Description of control file keywords of VLBI data processing software PIMA

L. Petrov

Abstract:

This document provides detailed description of syntax of the language used for specification of control file for VLBI data processing package PIMA

Table of contents:

1   General rules

  Control file for PIMA consists of lines of variable length. Lines which
starts from characters # or * are considered as comments and ignored by
parsing software. Each line consists of a keyword and the value. All
keywords must be specified, no defaults are allowed. A value should be
separated from the keyword by one or more blanks. The first line and
the last of the control file should have the label of the format version.
The current label version is

PIMA_CONTROL file. Format Version of 2015.09.01

  If the file does not have correct format label, PIMA will issue
an errors message and stop. New keywords may be added in the future.
PIMA control files may require an upgrade if the control file
satisfies specifications of the old version of PIMA, but it misses
some keywords added in the newer version. An upgrade of control file
can be done automatically with task upgr. This task accept the
name of the control file in the old, obsolete format and generate
the output control file that conforms to the current format b
adding lines with new keywords that are set to some defaults.

  All keywords may be defined more than once. With exception of

the keywords UV_FITS and INTMOD_FILE the latest definition supersedes
the previous definition.

   PIMA supports a number of kludge environment variables that tweak
the normal operation. They are are either for printing debugging
information or modifying existing code that are invoked on an
exceptional basis. These environment variables have prefix PIMAVAR_.
They can be either defined outside PIMA or placed in the control file.
In that case colon is appended to the end of the environment variable
name.

2 General keyword

2.1 SESS_CODE:

SESS_CODE: value

   Session code. It may be different from the experiment code embedded
in FITS-IDI code. This session code is appended to names of
scratch files.

2.2 BAND:

BAND: value

   One character letter in upper case that describes the band of the
experiment. For processing multifrequency data usually two or more
control files are used. The band helps to distinguish different
control files and, if necessary, to coming multi-frequency
observations.

2.3 UV_FITS:

UV_FITS: file

   Full path name of the data file with the correlator output that
conforms to FITS-IDI standard. More than one UV_FITS: keyword may
be specified and, unlike to other keywords, the next definition does
not supersede the previous one. Restrictions:

1) Files should follow in chronological order.
2) The number of spectral channels for all intermediate frequencies
   (IF) should be the same.
3) The source names and station names should be consistent
   between files, i.e. the same name should be used for the same
   source.

2.4 STAGING_DIR:

STAGING_DIR: **directory**

directory -- This keyword specifies the name of staging directory.
              When the name of staging directory is specified, PIMA
              first checks whether the directory has all files with
              visibilities. If not, then PIMA removes all files from
              that directory and copies there files with visibilities.
              If drive where the staging directory is significantly
              faster than the directory with UV data specified UV_FITS
              keyword(s), PIMA will run faster. It is recommended
              to have SSD RADI-0 disk arrays for staging directory.
              NB: of a side effect: PIMA will remove from that
              directory all files which are not UV-files for this
              experiment.

NO   -- no staging directory is specified

2.5 SOU_NAMES:

SOU_NAMES: **file**

  Name of the source definition file. The purpose of this file is
to match the source names in FITS-IDI files with the source names
used in data reduction. Correlator may use non-standard names.
The source definition file has a separate column that associate
non-standard names used by the correlator with standard names.
In addition, the source definition file may have several entries
for the same source in the column with source name alias and these
entries may have a "splitting" flag. In that case PIMA will treat
these same uv-data as belonging to two different sources with
different a priori coordinates. Supported data format: SOURCE_NAMES.

2.6 STA_NAMES:

STA_NAMES: **file**

  Name of the station definition file. The purpose of this file is
to match station names in FITS-IDI files with the station names
used in data analysis. Correlators often use 2-characters long
names, while data analysis software uses 8-characters long names.

2.7 PCAL:

PCAL: **[NO or USE_ONE or USE_ALL or tone_index][:action:[station[:station]...]**

  This keyword specified how information about measured phases and
amplitudes of phase calibration information should should be used
for fringe fitting. Supported values:

     NO   -- no measured calibration is applied.

     USE_ONE -- to apply one tone per intermediate frequency (IF).
                If two tone are available, then the tone with the
                lowest frequency will be applied. If more than two
                phase calibration tones are available, then the tone
                with index equal to the ratio of the total number of
                phase cal tones per IF will be applied. The phase
                of phase-calibration signal will be subtracted
                with appropriate sign from cross-correlation
                phases in every spectral channels of a given IF.

     USE_ALL -- to apply all tones per intermediate frequency (IF).
                PIMA computes group delay and phase delay in phase
                calibration signal. Using these two values,
                it computes ph-cal phase for every spectral channel
                in the IF and then subtracts them with an appropriate
                sign from cross-correlation phases in every spectral
                channels of a given IF.

   tone_index -- to apply one tone per intermediate frequency (IF).
                The value of tone_index sets the index of the phase
                calibration tone. If the index is out of range,
                then the tone with index equal to the ratio of the
                total number of phase cal tones per IF will be applied.

  The value may has an optional qualifier for fine-grained station selection.
  The qualifier is separated from the value with a column. Sub-value
  action can be TO_USE or NOT_TO_USE. A column separated list of
  stations follows the action sub-value. The action is case insensitive.
  Either column or comma can be used as a separation in the station list.
  If action is TO_USE, then pcal only from the stations form the list will
  be considered for using. If action is NOT_TO_USE, then pcal from the

```
stations on the list will not be used. If phase calibbration for
a given station is not available or was deselected with task gean pcal_off,
then fine-grained phase calibration will have no effect. Task gean
pcal_off has a precedence over fine-grained selection.
```

2.8 TSYS:

TSYS: **[NO or MEASURED or MODELED or CLEANED]**

```
  This keyword specified how system temperature calibration should
be used. Keywords MEASURED or CLEANED or MODELED instruct PIMA from
which slot to take the Tsys. In order to use CLEANED or MODELED
Tsys, the Tsys model has to be computed with task tsmo before.

       NO -- System temperature is ignored

  MEASURED -- System temperature measured when the antenna was on the
              specified source will be used. PIMA will find Tsys
              measurement of the same source to the epoch the nearest
              of the fringe reference time.
              Value INTRP is equivalent to MEASURED and supported for
              compatibility with control files generated before 2017.08.30

  MODELED  -- System temperature from the "model" slot will be used.
              It it is assumed task tsmo ran before and populated
              "modeled" Tsys slot. See documentation of task tsmo for
              details. In short, task tsmo decompose Tsys into a product
              of regular functions of IF, elevation and time assuming
              a) Tsys ratio between IFs is constant during the experiment;
              b) Tsys dependence on elevation is not changing with time;
              c) Tsys is can be decomposed on somewhat smoothed functions
              of time and elevation. Task tsmo flags outliers and computes
              the decomposition using the dataset free from outliers"

  CLEANED  -- System temperature from the "clean" slot. Tsys from that
              slot is a mixture of MEASURED and MODELED Tsys. Tsys
              in the CLEANED slot is the same as in the MEASURED slot
              for the points that has measured Tsys that is not flagged
              out by tsmo task. System temperature is equal to the Tsys
              from the MEASURED slot for the points without measured Tsys
              or for the points that were flagged out.
```

2.9 GAIN:

GAIN: **[NO or USE]**

```
 NO -- do not use antenna gains in the input file or internal PIMA
         data structures. This is equivalent to using gain 1 Jy/K

 USE -- use antenna gains in the input file or internal PIMA
         data structures if during calibration new gain tables have
         been loaded in PIMA.
```

2.10 SAMPLER_CAL:

SAMPLER_CAL: [NO or USE]

```
 NO -- do not apply calibration for amplitude distortion due
         to digitization in the samplers.

 USE -- apply calibration for amplitude distortion due
         to digitization in the samplers.
```

2.11 OBS:

OBS: [ALL or index or index1:index2 or file]

This keyword specifies which observations are to be used for processing. It should be considered as the second filter that is applied after the first filter specified by the keyword INCLUDE_OBS_FILE. Keyword OBS restricts a set of processed observations among those that passed the first filter. The third filter specified by the keyword EXCLUDE_OBS_FILE impose further restrictions that are imposed on the set of observations that passed the second filter.

```
 ALL           -- all observations will be used.

 index         -- Index of the observation that will be used.

 index1:index2 -- Range of indices that will be used.
                   index2 should be greater or equal than index2.

 file:         -- Name of the ascii file that contains indices of
                   observations. The file has one observation index
                   per line. Lines that start from # are considered
                   as comments and bypassed.
```

2.12 INCLUDE_OBS_FILE:

`INCLUDE_OBS_FILE:   [NO or file]`

  This keyword specifies the first filter of observations that are
to be used. Those observations that passed this filter are considered
as eligible for checking for the second filter specified by the keyword
OBS: and the third filter specified by the keyword EXCLUDE_OBS_FILE:

  NO   -- all the observations are to be used.

  file -- Name of the file that specifies indices of observations that are
        put in the include list. The file has one observation index
        per line. Lines that start from # are considered as comments
        and bypassed.

2.13 EXCLUDE_OBS_FILE:

`EXCLUDE_OBS_FILE:   [NO or file]`

  This keyword specifies the third filter of observations that are
to be used. Those observations that passed this filter form the final
list of used observations.

  NO   -- Exclude list is empty. That means that all observations
        that passed the first filter specified by the keyword
        INCLUDE_OBS_FILE: and the second filter specified by
        the keyword OBS: are considered as eligible.

  file -- specifies indices of observations that form the exclude list.
        The observations indices specified in this files are excluded
        from the list that passed the first and the second filter.

2.14 WARNING:

`WARNING:            [NO, ON, OFF]`

  NO -- No warning messages will be printed

  OFF -- No warning messages will be printed

  ON -- Warning messages will be printed.

2.15 DEBUG_LEVEL:

```
DEBUG_LEVEL:        value
```

Level of verbosity of information messages issued by PIMA

0 -- completely silent mode. Only errors or warnings
     (if WARNING: ON is specified) are printed.

1 -- terse mode. Only very important messages are printed.

2 -- normal mode.

3 -- talkative mode. More messages are displayed.

4 -- chat-box mode. A lot of messages are displayed that
     may be annoying.

5-99 -- debugging messages are printed. These messages are
        are for facilitation of debugging.

2.16 CHECK_SEVERITY:

```
CHECK_SEVERITY:     value
```

Level of severity of checks for internal data consistency.
During loading the data, situations occur when a portion of
data is corrupted. PIMA detects data corruption and issues
warning, but then it should decide what to do further: to
continue or to stop. The level of severity governs the algorithm.

0 -- permissive mode. Unless the error is fatal that prevents
     further data processing, PIMA discards corrupted data
     and continues.

1 -- normal mode. PIMA may or may stop after an error depending
     on its severity. It stops if corrupted data *may* produce
     bogus results.

2 -- strict mode. PIMA stops if it finds corrupted data.

2.17 FRINGE_ERRORS:

FRINGE_ERRORS: [IGNORE, STOP]

  This keywords set the action if fringe search ended with error

  IGNORE -- the error is ignored and PIMA proceeds to the new
            observation. Code failure is written in the status field
            for the affected observation.

  STOP   -- pima stops on the error.

2.18 FFT_METHOD:

FFT_METHOD: [FFTW or MKL]

  PIMA supports two software packages that implements fast Fourier
transform: MKL (Math Kernel Library by Intel) and open source FFTW.
MKL support is optional and depends on PIMA configuration in compile
time. Keyword FFT_METHOD defines which library to use. Both libraries
produce correct answers, but depending on size of the problem
have different speed. If in doubt, please use FFTW method.

  MKL   -- use the MKL library

  FFTW -- use open source FFTW library

2.19 FFT_CONFIG_FILE:

FFT_CONFIG_FILE: [NO or file]

   When FFTW is used, its configuration, so-called "wisdom" that keeps
optimal setting should be loaded. Refer to PIMA INSTALL document,
Post-Installation section with a detailed explanation how to create this
configuration file. PIMA will run without FFTW configuration, but its
performance will be significantly degraded.

