DDBase Command Reference

This appendix consists of a **DDBase** command reference in the form of the **DDBase** syntax page (printed whenever **DDBase** is run with option `-h` or `--help`, or without any arguments), with each command followed by an explanation of the command's form, meaning and use.

Commands that include whitespace (for example file names or paths with embedded spaces) should be enclosed in double quotes. For example, suppose the user wishes to run **DDBase** with the input file `test[sp]input.file`, where [sp] stands for whitespace, and this input file contains the following command:

```
--Log “test[sp]log.file”
```

The user would enter the following at the command prompt:

```
DDBase “-ftest[sp]input.file”
```

This appendix assumes UNIX conventions for the operating system dependent details of command line processing. The only difference is the forward slash (/) is used under UNIX while under Windows the backslash (\) is used.

This documentation uses the word ‘station’ to mean site or receiver position. Input to **DDBase** must consist of (RINEX format) data from two or more different stations. Each station must be given a unique label (determined by the user) and an approximate position must be given for each. **DDBase** computes a relative position, meaning that the results (positions of the stations that are not ‘fixed’ (see below)) are relative to the position(s) of the station(s) that are ‘fixed.’

The word ‘baseline’ refers to the distance or the separation vector between two stations. The positioning algorithm and the operation of **DDBase** are very different for short (less than a few kilometers) and long baselines. **DDBase** currently (ver 4.2) is not tested at long baselines.

```
DDBase, ARL:UT DD phase estimation processor, Ver 4.2 2/26/07, Run
2007/06/27 08:00:22
Usage: DDBase42 [OPTION] ...
 Prgm DDBase will read RINEX obs data from any number of files and process
them
 in a double-differenced carrier phase estimation algorithm to produce
precise
 estimates of relative positions. Input is on the command line, or of the
same
 format in a file (see -f<file> below). DDBase is built on the GPS Toolkit
(GPSTk).
 NB. Input option --DT <data_interval_(seconds)> is required.
 NB. Stations are defined, and many inputs for each are identified, by a
label
  (called station label or id below), which is case sensitive and must be
used
  consistently throughout. It cannot be 'X','Y' or 'Z' nor contain '-' or
'_';
  four characters work best.
 NB. There must be at least two stations defined, with observation file(s)
  provided for each, and at least one station must be fixed.

 Options may be given in an input file (see -f<file>); the '#' character
marks
```

```
 a comment, to EOL. All input options are shown below, followed by a
 description, and the default value, if there is one, in ().
```

```
Optional arguments:
 [-f|--file] <file>       Name of file containing more options (#- EOL:
comment)
```

This command allows the user to put any number of other commands in a file rather than including them on the command line itself. Note that there is no whitespace between the '`-f`' and the file name. The format of the commands is just as on the command line, except that comments are allowed – on each line, everything from (and including) the symbol '#' and the end of line is ignored. Note that commands that include whitespace must be enclosed in double quotes.

```
 --Log <file>          Name of output log file (ddbase.log)
```

As **DDBase** runs it writes results and other useful intermediate information to a log file; this command gives the name of that log file. The default file is `ddbase.log` in the current directory. To put the log file in another directory, just include the path as part of the file name, for example `--Log logs/run17.log` makes the log file the file '`run17.log`' in the `./logs` directory. The verbose and debug options (see below) determine how much output is sent to this file. There is also output directed to the screen; this is only a subset of the output sent to the log file.

```
# Observations:
 --ObsPath <path>      Path for input obs file(s) (.)
```

This command gives **DDBase** the path at which to look for the RINEX observation files (see the next command). There can be only one path specified. Note that the path of the file could also be included in the file name itself (in the next command) to achieve the same effect as this command.

```
 --ObsFile <name,id>   Rinex observation file name(s),
                        followed by a station label.
```

(REQUIRED) This command does two things: first it gives **DDBase** the names of the RINEX observation files at each of the stations to be processed, and second, it defines labels which uniquely identify the station, both in other commands and in the results presented in the log file. There may be more than one RINEX file per station, in any order; however **DDBase** will assume that the net content of the files for each station is in a single continuous time block.

