iLoc Location Algorithm

Version 4.1 May, 2023

István Bondár

ibondar2014@gmail.com

Institute for Geological and Geochemical Research,
Research Centre for Astronomy and Earth Sciences,
Eötvös Loránd Research Network (ELKH)

Bondár I., T. Šindelářová, D. Ghica, U. Mitterbauer, A. Liashchuk, J. Baše, J. Chum, C. Czanik, C. Ionescu, C. Neagoe, M. Pásztor, A. Le Pichon (2022), Central and Eastern European Infrasound Network: Contribution to Infrasound Monitoring, *Geophys. J. Int.*, **230**, 565-579, https://doi.org/10.1093/gji/ggac066.

Bondár I., R. Steed, J. Roch, R. Bossu, A. Heinloo, J. Saul and A. Strollo (2020), Accurate locations of felt earthquakes using crowdsource detections, *Front. Earth Sci.*, **8**, 272, https://doi.org/10.3389/feart.2020.00272.

Bondár, I. P. Mónus, Cs. Czanik, M. Kiszely, Z. Gráczer, Z. Wéber, and the AlpArrayWorking Group (2018). Relocation of Seismicity in the Pannonian Basin Using a Global 3D Velocity Model, *Seism. Res. Let.*, **89**, 2284-2293, https://doi.org/10.1785/0220180143.

Bondár, I., and D. Storchak (2011). Improved location procedures at the International Seismological Centre, *Geophys. J. Int.*, https://doi.org/10.1111/j.1365-246X.2011.05107.x.

Bondár, I. and K. McLaughlin (2009). Seismic location bias and uncertainty in the presence of correlated and non-Gaussian travel-time errors, *Bull. Seism. Soc. Am.*, **99**, 172-193.

Step-by-Step Installation Guide for iLoc

Download and unpack the iLoc4.1 tarball

Download the latest release of the *iLoc* package from
https://seiscode.iris.washington.edu/projects/iloc

Open the terminal application and copy the iLoc tarball into your home directory, then unpack the iLoc tarball. The files will be extracted to the *~/iLoc3.2* directory.

```
cp iLoc4.1.tgz ~/
cd
tar xzvf iLoc4.1.tgz
```

Set the environment

ILOCROOT - directory pathname for the *iLoc* root directory.

Edit your *~/.bashrc* file using a text editor (*gedit*, *vi*, *BBedit*) so that it includes

MacOSX

```
#
# RSTT, iLoc
#
export ILOCROOT=~/iLoc4.1
export PATH=$PATH:~/bin
```

Linux

```
#
# RSTT, iLoc
#
export ILOCROOT=~/iLoc4.1
export PATH=$PATH:~/bin
export LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:$HOME/lib:$LD_LIBRARY_PATH
```

Close terminal application and open it again to get your *~/.bashrc* sourced.

Install Dependencies

LAPACK and LBLAS libraries (required)

MacOSX

lapack and *lblas* are part of the MacOS X Accelerate framework, no action is required.

Linux CentOS

```
sudo yum groupinstall 'Development Tools' -y
sudo yum install lapack.x86_64 -y
```

Linux Ubuntu

```
sudo apt-get update
sudo apt-get install liblapack3
```

MySQL client (optional)

Only required if the standard SeisComp database interface is to be used. For using *mysql* interactively, store the MySQL connection info in *~/.my.cnf*. *iLoc* reads the MySQL connection info from *~/.my.cnf*.

See also

https://dev.mysql.com/doc/relnotes/connector-cpp/en/news-8-0-33.html
https://dev.mysql.com/doc/connector-cpp/8.0/en/connector-cpp-installation-binary.html

macOS

```
brew install mysql-client
```

Linux CentOS

```
sudo yum install mysql-devel -y
```

Linux Ubuntu

```
sudo apt-get update
sudo apt-get install libmysqlclient-dev
```

PostgreSQL client (optional)

Only required if the CTBTO NDC-in-a-Box IDC, the CTBTO NDC-in-a-Box SeisComp3 database or the ISC database interface is to be used. For installation instructions see the PostgreSQL website:

https://www.postgresql.org/download

Note that you need the developer package, because only that includes the necessary include files (libpq-fe.h and postgres_ext.h) for the iLoc compilation.

iLoc expects that the PGPORT environment variable is set to the port PostgreSQL uses (typically 5432). For using *psql* interactively, you may want to store the PostgreSQL connection info in *~/.pgpass*.

Linux CentOS

```
sudo yum install postgresql.x86_64
sudo yum install postgresql-devel.x86_64
```

Linux Ubuntu

```
sudo apt-get update
sudo apt-get install postgresql-client
sudo apt-get install libpq-dev
```

Oracle client (optional)

Only required if the CTBTO IDC database interface is to be used. *iLoc* expects that the ORACLE_HOME and ORACLE_SID environment variables are set.

Download and install the Oracle instant client from
https://www.oracle.com/database/technologies/instant-client.html
or alternatively, the Oracle database express edition
https://www.oracle.com/database/technologies/xe-downloads.html

Fix IDC database schema issue

The IDC NIAB and Oracle databases have a unique index on the **assoc** table by *arid*, which prevents *iLoc* to store multiple solutions for the same event. To solve the problem, connect to the database with *psql* or *sqlplus* and delete the **assaridx** index.

```
drop index assaridx;
```

Install iLoc and RSTT

Compile iLoc and RSTT (Linux and MacOS X)

Running make with the all option will unpack and compile the RSTT package; will check the existence of MySQL, Postgresql or Oracle clients and compile iLoc with the various options available. Note that all varieties of iLoc compiling options include the ISF input/output feature. The RSTT libraries will be installed in $HOME/lib, the iLoc executables in $HOME/bin.

```
cd $ILOCROOT
make all
```

iLoc variations

iLocSC: SeisComp MySQL database schema interface

iLocIDC: IDC Oracle database schema interface

iLocISC: ISC Postgresql database schema interface

iLocNiaB: IDC NDC-in-a-Box IDC Postgresql database schema interface

Check if correct libraries were linked

```
cd ~/bin
```

MacOSX

```
otool -L iLoc
```

Linux

```
ldd iLoc
```

Edit *$ILOCROOT/auxdata/iLocpars/Config.txt*

Make sure that especially these parameters are set to your preferences:

StationFile,
RSTTmodel
UseRSTTPnSn
UseRSTTPgLg
LocalVmodelFile
MaxLocalTTDelta
DoGridSearch
OutAgency

For PostgreSQL and Oracle database interfaces make sure that the *DBuser*, *DBpasswd* and *DBname* parameters are properly set.

```
cd $ILOCROOT/auxdata/iLocpars
gedit Config.txt
```

Release Notes

iLoc4.1

- updated the default depth grid from the rebuilt ISC bulletin, 1964-2020.
- support ISC ISF2.1 format
- introduced new instruction options, isf2.1, isf2.0 and ims
- for the isf2.1 option, station list is created from the ISC ISF2.1 file.
- ML and mB are calculated as an option due to unknown amplitude units in the IMS/ISF files.
- station magnitudes reported at verbose level 1.
- BEGIN and DATA_TYPE lines in the ISF2.1 output file are written only at the beginning of the file, not for each event.
- produce optional ISF2.1 file even if FixHypocenter is set.
- resolved FixEpicenter bug.
- if the magnitude phase time residual is larger than MagMaxTimeResidual its amplitude is ignored in the calculation of the station magnitude.

iLoc4.0

- Incorporated the RSTT3.2 model, TT predictions and path-dependent uncertainties.
- Harmonized auxdata directory with SeisComp iLoc plugin. The standalone iLoc and the SeisComp plugin sciLoc now read the same auxdata directory.
- Moved the RSTT model to the auxdata/RSTTmodels directory.
- Added iasp91 travel-time tables.
- Jeffreys-Bullen travel-time tables are no longer supported.
- Separated model-specific and phase-specific paramaters into Global1DModelPhaseList.txt and IASPEIPhaseMap.txt files.

iLoc3.3

- New default depth grid calculated from the EHB and ISC free-depth events from the rebuilt ISC bulletin (Storchak et al., 2017, 2020) is implemented.
- Ellipticity corrections: instead of P and S, Pup and Sup is now used for Pg,Pb and Sg, Lg, Sb, respectively.
- Reading and station magnitude comments in the output isf file are written when verbose is on.
- Bug fix in iLocReadISF.c: when reading from isf file, if an event had no magnitude block, the previous magnitude block was copied.
- Bug fix in iLocTravelTimes.c: Incorrect indices were used in getting travel time table values for exact depth values when only residuals were calculated.
- Bug fix in iLocTimeFuncs.c: When fixing origin time to a value, fractional milliseconds were incorrectly interpreted.

iLoc3.2

- Added a new instruction, MagnitudesOnly. Invoking the MagnitudesOnly instruction will not locate the event, but will calculate the magnitudes w.r.t. the preferred origin.
- Bugfixes in the magnitude calculation of ML and mB.

iLoc3.1

- Incorporated RSTT3.1 TT predictions and path-dependent uncertainties.
- RSTT do not provide travel time predictions for direct phases when the source depth is below the Moho. As a result, iLoc lost the closest stations for deep events, thus hampering depth determination. To fix the problem, if RSTT does not return a valid travel time, iLoc will use ak135 predictions.
- Improved the Makefile(s).
- Renamed the etc directory to auxdata.
- The RSTT source code and model is now found in $ILOCROOT/rstt directory.
- Updated Manual.
- Minor bug fixes.

iLoc2.1

- Generalized makefile checks for operational sytem and distribution type and the existence of lapack and lblas libraries and MySQL, Postgresql and Oracle clients.
- Bugfixes: slowness was not used in location
- Bugfixes: several formatting issues in ISF output files
- Bugfixes: iLoc no longer chokes on ISF Reference section
- Corrected typo NIOB to NIAB (NDC-in-a-Box)
- Updated Manual.

iLoc2.0

- Adopted BSD open source license.
- Changed parameter names to CamelBack notation to facilitate easier typing of instructions.
- Renamed all source files and some auxiliary data files.
- Renamed the *etc* directory to *auxdata*.
- The default RSTT model is now found in *$ILOCROOT/auxdata/RSTTmodel* directory.
- Added usage info when *iLoc* is invoked without arguments.
- Before parsing text files, the occasional Windows CRLF convention is changed to the unix/linux/MacOS LF convention.
- Added support for the NDC-in-a-box IDC PostgreSQL database schema.
- Added support for the NDC-in-a-box SeisComp3 PostgreSQL database schema.
- Added support for the IDC Oracle database schema.
- *iLoc* includes and uses the Oracle open-source ODPI-C source package to provide the Oracle database interface.
- Slowness and azimuth measurements can now be used in location.
- For azimuth-only events the origin time is fixed.
- Infrasound I and hydroacoustic H and O phases are now used in location. The phase names are fixed, that is, iLoc shall not reidentify I, H or O phases.
- Introduced phase and distance dependent *a priori* azimuth and slowness measurement errors in *$ILOCROOT/auxdata/iLocpars/PhaseConfig.txt*.
- Removed dependency on environment variable SLBMROOT in *iLocMain.c*
- Improved determination of readings by taking into account author changes when it splits phases into readings.

- The minimum number of station magnitudes required to calculate network magnitudes is now a configuration parameter, *MinNetmagSta.* It can also be overridden as an instruction.
- When calculating MS from IAMS_20, units are assumed to be in nanometres.
- Introduced configuration parameters and instructions to support the generation of Google Earth kml, or optionally, kmz files. The configuration parameters *KMLLatitude*, *KMLLongitude*, *KMLRange*, *KMLHeading*, *KMLTilt* define the default view point. The configuration parameter *KMLBulletinFile* provides the pathname for the Google Earth representation of all events processed in one *iLoc* run; if a database interface is used, the *KMLEventFile* instruction generates a Google Earth file for an individual event. If the configuration parameter, *ZipKML* is set to one, the kml file(s) will be zipped into a kmz file, and the kml file(s) will be deleted.
- Several bugfixes and optimizations.
- Added utilities to load station lists and IMS1.0/ISF1.0 bulletins into IDC Oracle and IDC NDC-in-a-Box PostgreSQL databases.
- Added *$ILOCROOT/examples* directory with sample input ISF files and their corresponding output files created by *iLoc*.
- Added *$ILOCROOT/database* directory with database schema descriptions and sample startup files for MySQL, PostgreSQL and Oracle.
- Updated Manual.

iLoc1.60

- Introduced the computation of local magnitudes, ML and the broadband body wave magnitudes, mB. New, mB and ML specific parameters (*mB_mag_min_dist, mB_mag_max_dist, ML_mag_max_dist*) are given in the *$ILOCROOT/etc/iLocpars/Config.txt* file.
- Phases used to compute ML are listed in *$ILOCROOT/etc/iLocpars/ak135_model.txt*.

iLoc1.50

- Replaced TT calculations from local velocity models with much simpler and faster code. Now only the direct phase and headwave(s) are calculated, i.e. the phases Pg/Pb/Pn/P and Sg/Sb/Sn/S. Lg is replicated as Sg. The format of the local velocity model has also changed. The first non-comment line in the local velocity model file is expected to be the number of layer boundaries in the model, followed by the description of each layer in the model, i.e. the depth of the top of the layer, its P and S velocity and an indicator for the Conrad or Moho discontinuities.

iLoc1.40

- Added SeisComp3 database support. *iLoc* can communicate with a SeisComp3 database schema through a MySQL client. The SeisComp3 PostgreSQL client is not yet supported.
- RSTT support is included by default. *iLoc* can optionally take travel-time predictions for crustal and mantle phases from a 3D velocity model compliant with the RSTT parameterization (Myers et al., 2010).
- Added support for reading input files in ISF2, the format the NEIC PDE is produced since 2013. When ISF output is required, it is produced in ISF2.
- Added support for local velocity models. One can provide a 1D local velocity model from which travel time tables for local phases will be calculated and used in the

location (up to *max_localTT_delta* distance). Alternatively, the local 1D velocity model at the initial hypocentre can be extracted from the RSTT model and used in the travel time table calculations.

- Added a new instruction, *fix_depth_median* that allows to fix the depth to the median of the reported depths.

1 Getting Started

1.1 Usage

```
iLoc [isf2.1|isf2.0|ims|isc|seiscomp|idc|niab] < instructionfile
echo "instructions…" | iLoc [isf2.1|isf2.0|ims|isc|seiscomp|idc|niab]
```

Text file input/output

If the argument is '**ims**', '**isf2.0**' or '**isf2.1**', *iLoc* will expect input from an ISF1.0/IMS1.0, ISF2.0 or an ISF2.1 text file. The log will be written to the file specified in the *Config.txt* file. The *ISFInputFile* parameter is expected among the instructions, as well as the *StationFile* parameter, either specified in the *Config.txt* file or given as an instruction. We recommend that the users download the file of station parameters (*isc_stalist*) from the ISC website (download.isc.ac.uk/iscloc/isc_stalist), where it is regularly updated. The *$ILOCROOT/utilities* directory contains perl scripts to update the IR2_stalist file required for ISF2 input. Note that the ISF2.1 format lists the station coordinates for each arrival, so for the '**isf2.1**' option there is no need to specify a *StationFile* parameter.

Instructions are taken from stdin or redirected from an instruction file. Only the first line of the instruction file is parsed, and the instructions will apply to all event in the *ISFInputFile*. Thus if you want to specify individual parameters for each event, you should create separate ISF files for each event.

Output is optionally written in ISF2 to the file specified by the *ISFOutputFile* parameter. If the *KMLBulletinFile* or *KMLEventFile* specified, *iLoc* will create a Google Earth kml file for all events in *ISFInputFile*. If *ZipKML* = 1, the kml file(s) will be zipped into a kmz file, and the kml file(s) will be deleted.

Database input/output

- If the argument is '**idc**', *iLoc* will expect input from an Oracle database compliant with the IDC database schema (see **Appendix B)**. The database connection parameters *DBuser, DBpasswd* and *DBname* are expected to set in *Config.txt*. The ORACLE_HOME and ORACLE_SID environment variables should also be set.
- If the argument is '**niab**', *iLoc* will expect input from a PostgreSQL database compliant with the IDC NDC-in-a-Box PostgreSQL database schema (see **Appendix B)**. The database connection parameters *DBuser, DBpasswd* and *DBname* are expected to set in *Config.txt*.
- If the argument is '**seiscomp**', *iLoc* will expect input from a MySQL database compliant with the SeisComp3 database schema (see **Appendix A**). The MySQL client parameters (host, port, socket, user, password) are expected in the client section of the user's *~/.my.cnf* file.
- If the argument is '**isc**', *iLoc* will expect input from a PostgreSQL database compliant with the ISC PostgreSQL database schema (see **Appendix C)**. The database connection parameters *DBuser, DBpasswd* and *DBname* are expected to set in *Config.txt*.

Instructions will be taken from stdin or can be redirected from an instruction file. An instruction file contains lines for each event. The instruction lines must begin with the event identifier, followed by the event specific instructions. Thus, each event can be processed with its own separate instructions. The log will be written to the file specified in the *Config.txt* file.