NB: FFTW configuration depends on the number of threads. Single threaded
configuration file will be rejected if FFTW is called in multi-thread
mode and performance will be severely impacted. An M-threaded configuration
file will be rejected if the number of threads for FFTW is not equal to M.

```
NO   -- no FFTW configuration file is specified. FFTW configuration
        file is ignored if FFT_METHOD MKL is used. You can use NO value
        even if FFTW methods is used, but PIMA performance will be worse
        by a factor of 2-20.

file -- name of the FFTW configuration file.
```

2.20 NUM_THREADS:

NUM_THREADS: value

```
  The number of threads used by FFT. Usually, FFT_NUM_THREADS is set
to the number of processors (cores) available. But you may want to
reduce the number of threads for load balancing.
```

3 Observation processing keywords

3.1 AP_TOLERANCE:

AP_TOLERANCE: value

```
  PIMA assumes the time epochs of all the uv data within each scan forms
a rail of ticks with spacing equal to some constant, accumulation
period length. Parameter AP_TOLERANCE sets the tolerance in seconds for UV
data to have time epochs with interval slightly greater or less than the
accumulation period length. If PIMA finds UV data that have time epoch
that differs from previous accumulation period by more than AP_TOLERANCE,
it will discard that data point. NB: the start time of a scan may be not
commensurate to the accumulation period length counted from the previous
scan, only time stamps within each scan are checked.

  For geodesy and astrometry applications AP_TOLERANCE should not exceed
1.D-6 seconds. For imaging application the tolerance may be relaxed.
```

3.2 MIN_SCAN_LEN:

MIN_SCAN_LEN: value

Parameter MIN_SCAN_LEN determines the minimum scan duration in seconds.
It instructs the PIMA internal algorithm for splitting visibility data
into scans that a scan should not be shorter than that value. If it finds
a set of visibility data shorter than this value that cannot be attached
to another scans in accordance with the constraints set by other parameters
of PIMA, it will flag these visibilities.

3.3 MAX_SCAN_LEN:

MAX_SCAN_LEN: **value**

Parameter MAX_SCAN_LEN determines the maximum scan duration in seconds.
It instructs the PIMA internal algorithm for splitting visibility data
into scans that a scan should be not longer than not that value. Scan
duration is set as an interval between the beginning of the first
accumulation period and the end of the last accumulation period. A set
of data of the same source may last longer than MAX_SCAL_LEN. In that
case that set of data will be split into several scans. Splitting
of data into scans occurs in chronological order. Thus, after putting
the earliest visibility set into a given scan, the scan end is fixed,
and the visibilities for the following epochs are put into the next scan.

3.4 MAX_SCAN_GAP:

MAX_SCAN_GAP: **value**

Parameter MAX_SCAN_GAP controls maximum duration of a gap in data
within each scan for the algorithm that splits visibilities into scans.
If a gap in visibility data of the same source long than this value
occurred, it triggers setting the scan boundary at the beginning of the
gap. The start time epoch of the new scan is set to the the first time
epoch after the gap.

3.5 SCAN_LEN_SKIP:

SCAN_LEN_SKIP: **value**

This parameter instructs PIMA to discard the visibilities at the
beginning of a scan from the nominal scan start for the interval of time

equal to this value in seconds. The value of SCAN_LEN_SKIP cannot be
negative.

3.6 SCAN_LEN_USED:

SCAN_LEN_USED: value

   This parameter instructs PIMA to discard the visibilities at the end of
a scan. It control the duration of the interval in seconds with used
visibilities. The beginning of the interval is counted from the nominal
scan length if SCAN_LEN_SKIP is 0. If SCAN_LEN_SKIP > 0, the beginning
of the interval is offset at SCAN_LEN_SKIP with respect to the nominal
scan start. The visibility data beyond the interval specified by this
keyword are discarded.

3.7 FRT_OFFSET:

FRT_OFFSET: [AUTO or value or file]

   Keyword FRT_OFFSET instructs PIMA how to determine to fringe reference
time, i.e. the time to which residual phase related.

   AUTO -- Instructs PIMA to determine the fringe reference time
            automatically. PIMA determines it as a mean weighted
            epoch within a scan for each observation separately.
            NB: In general, the fringe reference time for different
            observation of a given scan is not the same.

   value -- sets the offset of the fringe reference time with
            respect to the nominal scan start.

   file -- Specifies the file with fringe reference time.
            The file is in ASCII format. Each line has two words
            separated by one or more blanks. The first word
            defines the observation index, the second words
            defines the fringe reference frame in seconds with
            respect the scan nominal start. Lines that start
            from # are considered as comments and ignored.

3.8 STA_REF:

STA_REF: Station name

Specifies the name of the reference station. The reference station
is used for computation of the complex bandpass.

3.9 VTD_CONFIG_FILE:

VTD_CONFIG_FILE: file

Specifies the name of VTD (VLBI Time Delay) control file. The VTD
control file specifies parameters of the VLBI model for path delay.

3.10 EXPER_DIR:

EXPER_DIR: directory

Name of the directory where PIMA task load writes index files.
Index files have names that start from the session name and end on
specific extensions.

3.11 UV_EXCLUDE_FILE:

UV_EXCLUDE_FILE: []file or AUTO or NO]

file -- File name with indices of visibility dates that are
        excluded from processing. Lines that start from #
        are considered as comments and ignored.

AUTO -- invokes a special algorithm for automatic accounting for
        bad uv data points for PIMA task load. The algorithm first
        searches for
        a file in EXPER_DIR directory with the name that has
        the first part session name and the last part _uv.exc
        If it finds that file, it reads it. The file is contains
        the indexes of the visibility data that are excluded
        before further analysis. These are s- called a priori
        excluded visibilities. If PIMA finds bad visibilities,
        it generates a list of bad visibility names. These are
        so-called a posteriori bad visibilities. If there is
        at least one bad a posteriori visibility, PIMA combines

```
        the indexes of a priori and a posteriori bad visibilities,
        writes them down into a file with the same name:
        the first part session name and the last part _uv.exc,
        and sets exit code 23.

        If the control file has UV_EXCLUDE_FILE: AUTO and the
        first run of PIMA task load returned exit code 23, then
        the second run is recommended. Bad points identified
        in the first run, will be bypassed during the second run.

   NO   -- No visibility data are excluded.
```

3.12 BANDPASS_USE:

BANDPASS_USE: [AMP or PHS or AMP_PHS or NO]

This keyword controls whether to apply complex bandpass and if
to apply, then how. A bandpass is a complex function of frequency.
For each station, each IF and each spectral channel, a complex value
is defined. Applying the bandpass is an operation when raw visibilities
are divided by this value.

```
   AMP      -- apply the amplitude part of the bandpass, i.e. to change
               only amplitudes, but do not change phases.

   PHS      -- apply phase part of the bandpass, i.e. to change only
               phases, but do not change amplitudes.

   AMP_PHS -- apply the complex bandpass, i.e. change both amplitudes
               and phases.

   NO       -- do not apply the complex bandpass.
```

3.13 BANDPASS_FILE:

BANDPASS_FILE: file or NO

This keyword controls the name of the bandpass file. Bandpass is
a complex function of frequency. It defines a complex number for each
station, each spectral channel of each IF. The visibility is divided
by a product of bandpasses for the pair of stations of the baseline.
Bandpass file name is an output parameter for task bpas. Therefore, it
may not exist for this task. The bandpass file name is an input
parameters for many other tasks, such as frib, splt, mkdb. PIMA checks

whether the specified file exists and stops with an error message,
if cannot find it and read.

   file -- name of the bandpass file.

   NO   -- No bandpass file exists. You cannot specify NO with bpas task.

3.14 POLARCAL_FILE:

POLARCAL_FILE: file or NO

   This keyword controls the name of the polarization bandpass file.
In a case of dual-polarization data PIMA supports two bandpasses.
The first bandpass is for RR data. It is defined in the file specified
by BANDPASS_FILE name. The second bandpass determines the complex
function of frequency that describes the ration of complex response
of LL band with respect to RR. That second bandpass is specified by
POLARCAL_FILE. Therefore, the LL bandpass is the product of these
two bandpasses. When PIMA calibrates RR data, it uses only the first
bandpass. When PIMA calibrates LL data, it uses the product of two
bandpasses. When PIMA calibrates data for polarization I, which it
computes on the fly, it uses both bandpasses. POLARCAL_FILE is not
used for single-polarization data. Bandpass file name is an
output parameter for task bpas. Therefore, it may not exist for this
task. The bandpass file name is an input parameters for many other
tasks, such as frib, splt, mkdb. PIMA checks whether the specified
file exists and stops with an error message it it cannot find
it and read.

   file -- name of the second polarization bandpass file.

   NO   -- No polarization bandpass is to be used or to be created.
           Task bpas will not compute polarization bandpass
           if POLARCAL_FILE: NO.

3.15 BANDPASS_MASK_FILE:

BANDPASS_MASK_FILE: file or NO

   This keyword controls the name of the mask file. Mask file consists
of 0 and 1 for each station, each IF, each frequency channel. PIMA
defines 4 masks: 1) for autocorrelation; 2) for computation of bandpass;
3) for fringe fitting, and 4) for task splt. Normally mass 2-3-4 are
the same. When PIMA processes autocorrelation, it multiplies
autocorrelation data by the mask value 0 and 1. Then it interpolates

the autocorrelation and puts on place those spectral channels that
were masked out (i.e. had 0 in the mask file) the value computed by
interpolation between neighboring channels that were not masked out.
When PIMA processes cross correlation data, it multiplies visibilities
on mask values. If a given channels is masked out, it is effectively
excluded from further processing.

  Bandpass mask file must define mask values for all spectral channels,
all IFs, all stations. PIMA supports task bmge (Bandpass Mask GEneration)
that allows to describe in an ascii files only those spectral channels,
IFs or their ranges that have to be masked out. Task bmge reads the
input mask definition file and generates the output bandpass mask file.

    file -- name of the the bandpass mask file. This file name is the
           output for task bmge. When task bmge is used, this file may
           not exist. For other tasks this file should exists. PIMA will
           stop with an error message if it will not find it.

    NO   -- No bandpass file exists. You cannot specify NO with bmge task.

3.16 PCAL_MASK_FILE:

PCAL_MASK_FILE: file or NO

  This keyword controls the name of the mask file for phase calibration
data. It is applied only when all phase calibration tones are used
(PCAL: USE_ALL). Phase calibration signal is often affected by internal
radio interference. The mask file allows to exclude (to mask out) the
tones that are affected by the RFI. When the interpolation spline is
computed to approximate the frequency dependence of phase calibration
signal, these tines are excluded from computation of the spline.

  Phase calibration mask file must define mask values for all spectral
channels, all IFs, all stations. PIMA supports task pmge (Phase
calibration Mask GEneration) that allows to describe in an ascii files
only those spectral channels, IFs or their ranges that have to be masked
out. Task pmge reads the input mask definition file and generates the
output polarization mask file.

    file -- name of the the phase calibration mask file. This file name
           is the output for task pmge. When task bmge is used, this
           file may not exist. For other tasks this file should exists.
           PIMA will stop with an error message if it will not find it.

    NO   -- No polarization make file exists. You cannot specify NO
           with pmge task.

3.17 INTMOD_FILE:

INTMOD_FILE: **[directory or NO]**

   Normally, the a priori interferometric model is defined inside
the FITS-IDI data. However, some correlators omit these sections
in FITS-IDI files, which prevents astrometry and geodetic applications
that require computation of the total phases, group delays, phase
delay rates and group delay rates. As a workaround PIMA supports
import of interferometric models in the form that were used by
the correlator. Keyword INTMOD_FILE specifies a file or a directory
tree(s) that contains files with the a priori model for path delay.
Interpretation of files depends on the correlator. In a case of
VERA hardware correlator in Mitaka, PIMA searches for files
that has "ANT." pattern and ignores all others. In a case of
SFXC correlator, PIMA searches for files with extension ".del"
and ignores all others. PIMA also searches for the clock data
file with extension ".clk". This keyword is used by task moim
(MOdel IMport). More than one INTMOD_FILE keyword is allowed.