Station labels are case-sensitive and must be consistently used for all of the stations, both fixed and not fixed, which are given to **DDBase** to be processed. They must NOT contain whitespace, hyphen (-) or underscore (_), and must not be "X," "Y," or "Z." The log file output is designed to "look best" with 4-character labels.

There must be at least two stations in the input to **DDBase**; at least one must be fixed and at least one must not be fixed.

```
# Ephemeris and Earth orientation:
```

```
--NavPath <dir>       Path of navigation file(s) (.)
```

This command gives **DDBase** the path at which to look for the navigation (ephemeris) files (see the next command). There can be only one path specified. Note that the path of the file could also be included in the file name itself (in the next command) to achieve the same effect as this command.

```
 --NavFile <file>      Navigation (Rinex Nav OR SP3) file(s)
```

(REQUIRED) This command gives **DDBase** the name of the navigation or ephemeris file. There may be more than one. The files may be either RINEX navigation (broadcast ephemeris) files, or SP3-format (precise ephemeris) files. SP3-format files are available from NGA on their web site, or from IGS web sites.

A note about processing near day boundaries: if **DDBase** is processing data collected at the beginning or end of the day, it will require the satellite ephemerides for slightly more than just that day. Because SP3-format ephemeris files usually cover precisely one day, this implies that input of the SP3 file for the adjoining day will be required as well. In other words, even though the data do not cross the day boundary, in order to process right up to that boundary, ephemeris from the adjoining day will be required as well. A similar thing often happens with the broadcast ephemeris, particularly at the beginning of the day, because often the ephemeris records are put into one file per day (e.g., IGS navigation data) based on their time of reception. (This also applies to the EOP files – see below.)

```
 --EOPPath <dir>       Path of earth orientation file(s)
```

The path to the Earth Orientation Parameter (EOP) files (cf. the following command) is given by this command. There can be only one path specified. Note that the path of the file could also be included in the file name itself (in the next command) to achieve the same effect as this command.

```
 --EOPFile <file>      Earth orientation parameter (EOPP or IERS)
                        file(s).  If no EOP file is given, DDBase
                        will search for the IERS file
                        ‘finals.daily’ in the current directory.
```

(REQUIRED) **DDBase** applies a correction for the (changing) orientation of Earth in inertial space. It obtains this correction from small files that are obtained from a variety of sources called Earth Orientation Parameter (EOP) files. This command gives the name of the Earth Orientation Parameter (EOP) file. This command may be repeated with more than one file.

EOP files, one per week, are obtainable from the NGA web site **Error! Reference source not found.**, and are named EOPPyw.TXT, where ‘y’ is the single last digit of the year, and ‘w’ is the 2-digit week-of-year (this is the old convention) or the 4-digit GPS week number (the new convention) of interest. Also, the IERS produces a much larger file which contains EOPs for long periods, including predicted EOPs for future times, on their website **Error! Reference source not found.**. The file ‘finals.daily’ (IAU1980) may be downloaded and used by **DDBase** in case EOPP files cannot be obtained. **DDBase** will look for ‘finals.daily’ in the current directory when this option is not found in the input.

When processing near the day boundary, EOP files from the adjoining day may be needed – cf. the note under `--NavFile`.

```
# Station configuration [--Pos..  (1 only) MUST be given for
 each site]:
--PosXYZ <X,Y,Z,id>   Station position in ECEF coordinates (m),
                        followed by a label identifying the
                        station.
--PosLLH <La,Lo,H,id> Station position in geodetic coordinates:
                        Latitude(deg),Longitude(E,deg),Height(m),
                        label
--PosPRS <id>         Let position of station labelled <id> be
                        set to the computed average pseudorange
                        solution for that site.
```

(REQUIRED) Every station to be processed by **DDBase** MUST be given a position. Stations that are ‘fixed’ (see below) by **DDBase** should be well known ‘reference’ sites, with accurately known position, since the **DDBase** results are relative to these known stations. This is especially true when more than one ‘fixed’ station is given. The stations that are not ‘fixed’ in **DDBase** are probably known only approximately or even unknown – that’s why you need **DDBase**. However, the quality of these positions, for both fixed and not-fixed stations, will directly influence the difficulty the problem that **DDBase** must solve and may to some extent affect the accuracy of the results, particularly for long baselines. Thus it is important that high quality positions be provided in the input.

