VEX File Definition/Example

Rev 1.5b1
30 January 2002

Introduction

This document describes and defines the 'VEX-file' format (VEX = 'VLBI Experiment'), which has been invented to prescribe a complete description of a VLBI experiment, including scheduling, data-taking and correlation.. This includes all setup and configuration information, as well as the schedule of observations. VEX is designed to be independent of any particular VLBI data-acquisition system or correlator, and is expandable to accommodate new equipment, recording and correlation modes. Every attempt has been made to consider the requirements and concerns of both the astronomy and geodetic VLBI communities in the construction of the VEX format.

Files in the VEX format are targeted at three particular types of files in the experiment process:
1. The schedule file (generated by your scheduling program of choice): This VEX-format file will be completely self-contained file which details the experiment setup and execution for all sites. (The example VEX file in this document is primarily of this type.)
2. The station experiment summary file, detailing the actual as-observed-experiment: Each participating station
will create a VEX-format file with 'as-observed' (i.e. log) information. As of this writing, much work needs to be done to specify the details of this file.
3. The Mark IV and EVN correlators (at least) are planning to use a VEX file format as the primary correlator-control file.

The first major section of this document is a combination VEX definition and example, and is intended to show the concept, style and capabilities of the VEX-format. It is not intended to illustrate a real experiment; as a result, some of the modes and configurations are somewhat contrived and fictitious in order to illustrate the capabilities of the VEX format. Furthermore, there are undoubtedly some errors which have escaped our attention.

This document should be studied in conjunction with the 'VEX Parameter Tables' document, which defines the VEX parameters in detail. Because VEX is intended to evolve as systems and capabilities evolve, these tables cannot be judged as static and will undoubtedly evolve in time.

The definition of the VEX file format has been primarily the work of Alan Whitney and Colin Lonsdale of Haystack Observatory and Ed Himwich and Nancy Vandenberg of NASA/GSFC/NVI, with important contributions from Huib van Langevelde, Ari Mujunen and Craig Walker.

VEX Definition/Example

The following is a combination VEX definition and example file. All lines or phrases beginning with an asterisk ('*') are comments. The companion document titled 'VEX Parameter Tables' should be consulted for details of actual parameters values and their meanings.

```
VEX_rev = 1.5;           *defines file format; must start at first character on first line of VEX file.
*
*VLBI experiment (VEX) definition and example file                                            ARW 31 Jan 98
*
*******************************************************************************************************************
*
*VEX Revision Log
*
*Rev   Date     Revisions
*1.0  24Nov95   Original Release
*1.1   5Dec95   Changes made in response to community feedback to release 1.0:
*               1. Change $RX blockname to $IF
*               2. Change epoch specification to '1995y306d12h23m06.2s' style to remove colons.
*               3. Add units to specification of ra/dec source positions (e.g. ra=05h23m2.56s; dec=-20d45'12.2";)
*               4. 'Indirect references' renamed to 'links' to avoid confusion with references to keywords.
*                  Mandate that all linkwords have '&' as first character.
*               5. Referencing to $PHASE_CAL block modified to be consistent with other referencing.
*               6. Clarify usage with antenna clusters (multiple antennas utilizing common DAS)
*               7. Add velocity wrt LSR parameter to source definition for post-correlation processing purposes.
*               8. Add example of orbiting site to $SITE block.
*1.2  28Dec 95  1. Slightly modify 'roll=' parameter in $ROLL section to accomodate more general barrel-roll sequence.
*               2. Update 16-track VLBA barrel-roll sequence (def 'VLBA/16TKS' in $ROLL) to match latest information
*                  from VLBA.
*               3. Fix a number of minor typos and clarify some explanations.
*1.3   2Feb 96  1. Slight modify 'pointing_sector' parameter in $ANTENNA section.
*               2. Add 'axis_orientation' parameter in $ANTENNA section.
*               3. Slight modify 'headstack' parameter in $DAS section.
*               4. Change name of parameter 'ut1' to 'ut1-utc' in $EOP section.
*1.4   8May 96  1. Explicity list prohibited characters in keywords and linkwords.
```

```
*                2. Allow quoted (“...”) character strings with embedded white space.
*                3. Specify legal character set more tightly.
*                4. Prohibit any null bytes in files.
*                5. Tighten up literal-block specifications.
*                6. Move primary phase-cal specification to $IF block for more uniformity.
*                7. Rename $PHASE_CAL block TO $PHASE_CAL_DETECT and restructure. Add link to $FREQ section.
*                8. In $SCHED section, change 'def...enddef;' construction to 'scan...endscan;'
*                9. Restructure clock specification for more syntactic and structural uniformity.
*               10. Add 'exper_desc' (experiment description) parameter to $EXPER block.
*               11. Delete 'bits_per_sample' parameter since it is redundant with $TRACKS specification.
*               12. Rename $SCHEDULING_PARMS block to $SCHEDULING_PARAMS and specify that it contain only a
*                   literal-block def at current time.  Scheduling program must parse.
*               13. In $SITE block, use separate parameters for site position, epoch and referency (for more uniformity
*                   with source position.
*               14. Add 'roll=on|off' parameter.
*               15. Simplify drive# specification in $DAS and $SCHED blocks.
*               16. In $SOURCE section, change 'epoch' of source position to 'ref_coord_frame' and
*                   'source_pos_epoch'.  For satellite, change 'epoch' to 'orbit_epoch'.
*1.5   5Nov 96   1. White-space rules clarified.
*                2. 'phase-cal' parameter deleted. 'phase-cal_detect' parameter updated.
*                3. 'antenna' spelled out fully in all relevant antenna-related parameters in $ANTENNA.
*                4. 'antenna_name' paramter added in $ANTENNA.
*                5. 'electronics_rack_name' and 'record_transport_name' added in $DAS
*                6. Add ‘physical IF’ parameter to $IF section.
*                7. 'tape_length' and 'tape_motion' parameters modified for S2 in $DAS.
*                8. 'S2_record_mode' and 'S2_data_source' parameters added in $TRACKS. 'S2_data_def' deleted.
*                9. 'S2_group_order' added to $PASS_ORDER.
*               10. Missing 'SCHEDULING_PARAMS' section added.
*               11. Change field limits to 128 characters.
*               12. Prohibit quoted-character-strings in everything except parameter values.
*               13. Allow 16-char source names in $SOURCE.
*               14. Change $PROC_TIMING to $PROCEDURES
*               15. Add ‘procedure_name_prefix’ parameter in $PROCEDURES section.
*               16. In $DAS, change 'record_transport' to 'record_transport_type' and 'electronics_rack' to
*                   'electronics_rack_type'.
*               17, In $EXPER, change 'exper_desc' to 'exper_description', 'pi_name' to 'PI_name' and
*                   'pi-email' to 'PI_email'.
*               18. In $PROCEDURES, change 'headstk_motion' to 'headstack_motion'.
*               19. In $ROLL, change 'reinit_period' to 'roll_reinit_period' and 'inc_period' to 'roll_inc_period'.
*               20. In $SITE, change 'site_pos_ref' to 'site_position_ref' and add 'site_velocity'.
*                   Change 'site_name' from max of 8 to 16 characters.
*               21. In $SOURCE, change 'source_pos_ref' to 'source_position_ref' and 'source_pos_epoch' to
*                   'source_position_epoch'.
*1.5a  31Jan98   1. Update and clarified much explanatory text following 9 Sep 97 e-mail from WEH.
*                2. Adopted 128-characters as standard length limit.
*                3. Slightly revise excluded characters for various items.
*                4. Fix several typographical errors.
*                5. Remove polarization parameter from $FREQ ‘chan_def=’ statement.
*                6. Add polarization parameter to $IF ‘if_def=’ statement.
*                7. Change $PROCEDURES ‘standard procedures=’ to ‘procedure_name_prefix’.
*                8. Remove $ANTENNA ‘axis_orientation’ parameter and augment ‘axis_type’ parameter for x/y type mounts.
*                9. Update ‘VEX Parameter Tables’ document.
*1.5b  17Dec01  10. Draft updates to include Mark 5A.
*1.5b1 29Jan02  11. Further draft updates to include Mark 5A.
```

***$Blocks**

```
*
*The VEX file is divided into a number of separate '$blocks', each beginning with '$blockname' statement.
*There are several types of $blocks:
* 1.'Primitive' $blocks: Consist entirely of keyword 'def blocks' which define low-level station, source, and
*   recording parameters. The currently defined primitive $blocks and their use in various stages of the experiment
*   are summarized below:
*                                                                    Req'd in   SCHEDULING  DATA-TAKING  CORRELATION
*         $ANTENNA           antenna parameters                                       X                        X
*         $BBC               BBC/IF assignments                                       X           X
*         $CLOCK             clock-synchronization model for correlation                                       X
*         $CORR              correlation parameters (NYI)                                                      X
*         $DAS               data-acquisition system information                      X           X            X
*         $EOP               earth-orientation information                                                     X
*         $EXPER             general experiment information                           X           X            X
*         $FREQ              channel frequencies, bandwidths, sample rates, etc.      X           X            X
*         $HEAD_POS          headstack-positioning information                        X           X            X
*         $PASS_ORDER        tape pass order                                          X           X            X
*         $PHASE_CAL_DETECT  phase-cal frequencies to be detected                     X           X            X
*         $ROLL              recording barrel-roll details                            X           X            X
*         $IF                IF bands/sidebands, 1st LO freq, phase-cal freqs         X           X            X
*         $SEFD              SEFD info                                                X
*         $SITE              antenna site details                                     X                        X
*         $SCHEDULING_PARAMS parameters for scheduling program                        X
*         $SOURCE            source names, positions, etc.                            X           X            X
*         $PROCEDURES        general-procedure timing parameters                      X           X
*         $TRACKS            recording multiplex details                             X           X            X
*   The '.obs' file, currently not fully defined, will contain additional $blocks.  Some of the $blocks are:
*         $ANTENNA_CAL_OBS ' As-observed' Antenna calibration measurements
*         $CABLE_CAL_OBS   ' As-observed' cable-calibration measurements
*         $CLOCK_OBS       ' As-observed' clock-sync measurements
*         $SCHED_OBS       ' As-observed' schedule
*         $PHASE_CAL_OBS   ' As-observed' phase calibration measurements
*         $TAPELOG_OBS     ' As-observed' tape-usage log
*         $TSYS_OBS        ' As-observed' system-temperature measurements
*         $WX_OBS          ' As-observed' weather measurements
* 2. $GLOBAL block: Specifies global/general experiment parameters as 'refs' to primitive-block keywords.
* 3. $STATION/$MODE blocks: Define station and mode keywords in 'def' blocks which contains 'refs' to primitive-block
*    keywords.  The combination of a $STATION key, $MODE key, plus the $GLOBAL parameters specify the detailed
*    configuration for an observation at a particular station.
* 4. $SCHED block: specifies a detailed time-ordered list of observations, using keyword references to $STATION, $MODE
*    and $SOURCE 'defs'. (Time-ordering requirement may be removed in future.)
*
* Additional $block types may be added for special purposes.  The order of the $blocks is irrelevant except
* for convenience and readability.  In this example VEX file, the numerous primitive $blocks are simply ordered
* alphabetically.
*
*********************************************************************************************************************
*
*High-level Organization Rules:
*
* 1. The first line in the VEX file identifies the file type and must be of the form 'VEX_rev = <VEX rev level>;'
*    followed by an optional comment (prefaced with a '*').
* 2. Except for the first line, the active (i.e. non-comment) statements in a VEX file consist entirely of blocks
*    separated (and named) by '$BLOCKNAME;' statements.  For convenience, we will refer to those blocks as '$blocks'
*    or 'sections'.
* 3. The '$BLOCKNAME;' statement is the first statement in a $block and names the $block.
* 4. A $block is terminated by the specification of another '$BLOCKNAME;' statement.
* 5. '$BLOCKNAME' just be comprised of characters from the ‘Legal Character Set’
*    (see ‘Syntax Rules’ below).
* 6. $blocks may be in any order, though they are normally organized for convenience and readability.
* 7. Additional $blocks may be added as necessary.
* 8. The maximum length of $blockname is specified under ‘Syntax Rules’ below.
*
*********************************************************************************************************************
*
*$block Types
*
* $blocks are divided into three types:
*   1. 'Primitive' $blocks  - each primitive $block serves to define a set or sets of parameters (via keyword 'defs')
*                             with no direct reference to any other $blocks (except perhaps to an external database
*                             file).
*   2. 'High-level' $blocks - made up entirely of references to primitive-block keywords to construct the full
*                             suite of parameters necessary to specify an experiment.  The $GLOBAL, $STATION,
*                             and $MODE blocks are the currently-defined 'high-level' blocks.
*   3. $SCHED block         - entries necessary to drive the observation schedule.
*                             Has a special format and includes references to $STATION, $MODE and $SOURCE blocks.
*
*********************************************************************************************************************
*
*Syntax Rules:
*
* General
*    1. Upper/lower case is relevant in VEX statements.
*    2. All VEX statements start with $blockname, 'def', 'enddef', 'ref', 'scan', 'endscan', 'start_literal()',
*       'end_literal()' or '<parameter>='.
*    3. VEX puncuation characters are:
*          '=' - assignment of a keyword or parameter value
*          ':' - delineator between subfields in the assignment of parameter value(s).
*          ';' - terminates a VEX statement
*          ' ' - (white space) in special cases.  See White Space Rules below.
*    4. Multiple statements may occur on a single line.
*    5. A single statement may span multiple lines.
*    6. The maximum length of block names (after the leading ‘$’), link names (after the leading ‘&’), parameter
*       names, parameters values, keywords, and file names is 128 characters.