   More than one keyword INTMOD_FILE can be specified.

   directory -- directory name where PIMA will search for files with
               a priori model, including all sub-directories.

   NO   -- No directory name is specified. NO can be specified
               only when INTMOD_TYPE is NO.

3.18 INTMOD_TYPE:

INTMOD_TYPE: **[VERA1000 or VERA2000 or SFXC or NO]**

   This keywords defines the type of external path delay model.
This value of this keyword controls how PIMA should interpret
the files with a priori path delay used by the correlator specified
by the keyword INTMOD_FILE.

   VERA1000 -- delay files are in CODA format for VERA1000 recording
               systems. The files are named ANT.x, where x is an
               integer number.

   VERA2000 -- delay files are in CODA format for VERA2000 recording
               systems. The files are named ANT.x, where x is an
               integer number. NB: VERA1000 and VERA2000 have different
               format.

   SFXC   -- delay files are in ASCII format for SFXC correlator.
               The files have extension .del

```
   NO      -- format is undefined. Should be used when INTMOD_TYPE: NO
             and cannot be specified otherwise.
```

3.19 CORR_FLAG_MIN:

CORR_FLAG_MIN: value

  This keyword defines the threshold for discarding the data that
are marked by the correlator as potentially corrupted. The data
with CORR_FLAG_MIN equal or less are discarded. Usually,
CORR_FLAG_MIN: 1 is a good choice. That means any data flagged
by the correlator will be discarded. However, if you think that
the correlator discards too many 'good' data, you can lower
the threshold.

  FITS-IDI specifications define the flag the following way:

```
-1 No severity level assigned
 0 Data are known to be useless
 1 Data are probably useless
 2 Data may be useless
```

   The VLBA hardware correlator may assign flag -1 (i.e. undefined
severity). In that case FITS-IDI keyword CORR_FLAG_REASON string that
describes the problem that resulted in flagging must be defined. When
PIMA finds a severity flag -1 it parses the string with the reason.
Then it assigns the severity level, i.e the correlator flag,
the following way:

```
   2-16 ghz synthesizer #2  --  FLAG = 2
   2-16 ghz synthesizer #2  --  FLAG = 2
   antenna not in point mod --  FLAG = 0
   Antenna not pointed      --  FLAG = 0
   antenna position error t --  FLAG = 0
   Antenna off source       --  FLAG = 0
   BBC Synth Unlocked       --  FLAG = 2
   channel 1 bbc synthesize --  FLAG = 2
   channel 2 bbc synthesize --  FLAG = 2
   channel 3 bbc synthesize --  FLAG = 2
   channel 4 bbc synthesize --  FLAG = 2
   channel 5 bbc synthesize --  FLAG = 2
   channel 6 bbc synthesize --  FLAG = 2
   channel 7 bbc synthesize --  FLAG = 2
   channel 8 bbc synthesize --  FLAG = 2
   Ellipsoid posn error     --  FLAG = 2
   ellipsoid position error --  FLAG = 2
   observing system idle    --  FLAG = 2
   Recorder 1 head posn err --  FLAG = 2
   Recorder 1 not running   --  FLAG = 2
```

```
Recorder 2 head posn err -- FLAG = 2
Recorder 2 not running   -- FLAG = 2
Source change in progres -- FLAG = 0
source change in progres -- FLAG = 0
Subreflector error       -- FLAG = 2
Synthesizer 1 unlocked   -- FLAG = 2
Synthesizer 2 unlocked   -- FLAG = 2
Synthesizer 3 unlocked   -- FLAG = 2
System idle              -- FLAG = 0
subreflector position er -- FLAG = 2
2-16 ghz synthesizer #1  -- FLAG = 2
2-16 ghz synthesizer #2  -- FLAG = 2
2-16 ghz synthesizer #3  -- FLAG = 2
```

   If it does not find CORR_FLAG_REASON string or finds a string
that is not in the table above, its stops with an error message.

   value -- the threshold of the correlator flag. If a UV point
         has the correlator flag equal or less than that code,
         PIMA flags that observation out, i.e. replaces the
         visibility data with zeroes.

3.20 TIME_FLAG_FILE:

TIME_FLAG_FILE: [file or NO]

   This keyword defines the name of the file that sets multiplicative
reweighting factors for all visibilities at specified time epochs
at specified observations. Observations weight will by multiplied by
this factor. Usually, these reweighting factors are zero. In that case,
TIME_FLAG_FILE sets the accumulation periods that PIMA must discard.

   The ascii TIME_FLAG_FILE consists of lines that contains three words:
observation index, ap index from the nominal observation start, weight.
Usually this file is generated automatically either by a user
application or by PIMA task onof.

   file -- name of the time flag. When PIMA uses onof task in the mode
         of time flag file generation, the file name must be specified.
         PIMA will stop if the file name is not specified. For all
         other tasks, keyword TIME_FLAG_FILE with the file name
         specified means that PIMA should read this file and update
         weights for the specified observations, specified accumulation
         periods.

   NO   -- means no multiplicative reweighting factors will be applied.

3.21 TEC_FILE:

TEC_FILE: file or NO

   TEC file name. At the moment is not supported. Reserved for future use.
TEC_FILE: NO should be specified.

3.22 FRINGE_FILE:

FRINGE_FILE: file

   Name of the ascii output file where results of fringe fitting
are written. If the file exists, PIMA will append records to its end.
PIMA first writes the header that starts with # character
and writes a terse footer at the end.

3.23 FRIRES_FILE:

FRIRES_FILE: file

   Name of the ascii output file where residuals of fringe fitting
are written. If the file exists, PIMA will append records to its end.
PIMA first writes the header that starts with # character
and writes a terse footer at the end.

3.24 BEG_FRQ:

BEG_FRQ: value

 The start intermediate frequency (IF) index that is used by PIMA.
Cannot be less than 1.

3.25 END_FRQ:

END_FRQ: value

 The end intermediate frequency (IF) that is used by PIMA. Cannot be

less than BEG_FRQ and cannot be greater than the last IF.

3.26 FRQ_GRP:

FRQ_GRP: [value or value_min:value_max or value_min−value_max]

  Frequency group. In a case if the input FITS-IDI data have more than
one intermediate frequency (IF), the IFs can form one or more groups.
This keywords specifies which frequency groups will be used. A valid
FITS-IDI file contains at least one frequency group with index 1.
NB: PIMA can perform fringe fitting only within one frequency group.

    value   -- the index of the used frequency group.
                          Cannot be less than 1 and cannot be greater than
                          the total number of frequency groups.

    value_min:value_max -- merge several frequency groups together for
                          overlapping data. It is assumed that there are
                          visibility data for the same time epochs
                          within groups that are being merged.
                          value_min and value_max are the indexes of
                          the first and the last frequency group
                          to merge. They form a virtual frequency group.
                          The number of IFs in the merged frequency
                          group is the sum of the number of IFs in
                          participating groups.

    value_min−value_max -- combine several frequency groups together for
                          non overlapping data. It is assumed that there
                          are no visibility data for the same time epochs
                          within the groups that are being combined.
                          A new virtual group is created with the index
                          that is by one greater than the number of
                          non−virtual groups. The new combined group
                          has all IFs of its constituents.

NB: keep in mind the distinction between merged and combined frequency
groups.

3.27 POLAR:

POLAR: [RR or LL or RL or LR or I]

  Polarization code. In the case if one polarization was correlated,
it must be either RR or LL depending on used polarization. If two
polarization were correlated, it can be RR or LL or I. In a latter

case PIMA apply the bandpass to RR and both bandpass and polarization
bandpass to LL polarizations, rotates RR fringe phase at the
parallactic angle and LL fringe phases at the same angle but with the
opposite sign and forms a calibrated linear combination I = (RR + LL)/2.
The SNR of this combination is at 40% better than the SNR of RR or LL
data. NB: polarization I should be used only after both bandpass
and polarization bandpasses were computed. Otherwise, the results will
be totally unsatisfactory.

3.28 WVR_FILE:

WVR_FILE: **file name**

   file_name: Full path name of the WVR data. At the moment,
               only form "* WVR_EFL data. Format version 2014.07.24"
               is supported. If no WVR data are available, NO should
               be used. More than one file can be specified,
               one file per station.

3.29 WVR_USE

WVR_USE: **usage**

   usage: WVR usage. Specifies the algorithm for interpolation and
              smoothing of original WVR data and the contribution of
              WVR path delay phases should be subtracted from the
              visibility phases. Supported values:

              NO   -- do not apply visibility phases

              WVR_3SPL -- Expand original data into smoothing
                     spline of the 3rd degree.

              WVR_LIN -- Expand original data into smoothing
                     spline of the 1st degree.

              WVR_AVR -- Expand original data into smoothing
                     spline of the 3rd degree.

3.30 WVR_SMOOTHING_INTERVAL

WVR_SMOOTHING_INTERVAL: **value**

```
value:                      interval of the smoothing spline in seconds.
                            10-40 seconds is usually adequate
```

3.31 WVR_SMOOTHING_SIGMA

WVR_SMOOTHING_SIGMA: value

```
   value:  Reciprocal weight of the constraint imposed
           on WVR smoothing spline in seconds. The smaller
           parameter, the close smoothing spline to the
           averaged path delay. The bigger parameter,
           the greater fluctuations in WVR phases are
           allowed. 2.0D-11 is usually adequate.
```

3.32 PHASE_ACCELERATION:

PHASE_ACCELERATION: value

```
  Ad hoc phase acceleration that will be applied to visibilities.
Units: rad/s^2. Normally 0.0 value used. Values different from
zero are used for radio interferometer with a station at the
space.
```

3.33 PHASE_ACCEL_MIN:

PHASE_ACCEL_MIN: value

```
  Minimal phase acceleration in rad/s^2 that  will be used for
iterations. PIMA supports the mode when it tries a range of phase
accelerations with a step that it determines automatically. The step
is found in such a way that the maximum contribution of the phase
acceleration term be less than 0.33 rad. PIMA runs N times fringe
fitting with different phase accelerations for each observation.
PHASE_ACCEL_MIN: 0.0 and PHASE_ACCEL_MAX: 0.0 tells PIMA that no
tries with a range of phase acceleration should be done
```

3.34 PHASE_ACCEL_MAX:

PHASE_ACCEL_MAX: **value**

  Maximal phase acceleration in rad/s^2 that will be used for
iterations. PIMA supports the mode when it tries a range of phase
accelerations with a step that it determines automatically. The step
is found in such a way that the maximum contribution of the phase
acceleration term be less than 0.33 rad. PIMA runs N times fringe
fitting with different phase accelerations for each observation.
PHASE_ACCEL_MIN: 0.0 and PHASE_ACCEL_MAX: 0.0 tells PIMA that no
tries with a range of phase acceleration should be done

3.35 EPHEMERIDES_FILE:

EPHEMERIDES_FILE: **[file or NO]**

  Name of the ephemeride file for space interferometer. The file
should conform CCSDS_OEM_VERS specifications. This ephemerides
file is used for computation of the position of the station that
is in space. If one of the stations is marked in antenna description
file used by VTD as space station, ephemerides file must be specified.

  NO -- no ephemerides file is specified. If you analyze an experiment
        only with ground station, use NO.

3.36 EPHEMERIDES_USE:

EPHEMERIDES_USE: **[NO or RA_PUSCH or RA_GBT or RA_GB140 or EARTH_OR**
                         or STA_ORB or INTERPLA]

  This keyword specifies the type of space interferometer. It instructs
the algorithm for computation of path delay.