The above three command (`--Pos…`) allow the user to enter this position. Each station must have exactly ONE of the `--Pos…` commands in the input. The `PosXYZ` command is followed by four comma-separated fields (no whitespace) containing the Earth-centered Earth-fixed (ECEF) WGS84 position coordinates of the station in meters; the fourth field is the station label (consistent with other input, e.g. `ObsFile`). Similarly, the `PosLLH` command is followed by geodetic latitude (in degrees North), longitude (degrees East), and height above the ellipsoid (meters) of the station, followed by the station label. The third command, `--PosPRS`, is followed only by the station label; this tells **DDBase** to use the position it computes from the pseudorange as the position for this station. Note that this position is relatively inaccurate (meter-level errors) and therefore not to be preferred unless there is no other known position available.

There must be at least two stations in the input to **DDBase**; at least one must be fixed and at least one must not be fixed.

```
 --TropModel <trop,id> Use trop model <trop> for station <id>,
                        choices are: ‘Zero’,’Black’,’NewB’,
                        ‘ModHop’,’ModHopH’,’Saas’ (Saas)
                        [cf. GPSTk]
```

A model of the troposphere (lower atmosphere) is used by **DDBase** to correct for the delay introduced into the GPS signals; this is a standard GPS processing technique. A tropospheric model must be assigned to each station. This command includes a short abbreviation of the model name and the station label, separated by a comma. There are several different models to choose from; these are part of the GPS toolkit. The ‘Zero’ model means all zeros – this should not be used in practice. The others will probably have nearly identical effect on the results of **DDBase**, and the choice is not important. The default is the Saastamoinen model (Saas).

In practice it is probably best to choose the same model for each station, unless the stations are very far apart and it is known that one model is particularly better. Beware of extremes in height – very high altitude stations and/or situations in which the stations given to **DDBase** vary significantly in height should probably use a tropospheric model that accounts for height explicitly – Saas or ModHopH. In these extreme cases weather information is valuable and should be given (see the next command) if it is available.

```
 --Weather <T,P,H,id>  Weather parameters: Temperature(degC),
                        Pressure(mbar), Humidity(%), followed by
                        a label identifying the station.
                        (20.0,1010.00,50.0)
```

This command allows the user to give **DDBase** standard weather information for use in the tropospheric model. The information consists of temperature (Celsius), pressure (millibars) and relative humidity (a percentage: 0-100). These three numbers are followed by the station label, all comma separated.

```
 --Fix <id>            Hold the station <id> fixed in estimation
                        (don't)
```

DDBase produces a relative position solution, meaning one (or more) stations are positioned relative to one (or more) others, NOT relative to the Earth or another 'absolute' coordinate frame. In **DDBase**, the stations for which this command is entered are considered to be the known stations, those without the `--Fix` command are considered unknown and a position for them is estimated. The 'id' here is the usual station label (cf. under `--ObsFile` above).

There must be at least two stations in the input to **DDBase**; at least one must be fixed and at least one must not be fixed.

```
# Configuration:
 --noEstimate          Quit before performing the estimation.
```

This command allows the user to cause **DDBase** to terminate before entering the final portion of the algorithm, the estimation. This will happen after the raw data processing, the pseudorange solution, the satellite timetable and editing of the raw double differences. This option is for the user that wants to view the results of the preprocessing without having to wait for the estimation to finish.

```
 --Freq <L1|L2|L3>     Process L1, L2 or L3(L1+L2) frequency data
```