*    7. All blocknames (and only blocknames) begin with the ‘$’ character.
*    8. All link words  (and only link words) begin with the '&' character.
*    9. White Space Rules -
*       a. No white space is permitted within block names, keywords, file names, parameter names or values *except*:
*          at least one white space must separate a numerical parameter value from its units (e.g. '10 sec').
*       b. At least one white space must follow 'def', 'ref' or 'scan' (e.g. 'ref $FREQ').
*       c. No white space (or anything else) may occur before the 'VEX_rev=' statement at the beginning of the VEX
*          file.
*       d. With the above exceptions, an arbitrary amount (including zero) of white space is allowed anywhere within
*          a VEX file.
*       e. For purposes of parsing, an <end-of-line> character is equivalent to one white-space character, except that
*          an <end-of-line> unconditionally terminates a comment.
*   10. There are no column specifications in the VEX format, other than those imposed by the discipline of the
*       user for easy readability.  A 'literal block', however, may have column restrictions/specifications which
*       fall outside of the VEX format specification.
*   11. Keywords, link names and scan_ID’s are arbitrary and do not contain information.  If is helpful if they are
*       descriptive, but they are there just for bookkeeping.  All actual information is in VEX parameter statements
*       and the use of ‘refs’.
*
* Legal Character Set
*    1. The character set is confined to ASCII, except that the null character is prohibited anywhere,
*       including any literal blocks, comments and quoted text.  7-bit ASCII is preferred, though not required, in
*       VEX statements; comments may freely use 8-bit ASCII.
*    2. Block names, following the initial ‘$’, can include any characters except “ \t\n;=” (5 characters excluded).
*    3. Parameter names may contain any characters except “ \t\n;:=” (5 characters excluded), and in addition may not
*       start with any of ‘*$&”’. [“*” starts a comment; “$” starts a block name; “&” starts a link name;
*       ‘”’ surrounds a quoted string.]  There are some excluded string as well: “ref”, “def”, “enddef”,
*       “scan” and “endscan”.
*    4. Parameter values and keywords may contain any characters except ‘\t\n;:=&*$”’ and space (9 characters).
*    5. Link names, following the intial ‘&’, may contain any characters except ‘\t\n;:=&*$’ and space (9 characters).
*    6. Quoted character strings are allowed as parameter values in parameter-assignment statements.
*       Quoted character strings are allowed, but not encouraged, as block names, link names (following the initial
*       ‘$’ or ‘&’), parameter names and keywords.
*    7. Quoted character strings must start and end with a double-quote (“).  Quoted strings follow the usual C
*       quoting conventions for character strings, except that a null character may not be specified.  The length
*       of a quoted string is the length of the resulting characters, not the quoted representation.
*    8. Literal blocks fall outside of the normal VEX construction rules and may contain any characters except null.
*
* Comments
*    1. Any non-quoted asterisk ('*') begins a comment.  A comment is terminated at the first trailing new-line.
*       A comment may be preceded by optional whitespace.
*    2. A comment may not be embedded within a VEX statement.
*    3. A comments may contain any ASCII character except null.
*    4. For purposes of parsing, a comment is considered to be equivalent to a single space character.
*    5. A point of philosophy: Software should strive to retain comments whenever a VEX file is processed.
*       [If optional whitespace before a comment contains any new-lines, software should retain exactly one new-line.]
*
* 'def' Construct:
*    1. The 'def' block is used to assign a keyword name to a set of specified parameters values.
*    2. The 'def' block is the ONLY way to specify parameter values.
*    3 'def' blocks are allowed in most $block sections, but specifically prohibited in the $GLOBAL and $SCHED
*       blocks.
*    4. The structure of each def block is:
*         def <keyword>;                                    *define keyword
*           The following are possible statements within a 'def':
*           <parameter>=<list_of_values>;                   *direct setting of parameter values;
*                                                           *permitted only in 'primitive' $blocks
*           ref <external filename>:<$blockname>=<keyword>; *set parameter values by reference to external source;
*                                                           *permitted only in 'primitive' $blocks.
*                                                           *Not permitted in '.skd' (which must be self-contained)
*           ref <primitive $blockname>=<keyword>;           *set parameters according to referenced 'def' block;
*                                                           *permitted only in $GLOBAL, $STATION, $MODE $blocks
*           ref <primitive $blockname>=<keyword>:<qualifier>; *qualify the referenced 'def'-block parameters according
*                                                           *to station, time, etc (see 'ref' Construct rules).
*             .
*         enddef;
*    5. At least one space (or <end-of-line> character) must follow the word 'def'.
*    6. <keyword> must be unique within the $block.
*    7. Any number of '<parameter>=' or 'ref' statements may occur within the 'def'.
*    8. The order of statements within a 'def' is irrelevant.
*    9. A 'def' without a corresponding 'ref' may as well not exist.
*   10. The order of 'def' blocks within a $block is irrelevant.
*
* <parameter>=<list_of_values> Construct:
*    1. Permitted only within ‘defs’ defined within primitive $blocks, except for special ‘VEX_rev=’ statement
*       required as the first VEX statement in a VEX observation file.
*    2. The specification of any particular parameter is permitted only within the primitive $block to which
*       it is assigned (see VEX Parameter Tables).
*    3. The number and type of <list_of_values> must be consistent with the definition specified in the
*       VEX Parameter Tables.
*    4. Entries within the <list_of_values> are separated by colon (':')characters.  Each such colon may be
*       surrounded by zero or more white spaces.
*    5 .Each value in the <list_of_values> must be of one of the following types:
*         a. Real/Integer
*              Examples of acceptable values: -1.234, +5.67e-12, -.987E+04, 0.267e2, -871
*              Followed by units specification if required.
*              Unit specification, if present, must be separated by one or more spaces.
*         b. Character
*              A character string may be specified either with or without surrounding quotes:
*                With quotes - All characters between the quotes, including white space, will be parsed as
*                              part of the character string (example: “Character string”).
*                              Legal characters are “A-Za-z0-9_/-+”.  Normal C-language quoting conventions are
*                               observed, except that no null bytes may be present.
*                 Without quotes - No embedded spaces are allowed.  Character string is assumed to start with
*                               first non-space character and end with last non-space character before terminating
*                               colon or semi-colon.
*         c. Epoch
*              An epoch is specification of a particular instant of time is and is always expressed as
*              UTC '##y###d##h##m##.###s' where '#' is a numeral, '###d' is day-of-year (1-366),
*              '##h' is hour (0-23), '##m' is minute (0-59) and '##.###s' are seconds (0-59.999..).
*              Only the seconds sub-field may contain a non-integer (i.e. real) value.
*              The year may be either 4-characters (e.g.'1997y') or 2-characters (e.g.'97y').  If appropriate,
*              epoch may be truncated from the right (e.g. '97y263d12h') with implied 0's for unspecified fields.
*         d. RA/Dec
*              Source position specified in ra/dec use an abbreviated form of units, of the form
*              'ra=##h##m##.###s; dec=##d##'##.###";' (e.g. ra=05h23m2.56s; dec=-20d45'12.2";)
*         e. &link
*              See 'Links' section below.
*    6. Unit Specifications
*       a. Integer or real values with units of time, freq, sample rate, length, angle, angular-rate and flux
*          may require the explicit specification of units, as specified in the VEX Parameter Tables:
*               time      -   'psec', 'nsec', 'usec', 'msec', 'sec', 'min', 'hr', 'yr'
*               frequency -   'mHz', 'Hz', 'kHz', 'MHz', 'GHz'
*               sample rate-  'ks/sec', 'Ms/sec'
*               length    -   'um', 'mm', 'cm', 'm', 'km', 'in', 'ft'
*               angle     -   'mdeg', 'deg', 'amin', 'asec', 'rad'  (exception - source position: see below)
*               ang rate  -   '<angle>/<time>', where <angle> and <time> are any valid units
*               velocity  -   '<length>/<time>', where <length> and <time> are any valid units
*               flux      -   'mJy' 'Jy'
*               bit density - 'bpi', 'kbpi'
*          Unit specifications follow, separated by one or more spaces, the numerical value to which they refer.
*          In cases where a parameter statement ends with a variable-length list of numerical values which take the
*          same units, the first field of the list must have specified units, but the following values may omit them,
*          causing them to default to the same units.  [Note, however, for parameter statements with a fixed number
*          of fields, units must always be specified.  Thus, for example, 'source_model' and 'pointing_sector'
*          require units for all relevant fields, but 'horizon_map_az' 'ut1-utc' and 'switching_cycle' do not.]
*
* ref <external filename>:<$blockname>=<keyword> Construct:
*    1. Permitted only within primitive 'defs' to refer to lists of <parameter>=<list_of_values> statements from
*       an external source.
*    2. <$blockname> is the $block name to be searched in the external file (Normally, this would be the same
*       as the VEX file primitive $blockname in which this statement occurs).
*    3. <keyword> is a defined keyword in the <$blockname> block of the external file.
*    4. The syntax rules and construct rules for the external file are identical to the rules for primitive $blocks.
*    5. The order of 'refs' within a 'def' is irrelevant.
*    6. At least one space (or <end-of-line> character) must follow the word 'ref'.
*
* ref <primitive $blockname>=<keyword>[:<qualifier(s)>] Construct:
*    1. Permitted only within $GLOBAL, $STATION, $MODE blocks.
*    2. <keyword> is a defined 'def' keyword within the specified <primitive $blockname>.
*    3. Within a $STATION or $MODE 'def' block, there may be multiple 'refs' to the same primitive $block
*       (with distinct keywords).  In this case, the parameter values associated with the multiple 'refs'
*       are simply concatenated.  No conflicting or duplicate parameters are permitted.
*    4. <qualifier(s)> is an optional parameter that may be used only in the $MODE block to qualify
*       a 'ref' by station(s).  <qualifier(s) may be a single $STATION 'def' keyword, or may be muliple $STATION
*       keywords separated by colons
*       Note: Current operational considerations require that *all* $MODE 'refs' be qualified by the complete list
*             of stations to which they apply.  This is necessary so that useful consistency checking can be applied.
*             In the future, checking may be re-written to relax this constraint.
*    5. The order of 'refs' within a 'def' or within the $GLOBAL block is irrelevant.
*    6. At least one space (or <end-of-line> character) must follow the word 'ref'.
*
* Links
*    'Links' are cross-references made through the use of 'linkwords' in the <list_of_values> part of various
*     <parameter>=<list_of_values> statements.  They serve to link together the myriad of details
*     regarding station and recording configurations in the selected 'defs'.  There is, for example,
*     heavy reliance on 'links' between the $FREQ, $BBC, $IF and $TRACKS blocks in order to specify
*     the detailed RF, IF, BBC and configurations.  For clarity, linkwords require a '&' as the first character
*     of their name, but are otherwise arbitrary using the normally-valid character set. Embedded spaces are
*     not permitted.  To understand the concept of 'links' it may useful to examine the following $blocks in order:
*     $FREQ >> $BBC >> $IF, $FREQ >> $TRACKS.
*
* Literal Block Construct
*    1. A block of text may be declared as 'literal' if the block is preceded by a 'start_literal();' statement
*       and terminated by an 'end_literal();' statement, as follows:
*                start_literal(<string>);          *'start_literal' statement must be last VEX statement on line
*                   <first line of literal text>
*                     .
*                     .
*                   <last line of literal text>
*                 end_literal(<string>);           *'end_literal' statement must be first VEX statement on line
*       where <string> is optional (i.e. may be empty) and may be arbitrarily chosen to avoid any spurious
*       interpretation of an embedded 'end_literal()' string.  <string> may not contain ')', null or new-line
*       characters, but all other characters are valid including white space.
*    2. The literal block starts on the first line following the 'start_literal' statement and ends on the last
*       line before the 'end_literal' statement.
*    3. Any ASCII characters except null are allowed within a literal block.
*    4. A literal block may only occur within a 'def' block.
*    5. Parsing of a literal block is outside the VEX-format specification.
*       Note: At this time, the literal block is used only in the $SCHEDULING_PARAMS block.
*
*----------------------------------------------------------------------------------------------------------------------
$GLOBAL;                            *global default parameters
*----------------------------------------------------------------------------------------------------------------------
*
*$GLOBAL block rules:
*    1. Purpose is to specify global/general information as appropriate.
*    2. Only 'refs' to primitive-block keywords are permitted.
*    3. 'refs' to any primitive $blocks are permitted and are concatenated with 'refs' in the $STATION and $MODE
*        blocks.  No conflicting or duplicated parameters are permitted in the concatenation.
*
    ref $EXPER = EXP1387;             *general experiment information
    ref $SCHEDULING_PARAMS = SKED1;   *choose scheduling program/parameters
    ref $PROCEDURES = STANDARD1       *general-procedure timing parameters
    ref $EOP = EOP129;                *earth-orientation parameters
*
*----------------------------------------------------------------------------------------------------------------------
$STATION;                           *station parameters
*----------------------------------------------------------------------------------------------------------------------
*
*$STATION $block rules:
*    1. Consists exclusively of 'def' blocks.
*    2. Within 'def' blocks, only 'refs' to primitive-block keywords are permitted.
*    3. For each observation in the $SCHED block, the set of parameters specified by the selected $STATION
*       and $MODE keywords are concatenated with the parameters specified in the $GLOBAL block to form
*       the full set of observing parameters for each station in that observation.
*       Conflicting or duplicate parameters within the concatenated parameter set are not permitted.
*    4. Multiple 'refs' to ‘defs’ within the same primitive $block are simply concatenated.
*    5. Parameter values implied by the 'refs' in the $GLOBAL, $STATION or $MODE blocks must not conflict
*       (i.e. there must be no more than one value specified for a single parameter).