  RA_PUSCH -- Radioastron type of interferometer. The space station
              has its H-maser that does not keep time between scans
              and is synchronized with the ground H-maser at PUSCHINO
              radiotelescope.

  RA_BGT   -- Radioastron type of interferometer. The space station
              has its H-maser that does not keep time between scans
              and is synchronized with the ground H-maser at GB-VLBA
              radiotelescope.

  RA_GB140 -- Radioastron type of interferometer. The space station

has its H-maser that does not keep time between scans
and is synchronized with the ground H-maser at NRAO140
radiotelescope.

EARTH_OR -- the space radio telescope is Earth orbiting. Its clock are
synchronized before the experiment.

INTERPLA -- the space radio telescope is an interplanetary station.

NO   -- there are no space radio telescopes in this experiment.

4 Baseline fringe fitting keywords

The baseline fringe fitting procedure is the main task of PIMA.
The goal of the fringe fitting procedure is to find group delay,
group delay rate, phase delay rate and possibly phase delay
acceleration in such a way that would minimize the loss of coherency
during averaging over time and frequency.

Baseline fringe fitting is done for each observation independently.
It consists of two steps: coarse fringe search and refinement.
In general, optimal determination of group delay, group delay rate,
phase delay rate and phase delay acceleration is a non-linear
procedure. The goal of the mandatory coarse fringe search is to find
an approximate value of group delay and phase delay rate in order
to shrink the search area and make possible linear estimator.

PIMA supports several algorithms of fine search that improves
result of the coarse fringe search. Results of fringe search and
statistics of residuals are written in the ascii output files
defined by keywords FRINGE_FILE and FRIRES_FILE.

4.1 FRIB.SEARCH_TYPE:

FRIB.SEARCH_TYPE: 2FFT

At the moment the only supported fringe search type is 2FFT. It is
based in the procedure that uses 2D FFT in order to find the maximum of
the visibility for the observation averaged with respect to time and
frequency with trial group delay and phase delay rate.

4.2 FRIB.DELAY_WINDOW_CENTER:

FRIB.DELAY_WINDOW_CENTER: value

   PIMA allows to restrict the area in group delay, phase delay rate
space where the maximum of visibilities averaged over frequency and
time is sought. This rectangular area is determined by four parameters:
FRIB.DELAY_WINDOW_CENTER, FRIB.RATE_WINDOW_CENTER,
FRIB.DELAY_WINDOW_WIDTH, FRIB.RATE_WINDOW_WIDTH.

   Keyword FRIB.DELAY_WINDOW_CENTER sets the center of the window over
group delay in seconds.

   If you want to set the maximum search window, i.e. you do not see
a reason to restrict it, please set all four parameters
FRIB.DELAY_WINDOW_CENTER, FRIB.RATE_WINDOW_CENTER,
FRIB.DELAY_WINDOW_WIDTH, FRIB.RATE_WINDOW_WIDTH to zero.

4.3 FRIB.RATE_WINDOW_CENTER:

FRIB.RATE_WINDOW_CENTER: value

   PIMA allows to restrict the area in group delay, phase delay rate
space where the maximum of visibilities averaged over frequency and
time is sought. This rectangular area is determined by four parameters:
FRIB.DELAY_WINDOW_CENTER, FRIB.RATE_WINDOW_CENTER,
FRIB.DELAY_WINDOW_WIDTH, FRIB.RATE_WINDOW_WIDTH.

   Keyword FRIB.RATE_WINDOW_CENTER sets the center of the window over
phase delay rate. This quantity is dimensionless.

   If you want to set the maximum search window, i.e. you do not see
a reason to restrict it, please set all four parameters
FRIB.DELAY_WINDOW_CENTER, FRIB.RATE_WINDOW_CENTER,
FRIB.DELAY_WINDOW_WIDTH, FRIB.RATE_WINDOW_WIDTH to zero.

4.4 FRIB.DELAY_WINDOW_WIDTH:

FRIB.DELAY_WINDOW_WIDTH: value

   PIMA allows to restrict the area in group delay, phase delay rate
space where the maximum of visibilities averaged over frequency and
time is sought. This rectangular area is determined by four parameters:
FRIB.DELAY_WINDOW_CENTER, FRIB.RATE_WINDOW_CENTER,
FRIB.DELAY_WINDOW_WIDTH, FRIB.RATE_WINDOW_WIDTH.

   Keyword FRIB.DELAY_WINDOW_WIDTH sets the half-width of the

delay in seconds. That means the search window is
[FRIB.DELAY_WINDOW_CENTER − FRIB.DELAY_WINDOW_WIDTH,
 FRIB.DELAY_WINDOW_CENTER + FRIB.DELAY_WINDOW_WIDTH].

  If the value is zero, then the maximum search window is set up:
1.0/Chan_with where Chan_width is the spectral resolution of
visibility data in Hz. NB: setting FRIB.DELAY_WINDOW_WIDTH greater
than the maximum search window has no effect. Parameter
FRIB.DELAY_WINDOW_WIDTH used for restricting the search window, not
for its increase.

  If you want to set the maximum search window, i.e. you do not see
a reason to restrict it, please set all four parameters
FRIB.DELAY_WINDOW_CENTER, FRIB.RATE_WINDOW_CENTER,
FRIB.DELAY_WINDOW_WIDTH, FRIB.RATE_WINDOW_WIDTH to zero.

4.5 FRIB.RATE_WINDOW_WIDTH:

FRIB.RATE_WINDOW_WIDTH: value

  PIMA allows to restrict the area in group delay, phase delay rate
space where the maximum of visibilities averaged over frequency and
time is sought. This rectangular area is determined by four parameters:
FRIB.DELAY_WINDOW_CENTER, FRIB.RATE_WINDOW_CENTER,
FRIB.DELAY_WINDOW_WIDTH, FRIB.RATE_WINDOW_WIDTH.

  Keyword FRIB.DELAY_WINDOW_WIDTH sets the half−width of the
delay in seconds. That means the search window is
[FRIB.DELAY_WINDOW_CENTER − FRIB.DELAY_WINDOW_WIDTH,
 FRIB.DELAY_WINDOW_CENTER + FRIB.DELAY_WINDOW_WIDTH].

  If the value is zero, then the maximum search window is set up:
1.0/Chan_with where Chan_width is the spectral resolution of
visibility data in Hz. NB: setting FRIB.DELAY_WINDOW_WIDTH greater
than the maximum search window has no effect. Parameter
FRIB.DELAY_WINDOW_WIDTH used for restricting the search window, not
for its increase.

  If you want to set the maximum search window, i.e. you do not see
a reason to restrict it, please set all four parameters
FRIB.DELAY_WINDOW_CENTER, FRIB.RATE_WINDOW_CENTER,
FRIB.DELAY_WINDOW_WIDTH, FRIB.RATE_WINDOW_WIDTH to zero.

4.6 FRIB.AUTOCORR_CALIB:

**FRIB.AUTOCORR_CALIB: [NO or CNST1 or CNST2 or SQRT_MEA or
                       SQRT_KOG]**

Correlators apply arbitrary normalization factors that affect
both autocorrelation and cross-correlation PIMA allows to re-normalize
cross-correlation visibilities.

    NO   -- no re-normalization is performed. NB: it is a bad idea to
                 use the data without renormalization for imaging.

    CNST1   -- cross-correlation is divided by a constant
                 PIMA__ACCR_CNST1 defined in pima.i

    CNST2   -- cross-correlation is divided by a constant
                 PIMA__ACCR_CNST2 defined in pima.i

    SQRT_MEA -- PIMA computed re-normalization factors under assumptions
                 that 1) the correlator applied the same normalization
                 factor to both autocorrelation and cross-correlation;
                 2) the mean of the autocorrelation spectrum is 1.

                 The autocorrelation spectrum is corrected for sampling
                 distortion. This is done by transforming the spectrum to
                 the time domain, applying the digital sampler correction
                 that may differ from the digital sampler correction of
                 cross-correlator data since the autocorrelation data have
                 much larger amplitude, normalizing the autocorrelation
                 function to 1 at the zero lag and transforming the
                 corrected autocorrelation data back to the spectral
                 domain. Then the mean autocorrelation spectrum over an
                 IF is computed for each station, and the cross-correlation
                 visibilities are divided by the square root of the
                 product of the mean autocorrelation spectrum for both
                 stations of the baseline.

    SQRT_KOG -- Similar to SQRT_MEA, but the original algorithm of Leonid
                 Kogan used. The author of PIMA considers Kogans's
                 algorithm erroneous, although the error affects the fringe
                 amplitude at the amount of 0.5-3%.

4.7 FRIB.AMPL_FUDGE_TYPE:

FRIB.AMPL_FUDGE_TYPE: [VLBA or DIFX or KOGAN or NO]

   Fudge factor that accounts for disparity of amplitude distortion
of the autocorrelation and cross-correlation. This factor is correlator
dependent.

    VLBA -- the factor that is due to register saturation in VLBA
             hardware correlator. It depends on polarization and

```
               on data weights. NB: if you process the data processed
               with the software correlator located at Socorro you should
               NOT apply this correction.

      KOGAN -- the factor that is due to register saturation in VLBA
               *hardware correlator*. The difference with respect to
               a case VLBA (see above) is that all weights are
               considered 1.0

      DIFX  -- For DiFX correlator: the fudge factor is 1.0

      NO    -- No fudge factor is applied.
```

4.8 FRIB.AMPL_EDGE_WINDOW_COR:

FRIB.AMPL_EDGE_WINDOW_COR: [USE or NO]

```
  There are three factors that reduces fringe amplitude when we observe
a source that is off the main beam. The first factor is due to data loss
at the edge of a frame with raw voltage record due to shifting of data
streams. If the frame has N points, we can cross-multiplies all N points
of both streams when no shift is applied. If one of the streams is
shifted at K points, we can cross-multiply only N-K points and the
amplitude of the average will become less by the (N-K)/N factor.
This factor is linear with respect to the group delay delay and the
amplitude is reduced by the factor of 0.5 at the edge of the natural
search window determined as a quantity reciprocal to the spectral
resolution. The second factor is the power pattern of the primary beam.
The third factor is appearance of the phase acceleration due to
the third derivative of path delay over time, two times, and over
source coordinates.

  This keyword determines whether to apply the amplitude correction
due to the amplitude loss for

 USE -- to apply the amplitude correction for the amplitude loss
        in cross-multiplication of the data streams when due to
        data stream shift. If in doubt, set USE.

 NO  -- do not apply the amplitude correction due to non data stream
        shifts. You should well understand what you are doing when
        you use this option.
```

4.9 FRIB.AMPL_EDGE_BEAM_COR:

FRIB.AMPL_EDGE_BEAM_COR: [YES or NO]

  This keyword specifies whether to apply the amplitude correction
for the antenna beam. Antenna beam is here the ratio of the power
of the signal from a uniformly emitting sphere at angle theta from
the beam direction to the power at tetra=0.

  YES -- to apply the correction for the primary beams of both
       antennas of a baseline. If in doubt, set YES.

  NO -- not to apply the correction for the primary beams of both
       antennas of a baseline. You should well understand what
       you are doing when you use this option.

4.10 FRIB.OVERSAMPLE_MD:

FRIB.OVERSAMPLE_MD: **value**

  Oversampling factor over group delay axis. When PIMA grids the data
for fringe search, it can reduce the step of the grid and pad the
samples between grid points. When no oversampling is used, the
amplitude of the coarse fringe may by less by a factor of 2.5 when
the maximum happens just between grid points. The week sources may be
overlooked. When the oversampling factors are 4 for both group delay
and phase delay rate, the amplitude drop of the coarse fringe search
is only 5%. It should be noted that the amplitude is affected by
the coarse fringe search only. If the sources is detected by the coarse
fringe search, the fine fringe fitting corrects the amplitude
underestimation.

  value -- oversampling factor. General recommendation: 4, unless
       you can tolerate the detection limit drop up to the factor
       of 2.5. If you can tolerate it, set the oversampling
       factor to 1. The factor cannot be less than 1.

