

# MNF Native Format for *mloc*

Eric Bergman  
Global Seismological Services  
November 30, 2017  
v1.3.3

This document describes the native data format for use in the multiple event relocation program *mloc*. Data files with this format have the filename suffix ".mnf" and the format will be referred to as MNF for "*mloc* native format". This document describes v1.3.3 of the MNF format, a minor change from v1.3.2, adding a record type to explicitly carry event ID information. The first version of *mloc* to fully support this file format is v10.3.4 with a release date of October 11, 2017, but earlier versions of *mloc* should be able to process event files in this format. A warning would be issued on encountering the unknown record type.

## Changes from v1.3.2

- A new record type is defined, the **ID record** (for event ID or evid), denoted by "I" in column 1. It became desirable to be able to carry more than one event ID, since there is no universal registry. The event ID field has been removed from **event records**.
- The version field of format record is now expected to have the full version number (e.g., "1.3.3") rather than the abbreviated "1.3" used previously.

## Changes from v1.3.1

- The field for event ID (evid) in the **event record** has been expanded from 10 characters to 40 characters, to accommodate current practice at NEIC which features no limit on the length of event identifiers. The content of the field is still right-justified in column 121. The length of the event annotation field is reduced correspondingly.

## Changes from v1.3

- Focal depth is no longer a required field in the **hypocenter record**. There are cases, e.g., macroseismic locations, for which no estimate of depth is provided (technically, it becomes an "epicenter" record).

## Changes from v1.2

- Moved the event ID field to the end of the **event record**, and left the interior of the record for use as an 'annotation' so it can carry the names that are commonly assigned to large or significant earthquakes.
- Added a single-character depth code field to the **hypocenter record** and adjusted positions of the depth uncertainty fields.
- Added the **depth record**, denoted by "D" in column 1.
- Changed the format of the **magnitude record** to allow a larger field for author and comments, in order to accommodate reference-style authorship, e.g., "Baker et al. (1993)". Put the magnitude field in front of the magnitude type field.

## Changes from v1.1

- The **hypocenter record** format has been changed to accommodate asymmetric uncertainties in focal depth.
- A field for GTCU (a new nomenclature I designed to carry information about calibrated events—it supersedes the old GTX formulation) is added to the **hypocenter record** (a4).
- The length of the field for cluster ID in a **hypocenter record** is expanded (first to 20 characters, then back to 18), to accommodate *mloc* cluster names (and series numbers).

## Background

In its early development in the late 1980s, *mloc* used a data format inherited from Ken Creager's LOC program, on which *mloc* was based. In the late 1990s, when *mloc* began to be heavily developed for application in research on calibrated ("ground truth") locations in close collaboration with E. R. Engdahl, the ISC's 96-byte fixed format (FFB), on which Engdahl had standardized his processing, was adopted as the default data format for *mloc*. The FFB format had to be extended in certain ways to meet the needs of *mloc* (as well as Engdahl's codes), and it was always an awkward format, particularly in the fact that very different formats are used for initial and secondary phase readings, on the need to break up station codes longer than 4 characters, and in the strict requirements on ordering of records that is imposed by the concept of carrying in each record the record type for the following record.

At this time, as support for FFB is being discontinued by the ISC, it has become necessary and desirable to devise a format that specifically meets the requirements of *mloc* and which is more compatible with current practice in seismological data formats and exchange mechanisms. This results in data files that are reasonably compact and easy to process and edit as required by the typical processing strategies that have proven to be effective with *mloc*. Data files with this format have the filename suffix ".mnf" and the format will be referred to as MNF for "*mloc* native format".

Starting with version 9.8.0, *mloc* supports only MNF format. Starting with version 10.0.5, *mloc* drops support for versions of MNF format before v1.3.

## Description and Features

MNF is a fixed format based on the common concept of a small set of distinct record types with different formats, identified by a single-character flag in the first column. Each record is a single line; each record type has a minimum line length determined by the defined fields (even if they are not all required). There are minimal constraints on the order in which different record types may occur. Each type of information (e.g., event, hypocenter, depth, magnitude, phase arrival) has a specific record type, and there are a few utility record types. The format can be used to present data for a single event, an earthquake catalog (multiple events, hypocenters only) or a seismic bulletin (multiple events, including phase readings). Events with phase data can be intermixed with events lacking phase data. The **bulletin record** "B" is expected to be the first record of all multi-event data files and it should not appear in single-event files. The currently-defined record types, their flags, their minimum (i.e., required fields) and full (i.e., defined fields) line lengths and their normal order of appearance are given in the following table:

