Parameter Descriptions for LocOO3D Version 1.11

Setting Parameters:

The parameters required by LocOO3D are preset to default values as the application is started. These defaults are given below in the parameter description section. Users may apply a different parameter value by using a property file (e.g., test.property). Only parameters whose values differ from their defaults need to be listed in the parameter file, since the defaults will be activated for any parameter not found in parameter file.

NOTES:

1) LocOO parameters are case sensitive.

2) All parameters in the parameter file must contain an '=' character, separating the parameter name from the parameter value (e.g. io_print_to_screen = true). White space around the '=' sign is optional (ignored).

3) Properties can be recursive. If a property value contains a string '<property:xyz>' then the phrase '<property:xyz>' is replaced with the value of property 'xyz'. For example, if the following records appear in the property file:

   testDirectory = \home\abc
   io_log_file = <property:testDirectory>\testdir

   then the actual value of property 'io_log_file' will be '\home\abc\testdir'.

4) If a property value contains the string '<env:xyz>' then the phrase '<env:xyz>' is replaced with the value returned by System.getenv(xxx).

Parallel Processing

parallelMode

<string> [Default = sequential] (sequential, concurrent)

When *parallelMode* is *sequential*, origins are located sequentially but the predictions calculated during location calculations are computed concurrently. When *parallelMode* is *concurrent*, the locations are computed in parallel and the predictions are computed sequentially.

maxProcessors

<int>. (Default = all available processors)

The number of processors LocOO3D can use.

batchSizeNdef

<int>. (Default = 1000)

The maximum number of defining phases to include in a batch of events to compute in parallel. See description of property *dbInputWhereClause* for more information.

Predictors

loc_predictor_type

<string> [Default = none] (lookup2d, bender)

String indicating list of predictors that are to be used. For example, if value is "lookup2d, bender(P, Pn)" then lookup2d will be used for all phases not specified later in the list, Bender will be used for phases P and Pn.

seismicBaseData

<string> [Default = none] ()

Path to the seismicBaseData directory. If this parameter is not specified then a copy of the seismicBaseData directory that is included in the locoo3d jar file will be used.

lookup2dModel

<string> [Default = ak135] (ak135)

Name of the 1D model that Lookup2D should use to calculate predictions of seismic observables.

lookup2dTableDirectory

<string> [Default = none] ()

Name of the directory where the travel time lookup tables reside. This directory will contain a separate file for each phase that will be supported. The file names can be names like 'PKP' or 'ak135.PKP'.

If *lookup2dTableDirectory* and *lookup2dEllipticityCorrectionsDirectory* are both specified, then they will dictate the locations of the table directory and ellipticity corrections directories, as advertised. If either of them is not specified then the locations of both directories will be deduced from property *seismicBaseData*.

lookup2dEllipticityCorrectionsDirectory

<string> [Default = none] ()

Path of the directory where ellipticity correction coefficients are located for use with the Lookup2D predictor.

If *lookup2dTableDirectory* and *lookup2dEllipticityCorrectionsDirectory* are both specified, then they will dictate the locations of the table directory and ellipticity corrections directories, as advertised. If either of them is not specified then the locations of both directories will be deduced from property *seismicBaseData*.

lookup2dUseEllipticityCorrections

<boolean> [Default = true] (true | false)

lookup2dUseElevationCorrections

<boolean> [Default = true] (true | false)

lookup2dSedimentaryVelocity

<double> [Default = 5.8 km/sec] ()

lookup2d TTModelUncertaintyScale

<2 doubles: scale and offset> [Default = null] ()

Travel time model uncertainty scale and offset. If one value is specified, it will be used to scale the travel time model uncertainty. If two values are specified, the second will be added to the travel time model uncertainty. In other words, ttModelUncertainty = ttModelUncertainty * scale + offset.

benderModel

<string> [Default = none] ()

Path to geoModel that Bender should use to calculate predictions of seismic observables.

benderUncertaintyType

<string> [Default = UncertaintyNAValue] (UncertaintyNAValue, DistanceDependent)

Type of travel time uncertainty desired. If UncertaintyNAValue is specified (default), then all requests for travel time uncertainty return the NA_VALUE (-999999.). If DistanceDependent is specified then distance dependent uncertainty is returned.

benderUncertaintyDirectory

<string> [Default = none] ()

Directory where distance dependent uncertainty values can be found for use with Bender predictions. Expecting to find subdirectories such as
<benderUncertaintyDirectory>/<attribute>/< <benderUncertaintyModel>
For example: if uncertainty information is in file
/index/SNL_tool_Root/seismicBaseData/tt/ak135 then specify
benderUncertaintyDirectory = /index/SNL_tool_Root/seismicBaseData
benderUncertaintyModel = ak135

benderUncertaintyModel

<string> [Default = none] ()

Subdirectory where distance dependent uncertainty values can be found for use with Bender predictions. Expecting to find subdirectories such as
<benderUncertaintyDirectory>/<attribute>/< <benderUncertaintyModel>