4.11 FRIB.OVERSAMPLE_RT:

FRIB.OVERSAMPLE_RT: **value**

  Oversampling factor over phase delay rate axis. See description
of FRIB.OVERSAMPLE_MD. Usually the oversampling over both axes is
applied.

  value -- oversampling factor. General recommendation: 4, unless
       you can tolerate the detection limit drop up to the factor

of 2.5. If you can tolerate it, set the oversampling
factor to 1. The factor cannot be less than 1.

4.12   FRIB.FINE_SEARCH:

FRIB.FINE_SEARCH: **[LSQ or ACC or PAR or BIN or TEC or NO]**

  This keyword specifies the algorithm for fine fringe fitting that
follows the coarse fringe fitting.

  LSQ -- the parameters following parameters are determined with
        least square: group delay, group delay rate, phase delay
        rate, and fringe phase at the fringe reference time and
        the reference frequency. The reference frequency is the
        lowest frequency of the data that are processed.
        Fringe phase for each visibility point is assigned
        according the the a priori SNR. The algorithm
        seeks the variance factor that corrects the a priori weights
        in such a way that the ratio of the weighted sum of the
        postfit residuals to its mathematical expectation be close
        to unity. The algorithm computes these reweighting parameters
        for two cases: multiplicative reweighting and additive
        reweighting, i.e. in the the first case the a posteriori
        weight is the a priori weight multiplied by the reweighting
        parameter, and in the second case the a posteriori weight
        is the a priori weight with the reweighted parameter added
        in quadrature. The algorithm returns three estimates of
        group delay, group delay rate, phase delay rate, phase and
        amplitude of observation-averaged visibilities: without
        reweighting, with multiplicative reweighting and with additive
        reweighting.

  ACC -- the same as LSQ, but phase delay acceleration is evaluated
        instead of group delay rate.

  PAR -- the fine fringe fitting is done by the parabolic fit
        through three points of the delay resolution function around
        the maxima. The search of the maxima is performed separately
        for delay and delay rate. The Lagrange interpolating
        polynomial is computed, and the linear equation which
        equalizes the derivative to zero is solved. The algorithm
        returns the estimates of group delay, phase delay rate, phase
        and amplitude of observation-averaged visibilities.

  BIN -- the fine fringe fitting is done by binary division method.
        The algorithm returns the estimates of group delay, group
        delay rate, phase delay rate, phase and amplitude of
        observation-averaged visibilities.

  TEC -- not implemented

```
  NO  -- no fine fringe fitting is performed.
```

4.13 FRIB.AUTOCORR_THRESHOLD:

FRIB.AUTOCORR_THRESHOLD: **value**

```
   The threshold for autocorrelation. The autocorrelations
that are less than this threshold are discarded. This
results in discarding related cross-correlations if
AUTOCORR_CALIB is SQRT_MEA or SQRT_KOG. Value 0.05
for DiFX correlator is usually adequate.
```

4.14 FRIB.WEIGHTS_THRESHOLD:

FRIB.WEIGHTS_THRESHOLD: **value**

```
   The threshold for visibility weights. The visibility points
with weights less that this threshold are discarded.  Value 0.2
is usually adequate.
```

4.15 FRIB.NOISE_NSIGMA:

FRIB.NOISE_NSIGMA: **value**

```
   This parameter controls computation of the mean amplitude
of noise. The 2d Fourier transform of visibility data over
time an over frequency is analyzed for a search of the signal
from the radio source. The result is analyzed also for determination
of the mean amplitude of the noise. Among the results of the
2D Fourier transform the minimum among 32768 or 1/2 of the
number of elements are randomly selected. The amplitudes
of these random points of the spectrum are sorted in
the ascending order. Then the iterative procedure for
outlier elimination is launched. If the amplitude of the top
point of the list is greater than FRIB.NOISE_NSIGMA times the
root mean square over amplitudes in the list, the point is
discarded, the root means squares is recomputed, and the
procedure is repeated. Outlier elimination is made mainly to
eliminate from statistics computation those points that are
associated with signal rather than this signal.
```

value 4.0 is usually adequate.

4.16 FRIB.SNR_DETECTION

FRIB.SNR_DETECTION: **value**

Sets the threshold for detection. Observations with the SNR
less than this threshold are considered as non-detections.
Task splt discards observations with the SNR less than this
threshold. Task mkdb set flag "non-detection" for observations
with the SNR below this threshold. The detection limit depends
on the number of visibility data used for fringe fitting. PIMA
cannot determine the a priori threshold. It is set by a user.
It is possible to determine the SNR detection limit by analyzing
statistics of residuals from observations of the same type.
The SNR is defined as the ratio of the fringe amplitude to
the mean value of amplitude of noise. If to consider that
real and image part of noise are independent and have Gaussian
distribution with variance \sigma, the mean amplitude is
\sqrt(\pi/2) \approx 1.253 \sigma. The variance of the amplitude
noise is \sqrt(2 - \pi/2} \approx 0.655 \sigma.

4.17 FRIB.FRQ_TRANSFER_BAND:

FRIB.FRQ_TRANSFER_BAND: **value**

Controls how phase transfer from one frequency of simultaneous
observations to another should be performed by tasks frtr.

        NO   -- no phase transfer is made

        control_file -- name of the secondary control file for the
                        lower band of this VLBI experiment. That
                        control file determines the frequency band
                        whether fringe phases should be taken
                        for the phase transfer to the frequencies
                        of the band specified in the main control
                        file.

4.18 FRIB.FRQ_TRANSFER_METHOD:

FRIB.FRQ_TRANSFER_METHOD: **method**

Defines the smoothing method applied to phase before frequency
transfer.

```
SPLINE    -- phase at the lower band is expanded in B-spline basis
             with the number of knots specified by keyword
             FRIB.FRQ_TRANSFER_DEG

LEGENDRE -- phase at the lower band is expanded in Legendre basis
            with the degree specified by keyword
            FRIB.FRQ_TRANSFER_DEG

ASIS     -- phase at lower band is not expanded and transferred
            as is.
```

4.19 FRIB.FRQ_TRANSFER_DEG:

FRIB.FRQ_TRANSFER_DEG: degree

Defines the degree of the polynomial or the number of knots of
B-spline for smoothing phase before frequency transfer.

```
degree -- The meaning of this keyword depends on
          FRIB.FRQ_TRANSFER_METHOD. If the expansion basis is
          LEGENDRE, then FRIB.FRQ_TRANSFER_DEG is the
          degree of the Legendre polynomial for expansion of
          the phase at the low band. If the expansion basis is
          SPLINE, then FRIB.FRQ_TRANSFER_DEG is the number of
          spline knots in each intermediate frequency for
          expansion of the amplitude of the phase at the low
          band. It is ignored if method is ASIS.
```

4.20 FRIB.FRQ_TRANSFER_MSEG:

FRIB.FRQ_TRANSFER_MSEG: segment_length

Defines the length of the segment that will be coherently averaged
before frequency transfer.

```
segment_length -- The averaging factor for the phase at low band
                  to be used for interpolation before transfer.
                  An integer number in a range [1, NCHN], where
                  NCHN is the number of spectral channels in
                  an IF, or AUTO. Value AUTO forces PIMA to
```

determine segment length in order to have
the SNR within a segment no less than 1.5

4.21   FRIB.2D_FRINGE_PLOT:

FRIB.2D_FRINGE_PLOT: [XW or GIF or PS or TXT or NO]

   This keyword specifies the format of the 2D fringe
plot. The 2D fringe plots shows the amplitude of the visibility
averaged over frequency and time within the observation as
a function of group delay and phase delay rate. The name of
the output file is
fr2d_{scan_name}_{band_name}_{station1}_{station2}.
All components are in lower case. The plot is placed in
subdirectory {EXPER_DIR}/{SESS_CODE}_fpl/ where EXPER_DIR is
the experiment scratch directory and {SESS_CODE} is the session
code.

   XW -- to display the plot of the 2D function on the screen
          using the PGPLOT library.

   GIF -- to generate the plot in GIF format. Extension of the output
          file is .gif

   PS -- to generate the plot in Postscript format. Extension of the
          output file is .ps

   TXT -- to write the table with the averaged amplitude as a function
          of group delay and phase delay rate. Extension of the output
          file is .txt

   NO -- do not generate the 2D fringe file output.

4.22   FRIB.PLOT_DELAY_WINDOW_WIDTH:

FRIB.PLOT_DELAY_WINDOW_WIDTH: value

   This parameter sets the width of the 2d fringe plot along
group delay axis. Units are sec.

4.23   FRIB.PLOT_RATE_WINDOW_WIDTH:

FRIB.PLOT_RATE_WINDOW_WIDTH: value

   This parameter sets the width of the 2d fringe plot along
phase delay rate axis. Units are sec.

4.24 FRIB.OVERSAMPLE_PLOT_MD:

FRIB.OVERSAMPLE_PLOT_MD: value

   This parameter sets the oversampling factor along the group delay
axis for generation of the 2D fringe plot. Natural choice is '
FRIB.OVERSAMPLE_PLOT_MD: 1. If to increase the value, the lines
will become thicker.

4.25 FRIB.OVERSAMPLE_PLOT_RT:

FRIB.OVERSAMPLE_PLOT_RT: value

   This parameter sets the oversampling factor along the phase delay
rate axis for generation of the 2D fringe plot. Natural choice is
FRIB.OVERSAMPLE_PLOT_RT: 1. If to increase the value, the lines will
become thicker.

4.26 FRIB.1D_RESFRQ_PLOT:

FRIB.1D_RESFRQ_PLOT: [XW or GIF or PS or SAV or TXT or NO]

   This keyword specifies the format of the 1D fringe plot that shows
amplitude and phase of the visibilities averaged over time as
a function frequency. The name of the output file is
fr1d_frq_{scan_name}_{band_name}_{station1}_{station2}_all.
All components are in lower case. The plot is placed in
subdirectory {EXPER_DIR}/{SESS_CODE}_fpl/ where EXPER_DIR is
the experiment scratch directory and {SESS_CODE} is the session
code.

   XW -- to display the 1D plot of the fringe phase and fringe
          amplitude on the screen using the PGPLOT library.