*
*Notes:
*    1. The requirement that only 'refs' are allowed in the $STATION/$MODE blocks may seem somewhat awkward since
*       some of the 'refs' refer to single-parameters primitive-'defs'.  The advantage, however, is that this
*       rule forces all parameter values to be set in the primitive $block to which they are uniquely assigned,
*       eliminating any possible confusion about their meaning.
*    2. A 'station' consists of an ANTENNA on a SITE with a data-acquisition system (DAS).
*       The $STATION block is intended for the specification of parameters which truly are associated with each
*       station and will remain fixed for the duration of the experiment at each station.
*
*Special cases:
*    1. Multiple antennas at one location:
*       Each antenna is defined separately in the $STATION block and must have its own $SITE 'def'.
*       If the antennas are identical, they may reference the same $ANTENNA 'def'.
*    2. Multiple antennas sharing a single data-acquisition system (antenna cluster):
*       If two or more antennas share the same DAS, the $STATION 'defs' for the corresponding antennas must have
*       'refs' to exactly the same set of $DAS keywords, including particularly the 'recording_system_ID' parameter
*       (which identifies the particlar DAS), except that one (and only one) of the stations must declare itself
*       as the tape-control master with the inclusion of a 'tape_control=master;' statement within the referenced
*       $DAS 'defs' for that station (this will cause the tape to be controlled according to the schedule of that
*       station; the others will simply point).  The $FREQ 'def' selected for each station, along with the
*       corresponding selected $BBC and $IF 'defs', define the channels recorded from each antenna.  Normally,
*       all stations will reference the same 'defs' in the $TRACK, $ROLL, $HEAD_POS, $PASS_ORDER blocks (depending
*       on type of DAS), but care must be taken to ascertain that the combination of specified recordings is
*       compatible with the DAS.
*
  def EF;
    ref $SITE = EFLSBERG;
    ref $ANTENNA = EFLSBERG;
    ref $DAS = VLBA/1_DRIVE;
    ref $DAS = 33KBPI;
    ref $DAS = 8820FT;
    ref $DAS = EF_VLBA_ID;
    ref $PHASE_CAL_DETECT = STANDARD;
    ref $CLOCK=EF;
  enddef;
  *
  def FD;
    ref $SITE = VLBA-FD;
    ref $ANTENNA = VLBA;
    ref $CLOCK=FF;
    ref $DAS = VLBA/2_DRIVES;
    ref $DAS = 33KBPI;
    ref $DAS = 17640FT;
    ref $DAS = FD_VLBA_ID;
    ref $PHASE_CAL_DETECT = STANDARD;
  enddef;
  *
  def HS;
    ref $SITE = HAYSTACK;
    ref $ANTENNA = HAYSTACK;
    ref $CLOCK=HS;
    ref $DAS = MARK5A;
    ref $DAS = HS_MARK5A_ID;
    ref $PHASE_CAL_DETECT = STANDARD;
  enddef;
  *
  def JB;
    ref $SITE = JODRELL_MK2;
    ref $ANTENNA = JODRELL_MK2;
    ref $CLOCK=JB;
    ref $DAS = MARK4;
    ref $DAS = 56KBPI;
    ref $DAS = 17640FT;
    ref $DAS = JB_MARK4_ID;
  enddef;
  *
*------------------------------------------------------------------------------------------------------------------------
$MODE;                              *data and recording parameters
*------------------------------------------------------------------------------------------------------------------------
*
*$MODE block rules: See $STATION block rules.
*
*Notes:
*    1. The $MODE block is intended to specify the detailed station setup parameters that may change from
*       observation to observation.
*    2. A powerful capability of the 'ref's in the $MODE block is the ability to qualify them by station, so that the
*       specification of a single $MODE key automatically configures each station according to the details of its
*       particular observing equipment and capabilities.
*    3. A 'ref <$blockname> = <keyword>:<$stationkey(s)>;' statement is applied only to the station(s) specified.
*       Note: Current operational consideration require that *all* $MODE'refs' be qualified by the complete list
*             of stations to which they apply.  This is necessary so that useful consistency checking can be applied.
*             In the future, checking may be re-written to relax this constraint.
*
  def SX;
  *As an illustration, this 'def' sets up data and recording parameters for four disparate stations
  *with various types of equipment and different (but compatible) recording modes.
    ref $FREQ = S4:EF;                    *station EF
    ref $FREQ = X4:EF;
    ref $BBC = EF/S4X4:EF;
    ref $IF = EF/X:EF;
    ref $IF = EF/S:EF;
    ref $SEFD = EF:EF;
    ref $TRACKS = VLBA/XX-8-2/8:EF;
    ref $TRACKS = VLBA_TRK_FORMAT:EF;
    ref $TRACKS = DATA_MODULATION_ON:EF;          *turn-on data-modulation
    ref $TRACKS = FRMTR_TK7_TO_SYSTRK0:EF;      *compensate for known bad head
    ref $HEAD_POS = VLBA/1_HDSTK:EF;
    ref $PASS_ORDER = VLBA/8:EF;
    ref $ROLL = VLBA/8:EF;
    ref $PROCEDURES = STANDARD_CAL:EF;
    ref $DAS = START/STOP:EF;
*
    ref $FREQ = S4:FD;                    *station FD
    ref $FREQ = X4:FD;
    ref $BBC = VLBA/S4X4:FD;
    ref $IF = VLBA/X:FD;
    ref $IF = VLBA/S:FD;
    ref $SEFD = VLBA-FD:FD;
    ref $TRACKS = VLBA/XX-8-2/16:FD;
    ref $TRACKS = VLBA_TRK_FORMAT:FD;
    ref $TRACKS = TRNSPRT_TK23_TO_SYSTRK33:FD;       *compensate for known bad head
    ref $HEAD_POS = VLBA/2_DRIVES:FD;
    ref $PASS_ORDER = VLBA/16:FD;
    ref $ROLL = VLBA/16:FD;
    ref $PROCEDURES = VLBA_CAL:FD;
    ref $DAS = START/STOP:FD;
*
    ref $FREQ = S4:HS;                      *station HS (Mark 5A)
    ref $BBC = HS/S4:HS;
    ref $IF = CDP/SX_WIDE:HS;
    ref $SEFD = HS:HS;
    ref $TRACKS = MARK5A/XX-4-2/8:HS;
    *Note: For Mark 5A, no ref's to $HEAD_POS or $PASS_ORDER
    ref $TRACKS = MARK4_TRK_FORMAT:HS;
    ref $PROCEDURES = STANDARD_CAL:HS;
    ref $DAS = EARLY_START_20:HS;
*
    ref $FREQ = X4:JB;                      *station JB
    ref $FREQ = S4:JB;
    ref $BBC = JB/S4X4:JB;
    ref $IF = JB/X:JB;
    ref $IF = JB/S:JB;
    ref $SEFD = JB:JB;
    ref $TRACKS = MARK4/XX-8-2/64:JB;
    ref $TRACKS = MARK4_TRK_FORMAT:JB;
    ref $HEAD_POS = MARK4/2_HDSTKS:JB;
    ref $PASS_ORDER = MARK4/64:JB;
    ref $PROCEDURES = STANDARD_CAL:JB;
    ref $DAS = START/STOP:JB;
  enddef;
*
  def SX_VLBA;
* This illustration shows an example of a 'def' for a case of all stations operating with the same equipment
* and recording modes.  In this case, each 'ref' applies to all stations.
    ref $FREQ = SX14;
    ref $BBC = SX14;
    ref $IF = CDP/SX_WIDE;
    ref $SEFD = VLBA-FD;
    ref $TRACKS = VLBA/XX-8-2/16;
    ref $TRACKS = VLBA_TRK_FORMAT;
    ref $HEAD_POS = VLBA/2_DRIVES;
    ref $PASS_ORDER = VLBA/16;
    ref $ROLL = VLBA/16;
    ref $PROCEDURES = VLBA_CAL;
    ref $DAS = CONTINUOUS;
  enddef;
*
*-----------------------------------------------------------------------------------------------------------
$SCHED;                             *schedule block
*-----------------------------------------------------------------------------------------------------------
*
*The $SCHED block specifies the actual scans to be taken.
*
*$SCHED block rules:
*  1. Each scan is specified by 'scan <scan_ID>'. This (arbitrary) 'scan ID' will carry through
*     into the log files and log-summary files to serve as a convenient reference to identify particular scans.
*     All <scan_ID>'s within the $SCHED block should be unique.  [Note: The <scan_ID> is primarily for the
*     convenience of the correlator and not of direct relevance to scheduling.]
*  2. The 'scan' must include the appropriate 'start=', 'source=', 'mode=' and 'station=' statements.
*  3. The nominal start time of a scan is specified by a 'start=' statement, as follows:
*           start=<epoch>
*     where <epoch> is in the VEX <epoch> format [typically '##y###d##h##m##.##s'].
*     The 'start=' time must be chosen to be the expected time of first good data at any station in the 'scan'.
*  4. The 'scan' blocks must be in time order by 'start=' times, although adjacent scans may have the
*     same 'start=' time (for example, if each of two antennas is to be pointed at different sources starting at
*     the same time). [Note: The time ordering constraint may be removed in the future.
*  5. Source references are of the form
*           source=<$SOURCEkey>
*     where <$SOURCEkey> is the desired keyword defined in the $SOURCE block
*  6. Mode references are of the form
*           mode=<$MODEkey>
*     where <$MODEkey> is the desired keyword defined in the $MODE block.
*  7. Each station participating in the scan is specified in a separate 'station=' statement of the form:
*       station=$STATIONkey:datastart:datastop:startpos:pass:pointsectr:tapedrive(s)
*         where
*           $STATIONkey  station keyword from $STATION block
*           datastart    Nominal start time of good data (i.e. antenna on-source) wrt 'start=' time.
*                        Must be specified and must be >=0.
*                        Note that the tape may have started earlier if an early start has been specified
*                        by a 'tape_motion=start&stop:<earlystart>' parameter in the $DAS block.
*           datastop     Nominal stop time of good data (i.e. antenna off-source) wrt 'start=' time.
*                        Must be specified and must be greater than 'datastart'.
*           startpos     In start/stop mode: nominal tape-start position (feet or meters).  Null if dynamic tape
*                        allocation.  Required for continuous motion (nominal footage at start of scan).
*                        For S2, startpos may be specified in terms of running time from the beginning of tape
*                        (specifiable in minutes or seconds only, not in hour-minute-second format).
*                        For Mark 5A: If =0, will request a new set of discs to be mounted.  If <>0, will be
*                        ignored since a new recording can only be appended to existing recording; may be used
*                        for informational purposes (a priori byte offset position) or may be filled in
*                        a posteriori as desired.
*           pass         specifies tape pass as a numeric headstack-position reference plus an alphabetic subpass
*                        identifier (examples: 1A, 12D) as defined in the referenced $TRACKS and $HEAD_POS 'def'
*                        blocks pertaining to that station.  For S2, specifies group number.  Null if pass if not
*                        specified (i.e. dynamic tape allocation) or if irrelevant to the recording system.
*                        Not relevant for Mark 5A.
*           pointsectr   specifies, if necessary, the antenna-pointing sector (as defined by a 'linkword' in the
*                        $ANTENNA block).  If null, cable-wrap considerations are considered not relevant.
*           drive        physical drive# (>=1).  If null, dynamic tape allocation is assumed.
*                        If =0, signifies 'point antenna, do not record' (EVN usage).
*                        [Technically, the drive# specified here is added to the 'drive-number offset' specified
*                         in the $DAS section, which is normally specified =0.  In the case of multiple simultaneous
*                         drives, two (or more) different 'drive-number offsets' in the $DAS section
*                        (normally 0 and 1) automatically specify the proper physical drives to be used.]
*  8.  Multiple 'station=XY...' statements (where XY is some station) may be written within a single scan block so
*      long as they specify non-overlapping recording periods.  This might be the case, for example, if the tape must
*      reverse direction one or more times at a station during a long scan.
*
*Notes:
*   1. The $SCHED block needs to contain only the information needed to acquire the data.
*      Depending on the scheduling procedure and/or particular antenna, some information such as passkey, startfoot
*      and/or pointing-sector may not be need to be specified.
*   2. The set of stations specified for a single scan must include all stations needed to form the desired
*      set of baselines to be correlated (though not all possible baselines will necessarily be correlated).
*      In cases of complicated clustering or subnetting, the correlator may, in some cases, be required to create
*      composite scans spanning more than one 'scan' block.
*   3. For purposes of processing multiple sources within a beam, multiple 'source=' statements are permitted for
*      each scan.  The antenna will be directed to point to the first source listed; if pointing to the
*      center of a group of multiple sources within a beam, the first 'source=' statement may be a reference to a
*      source of 'source_type=dummy' (in the $SOURCE def) that is used for pointing purposes only and will be ignored
*      by the correlator.
*   4. Each scan specified in the $SCHED block stands completely on its own as far as all details of station
*      and recording configurations are concerned.  This allows complicated switching between frequencies and/or
*      recording modes to be executed with relative ease, and also allows a rearrangement of the schedule without
*      the potential of lost configuration information.
*   5. For the special case of multiple antennas sharing the same DAS (antenna cluster):
*       a. One of the stations in the cluster must be declared as the master tape controller with the inclusion of
*          a 'tape_control=master;' statement included within referenced $DAS 'defs' for that station.
*          This will cause the tape to be controlled according to the schedule of that station; (i.e. the other
*          stations in the cluster will only point).  In this case, the 'strt pos', 'pass' and 'drive' sub-fields
*          in the 'station=' statement will be null.