For example: if uncertainty information is in file
/index/SNL_tool_Root/seismicBaseData/tt/ak135 then specify
benderUncertaintyDirectory = /index/SNL_tool_Root/seismicBaseData
benderUncertaintyModel = ak135

use_tt_site_terms

<boolean> [Default = true]

If true then travel time site terms computed for each station during tomography are applied to computed values. The site terms are stored in the GeoTessModel specified with parameter *benderModel*.

use_tt_model_uncertainty

<boolean> [Default = true]

If true, travel time residuals and derivatives are weighted by the total uncertainty which consists of a combination of the model uncertainty and the pick uncertainty. If false, only the pick uncertainty is used.

use_az_model_uncertainty

<boolean> [Default = true]

If true, azimuth residuals and derivatives are weighted by the total uncertainty which consists of a combination of the model uncertainty and the pick uncertainty. If false, only the pick uncertainty is used.

use_sh_model_uncertainty

<boolean> [Default = true]

If true, slowness residuals and derivatives are weighted by the total uncertainty which consists of a combination of the model uncertainty and the pick uncertainty. If false, only the pick uncertainty is used.

Master Event Relative Relocation

If parameter *masterEventWhereClause* is specified, then LocOO3D will apply master event relocation corrections to all input origins that it relocates. A single materEvent origin along with associated assocs will be loaded into memory. For each masterEvent assoc, residuals of time, azimuth and slowness will be computed for defining observations using the Predictor defined in the properties file with property *loc_predictor_type*. These residuals will become masterEventCorrections which will be added to predictions of observations that have the same station-phase-type as the masterEventCorrection. Note that corrections to travel time, azimuth and slowness will be applied, so long as the corresponding master event assoc is time-defining, azimuth-defining and/or slowness-defining.

masterEventWhereClause

<string> [Default = null] (valid sql; eg. 'orid = 100')

If this parameter is present, then master event relocation option will be implemented. Must provide a valid sql where clause that will be executed against the *masterEventSchema* database schema. The where clause must return one and only one origin row. It the where clause returns anything other than one origin row, an exception is thrown.

masterEventAssocClause

<string> [Default = null] (valid sql; eg. 'phase = 'P'')

Optional parameter to limit the assoc rows returned with the master event origin row.

masterEventUseOnlyStationsWithCorrections

<boolean> [Default = false] (true or false)

If true then the only assocs that will be used to locate input origins are those that have valid master event corrections.

masterEventSchema

<string> [Default = dbInput] (schemaName, eg. 'dbMaster')

If this parameter is specified, then all the properties of a valid Schema must be supplied (see description of *dbInput* properties in the *DBIO Utility* section). For example, if property *masterEventSchema = dbMaster*, then properties *dbMasterUserName, dbMasterPassword, dbMasterOriginTable*, etc, will be recognized and all information about the *masterEvent* origin will be retrieved from database tables as specified.

If this parameter is not specified, then it is assumed that the information about the masterEvent origin, assoc, arrival and site will come from the same database tables as the input origins which are to be relocated (*dbInput* schema).

LibCorr3D

lookup2dPathCorrectionsType

<string> [Default = none] (libcorr)

Set the value to 'libcorr' to apply libcorr3d corrections. If this parameter is omitted, then corrections will not be applied.

lookup2dLibCorrPathCorrectionsRoot

<string> [Default = none]

The name of the directory where all the libcorr3D correction surfaces reside. This directory should contain a separate file for each correction surface.

lookup2dLibCorrPathCorrectionsRelativeGridPath

<string> [Default = ".]

The relative path from the directory where the correction surface files reside to the directory where the grid files reside.

lookup2dLibCorrInterpolatorType

<string> [Default = "linear"] (linear | natural_neighbor)

Type of horizontal interpolation to use.

lookup2dLibCorrPreloadModels

<boolean> [Default = false]

Whether all libcorr models should be loaded at startup or loaded on an 'as needed' basis.

lookup2dUsePathCorrectionsInDerivatives

<boolean> [Default = false]

Whether or not path corrections should be included in total values when computing derivatives of travel time with respect to source location.

General

lsq_max_iterations

<int> [Default = 100] (0 <= X <= 100000)

Maximum allowable number of iterations. If this number is set to 0, the LocOO3D simply computes the residuals and location uncertainty information at the initial location and outputs the results.

gen_initial_location_method

<string> [Default = data_file] (data_file | properties_file | internal)

Specifies the method for setting the initial event location.

Options:

data_file = If flat_file io is used, then take the initial location from the observation file summary line. For database io, use the location given in the origin table

properties_file = Use the values given in the *.properties file. See also gen_lat_init, gen_lon_init, gen_depth_init and gen_origin_time_init.

internal = For events with no defining azimuth observations, the initial location is set to the location of the station that observed the first arrival. If azimuth observations are available, then the initial location is based on the intersections of great circles computed from the azimuth observations. See Ballard [2002] for complete details.

gen_lat_init

<double> [Default = 0.0000] (-90.00 <= X <= 90.000)

Initial latitude estimate for the algorithm (degrees). If this parameter is specified, then the initial latitude is set to the value of this parameter, regardless of the setting of gen_initial_location_method.