If the configuration parameter *UpdateDB* = 1, results will be written to the database; if *UpdateDB = 0*, result will not be written to the database, thus allowing for trial-and-error runs. By default, *iLoc* reads and writes the account identified by the client user. Optionally input data can be read from the *InputDB* and written to the *OutputDB* account, and new unique identifiers can be obtained from the *NextidDB* account. The user must have read permission to *InputDB* and read/write permission to *OutputDB* and *NextidDB*.

If *ISFOutputFile* is specified, results will also be written to an ISF2 file. If the configuration parameter *KMLBulletinFile* is specified, *iLoc* will create a Google Earth kml file for all events listed in the instruction file. If an instruction line specifies *KMLEventFile*, a Google Earth kml is created for that specific event. If *ZipKML* = 1, the kml file(s) will be zipped into a kmz file, and the kml file(s) will be deleted.

We recommend that the users download the file of station parameters (*isc_stalist*) from the ISC website (www.isc.ac.uk), where it is regularly updated. The *$ILOCROOT/utilities* directory contains *perl* scripts to update the database with the latest ISC station list and update the IR2_stalist file required for ISF2 input.

Examples

```
echo "ISFInputFile=p.isf ISFOutputFile=u.isf2.1" | iLoc ims
echo "ISFInputFile=p.isf2.0 ISFOutputFile=u.isf2.1 StationFile=./IR.txt" | iLoc isf2.0
echo "ISFInputFile=p.isf2.1 ISFOutputFile=u.isf2.1" | iLoc isf2.1
echo "bud2014xjmk DoGridSearch=0 StartDepth=NEIC" | ilocSC seiscomp
echo "567493 FixDepth=0 UpdateDB=0 Verbose=1" | ilocNiaB niab
iLocSC seiscomp < instruction_file > logfile
iLocISC isc < instruction_file > logfile
iLocIDC idc < instruction_file > logfile
```

Finally, the one-liner for not the faint hearted that connects to the SeisComp3 database, selects a subset of events, generates the instruction lines and runs *iLoc*:

```
    echo "`mysql -u username -sN -e"select ep.publicID, 'UpdateDB=1 DoGridSearch=0
UseRSTTPnSn=1 UseRSTTPgLg=1' from Event e, PublicObject ep where e._oid=ep._oid and
ep.publicID like '%306%'" | sed 's/\t/ /g'`" | iLocSC seiscomp > subset.log
```

1.2 Configuration parameters

The *iLoc* configuration parameters are read from *$ILOCROOT/auxdata/iLocpars/Config.txt*. See also the comments in *$ILOCROOT/auxdata/iLocpars/Config.txt*.

Table 1. Configuration parameters

Name	Type	Description	Default
StationFile	string	Pathname for the station file. Required for ISF/ISF2 input. **isf** mode expects isc_stalist; **isf2** expects IR2_stalist.	isc_stalist
RSTTmodel	string	Full pathname for the RSTT model.	rstt201404um.geotess
UseRSTTPgLg	integer	Use RSTT crustal phase predictions [0/1].	1
UseRSTTPnSn	integer	Use RSTT mantle phase predictions [0/1].	1
UpdateDB	integer	Write results to DB [0/1].	0

OutAgency	string	Author for new hypocentres and associations.	ILOC
DBuser	string	Database username. Required from modes **idcdb**, **idcniab**, **sc3niab** and **iscdb**.	
DBpasswd	string	Database password. Required from modes **idcdb**, **idcniab**, **sc3niab** and **iscdb**.	
DBname	string	Database host. Required from modes **idcniab**, **sc3niab** and **iscdb**.	
InputDB	string	Input database account if it differs from DBuser. DBuser must have read permission.	
OutputDB	string	Output database account if it differs from DBuser. DBuser must have read and write permissions.	
NextidDB	string	Account for new unique ids if it differs from DBuser. DBuser must have read and write permissions.	
InAgency	string	Author for input associations. ISC-specific.	ISC
repid	integer	Reporter id for new hypocentres and associations. ISC-specific.	100
KMLBulletinFile	string	Pathname for the output Google Earth kml bulletinfile.	
KMLLatitude	double	Viewpoint latitude	47.49833
KMLLongitude	double	Viewpoint longitude	19.04045
KMLRange	double	Viewpoint elevation (m)	1000000
KMLHeading	double	Viewpoint azimuth	0
KMLTilt	double	Viewpoint tilt	0
ZipKML	integer	Zip KML files and keep kmz only [0/1]	1
TTimeTable	string	TT table name [ak135 | jb].	ak135
LocalVmodelFile	string	Pathname for user-provided local 1D velocity model	
LocalTTfromRSTT	integer	Use local TT tables from the local velocity model extracted from RSTT [0/1]	0
MaxLocalTTDelta	double	Use local TT tables up to this distance [degrees]	3.
EtopoFile	string	Filename for ETOPO file.	etopo5_bed_g_i2.bin
EtopoNlon	integer	Number of longitude samples in ETOPO.	4321
EtopoNlat	integer	Number of latitude samples in ETOPO.	2161
EtopoRes	double	Cellsize in ETOPO.	0.0833333
NohypoAgencies	string	Comma separated list of agencies not to be used in setting the initial hypocentre.	UNK,NIED,HFS,HFS1, HFS2,NAO
DoGridSearch	integer	Perform Neighbourhood Algorithm (NA) search [0/1].	1
NAsearchRadius	double	Search radius around starting epicentre (degrees).	5.
NAsearchDepth	double	Search radius around starting depth (km).	300.
NAsearchOT	double	Search radius around starting origin time (s).	30.
NAiterMax	integer	Maximum number of iterations in NA search.	5
NAlpNorm	double	p-value for norm to compute misfit in NA search [$1 \le p \le 2$].	1
NAinitialSample	integer	Number of initial samples in NA search.	1500

NAnextSample	integer	Number of subsequent samples in NA search.	150
NAcells	integer	Number of cells to be resampled in NA search.	25
iseed	integer	Random number seed.	5590
MinDepthPhases	integer	Minimum number of time-defining first arriving P and depth phase pairs for depth-phase depth resolution.	3
MindDepthPhaseAgencies	integer	Minimum number of agencies reporting depth phases or PcP/ScS for depth resolution.	1
MinCorePhases	integer	Minimum number of time-defining first arriving P and core phase (PcP, ScS) pairs for depth resolution.	3
MaxLocalDistDeg	double	Maximum epicentral distance for local stations (degrees).	0.2
MinLocalStations	integer	Minimum number of time-defining first arriving P within *MaxLocalDistDeg* for depth resolution.	1
MaxSPDistDeg	double	Maximum epicentral distance for near-regional stations (degrees).	2
MinSPpairs	integer	Minimum number of time-defining first arriving P and S pairs for depth resolution.	3
DefaultDepth	double	Used if seed hypocentre depth is null.	0.
MaxShallowDepthError	double	Maximum depth error for crustal events to accept a free-depth solution.	30.
MaxDeepDepthError	double	Maximum depth error for deep events to accept a free-depth solution.	60.
SigmaThreshold	double	Residuals beyond *SigmaThreshold * prior measurement error* will be made defining.	6.
DoCorrelatedErrors	integer	Account for correlated travel-time prediction errors [0/1].	1
AllowDamping	integer	Allow damping of solution [0/1].	1
MinIterations	integer	Minimum number of iterations.	4
MaxIterations	integer	Maximum number of iterations.	20
MinNdefPhases	integer	Minimum number of defining observations. If the number of time defining observations exceeds MinNdefPhases, slowness measurements will not be used in location.	4
ConfidenceLevel	double	Confidence level for formal uncertainties [90 | 95 | 98].	90.
MinNetmagSta	integer	Minimum number of station magnitudes required for network magnitudes.	3
mbQ_table	string	Magnitude attenuation correction table for mb [GR | VC | MB | none].	MB
MagMaxTimeResidual	double	Allowable time residual (if any) for a magnitude phase to be used in magnitude calculations	10.
CalculatemB	integer	Calculate broadband mB [0/1]	0
CalculateML	integer	Calculate local magnitude ML [0/1]	0
mbMinDistDeg	double	Minimum mb distance (degrees).	21.
mbMaxDistDeg	double	Maximum mb distance (degrees).	100.

mbMinPeriod	double	Minimum mb period (s).	0.3
mbMaxPeriod	double	Maximum mb period (s).	5.
BBmBMinDistDeg	double	Minimum mB distance (degrees).	5.
BBmBMaxDistDeg	double	Maximum mB distance (degrees).	105.
MLMaxDistkm	double	Maximum ML distance (km).	600.
MSMinDistDeg	double	Minimum MS distance (degrees).	20.
MSMaxDistDeg	double	Maximum MS distance (degrees).	160.
MSMinPeriod	double	Minimum MS period (s).	10.
MSMaxPeriod	double	Maximum MS period (s).	60.
MSMaxDepth	double	Maximum MS depth (km).	60.
MSPeriodRange	double	Period tolerance around MS_Z when searching for horizontal MS amplitudes (s)	5.
MagnitudeRangeLimit	double	Generate warning if station magnitude range exceeds this.	2.2
logfile	string	Pathname for the iLoc logfile.	stdout
errfile	string	Pathname for the iLoc error file.	stderr

Parameters specific to the global 1D velocity models ak135 and iasp91 are read from the *$ILOCROOT/auxdata/iLocpars/Global1DModelPhaseList.txt* file that also includes the list of phases with valid travel times.

Table 2. Configuration parameters specific to global 1D velocity models

Name	Type	Description
Moho	double	Depth of Moho discontinuity in ak135/iasp91 (km).
Conrad	double	Depth of Conrad discontinuity in ak135/iasp91 (km).
MaxHypocenterDepth	double	Maximum allowable event depth (km).
SSurfVel	double	Sg velocity for elevation corrections in ak135 (km/s).
PSurfVel	double	Pg velocity for elevation corrections in ak135 (km/s).
PhaseMap	table	Map reported phases to IASPEI standard phase names.
PhaseTT	table	List of phases with ak135/iasp91 travel-time predictions.

Phase-specific parameters are read from the *$ILOCROOT/auxdata/iLocpars/IASPEIPhaseMap.txt* file. See also the comments in *$ILOCROOT/auxdata/iLocpars/IASPEIPhaseMap.txt*.

Table 3. Phase-specific configuration parameters

Name	Type	Description
PhaseMap	table	Map reported phases to IASPEI standard phase names.
AllowablePhases	table	List of IASPEI phase names to which reported phases can be renamed.
AllowableFirstP	table	List of IASPEI phase names to which reported first-arriving P phases can be renamed.
OptionalFirstP	table	Additional list of IASPEI phase names to which reported first-arriving P phases can be renamed.
AllowableFirstS	table	List of IASPEI phase names to which reported first-arriving S phases can be renamed.
OptionalFirstS	table	Additional list of IASPEI phase names to which reported first-arriving S phases can be renamed.

PhaseWithoutResidual	table	List of IASPEI phase names for which residuals will not be calculated (and not used in the location, e.g. amplitude phases).
PhaseWeight	table	List of *a priori* time, azimuth and slowness measurement errors within specified delta ranges for IASPEI phases. (phase, mindelta, maxdelta, time azimuth, slowness measurement error values)
MBPhase	table	List of phase names that contribute to mb/mB calculations.
MSPhase	table	List of phase names that contribute to MS calculation.
MLPhase	table	List of phase names that contribute to ML calculation.

1.3 Instructions

Instructions can either be piped from the command line or read from an instruction file. An instruction line is expected for every event in the form:

eventid [par=value [par=value [par [par=value...]]]

where *eventid* is the event identifier string and the par=value pairs denote the optional parameter name and value pairs.

Note:

For ISF/ISF2 input only the first line of the instruction file is interpreted, and the instructions are applied to *all* events in the input ISF/ISF2 file. For ISF/ISF2 input the *eventid* parameter is not required.

Tables 4 and 5 provide the list of instructions accepted by *iLoc*.

Table 4. Instructions

Name	Type	Description
Verbose	integer	Level of verbosity (0..5).
ISFInputFile	string	Pathname to the ISF/ISF2 input text file.
ISFOutputFile	string	Pathname to the ISF2 output text file.
KMLEventFile	string	Pathname for the output Google Earth kml event file.
StartLat	double or string	A latitude value or an agency code to start the locator from. If no value or code is given, the initial latitude will be set to the median reported latitude.
StartLon	double or string	A longitude value or an agency code to start the locator from. If no value or code is given, the initial longitude will be set to the median reported longitude.
StartDepth	double or string	A depth value or an agency code to start the locator from. If no value or code is given, the initial depth will be set to the median reported depth.
StartOT	YYYY-MM-DD_HH:MI:SS.SSS or string	A date-time value or an agency code to start the locator from. If no value or code is given, the initial origin time will be set to the median reported origin time.
FixOriginTime	YYYY-MM-DD_HH:MI:SS.SSS or string	A date-time value or an agency code to fix the origin time to. If no value or code is given, the origin time will be fixed to the median reported origin time.
FixEpicenter	string	An agency code to fix the epicentre to. If no code is given, the epicentre will be fixed to the median reported latitude, longitude values.

FixDepth	double or string	A depth value or an agency code to fix the depth to. If no value or code is given, the depth will be fixed to the median reported depth.
FixDepthToDefault	integer	Fix the depth to the region-dependent default depth (from default depth grid or from Flinn-Engdahl region number).
FixDepthToMedian	integer	Fix the depth to the median of the reported depths.
FixHypocenter	string	An agency code to fix the hypocentre to. If no code is given, the hypocentre will be fixed to the median reported hypocentre (lat, lon, depth, origin time) parameters.
WriteNAResultsToFile	integer	Write Neighbourhood Algorithm results to file [0/1].
DoNotRenamePhase	integer	Do NOT reidentify phases [0/1]
MagnitudesOnly	integer	Calculate magnitudes w.r.t. the preferred origin

The instructions below can temporarily (for one event only) override the parameter values set in the *Config.txt* file.

Table 5. Instructions that can temporarily override configuration parameters

Name	Type	Description
UpdateDB	integer	Write results to database [0/1].
StationFile	string	Pathname to the station file when not read from DB.
RSTTmodel	string	Full pathname for the RSTT model.
UseRSTTPgLg	integer	Use RSTT crustal phase predictions [0/1].
UseRSTTPnSn	integer	Use RSTT mantle phase predictions [0/1].
LocalTTfromRSTT	integer	Use local TT tables from the local velocity model extracted from RSTT [0/1]
MaxLocalTTDelta	double	use local TT tables up to this distance [degrees]
LocalVmodelFile	string	pathname for user-provided local 1D velocity model
DoCorrelatedErrors	integer	Account for correlated travel-time prediction errors [0/1].
DoGridSearch	integer	Perform Neighbourhood Algorithm search [0/1].
iseed	integer	Random number seed.
NAsearchRadius	double	Search radius around starting epicentre (degrees).
NAsearchDepth	double	Search radius around starting depth (km).
NAsearchOT	double	Search radius around starting origin time (s).
NAinitialSample	integer	Number of initial samples in NA search.
NAnextSample	integer	Number of subsequent samples in NA search.
NAcells	integer	Number of cells to be resampled in NA search.
NAiterMax	integer	Maximum number of iterations in Neighbourhood Algorithm search.
MinNetmagSta	integer	Minimum number of station magnitudes required for network magnitudes.
MindDepthPhaseAgencies	integer	Minimum number of agencies reporting depth phases or PcP/ScS for depth resolution.
MinDepthPhases	integer	Minimum number of time-defining first arriving P and depth phase pairs for depth-phase depth resolution.
MinCorePhases	integer	Minimum number of time-defining first arriving P and core phase (PcP, ScS) pairs for depth resolution.
MaxLocalDistDeg	double	Maximum epicentral distance for local stations (degrees).

MinLocalStations	integer	Minimum number of time-defining first arriving P within localdist for depth resolution.
MaxSPDistDeg	double	Maximum epicentral distance for near-regional stations (degrees).
MinSPpairs	integer	Minimum number of time-defining first arriving P and S pairs for depth resolution.
MagMaxTimeResidual	double	Allowable time residual (if any) for a magnitude phase to be used in magnitude calculations (seconds).
CalculatemB	integer	Calculate broadband mB [0/1].
CalculateML	integer	Calculate local magnitude ML [0/1].
MSPeriodRange	double	Period tolerance around MS_Z when searching for horizontal MS amplitudes (s)
KMLBulletinFile	string	Pathname for the output Google Earth kml bulletinfile.
InputDB	string	Input database account if it differs from DBuser. DBuser must have read permission.
OutputDB	string	Output database account if it differs from DBuser. DBuser must have read and write permissions.
NextidDB	string	Account for new unique ids if it differs from DBuser. DBuser must have read and write permissions.

1.4 Auxiliary data files

The locator reads the data files listed below (see also **Appendix D**).