*       b. Apart from the tape-control aspect mentioned above, each of the antennas in a cluster act as independent
*          stations and scheduled as such within the $SCHED block.
*
*
*                          stn   data      data      strt    pass  point  drv#
*                          key   strt      stop       pos          sectr
  scan 263-061500;
     start=1995y263d06h15m00s;
     mode=SX;
     source=HD123456;
     station=              EF:    0 sec:  180 sec:    0 ft:    1A:  &n  :    1;
     station=              FD:    0 sec:  180 sec:    0 ft:    1A:  &cw :    1;
  endscan;
  scan 263-062000;
     start=1995y263d06h20m00s;
     mode=SX;
     source=HD123456;
     station=              HS:   20 sec:  120 sec:        :      :  &n  :    1;      *Mark 5A
     station=              JB:    0 sec:  150 sec:    0 ft:    1A:      :    1;
  endscan;
  scan 263-063000;
     start=1995y263d06h30m00s;
     mode=SX;
     source=3C123;
     station=              EF:   10 sec:  300 sec: 2000 ft:    1B:  &n  :    1;
     station=              FD:    0 sec:  300 sec: 2000 ft:    1B:  &np :    1;      *'plunge' mode
  endscan;
  scan 263-063500;
     start=1995y263d06h35m00s;
     mode=SX;
     source=3C123;
     station=              HS:   20 sec:  300 sec:        :      : &ccw :    1;      *Mark 5A
     station=              JB:    0 sec:  300 sec: 1700 ft:    1B:      :    1;
  endscan;
*
***********************************************************************************************************************
*
*PRIMITIVE BLOCKS
*
*Primitive $block rules:
*  1. Content of each $block consists entirely of keyword 'defs'.
*  2. No 'refs' to other $blocks are permitted except to pick up parameters from an external file
*     ('ref <external_file>:<$blockname>=<keyword>').  'Links', however, are sometimes required between
*     $blocks (for example, between $FREQ, $BBC, $IF and $TRACKS blocks).
*  3. Each primitive $block may contain 'parameter=list of value' statements only for parameters which are
*     'native' to that block (see VEX Parameter Tables).
```

```
*
*******************************************************************************************************************
*
*------------------------------------------------------------------------------------------------------------------
$ANTENNA;                        *antenna parameters
*------------------------------------------------------------------------------------------------------------------
*
*Note: The 'Sector ID' defined in the 'pointing_sector=' statements is a 'link' which may be referenced in the $SCHED
*      block to specify that a particular scan is to be taken in a particular antenna-pointing sector.
*
  def EFLSBERG;
    antenna_diam = 100 m;
    axis_type = az : el;
    axis_offset = 0 m;
*                   type   slew     settle
    antenna_motion = az: 90 deg/min: 2 sec;
    antenna_motion = el: 30 deg/min: 1 sec;
  *                   Sector  Axis   LoLimit  HiLimit  Axis  LoLimit HiLimit
  *                     ID
    pointing_sector = &ccw   : az :  -90 deg:   90 deg: el :  0 deg:  88 deg;     *ccw cable wrap zone
    pointing_sector = &n     : az :   90 deg:  270 deg: el :  0 deg:  88 deg;     *neutral cable wrap zone
    pointing_sector = &cw    : az :  270 deg:  450 deg: el :  0 deg:  88 deg;     *cw cable wrap zone
  enddef;
  *
  def VLBA;
    antenna_diam = 25 m;
    axis_type = az : el;
    axis_offset = .123 m;
                    type   slew     settle
    antenna_motion = az: 90 deg/min: 2 sec;
    antenna_motion = el: 30 deg/min: 1 sec;
*                    Sector  Axis   LoLimit  HiLimit  Axis  LoLimit HiLimit
  *                     ID
    pointing_sector = &ccw   : az : -90 deg:   90 deg: el :  0 deg:  88 deg;     *ccw cable wrap zone
    pointing_sector = &n     : az :  90 deg:  270 deg: el :  0 deg:  88 deg;     *neutral cable wrap zone
    pointing_sector = &cw    : az : 270 deg:  450 deg: el :  0 deg:  88 deg;     *cw cable wrap zone
    pointing_sector = &ccwp  : az : -90 deg:   90 deg: el : 92 deg: 120 deg;     *ccw cable wrap zone, plunge mode
    pointing_sector = &np    : az :  90 deg:  270 deg: el : 92 deg: 120 deg;     *neutral cable wrap zone, plunge mode
    pointing_sector = &cwp   : az : 270 deg:  450 deg: el : 92 deg: 120 deg;     *cw cable wrap zone, plunge mode
  enddef;
  *
  def HAYSTACK;
    ref <external_file>:$ANTENNA = HAYSTACK;
  enddef;
  *
  def JODRELL_MK2;
    ref <external_file>:$ANTENNA = JODRELL_MK2;
  enddef;
  *
*------------------------------------------------------------------------------------------------------------------
$BBC;                            *BBC and IF assignments
*------------------------------------------------------------------------------------------------------------------
*Notes:
*  1. 'BBC ID' is a link to the selected $FREQ key.
*  2. 'IF ID' is a link to the selected $IF key.
*  3. All 'BBC ID's defined in the selected $FREQ 'def' must be present in the selected $BBC 'def', but not
*     necessarily vice versa (i.e. the defined BBC's may be a superset of channels defined in the referenced
*     $FREQ block).
*
*
  def EF/S4X4;
    ref <external_file>:$BBC = EF/SX;
  enddef;
  *
  def VLBA/S4X4;
  *                BBC   Physical     IF
  *                 ID     BBC#       ID
    BBC_assign = &BBCa  :   1    : &IF_XR1;
    BBC_assign = &BBCb  :   2    : &IF_XL1;
    BBC_assign = &BBCc  :   3    : &IF_SR1;
    BBC_assign = &BBCd  :   4    : &IF_SL1;
  enddef;
  *
  def HS/S4;
*                  BBC     BBC        IF
*                   ID      #         ID
    BBC_assign = &BBCc  :   5    : &IF_SR;
    BBC_assign = &BBCd  :   6    : &IF_SL;
  enddef;
  *
  def JB/S4X4;
```

```
    ref <external_file>:$BBC = JB/SX;
  enddef;
  *
  def SX14;
  *               BBC   Physical     IF
  *                ID     BBC#       ID
    BBC_assign = &BBCa  :   1    : &IF_XR;
    BBC_assign = &BBCb  :   2    : &IF_XR;
    BBC_assign = &BBCc  :   3    : &IF_XR;
    BBC_assign = &BBCd  :   4    : &IF_XR;
    BBC_assign = &BBCe  :   5    : &IF_XR;
    BBC_assign = &BBCf  :   6    : &IF_XR;
    BBC_assign = &BBCg  :   7    : &IF_XR;
    BBC_assign = &BBCh  :   8    : &IF_XR;
    BBC_assign = &BBCi  :   9    : &IF_SR;
    BBC_assign = &BBCj  :  10    : &IF_SR;
    BBC_assign = &BBCk  :  11    : &IF_SR;
    BBC_assign = &BBCl  :  12    : &IF_SR;
    BBC_assign = &BBCm  :  13    : &IF_SR;
    BBC_assign = &BBCn  :  14    : &IF_SR;
  enddef;
*
*-----------------------------------------------------------------------------------------------------------------------
$CLOCK;                           *clock parameters
*-----------------------------------------------------------------------------------------------------------------------
*
  def EF;
*                  Valid from     clock_early  clock_early_epoch    rate
    clock_early = 1995y263d06h00m :  2.5 usec :  1995y132d00h08m0s : 1.2e-12;
    clock_early = 1995y263d12h00m :  1,5 usec;                              *clock 'break'
  enddef;
*
  def FD;
    ref <external_file>:$CLOCK = VLBA-FD;
  enddef;
  *
  def HS;
    clock_early= : -3.5 usec;                     *Valid though entire experiment
  enddef;
  *
  def JB;
    clock_early= : 10.2 usec;
  enddef;
  *
*-----------------------------------------------------------------------------------------------------------------------
$DAS;                              *data-acquisition system parameters
*-----------------------------------------------------------------------------------------------------------------------
*
*Note:
*    A 'drive-number offset' is associated with each headstack in the recording system.  This number is added to the
*    tape drive # specified in the 'station=' statement in the $SCHED block in order to determine the physical
*    drive number.  For single-drive systems, the 'drive-number offset' will normally be =0, so that the drive#
*    assigned in the $SCHED section is the actual physical drive#.  In the case of multiple simultaneous drives,
*    the drive-number offset must be different for each headstack so that the drive specification in the $SCHED
*    block properly maps into the physical drive#'s being used.
  *
  def VLBA/1_DRIVE;                       *single-drive VLBA DAS
    record_transport_type=VLBA;
    electronics_rack_type=VLBA;
    number_drives=1;
*             hdstk#               drv#offst
    headstack =   1   : read/write :    0    ;          *1 headstack on 1 drive
  enddef;
  *
  def VLBA/2_DRIVES;                      *simultaneous dual-drive VLBA DAS
    record_transport_type=VLBA;
    electronics_rack_type=VLBA;
    number_drives=2;
*             hdstk#               drv#offst
    headstack =   1   : read/write :    0    ;          *2 headstacks, 1 on each of 2 drives
    headstack =   2   : read/write :    1   ;
  enddef;
*
  def VLBA/GEOD;                          *'geodetic' VLBA DAS with 14 BBC's
    record_transport_type=VLBA;
    electronics_rack_type=VLBAG;
    number_drives=1;
*             hdstk#               drv#offst
    headstack =   1   : read/write :    0    ;          *1 headstack on 1 drive
  enddef;
  *
  def Mark4;
    record_transport_type=Mark4;
    electronics_rack_type=Mark4;
    number_drives=1;
*              hdstk#               drv#offst
    headstack =   1   : write      :    0    ;          *2 headstacks on 1 drive
    headstack =   2   : read/write :    0    ;
  enddef;
  *
  def Mark3A;
    record_transport_type=Mark3A;
    electronics_rack_type=Mark3A;
    number_drives=1;
*              hdstk#                drv#offst
    headstack =   1   :   write    :    0    ;          *1 write headstack
    headstack =   2   :   read     :    0    ;          *1 read headstack
  enddef;
  *
  def MARK5A;            *single Mark 5A w/Mark 4 formatter
    record_transport_type=Mark5A : 1.0 ;
    electronics_rack_type=Mark4;
    number_drives=1
    headstack =   1   ::    0    ;
    headstack =   2   ::    0    ;
  enddef;
  *
  def MARK5A/DUAL;            *dual Mark 5A's w/Mark 4 formatter (for 2 Gbps)
    record_transport_type=Mark5A;
    electronics_rack_type=Mark4;
    number_drives=2
    headstack =   1   ::    0    ;     *drive 0
    headstack =   2   ::    0    ;     *drive 0
    headstack =   3   ::    1    ;     *drive 1
    headstack =   4   ::    1    ;     *drive 1
  enddef;

*def S2;
    record_transport_type=S2;
    electronics_rack_type=Mark4;
  *              duration   speed  #tapes
    tape_length = 515 min :  slp  :  8;
  *                 type   early strt  late stop  min gap
    tape_motion = adaptive : 2.5 min  : 0.5 min : 3 min;
  enddef;
  *
  def 33KBPI;
    record_density=33333 bpi;                *'raw' bit density per track
  enddef;
  *
  def 56KBPI;
    record_density=56250 bpi;
  enddef;
  *
  def 8820FT;
    tape_length = 8220 ft;
  enddef;
  *
  def 17640FT;
    tape_length = 17640 ft;
  enddef;
  *
  def EF_VLBA_ID;
    recording_system_ID = 4;
  enddef;
  *
  def FD_VLBA_ID;
    recording_system_ID = 10;
  enddef;
  *
  def HS_MARK5A_ID;            *Mark 4 formatter ID
    recording_system_ID = 6;
  enddef;
  *
  def JB_MARK4_ID;
    recording_system_ID = 19;
  enddef;
  *
  def START/STOP;
    tape_motion = start_stop;                 *no early start
  enddef;
  *
  def EARLY_START_20;
    tape_motion = start_stop:20 sec;          *tape starts 20 seconds before expected first good data.
  enddef;
  *
  def CONTINUOUS;
    *Other tape_motion modes may be added as necessary for particular observatories or recording systems.
    tape_motion = continuous;                 *continuous tape motion
  enddef;
  *
  def MASTER;
    *This parameter is relevant only in a cluster of antennas sharing the same DAS, and is included in the $STATION
    *'def' selected to contol the tape transport.  Other stations within the cluster will simply record data according
    *to the schedule defined by the 'master'.
    tape_control=master;
  enddef;
*
*------------------------------------------------------------------------------------------------------------------
$EOP;                             *earth-orientation parameters
*------------------------------------------------------------------------------------------------------------------
*
  def EOP129;
    TAI-UTC= 0.526 sec;
    A1-TAI= 22.0 sec;
    eop_ref_epoch=1995y129d;
    num_eop_points=5;
    eop_interval=24 hr;
    ut1-utc= 0.1039 sec: 0.1052 sec: 0.1064 sec: 0.1078 sec: 0.1084 sec;
    x_wobble=0.231 asec: 0.232 asec: 0.233 asec: 0.234 asec: 0.235 asec;
    y_wobble=-.123 asec: -.124 asec: -0.125 asec: -0.126 asec: -0.127 asec;
  enddef;
*
*------------------------------------------------------------------------------------------------------------------
$EXPER;                           *general experiment information
*------------------------------------------------------------------------------------------------------------------
*
  def EXP1387;
    exper_name=RDWPS1;
    exper_description="This description can be up to 128 characters long."
    exper_nominal_start=1995y132d05h00m;
    exper_nominal_stop=1995y132d12h00m;
    PI_name=Joe_Schmoe;
    PI_email=jschoe@mycomputer.pu.edu;
    contact_name=John_Doe;
    contact_email=jdoe@myplace.edu;
    scheduler_name=Jane_Smith;
    scheduler_email=jsmith@abc.school.edu;
    target_correlator=EVN;
  enddef;
*
*------------------------------------------------------------------------------------------------------------------
$FREQ;                            *frequency and channel parameters
*------------------------------------------------------------------------------------------------------------------
*
*The $FREQ block specifies the sideband, channel bandwidth and exact RF sky frequency of each channel to
*be recorded in an observation.  In a simple experiment with all stations recording the same set of channels, a 'ref'
*to a single $FREQ 'def' will generally serve for all stations; in this case, the 'ref $FREQ=<keyword>;'
*statement may be placed in the $GLOBAL block, individually for each station in the $STATION 'defs', or as a single
*'ref' in the appropriate $MODE 'def'.  In more complicated situations, more than one $FREQ 'def' may have to be
*referenced (as, for instance, in the example of this file).