For initial location to be specified in properties file, all four parameters gen_lat_init, gen_lon_init, gen_depth_init and gen_origin_time_init must be specified in the properties file.

gen_lon_init

<double> [Default = 0.0000] (-180.0 <= X <= 360.00)

Initial longitude estimate for the algorithm (degrees). If this parameter is specified, then the initial longitude is set to the value of this parameter, regardless of the setting of gen_initial_location_method.

For initial location to be specified in properties file, all four parameters gen_lat_init, gen_lon_init, gen_depth_init and gen_origin_time_init must be specified in the properties file.

gen_depth_init

<double, or "topography"> [Default = Globals.NA_Value] (-10000 <= X <= 10000.)

Initial depth estimate for the algorithm (km). If this parameter is specified, then the initial depth is set to the value of this parameter. If the value of this parameter is a string that starts with "topo" (case insensitive) then initial depth will be set to the depth of the topographic/bathymetric surface interpolated from the topography model at the latitude, longitude position of the initial location. If gen_depth_init is set to 'topography' then parameter topo_model must specify the topography model to use.

For initial location to be specified in properties file, all four parameters gen_lat_init, gen_lon_init, gen_depth_init and gen_origin_time_init must be specified in the properties file.

gen_origin_time_init

<string> [Default = 1970-01-01 00:00:00.000] (any reasonable date or epoch time)

Initial origin time estimate for the algorithm. The specified value can be either a date (yyyy-mm-dd hh:mm:ss.sss) or an epoch time (seconds since 1970-01-01 00:00:00.000). If this parameter is specified, then the initial origin time is set to the value of this parameter, regardless of the setting of gen_initial_location_method.

LocOO3D first tries to interpret the value of this parameter as a date in the format yyyy-mm-dd hh:mm:ss.sss. If that fails, it tries to convert the value to an integer (value may not include '.'). If successful, the value is interpreted as a jdate. If unsuccessful, LocOO3D attempts to convert the value to a number of type double (must include a '.'). If successful then the value is interpreted as an epoch time (seconds since 1970). If that also fails, LocOO3D throws an exception.

For initial location to be specified in properties file, all four parameters gen_lat_init, gen_lon_init, gen_depth_init and gen_origin_time_init must be specified in the properties file.

topo_model

<string> [Default = null] (none)

If the values of properties gen_depth_init or gen_min_depth start with 'topo' (case insensitive) then locoo will interrogate the specified GeoModel object to determine the local topography at the latitude, longitude position of the event. The value must specify the name of a file that contains a valid topography model in GeoModel format (these files usually have a .bin extension).

LocOO3D version 1.3 was delivered with 4 topography models named topo_000250.bin, topo_000500.bin, topo_001000.bin and topo_002000.bin. The numbers in the model names indicate the resolution of the model in millidegrees. The models have resolutions of 0.25°, 0.5°, 1°, 2°, respectively.

gen_fix_lat_lon

<bool> [Default = false]

Hold (true) / don't hold (false) the latitude and longitude fixed during the solution algorithm. If held fixed, then the initial values specified by gen_lat_init and gen_lon_init are used. See also gen_initial_location_method, gen_lat_init, and gen_lon_init.

gen_fix_depth

<bool> [Default = false]

Hold (true) / don't hold (false) the depth fixed during the solution algorithm. If held fixed, then the initial value specified by gen_depth_init is used. See also gen_initial_location_method, gen_depth_init.

gen_fix_origin_time

<bool> [Default = false]

Hold (true) / don't hold (false) the origin time fixed during the solution algorithm. If held fixed, then the initial value specified by gen_origin_time_init is used. See also gen_initial_location_method, gen_origin_time_init.

gen_allow_big_residuals

<bool> [Default = true]

Allow (true) / don't allow (false) observations that result in 'big' residual values.

Here is how this works: The event is first located using all defining observations. If any observations have weighted residuals greater than the value specified with gen_big_residual_threshold (default value is 3), then some subset of those observations are set to non-defining and the event is relocated. This process continues until either

there are no observations whose weighted residuals exceed the threshold or N = M where N is the number of defining observations and M is the number of degrees of freedom in the problem (4 for free depth solutions, 3 for fixed depth, etc). If there are observations with large residuals, then the subset of them that is set to non-defining is determined as follows. The minimum number that will be set to non-defining is 1. The maximum number that can be set to non-defining is (N-M) * gen_big_residual_max_fraction. gen_big_residual_max_fraction defaults to 0.1.