Table 6. Data files

Pathname	Description
$ILOCROOT/auxdata/iLocpars/Config.txt	Configuration parameters.
$ILOCROOT/auxdata/iLocpars/Global1DModelPhaseList.txt	Model-specific configuration parameters.
$ILOCROOT/auxdata/iLocpars/IASPEIPhaseMap.txt	Phase-specific configuration parameters.
$ILOCROOT/auxdata/ak135/*.tab or $ILOCROOT/auxdata/iasp91*.tab	Travel-time tables. The ak135 and iasp91 travel-time tables were generated by *libtau*.
$ILOCROOT/auxdata/ak135/ELCOR.dat $ILOCROOT/auxdata/iasp91/ELCOR.dat	Ellipticity correction coefficients for ak135/iasp91.
$ILOCROOT/RSTTmodels/pdu202009Du.geotess	RSTT3.2 3D global velocity model.
$ILOCROOT/auxdata/topo/etopo5_bed_g_i2.bin	Topography/bathymetry data to calculate bounce point corrections. The ETOPO1 bedrock file was resampled to 5'x5' resolution (binary file of etoponlat * etoponlon 2-byte integers).
$ILOCROOT/auxdata/FlinnEngdahl/FE.dat	Flinn-Engdahl regionalization.
$ILOCROOT/auxdata/FlinnEngdahl/DefaultDepth0.5.grid	Default depth grid defined on a 0.5 x 0.5 degree grid.
$ILOCROOT/auxdata/FlinnEngdahl/GRNDefaultDepth.ak135.dat $ILOCROOT/auxdata/FlinnEngdahl/GRNDefaultDepth.iasp91.dat	Default depths for Flinn-Engdahl regions for locations where no default depth grid point exists.
$ILOCROOT/auxdata/variogram/variogram.model	Generic variogram model to calculate *a priori* data covariance matrix.
$ILOCROOT/auxdata/magnitude/GRmbQ.dat or $ILOCROOT/auxdata/magnitude/MBmbQ.dat or $ILOCROOT/auxdata/magnitude/VCmbQ.dat	Magnitude attenuation Q(d,h) curves. Currently the GR (Gutenberg-Richter),

	VC (Veith-Clawson) and MB (Murphy-Barker) curves are supported.
$ILOCROOT/auxdata/Stations/isc_stalist or $ILOCROOT/auxdata/Stations/IR2_stalist	Station parameters. Required for ISF or ISF2 input.
$ILOCROOT/auxdata/localmodels/*.dat	Sample local velocity models

2 Event location

2.1 Initializations

The initialization step sets the starting hypocentre according to the instructions given for the event. Since the reported hypocentres may exhibit a large scatter, and it is not guaranteed that any of them is close to the global optimum, the initial hypocentre parameters (latitude, longitude, origin time, depth) are set to the median of the corresponding reported hypocentre parameters. However, if the author of the prime (preferred) hypocentre is IASPEI, the initial hypocentre will be set to the prime hypocentre. Furthermore, if instructions (**Table 4**) are given to fix or provide starting values for any of the hypocentre parameters, they will be set according to the instructions.

The event type is read from the **event** (ISC), **Event** (SeisComp3) or **origin** (IDC) table or taken from the prime hypocentre if the input is ISF). If the event type indicates an anthropogenic event, the depth will be fixed to the surface, unless an instruction is given to fix the depth to some other specific value.

2.2 Phase identification

Reported phases are mapped to IASPEI standard phase names (Storchak et al., 2003) using the map provided in *PhaseMap* (**Table 3**). The *PhaseMap* table lists all possible variations of reported phase names and the corresponding IASPEI phase name. Unrecognised phase names are mapped to a null value. However, if the *DoNotRenamePhase* instruction is set to true, none of the reported phases will be reidentified by *iLoc*. Infrasound I and hydroacoustic H and O pahases are not reidentified. ISC-specific: if the *phase_fixed* flag is set in the ISC **association** table, the phase will not be renamed.

Once the reported phase names are mapped to the IASPEI standards and *DoNotRenamePhase* is false, phases are reidentified with respect to the initial hypocentre. If the Neighbourhood Algorithm (NA) search is turned on, phases are identified with respect to each trial hypocentre. Once NA is finished, the phases will be identified with respect to the refined initial hypocentre guess obtained from the NA search. In order to avoid oscillating solutions, phase names are kept the same during the iterations of the linearized least-squares algorithm unless the depth crosses the Moho or Conrad discontinuities between two iterations. In that case local and regional phases are reidentified.

Phases in a reading (phases reported by a single agency for an event at a single station) are treated as a group in *iLoc*. In each reading the phases are identified with respect to the initial hypocentre. Phases that are listed in *PhaseWithoutResidual* (**Table 3**) are skipped, thus preventing them to be used in the location.

For a phase in a reading the phase type is determined by the first leg of the ray path indicated by the first letter in the phase name; for depth phases the phase type is determined by the second letter in the phase name (e.g. Pn, sP and pwP are P-type phases, Sn, ScP, Lg and pS are S-type phases). Currently only P-type, S-type, H-type and I-type phases are considered.

iLoc checks if the phase is in the list of *AllowablePhases* (**Table 3**). The list of allowable phases was introduced to prevent the locator renaming phases to unlikely 'exotic' phases, just because a travel-time prediction fits the observed travel-time better. For instance, we do not want to reidentify a reported Sn as SgSg, SPn or *horribile dictu*, sSn. Recall that phases may suffer from picking errors or the observed travel-times may reflect true 3D structures not modelled by the velocity model. Introducing the list of allowable phases helps to maintain the sanity of the bulletin and mitigates the risk of misidentifying phases. However, if a reported phase is not in the list of allowable phases, it is temporarily added to the list accepting the fact that station operators may confidently pick later phases. In other words, exotic phase names can appear in the final bulletin only if they were reported as such. Note that iLoc does not attempt to reidentify infrasound and hydroacoustic (I, H and O) phases.

For each phase in a reading *iLoc* loops through the (possibly amended) list of allowable phases and calculates the time residual with respect to the initial hypocentre. Certain rules apply:

- Cannot rename a P-type phase to S-type, and vice versa.
- A phase name can appear only once in a reading.
- I, H, and O phases are not renamed.

Further restrictions apply to first-arriving P and S phases. First-arriving P and S phases can be identified as those in the list of allowable first-arriving P and S phases (**Table 3**). Occasionally a station operator may not report the true first-arriving phase due to high noise conditions. To account for this situation the list of optional first-arriving P and S phases (**Table 3**) is also checked.

Finally, having observed the rules above, the phase is identified as the phase in the allowable phase list that has the smallest residual. If no eligible phase has been found, i.e. if the smallest residual is beyond ±60 seconds, the phase is treated as unidentified.

Once the phases are identified with respect to the initial hypocentre, *iLoc* sets the time defining flag and the *a priori* estimate of the time measurement error for each phase. The prior measurement errors are given in *PhaseWeight* (**Table 3**). The time defining flag is set to true if a valid entry is found in the *PhaseWeight* table. However, the phase is made non-defining if its residual is larger than *SigmaThreshold* times the prior time measurement error. ISC-specific: a phase can be was explicitly set to non-defining by an analyst (i.e. the *non_def* flag is set in the ISC **association** table).

Since phases can be picked and reported by several agencies for the same station, there can be multiple entries for the same phase arrival. At this stage *iLoc* considers time-defining phases only. Arrival picks are considered duplicates if they are reported at the same site for the same event and if they arrival times is within 0.1 seconds. To account for alternative station codes, the primary stations codes are used. For duplicates the arrival time is taken as the mean of the arrival times, and the phase name is forced to be the one with the smallest residual. If accounting for correlated errors is turned off, duplicates are explicitly down-weighted. However, if correlated errors are accounted for, down-weighting is not necessary as duplicates are simply projected to the null space.

Azimuth and slowness measurements are also used in the location. The prior azimuth and slowness measurement errors are given in *PhaseWeight* (**Table 3**). The slowness or azimuth defining flag is set to true if a valid entry is found in the *PhaseWeight* table. However, the slowness or azimuth is made non-defining if its residual is larger than *SigmaThreshold* times the prior slowness or azimuth measurement error.

Only time, azimuth and slowness defining observations are used in the location. A phase can have any combinations of time, azimuth and slowness defining flags. If the number defining observations is less than *MinNdefPhases*, *iLoc* will not locate the event, but will calculate the residuals with the hypocentre fixed. Furthermore, if the number of time defining observations exceeds *MinNdefPhases*, there are sufficient number of observations to locate the event without using the less reliable slowness measurements. In that case *iLoc* will make all slowness observations non-defining.

2.3 Travel-time prediction

2.3.1 Global 1D travel-time tables

Currently *iLoc* supports the iasp91 (Kennett and Engdahl, 1991) and the *ak135* (Kennett et al., 1995) travel-time tables, as well as RSTT3.2 (Myers et al., 2010, Begnaud et al., 2020, 2021) crustal and mantle phase travel-time predictions.

The *ak135* and iasp91 travel-time tables were generated by *libtau* (Buland and Chapman, 1983. *ak135* and iasp91 offers an abundance of phases from the IASPEI standard phase list (Storchak et al., 2003) that can be used in the location, most notably the PKP branches and depth-sensitive phases. Composite tables for first-arriving P and S phases were also constructed and they are used to get a valid travel-time table value at local/regional crossover distances without reidentifying the phase during the subsequent iterations of the location algorithm.

The travel-time table values for a given phase, delta and depth are calculated by bicubic spline interpolation. Vertical slowness table values are calculated only if vertical partial derivatives are requested. For depth phases bounce point distances are also calculated.

The P and S velocities in the surface layer to calculate elevation corrections are given by *PSurfVel* and *SSurfVel* (**Table 2**). Ellipticity corrections (Dziewonski and Gilbert, 1976; Kennett and Gudmundsson, 1996), using the WG84 ellipsoid parameters, are added to the *ak135* predictions.

Bounce point (elevation correction at the surface reflection point) corrections are calculated by the EHB algorithm (Engdahl et al., 1998) for depth phases. Water depth corrections for pwP are calculated for water columns exceeding 1.5 km. We use the ETOPO1 global relief model (Amante and Eakins, 2009), resampled to 5' x 5' resolution, to obtain the elevation or the water depth at the bounce point. The ETOPO parameters are specified in the *Config.txt* file (**Table 1**).

2.3.2 Global 3D RSTT regional travel-time predictions

RSTT Pg/Lg and Pn/Sn travel-time predictions can be enabled separately via the *UseRSTTPgLg* and *UseRSTTPnSn* parameters. Obviously, if both parameters are set to 0, RSTT predictions will not be used. By default, *UseRSTTPnSn* = 1 and *UseRSTTPgLg* = 1. RSTT predictions include the elevation and ellipticity corrections. If enabled, iLoc uses the RSTT predictions if

- epicentral distance is less than 15°, and
- the phase is Pn/Sn or first-arriving crustal Pg/Pb/Sg/Sb/Lg

2.3.3 Local 1D velocity model travel-time predictions

Optionally the user can provide a local 1D velocity model, from which travel time tables for local phases will be calculated. Alternatively, the local 1D velocity model at the initial hypocentre can be extracted from the RSTT model, and used for TT generation. The local TT predictions are applied up to *MaxLocalTTDelta* distance.

All other phases get RSTT and/or *ak135* predictions.

2.4 Data covariance matrix

If the observations are assumed to be independent, the data covariance matrix is a diagonal matrix where the diagonal elements are the *a priori* estimates of the measurement error variances. The phase and distance dependent time, azimuth and slowness measurement errors are specified in *PhaseWeight* (**Table 3**). The inverse of the data covariance matrix provides the diagonal weight matrix.

When correlated travel-time prediction errors are present, the data covariance matrix is no longer diagonal, and the redundancy in the observations reduces the effective number of degrees of freedom (Bondár and McLaughlin, 2009). Assuming that the similarity between ray paths is well approximated by the station separation, the covariances between station pairs can be estimated from a generic P variogram model derived from ground truth residuals. The estimates for the elements of the data covariance matrix are obtained as

$$C_D(i,j) = \sigma_{sill}^2 - \gamma(h_{ij}) + \delta_{ij}\sigma_{phase}^2$$

where σ_{sill}^2 denotes the background variance where the variogram levels off (i.e. where the pairs become independent), $\gamma(h_{ij})$ is the variogram value for the distance, h_{ij}, between the i^{th} and j^{th} stations, δ_{ij} is the Kronecker delta and σ_{phase}^2 is the *a priori* estimate of the measurement error covariance for an observed phase. The last term indicates that the measurement error variances add to the diagonal of the covariance matrix.

Because in this representation the covariances depend only on station separations, the covariance matrix (and its inverse) needs to be calculated only once. We assume that different phases owing to the different ray paths they travel along, as well as station pairs with a separation larger than 1,000 km are uncorrelated. Hence, the data covariance matrix is a sparse, block-diagonal matrix.

Furthermore, if the stations in each phase block are ordered by their nearest neighbour distance, the phase blocks themselves become block-diagonal. We determine the nearest neighbour ordering of the stations by performing a single-linkage hierarchical cluster analysis (de Hoon et al., 2004; Sibson, 1973) using the distance matrix constructed from the station separations.

To reduce the computational time of inverting large matrices we exploit the inherent block-diagonal structure by inverting the covariance matrix block-by-block. The singular value decomposition of the data covariance matrix is written as

$$\mathbf{C}_D = \mathbf{U}_D \mathbf{\Lambda}_D \mathbf{V}_D^T$$

where $\mathbf{\Lambda}_D$ is the diagonal matrix of eigenvalues and the columns of $\mathbf{U}_D$ contain the eigenvectors of $\mathbf{C}_D$. Let $\mathbf{C}_D = \mathbf{B}\mathbf{B}^T$, with $\mathbf{B} = \mathbf{U}_D \mathbf{\Lambda}_D^{1/2}$, then the projection matrix

$$\mathbf{W} = \mathbf{B}^{-1} = \mathbf{\Lambda}_D^{-1/2} \mathbf{U}_D^T$$

will orthogonalize the data set and project redundant observations into the null space.

2.5 Neighbourhood Algorithm (NA) search

Linearized inversion algorithms are quite sensitive to the initial guess. However, it is not guaranteed that any of the reported hypocentres would provide a suitable starting point for the linearized inversion. In order to find an initial hypocentre guess for the linearized inversion we run the Neigbourhood Algorithm (Sambridge, 1999; Sambridge and Kennett, 2001; Kennett, 2006) around the starting hypocentre.

The parameters governing the NA search are read from the *Config.txt* file (**Table 1**) or can be given as instructions (**Table 5**). As with the linearized location algorithm, the forward calculations in NA use all H, I, P and S-type phases and can account for correlated travel-time prediction errors. Note that for events that have a reasonable network geometry (networks that are not heavily unbalanced) we temporarily disable the accounting for correlated errors in order to improve performance. During the NA search, we identify the phases with respect to each trial hypocentre and calculate the misfit of the trial hypocentre. The misfit is defined as

$$misfit = \frac{\left(\sum_{i=0}^{N-1} |tres_i|^p\right)^{1/p}}{Nrank - M} + \alpha \frac{Nass - Ndef}{Nass}$$

where *Nass* is the total number of P and S-type phases, *Ndef* is the number of defining observations, *Nrank* is the number of independent defining observations, *M* is the number of degrees of freedom (the number of model parameters), α (=4.1) is a penalty factor, and *p* is defined by the parameter *NAlpNorm*. Recall that the phase identification process makes an observation non-defining if its residual exceeds the threshold defined by *SigmaThreshold* times the prior measurement error. Thus, each trial hypocentre can have different number of defining phases. The second term in the misfit expression is introduced to penalize against freakish local minima provided by just a few phases.

The NA parameters, *NAinitialSample*, *NAnextSample* and *NAcells*, are tuned to achieve a reasonable compromise between speed and the exhaustive exploration of the search space. An initial sample size (*NAinitialSample*) of 3000 or larger provides a very thorough initial sampling of the search space, but for large events with thousands of phases the NA search would be very slow. An initial sample size somewhere between 500 and 1000 typically provides good results. The parameter *NAcells* determines the number of the cells with the best misfits whose neighbourhood is explored in subsequent NA iterations; *NAnextSample* is the number of samples generated in subsequent iterations. Thus, for *NAcells* =25 and *NAnextSample* =100, each candidate cell is resampled by four new samples. Keeping the number of cells to be resampled relatively high prevents NA prematurely falling into a local minima.

Local and near-regional events with lots of stations around the Pg/Pb/Pn crossover distance range could generate misfit surfaces with quite a number of local minima. In such cases, it is advisable to run the locator with several parameter settings for NA. On the other hand, for events where the user may have a good initial guess for the location (especially for anthropogenic events), the NA search could be safely disabled.

If the instruction *WriteNAResultsToFile* is set to 1, the trial hypocentres together with their misfit and the number of defining phases, as well as the best models after every iteration are written to a file.

2.6 Depth determination

2.6.1 Depth resolution

Solving for depth is attempted by *iLoc* only if the data provide sufficient depth resolution. Depth resolution can be provided by a local network, depth phases, core reflections and to a lesser extent near-regional secondary phases. We have developed a number of criteria to test whether the reported data for an event have sufficient depth resolution.

1. Local network: at least *MinLocalStations* stations with time-defining phases within *MaxLocalDistDeg* epicentral distance.
2. Depth phases: at least *MinDepthPhases* time-defining first-arriving P and depth phase pairs reported by at least *MindDepthPhaseAgencies* agencies (to reduce the chance of misinterpretation by a single inexperienced analyst).
3. Core reflections: at least *MinCorePhases* time-defining first-arriving P and core reflection pairs (PcP, ScS) reported by at least *MindDepthPhaseAgencies* agencies.
4. Local/near regional S: at least *MinSPpairs* time-defining P and S pairs within *MaxSPDistDeg* epicentral distance.