This command determines the frequency of the data to be used by **DDBase**; either the literal `L1` or `L2` must follow the `–Freq`; of course the data for GPS frequencies L1 or L2 must be present in the input data files. The option for frequency L3, which is a combination of L1 and L2 which eliminates the ionospheric delay in the phase, is NOT implemented at this time. (Note that L3 processing implies that biases cannot be fixed; cf. the command `--FixBiases` below.)

```
 --nIter <n>           Maximum number of estimation iterations (5)
```

This command allows the user to limit the number of iterations taken by the estimation processor. It does NOT include the 'extra' iteration that will be performed as a result of the `--FixBiases` command – cf. below. Normally the user should not change this value from its default of 5.

```
 --Converge <cl>       Convergence limit on RSS change in state
                        (5.00e-08 m)
```

This command allows the user to set the limit which determines when convergence is achieved by the estimation processor. **DDBase** computes, on each iteration of the processor, the change in the 'state vector,' which consists of the coordinates of the un-fixed positions and the phase biases. The square root of the sum of the squares of these changes is then compared to the convergence limit. When the estimation has 'converged,' this sum will be approximately zero. Normally this should not be changed by the user from its default, which is 5e-8 meters.

```
 --FixBiases           Perform an extra, last iteration that
                        fixes the phase biases
```

In the double differenced carrier phase algorithm it is necessary to estimate the biases on the double differenced phases (which are in fact integers) as well as the coordinates of the un-fixed station positions. Normally **DDBase** will converge to the point where these quantities are well determined and the biases are very close to integer values. This option allows the user to tell **DDBase** to force the biases to their integer values in the last iteration of the estimation (this iteration is NOT included in the count used by `nIter` – see above). This should improve the position estimate by, in effect, removing the small remaining error in the biases. This option should probably be included in all short baseline processing, but for longer baselines the integer biases may not be determined very well and it may be that the 'float' or biases-not-fixed solution will be better.

```
# State model, a priori constraints:
--RZDnIntervals <n>   Number of (equal time) residual zenith
                        delay intervals (0)
                        (enter 0 to turn off estimation of RZD)
--RZDtimeconst <tau>  Time constant (hours) for multiple RZD
                        intervals (2.00)
--RZDsigma <sig>      A priori sigma (m) for residual zenith
                        delay (0.50)
```

DDBase includes the option of estimating a residual zenith tropospheric delay (RZD). This is a processing technique that is used for long baselines and in situations where the troposphere is large or changing or not well suited to the standard models, and is therefore having a particularly strong effect on the solution. The idea is to estimate a small tropospheric delay at each station in addition to the delay that is removed by applying the tropospheric model (cf. `--TropModel` above). This is a sophisticated technique and should only be applied by users familiar with it.

The technique estimates a single RZD value at each station and for each of `n` equal time intervals. These RZD values are correlated with each other, using a first order Markov, or random walk process, with time constant `tau` and *a priori* sigma `sig`. The effect of the correlation is to limit the values that the estimated RZD values can take on. This is done because there are two things known about the RZD: it is small and it does not change rapidly. The *a priori* sigma tells the estimator not to let the estimated RZD get much bigger than this, and the time constant tells it not to let the estimated RZD change very much over this time.

The above commands allow the user to input the values defining the correlation in an obvious way. The default number of intervals is zero, which means no RZD estimation is performed. The defaults for the other parameters are 2 hours and 0.5 meters.

```
 --Tight <ppm>         Tight a priori constraint, in units fraction of the
                        baseline (1.e-4)
 --Loose <ppm>         Loose a priori constraint, in units fraction of the
                        baseline (1.e-1)
```