*
*Notes:
*  1. All 'chan_def=' statements referenced with respect to a given observation are assumed to be active recorded
*     channels.
*  2. 'Band_ID' is a RF-freq-band ID which 'links' to the $SOURCE section (used for scheduling purposes only).
*  3. 'Trks_ID' is an identifier which 'links' to the selected $TRACKS block to define the data layout on tape for
*     that frequency channel.  If frequency switching is being used, more than one channel may have the same
*     'Trks_ID'.
*  4. 'BBC_ID' is a BBC identifier which 'links' to the selected $BBC block to define the physical
*     BBC assignment for each frequency channel.
*  5. 'Phase-cal_ID' is a phase-cal identifer which 'links' to the selected 'def' in the $PHASE_CAL_DETECT block
*     to specify the phase-cal frequencies to be detected at a particular station.  A null 'Phase-cal_ID' field
*     implies no phase-cal will be detected in the corresponding channel; the same result can be achieved by
*     specifying a 'Phase-cal ID' which links to a 'Phase-cal detect=' statement with no tones specified.
*
  def X4;
  *          Band     Sky freq     Net      Chan    Chan    BBC     Phase-cal
  *           ID     at 0Hz BBC     SB       BW      ID      ID        ID
    chan_def= &X :   8210.99 MHz :  U  :   2 MHz : &CH1  : &BBCa  : &USB_CAL  ;
    chan_def= &X :   8210.99 MHz :  L  :   2 MHz : &CH2  : &BBCa  : &LSB_CAL  ;
    chan_def= &X :   8212.99 MHz :  U  :   2 MHz : &CH3  : &BBCb  : &USB_CAL  ;
    chan_def= &X :   8212.99 MHz :  L  :   2 MHz : &CH4  : &BBCb  : &LSB_CAL  ;
    sample_rate = 4 Ms/sec;
  enddef;
  *
  def S4;
  *          Band     Sky freq     Net      Chan    Chan    BBC     Phase-cal
  *           ID     at 0Hz BBC     SB       BW      ID      ID        ID
    chan_def= &S :   2210.99 MHz :  U  :   2 MHz : &CH5  : &BBCc  : &USB_CAL  ;
    chan_def= &S :   2210.99 MHz :  L  :   2 MHz : &CH6  : &BBCc  : &LSB_CAL  ;
    chan_def= &S :   2212.99 MHz :  U  :   2 MHz : &CH7  : &BBCd  : &USB_CAL  ;
    chan_def= &S :   2212.99 MHz :  L  :   2 MHz : &CH8  : &BBCd  : &LSB_CAL  ;
    sample_rate = 4 Ms/sec;
  enddef;
  *
  def X4_SWITCHED;                      *Frequency-switched geodetic sequence
  *Note: Example to show the 'chan_def=' format when frequency-switching is active.
  *      In this example, switching is between two states (1 and 2);
  *      two BBC's (BBCa, BBCd) are at the same frequency for both states while the other two
  *      (BBCb, BBCc) are set for different observing frequencies in the two states.
  *      Switching timing is started at the observation start time:
  *      10 sec are spent in state 1, alternating with 10 sec in state 2.
  *
  *                 switching ref    State 1    State 2
  *                                  period     period
    switching_cycle = wrt_obs_start :  10 sec  :  10 sec ;
  *
  *          Band     Sky freq     Net      chan    Trks    BBC     Phase-cal   State#(s)
  *           ID     at 0Hz BBC     SB       BW      ID      ID        ID
    chan_def= &X :   8212.99 MHz :  U  :   2 MHz : &CH1  : &BBCa  : &USB_CAL  :  1  :  2  ;
    chan_def= &X :   8252.99 MHz :  U  :   2 MHz : &CH2  : &BBCb  : &LSB_CAL  :  1  ;
    chan_def= &X :   8352.99 MHz :  U  :   2 MHz : &CH2  : &BBCb  : &USB_CAL  :  2  ;
    chan_def= &X :   8732.99 MHz :  U  :   2 MHz : &CH3  : &BBCc  : &LSB_CAL  :  1  ;
    chan_def= &X :   8912.99 MHz :  U  :   2 MHz : &CH3  : &BBCc  : &USB_CAL  :  2  ;
    chan_def= &X :   8932.99 MHz :  U  :   2 MHz : &CH4  : &BBCd  : &LSB_CAL  :  1  :  2  ;
    sample_rate = 4 Ms/sec;
  enddef;
  *
  def SX14;                        *example of standard Mark IIIA geodetic 14-frequency setup
  *          Band     Sky freq     Net      chan    Trks    BBC     Phase-cal
  *           ID     at 0Hz BBC     SB       BW      ID      ID        ID
    chan_def= &X :   8212.99 MHz :  U  :   2 MHz : &CH1  : &BBCa  : &USB_CAL  ;
    chan_def= &X :   8252.99 MHz :  U  :   2 MHz : &CH2  : &BBCb  : &LSB_CAL  ;
    chan_def= &X :   8352.99 MHz :  U  :   2 MHz : &CH3  : &BBCc  : &USB_CAL  ;
    chan_def= &X :   8512.99 MHz :  U  :   2 MHz : &CH4  : &BBCd  : &LSB_CAL  ;
    chan_def= &X :   8732.99 MHz :  U  :   2 MHz : &CH5  : &BBCe  : &USB_CAL  ;
    chan_def= &X :   8852.99 MHz :  U  :   2 MHz : &CH6  : &BBCf  : &LSB_CAL  ;
    chan_def= &X :   8912.99 MHz :  U  :   2 MHz : &CH7  : &BBCg  : &USB_CAL  ;
    chan_def= &X :   8932.99 MHz :  U  :   2 MHz : &CH8  : &BBCh  : &LSB_CAL  ;
    chan_def= &S :   2220.99 MHz :  U  :   2 MHz : &CH9  : &BBCi  : &USB_CAL  ;
    chan_def= &S :   2230.99 MHz :  U  :   2 MHz : &CH10 : &BBCj  : &LSB_CAL  ;
    chan_def= &S :   2250.99 MHz :  U  :   2 MHz : &CH11 : &BBCk  : &USB_CAL  ;
    chan_def= &S :   2305.99 MHz :  U  :   2 MHz : &CH12 : &BBCl  : &LSB_CAL  ;
    chan_def= &S :   2340.99 MHz :  U  :   2 MHz : &CH13 : &BBCm  : &USB_CAL  ;
    chan_def= &S :   2345.99 MHz :  U  :   2 MHz : &CH14 : &BBCn  : &LSB_CAL  ;
    sample_rate = 4 Ms/sec;
  enddef;
*
*-----------------------------------------------------------------------------------------
$HEAD_POS;                        *headstack-position parameters
*-----------------------------------------------------------------------------------------
*
  def VLBA/1_HDSTK;
  *                  pos    hdstk1    hdstk2   hdstk3  hdstk4
  *                  ref#     pos      pos      pos      pos
    headstack_pos =   1  : -319 um;
    headstack_pos =   2  :   31 um;
    headstack_pos =   3 :  -271 um;
    headstack_pos =   4  :   79 um;
    headstack_pos =   5  : -223 um;
    headstack_pos =   6  :  127 um;
    headstack_pos =   7  : -175 um;
    headstack_pos =   8  :  175 um;
    headstack_pos =   9 :  -127 um;
    headstack_pos =  10  :  223 um;
    headstack_pos =  11  :  -79 um;
    headstack_pos =  12  :  271 um;
    headstack_pos =  13  :  -31 um;
    headstack_pos =  14  :  319 um;
  enddef;
  *
  def VLBA/2_DRIVES;
  *                  pos    hdstk1    hdstk2    hdstk3    hdstk4
  *                  ref#     pos       pos       pos       pos
    headstack_pos =   1  : -319 um : -319 um;
    headstack_pos =   2  :   31 um :   31 um;
    headstack_pos =   3 :  -271 um : -271 um;
    headstack_pos =   4  :   79 um :   79 um;
    headstack_pos =   5  : -223 um : -223 um;
    headstack_pos =   6  :  127 um :  127 um;
    headstack_pos =   7  : -175 um : -175 um;
    headstack_pos =   8  :  175 um :  175 um;
    headstack_pos =   9 :  -127 um : -127 um;
    headstack_pos =  10  :  223 um :  223 um;
    headstack_pos =  11  :  -79 um :  -79 um;
    headstack_pos =  12  :  271 um :  271 um;
    headstack_pos =  13  :  -31 um :  -31 um;
    headstack_pos =  14  :  319 um :  319 um;
  enddef;
  *
  def MARK4/2_HDSTKS;
  *                  pos   hdstk1    hdstk2    hdstk3    hdstk4
  *                  ref#   pos       pos       pos       pos
    headstack_pos =   1  : -319 um : -271 um;
    headstack_pos =   2  :   31 um :   79 um;
    headstack_pos =   3 :  -223 um : -175 um;
    headstack_pos =   4  :  127 um :  175 um;
    headstack_pos =   5  : -127 um :  -79 um;
    headstack_pos =   6  :  223 um :  271 um;
  enddef;
  *
*-----------------------------------------------------------------------------------------------------------------
$IF;                                *receiver and IF parameters
*-----------------------------------------------------------------------------------------------------------------
*
*Notes:
* 1. The $IF block serves to define the IF bands and phase-cal frequencies used in the observations.
* 2. The 'IF ID' is a link to the selected $BBC block and $SEFD block (used for scheduling only).
* 3. The 'Physical Name' is the IF letter or number specifying which IF is selected.  For the VLBA system, IF's
*    A,B,C,D may each be selected with either a 'Normal' or 'External' input, leading to designations 'AN','AE','BN',
*    'BE', 'CN', 'CE', 'DN', 'DE'.  For the Mark III system, IF's 1,2,3 may each be selected with either a 'Normal'
*    or 'Alternate' input, leading to the designations '1N', '1A', '2N', '2A', '3I', '3O'.
* 4. The 'Total LO' is the effective total LO frequency (not including BBC LO's) for the associated IF band.
*    The 'Total LO', in combination with the sky frequency in the $FREQ block, allows the LO frequencies in the
*    individual BBC's to be computed.
* 5. 'Net SB' is the net frequency sideband ('U' or 'L') of the IF itself.
* 6. Null value or omission of 'Phase-cal freq spacing' indicates no phase-cal present.
  *
  def VLBA/X;
  *           IF    Physical Pol    Total     Net    Phase-cal     P-cal base
  *           ID      Name           LO        SB   freq spacing      freq
    if_def= &IF_XR1 :  AN   : R :   7600 MHz :  U  :  1 MHz      :  0 Hz  ;
    if_def= &IF_XR2 :  BN   : R :   9600 MHz :  L  :  1 MHz      :  0 Hz  ;
    if_def= &IF_XL1 :  CN   : L :   7600 MHz :  U  :  1 MHz      :  0 Hz  ;
    if_def= &IF_XL2 :  DN   : L :   9600 MHz :  L  :  1 MHz      :  0 Hz  ;
  enddef;
  *
  def VLBA/S;
  *           IF    Physical Pol    Total     Net
  *           ID       ID            LO        SB
    if_def= &IF_SR  :  AE   : R :   2900 MHz :  L  ;           *no phase-cal
    if_def= &IF_SL  :  BE   : L :   2900 MHz :  L  ;           *no phase-cal
  enddef;
  *
  def CDP/SX_WIDE;
  *           IF    Physical Pol    Total     Net    Phase-cal
  *           ID       ID   :        LO        SB   freq spacing
    if_def= &IF_XR  :  1A   : R :   8080 MHz :  U  :  1 MHz  ;
    if_def= &IF_SR  :  2A   : L :   2020 MHz :  U  :  1 MHz  ;
  enddef;
*
*-----------------------------------------------------------------------------------------------------------------
$PASS_ORDER;                         *pass order
*-----------------------------------------------------------------------------------------------------------------
*
*Defines tape-pass ordering in Mark IIIA, Mark IV and VLBA systems
*
*Notes:
* 1. Each tape pass is specified by a numeric headstack-position reference (defined by the selected $HEADPOS def)
*    plus an alphabetic subpass identifier (defined by selected $TRACKS def).
* 2. First pass is assumed to be in forward direction.