The parameters governing the depth resolution criteria are read from the *Config.txt* file (**Table 1**) or can be given as instructions (**Table 5**). We attempt a free-depth solution if any of the above criteria are satisfied; otherwise we fix the depth to a default depth, dependent on the epicentre location.

2.6.2 Default depth for fixed-depth events

If there is insufficient depth resolution provided by the data, or the depth uncertainty for a free-depth solution exceeds a threshold (**Table 1**), the hypocentre depth is set to the depth from the default depth grid if a grid point for the epicentre location exists; otherwise it is set to the median of reported depths if the initial depth is larger than 100 km; else it is set to a depth (Bolton et al., 2006) assigned to the corresponding Flinn-Engdahl (Young et al., 1996) geographic region.

The default depth grid (Bondár and Storchak, 2011) is defined on a 0.5º x 0.5º grid and was derived from the free-depth solutions obtained from the rebuilt ISC bulletin (Storchak et al., 2017, 2020) as well as the EHB (Engdahl et al., 1998) free-depth solutions. In total, 686,188 events with reliable depths were used to produce the default depth for some 10,000 grid cells.

The default depth in a grid cell is defined as the median of all depths in the cell, provided that there were at least five events in the cell, and the 75–25 percent quartile range was less than 100 km. The latter constraint is imposed to avoid regions with both shallow and deep seismicity.

The default depth grid follows gridline registration, i.e. the nodes are centred on the grid line intersections and the data points represent the median value in a cell of dimensions (*EtopoRes* x *EtopoRes*) centred on the nodes. Therefore, a point (*lat,lon*) falls in a grid cell if the expression below is true.

$$(|lat - grid\_lat_i| \le EtopoRes/2 \text{ and } |lon - grid\_lon_j| \le EtopoRes/2)$$

2.6.3 Depth-phase depth

If the reported depth phases provide sufficient depth resolution, iLoc determines the hypocentre depth using the depth-phase stacking method (Murphy and Barker, 2006). The depth-phase stack provides a depth estimate independent from the one obtained by the location algorithm.

The depth-phase stack is always performed if there is depth resolution provided by the depth phases and the linearised least-squares location algorithm has reached a convergent solution, even if the depth was fixed either by instructions or by the location algorithm.

The depth-phase stacking method can be described in three steps.

1. The predicted moveout curves (depth phase – first arriving P arrival time) are generated for each station as a function of depth (they also depend on the epicentral distance).
2. For each observed moveout a depth trace is generated by putting a boxcar at the corresponding depth. The width of the boxcar is defined by the *a priori* measurement error for the depth phase; it is centred on the observed moveout and then projected to the *x*-axis (depth).
3. The depth traces are stacked and the depth-phase depth and its uncertainty are defined as the median and the standard median absolute deviation (SMAD) of the stack, respectively.

2.6.4 Reporting how the hypocentre depth was obtained

The *dtype* field in the IDC **origin** table, *depfix* in the ISC **hypocenter** table and the *depthType* field in the SeisComp3 **Origin** table describe how the depth was obtained for an event. An explanation is also given in the *iLoc* log file as well as in the ISC bulletin when searched on the ISC website, www.isc.ac.uk. **Table 7** gives the list of possible depth type values and their descriptions.

Table 7. Depth determination

Type	Description
F	Free-depth solution.
A	Depth fixed by analyst.
B	Beyond depth limits; depth fixed to 0/600 km.
G	Depth fixed to default depth grid value.
H	Depth fixed to depth of a reported hypocentre.
M	Depth fixed to median of reported hypocentre depths.
R	Depth fixed to default Flinn-Engdahl region depth value.
S	Anthropogenic event; depth fixed to the surface.

2.7 Linearized least-squares location algorithm

2.7.1 Iterative inversion scheme

An iterative linearized least-squares inversion of travel-times (Bondár and McLaughlin, 2009) is performed by *iLoc* (Bondár and Storchak, 2011) to obtain a solution for the hypocentre. The algorithm solves the matrix equation

$$\mathbf{G}_W\mathbf{m} = \mathbf{W}\mathbf{G}\mathbf{m} = \mathbf{W}\mathbf{d} = \mathbf{d}_W$$

where **G** is the (*NxM*) design matrix containing the partial derivatives of *N* data by *M* model parameters, **m** is the (*Mx1*) model adjustment vector $[\Delta T, \Delta x, \Delta y, \Delta z]^T$, **d** is the (*Nx1*) vector of time residuals. If *DoCorrelatedErrors* (**Table 1**) is false, **W** represents the diagonal weight matrix.

If correlated travel-time errors are accounted for, **W** is the (*NxN*) projection matrix that orthogonalizes the data set. In other words, we solve the inversion problem in the eigen-coordinate system in which the transformed observations are independent, that is, $\mathbf{d}_W$ represents linear combinations of the observed residuals, the "eigen residuals". The equation above is solved by singular value decomposition, which yields the general inverse

$$\mathbf{G}_W^{-1} = \mathbf{V}_W \mathbf{\Lambda}_W^{-1} \mathbf{U}_W^T$$

and the model adjustment of

$$\mathbf{m}_{est} = \mathbf{G}_W^{-1} \mathbf{d}_W$$

After the j^{th} iteration, the model vector is adjusted such that $\mathbf{m}_{j+1} = \mathbf{m}_j + \mathbf{m}_{est}$. Damping can be applied for large condition numbers.

For free-depth solutions the depth is kept fixed for the first *MinIterations* -1 (**Table 1**) iterations, to mitigate the trade-off between depth and epicentre. After each iteration the depth is checked against negative or excessively large values and fixed to 0 or 600 km. The depth remains fixed for the rest of the iterations if the number of occurrences of airquakes/deepquakes exceeds 2.

The $\mathbf{G_w}$ matrix and the $\mathbf{d_w}$ vector are recalculated with respect to the new solution. An observation (arrival time, slowness or azimuth) is made non-defining if its residual exceeds *SigmaThreshold* times the *a priori* measurement error (see **Tables 1** and **3**). Phases are reidentified if the depth crosses the Moho or Conrad discontinuity. Normally the projection matrix needs to be calculated only once. However, if defining phases were renamed or defining phases were made non-defining, the projection matrix is recalculated.

The convergence test is based on the Paige-Saunders convergence test value (Paige and Saunders, 1982) and on the history of model and data norms. The convergence test is applied after every iteration.

2.7.2 Formal uncertainties

Once a convergent solution is obtained, the location uncertainty is defined by the *a posteriori* model covariance matrix

$$\mathbf{C}_M = \mathbf{G}^{-1} \mathbf{C}_D \mathbf{G}^{-1^T} = \mathbf{V}_W \mathbf{\Lambda}_W^{-2} \mathbf{V}_W^T$$

The model covariance matrix yields the four-dimensional error ellipsoid whose projections provide the two-dimensional error ellipse and one-dimensional errors for depth and origin time. The error ellipse encompassing the confidence region at a given α percentile level is defined by

$$(\mathbf{r} - \mathbf{r}_{loc})^T \mathbf{C}_M (\mathbf{r} - \mathbf{r}_{loc}) = \kappa_\alpha^2$$

where $\mathbf{r}_{loc}$ denotes the location vector of the epicentre. We follow Jordan and Sverdrup (1981) to define κ_α^2 as

$$\kappa_\alpha^2 = Ms^2 F_\alpha(M, K + N - M)$$

where the variance scaling factor s^2 is defined as

$$s^2 = \frac{K + \frac{1}{N}\sum d_w^2}{K + N - M}$$

and F_α is an F statistic (Zwillinger and Kokoska, 2000) with M and $K+N-M$ degrees of freedom at the critical level α = *ConfidenceLevel*% (**Table 1**) with M = 2 and with *N independent* observations, that is, the total number of observations less the number of observations projected to the null space. *K* is set to a large value (99999) so that the formal uncertainty estimates approximate "coverage" error ellipses.

2.7.3 Location quality metrics

Besides calculating the formal uncertainties, *iLoc* also computes various location quality metrics (Bondár et al., 2004; Bondár and McLaughlin, 2009) based on the network geometry. The maximal azimuthal gap, the maximal secondary azimuthal gap and the deviation from a uniformly distributed station network are calculated for local, near-regional, teleseismic distance ranges as well as for the entire network.

- Local network: 0 - 150 km
- Near regional: 3 - 10 degrees
- Teleseismic: 28 - 180 degrees
- Entire network: 0 - 180 degrees

The location quality metrics help to assess the accuracy of the hypocentre solution and identify ground truth candidate events.

2.8 Magnitude calculation

Currently *iLoc* calculates body and surface wave magnitudes. MS is calculated for shallow events (depth < *MSMaxDepth*) only. At least *MinNetmagSta* station magnitudes are required for a network magnitude. The network magnitude is defined as the median of the station magnitudes of the same type, and its uncertainty is defined as the standard median absolute deviation (SMAD) of the alpha-trimmed (*alpha* = 20%) station magnitudes.

The station magnitude is defined as the median of reading magnitudes for a station. The reading magnitude is defined as the magnitude computed from the maximal log(A/T) in a reading. Amplitude magnitudes are calculated for each reported amplitude - period pairs.

Note that station, reading, amplitude and MS horizontal and vertical components are always stored in the ISC and the SeisComp3 database schema, even if there were insufficient number of station magnitudes to calculate a network magnitude.

2.8.1 Body-wave magnitudes, mb

Body-wave magnitudes are calculated for each reported amplitude-period pairs, provided that the phase is in the list of phases that can contribute to mb (*MBPhase*), the station is between the epicentral distances *mbMinDistDeg* and *mbMaxDistDeg*, and the period is between *mbMinPeriod* and *mbMaxPeriod*.

A reading contains all parametric data reported by a single agency for an event at a station, and it may have several reported amplitude and periods. For each pair an amplitude mb is calculated.

Amplitude mb = log(A/T) + Q(Δ,h)

If no amplitude-period pairs are reported for a reading, the body-wave magnitude is calculated using the reported logat values.

Amplitude mb = *logat* + Q(Δ,h)

The magnitude attenuation Q(d,h) value is calculated using the Gutenberg-Richter (Gutenberg and Richter, 1956), or optionally, using the Veith-Clawson (Veith and Clawson, 1972) or the Murphy-Barker (Murphy and Barker, 2003) tables. For the Gutenberg-Richter tables amplitudes are measured in micrometers, while for the Veith-Clawson and Murphy-Barker tables amplitudes are measured in nanometers.

$Q(\Delta,h) = Q_{GR}(\Delta,h) - 3$

$Q(\Delta,h) = Q_{VC}(\Delta,h)$

$Q(\Delta,h) = Q_{MB}(\Delta,h)$

For each reading *iLoc* finds the reported amplitude-period pair for which A/T is maximal:

Reading mb = log(max(A/T)) + Q(Δ,h)

Or, if no amplitude-period pairs were reported for the reading:

Reading mb = max(*logat*) + Q(Δ,h)

Several agencies may report data from the same station. The station magnitude is defined as the median of the reading magnitudes for a station.

Station mb = median(Reading mb)

Once all station mb values are determined, the station magnitudes are sorted and the lower and upper α percentiles are made non-defining. The network mb and its uncertainty are then calculated as the median and the standard median absolute deviation (SMAD) of the alpha-trimmed station magnitudes, respectively.

2.8.2 Surface-wave magnitudes, MS

Surface-wave magnitudes are calculated for each reported amplitude-period pairs, provided that the phase is in the list of phases that can contribute to MS (*MSPhase*), the station is between the epicentral distances *MSMinDistDeg* and *MSMaxDistDeg*, and the period is between *MSMinPeriod* and *MSMaxPeriod*.

For each reported amplitude-period pairs MS is calculated using the Prague formula (Vanek et al., 1962); see also the latest IASPEI standards http://www.iaspei.org/commissions/CSOI/Summary_WG_recommendations_20130327.pdf.

Amplitude MS is calculated for each component (Z, E, N) separately, where the amplitude is measured in nanometers.

Amplitude MS = log(A/T) + 1.66 * log(Δ) + 0.3

To calculate the reading MS, *iLoc* first finds the reported amplitude-period pair for which A/T is maximal on the vertical component.

$MS_Z = \log(\max(A_Z/T_Z)) + 1.66 * \log(\Delta) + 0.3$

Then it finds the max(A/T) for the E and N components for which the period measured on the horizontal components is within ± *MSPeriodRange* seconds from the period measured on the vertical component. The horizontal MS is calculated as

$$\max(A/T)_H = \begin{cases} \sqrt{2(\max(A_E/T_E))^2} & \textit{if } MS_N \textit{ does not exist} \\ \sqrt{(\max(A_E/T_E))^2 + (\max(A_N/T_N))^2} & \textit{if } MS_E \textit{ and } MS_N \textit{ exist} \\ \sqrt{2(\max(A_N/T_N))^2} & \textit{if } MS_E \textit{ does not exist} \end{cases}$$

$MS_H = \log(\max(A/T)_H) + 1.66 * \log(\Delta) + 0.3$

The reading MS is defined as

$$\text{Reading } MS = \begin{cases} (MS_Z + MS_H)/2 & \textit{if } MS_Z \textit{ and } MS_H \textit{ exist} \\ MS_H & \textit{if } MS_Z \textit{ does not exist} \\ MS_Z & \textit{if } MS_H \textit{ does not exist} \end{cases}$$

Several agencies may report data from the same station. The station magnitude is defined as the median of the reading magnitudes for a station.

Station MS = median(Reading MS)

Once all station MS values are determined, the station magnitudes are sorted and the lower and upper α percentiles are made non-defining. The network MS and its uncertainty are then calculated as the median and the standard median absolute deviation (SMAD) of the alpha-trimmed station magnitudes, respectively.

2.8.3 Local magnitudes, ML

Local magnitudes are calculated for each reported amplitude-period pairs, provided that the phase is in the list of phases that can contribute to ML (*MLPhase*), the station-event distance is less than *MLMaxDistkm*. For the calculation of local magnitudes *iLoc* uses the SeisComp3 global MLv computation scheme. A magnitude correction value is computed by linear interpolation between three distance ranges as given in **Table 8**.

Table 8. MLv corrections

Distance [km]	Correction	A0
0 - 60	1.3 - 2.8	1.3+0.025x
60 - 400	2.8 - 4.5	2.5+0.005x
400 - 1000	4.5 - 5.85	3.6+0.0225x

A reading contains all parametric data reported by a single agency for an event at a station, and it may have several reported amplitudes. For each amplitude an amplitude ML is calculated.

Amplitude ML = $\log(A)$ + A0

where A0 is calculated according to the formulae in **Table 7**.

For each reading *iLoc* finds the reported maximal amplitude.

Reading ML = $\log(\max(A))$ + A0

Several agencies may report data from the same station. The station magnitude is defined as the median of the reading magnitudes for a station.

Station ML = median(Reading ML)

Once all station ML values are determined, the station magnitudes are sorted and the lower and upper α percentiles are made non-defining. The network ML and its uncertainty are then calculated as the median and the standard median absolute deviation (SMAD) of the alpha-trimmed station magnitudes, respectively. Note that the because ISF files do not contain information on amplitude units, ML computation can be unreliable when using ISF input files.

2.8.1 Broadband body-wave magnitudes, mB

Broadband body-wave magnitudes are defined in http://gfzpublic.gfz-potsdam.de/pubman/item/escidoc:816929 and calculated for each reported mB amplitude, provided that the phase is in the list of phases that can contribute to mb (*MBPhase*), the station is between the epicentral distances *BBmBMinDistDeg* and *BBmBMaxDistDeg*.

A reading contains all parametric data reported by a single agency for an event at a station, and it may have several reported amplitude and periods. For each measurement an amplitude mB is calculated.

$$\text{Amplitude mB} = \log(V_{max}/2\pi) + Q(\Delta,h)$$

The magnitude attenuation Q(*d,h*) value is calculated using the Gutenberg-Richter (Gutenberg and Richter, 1956), or optionally, using the Veith-Clawson (Veith and Clawson, 1972) or the Murphy-Barker (Murphy and Barker, 2003) tables. For the Gutenberg-Richter tables amplitudes are measured in micrometers, while for the Veith-Clawson and Murphy-Barker tables amplitudes are measured in nanometers.

$$Q(\Delta,h) = Q_{GR}(\Delta,h) - 3$$

$$Q(\Delta,h) = Q_{VC}(\Delta,h)$$

$$Q(\Delta,h) = Q_{MB}(\Delta,h)$$

For each reading *iLoc* finds the reported amplitude for which V_{max} is maximal:

$$\text{Reading mB} = \log(\max(V_{max}/2\pi)) + Q(\Delta,h)$$

Several agencies may report data from the same station. The station magnitude is defined as the median of the reading magnitudes for a station.

$$\text{Station mB} = \text{median}(\text{Reading mB})$$

Once all station mB values are determined, the station magnitudes are sorted and the lower and upper α percentiles are made non-defining. The network mB and its uncertainty are then calculated as the median and the standard median absolute deviation (SMAD) of the alpha-trimmed station magnitudes, respectively. Note that the because ISF files do not contain information on amplitude units, mB computation can be unreliable when using ISF input files.

