that will be subtracted from each pixel's offset to obtain the pixel's sky offset if `SubOff` = `F` (see section 2.1.2).

3.2.2 Pixel Offset Computation

A loop over all array pixels is performed, and inside this loop each pixel is processed as described below.

- The pixel's value in each science frame in the data cube is examined; if no uncertainties were input, then if the pixel value is not NaN and not masked, the value is put into a pixel stack; if uncertainties were input, then these same conditions must be met by the science and uncertainty values, and in addition the uncertainty must be greater than zero (including not NaN); if and only if these conditions are met, the science data value and the uncertainty value are both placed into separate stacks; values accepted for a stack are placed there in increasing time order (the pixels are accessed via the index array of sorted UTCS_OBS values), and another index array is filled which specifies from which frame number in the data cube the pixel came; if `SubOff` is T (see section 2.1.2), then each science pixel's corresponding frame offset is subtracted from the science dta value as this value is loaded into the stack.
- If the number of values in the stack is less than `MinPix`, values of zero are returned for the sky offset and uncertainty, and if mask inputs were given, the pixel's value in all input masks has bits set for unreliable/erroneous sky offset and uncertainty; if the number of values in the stack is greater than or equal to `MinPix`, processing continues.
- The time-ordered pixel stack is copied into a temporary buffer which is sorted into ascending order, and the median is found.
- The standard deviation of the lower 50%-tile about the median is computed, σ_{50}.
- All values in the stack that are either less than median–`ThrshLo`*σ_{50} or greater than median+`ThrshHi`*σ_{50} are compressed out of the stack.
- A new median is found, the median of the remaining stack; this is the pixel's "sky offset" if the frame offsets have already been subtracted off (`SubOff` = T), and otherwise it is the pixel's absolute offset, and the global frame offset is subtracted from it to obtain the pixel's sky offset.
- The uncertainty of the sky offset is found as follows; if pixel uncertainties were input, then the uncertainties corresponding to pixels retained for the sky offset computation (median after outlier rejection) are used to compute an inverse-variance uncertainty for the sky offset; if no pixel uncertainties were input, then the sample variance about the offset is computed for the N values in the compressed stack, divided by N-1, and the square root is used as the uncertainty; in either case, the uncertainty is then scaled by $\sqrt{\pi/2}$ to account for the additional uncertainty of the median.
- Sanity checking of the pixel sky offset is performed as follows; if pixel uncertainties were input, then chi-square is computed and the reduced chi-square value must be less than `ChiSqMax` (see section 2.1.2); if pixel uncertainties were not input, then the dynamic range of the compressed stack is divided by the uncertainty of the sky offset, and the ratio must be less than `ChiSqMax`; in either case, if the condition is not satisfied and if masking is being performed, then the bit indicating an unreliable/erroneous sky-offset uncertainty is set for the pixel in all input masks.

3.3 Transient Pixel Identification

Unless transient identification was deselected (no masks input or "-tf" was specified on the command line), the following processing is performed in the same loop over pixels as the above section.

The full pixel stack that was prepared as described in section 3.2.1 above is examined in time order. Runs of consecutive pixel values that are all greater than FramHi or less than FramLo (see section 3.2.1) are identified. If the number of such consecutive pixels is at least as large as `MinPersist` (see section 2.1.2), and if mask inputs were given, then the corresponding mask pixels have bits set to indicate transient behavor. If the run began with the first pixel in the stack or ended at the last pixel in the stack, then it need not be as long as `MinPersist`; it need be only at least half as long as `MinPersist`. If any part of the stack was flagged as transient, then the entire mask stack for that pixel has the bit set indicating unreliable sky offset.