See gen_big_residual_threshold and _big_residual_max_fraction.

gen_big_residual_threshold

<double> [Default = 3.0000] (0.0000 <= X <= 100000)

Threshold weighted residual value above which observations are flagged as 'big'. See gen_allow_big_residuals.

gen_big_residual_max_fraction

<double> [Default = 0.10] (0.0 <= X <= 1.0)

A constraint on the maximum number of observations that can be set to non-defining when gen_allow_big_residuals is false and there are observations with large residuals.

gen_max_depth

<double> [Default = 700.00] (-1e6 <= X <= 1e6.)

Maximum depth constraint (km).

gen_min_depth

<double, or "topography"> [Default = -999999.] (-10000 <= X <= 10000.)

Minimum depth constraint (km). If the value of this parameter is a string that starts with "topo" (case insensitive) then the depth of the event location is constrained to be no less than the depth of the local surface topography/bathymetry.

gen_defining_phases

<string> [Default = all]

The subset of defining phases to use for the location algorithm. This overrides the assoc table values. Setting the value to 'all' is the default behavior and causes all previously indicated defining phases to be used. A comma-delimited list overrides this behavior and down-selects from the set of defining phases. All observations with phases that are not included in the list are set to non-defining.

gen_defining_stations

<string> [Default = all]

The subset of defining stations to use for the location algorithm. This overrides the assoc table values. Setting the value to 'all' is the default behavior and causes all previously indicated defining stations to be used. A comma-delimited list of station

names overrides this behavior and down-selects from the set of defining stations. All observations from stations that are not included in the list are set to non-defining.

gen_defining_attributes

<string> [Default = all] (t, a, s)

The subset of defining attributes (travel time, azimuth, slowness) to use for the location algorithm. This overrides the data file or assoc table values. Setting the value to 'all' is the default behavior and causes all previously indicated defining attributes to be used. A comma-delimited list overrides this behavior and down-selects from the set of defining stations. All observations with attributes that are not included in the list are set to non-defining.

Any word starting with the letters t or T is interpreted to be travel time, a or A is interpreted to be azimuth, and s or S is interpreted to be slowness.

gen_defining_observations_filter

<string> [Default = none]

A set of filters that can be used to toggle individual observations from defining to non-defining and vice versa. Each filter is of the form: ±ORID/STATION/PHASE/ATTRIBUTE. The set of filters consists of a number of individual filters, separated by commas. Each observation starts out as either defining or non-defining (after application of parameters gen_defining_stations, gen_defining_phases and gen_defining_attributes). Then the observation is subjected to each filter, in order. If the observation matches the filter, then the observation is made defining if the first character of the filter is '+' or non-defining if the first character of the filter is '-'. Each component of a filter can be the '*' character. Every observation matches the component of a filter that is the '*' character.

For example, the filter +100/ILAR/P/TT will guarantee that for the origin with orid 100, the P travel time observation from station ILAR will be defining. The filter
-*/ILAR/*/*, +*/ILAR/P/t
will first make all observations from station ILAR non-defining, then turn back on just the P travel time observations from station ILAR.

gen_error_ellipse_type

<string> [Default = coverage] (coverage|confidence|mixed)

Type of error ellipse desired:

coverage	Dimensions of error ellipse depend only on apriori information (K=-1, interpreted as infinity).
confidence	Dimensions of error ellipse depend only on a posteriori information (K=0).
mixed	Dimensions of error ellipse depend on both a priori and aposteriori information. The Jordan-Sverdrup K parameter is set to the value of this parameter.

gen_jordan_sverdrup_K

<int> [Default = -1] (-1 <= X <= 999999)

Jordan-Sverdrup K parameter for 'mixed' type confidence ellipses. -1 is interpreted as infinity. This parameter is ignored unless gen_ellipse_type == mixed.

gen_apriori_standard_error

<double> [Default = 1.0000] (0.0000 <= X <= 1000.0)

The a priori standard error scale factor. It represents an estimate of the ratio between the true and actual data standard errors. This parameter only applies when K > 0.

gen_confidence_level

<double> [Default = 0.9500] (0.0000 <= X <= 1.0000)

Uncertainty confidence level desired.

allowCorePhaseRenamingP

<boolean> [Default = false]

If corePhaseRenaming is true then this is the distance in degrees beyond which corePhaseRenaming will take place.

corePhaseRenamingThresholdDistanceP

<double> [Default = 110.] (0.0000 <= X <= 180.0000)

If *allowCorePhaseRenamingP* is true then this is the distance in degrees beyond which corePhaseRenaming will take place.

useSimplex

<boolean> [Default = false]

If true, then after computing the best fit location in the standard manner, the Simplex algorithm is applied to the previous best fit solution. If the simplex finds a better solution (lower rms residuals) then the simplex location is retained; otherwise, the standard solution is retained. The advantage of the simplex algorithm is that it does not require derivatives of predictions with respect to source position. The disadvantage is that it can be very expensive computationally (order of magnitude longer to execute is not uncommon).