*
  def VLBA/32;                           *32-tracks recording simultaneously
  *             F    R    F    R    F    R    F    R    F    R     F     R     F     R
    pass_order = 1A : 2A : 3A : 4A : 5A : 6A : 7A : 8A : 9A : 10A : 11A : 12A : 13A : 14A ;
  enddef;
  *
  def VLBA/16;                       *16-tracks recording simultaneously (2 subpasses/headstack-pos)
  *            F  R  F  R  F  R  F  R  F  R  F  R  F  R  F  R  F  R   F  R   F  R   F  R   F  R   F  R
    pass_order =1A:2A:1B:2B:3A:4A:3B:4B:5A:6A:5B:6B:7A:8A:7B:8B:9A:10A:9B:10B:11A:12A:11B:12B:13A:14A:13B:14B;
  enddef;
  *
  def VLBA/8;                        *8 tracks recording simultaneously (4 subpasses/headstack-pos)
  *         F  R  F  R  F   R  F   R  F   R  F   R   F   R   F   R   F   R   F   R   F   R   F   R   F   R   F   R
  pass_order=1A:2A:1B:2B:1C: 2C:1D: 2D:3A: 4A:3B: 4B: 3C: 4C: 3D: 4D: 5A: 6A: 5B: 6B: 5C: 6C: 5D: 6D: 7A: 8A: 7B: 8B:
             7C:8C:7D:8D:9A:10A:9B:10B:9C:10C:9D:10D:11A:12A:11B:12B:11C:12C:11D:12D:13A:14A:13B:14B:13C:14C:13D:14D;
  enddef;
  *
  def MARK4/64;                      *64-tracks recording simultaneously
  *             F    R    F    R    F    R
    pass_order = 1A : 2A : 3A : 4A : 5A : 6A ;
  enddef;
  *
  def S2/4;                          *4 groups of 4 cassettes each
*                   grp grp grp grp
    S2_group_order = 0 : 1 : 2 : 3 ;
  enddef;
*
*-----------------------------------------------------------------------------------------------------------------------
```

$PHASE_CAL_DETECT;   *phase-cal-detection parameters

```
*-----------------------------------------------------------------------------------------------------------------------
*
*Notes:
* 1. The pcal_ID is linked to the selected $FREQ 'def' to define the details of the phase-cal in each frequency
*    channel.
* 2. Phase-cal detection will be done on the set of specified 'tone#'s, which are listed in order of preference in
*    case more tones are specified than can be detected.  The tones are numbered positively from the low (DC) edge
*    of the BBC output band, with tone number '1' being the first tone *above* DC.  Tones may also be specified as
*    negative numbers corresponding to their position from the bandedge, with tone number '-1' being the first tone
*    *below* bandedge.  A tone number of '0' specifies state counting, rather than phase-cal detection, should take
*    place.
  *
  def STANDARD;
  *                     pcal_ID  tone# tone#
    phase_cal_detect = &USB_CAL :  1  :  -1  ;         *detect lowest and highest frequency tones in band
    phase_cal_detect = &LSB_CAL :  1   ;              *detect lowest-frequency tone in band
  enddef;
*
*-----------------------------------------------------------------------------------------------------------------------
```

$PROCEDURES;   *procedure parameters

```
*-----------------------------------------------------------------------------------------------------------------------
*
  def STANDARD1;
  *Required procedures
    tape_change =          420 sec;
    headstack_motion =         6 sec;
    new_source_command =     5 sec;
    new_tape_setup =        20 sec;
  *Optional procedures
    setup_always =   on :   20 sec;
    parity_check =   on :   70 sec;
    tape_prepass =   off:  600 sec;
  enddef;
  *
  *Calibration procedure timings
  def VLBA_CAL;
  *                     time   procedure
  *                     req'd     name
    preob_cal  =  on : 10 sec : preob;
    midob_cal  =  on : 20 sec : midob;
    postob_cal =  on : 20 sec : postob;
  enddef;
  *
  def STANDARD_CAL;
    preob_cal  =  on : 10 sec : preob;
    midob_cal  =  on : 15 sec : midob;
    postob_cal = off : 20 sec : postob;               *procedure not used; no time will be allocated
  enddef;
  *
  def SX2C;
    procedure_name_prefix = SX2C;                     *prescribes use of standard library of procedures SX2C
  enddef;
  *
*-----------------------------------------------------------------------------------------------------------------------
```

```
$ROLL;                            *recording-system barrel-roll parameters
*------------------------------------------------------------------------------------------
*
*Notes:
* 1. Barrel-roll tables are extremely detailed in order to take advantage of flexibility in Mark 4 formatter.
*    The VLBA formatter supports a small subset of the possible Mark 4 barrel-roll modes.
* 2. The 'home trk' parameter in the 'roll=' statement is the track to which data would be written if barrel-roll
*    is not active.  The 'output track' is the actual track to which data is written, specified at each step of
*    the roll.
*
  def VLBA/8;                           *standard 8-track 8-position VLBA barrel-roll
    roll = on;
    roll_reinit_period = 2 sec;         *barrel-roll sequence reinitialized every 2 sec
    roll_inc_period = 1;                *barrel-roll increment period (frames)
  *
  *                -------------output track----------------------
  *         hdstk home  roll  roll  roll  roll  roll  roll  roll  roll
  *                trk  step0 step1 step2 step3 step4 step5 step6 step7
    roll_def=  1  :  2 :   2 :   4 :   6 :   8 :  10 :  12 :  14 :  16;
    roll_def=  1  :  4 :   4 :   6 :   8 :  10 :  12 :  14    16 :   2;
    roll_def=  1  :  6 :   6 :   8 :  10 :  12 :  14 :  16 :   2 :   4;
    roll_def=  1  :  8 :   8 :  10 :  12 :  14 :  16 :   2 :   4 :   6;
    roll_def=  1  : 10 :  10 :  12 :  14 :  16 :   2 :   4 :   6 :   8;
    roll_def=  1  : 12 :  12 :  14 :  16 :   2 :   4 :   6     8 :  10;
    roll_def=  1  : 14 :  14 :  16 :   2 :   4 :   6 :   8 :  10 :  12;
    roll_def=  1  : 16 :  16 :   2 :   4 :   6 :   8 :  10 :  12 :  14;
  *
    roll_def=  1  : 18 :  18 :  20 :  22 :  24 :  26 :  28 :  30 :  32;
    roll_def=  1  : 20 :  20 :  22 :  24 :  26 :  28 :  30 :  32 :  18;
    roll_def=  1  : 22 :  22 :  24 :  26 :  28 :  30 :  32 :  18 :  20;
    roll_def=  1  : 24 :  24 :  26 :  28 :  30 :  32 :  18 :  20 :  22;
    roll_def=  1  : 26 :  26 :  28 :  30 :  32 :  18 :  20 :  22 :  24;
    roll_def=  1  : 28 :  28 :  30 :  32 :  18 :  20 :  22 :  24 :  26;
    roll_def=  1  : 30 :  30 :  32 :  18 :  20 :  22 :  24 :  26 :  28;
    roll_def=  1  : 32 :  32 :  18 :  20 :  22 :  24 :  26 :  28 :  30;
  *
    roll_def=  1  :  3 :   3 :   5 :   7 :   9 :  11 :  13 :  15 :  17;
    roll_def=  1  :  5 :   5 :   7:    9:   11:   13:   15:   17:    3;
    roll_def=  1  :  7 :   7 :   9 :  11 :  13 :  15 :  17 :   3 :   5;
    roll_def=  1  :  9 :   9 :  11 :  13 :  15 :  17 :   3 :   5 :   7;
    roll_def=  1  : 11 :  11 :  13 :  15 :  17 :   3 :   5 :   7 :   9;
    roll_def=  1  : 13 :  13 :  15 :  17 :   3 :   5 :   7 :   9 :  11;
    roll_def=  1  : 15 :  15 :  17 :   3 :   5 :   7 :   9 :  11 :  13;
    roll_def=  1  : 17 :  17 :   4 :   5 :   7 :   9 :  11 :  13 :  15;
  *
    roll_def=  1  : 19 :  19 :  21 :  23 :  25 :  27 :  29 :  31 :  33;
    roll_def=  1  : 21 :  21 :  23 :  25 :  27 :  29 :  31 :  33 :  19;
    roll_def=  1  : 23 :  23 :  25 :  27 :  29 :  31 :  33 :  19 :  21;
    roll_def=  1  : 25 :  25 :  27 :  29 :  31 :  33 :  19 :  21 :  23;
    roll_def=  1  : 27 :  27 :  29 :  31 :  33 :  19 :  21 :  23 :  25;
    roll_def=  1  : 29 :  29 :  31 :  33 :  19 :  21 :  23 :  25 :  27;
    roll_def=  1  : 31 :  31 :  33 :  19 :  21 :  23 :  25 :  27 :  29;
    roll_def=  1  : 33 :  33 :  19 :  21 :  23 :  25 :  27 :  29 :  31;
  enddef;
  *
  def VLBA/16;                          *standard 16-track 16-position VLBA barrel-roll
    roll = on;
    roll_reinit_period = 2 sec;         *barrel-roll sequence reinitialized every 2 sec
    roll_inc_period = 1;                *barrel-roll increment period (frames)
  *
  *       hdstk home  roll  roll  roll  roll  roll  roll  roll  roll  roll  roll  roll  roll  roll  roll  roll  roll
  *              trk  step0 step1 step2 step3 step4 step5 step6 step7 step8 step9 stp10 stp11 stp12 stp13 stp14 stp15
    roll_def= 1 :  2 :   2 :   4 :   6 :   8 :  10 :  12 :  14 :  16 :  18 :  20 :  22 :  24 :  26 :  28 :  30 :  32 ;
    roll_def= 1 :  4 :   4 :   6 :   8 :  10 :  12 :  14    16 :   2 :  20 :  22 :  24 :  26 :  28 :  30 :  32 :  18 ;
    roll_def= 1 :  6 :   6 :   8 :  10 :  12 :  14 :  16 :   2 :   4 :  22 :  24 :  26 :  28 :  30 :  32 :  18 :  20 ;
    roll_def= 1 :  8 :   8 :  10 :  12 :  14 :  16 :   2 :   4 :   6 :  24 :  26 :  28 :  30 :  32 :  18 :  20 :  22 ;
    roll_def= 1 : 10 :  10 :  12 :  14 :  16 :   2 :   4 :   6 :   8 :  26 :  28 :  30 :  32 :  18 :  20 :  22 :  24 ;
    roll_def= 1 : 12 :  12 :  14 :  16 :   2 :   4 :   6     8 :  10 :  28 :  30 :  32 :  18 :  20 :  22 :  24 :  26 ;
    roll_def= 1 : 14 :  14 :  16 :   2 :   4 :   6 :   8 :  10 :  12 :  30 :  32 :  18 :  20 :  22 :  24 :  26 :  28 ;
    roll_def= 1 : 16 :  16 :   2 :   4 :   6 :   8 :  10 :  12 :  14 :  32 :  18 :  20 :  22 :  24 :  26 :  28 :  30 ;
  *
    roll_def= 1 : 18 :  18 :  20 :  22 :  24 :  26 :  28 :  30 :  32 :   2 :   4 :   6 :   8 :  10 :  12 :  14 :  16 ;
    roll_def= 1 : 20 :  20 :  22 :  24 :  26 :  28 :  30 :  32 :  18 :   4 :   6 :   8 :  10 :  12 :  14 :  16 :   2 ;
    roll_def= 1 : 22 :  22 :  24 :  26 :  28 :  30 :  32 :  18 :  20 :   6 :   8 :  10 :  12 :  14 :  16 :   2 :   4 ;
    roll_def= 1 : 24 :  24 :  26 :  28 :  30 :  32 :  18 :  20 :  22 :   8 :  10 :  12 :  14 :  16 :   2 :   4 :   6 ;
    roll_def= 1 : 26 :  26 :  28 :  30 :  32 :  18 :  20 :  22 :  24 :  10 :  12 :  14 :  16 :   2 :   4 :   6 :   8 ;
    roll_def= 1 : 28 :  28 :  30 :  32 :  18 :  20 :  22 :  24 :  26 :  12 :  14 :  16 :   2 :   4 :   6     8 :  10 ;
    roll_def= 1 : 30 :  30 :  32 :  18 :  20 :  22 :  24 :  26 :  28 :  14 :  16 :   2 :   4 :   6 :   8 :  10 :  12 ;
    roll_def= 1 : 32 :  32 :  18 :  20 :  22 :  24 :  26 :  28 :  30 :  16 :   2 :   4 :   6 :   8 :  10 :  12 :  14 ;
  *
    roll_def= 1 :  3 :   3 :   5 :   7 :   9 :  11 :  13 :  15 :  17 :  19 :  21 :  23 :  25 :  27 :  29 :  31 :  33 ;
    roll_def= 1 :  5 :   5 :   7 :   9 :  11 :  13 :  15    17 :   3 :  21 :  23 :  25 :  27 :  29 :  31 :  33 :  19 ;
    roll_def= 1 :  7 :   7 :   9 :  11 :  13 :  15 :  17 :   3 :   5 :  23 :  25 :  27 :  29 :  31 :  33 :  19 :  21 ;
    roll_def= 1 :  9 :   9 :  11 :  13 :  15 :  17 :   3 :   5 :   7 :  25 :  27 :  29 :  31 :  33 :  19 :  21 :  23 ;
    roll_def= 1 : 11 :  11 :  13 :  15 :  17 :   3 :   5 :   7 :   9 :  27 :  29 :  31 :  33 :  19 :  21 :  23 :  25 ;
    roll_def= 1 : 13 :  13 :  15 :  17 :   3 :   5 :   7     9 :  11 :  29 :  31 :  33 :  19 :  21 :  23 :  25 :  27 ;
    roll_def= 1 : 15 :  15 :  17 :   3 :   5 :   7 :   9 :  11 :  13 :  31 :  33 :  19 :  21 :  23 :  25 :  27 :  29 ;
    roll_def= 1 : 17 :  17 :   3 :   5 :   7 :   9 :  11 :  13 :  15 :  33 :  19 :  21 :  23 :  25 :  27 :  29 :  31 ;
  *
    roll_def= 1 : 19 :  19 :  21 :  23 :  25 :  27 :  29 :  31 :  33 :   3 :   5 :   7 :   9 :  11 :  13 :  15 :  17 ;
    roll_def= 1 : 21 :  21 :  23 :  25 :  27 :  29 :  31 :  33 :  19 :   5 :   7 :   9 :  11 :  13 :  15 :  17 :   3 ;
    roll_def= 1 : 23 :  23 :  25 :  27 :  29 :  31 :  33 :  19 :  21 :   7 :   9 :  11 :  13 :  15 :  17 :   3 :   5 ;
    roll_def= 1 : 25 :  25 :  27 :  29 :  31 :  33 :  19 :  21 :  23 :   9 :  11 :  13 :  15 :  17 :   3 :   5 :   7 ;
    roll_def= 1 : 27 :  27 :  29 :  31 :  33 :  19 :  21 :  23 :  25 :  11 :  13 :  15 :  17 :   3 :   5 :   7 :   9 ;
    roll_def= 1 : 29 :  29 :  31 :  33 :  19 :  21 :  23 :  25 :  27 :  13 :  15 :  17 :   3 :   5 :   7     9 :  11 ;
    roll_def= 1 : 31 :  31 :  33 :  19 :  21 :  23 :  25 :  27 :  29 :  15 :  17 :   3 :   5 :   7 :   9 :  11 :  13 ;
    roll_def= 1 : 33 :  33 :  19 :  21 :  23 :  25 :  27 :  29 :  31 :  17 :   3 :   5 :   7 :   9 :  11 :  13 :  15 ;
  *
  enddef;
*
   def ROLL_OFF;
     roll = off;
   enddef;
*
*-----------------------------------------------------------------------------------------------------------------------
$SCHEDULING_PARAMS;             *Scheduling parameters
*-----------------------------------------------------------------------------------------------------------------------
*
*Note: This is an example of a literal block.  In the future, normal VEX statements may be defined in this section.