   SAV -- to write the plot in the internal format of graphic

```
library DiaGI. Extension of the output file is .sav
The plot can be displayed with program diagi_rst that
accepts the name of the plot file in SAV format as
its argument.

GIF -- to generate the plot in GIF format. Extension of the output
       file is .gif

PS  -- to generate the plot in Postscript format. Extension of the
       output file is .ps

TXT -- to write the table with fringe phase and fringe amplitude
       as a function of frequency. Extension of the output
       file is .txt

NO  -- do not generate the 1D fringe plot versus frequency output.
```

4.27 FRIB.1D_FRQ_MSEG:

FRIB.1D_FRQ_MSEG: **value**

```
   The averaging factor for 1D fringe plot over frequency.
An integer number in a range [1, NCHN], where NCHN is the number
of spectral channels in an IF. If FRIB.1D_FRQ_MSEG > 1, then
the complex visibility is averaged over FRIB.1D_FRQ_MSEG consecutive
spectral channels, and the number of points in the plot will be
reduced by the same factor. Frequency averaging is used for displaying
fringe plot of a weak source.
```

4.28 FRIB.1D_RESTIM_PLOT:

FRIB.1D_RESTIM_PLOT: **value**

```
   This keyword specifies the format of the 1D fringe plot that shows
amplitude and phase of the visibilities averaged over frequency as
a function time. The name of the output file is
fr1d_tim_{scan_name}_{band_name}_{station1}_{station2}_all.
All components are in lower case. The plot is placed in
subdirectory  {EXPER_DIR}/{SESS_CODE}_fpl/ where EXPER_DIR is
the experiment scratch directory and {SESS_CODE} is the session
code.

  XW  -- to display the 1D plot of the fringe phase and fringe
         amplitude on the screen using the PGPLOT library.
```

```
SAV -- to write the plot in the internal format of graphic
       library DiaGI. Extension of the output file is .sav
       The plot can be displayed with program diagi_rst that
       accepts the name of the plot file in SAV format as
       its argument.

GIF -- to generate the plot in GIF format. Extension of the output
       file is .gif

PS  -- to generate the plot in Postscript format. Extension of the
       output file is .ps

TXT -- to write the table with fringe phase and fringe amplitude
       as a function of frequency. Extension of the output
       file is .txt

NO  -- do not generate the 1D fringe plot versus time.
```

4.29 FRIB.1D_TIM_MSEG:

FRIB.1D_TIM_MSEG: **value**

```
  The averaging factor for 1D fringe plot over time.
An integer number in a range [1, NAP], where NAP is the number
of accumulation periods. If FRIB.1D_TIM_MSEG > 1, then
the complex visibility is averaged over FRIB.1D_TIM_MSEG consecutive
accumulation periods, and the number of points in the plot will be
reduced by the same factor. Frequency averaging is used for
displaying fringe plot of a weak source.
```

4.30 FRIB.1D_DRF_PLOT:

FRIB.1D_DRF_PLOT: **value**

```
  This keyword specifies the format of the 1D plot of the delay
resolution function -- dependence of the fringe amplitude averaged
over frequency and time on group delay. The argument of the DRF is
group delay with respect to the value found by the fringe fitting
procedure. The name of the output file is
fr1d_drf_{scan_name}_{band_name}_{station1}_{station2}_all.
All components are in lower case. The plot is placed in
subdirectory  {EXPER_DIR}/{SESS_CODE}_fpl/ where EXPER_DIR is
the experiment scratch directory and {SESS_CODE} is the session
code.
```

```
XW  -- to display the 1D plot of the fringe phase and fringe
       amplitude on the screen using the PGPLOT library.

SAV -- to write the plot in the internal format of graphic
       library DiaGI. Extension of the output file is .sav
       The plot can be displayed with program diagi_rst that
       accepts the name of the plot file in SAV format as
       its argument.

GIF -- to generate the plot in GIF format. Extension of the output
       file is .gif

PS  -- to generate the plot in Postscript format. Extension of the
       output file is .ps

TXT -- to write the table with fringe phase and fringe amplitude
       as a function of frequency. Extension of the output
       file is .txt

NO  -- do not generate the 1D delay resolution function plot.
```

4.31 FRIB.1D_DRF_SPAN:

FRIB.1D_DRF_SPAN: **value**

  This parameter sets the half-width of the interval of the argument
for a delay resolution plot. Units are sec.

5 Bandpass processing keywords

5.1 BPS.MODE:

BPS.MODE: **value**

5.2 BPS.MODE:

BPS.MODE: **[INIT or ACCUM or FINE]**

The mode of the complex bandpass computation procedure. This
keyword specifies the stage where bandpass computation should stop.
There are three stages for bandpass calculation.

    INIT -- to run the bandpass in the init mode. PIMA analyzes the
                input file with the fringe fitting results and finds
                the observations with the maximum SNR for each baseline
                with the reference station. NB: PIMA obeys INCLUDE_OBS:
                and EXCLUDE_OBS: keywords and checks only those observations
                that satisfies the filter. PIMA computes the visibilities
                averaged over time with parameters of fringe fitting
                applies. These time-averaged visibilities are a function of
                frequency. The phase of the complex bandpass is the residual
                phase of the visibilities after subtraction the time and
                frequency averaged phase over the observation. The amplitude
                of the bandpass is the amplitude of time-averaged visibility
                after dividing it over the amplitude averaged over time and
                frequency within the IF to which the individual point of the
                bandpass belongs. The bandpass can be considered as the
                averaged visibility. The normalization for phase is computed
                by averaging over all IFs, and normalization for amplitude
                is computed by averaging over individual IFs. In the case of
                dual-polarization data when POLAR: I is specified, the
                procedure is repeated twice: first for the RR polarization
                bandpass second for the LL polarization with respect to the
                 RR polarization data.

    ACCUM -- to run bandpass in the accumulation mode. After computation
                the bandpass in the init mode, the bandpass is applied to
                K observations at every baseline with the reference station.
                PIMA selects these observations as the ones that have the
                greatest SNR.

    FINE -- to run bandpass in the fine mode. After computation the
                bandpass in the accumulation mode, the bandpass is re-computed
                using least squares. Accumulation bandpass is applied.
                Correction to the accumulation bandpass is computed by
                estimating parameters of the phase and amplitude bandpass
                using M observations with the highest SNR among all
                baselines with the references station. Estimated parameters
                for the phase part are coefficients of expansion over Legendre
                polynomials of B-spline basis. Estimated parameters for
                the amplitude part are coefficients of expansion of the
                logarithm of the residual amplitude ratios into either
                Legendre polynomials or the B-spline basis. After computing
                a solution, observations are checked for outliers. If the
                residual phases or amplitudes exceed a certain threshold, the
                observation is marked as outlier and the bandpass is recomputed.
                The procedure is repeated till either no outlier is found.
                If at a given baseline the number of remaining observations
                fell to the specified threshold Q, no more outliers at that
                baseline is eliminated.

5.3 BPS.NOBS_ACCUM:

BPS.NOBS_ACCUM: value

   This keyword specified how many observations with the highest SNR
at each baseline are taken for computation of the accumulation bandpass.

5.4 BPS.MSEG_ACCUM:

BPS.MSEG_ACCUM: value

   This keyword specifies how many adjacent spectral channels are
coherently averaged during phases of the initial and accumulation
bandpass calculation. Cannot be less than 1 or exceed the number
of spectral channels in the IF.

5.5 BPS.NOBS_FINE:

BPS.NOBS_FINE: value

   This keyword specified how many observations with the highest SNR
at each baseline are taken for computation of the accumulation bandpass.

5.6 BPS.MINOBS_FINE:

BPS.MINOBS_FINE: value

   The keyword specified the minimum number of observation at each
baseline that should remain during bandpass computation in the fine
mode. This parameter limits the number of outliers rejected.

5.7 BPS.MSEG_FINE:

BPS.MSEG_FINE: value

This keyword specifies how many adjacent spectral channels are coherently averaged during phases of the fine bandpass calculation. Cannot be less than 1 or exceed the number of spectral channels in the IF.

5.8 BPS.SNR_MIN_ACCUM:

BPS.SNR_MIN_ACCUM: value

This keyword specifies the minimum SNR for an observation to be eligible for being used for bandpass computation in accumulation mode. Observations with the SNR less than this limit are not considered. If the SNR is too small, there is a chance that the bandpass computation procedure will derail. If there are no high SNR observations in the experiment, parameter BPS.ACCUM_MSEG should be raised.

5.9 BPS.SNR_MIN_FINE:

BPS.SNR_MIN_FINE: value

This keyword specifies the minimum SNR for an observation to be eligible for being used for bandpass computation in fine mode. Observations with the SNR less than this limit are not considered.

5.10 BPS.DECOR_TIM_MIN:

BPS.DECOR_TIM_MIN: value

This keyword specifies the minimum time decorrelation value computed by the fringe fitting for the observation to be eligible for being used for bandpass computation in any mode. It can be in a range of 0 to 1. DECOR_TIM is the ration of the correlated amplitude coherently averaged over the scan to the arithmetic average (i.e. the incoherent average) of the correlated amplitude computed over elementary segments used by the procedure of fine fringe fitting. DECOR_TIM is close to 1 for a perfect observation. Phase variations due to atmosphere of frequency standard degrades DECOR_TIM. This degradation is harmless for bandpass calculation. However, DECOR_TIM may be reduced due to catching artificial signal, either internal or external RFI. The purpose of this filter is to prevent using observations affected by RFI. Recommended value: 0.8 at frequencies below 15 GHz and 0.5 at frequencies above 15 GHz. Value 0.0 disables this filter entirely.

5.11 BPS.AMPL_REJECT:

BPS.AMPL_REJECT: value

  The threshold for the residual amplitude after applying bandpass
in the fine mode for being marked as the outlier. The observation
is marked as the outlier if the the rms of the deviation of the
normalized amplitude from 1.0 in any intermediate frequency
exceeds BSP.AMPL_REJECT.

5.12 BPS.PHAS_REJECT:

BPS.PHAS_REJECT: value

  The threshold for the residual phase after applying bandpass
in the fine mode for being marked as an outlier. Unit: radian.
The observation is marked as an outlier if the the rms of the
residual phase in any intermediate frequency is greater than
BSP.PHAS_REJECT. Typical value is 0.4 rad.

5.13 BPS.INTRP_METHOD:

BPS.INTRP_METHOD: [LEGENDRE or SPLINE]

  The keyword specifies the basis for expansion the bandpass.

  LEGENDRE --Legendre polynomial

  SPLINE   -- B-spline of the 3rd degree.

5.14 BPS.DEG_AMP:

BPS.DEG_AMP: value

  The meaning of this keyword depends on BPS.INTRP_METHOD.
If the expansion basis is LEGENDRE, then BPS.DEG_AMP is the

degree of the Legendre polynomial for expansion of the
amplitude of the bandpass. If the expansion basis is SPLINE,
then BPS.DEG_AMP is the number of spline knots in each
intermediate frequency for expansion of the amplitude of the
bandpass.

5.15 BPS.DEG_PHS:

BPS.DEG_PHS: value

The meaning of this keyword depends on BPS.INTRP_METHOD.
If the expansion basis is LEGENDRE, then BPS.DEG_PHS is the
degree of the Legendre polynomial for expansion the phase
of the bandpass. If the expansion basis is SPLINE, then BPS.DEG_AMP
is the number of spline knots in each intermediate frequency
for expansion of the phase of the bandpass.

5.16 BPS.NORML:

BPS.NORML: [NO or IF or BAND]

This keywords specifies the way how the bandpass normalization
is made.

   NO   -- No normalization is applied.

   IF   -- The bandpass is normalize to have the mean amplitude
           1.0 over each individual intermediate frequency.
           This is a typical choice.

   BAND -- The bandpass is normalized to have the mean amplitude 1.0
           over the band, i.e. over all intermediate frequencies.

5.17 BPS.SEFD_USE:

BPS.SEFD_USE: value

Not used. Should be NO

6 Phase reference fringe fitting keywords

6.1 FRIP.SCAN_FILE:

FRIP.SCAN_FILE: **file or NO**

   Not implemented. Should be NO.

6.2 FRIP.STA_INC_FILE:

FRIP.STA_INC_FILE: **file or NO**

   Name of the file that with station names. Should be NO.

6.3 FRIP.STA_EXC_FILE:

FRIP.STA_EXC_FILE: **file or NO**

   Not implemented. Should be NO.

6.4 FRIP.STA_REFS:

FRIP.STA_REFS: **value**

   Not implemented. Should be SAME.

6.5 FRIP.RESOLUTION:

FRIP.RESOLUTION: **value**

   Not implemented. Should be 4096.

6.6 FRIP.OVERSAMPLE:

```
FRIP.OVERSAMPLE:                   value
```

   Not implemented. Should be 1.

6.7 FRIP.SCA:

```
FRIP.SCA:                          value
```

   Not implemented. Should be ALL.

6.8 FRIP.MAP_DIR:

```
FRIP.MAP_DIR:                      directory or NO
```

   Not implemented. Should be SAME.

6.9 FRIP.ATM_ZEN_FILE:

```
FRIP.ATM_ZEN_FILE:                 file or NO
```

   Not implemented. Should be NO.

6.10 FRIP.CAL_PLOT:

```
FRIP.CAL_PLOT:                     value
```

   Not implemented. Should be GIF.

6.11 FRIP.CAL_RES:

```
FRIP.CAL_RES:                      value
```

Not implemented. Should be 400.

6.12 FRIP.TAG_PLOT:

FRIP.TAG_PLOT: **value**

Not implemented. Should be GIF.