Correlated Observation Parameters

gen_correlation_matrix_method

<string> [Default = uncorrelated] (uncorrelated | file | function)

Specifies how correlation coefficients are to be specified. If 'file', then correlation coefficients between pairs of observations are read in from a file, with the file name being specified with the gen_correlation_matrix_file parameter. If value = 'function', then function parameters must be specified using the gen_correlation_scale parameter. Regardless of how the correlation coefficients are entered, they are used to populate an N x N matrix where N is the number of observations. The correlation matrix will have ones on the diagonal and user specified values between -1 and 1 in the off-diagonal elements. The diagonal elements of this matrix will be multiplied by the square of the total uncertainty of each observation (combined model and pick error). The off-diagonal elements are multiplied by the product of the model errors for the two observations (pick error is not included for off-diagonal elements).

gen_correlation_matrix_file

<string> [Default =] (any valid file path+name)

Specifies the name of the file from which correlation coefficients are to be read. The file consists of an arbitrary number of lines, each of which defines the correlation coefficient between two observations. Each line must contain the following information:

Station_1/phase_1/type_1 station_2/phase_2/type_2 corr_coeff

Where type_1 and type_2 are the observation types (TT, AZ or SH for travel time, azimuth and slowness), and corr_coeff is the correlation coefficient (-1 <= corr_coeff <= 1). For example,

ARCES/P/TT FINES/P/TT 0.5

would specify a correlation coefficient of 0.5 between all P wave travel time observations from ARCES and FINES. Correlation coefficients between pairs of observations which are not specified in the file are set to zero.

gen_correlation_scale

<double> [Default = 10 degrees] (> 0 degrees)

Specifies the correlation scale length for calculating the correlation coefficients between pairs of observations (degrees).

If gen_correlation_matrix_method is 'function' then the correlation coefficients between pairs of stations are computed from

$$c = \exp(-(\Delta / scale)^2)$$

where c is the correlation coefficient, Δ is the separation of the two stations where the observations were made, in degrees, and *scale* is the scale length specified with the gen_correlation_scale parameter. The scale length is in degrees. The default is 10 degrees.

Levenberg-Marquardt Non-Linear Least Squares Solver

lsq_print_iteration_table

<bool> [Default = true]

If false, iteration tables are not sent to general output.

lsq_convergence_n

<int> [Default = 2] (1 <= X <= 100000)

Number of consecutive times convergence criterion must be satisfied before convergence is declared.

lsq_applied_damping_multiplier

<double> [Default = 10.000] (1.0 <= X <= 1e6)

If the initial applied damping does not reduce the sum squared weighted residuals to a level below that observed in the previous iteration, keep multiplying the applied damping by this factor until it does, or until the damping is so large that the solution stops moving (see lsq_damping_dkm_threshold).

lsq_convergence_criterion

<double> [Default = 0.0001] (0 <= X <= 1e6)

Threshold convergence criterion. At the conclusion of each iteration the ratio of the sum squared residual at the conclusion of the current iteration to the sum squared residual at the conclusion of the previous iteration is calculated. One is subtracted from the result and the absolute value evaluated. If the resulting quantity is less than the threshold convergence criterion, the convergence criterion is declared to have been achieved.

lsq_damping_dkm_threshold

<double> [Default = 0.0100] (0.0000 <= X <= 1e6)

During automatic damping, the applied damping will continue to increase until either the sum squared weighted residual is reduced to a level less than that observed in the previous iteration, or until the applied damping is so large that the solution stops moving. The quantity dkm is the amount, in km, that the solution will move during an iteration. When dkm becomes less than, lsq_damping_dkm_threshold, it can be concluded that the sum squared weighted residuals cannot be further reduced and convergence can be declared.

lsq_damping_factor

<double> [Default = -1.000] (-1.000 <= X <= 1e6)

Damping factor to be applied to singular values. For AUTOMATIC DAMPING: lsq_damping_factor = -1 [DEFAULT]. This factor controls application of the Levenberg-Marquardt algorithm which helps the locator converge if it is oscillating around the optimum location. Set to -1.0 to have the locator automatically adjust the damping factor as necessary. Set to 0.0 or positive values to override the default behavior.

lsq_initial_applied_damping

<double> [Default = 0.0001] (0.0000 <= X <= 1e6)

When AUTOMATIC DAMPING is being applied, this is the initial damping factor applied when an increase in the sum squared weighted residuals is observed.

lsq_singular_value_cutoff

<double> [Default = 1e-6] (0.0000 <= X <= 1e30)

Singular value cutoff. Any singular values that are less than this number times the maximum singular value will have their value set to infinity, with the result that the associated location parameter will not change from its initial value.