Flag	Record Type	Minimum Length	Full Length
B	Bulletin	1	121
F	Format version	15	15
E	Event	1	121
I	ID	1	51
H	Hypocenter	74	121
D	Depth	9	121
M	Magnitude	8	121
P	Phase reading	55	121
#	Comment	1	121
S	Stop event	1	4
EOF	End of file	3	3

Unlike all other record types, which are distinguished by the flag in column 1, the **end-of-file record** (“EOF”) uses columns 1-3; it has no other arguments. It causes processing of a data file to end, so it would normally only be found once, at the end of the file, whether it holds a single event or multiple events.

The ‘natural’ line length for MNF files is 121 characters, because most of the information-carrying record types have fields defined to this length.

Data for a single event is carried in a block of records that must start with an **event record** “E” and end with a **stop record** “S”. Within the block, data is carried in a combination of **hypocenter** “H”, **depth** “D”, **magnitude** “M”, and **phase reading** “P” records. At least one **hypocenter record** “H” is required, but multiple estimates of hypocenter are permitted. **ID**, **magnitude** and **depth records** are optional and there is no limit to how many can be supplied of each. **Phase reading records** are also optional, in the case of an earthquake catalog. The MNF format can represent a catalog but it is not designed to do so in an efficient manner, since it requires a minimum of three lines (an **event record**, at least one **hypocenter record**, and a **stop record**) for each event.

Some of the fields in an MNF-formatted dataset are not required for *mloc* processing, but they carry information which is often important to retain for interpretation of results and for maintaining compatibility of data products with the NEIC and ISC and other agencies that have standardized on formats such as the IASPEI ISF format for data exchange. Conversely, MNF does not carry some (actually, many) fields which would be considered essential for a general-purpose seismic bulletin format such as ISF, because it is optimized for use in relocation studies using *mloc*.

Several record types carry a "usage" flag (always in column 3) that determines the way (or if) the information in that record will be used. Some records have an "ID" field to carry an ID number that may have been assigned elsewhere, typically in a relational database (e.g., EvID, OrID,

ArrID). Event IDs have their own record type. **Comment records** (flag “#”) are supported; they can be inserted anywhere in an MNF-formatted file. Obviously, any text that occurs after the first **EOF** record will not be processed and can be considered as a comment, regardless of the formatting, but the use of the **EOF** record in this manner is not recommended.

Except for the requirement to start each event block with an **event record** and end the block with a **stop record**, there are few requirements on the order of records within an event block. In most cases the recommended order would be **event record**, **id record(s)**, **hypocenter record(s)**, **depth record(s)**, **magnitude record(s)**, **phase reading records**, followed by a **stop record**.

There must be at least one **hypocenter record**, but multiple hypocenter estimates can be carried. The usage flag should be used to determine which of several **hypocenter records** will be honored for the starting location in *mloc* (or is otherwise the “preferred” location); otherwise precedence will be determined by the software reading the file, probably either the first **hypocenter record** encountered or the last. The same principle applies to **depth** and **magnitude** records: the usage flag should be used to specify a preferred value, if there is one, rather than relying on the sequence.

## Format Versions

Each data file, whether for a single event or multiple events, should contain a **format version record** (“F”) before the first **event record** (“E”). The **format version record** provides a version number for the MNF format in which the file is written. A program that reads an MNF file should check the version number to be sure it will correctly interpret the data records. A bulletin or catalog made up of event files written in different MNF format versions could be processed by including the necessary **format version records** (“F”) when the format changes for the next event, but this is not recommended.

Starting with MNF v1.3.3 and *mloc* v10.3.4 (October 11, 2017 release), format versions are expected to be complete, i.e., “1.3.3” rather than “1.3” which was adequate for previous versions of *mloc*.