3.4 Latent Image Identification

TBS

4 Output

4.1 Sky-Offset FITS Output

The main output ("-o1" command-line parameter) is a FITS file containing the sky-offset image. An example header is below. The angle brackets indicate information whose literal content depends on actual execution or generation circumstances.

```
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                  -32 / number of bits per data pixel
NAXIS   =                    2 / number of data axes
NAXIS1  =                 1016 / length of data axis 1
NAXIS2  =                 1016 / length of data axis 2
BAND    =                    1 / WISE band number (1, 2, 3 or 4)
NUMINP  =                   70 / Number of input frames used
UTCSBGN =           1260864418 / Earliest UTCS in frame stack [sec]
UTCSEND =           1261051979 / Latest UTCS in frame stack [sec]
FRMIDSEQ=         <'Num..Num'> / Range of frameIDs used
COMMENT  Generated by tempcal vsn 1.25 A80715 on 15-07-08 at 11:46:58
COMMENT  This is a sky offset image
```

Notes:

- BITPIX = -32 refers to single precision floating point.
- For bands 1, 2 and 3, the input frames will have NAXIS1 = NAXIS2 = 1016. For band 4, NAXIS1 = NAXIS2 = 508.
- The UTCSBGN, UTCSEND keywords specify the start/end observation time of frames in the input ensemble. These time tags will be available in the input frame headers [e.g., UTCS_OBS].
- The FRMIDSEQ specifies the range of input frame IDs: a string composed of the *min* and *max* ID delimited by two dots, e.g., '31412..31505' (note: currently the frame ID is not included in the header by the ingest subsystem; it is embedded in the file name, but the WISE system engineer advises against parsing the file name to obtain frame ID on the basis that file naming may vary under certain conditions; once the ingest subsystem is upgraded to include the frame ID in the FITS headers, this keyword will be added).

4.2 Sky-Offset Uncertainty FITS Output

If the "-o2" command-line parameter was specified, a FITS file containing the sky-offset uncertainty will be generated. An example header is given below.

```
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                  -32 / number of bits per data pixel
NAXIS   =                    2 / number of data axes
NAXIS1  =                 1016 / length of data axis 1
NAXIS2  =                 1016 / length of data axis 2
COMMENT  Generated by tempcal vsn 1.25 A80715 on 15-07-08 at 11:46:58
COMMENT  This is a sky offset uncertainty image
```

4.3 Mask Updates

If mask inputs and bit numbers are specified, then any of the four conditions diagnosed by tempcal result in updates to these masks. The bit numbers are specified in terms of their decimal equivalents, not bit addresses; in other words, if bit 3 is to be set (where the WISE standard for bit numbering assigns zero to the least-significant bit), then this is specified as the value 8 (2^3). If any given mask bit is desired not to be set despite its condition being diagnosed, a value of zero may be specified for the mask bit value on the command line The conditions diagnosed and command-line specification flags are:

- Transient behavior, “-p”
- Possibly unreliable sky offset, “-s”
- Possibly unreliable sky-offset uncertainty, “-su”
- Probable latent, “-pl” (note: the second character is a lower-case “L”, not a numeral “one”)

If a sky offset is considered possibly unreliable, its uncertainty always is also. A sky offset may be considered reliable, however, and still have a possibly unreliable uncertainty (e.g., when the outlier-rejected stack has a significant slope with respect to time). These bits are set in every mask for every input science image.

When transient behavior is diagnosed, only the masks corresponding to the frames in which transient behavior is believed to exist have the bit set for this condition. The same is true for latent-image tagging, with the sole exception that the first image is not tagged, because it is believed to be a measurement of the true source that caused the latent image (note: latent tagging is currently a lien).

Note that the input masks themselves are updated. Making backup copies of the masks prior to execution is advised.