General Input/Output Parameters

io_verbosity

<int> [Default = 1] (0-4)

Verbosity level for progress information.
0 : no output, not even error messages. Error messages sent to output_error_file
1 : minimal output related mostly to property values, IO, and errors
2 : + basic information about the final locations
3 : + initial site and observation information
4 : + observation, prediction, and iteration tables

io_log_file

<string> [Default = null: no text output]

Full path to general output file. All header information, iteration status information, and final results are sent to the general output file. The output is in ascii text format.

io_print_to_screen

<bool> [Default = true]

Echo iteration progress and/or messages to the screen during the location calculation. This is the same information that is sent to the output text file.

io_error_file

<string> [Default = locoo_errors.txt]

Full path to general output error file. All error messages generated during locoo execution are sent to this file.

io_print_errors_to_screen

```
<bool>   [Default = io_verbosity > 0]

Write error messages to the screen.
```

io_observation_tables

```
<int>   [Default = 2]   (0 <= X <= 100000)

Maximum number of observation and prediction tables to output when io_verbosity >= 4.
     0 = No tables are output
     1 = Tables output only on final iteration
     2 = Tables output only on first and final iterations
     3 = Tables output only on first, second and final iterations
     4 = Tables output only on first three iterations + final iteration
     etc...

This parameter is ignored if io_verbosity < 4.
```

io_observation_sort_order

```
<string>   [Default = distance]    (distance | station_phase | weighted_residual)

Specifies the order in which observations are reported in the observation and prediction
tables in the text output file.
```

io_iteration_table

```
<boolean>   [Default = true]

Whether or not to print the iteration table when io_verbosity >= 4.
```

io_nondefining_residuals

```
<boolean>   [Default = true]

If true, then the residuals of all observations with valid observed values are computed
prior to writing results to the database, regardless of whether they are defining or
not.  Setting this parameter to false will not impact the final computed location.
```

Gridded Residuals

grid_output_file_name

```
<string>   [Default = none]

Name of file to receive the gridded residuals.  If this parameter is not specified then
gridded residuals are not generated and all the rest of the properties in this section
are ignored.
```

grid_output_file_format

```
<string>   [Default = tecplot]     ( tecplot )
```

The output file format. The only currently supported output format is tecplot. The format is pretty self-explanatory. Other formats may be supported in the future. Requests to support other formats will be considered.

When NZ = 1 there will be 4 values per record:
X (either longitude in degrees, or East in km),
Y (either latitude in degrees or North in km),
R root mean squared weighted residual, unitless
C confidence level (useful for plotting a 95% contour line).

When NZ > 1 there will be 5 values per record:
X (either longitude in degrees, or East in km),
Y (either latitude in degrees or North in km),
Z depth in km,
R root mean squared weighted residual, unitless
C confidence level, in % (useful for plotting a 95% contour line).

grid_origin_source

<string> [Default = epicenter] (epicenter | hypocenter | other)

The center of the grid. If anything other than epicenter or hypocenter is specified then the center of the grid is determined by properties grid_origin_lat, grid_origin_lon and grid_origin_depth

grid_origin_lat

<double> [Default = none]

Latitude in degrees of the center of the grid. Ignored if grid_origin_source equals either epicenter of hypocenter.

grid_origin_lon

<double> [Default = none]

Longitude in degrees of the center of the grid. Ignored if grid_origin_source equals either epicenter of hypocenter.

grid_origin_depth

<double> [Default = 0. km]

Depth in km of the center of the grid. Ignored if grid_origin_source equals either epicenter of hypocenter.

grid_map_units

<string> [Default = degrees] (degrees or km)

Map width and height will be interpreted with these units. Also controls the units of the output.

grid_map_width

<double> [Default = none]

Map will be this many units wide, in units specified by grid_map_units. The origin of the map will be in the center.

grid_map_height

<double> [Default = none]

Map will be this many units high, in units specified by grid_map_units. The origin of the map will be in the center.

grid_map_depth_range

<double> [Default = 0.]

Map will be this many units deep, in km. The origin of the map will be in the center.

grid_map_nwidth

<int> [Default = none]

Map will have this many nodes in the x direction.

grid_map_nheight

<int> [Default = none]

Map will have this many nodes in the y direction.

grid_map_ndepth

<int> [Default = 1]

Map will have this many nodes in the z direction.

DataLoader Utility

dataLoaderType

<string> [Default = None] (file | oracle)

Specifies whether to perform IO with text files or with an Oracle database. For descriptions of parameters related to file IO see section DataFileLoader Utility. For descriptions of parameters related to database IO see section DBIO Utility.