## Depths

The **hypocenter record** normally (but not always) carries an estimate of focal depth, but there may be additional estimates of depth that should be carried. The **depth record** is intended to carry information on depth that is considered credible. The most common source of such depth estimates is waveform analysis of some kind. Multiple depth records are permitted and one of them can be designated as preferred by the usage flag ‘=’ in column 3. Depth records can carry an optional depth code flag which *mloc* uses to keep track of the nature of depth constraint. It is highly recommended to use the standard flags (defined below). For example, the focal depth in the preferred **hypocenter record** will be taken as the preferred depth by default in *mloc*, but that will be over-ridden if a following **depth record** meets these requirements:

- The depth record has been designated as preferred by the usage flag “=” in column 3
- The depth record carries a depth code that is considered “constrained”

- The hypocenter record's depth estimate does not carry a depth code that is considered to be "constrained"

Like the **hypocenter record**, a **depth record** can carry an asymmetric estimate of uncertainty. The depth uncertainties are optional. Focal depth and both uncertainties can be given to a tenth of a kilometer, but the decimal point should be present even if the value is only given to the nearest kilometer, to ensure correct reading by Fortran formatted read statements. The authorship field can be used for comments about the nature of the depth estimate. There is no concept of authorID for depth.

## Magnitudes

Magnitude estimates are carried in a **magnitude record**, one magnitude estimate per record. Multiple magnitude estimates can be carried. A usage flag can be used to select a preferred estimate from among multiple records, but is not required. If no magnitude record is marked as preferred, *mloc* uses the first one encountered as a measure of magnitude. In cases where it is desired to carry a magnitude associated with a specific reading, a **magnitude record** could be interspersed with **phase reading records**; in any case the author field could be used to indicate the desired association. Magnitudes can be carried to two decimal places. In any case the decimal point should always be given, to ensure correct reading by Fortran formatted read statements.

## Station Codes

The concept of station codes is evolving away from the traditional strategy of attempting to carry all necessary information about a seismic station (plus the installed instrumentation and who operates it) in a single code of 4 or 5 characters. The MNF format implements the New IASPEI Station Coding Standard:

Agency.Deployment.Station.Location.Channel

or "ADSLC" formulation, using the fixed format display standard described in the defining documentation <[http://www.isc.ac.uk/registries/download/IR\\_implementation.pdf](http://www.isc.ac.uk/registries/download/IR_implementation.pdf)>. In this format, what used to be known as the "station code" is carried in 3-5 characters. In *mloc*, the variable that holds the station code is declared with 6 characters, so the 6th character is available for specialized purposes. The main such purpose is to resolve station naming conflicts, in which data may be obtained from a station which has not been registered for international data exchange at the ISC, and furthermore the code selected by the station operator conflicts with a code that has been registered for a different station. This causes a problem if both stations appear in a data set. For this reason, the MNF format has two fields for the station code: once as a 6 character field that is read by *mloc*, and optionally, as a 5-character subfield of the ADSLC formulation. This latter instance of the station code is not used by *mloc*, but it's useful to carry in the MNF format for forensic purposes.

## Phase Names

Phase name is carried twice in the MNF format. The first instance (columns 24:31 of a "P" record, see below) is read by *mloc*, and then it is subject to the various procedures within *mloc* that may change the syntax of the phase name or change the phase ID completely. The second instance (columns 66:73) carries the phase name as reported by the original source. It is

sometimes useful to change the input phase name from the original phase name, to assist *mloc*'s phase identification algorithm in determining the correct (or at least the desired) phase identification. It is also useful to retain the original phase name unchanged for reference. The MNF format also carries a position for a special flag (“!”) which informs *mloc* that the input phase name should not be changed during relocation.

## ID Numbers

For informational and forensic purposes, MNF includes fields for identification numbers for various kinds of data that are provided by seismological centers. The only use that *mloc* makes of such information is to use event IDs (evids), if provided, to correctly match events with the relocation results of a previous run, as carried in an hdf-formatted file.

Conventionally, ID numbers assigned to events (“evid”), hypocenters (“orid”), phase readings (“arrid”) and other kinds of data in relational databases are integer numbers. Standards on how many digits are carried vary from system to system, but 10-digit integers are likely to be necessary before long, especially for arrids. The MNF format does not enforce a data type on those IDs; they are character fields as far as *mloc* is concerned. Even so, arrids and orrids should be right-justified to aid in correct reading of integers by a Fortran code. Although 10-character arrids are specified in MNF it has proven necessary to provide larger fields for evids and orids.

The larger field for evids is driven by NEIC, where current practice places no practical limit on the length of an event ID, which is no longer an integer, but a combination of letters (e.g. network codes) and digits. The 40-character field provided in the MNF format should be adequate to handle these, but *mloc* only reads the first 10 characters from this field, because this is assumed to be adequate to distinguish between events in a cluster. An evid of less than 10 characters can be placed anywhere in the 10-character field (columns 12:21) that *mloc* reads; it will be right-justified inside *mloc*.