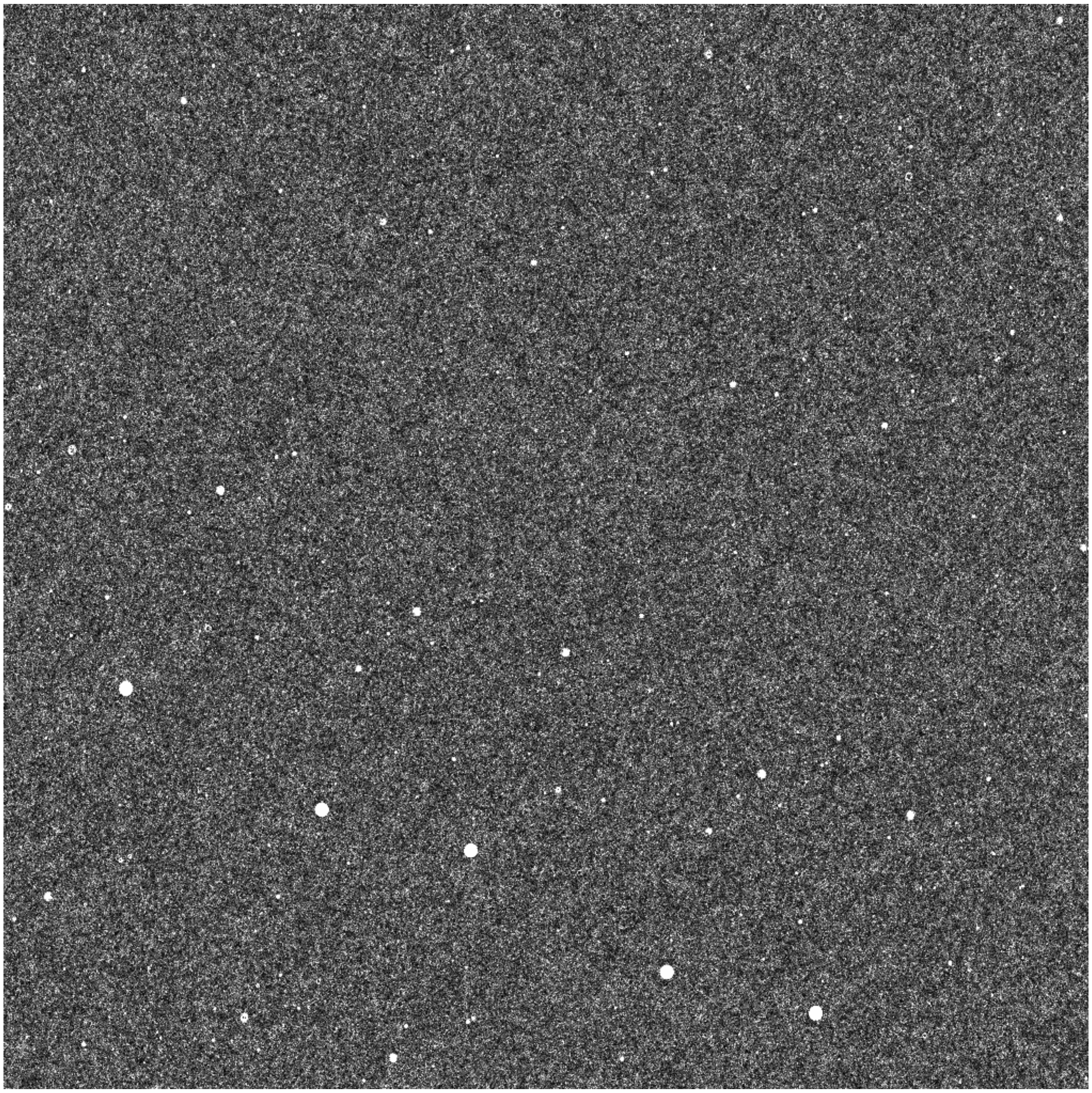
4.4 Optional Chi-Square FITS Output

If the “-o3” command-line specification is given, and if sky-offset computation has not been deselected, and if uncertainty images have been given, then the chi-square image is generated. This is the reduced chi-square for every unmasked pixel described in section 3.2.

$$\chi^2_{ij} = \frac{1}{N_{ij}} \sum_{n=1}^{N_{ij}} \frac{\left(p_{ijn} - s_{ij}\right)^2}{\sigma^2_{ijn} - \sigma^2_{s_{ij}}}$$