3 Input/output

3.1 Database

3.1.1 SeisComp4 database schema

The locator reads the **Event**, **Origin**, **OriginReference**, **Magnitude**, **Arrival**, **Pick**, **Amplitude**, **Network**, **Station, SensorLocation** and **PublicObject** tables. Note that some of the fields are read from the database only for the sake of producing an optional ISF2 output file. The difference between the SeisComp3 MySQL and the NDC-in-a-Box PostgreSQL database schema is that the field names (except *_oid*) are prepended with 'm_'.

Once the data are read from the database, the phases will be separated into readings and sorted by *delta*, *prista*, *rdid*, and *time*.

Table 7. Data read from SeisComp3 database schema tables

Table	Field	Description
Event	_oid	event id
Event	etype	event type
PublicObject	preferredOriginID	preferred origin id
Origin	_oid	origin id
Origin	creationInfo_agencyID	author
Origin	time_value	origin date-time to the second
Origin	time_value_ms	origin time microseconds part
Origin	latitude_value	event latitude
Origin	longitude_value	event longitude
Origin	depth_value	event depth
Origin	depth_uncertainty	depth uncertainty (fixed if null)
Origin	time_uncertainty	origin time uncertainty
Origin	epicenterFixed	fixed epicentre flag (for ISF output only)
Origin	timeFixed	fixed origin time flag (for ISF output only)
Origin	quality_associatedStationCount	number of stations (for ISF output only)
Origin	quality_usedStationCount	number of defining stations (for ISF output only)
Origin	quality_associatedPhaseCount	number of associated phases (for ISF output only)
Origin	quality_usedPhaseCount	number of defining phases (for ISF output only)
Origin	quality_minimumDistance	closest station distance (for ISF output only)
Origin	quality_maximumDistance	farthest station distance (for ISF output only)
Origin	quality_azimuthalGap	azimuthal gap (for ISF output only)
Origin	quality_secondaryAzimuthalGap	secondary azimuthal gap (for ISF output only)
Origin	uncertainty_minHorizontalUncertainty	semi-minor axis (for ISF output only)
Origin	uncertainty_maxHorizontalUncertainty	semi-major axis (for ISF output only)
Origin	uncertainty_azimuthMaxHorizontalUncertainty	azimuth of semi-major axis (for ISF output only)
Magnitude	magnitude_value	magnitude (for ISF output only)
Magnitude	type	magnitude type (for ISF output only)

Magnitude	magnitude_uncertainty	magnitude uncertainty (for ISF output only)
Magnitude	stationCount	number of station magnitudes (for ISF output only)
Magnitude	creationInfo_agencyID	magnitude author (for ISF output only)
Origin	quality_standardError	standard error of residuals (for ISF output only)
Arrival	_oid	phase id
Arrival	phase_code	phase associated to the preferred hypocentre
Arrival	distance	epicentral distance
Arrival	pickID	pick id
Pick	waveformID_networkCode	station network code
Pick	waveformID_stationCode	station code
Pick	waveformID_locationCode	station location code
Pick	waveformID_channelCode	phase pick channel
Pick	time_value	arrival date-time to the second
Pick	time_value_ms	arrival time microseconds part
Pick	phaseHint_code	reported phase code
Pick	horizontalSlowness_value	slowness
Pick	backazimuth_value	azimuth
Pick	polarity	short period first motion (for ISF output only)
Pick	onset	onset quality (for ISF output only)
Station	latitude	station latitude
Station	longitude	station longitude
Station	elevation	station elevation
Amplitude	_oid	amplitude id
Amplitude	amplitude_value	amplitude
Amplitude	period_value	period
Amplitude	snr	signal-to-noise ratio
Amplitude	waveformID_channelCode	channel of amplitude measurement
Amplitude	amplitude_type	amplitude type

If a convergent solution is reached, the new preferred origin, together with its associations is written to the database. The **Event**, **EventDescription**, **OriginReference**, **Origin**, **Arrival**, **Magnitude**, **StationMagnitude**, and **StationMagnitudeContribution** tables are populated with the new solution. Unique ids are taken from the **Object** and **PublicObject** tables.

If the locator failed to get a convergent solution, or it run with a fixed hypocentre, no data are written to the database.

3.1.2 IDC database schema

The locator reads the **origin**, **origerr**, **netmag**, **assoc**, **arrival**, **amplitude**, and **site** tables. Note that some of the fields are read from the database only for the sake of producing an optional ISF2 output file. There is no difference between the IDC Oracle and the NDC-in-aBox PostgreSQL database schema.

Once the data are read from the database, the phases will be separated into readings and sorted by *delta*, *prista*, *rdid*, and *time*.

Table 8. Data read from IDC database schema tables

Table	Field	Description
origin	evid	event id
origin	orid	origin id
origin	etype	event type
origin	auth	author of origin
origin	time	origin epoch time
origin	lat	event latitude
origin	lon	event longitude
origin	depth	event depth (depdp if depth is null)
origin	dtype	depth type (for ISF output only)
origin	nass	number of associated phases (for ISF output only)
origin	ndef	number of defining phases (for ISF output only)
origerr	sminax	semi-minor axis (for ISF output only)
origerr	smajax	semi-major axis (for ISF output only)
origerr	strike	azimuth of semi-major axis (for ISF output only)
origerr	stime	origin time uncertainty (for ISF output only)
origerr	sdepth	depth uncertainty (for ISF output only)
origerr	sdobs	standard error of residuals (for ISF output only)
netmag	magnitude	magnitude (for ISF output only)
netmag	magtype	magnitude type (for ISF output only)
netmag	uncertainty	magnitude uncertainty (for ISF output only)
netmag	nsta	number of station magnitudes (for ISF output only)
netmag	auth	magnitude author (for ISF output only)
site	lat	station latitude
site	lon	station longitude
site	elev	station elevation
assoc	arid	arrival id associated to the preferred origin id
assoc	sta	station code
assoc	delta	epicentral distance
assoc	phase	phase
arrival	arid	arrival id associated to the preferred origin id
arrival	iphase	reported phase name
arrival	chan	channel of phase arrival time pick
arrival	time	arrival epoch time
arrival	slow	slowness
arrival	azimuth	azimuth
arrival	fm	first motion (for ISF output only)
arrival	qual	arrival quality (for ISF output only)
arrival	auth	arrival author (for ISF output only)
amplitude	ampid	amplitude id
amplitude	amp	amplitude
amplitude	per	period
amplitude	chan	channel of amplitude measurement
amplitude	amptype	amplitude type

If there already existed a preferred origin with *OutAgency* author, *iLoc* will delete the previous solution before writing the new *OutAgency* solution.

If a convergent solution is reached, the new origin, together with its associations is written to the database. The **origin**, **origerr**, **netmag**, **stamag**, and **assoc** tables are populated with the new solution. Unique ids are taken from the **lastid** table.

If the locator failed to get a convergent solution, or it run with a fixed hypocentre, no data are written to the database.

3.1.3 ISC database schema

The locator reads the **event**, **hypocenter**, **hypoc_err**, **hypoc_acc**, **netmag**, **association**, **phase**, **amplitude**, **report** and **site** tables. Note that some of the fields are read from the database only for the sake of producing an optional ISF2 output file.

Note that several important assumptions are made:

- There is a prime (preferred) hypocentre identified by the *prime_hyp* in the **event** table;
- There is an author identified by *InAgency* (**Table 4**) in the **event** table for every physical event.
- The *etype* field in the **event** table describes the preferred event type (if it is null, 'ke' is taken by default) when the author is *InAgency*.
- The author of the phases associated to the prime hypocentre is identified by *InAgency*.

Once the data are read from the database, the reported hypocentres are sorted by their score, the prime hypocentre being the first. A reported phase name may be replaced if it was reidentified by NEIC, CSEM, EHB or IASPEI. The phases are sorted by *delta*, *prista*, *rdid*, and *time*.

Table 8. Data read from database tables

Table	Field	Description
event	prime_hyp	hypocentre id of the prime hypocentre
event	evid	physical event id for the prime hypocentre
event	etype	event type
hypocenter	hypid	hypocentre id
hypocenter	day	origin time to the second
hypocenter	msec	origin time milliseconds
hypocenter	lat	event latitude
hypocenter	lon	event longitude
hypocenter	depth	event depth (depdp if depth is null)
hypocenter	depfix	fixed depth flag (for ISF output only)
hypocenter	epifix	fixed epicentre flag (for ISF output only)
hypocenter	timfix	fixed origin time flag (for ISF output only)
hypocenter	mindist	closest station distance (for ISF output only)
hypocenter	maxdist	farthest station distance (for ISF output only)
hypocenter	azimgap	azimuthal gap (for ISF output only)
hypocenter	author	author of hypocentre report
hypocenter	nsta	number of readings (for ISF output only)

hypocenter	ndefsta	number of defining stations (for ISF output only)
hypocenter	nass	number of associated phases (for ISF output only)
hypocenter	ndef	number of defining phases (for ISF output only)
hypoc_err	sminax	semi-minor axis (for ISF output only)
hypoc_err	smajax	semi-major axis (for ISF output only)
hypoc_err	strike	azimuth of semi-major axis (for ISF output only)
hypoc_err	stime	origin time uncertainty (for ISF output only)
hypoc_err	sdepth	depth uncertainty (for ISF output only)
hypoc_err	sdobs	standard error of residuals (for ISF output only)
hypoc_acc	score	event score (to sort hypocentres)
netmag	magnitude	magnitude (for ISF output only)
netmag	magtype	magnitude type (for ISF output only)
netmag	uncertainty	magnitude uncertainty (for ISF output only)
netmag	nsta	number of station magnitudes (for ISF output only)
netmag	auth	magnitude author (for ISF output only)
site	prista	primary station code (sta if prista is null)
site	lat	station latitude
site	lon	station longitude
site	elev	station elevation
reporter	reporter	reporting agency (phase data)
association	reporter	data report id
association	phase	phase associated to the prime hypocentre, and/or phase associated to NEIC, EHB, IASPEI, CSEM hypocentres
association	delta	epicentral distance
association	phase_fixed	phase name fixed by analyst
association	nondef	phase made non-defining by analyst
phase	rdid	reading id
phase	phid	phase id
phase	sta	station code
phase	phase	reported phase name
phase	day	arrival time to the second
phase	msec	arrival time milliseconds
phase	slow	slowness
phase	azim	azimuth
phase	chan	channel of phase arrival time pick
phase	sp_fm	short period first motion (for ISF output only)
phase	emergent	emergent flag (for ISF output only)
phase	impulsive	impulsive flag (for ISF output only)
amplitude	ampid	amplitude id
amplitude	amp	amplitude
amplitude	per	period
amplitude	logat	log(A/T)
amplitude	chan	channel of amplitude measurement
amplitude	amptype	amplitude type

On output, *iLoc* overwrites the associations that belong to the prime solution.

If a convergent solution is reached, the previous *InAgency* solution, if any, is removed from the database. A new *hypid* is obtained if the previous prime was not *InAgency* and the **event** table is updated accordingly. The **hypocenter**, **hypoc_err**, **hypoc_acc**, **network_quality**, **association**, **netmag**, **stamag**, **readingmag**, **ampmag** and **ms_zh** tables are populated with the new solution.

If the locator failed to get a convergent solution, the **event**, **hypocenter** and **association** tables are reset to the previous prime. If the locator run with a fixed hypocentre, no data are written to the database.

3.2 ISF/ISF2 files

A detailed description of the ISF/ISF2 formats can be found at the ISC website, www.isc.ac.uk. The ISF2 format is an extension of the ISF format that incorporates the new International Seismograph Station Registry station naming conventions, *agency.network.station.location*.

When *iLoc* reads its input from an ISF/ISF2 file, it is assumed that the prime hypocentre is the last one among the hypocentres listed in the origin block of an event. The station coordinates are provided in a separate file, specified by *StationFile* (**Table 1**). The output is written to an ISF2 format file.

4 References

Amante, C. and B.W. Eakins, 2009, ETOPO1 1 Arc-Minute Global Relief Model: Procedures, Data Sources and Analysis. *NOAA Technical Memorandum NESDIS NGDC-24*, 19pp, March 2009.

Begnaud, M.l., S.C. Myers, B. Young, J.R. Hipp, D. Dodge, W.S. Phillips, 2020, Updates to the Regional Seismic Travel Time (RSTT) Model: 1. Tomography, *Pure Appl. Geophys.*, https://doi.org/10.1007/s00024-020-02619-5.

Begnaud, M.l., D.N. Anderson, S.C. Myers, B. Young, J.R. Hipp, W.S. Phillips, 2021, Correction to: Updates to the Regional Seismic Travel Time (RSTT) Model: 2. Path-dependent travel-time uncertainty, *Pure Appl. Geophys.*, https://doi.org/10.1007/s00024-021-02696-0.

Begnaud, M.l., D.N. Anderson, S.C. Myers, B. Young, J.R. Hipp, W.S. Phillips, 2021, Updates to the Regional Seismic Travel Time (RSTT) Model: 2. Path-dependent travel-time uncertainty, *Pure Appl. Geophys.*, **178**, 313-339, https://doi.org/10.1007/s00024-021-02657-7.

Begnaud, M. L., D. N. Anderson, S. C. Myers, B. A. Young, J. R. Hipp, and W. S. Phillips (2021). Updates to the Regional Seismic Travel Time (RSTT) Model: 2. Path-dependent Travel-time Uncertainty, *Pure Appl. Geophys*, pp. 27. doi: 10.1007/s00024-021-02657-7

Bolton, M.K., D.A. Storchak, and J. Harris, 2006, Updating default depth in the ISC bulletin, *Phys. Earth Planet. Int.*, **158**, 27-45.

Bondár, I. P. Mónus, Cs. Czanik, M. Kiszely, Z. Gráczer, Z. Wéber, and the AlpArrayWorking Group (2018). Relocation of Seismicity in the Pannonian Basin Using a Global 3D Velocity Model, *Seism. Res. Let.*, **89**, 2284-2293, doi:10.1785/0220180143, 2018.

Bondár, I., and D. Storchak, 2011, Improved location procedures at the International Seismological Centre, *Geophys. J. Int.*, doi: 10.1111/j.1365-246X.2011.05107.x.

Bondár, I., and K. McLaughlin, 2009, Seismic location bias and uncertainty in the presence of correlated and non-Gaussian travel-time errors, *Bull. Seism. Soc. Am.*, **99**, 172-193.

Bondár, I. and K. McLaughlin, 2009, A new ground truth data set for seismic studies, *Seism. Res. Let.*, **80**, 465-472.

Bondár, I., S.C. Myers, E.R. Engdahl and E.A. Bergman, 2004, Epicenter accuracy based on seismic network criteria, *Geophys. J. Int.*, **156**, 483-496, doi: 10.1111/j.1365-246X.2004.12070.x.

Buland, R. and C.H. Chapman, 1983. The computation of seismic travel times, *Bull. Seism. Soc. Am.*, **73**, 1271-1302.

de Hoon, M.J.L., S. Imoto, J. Nolan and S. Miyano, 2004, Open source clustering software, *Bioinformatics*, **20**, 1453-1454.

Dziewonski, A.M. and F. Gilbert, 1976, The effect of small, aspherical perturbations on travel times and a re-examination of the correction for ellipticity, *Geophys., J. R. Astr. Soc.*, **44**, 7-17.

Engdahl, E.R., R. van der Hilst, and R. Buland, 1998. Global teleseismic earthquake relocation with improved travel times and procedures for depth determination, *Bull. Seism. Soc. Am.,* **88**, 722-743.

Gutenberg, B. and C.F. Richter, 1956, Magnitude and energy of earthquakes, *Ann. Geof.*, **9**, 1-5.

Jordan T.H. and K.A. Sverdrup, 1981, Teleseismic location techniques and their application to earthquake clusters in the South-Central Pacific, *Bull. Seism. Soc. Am.*, **71**, 1105-1130.

Kennett, B.L.N., 2006. Non-linear methods for event location in a global context, *Phys. Earth Planet. Int.*, **158**, 45-64.

Kennett, B.L.N., E.R. Engdahl, and R. Buland, 1995. Constraints on seismic velocities in the Earth from traveltimes, *Geophys. J. Int.*, **122**, 108-124.

Kennett, B.L.N. and O. Gudmundsson, 1996, Ellipticity corrections for seismic phases, *Geophys. J. Int.*, **127**, 40-48.

Murphy J.R. and B.W. Barker, 2003, Revised B(d,h) correction factors for use in estimation of mb magnitudes, *Bull. Seism. Soc. Am.*, **93**, 1746-1764.

Murphy J.R. and B.W. Barker, 2006, Improved focal-depth determination through automated identification of the seismic depth phases pP and sP, *Bull. Seism. Soc. Am.*, **96**, 1213-1229.

Myers, S.C, M.L. Begnaud, S. Ballard, M.E. Pasyanos, W.S. Phillips, A.L. Ramirez, M.S. Antolik, K.D. Hutchenson, J. Dwyer, C. A. Rowe, and G. S. Wagner, 2010, A crust and upper mantle model of Eurasia and North Africa for Pn travel time calculation, *Bull. Seism. Soc. Am.*, **100**, 640-656.

Paige, C. and M. Saunders, 1982, LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares, *ACM Trans. Math. Soft.*, **8**, 43-71.

