```
data structures, crucial variables and file formats
Author:   Jonathan Jacobs
last mod: 11/29/96

lh/rh: holds LE/RE data. There are no null columns because these are LARGE files.
lv/rv: vertical
lt/rt: torsion

pos: created by pickdata/readbox. combined LE & RE data, in columnar format
vel: derivative of 'pos'.  created by pickdata/readbox (makes call to D2PT)
acc: derivative of 'vel'.  created by pickdata/readbox (makes call to D2PT)

max_len_l: largest LE array length.  created by getfile
max_len_r: largest RE array length
max_len:   largest of the above two

namelist:  list of all loaded files, separated by 1 space.
namearray: vertical list of all loaded file names (blank-padded
           to make each entry 13 characters long).

what_f_names: list of all files selected by PICKDATA, separated by a space.
what_f_array: blank-padded, vertical list of all PICKDATA-selected files.

which_eye:    tells whether pos(:,x) has LE or RE data

has_LH:       RH + LH columns wide.  Tells whether the corresponding
has_RH:       column in 'pos' has LH data or RH data.
has_LV:
has_RV:
has_LT:
has_RT:
              overlaps functionality of which_eye.

which_lh_col: has RE + LE columns.  Tells WHICH column of lh (or rh)
which_rh_col: is in column x of 'pos'.
which_lv_col:
which_rv_col:
which_lt_col:
which_rt_col:

%saccade control points
sacv_on_lh:   Holds saccade onset points for LE/RE.  Derived from vel trace.
sacv_on_rh:   Width is that of LE for sac_on_l, and of RE for sac_on_r.
              Use which_*h_col to synch with 'pos' array
              eg sac_on_l(:,which_lh_col(x)) belongs to pos(:,x)
sacp_on_lh:   Holds saccade onset points for LE/RE.  Derived from pos trace.
sacp_on_rh:   Width is that of LE for sac_on_l, and of RE for sac_on_r.
              Use which_*h_col to synch with 'pos' array
              eg sac_on_l(:,which_lh_col(x)) belongs to pos(:,x)
sacv_on_mat:  Created by pickdata/readbox.  contains the saccade
sacp_on_mat:  onset points for the selected data.

sacv_off_l:     analogous to sacv_on_l
sacp_off_l:     analogous to sacp_on_l
slow_peak_l:    analogous to sacp_on_l
max_v_pt_l:     analogous to sacv_on_l
cycle_beg_l:    analogous to sacp_on_l
cycle_end_l:    analogous to sacp_on_l

sacv_off_r:     analogous to sac_on_r
sacp_off_r:     analogous to sac_on_r
slow_peak_r:    analogous to sac_on_r
max_v_pt_r:     analogous to sacv_on_r
cycle_beg_r:    analogous to sac_on_r
```

```
cycle_end_r:    analogous to sac_on_r

sac_off_mat:    analogous to sac_on_mat
sacp_off_mat:   analogous to sac_on_mat
slow_peak_mat:  analogous to sac_on_mat
max_v_pt_mat:   analogous to sacv_on_mat
cycle_beg_mat:  analogous to sac_on_mat
cycle_end_mat:  analogous to sac_on_mat
```

Inside pickdata/readbox:

```
outer:    from 1 to total_num_files.  Used to walk through
          the possible candidates to be included in 'pos' et al
count:    how many sets of data are actually included.  this
          could be as great as 2*outer if all the loaded files
          contain binocular data.
```

Files:

Data:

ASCII data files are column-oriented numeric data. There can be no other information (like headers) in the file, or MATLAB will fail doing a "load".

RETRIEVE format files are binary, and are not human-readable. To decode one of these files you need to know the header structure (see 'readhdrD.m') and have access to a hex editor and hex<->decimal calculator.

Control points:

Saccade control point files have a "s" for the last letter of their extension. For example "LSH01_1.txt" will have a control points file named "LSH01_1.s" The file format is as follows:

```
There is a header line that contains:
         Type of saccade:     'B'raking or 'F'oveating
         Type of waveform:    'PP' or 'PC' (or whatever you want)
         Which eye/direction: 'lh', 'rh', 'lv', 'rv', 'lt', 'rt'
         Number of entries
There are eight columns per data channel:
         velocity-derived saccade onset
         position-derived saccade onset
         position-derived saccade offset
         velocity-derived saccade offset
         index of the maximum of the slow phase (pos-derived)
         index of the maximum of the saccade velocity (vel-derived)
         beginning of the saccade cycle (pos-derived)
         end of the saccade cycle (pos-derived)
```

A control points file can have as many waveform-saccade type combinations as you wish, per eye. ((As of 1/4/96, need a better way to choose which get chosen.) Will use GUI with checkboxes to allow user to include different types of saccade cycles))

Bias Adjust file:

These files are used to offset and scale the raw data. The file can contain modifications for either RTRV or ASCII data, taken with either IR or coil systems.

The header has four entries:

```
FILENAME    #channels    RecordingType    DataType
```

If the recording type is "IR" then the channel entries must have a zero-adjustment, a maximum calibration scale factor and a minimum calibration scale factor:

```
   chan1   zero_adjust   max_adjust   min_adjust
   chan2   zero_adjust   max_adjust   min_adjust
     .         .             .             .
     .         .            .             .
   chanN   zero_adjust   max_adjust   min_adjust
```

If the recording type is "coil" all that is necessary is a zero-adjustment.

```
   chan1   zero_adjust
   chan2   zero_adjust
     .         .
   chanN   zero_adjust
```

If the data type is "ASCII" then the offset/scaling must be followed by a sampling frequency, whereas for "RTRV" this is not needed, for this info is already stored in the RETRIEVE header.

```
   chan1   zero_adjust   max_adjust   min_adjust   samp_freq

          or

   chan1   zero_adjust   samp_freq
```

If you don't want to offset/scale your data (or don't yet know the proper offset and scale factors), simply enter "0 1 1" for the channel. Then you can use "os.m" to calculate the proper values. Once you have done this you can enter these values into "adjbias.txt".