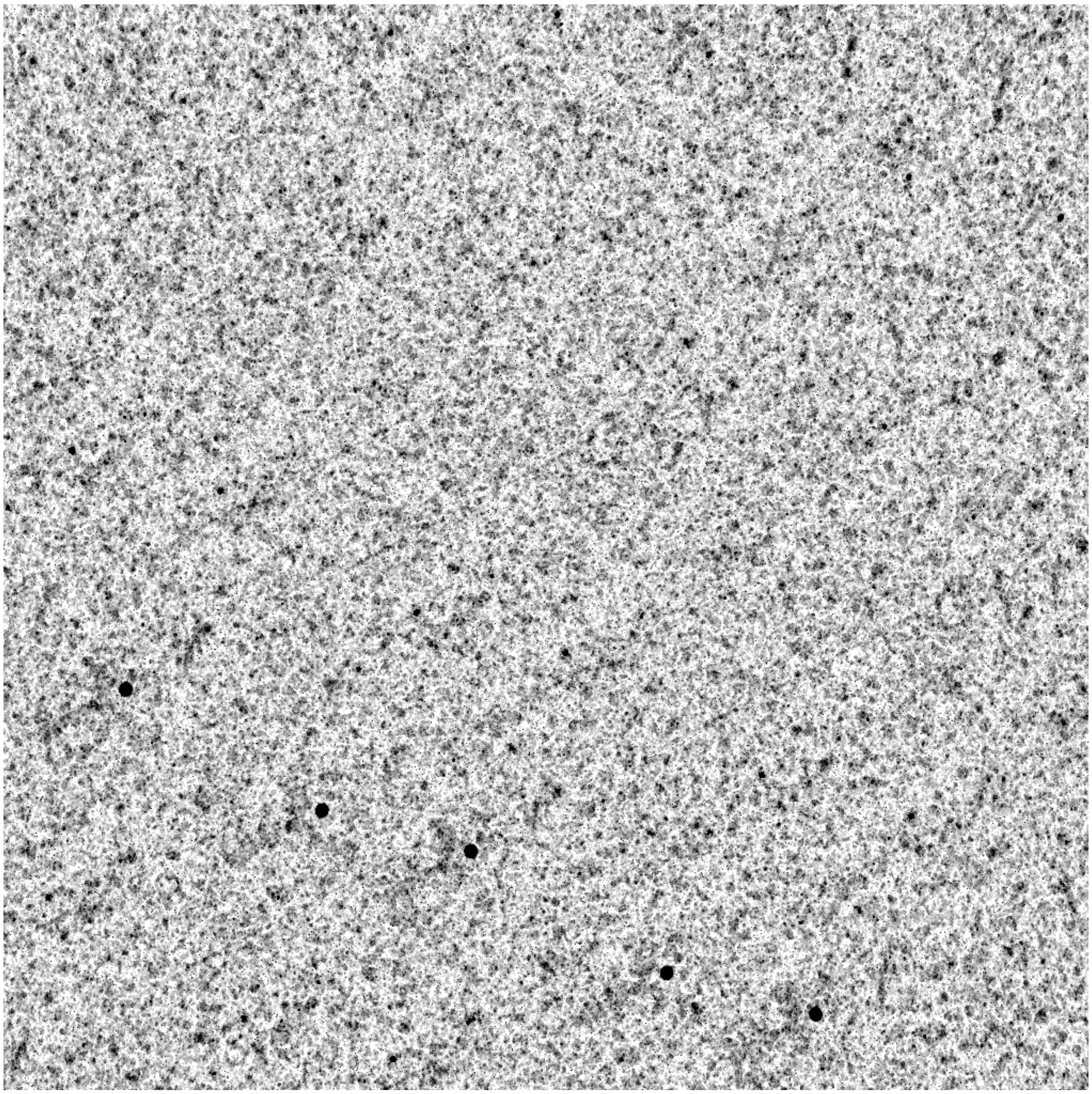
where the pixel is in the hardware array at column i and row j, p_{ijn} is the pixel’s sample at stack location n, and the total number of usable samples in the stack is N_{ij}. The summation shown is over usable samples only. The pixel’s sky offset is s_{ij}, and the denominator in the summation is the uncertainty variance of p_{ijn} with the uncertainty variance of s_{ij} subtracted to compensate for the fact that the differences in the numerator are defined by observed values minus a central value computed from these very observed values.