Sambridge, M., 1999, Geophysical inversion with a neighbourhood algorithm. I. Searching the parameter space, *Geophys. J. Int.*, **138**, 479-494.

Sambridge, M. and B.L.N. Kennett, 2001, Seismic event location: non-linear inversion using a neighbourhood algorithm, *Pageoph*, **158**, 241-257.

Sibson, R., 1973, SLINK: An optimally efficient algorithm for the single-link cluster method. *Comp. J.*, **16**, 30-34.

Storchak, D.A., J. Schweitzer and P. Bormann, 2003. The IASPEI Standard Seismic Phase List, *Seism. Res. Let.*, **74**, 761-772.

Vanek, J., A. Zatopek, V. Karnik, N.V. Kondorskaya, Y.V. Riznichenko, Y.F. Savarensky, S.L. Solovev and N.V. Shebalin, 1962, Standardization of magnitude scales, *Bull. (Izvest.) Acad. Sci. USSR Geophys. Ser.*, **2**, 108-111.

Veith, K.F. and G.E. Clawson, 1972, Magnitude from short-period P-wave data, *Bull. Seism. Soc. Am.*, **62**, 435-452.

Young, J.B., B.W. Presgrave, H. Aichele, D.A. Wiens and E.A. Flinn, 1996. The Flinn-Engdahl regionalization scheme: the 1995 revision, *Phys. Earth Planet. Int.*, **96**, 223-297.

Zwillinger, D. and S. Kokoska, 2000, *CRC Standard Probability and Statistics Tables and Formulae*, Chapman and Hall/CRC.

Appendix A: SeisComp4 Database Schema

A detailed description of the entire SeisComp3 database schema can be found at the SeisComp website, www.seiscomp.org. The SeisComp3 database schema closely follows the QuakeML schema, https://quake.ethz.ch/quakeml/Documents. This document describes only the database tables *iLoc* reads and writes.

In order to maintain the consistency of the database, the unique identifiers assigned to each measurement or computed entities are governed by the **Object** and **PublicObject** tables that exist on the main SeisComp3 account. The main account is specified by the *nextid_db* configuration parameter. The locator obtains the unique identifiers from the **Object** table; some of these identifiers also get public identifiers from the **PublicObject** table.

- **Origin._oid** – origin ids;
- **Magnitude._oid** – network magnitude ids;
- **StationMagnitude._oid** – station magnitude ids;
- **StationMagnitudeContribution._oid** – associated station magnitude ids;
- **Arrival._oid** – arrival ids.

Public identifiers:

- Origin#date_time.oid
- Origin#date_time.oid#netmag.mb
- Origin#date_time.oid#netmag.mB
- Origin#date_time.oid#netmag.MS
- Origin#date_time.oid#netmag.ML
- PickID#ampMag.mb
- PickID#ampMag.MS
- Origin#date_time.oid#rdMag.mb#net.sta
- Origin#date_time.oid#rdMag.MS#net.sta
- Origin#date_time.oid#rdMag.mB#net.sta
- Origin#date_time.oid#rdMag.ML#net.sta
- Origin#date_time.oid#staMag.mb#net.sta
- Origin#date_time.oid#staMag.MS#net.sta
- Origin#date_time.oid#staMag.mB#net.sta
- Origin#date_time.oid#staMag.ML#net.sta

Event table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	[1]
_last_modified	timestamp	modification date	
preferredOriginID	varchar(255)	public id of the preferred origin	points to Origin via PublicObject
preferredMagnitudeID	varchar(255)	public id of the preferred magnitude	points to Magnitude via PublicObject
preferredFocalMechanismID	varchar(255)	public id of the preferred focal mechanism	points to FocalMechanism via PublicObject
type	varchar(64)	event type	[k|s|d|f][e|h|i|m|n|r|x] [ls] [uk]
typeCertainty	varchar(64)	event type uncertainty	
creationInfo_agencyID	varchar(64)	reporting agency	
creationInfo_agencyURI	varchar(255)		
creationInfo_author	varchar(128)	author	
creationInfo_authorURI	varchar(255)		
creationInfo_creationTime	datetime		
creationInfo_creationTime_ms	int(11)		
creationInfo_modificationTime	datetime		
creationInfo_modificationTime_ms	int(11)		
creationInfo_version	varchar(64)		
creationInfo_used	tinyint(1)		

Remarks

- The *etype* field contains the preferred event type.
- The *etype* field is a two letter code where the first letter stands for

  k/s – known/suspected
  f/d – felt/damaging

  and the second letter stands for

  e – earthquake
  h – chemical explosion
  i – induced event, mine collapse
  m – mine explosion
  n – nuclear explosion
  q – quarry blast
  r – rockburst, coalbump
  x – explosion

  exceptions:

  uk – unknown
  ls – landslide

EventDescription table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	Event._oid
_last_modified	timestamp	modification date	
text	varchar(128)	event description	e.g. region name
type	varchar(64)	category	

OriginReference table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	Event._oid
_last_modified	timestamp	modification date	
originID	varchar(255)	public id of origin	points to Origin via PublicObject

Remarks

- The OriginReference table associates hypocenters to an event.

Origin table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	[1]
_last_modified	timestamp	modification date	
time_value	datetime	origin time	
time_value_ms	int(11)	origin time microseconds	
time_uncertainty	double	origin time uncertainty	
time_lowerUncertainty	double		
time_upperUncertainty	double		
time_confidenceLevel	double		
latitude_value	double	latitude	
latitude_uncertainty	double		
latitude_lowerUncertainty	double		
latitude_upperUncertainty	double		
latitude_confidenceLevel	double		
longitude_value	double	longitude	
longitude_uncertainty	double		
longitude_lowerUncertainty	double		
longitude_upperUncertainty	double		
longitude_confidenceLevel	double		
depth_value	double	depth	
depth_uncertainty	double	depth uncertainty	fixed if null
depth_lowerUncertainty	double		
depth_upperUncertainty	double		
depth_confidenceLevel	double		
depth_used	tinyint(1)		
depthType	varchar(64)	type of depth determination	
timeFixed	tinyint(1)	fixed origin time?	
epicenterFixed	tinyint(1)	fixed epicentre?	
referenceSystemID	varchar(255)		
methodID	varchar(255)	locator	iLoc
earthModelID	varchar(255)	velocity model	ak135, RSTT

quality_associatedPhaseCount	int(10)	number of phases	nass
quality_usedPhaseCount	int(10)	number of defining phases	ndef
quality_associatedStationCount	int(10)	number of stations	nsta
quality_usedStationCount	int(10)	number of defining stations	ndefsta
quality_depthPhaseCount	int(10)	number of depth phases	
quality_standardError	double	RMS residual	
quality_azimuthalGap	double		
quality_secondaryAzimuthalGap	double		
quality_groundTruthLevel	varchar(16)	GT level	[0..5]
quality_maximumDistance	double		
quality_minimumDistance	double		
quality_medianDistance	double		
quality_used	tinyint(1)		
uncertainty_horizontalUncertainty	double		
uncertainty_minHorizontalUncertainty	double	semi-minor axis of error ellipse	
uncertainty_maxHorizontalUncertainty	double	semi-major axis of error ellipse	
uncertainty_azimuthMaxHorizontalUncertainty	double	strike of error ellipse	
uncertainty_confidenceEllipsoid_semiMajorAxisLength	double		
uncertainty_confidenceEllipsoid_semiMinorAxisLength	double		
uncertainty_confidenceEllipsoid_semiIntermediateAxisLength	double		
uncertainty_confidenceEllipsoid_majorAxisPlunge	double		
uncertainty_confidenceEllipsoid_majorAxisAzimuth	double		
uncertainty_confidenceEllipsoid_majorAxisRotation	double		
uncertainty_confidenceEllipsoid_used	tinyint(1)		
uncertainty_preferredDescription	varchar(64)		
uncertainty_used	tinyint(1)		
type	varchar(64)		
evaluationMode	varchar(64)		
evaluationStatus	varchar(64)		
creationInfo_agencyID	varchar(64)	reporting agency	
creationInfo_agencyURI	varchar(255)		
creationInfo_author	varchar(128)	author	
creationInfo_authorURI	varchar(255)		
creationInfo_creationTime	datetime		
creationInfo_creationTime_ms	int(11)		
creationInfo_modificationTime	datetime		
creationInfo_modificationTime_ms	int(11)		
creationInfo_version	varchar(64)		
creationInfo_used	tinyint(1)		

Arrival table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	Origin._oid
_last_modified	timestamp	modification date	
pickID	varchar(255)	public id of a pick	points to Pick via PublicObject
phase_code	char(32)	phase	
timeCorrection	double		
azimuth	double	event-to-station azimuth	
distance	double		
takeOffAngle	double		
timeResidual	double		
horizontalSlownessResidual	double		
backazimuthResidual	double		
timeUsed	tinyint(1)		
horizontalSlownessUsed	tinyint(1)		
backazimuthUsed	tinyint(1)		
weight	double		
earthModelID	varchar(255)	velocity model	ak135, RSTT
preliminary	tinyint(1)		
creationInfo_agencyID	varchar(64)	reporting agency	
creationInfo_agencyURI	varchar(255)		
creationInfo_author	varchar(128)	author	
creationInfo_authorURI	varchar(255)		
creationInfo_creationTime	datetime		
creationInfo_creationTime_ms	int(11)		
creationInfo_modificationTime	datetime		
creationInfo_modificationTime_ms	int(11)		
creationInfo_version	varchar(64)		
creationInfo_used	tinyint(1)		

Remarks

- The Arrival table associates picks and amplitudes to a hypocenter.

Pick table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	
_last_modified	timestamp	modification date	
time_value	datetime	arrival time	
time_value_ms	int(11)	arrival time microseconds	
time_uncertainty	double	arrival time uncertainty	
time_lowerUncertainty	double		
time_upperUncertainty	double		
time_confidenceLevel	double		
waveformID_networkCode	char(8)	network	
waveformID_stationCode	char(8)	station	
waveformID_locationCode	char(8)	location	
waveformID_channelCode	char(8)	channel	
waveformID_resourceURI	varchar(255)		
filterID	varchar(255)		
methodID	varchar(255)		
horizontalSlowness_value	double	horizontal slowness	
horizontalSlowness_uncertainty	double		
horizontalSlowness_lowerUncertainty	double		
horizontalSlowness_upperUncertainty	double		
horizontalSlowness_confidenceLevel	double		
horizontalSlowness_used	tinyint(1)		
backazimuth_value	double	event-to-station azimuth	
backazimuth_uncertainty	double		
backazimuth_lowerUncertainty	double		
backazimuth_upperUncertainty	double		
backazimuth_confidenceLevel	double		
backazimuth_used	tinyint(1)		
slownessMethodID	varchar(255)		
onset	varchar(64)	emergent/impulsive	
phaseHint_code	char(32)	reported phase	
phaseHint_used	tinyint(1)		
polarity	varchar(64)	first motion	
evaluationMode	varchar(64)		
evaluationStatus	varchar(64)		
creationInfo_agencyID	varchar(64)	reporting agency	
creationInfo_agencyURI	varchar(255)		
creationInfo_author	varchar(128)	author	
creationInfo_authorURI	varchar(255)		
creationInfo_creationTime	datetime		
creationInfo_creationTime_ms	int(11)		
creationInfo_modificationTime	datetime		
creationInfo_modificationTime_ms	int(11)		
creationInfo_version	varchar(64)		
creationInfo_used	tinyint(1)		

Amplitude table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	
_last_modified	timestamp	modification date	
type	char(16)	amplitude type	
amplitude_value	double	amplitude	
amplitude_uncertainty	double	amplitude uncertainty	
amplitude_lowerUncertainty	double		
amplitude_upperUncertainty	double		
amplitude_confidenceLevel	double		
amplitude_used	tinyint(1)		
timeWindow_reference	datetime		
timeWindow_reference_ms	int(11)		
timeWindow_begin	double		
timeWindow_end	double		
timeWindow_used	tinyint(1)		
period_value	double	period	
period_uncertainty	double	period uncertainty	
period_lowerUncertainty	double		
period_upperUncertainty	double		
period_confidenceLevel	double		
period_used	tinyint(1)		
snr	double	signal-to-noise ratio	
pickID	varchar(255)	public id of a pick	points to Pick via PublicObject
waveformID_networkCode	char(8)	network	
waveformID_stationCode	char(8)	station	
waveformID_locationCode	char(8)	location	
waveformID_channelCode	char(8)	channel	
waveformID_resourceURI	varchar(255)		
filterID	varchar(255)		
methodID	varchar(255)		
scalingTime_value	datetime		
scalingTime_value_ms	int(11)		
scalingTime_uncertainty	double		
scalingTime_lowerUncertainty	double		
scalingTime_upperUncertainty	double		
scalingTime_confidenceLevel	double		
scalingTime _used	tinyint(1)		
magnitudeHint	char(16)	magnitude type	
evaluationMode	varchar(64)		
creationInfo_agencyID	varchar(64)	reporting agency	
creationInfo_agencyURI	varchar(255)		
creationInfo_author	varchar(128)	author	
creationInfo_authorURI	varchar(255)		
creationInfo_creationTime	datetime		
creationInfo_creationTime_ms	int(11)		
creationInfo_modificationTime	datetime		
creationInfo_modificationTime_ms	int(11)		
creationInfo_version	varchar(64)		
creationInfo_used	tinyint(1)		

Magnitude table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	Origin._oid
_last_modified	timestamp	modification date	
magnitude_value	double	network magnitude	
magnitude_uncertainty	double	magnitude uncertainty	
magnitude_lowerUncertainty	double		
magnitude_upperUncertainty	double		
magnitude_confidenceLevel	double		
magnitude_used	tinyint(1)		
type	char(16)	magnitude type	
originID	varchar(255)	public id of the origin	points to Origin via PublicObject
methodID	varchar(255)		
stationCount	int(10)	number of stations	
azimuthalGap	double		
evaluationStatus	varchar(64)		
creationInfo_agencyID	varchar(64)	reporting agency	
creationInfo_agencyURI	varchar(255)		
creationInfo_author	varchar(128)	author	
creationInfo_authorURI	varchar(255)		
creationInfo_creationTime	datetime		
creationInfo_creationTime_ms	int(11)		
creationInfo_modificationTime	datetime		
creationInfo_modificationTime_ms	int(11)		
creationInfo_version	varchar(64)		
creationInfo_used	tinyint(1)		

StationMagnitudeContribution table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	Magnitude_oid
_last_modified	timestamp	modification date	
stationMagnitudeID	varchar(255)	public id of origin	points to StationMagnitude via PublicObject
residual	double	magnitude residual	
weight	double	weight	

Remarks

- The StationMagnitudeContribution table associates station magnitudes to a network magnitude.

StationMagnitude table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	Origin._oid
_last_modified	timestamp	modification date	
originID	varchar(255)	public id of the origin	points to Origin via PublicObject
magnitude_value	double	magnitude	ampmag, rdmag, stamag
magnitude_uncertainty	double	magnitude uncertainty	
magnitude_lowerUncertainty	double		
magnitude_upperUncertainty	double		
magnitude_confidenceLevel	double		
type	char(16)	magnitude type	
amplitudeID	varchar(255)	public ID of the amplitude	points to Amplitude via PublicObject
methodID	varchar(255)		
waveformID_networkCode	char(8)	network	
waveformID_stationCode	char(8)	station	
waveformID_locationCode	char(8)	location	
waveformID_channelCode	char(8)	channel	
waveformID_resourceURI	varchar(255)		
waveformID_used	tinyint(1)		
creationInfo_agencyID	varchar(64)	reporting agency	
creationInfo_agencyURI	varchar(255)		
creationInfo_author	varchar(128)	author	
creationInfo_authorURI	varchar(255)		
creationInfo_creationTime	datetime		
creationInfo_creationTime_ms	int(11)		
creationInfo_modificationTime	datetime		
creationInfo_modificationTime_ms	int(11)		
creationInfo_version	varchar(64)		
creationInfo_used	tinyint(1)		

Remarks

- *iLoc* calculates mb and MS *amplitude magnitudes* from the corresponding amplitude-period pairs, and stores them in the StationMagnitude table with a public id PickID#**ampMag**.magtype.
- *iLoc* calculates mb and MS *reading magnitudes* from the above described amplitude magnitudes for each reading, and stores them in the StationMagnitude table with a public id OriginID#**rdMag**.magtype. The reading magnitude is defined as the amplitude magnitude for which A/T is maximal.
- Finally, *iLoc* calculates mb and MS *station magnitudes* from the above described reading magnitudes for each station, and stores them in the StationMagnitude table with a public id OriginID#**staMag**.magtype. The station magnitude is defined as the median of the reading magnitudes for the same station.