DataFileLoader Utility

dataLoaderFileInputOrigins

<string> [Default = None]

Name of file containing the input origins. Required.

dataLoaderFileInputAssocs

<string> [Default = None]

Name of file containing the input assocs. Required.

dataLoaderFileInputArrivals

<string> [Default = None]

Name of file containing the input arrivals. Required.

dataLoaderFileInputSites

<string> [Default = None]

Name of file containing the input sites. Required.

dataLoaderFileOrids

<string> [Default = None]

A list of orids to process. Optional. If not specified, then all origins in the file are processed.

dataLoaderFileOutputOrigins

<string> [Default = None]

Name of file to receive the output origins.

dataLoaderFileOutputOrigerrs

<string> [Default = None]

Name of file to receive the output origerrs.

dataLoaderFileOutputAssocs

<string> [Default = None]

Name of file to receive the output assocs.

dataLoaderFileOutputAzgaps

<string> [Default = None]

Name of file to receive the output azgaps.

dataLoaderFileOutputArrivals

<string> [Default = None]

Name of file to receive the output arrivals. The values of these arrivals are unchanged from the input arrivals but their format may have changed.

dataLoaderFileOutputSites

<string> [Default = None]

Name of file to receive the output sites. The values of these sites are unchanged from the input sites but their format may have changed.

dataLoaderFileInputTokenDelimiter

<string> [Default = tab]

Assembles the token delimiter for input from the specified value. The specified value is a space delimited string that uses "tab" for "\t", "comma" for ",", and "space" for " ". Any other character can be represented also. For example, if the desired delimiter is ",\t *" then the input string would be "comma tab space *". Using the long names for whitespace characters was necessary since the input tokenDelimiter is read from a properties file.

dataLoaderFileOutputTokenDelimiter

<string> [Default = same value as *dataLoaderFileTokenDelimiter*]

Assembles the token delimiter for output from the specified value. The specified value is a space delimited string that uses "tab" for "\t", "comma" for ",", and "space" for " ". Any other character can be represented also. For example, if the desired delimiter is ",\t *" then the input string would be "comma tab space *". Using the long names for whitespace characters was necessary since the input tokenDelimiter is read from a properties file.

dataLoaderFileOutputOriginColumns

<string> {Default = lat, lon, depth, time, orid, evid, jdate, nass, ndef, ndp, grn, srn, etype, depdp, dtype, mb, mbid, ms, msid, ml, mlid, algorithm, auth, commid}

The information that should be written to the output origin table.

dataLoaderFileOutputOrigerrColumns

<string> {Default = orid, sxx, syy, szz, stt, sxy, sxz, syz, stx, sty, stz, sdobs, smajax, sminax, strike, sdepth, stime, conf, commid }

The information that should be written to the output origerr table.

dataLoaderFileOutputAzgapColumns

<string> {Default = orid, azgap1, azgap2, sta, nsta, nsta30, nsta250}

The information that should be written to the output azgap table.

dataLoaderFileOutputAssocColumns

<string> {Default = arid, orid, sta, phase, belief, delta, seaz, esaz, timeres, timedef, azres, azdef, slores, slodef, emares, wgt, vmodel, commid }

The information that should be written to the output assoc table.

dataLoaderFileOutputArrivalColumns

<string> {Default = sta, time, arid, jdate, stassid, chanid, chan, iphase, stype, deltim, azimuth, delaz, slow, delslo, ema, rect, amp, per, logat, clip, fm, snr, qual, auth, commid }

The information that should be written to the output arrival table.

dataLoaderFileOutputSiteColumns

<string> {Default = sta, ondate, offdate, lat, lon, elev, staname, statype, refsta, dnorth, deast }

The information that should be written to the output site table.

DBIO Utility

dbInputUserName

<string> [Default = user's environment variable DBTOOLS_USERNAME]

Database input account username.

dbOutputUserName

<string> [Default = none]

Database output account username. If not specified, no output is written to the database.

dbInputPassword, dbOutputPassword

<string> [Default = user's environment variable DBTOOLS_PASSWORD]

Database input/output account passwords. If not specified in the property file, and the property DBTOOLS_PASSWORD is specified in the user's environment then the value from the environment is used.

dbInputInstance, dbOutputInstance

<string> [Default = user's environment variable DBTOOLS_INSTANCE]

Database instance for input/output.

dbInputDriver, dbOutputDriver

<string> [Default = user's environment variable DBTOOLS_DRIVER, or oracle.jdbc.driver.OracleDriver]

Database driver for input/output. Generally equals oracle.jdbc.driver.OracleDriver.

dbIntputTableTypes

<string> [Default =]

If the dbInputTableTypes parameter is specified, then the input table types specified with this parameter will default to the value of the dbInputTablePrefix parameter with the appropriate table type appended on the end. Currently recognized table types include: origin, assoc, arrival, site.

dbInputTablePrefix

<string> [Default none]

If this parameter is specified then the four input tables (dbInputOriginTable, dbInputAssocTable, dbInputArrivalTable, dbInputSiteTable) will default to the value of this parameter with the appropriate table type (ORIGIN, ASSOC, ARRIVAL, SITE) appended on the end. If any of the four tables are also explicitly specified, then the explicitly specified name has precedence.

dbInputOriginTable

<string> [Default not allowed]

Name of the input origin table. Specifying this parameter will override any default values set by other parameters.

dbInputAssocTable

<string> [Default not allowed]

Name of the input assoc table. Specifying this parameter will override any default values set by other parameters.

dbInputArrivalTable

<string> [Default not allowed]

Name of the input arrival table. Specifying this parameter will override any default values set by other parameters.

dbInputSiteTable

<string> [Default not allowed]

Name of the input site table. Specifying this parameter will override any default values set by other parameters.

dbInputWhereClause

<string> [No default value]

An orid query "where" clause that specifies the origins that should be processed as input for LocOO. This where clause is executed against the origin table.