The **DDBase** position estimator determines the value of the 'not-fixed' station coordinates from the double differenced phase data, but it does so with a mild constraint in place. This constraint assumes that the estimated position will not vary widely from the given starting position (the `--Pos`… position, cf. above). This is a standard technique that is used to prevent aliasing of large errors into the estimated position; it usually takes the value of the constraint to be proportional to the corresponding baseline length. A tighter constraint is applied when the biases are fixed; a looser one when the biases are allowed to float – hence the names 'Tight' and 'Loose.' This option inputs the ratio of the constraint to the length of the corresponding baseline component. A typical (and the default) value for the tight (final iterations with biases fixed) constraint is 1.e-4. A typical (and the default) value for the loose (iterations prior to fixing the biases) constraint is 1.e-1. At very short baselines these should probably be increased by a factor of 10.

```
# Time limits:
 --BeginTime <arg>     Start time: arg is 'GPSweek,sow' OR
                        'YYYY,MM,DD,HH,Min,Sec'
 --EndTime <arg>       End time: arg is 'GPSweek,sow' OR
                        'YYYY,MM,DD,HH,Min,Sec'
```

These commands allows the user to limit the data that is input into **DDBase** to lie within the given time limits, even if there is data outside these limits in the input data files (cf. `--ObsFile` above). The option is followed by comma-separated numbers, with no included whitespace: either two values giving GPS week and seconds of week, or six values giving year (4-digit), month (1=January), day of month, hour of day (0-23), minutes of hour (0-59), and seconds of minute (0 <= sec < 60).

There are no defaults for these options; either or both may be omitted; in those cases the time limits are determined by whatever data are in the input RINEX observation files.

```
# Satellite time table:
 --TimeTable <file>    Time table file name (if this option does
                        not appear a time table will be computed
                        and output to log file)
 --Ref <sat>           Use <sat> as 'reference' in DDs; don't
                        use a timetable
```

As part of the raw data processing, **DDBase** forms a non-redundant set of double differences of the carrier phase. This means for each pair of stations and each pair of satellites, a double difference (DD) is computed. There is a unique phase bias for each such unique DD, which of course must be estimated later. In order to limit the number of biases, **DDBase** chooses one (or a small number) of ‘reference’ satellites to be the second satellite in each DD. This choice of reference satellite is limited by data availability. Also, it is well known that errors can be reduced in the estimation process by choosing a reference satellite at high elevation. Ideally **DDBase** would like to have a single high-elevation satellite available throughout the dataset that could be used as reference. In practice, **DDBase** applies a search algorithm to limit the number of different reference satellites and optimally choose the times of switching between references. This schedule of reference satellites and their times is called the satellite time table.

Normally the time table produced by **DDBase** can be seen in the log file. Here is an example, (the station labels are BSN and IQ):

```
# DDBase, ARL:UT DD phase estimation processor, Ver 3.5 3/1/06, Run
2006/03/20…
# REF site site sat week use_first use_last data_start data_end
REF BSN IQ G16 1364 197245.000 201846.000 197245.000 199530.000 40.0 54.2
2286
REF BSN IQ G27 1364 201846.000 207355.000 204162.000 207317.000 42.1 56.7
3156
```

The lines beginning with ‘#’ are treated as comments and ignored by **DDBase**. The table itself consists of one line for each reference satellite. Each line consists of the fixed string ‘REF’ followed by the station labels (cf. above) for the two sites in the double difference: in this case ‘BSN’ and ‘IQK.’ Then comes the satellite identifier, in this case ‘G16’ and ‘G27’ for satellite PRN 16 and 27, followed by the GPS week number. The next four fields give times in the form of GPS seconds of week; first the start and stop times at which to use this satellite as reference, then the start and stop times of data for this satellite. The three numbers at the end of each line have to do with the selection algorithm and can be ignored.

The above commands give the user the option of telling **DDBase** which time table to use. The –-TimeTable option is followed by the name of a file containing a time table in the same format as the output table in the **DDBase** log file (see the previous paragraph). The other option `--Ref` tells **DDBase** to use a single satellite as the reference. The `<sat>` field is the usual satellite identifier in the GPS toolkit, namely the PRN number (from the RINEX file) optionally preceded or followed by the character ‘G’ (for GPS); for example `--Ref G18`. Input of the time table may limit the amount of data that can be used by **DDBase**.

```
# Data editing:
 --MinElev <elev>      Ignore data below elevation <elev>
                        degrees, DDs only (10.00)
```

This command sets the limit or ‘mask’ on satellite elevation angles as seen at each station. While preprocessing the raw data, if the satellite elevation angle is below this limit (as determined by the input (`--Pos`…) position and the ephemeris data), then that measurement is rejected. The default value is 10 degrees. This limit is applied to the double differenced phases only, not to the pseudorange solution (see `--PRSMinElev` below).