Network table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	4
_last_modified	timestamp	modification date	
code	char(2)	station code	
start	datetime	start of network epoch	
end	datetime	end of network epoch	
description	varchar(80)		
institutions	varchar(100)		
region	varchar(100)		
type	varchar(50)		
netClass	char(1)	permanent/temporary	[p|t]
archive	varchar(20)		
restricted	tinyint(1)		
shared	tinyint(1)		
remark_content	blob		
remark_used	tinyint(1)		

Station table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	Network._oid
_last_modified	timestamp	modification date	
code	char(5)	station code	
start	datetime	start of station epoch	
end	datetime	end of station epoch	
description	varchar(80)		
latitude	double	station latitude	
longitude	double	station longitude	
elevation	double	station elevation	
place	varchar(80)		
country	varchar(50)		
affiliation	varchar(50)		
type	varchar(50)	instrument type	VBB, SP, LP, etc
archive	varchar(20)		
archiveNetworkCode	char(2)		
restricted	tinyint(1)		
shared	tinyint(1)		
remark_content	blob		
remark_used	tinyint(1)		

SensorLocation table

Column	Type	Description	Comment
_oid	int(11)	event id	**primary key**
_parent_oid	int(11)	parent object	Station._oid
_last_modified	timestamp	modification date	
code	char(5)	location code	
start	datetime	start of sensor epoch	
end	datetime	end of sensor epoch	
latitude	double	sensor latitude	
longitude	double	sensor longitude	
elevation	double	sensor elevation	

Appendix B: IDC Database Schema

A detailed description of the entire SeisComp3 database schema can be found at the CTBTO website, http://www.ctbto.org. This document describes only the database tables *iLoc* reads and writes. Note that the **event** table is not populated by the IDC, so it is not listed here.

In order to maintain the consistency of the database, the unique identifiers assigned to each measurement or computed entities are governed by the **lastid** table, that exists on the main account. The main account is specified by the *nextid_db* configuration parameter. The locator obtains the unique identifiers from the **lastid** table.

- **evid** – event ids;
- **orid** – origin ids;
- **magid** –magnitude ids;
- **ampid** –amplitude ids;
- **arid** – arrival ids.

Origin table

Column	Type	Description	Comment
lat	double	latitude	(decimal degrees)
lon	double	longitude	(decimal degrees)
depth	double	depth	(km)
time	double	origin epoch time	(s)
orid	int(10)	origin identifier	**primary key**
evid	int(10)	event identifier	
jdate	int(8)	year and day of year	(yyydoy)
nass	int(4)	number of associated phases	
ndef	int(4)	number of defining phases	
ndp	int(4)	number of depth phases	
grn	int(4)	geographic region number	
srn	int(4)	seismic region number	
etype	varchar(1)	event type	
depdp	double	depth from depth phases	(km)
dtype	varchar(1)	method of depth determination	
mb	double	body wave magnitude	
mbid	int(10)	mb magnitude identifier	
ms	double	surface wave magnitude	
msid	int(10)	Ms magnitude identifier	
ml	double	local magnitude	
mlid	int(10)	ML magnitude identifier	
algorithm	varchar(15)	location algorithm used	
auth	varchar(15)	author	
commid	int(10)	comment identifier	
lddate	datetime	load date	

Remarks

- The *etype* field is a two letter code where the first letter stands for
  k/s – known/suspected
  f/d – felt/damaging
  and the second letter stands for
  e – earthquake
  h – chemical explosion

i – induced event, mine collapse
m – mine explosion
n – nuclear explosion
q – quarry blast
r – rockburst, coalbump
x – explosion

exceptions:

uk – unknown
ls – landslide

- The *dtype* field can be
  F - free-depth solution
  A - depth fixed by the analyst
  S - anthropogenic event; depth fixed to surface
  G - depth fixed to ISC default depth grid
  R - depth fixed to ISC default region depth
  M - depth fixed to median depth of reported hypocentres
  B - beyond depth limits; depth fixed to 0/*max_depth_km*
  H - depth fixed to depth of a reported hypocentre
  D - depth fixed to depth-phase depth

Origerr table

Column	Type	Description	Comment
orid	int(10)	origin identifier	**primary key**
sxx	double	model covariance matrix	
syy	double	model covariance matrix	
szz	double	model covariance matrix	
stt	double	model covariance matrix	
sxy	double	model covariance matrix	
sxz	double	model covariance matrix	
syz	double	model covariance matrix	
stx	double	model covariance matrix	
sty	double	model covariance matrix	
stz	double	model covariance matrix	
sdobs	double	standard error of observations	(s)
smajax	double	semi-major axis of error ellipse	(km)
sminax	double	semi-minor axis of error ellipse	(km)
strike	double	azimuth of semi-major axis	(decimal degrees)
sdepth	double	depth uncertainty	(km)
stime	double	origin time uncertainty	(s)
conf	double	confidence level	
commid	int(10)	comment identifier	
lddate	datetime	load date	

Assoc table

Column	Type	Description	Comment
arid	int(10)	arrival identifier	**primary key**
orid	int(10)	origin identifier	**primary key**
sta	varchar(6)	station code	
phase	varchar(8)	associated phase	
belief	double	phase confidence	
delta	double	station-to-event distance	(decimal degrees)
seaz	double	station-to-event azimuth	(decimal degrees)
esaz	double	event-to-station azimuth	(decimal degrees)
timeres	double	time residual	(s)
timedef	varchar(1)	time defining/nondefining flag	[d|n]
azres	double	azimuth residual	(decimal degrees)
azdef	varchar(1)	azimuth defining/nondefining flag	[d|n]
slores	double	slowness residual	(s/deg)
slodef	varchar(1)	slowness defining/nondefining flag	[d|n]
emares	double	incidence angle residual	(decimal degrees)
wgt	double	location weight	
vmodel	varchar(15)	velocity model	
commid	int(10)	comment identifier	
lddate	datetime	load date	

Arrival table

Column	Type	Description	Comment
sta	varchar(6)	station code	**primary key**
time	double	origin arrival time	**primary key**
arid	int(10)	arrival identifier	**alt primary key**
jdate	int(8)	year and day of year	(yyydoy)
stassid	int(10)	stassoc identifier	
chanid	int(10)	instrument identifier	
chan	varchar(8)	channel code	
iphase	varchar(8)	reported phase	
stype	varchar(1)	signal type	
deltim	double	arrival time uncertainty	(s)
azimuth	double	observed azimuth	(decimal degrees)
delaz	double	azimuth uncertainty	(decimal degrees)
slow	double	observed slowness	(s/deg)
delslo	double	slowness uncertainty	(s/deg)
ema	double	emergence angle	(decimal degrees)
rect	double	rectilinearity	
amp	double	instrument corrected amplitude	(nm)
per	double	instrument corrected period	(s)
logat	double	log(amp/per)	
clip	varchar(1)	clipped flag	
fm	varchar(2)	first motion	
snr	double	signal-to-noise ratio	
qual	varchar(1)	signal onset quality	[e|i|null]
auth	varchar(15)	author	
commid	int(10)	comment identifier	
lddate	datetime	load date	

Amplitude table

Column	Type	Description	Comment
ampid	int(10)	amplitude identifier	**primary key**
arid	int(10)	arrival identifier	
parid	int(10)	predicted arrival identifier	
chan	varchar(8)	channel code	
amp	double	amplitude	(nm)
per	double	period	(s)
snr	double	signal-to-noise ratio	
amptime	double	time of amplitude measurement	(s)
start_time	double	start time of measurement window	(s)
duration	double	duration of measurement window	(s)
bandw	double	bandwidth	
amptype	varchar(8)	amplitude measurement type	
units	varchar(15)	units of measurement	
clip	varchar(1)	clipped flag	
inarrival	varchar(1)	amp is the same as amp in the arrival table	[y/n]
auth	varchar(15)	author	
lddate	datetime	load date	

Stamag table

Column	Type	Description	Comment
magid	int(10)	magnitude identifier	**primary key**
ampid	int(10)	amplitude identifier	**primary key**
sta	varchar(6)	station code	**primary key**
arid	int(10)	arrival identifier	
orid	int(10)	origin identifier	
evid	int(10)	event identifier	
phase	varchar(8)	associated phase	
delta	double	station-to-event distance	(decimal degrees)
magtype	varchar(6)	magnitude type	
magnitude	double	magnitude	
uncertainty	double	magnitude uncertainty	
magres	double	magnitude residual	
magdef	varchar(1)	magnitude defining/nondefining flag	[d|n]
mmodel	varchar(15)	magnitude model	
auth	varchar(15)	author	
commid	int(10)	comment identifier	
lddate	datetime	load date	

Netmag table

Column	Type	Description	Comment
magid	int(10)	magnitude identifier	**primary key**
net	varchar(8)	network identifier	
orid	int(10)	origin identifier	
evid	int(10)	event identifier	
magtype	varchar(6)	magnitude type	
nsta	int(8)	number of stations used	
magnitude	double	magnitude	
uncertainty	double	magnitude uncertainty	
auth	varchar(15)	author	
commid	int(10)	comment identifier	
lddate	datetime	load date	

Site table

Column	Type	Description	Comment
sta	varchar(6)	station code	**primary key**
ondate	int(8)	Julian start date (yyyydoy)	**primary key**
offdate	int(8)	Julian off date	
lat	double	latitude	(decimal degrees)
lon	double	longitude	(decimal degrees)
elev	double	elevation	(km)
staname	varchar(50)	station description	
statype	varchar(4)	station type	[ar|ss]
refsta	varchar(6)	reference station code for array members	
dnorth	double	offset from array reference	(km)
deast	double	offset from array reference	(km)
lddate	datetime	load date	

Appendix C: ISC Database Schema

A detailed description of the entire ISC database schema can be found at the ISC website, www.isc.ac.uk. This document describes only the database tables the ISC locator reads and writes.

With the introduction of the new ISC locator the ISC database schema has undergone some changes:

- new tables (**ampmag**, **readingmag**, **ms_zh**, **magnitude_type**, **network_quality**) were added to the schema;
- new fields were added to some tables (**hypocenter**, **hypoc_err**, **hypoc_acc**, **netmag**, **stamag**, **association**);
- some fields became obsolete (marked as '*not used*').

In order to maintain the consistency of the ISC database, the unique identifiers assigned to each measurement or computed entities are governed by the sequences implemented on the main ISC account. The main account is specified by the *nextid_db* configuration parameter. The locator obtains unique identifiers from the sequences below.

- **hypid** – hypocentre identifier;
- **magid** – network magnitude identifier;
- **stamag_stamagid_seq** – station magnitude identifier;
- **rdmagid** – reading magnitude identifier;
- **ampmagid** – amplitude magnitude identifier;
- **mszhid** – MS vertical/horizontal component magnitude identifier;

event table

Column	Type	Description	Comment
author	varchar(16)	author	
banished	varchar(1)	banished	[B|null]
evid	integer	reported event id	**primary key**
lddate	timestamp	load date	
moddate	timestamp	modification date	
prime_hyp	integer	prime hypocentre id	not null
ready	varchar(1)	ready to publish	[R|T|null]
remid	integer	remark id	
reporter	integer	data report id	
etype	varchar(4)	event type	[k|s|d|f][e|h|i|m|n|r|x] [ls] [uk]

Remarks

- There should be an *in_agency* (**Table 4**) author for every physical event
- The *etype* field contains the preferred event type when author = '<*in_agency*>'.
- The *etype* field is a two letter code where the first letter stands for
  - *k/s* – known/suspected
  - *f/d* – felt/damaging

  and the second letter stands for
  - *e* – earthquake
  - *h* – chemical explosion
  - *i* – induced event, mine collapse
  - *m* – mine explosion
  - *n* – nuclear explosion
  - *q* – quarry blast
  - *r* – rockburst, coalbump
  - *x* – explosion

  exceptions:
  - *uk* – unknown
  - *ls* – landslide
- The *ready* field indicates the status of the analyst review process
  - *T* – waiting for review
  - *R* – reviewed
  - *null* – not reviewed by the ISC

hypocenter table

Column	Type	Description	Comment
accuracy	varchar(10)	location accuracy	(km)
author	varchar(16)	author	
azimgap	numeric(6,3)	max azimuthal gap	
centroid	varchar(1)	centroid hypo	[C|null]
coll_evid	integer	reported event id	
day	timestamp	origin time to the second	
depdp	numeric(8,5)	depth-phase depth	(km)
depfix	varchar(6)	fixed-depth flag	[null|A|B|D|G|H|M|R|S]
deprecated	varchar(1)	deprecated flag	[D|M|P|null]
depth	numeric(8,5)	depth	(km)
epifix	varchar(6)	fixed-epicentre flag	[F|null]
etype	varchar(4)	event type	
grn	integer	geographic region number	
hypid	integer	hypocentre id	**primary key**
isc_evid	integer	physical event id	
lat	numeric(8,5)	latitude	(decimal degrees)
lddate	timestamp	load date	
lon	numeric(8,5)	longitude	(decimal degrees)
magid	integer	magnitude id	(*not used*)
magnitude	mumeric(4,2)	magnitude	(*not used*)
magtype	varchar(2)	magnitude type	(*not used*)
maxdist	numeric(6,3)	max station distance	(decimal degrees)
mindist	numeric(6,3)	min station distance	(decimal degrees)
moddate	timestamp	modification date	
msec	integer	origin time milliseconds	
nass	integer	# of associated phases	
ndef	integer	# of defining phases	
ndefsta	integer	# of defining stations	
ndp	integer	# of depth phases	
nsta	integer	# of readings(!)	NOT # of stations
pref_hypid	integer	preferred hypid for an author	
prime	varchar(1)	prime hypocentre	[P|null]
remid	integer	remark id	
reporter	integer	data report id	
srn	integer	seismic region number	
timfix	varchar(6)	fixed-origin time flag	[F|null]
velo_model	varchar(6)	velocity model	
nrank	integer	# of independent defining phases	

Remarks

- *nsta* is the number of readings (a reading contains all phases reported by a single agency for an event at a single station), not the number of distinct stations.
- The *depfix* field can be

  null - free-depth solution
  A - depth fixed by the analyst
  S - anthropogenic event; depth fixed to surface
  G - depth fixed to ISC default depth grid
  R - depth fixed to ISC default region depth
  M - depth fixed to median depth of reported hypocentres
  B - beyond depth limits; depth fixed to 0/*max_depth_km*
  H - depth fixed to depth of a reported hypocentre

D - depth fixed to depth-phase depth

- *pref_hypid* stands for the preferred hypocentre for multiple reports from the same agency
- The *deprecated* field can be
  D – deprecated, *M* – modified, *P* – preliminary, or null

hypoc_err table

Column	Type	Description	Comment
author	varchar(16)	author	
confidence	varchar(6)	confidence level %	
deprecated	varchar(1)	deprecated flag	[D|M|P|null]
hypid	integer	hypocentre id	**primary key**
lddate	timestamp	load date	
moddate	timestamp	modification date	
remid	integer	remark id	
reporter	integer	data report id	
sdepth	real	depth uncertainty	(km)
sdobs	numeric(8,4)	unweighted RMS	(s)
slat	numeric(8,5)	latitude error	(decimal degree)
slon	numeric(8,5)	longitude error	(decimal degree)
smajax	real	semi-major axis	(km)
sminax	real	semi-minor axis	(km)
stime	numeric(6,3)	origin time uncertainty	(s)
strike	numeric(4,1)	semi-major axis azimuth	(decimal degree)
stt	numeric(12,4)	model covariance matrix	
stx	numeric(12,4)	model covariance matrix	
sty	numeric(12,4)	model covariance matrix	
stz	numeric(12,4)	model covariance matrix	
sxx	numeric(12,4)	model covariance matrix	
sxy	numeric(12,4)	model covariance matrix	
syy	numeric(12,4)	model covariance matrix	
syz	numeric(12,4)	model covariance matrix	
szx	numeric(12,4)	model covariance matrix	
szz	numeric(12,4)	model covariance matrix	
sdepdp	real	depth-phase depth uncertainty	(km)

Remarks

- *smajax, sminax, stime and sdepth* are scaled to the *confidence* percent level

hypoc_acc table

Column	Type	Description	Comment
hypid	integer	hypocentre id	**primary key**
reporter	integer	data report id	
score	real	event score to set prime	
nstaloc	integer	# of defining local stations	local: 0-150 km
gtcand	integer	GT candidate	presumed location accuracy
nsta10	integer	# of stations within 10 km	

network_quality table

Column	Type	Description	Comment
hypid	integer	hypocentre id	not null
reporter	integer	data report id	
type	varchar(10)	distance range	not null
du	numeric(8,5)	deviation from uniformly distributed stations	
gap	numeric(8,5)	largest azimuthal gap	
secondary_gap	numeric(8,5)	largest secondary azimuthal gap	
nsta	integer	# of defining stations	
mindist	numeric(7,3)	min station distance	
maxdist	numeric(7,3)	max station distance	

primary key: (hypid, type)