*
  def SKED1;
    start_literal();
      sched_program = SKED:Rev_950715
      *time control
      default_scan_length = 196 sec
      lookahead = 20 sec
      min_scan_length = 60 sec
      minimum_between_scans = 0 sec
      modular_scan_length = 10 sec
      *
      *user interface
      max_display_width_col = 79
      confirm = on
      mutual_vis = subnet
      low_SNR_reject = auto
     *user control values
      variable_scan_length = on
      min_sun_angle = 15 deg;
      tape_usage_sync = on
      sked_optimize = sky_coverage
      window = 1 hr
      maximize_num_obs = on
      minimize_idle = on
      minimize_slew = on
    end_literal();
  enddef;*
*
*-----------------------------------------------------------------------------------------------------------------------
$SEFD;                          *SEFD parameters
*-----------------------------------------------------------------------------------------------------------------------
*
*Note: This information is used only for auto-scheduling purposes so that the length of observations can be tied to
*      an expected observation SNR based on antenna SEFD's and source models (see $SEFD and $SOURCE blocks).
*
  def EF;
    sefd_model = Shaffer;            *defines model be used
  *          IF ID      SEFD     Model parameters
    sefd = &IF_XR1  :  100 Jy  :  1.0  : 0.954  : 0.0464;
    sefd = &IF_XR2  :  110 Jy  :  1.0  : 0.974  : 0.0263;
    sefd = &IF_XL1  :  120 Jy  :  1.0  : 0.573  : 0.0470;
    sefd = &IF_XL2  :  130 Jy  :  1.0  : 0.549  : 0.0470;
  enddef;
  *
  def VLBA-FD;
    sefd_model = Shaffer;            *defines model be used
  *          IF ID     SEFD      Model parameters
    sefd = &IF_XR1  :  750 Jy  :  1.0  : 0.954  : 0.0464;
    sefd = &IF_XR2  :  760 Jy  :  1.0  : 0.974  : 0.0263;
    sefd = &IF_XL1  :  750 Jy  :  1.0  : 0.573  : 0.0470;
    sefd = &IF_XL2  :  750 Jy  :  1.0  : 0.549  : 0.0470;
  enddef;
  *
  def HS;
    ref <external_file>:$SEFD = HS;
  enddef;
  *
  def JB;
    ref <external_file>:$SEFD = JB;
  enddef;
*
*----------------------------------------------------------------------------------------------------------------------
$SITE;                              *site parameters
*----------------------------------------------------------------------------------------------------------------------
  def EFLSBERG;
    site_type=fixed;
    site_name=EFLSBERG;       *full station name
    site_ID=EF;               *standard 2-char identifier
    *                   x           y            z
    site_position = 123456.4 m: 5432112.6 m: 563675.2 m;
    site_position_epoch = 1994y1d; site_position_ref = GLB914F1;
    site_velocity = 0.17 cm/yr: 0.025 cm/yr: 0.12 cm/yr;
    horizon_map_az=0 deg:20:55:60:70:85:100:105:115:220:230:245:270:280:300:305:330:345:360;
    horizon_map_el= 5 deg: 3: 6: 7: 5: 6:  5:  4:  3:  4:  3:  5:  4:  3:  4:  5:  6:  5;
    zen_atmos=7 nsec;
    ocean_load_vert = 3 cm: 90 deg;
    ocean_load_horiz =0.5 cm: 52 deg;
    occupation_code=a478;
  enddef;
  *
  def VLBA-FD;
    ref <external_file>:$SITE = VLBA-FD;
  enddef;
  *
  def HAYSTACK;
    ref <external_file>:$SITE = HAYSTACK;
  enddef;
  *
  def JODRELL_MK2;
    ref <external_file>:$SITE = JODRELL_MK2;
  enddef;
*
def VSOP;                                              *satellite
    site_type = earth_orbit;
    site_name = SAT1;
    site_ID=VS;
    inclination = 1.06 deg;
    eccentricity = 0.000313;
    arg_perigee = 67.3883 deg;
    ascending_node = 43.9513 deg;
    mean_anomaly = 297.168289 deg;
    semi-major_axis = 42166.129 km;
    mean_motion = 0.;                                  *rev/sidereal-day
    orbit_epoch = 1995y44d7h;
  enddef;
*
*----------------------------------------------------------------------------------------------------------------------
$SOURCE;                            *source parameters
*----------------------------------------------------------------------------------------------------------------------
  *
  def HD123456;
     source_type = star : calibrator;           *source type may be declared as 'dummy' if for pointing purposes only
     source_name = HD123456; IAU_name = 0102-0304;
     ra=01h02m03.456s; dec = -03d04'04.567"; ref_coord_frame = J2000; source_position_ref = GLB923Z;
     ra_rate = 1.e-2 asec/yr;   dec_rate = -1.e-3 asec/yr;  source_position_epoch = 1995y;
     *            comp# Band   Flux        MajAxis    Ratio      PA     Raofst     DECofst
     *                  ID
     source_model = 1  : &X  : 1.10 Jy  :  .20 asec :  1.0   :  0 deg :  0 asec :  0 asec ;
     source_model = 1  : &S  : 2.40 Jy  :  .22 asec :  1.5   :  5 deg :  0 asec :  0 asec ;
     *Note: Source model is for scheduling purposes only; 'freq band' is 'link' to the selected $FREQ def.
  enddef;
  *
  def 3C123;
    source_type = star : target;
    ref <external_file>:$SOURCE = 3C123;
  enddef;
*
  def SAT1;
    source_type = earth_satellite;
    source_name = SAT1;
    inclination = 1.06 deg;
    eccentricity = 0.000313;
    arg_perigee = 67.3883 deg;
    ascending_node = 43.9513 deg;
    mean_anomaly = 297.168289 deg;
    semi-major_axis = 42166.129 km;
    mean_motion = 0.;                                 *rev/sidereal-day
    orbit_epoch = 1995y44d7h;
  enddef;
*
*-----------------------------------------------------------------------------------------------------------
$TRACKS;                               *track mapping/multiplex parameters
*-----------------------------------------------------------------------------------------------------------
*
*Notes:
*  1.'Trks_ID' is a link to the $FREQ block.
*  2. All 'Trks_ID's defined in a selected $FREQ 'def' must be present in the selected $TRACKS 'def'.
*  3. Except for the 'VLBA_drive_sys_track=' statement, all references to 'track numbers' in the $TRACKS block are
*     more properly labelled as 'formatter-output numbers' since signal switching within the recorder may lead to
*     re-arranged physical track assignments.  The 'track number' label has been retained for convenience.
*  4. When recording 2-bit samples, the bits are identified as 'sign' and 'mag', as defined in Mark IV memo 205.7.
*     The samples are actually encoded as '00', '01', '10', '11', in order, going from most negative to most positive
*     voltage.  The first bit (MSB) is truly the sign, while the second bit (LSB) is encoded for statistical
*     uniformity to keep the tape channel happier.  The designation 'sign' and 'mag' is used here, even though not
*     quite correct, so that the 1-bit sample case is designated as 'sign', which is both accurate and more
*     comfortable than 'MSB'.
*
  def VLBA/XX-8-2/8;                  *VLBA mode XX-8-2 recorded on 8 tracks
  *
  *              subpass  hdstk   trk    Bitstream1      Bitstream2
  *                        #      #   trksID:sign/mag
    fanin_def   =   A   :   1  :   2   : &CH1:sign  :   &CH1:mag;   *2-to-1 fanin
    fanin_def   =   A   :   1  :   4   : &CH2:sign  :   &CH2:mag;
    fanin_def   =   A   :   1  :   6   : &CH3:sign  :   &CH3:mag;
    fanin_def   =   A   :   1  :   8   : &CH4:sign  :   &CH4:mag;
    fanin_def   =   A   :   1  :  10   : &CH5:sign  :   &CH5:mag;
    fanin_def   =   A   :   1  :  12   : &CH6:sign  :   &CH6:mag;
    fanin_def   =   A   :   1  :  14   : &CH7:sign  :   &CH7:mag;
    fanin_def   =   A   :   1  :  16   : &CH8:sign  :   &CH8:mag;
    fanin_def   =   B   :   1  :  18   : &CH1:sign  :   &CH1:mag;
    fanin_def   =   B   :   1  :  20   : &CH2:sign  :   &CH2:mag;
    fanin_def   =   B   :   1  :  22   : &CH3:sign  :   &CH3:mag;
    fanin_def   =   B   :   1  :  24   : &CH4:sign  :   &CH4:mag;
    fanin_def   =   B   :   1  :  26   : &CH5:sign  :   &CH5:mag;
    fanin_def   =   B   :   1  :  28   : &CH6:sign  :   &CH6:mag;
    fanin_def   =   B   :   1  :  30   : &CH7:sign  :   &CH7:mag;
    fanin_def   =   B   :   1  :  32   : &CH8:sign  :   &CH8:mag;
    fanin_def   =   C   :   1  :   1   : &CH1:sign  :   &CH1:mag;
    fanin_def   =   C   :   1  :   3   : &CH2:sign  :   &CH2:mag;
    fanin_def   =   C   :   1  :   5   : &CH3:sign  :   &CH3:mag;
    fanin_def   =   C   :   1  :   7   : &CH4:sign  :   &CH4:mag;
    fanin_def   =   C   :   1  :   9   : &CH5:sign  :   &CH5:mag;
    fanin_def   =   C   :   1  :  11   : &CH6:sign  :   &CH6:mag;
    fanin_def   =   C   :   1  :  13   : &CH7:sign  :   &CH7:mag;
    fanin_def   =   C   :   1  :  15   : &CH8:sign  :   &CH8:mag;
    fanin_def   =   D   :   1  :  17   : &CH1:sign  :   &CH1:mag;
    fanin_def   =   D   :   1  :  19   : &CH2:sign  :   &CH2:mag;
    fanin_def   =   D   :   1  :  21   : &CH3:sign  :   &CH3:mag;
    fanin_def   =   D   :   1  :  23   : &CH4:sign  :   &CH4:mag;
    fanin_def   =   D   :   1  :  25   : &CH5:sign  :   &CH5:mag;
    fanin_def   =   D   :   1  :  27   : &CH6:sign  :   &CH6:mag;
    fanin_def   =   D   :   1  :  29   : &CH7:sign  :   &CH7:mag;
    fanin_def   =   D   :   1  :  31   : &CH8:sign  :   &CH8:mag;
  *
  enddef;
  *
  def VLBA/XX-8-2/16;                *VLBA mode XX-8-2 recorded on 16 tracks
  *
  *                 sub-  trksID   sign/
  *                 pass            mag   hdstk  trk1
    fanout_def   =   A   :&CH1  :  sign  :  1  :  2         ; *1-to-1 fanout
    fanout_def   =   A   :&CH1  :   mag  :  1  :  4         ;
    fanout_def   =   A   :&CH2  :  sign  :  1  :  6         ;
    fanout_def   =   A   :&CH2  :   mag  :  1  :  8         ;
    fanout_def   =   A   :&CH3  :  sign  :  1  : 10         ;
    fanout_def   =   A   :&CH3  :   mag  :  1  : 12         ;
    fanout_def   =   A   :&CH4  :  sign  :  1  : 14         ;
    fanout_def   =   A   :&CH4  :   mag  :  1  : 16         ;
    fanout_def   =   A   :&CH5  :  sign  :  1  : 18         ;
    fanout_def   =   A   :&CH5  :   mag  :  1  : 20         ;
    fanout_def   =   A   :&CH6  :  sign  :  1  : 22         ;
    fanout_def   =   A   :&CH6  :   mag  :  1  : 24         ;
    fanout_def   =   A   :&CH7  :  sign  :  1  : 26         ;
    fanout_def   =   A   :&CH7  :   mag  :  1  : 28         ;
    fanout_def   =   A   :&CH8  :  sign  :  1  : 30         ;