This parameter should generally not be set much lower than about 10 degrees, because data quality is typically very poor for such low elevations, and **DDBase** will probably eliminate the data anyway based on this poor quality.

```
 --AntRotElev <elev>   Apply MinElev to antenna rotated in
                        elevation by <elev> deg.
```

```
--AntRotAz <az>       Apply MinElev to antenna rotated in
                       azimuth by <az> deg.
```

The above two ‘--AntRot…’ commands allow the user to define a special ‘mask’ in editing the raw double difference data. Rather than a simple elevation angle lower limit (--MinElev) this mask would correspond to a minimum elevation *after the antenna had been rotated*. The rotation is defined by the elevation and azimuth angles given in the commands. In other words, imagine first rotating the antenna by az degrees in azimuth and elev degrees in elevation (so that the horizontal North direction is rotated to (az,elev)), and *then* applying the MinElev lower limit mask. Double difference data that is outside this mask is rejected. The presence of either or both options will produce the rotated mask; the default values of both are zero degrees. This feature is specialized for use in studying antenna phase centers.

```
--MaxGap              Maximum acceptable gap in data [number of
                       --DT intervals] (10) [Used in raw data
                       editing and synchronization]
```

DDBase edits the raw phase data whenever it finds a gap in the data that is larger than some limit. This command sets that limit – the maximum number of consecutive data epochs without data that will not cause **DDBase** to ‘break’ the data into two different segments. (A break in the phase data implies a new phase bias, which will have to be estimated.) The default value is 10 points or 10*DT seconds.

```
--MinDDSeg            Minimum acceptable length of DD data
                       segment (50)
```

This is another command related to editing the double differences. Each continuous segment of double differenced phase has an unknown phase bias, which must be estimated by **DDBase**. A very small segment of data does not contribute enough useful information (data) to justify the expense of estimating another bias. Thus when **DDBase** finds a very small segment of data, it will simply delete it. This command allows the user to set the limit on the minimum number of points in an acceptable segment. The default is 50 points.

```
--PhaseBiasReset      Limit on pt-to-pt change in phase without
                       reset, in cycles (10)
```

As **DDBase** processes the raw phase data, it looks at the point-to-point change in the phase. A very large value of this change indicates that the bias on the receiver has changed or ‘slipped’ and therefore **DDBase** needs to separate the phase data on the two sides of the slip into two different segments. The limit on the point-to-point change in phase is set by this command. The default value is 10 cycles.

```
--XSat <sat>          Exclude this satellite ()
```

This command allows the user the exclude a satellite entirely from processing. This may be useful if it is known that there is a problem with a particular satellite. The <sat> field is the usual satellite identifier in the GPS toolkit, namely the PRN number (from the RINEX file) optionally preceded or followed by the character ‘G’ (for GPS); for example --XSat G18. This command may be repeated any number of times with different satellites.

```
 --DT <t>               Data time interval in sec [will also
                          decimate input data]
```

(REQUIRED) This command tells **DDBase** the nominal spacing in time of the input RINEX observation data. Thus if the RINEX file contains data at every 10 seconds (ignoring gaps in the data), then the user enters `--DT 10`. This command may also be used to decimate the data to a larger data interval (some multiple of the true data interval). Thus if the user has 10 second data but wishes to have **DDBase** process only 30 second data, entering `--DT 30` will accomplish this. Note that when decimating the data the entered interval MUST be a simple multiple of the true data interval.