Remarks

- the network quality metrics are given for various distance ranges.
- the *type* field describes the distance ranges:
  local: 0 - 150 km
  near: 3 - 10 degrees
  tele: 28 - 180 degrees
  whole: 0 - 180 degrees

association table

Column	Type	Description	Comment
author	varchar(16)	author	not null
azimdef	varchar(1)	defining azimuth	[A|null]
azimres	numeric(6,3)	azimuth residual	
delta	numeric(6,3)	epicentral distance	(decimal degrees)
deprecated	varchar(1)	deprecated flag	[D|null]
esaz	numeric(6,3)	event-station azimuth	(decimal degrees)
hypid	integer	hypocentre id	not null
lddate	timestamp	load date	
moddate	timestamp	modification date	
net	varchar(6)	network code	
phase	varchar(8)	IASPEI phase code	
phase_fixed	varchar(1)	fixed phase code	{F|null]; phase cannot be renamed
phid	integer	phase id	not null
remid	integer	remark id	
reporter	integer	data report id	
seaz	numeric(6,3)	station-event azimuth	(decimal degrees)
slowdef	varchar(1)	defining slowness	[S|null]
slowres	numeric(6,3)	slowness residual	(s / degree)
sta	varchar(6)	station code	
timedef	varchar(1)	defining time	[T|null]
timeres	numeric(9,3)	time residual	(s)
weight	numeric(4,3)	weight by locator	1/prior measurement error
nondef	varchar(1)	do not use in location	[U|null]

primary key: (phid, author)

phase table

Column	Type	Description	Comment
ampid	integer	amplitude id	
author	varchar(16)	author	
azim	numeric(5,2)	azimuth	(decimal degrees)
chan	varchar(3)	channel	
day	timestamp	arrival time to the second	
delazim	numeric(5,2)	azimuth picking error	(decimal degrees)
delslow	numeric(5,2)	slowness picking error	(s/degree)
deltime	numeric(5,2)	time picking error	(s)
deprecated	varchar(1)	deprecated flag	[D|M|null]
emergent	varchar(1)	emergent signal	[e|E|null]
impulsive	varchar(1)	impulsive signal	[i|I|null]
init	varchar(1)	first phase in a reading	
lddate	timestamp	load date	
lp_fm	varchar(1)	long period first motion	[c|C|+|d|D|-|null]
moddate	timestamp	modification date	
msec	integer	arrival time milliseconds	(ms)
net	varchar(6)	network code	
phase	varchar(8)	reported phase code	
phid	integer	phase id	**primary key**
pref_rd	integer	preferred reading id	deals with duplicates
rdid	integer	reading id	
remid	integer	remark id	
reporter	integer	data report id	
slow	numeric(9,2)	slowness	(s/degree)
sp_fm	varchar(1)	short period first motion	[c|C|+|d|D|-|null]
sta	varchar(6)	station code	

amplitude table

Column	Type	Description	Comment
amp	numeric(12,1)	amplitude	(nm)
ampid	integer	amplitude id	**primary key**
amptype	varchar(8)	amplitude type	[0-to-p|p-to-p]
author	varchar(16)	author	
chan	varchar(3)	channel	
day	timestamp	amplitude time to the second	
delamp	numeric(10,1)	amplitude picking error	(nm)
delper	integer	period picking error	(s)
deprecated	varchar(1)	deprecated flag	[D|M|null]
factor	integer	exponent for moment	
lddate	timestamp	load date	
logat	numeric(9,2)	log(A/T)	
moddate	timestamp	modification date	
moment	numeric(8,5)	scalar moment mantissa	
msec	integer	amplitude time milliseconds	(ms)
per	numeric(6,2)	period	(s)
phid	integer	phase id	not null
reporter	integer	data report id	

netmag table

Column	Type	Description	Comment
author	varchar(16)	author	
deprecated	varchar(1)	deprecated flag	[D|M|P|null]
hypid	integer	hypocentre id	not null
lddate	timestamp	load date	
magid	integer	magnitude id	**primary key**
magnitude	numeric(4,2)	magnitude value	
magtype	varchar(6)	magnitude type	
moddate	timestamp	modification date	
nsta	integer	# of stations	
remid	integer	remark id	
reporter	integer	data report id	
uncertainty	numeric(4,2)	magnitude error	
nagency	integer	# number of agencies	

stamag table

Column	Type	Description	Comment
ampid	integer	amplitude id	(*not used*)
author	varchar(16)	author	
deprecated	varchar(1)	deprecated flag	[D|M|null]
hypid	integer	hypocentre id	not null
lddate	timestamp	load date	
magid	integer	magnitude id	
magnitude	numeric(4,2)	magnitude value	
magtype	varchar(6)	magnitude type	
moddate	timestamp	modification date	
phase	varchar(8)	phase code	(*not used*)
phid	integer	phase id	(*not used*)
remid	integer	remark id	
reporter	integer	data report id	
uncertainty	numeric(4,2)	magnitude error	(*not used*)
stamagid	integer	station magnitude id	**primary key**
magdef	varchar(1)	defining magnitude	[D|null]
rdid	integer	reading id	
sta	varchar(6)	station code	

Remarks

- **stamag** stores the station magnitudes calculated from the reading magnitudes.
- the *magdef* field indicates whether the station magnitude contributed to the calculation of network magnitude uncertainty. It is set to 1 if the *magnitude* is between the 20^{th} and 80^{th} percentile ranges, otherwise to 0.
- **stamag** is populated even if there are an insufficient number of stations to calculate a network magnitude. In that case *magid* is null.

magnitude_type table

Column	Type	Description	Comment
mtypeid	integer	magnitude type id	**primary key**
mtype	varchar(20)	magnitude type	
description	varchar(50)	description	

Remarks

- **magnitude_type** lists the IASPEI and reported magnitude types.

readingmag table

Column	Type	Description	Comment
rdmagid	integer	reading magnitude id	**primary key**
hypid	integer	hypocentre id	not null
magid	integer	magnitude id	
repid	integer	data report id	
rdid	integer	reading id	not null
sta	varchar(6)	station code	
mtypeid	integer	magnitude type id	not null
magnitude	numeric(4,2)	magnitude value	
magdef	integer	defining magnitude	[0|1]
author	varchar(16)	author	

Remarks

- **readingmag** stores the magnitudes calculated for a reading from individual amplitude reports.
- the *magdef* field indicates whether the reading magnitude contributed to the station magnitude. It is set to 1 for the median magnitude, otherwise to 0.
- **readingmag** is populated even if there is an insufficient number of stations to calculate a network magnitude. In that case *magid* is null.

ampmag table

Column	Type	Description	Comment
ampmagid	integer	amplitude id	**primary key**
ampid	integer	amplitude id	not null
hypid	integer	hypocentre id	not null
repid	integer	data report id	
rdid	integer	reading id	
mtypeid	integer	magnitude type id	not null
magnitude	numeric(4,2)	magnitude value	
ampdef	integer	defining amplitude	[0|1]
author	varchar(16)	author	

Remarks

- **ampmag** stores the magnitudes calculated from individual amplitude reports.
- the *ampdef* field indicates whether the amplitude contributed to the reading magnitude. It is set to 1 for the magnitude with max(A/T), otherwise to 0.
- **ampmag** is populated even if there is an insufficient number of stations to calculate a network magnitude. In that case *magid* is null.

ms_zh table

Column	Type	Description	Comment
mszhid	integer	mszh id	**primary key**
repid	integer	data report id	
rdid	integer	reading id	not null
msz	numeric(4,2)	magnitude	MS on vertical component
mszdef	integer	defining MS(z)	[0|1]
msh	numeric(4,2)	magnitude	MS on horizontal component
mshdef	integer	defining MS(h)	[0|1]
hypid	integer	hypocentre id	not null
author	varchar(16)	author	

Remarks

- **ms_zh** stores the MS values calculated on horizontal and vertical components.
- the *mszdef* and *mshdef* fields indicate whether the component magnitude reading magnitude contributed to the reading MS magnitude.
- **ms_zh** is populated even if there is an insufficient number of stations to calculate a network magnitude. In that case *magid* is null.

report table

Column	Type	Description	Comment
collector	varchar(8)	ISC personnel who parsed the report	
dirname	varchar(40)	location of data report file	
filename	varchar(20)	data report filename	
first_hyp	timestamp	time of first hypocentre in report	
first_phase	timestamp	time of first phase arrival in report	
hyp_count	integer	# of hypocentres in report	
hyp_max_lat	numeric(8,5)	extremes of hypocentres in report	
hyp_max_lon	numeric(8,5)	extremes of hypocentres in report	
hyp_min_lat	numeric(8,5)	extremes of hypocentres in report	
hyp_min_lon	numeric(8,5)	extremes of hypocentres in report	
last_hyp	timestamp	time of last hypocentre in report	
last_phase	timestamp	time of last phase in report	
lddate	timestamp	load date	
moddate	timestamp	modification date	
phase_time_precision	integer	precision code	[0|1|2|3|null]
product	varchar(20)	data report type (bulletin, catalog, &c)	
repid	integer	data report id	**primary key**
replaceable	varchar(1)		*not used*
reporter	varchar(8)	agency	
reporter_id	varchar(24)	additional notes on reporter	
status	integer	status of data	
sta_count	integer	# of stations in report	
sta_max_lat	numeric(8,5)	extremes of station locations in report	
sta_max_lon	numeric(8,5)	extremes of station locations in report	
sta_min_lat	numeric(8,5)	extremes of station locations in report	
sta_min_lon	numeric(8,5)	extremes of station locations in report	
issue_date	timestamp	time data were parsed	
parser	varchar(30)	program used to parse data to DB	

Remarks

- *repid* called *reporter* in the other tables

site table

Column	Type	Description	Comment
author	varchar(16)	author	
close_date	timestamp	station close date	
country	varchar(40)		
depth	numeric(5,1)		
digital	varchar(1)	digital station	[0/1]
elev	numeric(8,1)	elevation above seal level	(m)
lat	numeric(8,5)	latitude	(decimal degrees)
lddate	timestamp	load date	
lon	numeric(8,5)	longitude	(decimal degrees)
moddate	timestamp	modification date	
net	varchar(6)	network	
open_date	timestamp	station open date	
prinet	varchar(6)	primary network code	
prista	varchar(6)	primary station code	to deal with alternate codes
region	varchar(80)	station specific region info	
remid	integer	remark id	
reporter	integer	data report id	
sta	varchar(6)	station code	
staname	varchar(80)	station name	
statype	varchar(8)		*not used*
zone	varchar(30)	station specific zone info	
siteid	integer	station id	**primary key**

Appendix D: Data files

D1. General configuration parameters

The general *iLoc* configuration parameters are read from *$ILOCROOT/auxdata/iLocpars/Config.txt* file. The format is expected as **name = value** pairs. The description of the configuration parameters is given in **Table 1**.

D2. Phase and velocity model specific configuration parameters

These *iLoc* configuration parameters are read from the *$ILOCROOT/auxdata/iLocpars/PhaseConfig.txt* file. The description of the configuration parameters is given in **Table 2**.

D3. ak135 ellipticity correction coefficients

The *ak135* ellipticity correction parameters (Kennett and Gudmundsson, 1996) are specified in the *$ILOCROOT/auxdata/ak135/ELCOR.dat* file. For each phase a block of tau coefficients (Dziewonski and Gilbert, 1976) is given in a format:

> phase name, number of distance samples, min dist, max dist
> distance (deg)
> tau0 (at 0, 100, 200, 300, 500, 700 km)
> tau1 (at 0, 100, 200, 300, 500, 700 km)
> tau2 (at 0, 100, 200, 300, 500, 700 km)

D4. ak135 travel-time tables

The *ak135* travel-time tables (Kennett et al., 1995) are listed in the *$ILOCROOT/auxdata/ak135/ak135.<phase>.tab* files. Note that because the Mac OS X file system (HFS+) is case-insensitive, that is, it cannot distinguish between ak135.pP.tab and ak135.PP.tab, the depth phases are prefixed with 'little' so that ak135.pP.tab becomes ak135.littlepP.tab. The travel-time tables follow the format:

> number of distance and depth samples
> delta samples (max 25 in a line)
> depth samples (one line)
> TT table   (rows - delta, columns - depth)
> dtdd table (rows - delta, columns - depth)
> dtdh table (rows - delta, columns - depth)
> bounce point distance table (rows - delta, columns - depth) if depth phase

D5. Iasp91 travel-time tables

The *iasp91* travel-time tables (Kennett and Engdahl, 1991) are listed in the *$ILOCROOT/auxdata/iasp91/iasp91.<phase>.tab* files. The travel-time tables follow the same format and naming conventions as those for the *ak135* travel-time table files.

D6. RSTT model

The RSTT model (Myers et al., 2010) is located in the *$ILOCROOT/auxdata/RSTTmodels* directory.

D7. Flinn-Engdahl regionalization

The Flinn-Engdahl regionalization scheme, 1995 version (Young et al., 1996) is given in the *$ILOCROOT/auxdata/FlinnEngdahl/FE.dat* file. The file follows the format:

> For each latitude (from 90N to 90S) a set of longitude ranges is given (first part of the file).
> The second part of the file lists the geographic region numbers for each latitude within the corresponding longitude ranges.

D8. Default depth grid

The 0.5º x 0.5º default depth grid (Bondár and Storchak, 2011) is given in the *$ILOCROOT/auxdata/FlinnEngdahl/DefaultDepth0.5.grid* file.

Columns:

- lat, lon: center of the grid cell
- depth: median depth in the cell
- min: minimum depth in the cell
- 25Q: 25th percentile depth in the cell
- 75Q: 75th percentile depth in the cell
- max: maximum depth in the cell
- N: number of observations in the cell
- range: quartile range (75Q - 25Q)

Rows are ordered by descending latitude and increasing longitude.

D9. Default depths for Flinn-Engdahl regions

The default depths for each Flinn-Engdahl region (Bolton et al., 2006) are given in the *$ILOCROOT/auxdata/FlinnEngdahl/GRNDefaultDepth.ak135.dat* and the *$ILOCROOT/auxdata/FlinnEngdahl/GRNDefaultDepth.jb.dat* files.

Columns:

- grn: Flinn-Engdahl geographic region number
- depth: depth for GRN

D10. ETOPO topography/bathymetry

The ETOPO1 bedrock topography (Amante and Eakins, 2009) resampled to 5' x 5' resolution is given in the *$ILOCROOT/auxdata/topo/etopo5_bed_g.i2.bin* binary file.

```
grdfilter -I5m etopo1_bed.grd -Fg15 -D4 -Getopo5_bed.grd
        Gridline node registration used
        x_min: -180 x_max: 180 x_inc: 0.0833333 nx: 4321
```

```
y_min: -90 y_max: 90 y_inc: 0.0833333 ny: 2161
z_min: -10515.5 z_max: 6917.75 name: m
scale_factor: 1 add_offset: 0
```

The ETOPO parameters are specified in the *Config.txt* (**Table 4**) file.
Format:
EtopoNlon EtopoNlat2*-byte integers.

D11. Generic variogram model

The generic P variogram model (Bondár and McLaughlin, 2009) is given in the *$ILOCROOT/auxdata/variogram/variogram.model* file. The file follows the format:

number of samples
sill
max station separation in km
station separation (km), variance (s^2)

D12. Magnitude attenuation correction tables

The Gutenberg-Richter (Gutenberg and Richter, 1956), the Veith-Clawson (Veith and Clawson, 1972) and the Murphy-Barker (Murphy and Barker, 2003) magnitude attenuation tables are given in the *$ILOCROOT/auxdata/magnitude/GRmbQ.dat,* the *$ILOCROOT/etc/magnitude/VCmbQ.dat* and the *$ILOCROOT/auxdata/magnitude/MBmbQ.dat* files. The files follow the format:

number of distance samples
delta samples
number of depth samples
depth samples
number of distance samples, number of depth samples
magnitude attenuation table Q(d, h)

D13. Station lists

The comma-separated list of registered stations is given in the *$ILOCROOT/auxdata/Stations/isc_stalist* file. This file is regularly updated at the ISC website (www.isc.ac.uk).

Columns:
sta, altsta, lat, lon, elevation
where sta and alsta are the station code and the alternative station code; lat and lon are the station coordinates in decimal degrees; elevation is the station elevation above sea level in meters.

The station list that complies with the new standards of the International Registry of Seismographic Stations is given in the *$ILOCROOT/etc/Stations/IR2_stalist* file.

Columns:

fdsn.network.sta.location:latitude longitude elevation
* <EOE>

D14. Local velocity models

The locator can optionally use a local velocity model to calculate predicted travel times up to *MaxLocalTTDelta* distances. The parameter *MaxLocalTTDelta* is set in the *Config.txt* file. Predicted travel times are computed for the phases Pg, Pb, Pn, P, Sg/Lg, Sb, Sn, S.

The local velocity model can be either specified in the file given by the *LocalVmodelFile* parameter in the *Config.txt* file, or can be extracted from the RSTT model itself using the coordinates (lat, lon) of the initial hypocenter. In the latter case the *LocalTTfromRSTT* parameter in the Config.txt file should be set to 1.

The parameters controlling the use of a local velocity model (*LocalVmodelFile*, *LocalTTfromRSTT* and *MaxLocalTTDelta*) can also be set as a command line instruction.

Examples for the format of the local velocity model file are given below. The first non-comment line is expected to be the number of layer boundaries in the model, followed by the description of each layer in the model, i.e. the depth of the top of the layer, its P and S velocity and an indicator for the Conrad or Moho discontinuities.

```
#
# ak135
#    LAYER   DEPTH   VP    VS
#
# number of layers
4
     0.000    5.8000    3.4600 x
    20.000    6.5000    3.8500 CONRAD
    35.000    8.0400    4.4800 MOHO
    77.500    8.0400    4.4800 x

#
# Local velocity model from RSTT at (47.46, 18.36)
#    LAYER   DEPTH   VP    VS
#
7
  0.000 2.513 1.193 x
  0.937 4.624 2.536 x
  1.162 5.781 3.300 x
 10.504 5.808 3.466 x
 19.843 6.634 3.580 CONRAD
 29.185 7.889 4.516 MOHO
400.259 8.813 4.513 x
```