A sample image from test data is shown below.



4.5 Optional Sample-Size FITS Output

If the “-o4” command-line specification is given, and if sky-offset computation has not been deselected, then the sample-size image is generated. This is the image of the number of pixel values remaining in the stack after outlier rejection for every unmasked pixel as described in section 3.2. A sample image from test data is shown below.



4.6 Optional Time-Slice FITS Output

If the namelist parameters `ITimSlic1` and `ITimSlic2` are specified non-zero, and/or if the namelist parameters `JTimSlic1` and `JTimSlic2` are specified non-zero (see section 2.1.2), then column-range time-slice and/or row-range time-slice images are generated, respectively. "Time-slice" imaging is the generation of FITS files in which consecutive columns correspond to consecutive times in the image stack. Rows in these images may be either hardware array rows or hardware array columns; tempcal will take a specified range of rows or columns (or both, but the results go into separate output images) and generate an image in which the horizontal axis is time and the vertical axis is either row or column. Since rows and columns are handled in analogous fashion, we will describe the column-slice case, which employs `ITimSlic1` and `ITimSlic2`.

If a range of columns is specified, then the image consists of a concatenation of data-cube slices, each slice containing one full row vs. all times, with such slices concatenated for the specified column range, up to 1016 total columns, with additional columns placed into additional output images as needed. This concatenation over column-time slices is set to come as close as possible to a square image without the number of columns exceeding the number of rows. If the number of columns ITimSlic2-ITimSlic1+1 is too large to fit all the slices into one image, then more than one image is generated, with the last column in one image being the first in the next, and the last image containing the same number of slices as the others but ending on ITimSlic2, and therefore possibly having more overlap with the previous image than a single repeated slice.

The images are contained in FITS files named automatically by tempcal as follows: column-slice and row-slice images names begin with ColSlice and RowSlice, respectively; this is followed by an underscore, a four-digit number giving the first hardware column or row in the image, another underscore, and the last hardware column or row in the image. For example, the test data set consisted of 70 time-consecutive science images, so the the time extent is 70 samples, and this is mapped into the horizontal direction. The science images consist of 1016 columns and 1016 rows, so up to 14 slices are concatenated into a single FITS image to arrive at a nearly square image product of 980 columns and 1016 rows. In such an image, the row number is either a hardware row number (for column-range slices) or hardware column number (for row-range slices), and the column number is the time-sample number plus 70 times the number of preceding slices. For example, column 500 corresponds to time sample number 10 in the 8th time slice, i.e., there are seven preceding time slices, so the 10th time sample in the 8th time slice is at image column number $7\times70 + 10 = 500$. In general, with K frames in the stack, time-slice image column I corresponds to time sample N in slice M, where $M = [(I-1)/K]_{\text{truncated}} + 1$, and $N = \text{modulo}(I-1,K) + 1$.

For example, a test run was made in which the namelist input was:

```
 $tmpcalin
   JTimSlic1 = 501, JTimSlic2 = 600,
 $end
```