This command is required and there is no default value; **DDBase** will abort if this command is not found.

```
# Pseudorange solution (PRS) configuration:
--PRSniter <n>        PRS: Limit on number of iterations (10)
--PRSconverge <cl>    PRS: Convergence limit (m) (1.00e-09)
--PRSrmsLimit <rms>   PRS: RMS residual limit (m) (6.50)
--PRSalgebra          PRS: Use algebraic algorithm (don't)
--PRSMinElev <elev>   PRS: Reject data below elevation <elev>
                        degrees (10.00)
```

The set of commands that begin with `--PRS` allow the user to configure the pseudorange solution computation. That is, these commands affect only the pseudorange computation (PRSolution in GPSTk), not any other part of **DDBase**'s work; however note that this computation includes a RAIM (receiver autonomous integrity monitoring) algorithm and this may cause some data to be rejected. For specific details on these quantities, refer to the GPSTk documentation. Generally the elevation limit should be the same as that used for the double differenced phase (`--MinElev` above); note that the default values are the same.

```
# Output files:
--RAWFileOut <file>   Filename for output of raw data ()
--PRSFileOut <file>   Filename for output of pseudorange
                        solution ()
--CLKFileOut <file>   Filename for output of Rx clock bias and
                        model ()
--RDDFileOut <file>   Filename for output of raw DD data ()
--DDDFileOut <file>   Filename for output of (edited) DD data ()
--TDDFileOut <file>   Filename for output of triple difference
                        data ()
--DDRFileOut <file>   Filename for output of DD post-fit
                        residuals ()
```

These commands allow the user to tell **DDBase** to output the data, at various stages, to files for further study. They are all of the same form, a command followed by the name of the file to which the data should be written. If the file exists, it will be overwritten. Each of these files includes one or more header rows describing the file format and contents.

```
# Output misc:
 --BaseOut <id-id,x,y,z> Baseline to output; <id>s are station
```

```
                          labels, '-' is required, <x,y,z> are
                          optional baseline coordinates.
```

From the list of fixed and non-fixed stations, **DDBase** decides which baselines (station pairs) to actually use in the estimation process; these baselines are printed near the top of the log file as "Computed baselines". At the end of its run **DDBase** will write results on the final estimated positions of the non-fixed stations to the log file. This command allows the user to tell **DDBase** to also output information on specific baselines (not necessarily the computed baselines) as well. This is helpful if the user has known coordinates for some of the baselines with which he wants to make comparisons. The `--BaseOut` command is followed by two station labels, separated by a hyphen or minus sign, without whitespace, and optionally followed by comma-separated 'known' baseline coordinates (ECEF X,Y,Z, in meters). This will cause **DDBase** to output the estimated baseline coordinates for the indicated pair of stations, plus offsets from the known baseline coordinates if they are present in the input.

For example, if the input contains

```
--BaseOut Ant1-Aref,25.10517,-11.25633,-13.05277
```

then the output includes, besides the estimated positions, the final baseline and offset from the known values:

```
Ant1: Estimated Position   -740493.269936  -5457024.220078
3207268.687637
Ant1: Estimated   Sigmas              0.000012              0.000035
0.000019
Aref:     Fixed Position      -740518.375079  -5457012.963051
3207281.739822
Final Baseline Ant1-Aref      25.105142       -11.257027       -13.052185
30.452395
Final  Offset  Ant1-Aref        -0.000033         -0.000702
0.000588
```

```
 --validate            Read input and validate it, then quit.
```

This command allows the user to have **DDBase** read the input and determine if it is valid and adequate to proceed, but then stop. **DDBase** will echo the input to the log file and include a statement indicating whether it is valid.

```
 --verbose             (also -v) print extended output info.
```

This command causes **DDBase** to include more information in the log file than otherwise.

```
 --debug               (also -d) print very extended output info
                        (for developers).
```

This command causes **DDBase** to include very much more information in the log file. Normally it is used by the developers only.

```
--help                (also -h) print this help message and quit
```

When **DDBase** encounters this command, it finishes reading the input, but then prints the syntax page and stops.