6.13 FRIP.TAG_RES:

FRIP.TAG_RES: **value**

Not implemented. Should be 400.

6.14 FRIP.BEAM_PLOT:

FRIP.BEAM_PLOT: **value**

Not implemented. Should be GIF.

6.15 FRIP.FRQ_MSEG:

FRIP.FRQ_MSEG: **value**

Not implemented. Should be 128.

6.16 FRIP.TIM_MSEG:

FRIP.TIM_MSEG: **value**

Not implemented. Should be 1.

6.17 FRIP.RA_CENTER:

FRIP.RA_CENTER: **value**

Not implemented. Should be APRIORI.

6.18 FRIP.DEC_CENTER:

FRIP.DEC_CENTER: **value**

Not implemented. Should be APRIORI.

6.19 FRIP.RA_STEP:

FRIP.RA_STEP: **value**

Not implemented. Should be 0.1.

6.20 FRIP.DEC_STEP:

FRIP.DEC_STEP: **value**

Not implemented. Should be 0.1

6.21 FRIP.RA_RANGE:

FRIP.RA_RANGE: **value**

Not implemented. Should be 1.0.

6.22 FRIP.DEC_RANGE:

FRIP.DEC_RANGE: **value**

Not implemented. Should be 1.0.

7 Splt keywords

Operation splt gathers all visibilities for the specified source(s),
applies all calibrations, transforms baseline dependent group delays,
phase delay rates, and group delay rates from baseline dependent
quantities to station based quantities, rotates visibility phases for
results of fringe fitting, averages visibilities over time and frequency,
applies specified re-normalizations, and writes averaged visibilities
in output FITS files, one file per sources, in a form suitable for
imaging using software DIFMAP.

7.1 SPLT.SOU_NAME:

SPLT.SOU_NAME: **[Bname or Jname or ALL]**

Name of the source that is to be processed. All visibilities
related to that sources subject of constraint in INCLUDE_OBS_FILE,
EXCLUDE_OBS_FILE, OBS, and SNR_DETECTION_LIMIT keywords are
considered.

Bname -- B1950 8-character long source name.

Jname -- J2000 10-character long source name.

ALL   -- operation splt is performed in a cycle over all source
         names.

7.2 SPLT.FRQ_MSEG:

SPLT.FRQ_MSEG: **value**

The keyword specifies the number of spectral adjacent channels that will
averaged. It should be no less than 1. Value 1 means no frequency averaging.
The value should be exceed the number of channels in the intermediate
frequency. Typical choice: the total number of spectral channels in the IF.
In that case the number of IFs in the output FITS file is the same as the
number of used IFs in the input data. If SPLT.FRQ_MSEG is less than the number

of spectral channels in the intermediate frequency, the number of IFs in the
output file will be greater than the number of IFs in the input data.
PIMA does not support hierarchy spectral channel+IF in the output FITS files.
The number of spectral channels is always 1 in the output file.

7.3 SPLT.TIM_MSEG:

SPLT.TIM_MSEG: **value**

  The keyword specifies the number of adjacent accumulation periods for
time averaging. Value 1 means no time averaging. IF SPLT.TIM_MSEG exceeds
the number of accumulation periods in the scan, all accumulation periods
are averaged.

7.4 SPLT.WEIGHT_TYPE:

SPLT.WEIGHT_TYPE: [ONE, OBS_SNR, OBS_MS, SEG_RMS, AUTO]

  This keywords species the method for computing visibility
weights in the output FITS file.

    ONE   -- all weights are unity.

    OBS_SNR -- Calibrated amplitude is computed over the observation.
               The SNR for a given IF is computed based on the SNR
               over the observation. Rms of noise is computed as
               a ratio of calibrated amplitude to the SNR over the
               IF. Weight is reciprocal to the square of noise
               rms. Thus, the weight is computed for a given IF
               using all accumulation periods of the observation.

    OBS_RMS -- Weighted variance of calibrated visibilities is computed
               for a given IF using all accumulation periods of a given
               observation. Weight is reciprocal to the variance.

    SEG_RMS -- Weighted variance of calibrated visibilities is computed
               for a given IF using all accumulation periods over
               a given segment. Weight is reciprocal to the variance.

    AUTO   -- If the number of accumulation periods per segment is
               greater of equal than the threshold (currently 8),
               WEIGHT_TYPE AUTO is equivalent to WEIGHT_TYPE: SEG_RMS.
               If the number of accumulation periods per segment is
               less than the threshold (currently 8), weighted variance
               of calibrated visibilities is computed over so-called
               "weights segments". These weight segments are longer than

the output visibility segments. Weight for a given IF,
given segment is reciprocal to the variance of the weight
segment with the middle epoch nearest to the to the epoch
of the given segment.

7.5 SPLT.POLAR:

SPLT.POLAR: [RR or LL or RL or LR or I or PAR or ALL]

  This keyword specifies for which polarizations the visibilities
should be in the output FITS file. If the input data had single
polarization then SPLT.POLAR value should be that polarization
code. In a case of dual polarization the choice is

   RR -- RR-polarization only.

   LL -- LL-polarization only.

   RL -- RL-polarization only.

   LR -- LR-polarization only.

   I -- I-polarization only. The I-polarization data are computed
         on the fly.

  PAR -- RR-polarization and LL-polarization data. This is a usual
         choice when no cross-polarization analysis is intended.

  ALL -- all polarization data present in the input FITS files
         are exported to the output FITS file.

7.6 SPLT.AUTOCORR_NRML_METHOD:

SPLT.AUTOCORR_NRML_METHOD: [AVERAGED or NO]

  This keyword specifies whether to apply the renormalization factor
for system temperature due to discarding some spectral channels.
The system temperature is measured over entire intermediate frequency.
However, in general, the spectrum of noise is not constant over
the band. PIMA uses autocorrelation function to compute the factor of
Tsys(used)/Tsys(tot), where Tsys(used) is the system temperature over
the used portion of the bandwidth, and Tsys(tot) is the Tsys over
the entire IF, i.e. measured Tsys. The portion of the bandwidth for
renormalization is defined by [ICHN_1ST, ICHN_LAST] range *and* the
band mask.

```
AVERAGED -- to apply the system temperature to renormalization due
            to discarded spectral channels.

NO       -- do not apply the system temperature to renormalization
```

7.7 SPLT.BPASS_NRML_METHOD:

SPLT.BPASS_NRML_METHOD: [WEIGHTED or NO]

  This keywords specifies whether to apply the bandpass renormalization
factors for given intermediate frequencies using only a part of the
bandwidth. Initially, the bandpass is normalized to unity over the
entire bandwidth of the IF. However, often decorrelation occurs at the
edges of the band. PIMA allows to specify the portion of the band that
is considered "representative". It is expressed in parameters Bl, Bh
that stands for bandwidth low range and bandwidth high range that are
specified in SPLT.BPASS_NRML_RANGE. The representative bandwidth is
[F_low + Bl*Fw, F_low*Bh*fw]. Renormalization factor
R = (sum B_r/N_r) / Sum B_t/N_t, where B_r -- bandpass in the bandwidth
is [F_low + Bl*Fw, F_low*Bh*fw], N_r -- the number of points in that
bandwidth; B_t bandpass in the total bandwidth, N_t the total number of
points in the entire IF. The IF-dependent factors scale the bandpass
to make makes its normalized over the representative bandwidth within
a given IF [F_low + Bl*Fw, F_low*Bh*fw]. Usually, the factor is less
than 1.0. If unsure, use WEIGHTED.

```
WEIGHTED -- to apply the bandpass renormalization.

NO       -- not to apply the bandpass renormalization.
```

7.8 SPLT.BPASS_NRML_RANGE:

SPLT.BPASS_NRML_RANGE: low:high

  These keyword specifies the range of the bandwidth as a share of
the nominal bandwidth that are considered representative for bandpass
renormalization. The parameters of the range are between [0,1].
The parameter is discarded when SPLT.BPASS_NRML_METHOD: NO.
If unsure, specify 0.2:0.8

```
low  -- the low frequency of the IF that belongs to the
        representative range. Should be in a range [0, 1).

high -- the high frequency of the IF that belongs to the
```

        representative range. Should be in a range (0,1].
        High should be greater than low.

Example: let the IF frequency range 32 MHz, the number of spectral
channels is 256. If SPLT.BPASS_NRML_RANGE: 0.2:0.75, then
the low frequency of the representative range is 0.2*32 = 6.4 MHz,
the first spectral channel within the range is 52. The high
frequency of the representative range is 0.75*32 = 24 MHz,
the last spectral channel within the range is 192. Thus the bandpass
renormalization will be done in such a way that the mean bandpass
over spectral channels with indices 52–192 be unity.

7.9   SPLT.SUBARRY_CONSOLIDATION:

SPLT.SUBARRY_CONSOLIDATION: [NO or MIN or MAX]

   Since PIMA performs fringe fitting in the baseline mode, an individual
scan may not have detections less than N*(N+1)/2 baselines of a N-station
network. PIMA designate a subnetwork with detections at a given scan
as an subarray. Normally PIMA unites observations of different scans
with the same subnetwork in a subarray. PIMA computes station–based
visibility phases for each subarray individually. Keyword
SPLT.SUBARRY_CONSOLIDATION controls how subarrays are consolidated after
initial process for converting baseline-depended fringe phases into
station–based fringe phases.

   NO   -- means to disable subarray consolidation.

   MIN -- instructs PIMA to preform minimal subarray consolidation:
           if all stations of subarray A are present in the subarray B,
           then the subarray B is consolidated with subarray A.

   MAX -- instructs to preform maximum subarray consolidation: if a
           subarray has at least one common station with subarray B, both
           subarrays are consolidated.

7.10   SPLT.TOTAL_UV:

SPLT.TOTAL_UV: [NO or YES]

   Specifies whether or not to generate output files with the total
visibilities averaged over frequency and over all accumulation period
over the scan after applying phase rotation for the results of fringe
fittings. The total visibilities are referred to the band reference

frequency. If YES is specified, the files with total visibilities
are written in the same output directories averaged visibilities,
in addition to them.

7.11 SPLT.TOTAL_UV:

SPLT.SNR_MIN: [value]

   Specifies the minimum SNR over the entire scan for the data to
be used by splt to export to the output fits-file. The data
from the observation with the SNR less than that limit are not
exported.

   The SNR is defined as the ratio of the fringe amplitude to
the mean value of amplitude of noise. If to consider that
real and image part of noise are independent and have Gaussian
distribution with variance σ, the mean amplitude is
$\sqrt(\pi/2) \approx 1.253 \sigma$. The variance of the amplitude
noise is $\sqrt(2 - \pi/2} \approx 0.655 \sigma$.

7.12 SPLT.GAIN_CORR_FILE:

SPLT.GAIN_CORR_FILE: [NO or file]

   Specifies the name of the gain correction file. The gain correction file
contains empirical gain corrections. This keyword is used by tasks splt and
gaco.

   NO   -- no gain correction will be applied

   file -- name of the gain correction file. It consists of records of fixed
           lengths in plain ascii. Each record defines gain correction, its
           formal errors and the number of observations used to derived for
           a specific station and specific IF. Task splt checks whether
           keyword SPLT.GAIN_CORR_FILE defines a file. If it does it multiplies
           visibility by the product of two gain corrections for each station
           of a baseline.

7.13 SPLT.STA_BASED:

SPLT.STA_BASED: [YES or ALL or NO]

This keyword specifies whether the station based or baseline based algorithm for computation of averaged visibilities will be used. Usually, station-based algorithm is required for imaging since baseline based algorithm does not preserve phase misclosure. However, station based algorithm prevents putting observations with less than three baselines in the output FITS-IDI file. Therefore, sometimes it is desirable to have a possibility to write the averaged visibilities split into sources even of they belong to a subarray with only two stations.

    YES -- to use a station-based algorithm for computing averaged
           visibilities. All observations that passed input filter
           controlled by keywords INCLUDE_OBS_FILE, EXCLUDE_OBS_FILE,
           OBS, and SNR_DETECTION_LIMIT are process for computation
           of station-based quantities. The same observations are
           used for generating the output.