    fanout_def   =   A   :&CH8  :   mag  :  1  : 32         ;
    fanout_def   =   B   :&CH1  :  sign  :  1  :  3         ;
    fanout_def   =   B   :&CH1  :   mag  :  1  :  5         ;
    fanout_def   =   B   :&CH2  :  sign  :  1  :  7         ;
    fanout_def   =   B   :&CH2  :   mag  :  1  :  9         ;
    fanout_def   =   B   :&CH3  :  sign  :  1  : 11         ;
    fanout_def   =   B   :&CH3  :   mag  :  1  : 13         ;
    fanout_def   =   B   :&CH4  :  sign  :  1  : 15         ;
    fanout_def   =   B   :&CH4  :   mag  :  1  : 17         ;
    fanout_def   =   B   :&CH5  :  sign  :  1  : 19         ;
    fanout_def   =   B   :&CH5  :   mag  :  1  : 21         ;
    fanout_def   =   B   :&CH6  :  sign  :  1  : 23         ;
    fanout_def   =   B   :&CH6  :   mag  :  1  : 25         ;
    fanout_def   =   B   :&CH7  :  sign  :  1  : 27         ;
    fanout_def   =   B   :&CH7  :   mag  :  1  : 29         ;
    fanout_def   =   B   :&CH8  :  sign  :  1  : 31         ;
    fanout_def   =   B   :&CH8  :   mag  :  1  : 33         ;
  *
  enddef;
    *
*
  def VLBA/XX-8-2/32;                 *VLBA mode XX-8-2 recorded on 32 tracks
  *
  *                sub- trksID   sign/
  *                pass           mag   hdstk  trk1  trk2
    fanout_def   =  A  :&CH1  :  sign  :  1  :  2  :  4   ; *1-to-2 fanout
    fanout_def   =  A  :&CH1  :   mag  :  1  :  6  :  8   ;
    fanout_def   =  A  :&CH2  :  sign  :  1  :  3  :  5   ;
    fanout_def   =  A  :&CH2  :   mag  :  1  :  7  :  9   ;
    fanout_def   =  A  :&CH3  :  sign  :  1  : 10  : 12   ;
    fanout_def   =  A  :&CH3  :   mag  :  1  : 14  : 16   ;
    fanout_def   =  A  :&CH4  :  sign  :  1  : 11  : 13   ;
    fanout_def   =  A  :&CH4  :   mag  :  1  : 15  : 17   ;
    fanout_def   =  A  :&CH5  :  sign  :  1  : 18  : 20   ;
    fanout_def   =  A  :&CH5  :   mag  :  1  : 22  : 24   ;
    fanout_def   =  A  :&CH6  :  sign  :  1  : 19  : 21   ;
    fanout_def   =  A  :&CH6  :   mag  :  1  : 23  : 25   ;
    fanout_def   =  A  :&CH7  :  sign  :  1  : 26  : 28   ;
    fanout_def   =  A  :&CH7  :   mag  :  1  : 30  : 32   ;
    fanout_def   =  A  :&CH8  :  sign  :  1  : 27  : 29   ;
    fanout_def   =  A  :&CH8  :   mag  :  1  : 31  : 33   ;
  *
  enddef;
  *
  def VLBA/XX-8-2/64;                 *VLBA mode XX-8-2 recorded on 64 tracks
  *
  *                sub- trksID   sign/
  *                pass           mag   hdstk  trk1  trk2  trk3  trk4
    fanout_def   =  A  :&CH1  :  sign  :  1  :  2  :  4  :  6  :  8  ; *1-to-4 fanout
    fanout_def   =  A  :&CH1  :   mag  :  1  : 10  : 12  : 14  : 16  ;
    fanout_def   =  A  :&CH2  :  sign  :  1  :  3  :  5  :  7  :  9  ;
    fanout_def   =  A  :&CH2  :   mag  :  1  : 11  : 13  : 15  : 17  ;
    fanout_def   =  A  :&CH3  :  sign  :  1  : 18  : 20  : 22  : 24  ;
    fanout_def   =  A  :&CH3  :   mag  :  1  : 26  : 28  : 30  : 32  ;
    fanout_def   =  A  :&CH4  :  sign  :  1  : 19  : 21  : 23  : 25  ;
    fanout_def   =  A  :&CH4  :   mag  :  1  : 27  : 29  : 31  : 33  ;
    fanout_def   =  A  :&CH5  :  sign  :  2  :  2  :  4  :  6  :  8  ;
    fanout_def   =  A  :&CH5  :   mag  :  2  : 10  : 12  : 14  : 16  ;
    fanout_def   =  A  :&CH6  :  sign  :  2  :  3  :  5  :  7  :  9  ;
    fanout_def   =  A  :&CH6  :   mag  :  2  : 11  : 13  : 15  : 17  ;
    fanout_def   =  A  :&CH7  :  sign  :  2  : 18  : 20  : 22  : 24  ;
    fanout_def   =  A  :&CH7  :   mag  :  2  : 26  : 28  : 30  : 32  ;
    fanout_def   =  A  :&CH8  :  sign  :  2  : 19  : 21  : 23  : 25  ;
    fanout_def   =  A  :&CH8  :   mag  :  2  : 27  : 29  : 31  : 33  ;
  enddef;
  *
  def VLBA/XX-4-2/8;                 *VLBA mode XX-4-2 recorded on 8 tracks
  *
  *                sub- trksID   sign/
  *                pass           mag   hdstk  trk1
    fanout_def   =  A  :&CH1  :  sign  :  1  :  2         ; *1-to-1 fanout
    fanout_def   =  A  :&CH1  :   mag  :  1  :  4         ;
    fanout_def   =  A  :&CH2  :  sign  :  1  :  6         ;
    fanout_def   =  A  :&CH2  :   mag  :  1  :  8         ;
    fanout_def   =  A  :&CH3  :  sign  :  1  : 10         ;
    fanout_def   =  A  :&CH3  :   mag  :  1  : 12         ;
    fanout_def   =  A  :&CH4  :  sign  :  1  : 14         ;
    fanout_def   =  A  :&CH4  :   mag  :  1  : 16         ;
    fanout_def   =  B  :&CH1  :  sign  :  1  : 18         ;
    fanout_def   =  B  :&CH1  :   mag  :  1  : 20         ;
    fanout_def   =  B  :&CH2  :  sign  :  1  : 22         ;
    fanout_def   =  B  :&CH2  :   mag  :  1  : 24         ;
    fanout_def   =  B  :&CH3  :  sign  :  1  : 26         ;
    fanout_def   =  B  :&CH3  :   mag  :  1  : 28         ;
    fanout_def   =  B  :&CH4  :  sign  :  1  : 30         ;
    fanout_def   =  B  :&CH4  :   mag  :  1  : 32         ;
    fanout_def   =  C  :&CH1  :  sign  :  1  :  3         ;
    fanout_def   =  C  :&CH1  :   mag  :  1  :  5         ;
    fanout_def   =  C  :&CH2  :  sign  :  1  :  7         ;
    fanout_def   =  C  :&CH2  :   mag  :  1  :  9         ;
    fanout_def   =  C  :&CH3  :  sign  :  1  : 11         ;
    fanout_def   =  C  :&CH3  :   mag  :  1  : 13         ;
    fanout_def   =  C  :&CH4  :  sign  :  1  : 15         ;
    fanout_def   =  C  :&CH4  :   mag  :  1  : 17         ;
    fanout_def   =  D  :&CH1  :  sign  :  1  : 19         ;
    fanout_def   =  D  :&CH1  :   mag  :  1  : 21         ;
    fanout_def   =  D  :&CH2  :  sign  :  1  : 23         ;
    fanout_def   =  D  :&CH2  :   mag  :  1  : 25         ;
    fanout_def   =  D  :&CH3  :  sign  :  1  : 27         ;
    fanout_def   =  D  :&CH3  :   mag  :  1  : 29         ;
    fanout_def   =  D  :&CH4  :  sign  :  1  : 31         ;
    fanout_def   =  D  :&CH4  :   mag  :  1  : 33         ;
  enddef;
  *
  *
  def MARK5A/XX-4-2/8;                 *VLBA mode XX-4-2 8-tk mode recorded on Mark5A
  *
  *                sub- trksID   sign/
  *                pass           mag   hdstk  trk1
    fanout_def   =     :&CH1  :  sign  :  1  :  2         ; *1-to-1 fanout
    fanout_def   =     :&CH1  :   mag  :  1  :  4         ;
    fanout_def   =     :&CH2  :  sign  :  1  :  6         ;
    fanout_def   =     :&CH2  :   mag  :  1  :  8         ;
    fanout_def   =     :&CH3  :  sign  :  1  : 10         ;
    fanout_def   =     :&CH3  :   mag  :  1  : 12         ;
    fanout_def   =     :&CH4  :  sign  :  1  : 14         ;
    fanout_def   =     :&CH4  :   mag  :  1  : 16         ;
  enddef;
  *
  def MARK3A/MODE_C;               *Mark3A mode C recorded on 14 tracks
  *                                *Note that track numbers are 'VLBA' track numbers
  *
  *                 sub-  trksID   sign/
  *                 pass            mag   hdstk  trk1
    fanout_def   =   A   :&CH1  :  sign  :  1  :  4         ; *1-to-1 fanout
    fanout_def   =   A   :&CH2  :  sign  :  1  :  6         ;
    fanout_def   =   A   :&CH3  :  sign  :  1  :  8         ;
    fanout_def   =   A   :&CH4  :  sign  :  1  : 10         ;
    fanout_def   =   A   :&CH5  :  sign  :  1  : 12         ;
    fanout_def   =   A   :&CH6  :  sign  :  1  : 14         ;
    fanout_def   =   A   :&CH7  :  sign  :  1  : 16         ;
    fanout_def   =   A   :&CH8  :  sign  :  1  : 18         ;
    fanout_def   =   A   :&CH9  :  sign  :  1  : 20         ;
    fanout_def   =   A   :&CH10 :  sign  :  1  : 22         ;
    fanout_def   =   A   :&CH11 :  sign  :  1  : 24         ;
    fanout_def   =   A   :&CH12 :  sign  :  1  : 26         ;
    fanout_def   =   A   :&CH13 :  sign  :  1  : 28         ;
    fanout_def   =   A   :&CH14 :  sign  :  1  : 30         ;
    fanout_def   =   B   :&CH1  :  sign  :  1  :  5         ;
    fanout_def   =   B   :&CH2  :  sign  :  1  :  7         ;
    fanout_def   =   B   :&CH3  :  sign  :  1  :  9         ;
    fanout_def   =   B   :&CH4  :  sign  :  1  : 11         ;
    fanout_def   =   B   :&CH5  :  sign  :  1  : 13         ;
    fanout_def   =   B   :&CH6  :  sign  :  1  : 15         ;
    fanout_def   =   B   :&CH7  :  sign  :  1  : 17         ;
    fanout_def   =   B   :&CH8  :  sign  :  1  : 19         ;
    fanout_def   =   B   :&CH9  :  sign  :  1  : 21         ;
    fanout_def   =   B   :&CH10 :  sign  :  1  : 23         ;
    fanout_def   =   B   :&CH11 :  sign  :  1  : 25         ;
    fanout_def   =   B   :&CH12 :  sign  :  1  : 27         ;
    fanout_def   =   B   :&CH13 :  sign  :  1  : 29         ;
    fanout_def   =   B   :&CH14 :  sign  :  1  : 31         ;
  enddef;
  *
  def VLBA_TRK_FORMAT;
    track_frame_format=VLBA;
  enddef;
*
  def MARK4_TRK_FORMAT;
    track_frame_format=Mark4;
  enddef;
  *
```

```
def MARK3A_TRK_FORMAT;
  track_frame_format=Mark3A;
enddef;
*
def DATA_MODULATION_ON;
  data_modulation = on;
enddef;
*
def XTRK_PARITY/8;
  *Note: The 'VLBA_frmtr_sys_trk=' parameter defines the data to be placed on one of the four available 'system'
  *      tracks (tracks 0,1,34,35) on the VLBA recording system.  The formatter allows several options as
  *      illustrated in the following examples.
  *
  *                 output            frst_trk #trks
  *                  trk              covered
  VLBA_frmtr_sys_trk = 0 : xtrk_parity :   2  :  8; *xtrk parity covering tracks  2-9  written to formatter track 0
  VLBA_frmtr_sys_trk = 1 : xtrk_parity :  10  :  8; *xtrk parity covering tracks 10-17 written to formatter track 1
  VLBA_frmtr_sys_trk =34 : xtrk_parity :  18  :  8; *xtrk parity covering tracks 18-25 written to formatter track 34
  VLBA_frmtr_sys_trk =35 : xtrk_parity :  26  :  8; *xtrk parity covering tracks 26-33 written to formatter track 35
enddef;
*
def FRMTR_TK7_TO_SYSTRK0;
  VLBA_frmtr_sys_trk = 0 : duplicate  :   7  ;   *duplicate (pre-barrel-rolled) formatter track 7 to system track 0
enddef;
*
def TRNSPRT_TK23_TO_SYSTRK33;
*                     output source
*                       trk   trk
  VLBA_trnsprt_sys_trk = 33 : 23 ;      *duplicate VLBA recorder track 23 to system track 33.
enddef;
*
def S2/128-4-2;
  S2_recording_mode = 32x4-2 ;
  S2_data_source = Mk4_formatter : &BBCa : &BBCb ;     *data from Mark IV formatter phase-cal output port
enddef;
```