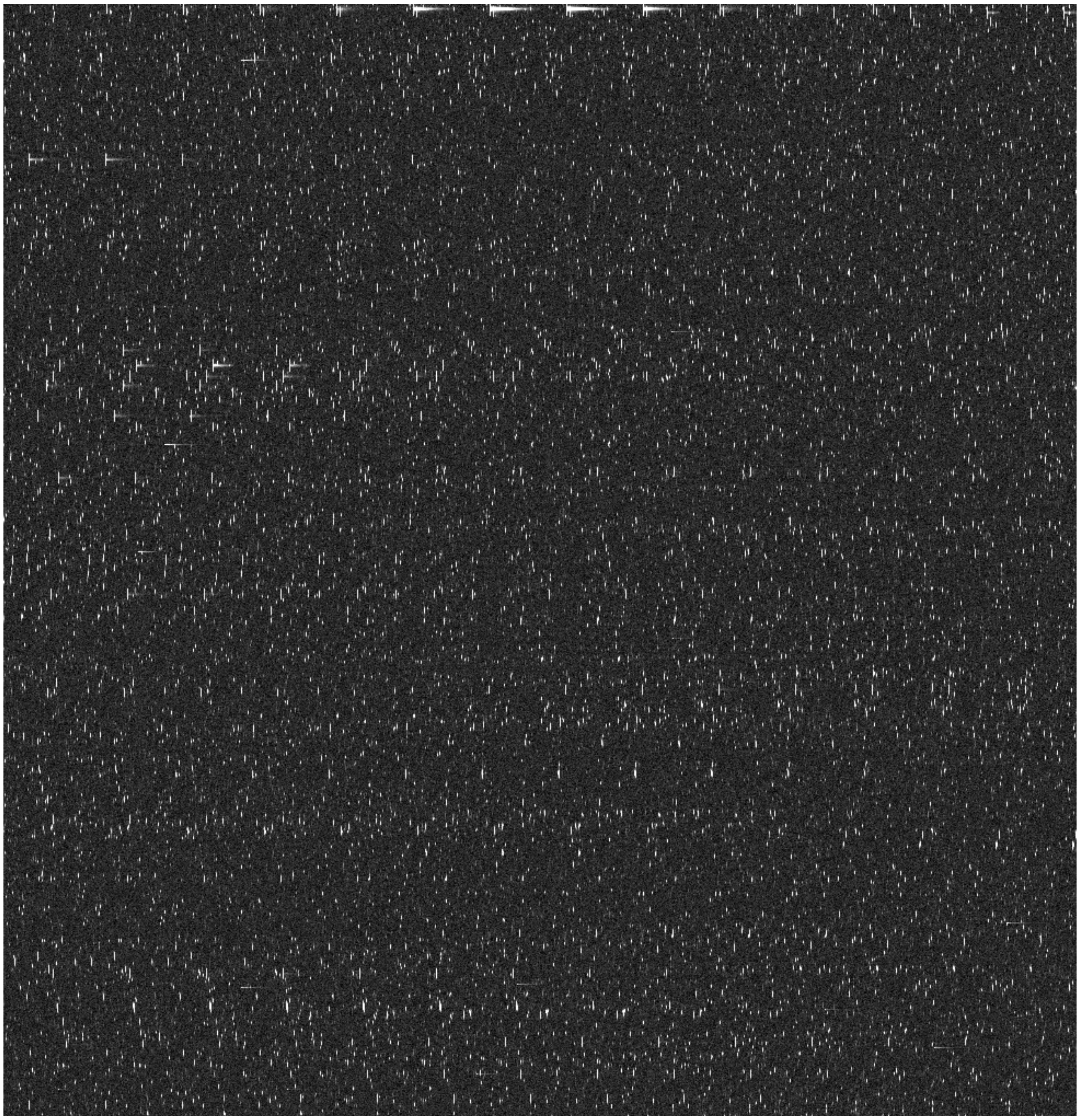
This requests 100 rows in the range 501 to 600. The first time slice contains the 70 time samples and the 1016 hardware columns at hardware row 501, so the leftmost 70 columns and 1016 rows in the time-slice image contain this slice. The next 70 columns and 1016 rows contain the time slice for hardware row 502. This tiling of time slices continues up to the time slice for hardware row 514, which is at the right side of the image. No more slices are concatenated, because this would give the image 1050 columns, more than the number of rows, and this is the arbitrary cutoff for keeping the time-slice images approximately square.