    ALL -- to use a station-based algorithm for computing averaged
           visibilities. All observations that passed input filter
           controlled by keywords INCLUDE_OBS_FILE, EXCLUDE_OBS_FILE,
           OBS, and SNR_DETECTION_LIMIT are process for computation
           of station-based quantities. All observations between
           the station for which station-based group delays, phase
           delay rates, and group delays are used for generating
           the output **regardless** whether they passed the input
           filter or not. If unsure, specify ALL.

    NO -- to use a baseline-based algorithm. Suitable mainly for
           processing single-baseline data.

8   ONOF processing keywords

   PIMA task onof computes good start and stop time for every scan. This routine is used for mitigation of incorrect on/off time written by the field system in logs. The task finds the time interval at the beginning and the end of the scan with the fringe amplitude below the threshold and the flags accumulation periods that fits criteria of "not on source" interval.

8.1   ONOF.GEN_FLAGS_MODE:

ONOF.GEN_FLAGS_MODE: [NO, CREATE, UPDATE]

    NO   -- onof procedure is not running

```
CREATE -- original flagging status for a given observation is set
          to 1.0, i.e. unflag.

UPDATE -- original flagging status for a given observation is
          honored. Onof does not see flagged scans and preserves
          their flag value.
```

8.2 ONOF.KERNEL_START_SHARE:

ONOF.KERNEL_START_SHARE: value

```
value -- The start epoch of the kernel interval defined
         as an offset with respect to the nominal start
         time as a share of the scan length. Should be
         in a range [0, 1]. For instance if the scan
         length is 100s, and ONOF.KERNEL_START_SHARE: 0.2,
         then the start epoch of the kernel is 20s.
```

8.3 ONOF.KERNEL_END_SHARE:

ONOF.KERNEL_END_SHARE: value

```
value -- The send epoch of the kernel interval defined
         as an offset with respect to the nominal start
         time as a share of the scan length. Should be
         in a range [0, 1]. For instance if the scan
         length is 200s, and ONOF.KERNEL_END_SHARE: 0.9,
         then the start epoch of the kernel is 180s.
```

8.4 ONOF.COHERENT_INTERVAL:

ONOF.COHERENT_INTERVAL: value

```
value -- Interval of coherency in seconds used for
         computation of mean amplitude. In a case
         if the kernel interval is longer than the
         coherency interval, the mean and wrms of
         visibilities is computed over the shorter
         coherency interval. If unsure, set it to 1200.
```

8.5 ONOF.COHERENT_INTERVAL:

ONOF.AMPL_THRESHOLD: value

```
value -- Threshold criterion. An AP with amplitude
         times ONOF.AMPL_THRESHOLD of the mean
         amplitude over the scan kernel is considered
         as a candidate for flagging.
         ONOF.AMPL_THRESHOLD: 0.0 means no amplitude
         threshold criterion is checked.
```

8.6   ONOF.NSIG_THRESHOLD:

ONOF.NSIG_THRESHOLD: value

```
value -- N-sigma criterion. An AP with amplitude
         ONOF.AMPL_THRESHOLD times wrms of the visibility
         over the scan kernel is considered as a candidate
         for flagging. ONOF.NSIG_THRESHOLD: 0.0 means
         no amplitude n-sigma criterion is checked.
```

8.7   ONOF.MIN_LOW_AP:

ONOF.MIN_LOW_AP: value

```
value: 0 means that the AP that satisfied flagging
       criteria ONOF.AMPL_THRESHOLD or ONOF.NSIG_THRESHOLD
       is flagged out. If ONOF.MIN_LOW_AP > 0, then
       onof checks how many consecutive APs have low
       amplitude. If ONOF.MIN_LOW_AP APs has low
       amplitude, then the rest of the scan portion
       (before the kernel interval or after kernel
       interval) is flagged out. If after k APs with
       low amplitude ( k < ONOF.MIN_LOW_AP ) follows
       an AP with high amplitude, these k APs are not
       flagged.
```

9   Creation of the output database keywords

```
The fringe fitting procedure writes results in an ascii file.
Task mkdb computes total group delay, phase delay rate, group
delay rates and a number of auxiliary quantities using the
results of the fringe fitting. The results of task are written
```

in either plain ascii format or in GVF format that is native
for geodesy/astrometry VLBI processing software VTD/post-Solve.

9.1 MKDB.OUTPUT_TYPE:

MKDB.OUTPUT_TYPE: [TEXT or GVF or AMPL]

  Format of the output file generated by task mkdb.

  TEXT -- total group path delays, phase delay rates and other
          quantities are written in a table in plain ascii format,
          one line per used observation.

  GVF -- total group path delays, phase delay rates and other
          quantities are written in a binary GVF format suitable
          for processing with geodesy/astrometry software
          VTD/post-Solve.

  AMPL -- fringe amplitude, fringe phase, Tsys, gain, uv baseline
          projections and other parameters are written in a plain
          ascii table, one line per used observation.

9.2 MKDB.SRT:

MKDB.SRT: [MID_SCAN or SRT_FRT or file]

  This keyword controls the logic for computation of scan reference
time (SRT). Since PIMA processes all observations independently,
in general, fringe reference time of observations of the same scan
is not the same. Group delays, phase delay rates, and fringe phases
of the same scan, the same subarray are referred to a common epoch
that is called scan reference time. PIMA computes this scan reference
time in such a way that the errors of group delays is minimized.
If unsure, use MID_SCAN.

  MID_SCAN -- PIMA finds the scan reference time near in the middle
              of a scan that. In a simplest case when all stations
              started and ended a scan at the same time and there
              were no data losses, the scan reference time is the
              middle scan epoch rounded to 1 sec. Computation of scan
              reference time is not trivial if good data started
              and ended at different stations at different epochs.
              First PIMA finds the interval of time of valid
              accumulation periods (AP) over all used observations
              of a scan. If it finds observations that are not in
              common range, it splits the scan into subarrays that

are treated as separate scans. Then within each subarray
it finds for each observation the range of time for
which the formal uncertainty of group delay increases
by no more than either an additive parameter
MKDB.GD_MAX_ADD_ERROR and by no more than scaling factor
MKDB.GD_MAX_SCL_ERROR. If it finds observations that are
not in common range, it splits the scan into subarrays
that are treated as separate scans. Finally, for each
remaining subarray PIMA assigns the scan reference time
that is in the middle of the common range, subject of
constraints imposed on allowed increase of group delay
formal uncertainty. The scan reference time is rounded
to the nearest second.

SRT_FRT -- scan reference time is the same as fringe reference time.
           If this value is specified, the scan reference time
           may be different for each observation.

file   -- name of the file with scan reference time. PIMA allows
           to compute the best fringe reference time externally
           and import it using MKDB.SRT keyword.

9.3 MKDB.GD_MAX_ADD_ERROR:

MKDB.GD_MAX_ADD_ERROR: value

  Parameter MKDB.GD_MAX_ADD_ERROR controls MID_SCAN algorithm for
setting the scan reference time. When the reference time epoch
is changed, the uncertainty of group delay is increased.
MKDB.GD_MAX_ADD_ERROR controls the tolerance to this increase
as an addition to the group delay uncertainty at fringe reference
time. Units: sec. If unsure, use 5.D-15

9.4 MKDB.GD_MAX_SCL_ERROR:

MKDB.GD_MAX_SCL_ERROR: value

  Parameter GD_MAX_SCL_ERROR controls MID_SCAN algorithm for
setting the scan reference time. When the reference time epoch
is changed, the uncertainty of group delay is increased.
GD_MAX_SCL_ERROR controls the tolerance to this increase
as a fraction pf the group delay uncertainty at fringe reference
time. Units: dimensionless. If unsure, use 0.2. Value 0.2
means that 20% increase of formal uncertainty due to a change

of the scan reference time with respect to the fringe reference
time is allowed.

9.5 MKDB.FILTER:

MKDB.FILTER: **[NO or ONLY_DET]**

  ONLY_DET -- only detected observations are put into the database,
                i.e. the observations with the SNR exceeding
                SNR_DETECTION_LIMIT. NB: if an observation has the SNR
                greater than the limit, this does not necessary means
                the observation is a detection.

  NO   -- no filter is applied. All observations regardless their
                SNR are put into the output database.

9.6 MKDB.FRINGE_ALGORITHM:

MKDB.FRINGE_ALGORITHM: [DRF or LSQ or MUL or ADD or NO]

   This keyword specifies what estimate of group delay,
phase delay rate, group delay rate and their formal
uncertainties to put in the output database. PIMA computes
four estimates of group delays then fine fringe search algorithm,
but only one estimate can be put in the output database. This
keyword determines which estimate is to he put in the output file.
If unsure, specify LSQ.

  DRF -- estimates of group delay, phase delay rate, fringe phase,
        and fringe amplitude from the coarse fringe fitting
        will be put into the output database. Group delay rate
        is zero.

  LSQ -- estimates of group delay, phase delay rate, group delay
        rate, fringe phase, and fringe amplitude from the LSQ fine
        fringe fitting algorithm without applying re-weighting
        will be put into the output database.

  MUL -- estimates of group delay, phase delay rate, group delay
        rate, fringe phase, and fringe amplitude from the LSQ fine
        fringe fitting algorithm with applying multiplicative
        re-weighting will be put into the output database. The
        formal uncertainties are scaled by a certain factor in order
        to make the ratio of the weighted sum of squares of
        residuals to their mathematical expectation close to unity.

```
ADD -- estimates of group delay, phase delay rate, group delay
       rate, fringe phase, and fringe amplitude from the LSQ fine
       fringe fitting algorithm without applying re-weighting
       will be put into the output database. An additive parameter
       is found in such a way that after being added in quadrature
       to the a priori weights the ratio of the weighted sum of
       squares of residuals to their mathematical expectation close
       to unity.

NO  -- estimates of group delay and phase delay rate from coarse
       fringe fitting will be put into the output database.
```

9.7 MKDB.2ND_BAND:

MKDB.2ND_BAND: **[NO or file]**

This keyword specifies whether to chain results from two bands
into one database.

```
NO   -- results from fringe fitting from one band specified in the
        current control file will be put in the output database.

file -- name of the second control file. Fringe fitting results
        from two bands, two control files, will be put in the output
        database. The first band is the band specified in the current
        database. The second band is the band specified in
        MKDB.2ND_BAND keyword.
```

NB: Although PIMA does not require bands be in a certain order
an order, VTD/Post-Solve does. VTD/Post-Solve requires the first
band to have a higher frequency and the second band to have a lower
frequency.

9.8 MKDB.VCAT_CONFIG:

MKDB.VCAT_CONFIG: **[NO or file]**

Name of the VCAT configuration file. Required if MKDB.OUTPUT_TYPE is
GVF and ignored otherwise. See VTD/Post-Solve documentation for format
of VCAT configuration file.

9.9 MKDB.OUTPUT_NAME:

MKDB.OUTPUT_NAME: **value**

   Meaning of this value depends on the value of MKDB.OUTPUT_TYPE keyword.
If MKDB.OUTPUT_TYPE is AMPL or TEXT, then MKDB.OUTPUT_NAME specifies
the name of the output file for task mkdb. If MKDB.OUTPUT_TYPE is GVF,
then MKDB.OUTPUT_TYPE specifies a suffix of the database name.
The GVF compliant database name is yyyymmddS_vNNN.env, where S is
a one-letter long lower case suffix and NNN is the version number.

9.10 MKDB.DESC_FILE:

MKDB.DESC_FILE: **[file or NO]**

  The name of the file with additional information about the experiment
that propagates to the GVF database. If no information is available,
NO can be specified. This parameter is used only by task mkdb when the
output data type is GVF.

Questions and comments about this guide should be directed to:

Leonid Petrov (http://astrogeo.org/petrov)

Last update: 2020.05.02