LocOO3D executes the following sql statement in order to load data from the database:

1) First, it executes something similar to:

   select orid, ndef from <*dbInputOriginTable*> where <*dbInputWhereClause*> order by ndef desc

   With the output, it forms the orids into batches where each batch will have roughly the same number of defining phases and the batches tend to be in order increasing number of orids. The number of defining phases per batch can be specified with property *batchSizeNdef*, which defaults to 1000. This arrangement is most efficient when processing predictions in parallel. Orids that have more defining phases than *batchSizeNdef* will form their own batch. Once all of those origins have been formed into batches, other batches will have no more than *batchSizeNdef* defining phases in them. No batch will have more than 1000 orids in it due to oracle limitations.

2) For each batch of orids generated during step 1, LocOO executes 2 sql statements like

   select origin.orid, evid, lat, lon, depth, time from <*dbOriginTable*> where orid in (<*list of orids in batch*>)

   select orid, assoc.arid, -1, site.sta, site.lat, site.lon, site.elev, site.ondate, site.offdate, assoc.phase, arrival.time, arrival.deltim,

assoc.timedef, arrival.azimuth, arrival.delaz, assoc.azdef, arrival.slow, arrival.delslo, assoc.slodef from <*dbInputAssocTable*> assoc, <*dbInputArrivalTable*> arrival, <*dbInputSiteTable*> site where orid in (<*orids in batch*>) and assoc.arid=arrival.arid and arrival.sta=site.sta and arrival.jdate between site.ondate and site.offdate and <*dbInputAssocWhereClause*>

dbInputAssocClause

<string> [empty string]

An optional phrase that will be appended onto the end of the where clause that selects rows from the assoc, arrival and site tables. See dbInputWhereClause for details of how this is used.

For example, to include only data from stations within distance of 40 degrees from the origin, specify dbInputAssocClause = assoc.delta < 40. To limit the data to include only stations ABC and XYZ, specify dbInputAssocClause = site.sta in ('ABC', 'XYZ'). This where clause becomes part of a sql statement that is executed against an assoc, arrival and site table.

batchSizeNdef

<int> [1000]

The approximate number of defining phases that will be included in each batch of origins that will be loaded and processed. See *dbInputWhereClause* for more details.

dbOutputTablePrefix

<string> [Default =]

If this parameter is specified then the output table types specified with the dbOutputTableTypes parameter will default to the value of this parameter with the appropriate table type appended to the end.

dbOutputTableTypes

<string> [Default =]

If the dbOutputTableTypes parameter is specified then the output table types specified with this parameter will default to the value of the dbOutputTablePrefix parameter with the appropriate table type appended on the end. Currently recognized table types include: origin, assoc, arrival, site, origerr and azgap.

dbOutputOriginTable

<string> [Default = none]

Name of the origin table where output is to be written. Specifying this parameter will override any default values set by other parameters.

In general, the value of orid in the output origin table will be set to a value that is unique in the output table. However, if parameter *dbOutputConstantOrid* is set to true, then the orid value from the input origin row is retained.

Note that evids are never changed by LocOO; evid in the output origin row will be equal to evid in the input origin row.

dbOutputArrivalTable

<string> [Default = none]

Name of the arrival table where output is to be copied. Specifying this parameter will override any default values set by other parameters.

dbOutputAssocTable

<string> [Default = none]

Name of the assoc table where output is to be written. Specifying this parameter will override any default values set by other parameters.

dbOutputAzgapTable

<string> [Default = none]

Name of the azgap table where output is to be written. Specifying this parameter will override any default values set by other parameters.

dbOutputOrigerrTable

<string> [Default = none]

Name of the origerr table where output is to be written. Specifying this parameter will override any default values set by other parameters.

dbOutputAuthor

<string> [Default = '-']

Name of the output author. This is used to populate the AUTH field of the new origin row.

dbOutputConstantOrid

<bool> [Default = false]

If true, then the value of orid in the output table will be unchanged from the value in the input origin table. If false, the value of orid will be set to a value that is unique in the output origin table.

dbOutputAutoTableCreation

<bool> [Default = false]

Boolean flag should be set to true if output database tables should be created if they do not already exist.

dbOutputTruncateTables

<bool> [Default = false]

Boolean flag should be set to true if output database tables should be automatically truncated at the start of the run. Unless the *dbOutputPromptBeforeTruncate* parameter has been set to false, the user will be prompted before table truncation actually occurs.

dbOutputPromptBeforeTruncate

<bool> [Default = true]

If *dbOutputTruncateTables* is true and this parameter is true, then the user is prompted before output table truncation actually occurs. If *dbOutputTruncateTables* is true and this parameter is false, table truncation occurs without warning.