The namelist shown above produced the eight time-slice images shown below:

```
-rw-rw-r--  1 jwf wise 3985920 Jul 25 11:54 RowSlice_0501-0514.fits
-rw-rw-r--  1 jwf wise 3985920 Jul 25 11:54 RowSlice_0514-0527.fits
-rw-rw-r--  1 jwf wise 3985920 Jul 25 11:54 RowSlice_0527-0540.fits
-rw-rw-r--  1 jwf wise 3985920 Jul 25 11:54 RowSlice_0540-0553.fits
-rw-rw-r--  1 jwf wise 3985920 Jul 25 11:54 RowSlice_0553-0566.fits
-rw-rw-r--  1 jwf wise 3985920 Jul 25 11:54 RowSlice_0566-0579.fits
-rw-rw-r--  1 jwf wise 3985920 Jul 25 11:54 RowSlice_0579-0592.fits
-rw-rw-r--  1 jwf wise 3985920 Jul 25 11:54 RowSlice_0587-0600.fits
```

Note that each image contains 14 slices, and the last slice of each image is the first slice in the next. The last image ends with hardware row 600, and since it also contains 14 slices like the others, it begins with hardware row 587, and thus has more than the single-slice overlap with its predecessor. This is done to keep all the time-slice images the same size.

A sample time-slice image is shown below. This is RowSlice_0501_0514.fits. The presence of latent images is immediately obvious, as these show up as streaks in the horizontal (time) direction. The vertical extent of each point source is the image blur in the hardware-column direction. Latent images caused by bright sources generally begin with a significant vertical extent and are followed by a decaying streak. Radiation hits often appear as narrow horizontal streaks. A particularly bright source with a latent tail can be seen near the top of the image; the multiple apparitions spaced 70 columns apart are all the same source seen in adjacent rows. The 70-column width of each time slice can be visually estimated fairly easily because of repetitive patterns like this, even the less striking ones.



5 Testing and Parameter Tuning

5.1 Tempcal Testing

The tempcal module has been unit-tested with simulated data hand-edited to force the various processing paths to be traversed. It has also been tested with simulated FITS data generated by N. Wright for a mid-ecliptic-latitude region around zero degrees longitude and 30 degrees latitude.

5.2 Tempcal Parameter Tuning

The tempcal input parameters which control robust estimation of the sky offset and transient pixel behavior will require tuning prior to their use in routine pipeline processing. The chi-square image should prove useful in this activity, since effective outlier rejection should leave a set of pixel samples which is well behaved with respect to prior uncertainty and local background noise. The two most important parameters are (see section 2.1.2) `ThrshHi` and `ThrshLo.` Also of special interest are `MinPersist`, `MinPix`, and `ChiSqMax`. Finally, the optimal number of consecutive frames has to be determined.

6 Example Command Lines

The following example reads in a list of image FITS file names (`-f1`), corresponding masks (`-f2`), and uncertainty images (`-f3`). The output sky-offset image (`-o1`) will be named skyoff.fits, the output sky-offset uncertainty image (`-o2`) will be named sig_skyoff.fits, a reduced chi-square image (`-o3`) to be named chsq.fits is requested, and a sample-size image (`-o4`) named nused.fits is also requested. The minimum number of consecutive outlier samples to define transient behavior (`-pn`) is set to 20. Samples with input mask bits 1 and 2 set will not be processed (`-m`) since the value 6 was specified, and $6 = 2^2 + 2^1$ (see section 4.3). Samples diagnosed as transient will have bit 21 set in all corresponding masks (`-p`), since 2097152 (2^{21}) was specified. Similarly, mask bits 23, 28, and 25 will be set for unreliable sky offsets (`-s`), unreliable sky-offset uncertainties (`-su`), and probable latents (`-pl`), respectively, since values of 8388608 (2^{23}), 268435456 (2^{28}), and 33554432 (2^{25}), respectively, were specified.

```
% tempcal -f1 intlist -o1 skyoff -o2 sig_skyoff -o3 chsq -o4 nused   \
          -f3 unclist -f2 masklist -pn 20 -p 2097152 -m 6 -s 8388608 \
          -su 268435456 -pl 33554432
```