Prior to v1.3.3 of the MNF format, only a single evid could be entered for each event; it was carried at the end of the event record. In v1.3.3, evids are read from one or more **id records**. The format also includes a character field to identify the source of the evid. The current version of *mloc* can still read an older MNF file in “1.3” format and extract an evid carried in an **event record**. The first-encountered **id record** will be the preferred value in *mloc* by default, or the usage code (“=”) can be used to specify among multiple records.

The *orid* field in a **hypocenter record** (columns 104:121) is 18 characters in length because it is used by *mloc* to record the cluster name/series code (left-justified, starting in column 104). For **magnitude** and **phase** records, the associated 10-character ID field is in columns 112:121. Although 10 digit integer IDs can be carried in the ID field, it should be noted that 32-bit computer systems may have problems processing integers of more than 9 digits.

## Defined Record Types

The concept of “optional” fields in the descriptions of record types is specifically in the context of use by *mloc*. Fields that are required by *mloc* are printed in bold below. In writing MNF-

formatted data files it is advisable to pad lines to the full length of that record type, which is based on the defined fields, not the required fields, and it is not unwise to pad all lines to 121 characters, the full length of the longest defined record types, regardless of record type.

### ***Bulletin Record ("B" in column 1)***

Column	Description
<b>1</b>	<b>Line format flag "B"</b>
5:121	Bulletin description, optional (a117)

### ***Format Version Record ("F" in column 1)***

Column	Description
<b>1</b>	<b>Line format flag "F"</b>
<b>10-15</b>	<b>Format version (a6)</b>

Note: It is useful for readability to include extra text, such that columns 1:9 read "F MNF v", but all that is required is the "F" in column 1 and the version number in columns 10-15.

### ***Event Record ("E" in column 1)***

Column	Description
<b>1</b>	<b>Line format flag "E"</b>
3	Usage flag, optional (a1)
5:121	Annotation, optional (a117)

Note: A usage flag of "-" indicates that there is no phase data available for this event.

### ***ID Record ("I" in column 1)***

Column	Description
<b>1</b>	<b>Line format flag "I"</b>
3	Usage flag, optional (a1)
5:10	ID source, optional (a6)
12:51	Event ID, optional (a40)

The only recognized usage code is "=", which makes this entry preferred, regardless of its position relative to other ID records. Otherwise the first **id record** encountered will be used. An **id record** in which the Event ID field is left blank is legal. If there are multiple **id records** and it is desired to have *mloc* ignore them, an empty **id record** with usage code set to make it the preferred ID could be used.

### ***Hypocenter Record ("H" in column 1)***

Column	Description
<b>1</b>	<b>Line format flag "H"</b>
3	Usage flag, optional
<b>5-8</b>	<b>Year (i4)</b>
<b>10:11</b>	<b>Month (i2)</b>
<b>13:14</b>	<b>Day (i2)</b>
<b>16:17</b>	<b>Hour (i2)</b>
<b>19:20</b>	<b>Minute (i2)</b>
<b>22:26</b>	<b>Seconds (f5.2)</b>
28:32	OT uncertainty, optional (f5.2)
<b>35:42</b>	<b>Latitude (f8.4)</b>
<b>44:52</b>	<b>Longitude (f9.4)</b>
54:56	$S_{\min}$ azimuth, optional (i3)
58:62	Error ellipse $S_{\min}$ , optional (f5.2)
64:68	Error ellipse $S_{\max}$ , optional (f5.2)
70:74	Focal Depth, optional (f5.1)
76:76	Depth code, optional (a1)
78:82	Plus depth uncertainty, optional (f5.1)
84:88	Minus depth uncertainty, optional (f5.1)
90:93	GTCU, optional (a4)
95:102	Author, optional (a8)
104:121	Origin or cluster ID, optional (a18)

Note: *mloc* recognizes a hypocenter record in which the usage flag is "=" as the preferred hypocenter for setting the starting location. If no hypocenter record carries the usage flag, the first hypocenter record encountered will be taken as preferred by *mloc*. Other software may behave differently.

Both depth uncertainties are provided as positive numbers. "Plus" depth uncertainty is on the deeper side; "Minus" uncertainty is shallower, and therefore should not be greater than the focal depth in absolute value. If only one depth uncertainty is encountered, it should be interpreted as a symmetric uncertainty. It is important to put the decimal place into the depth field, even if precision is only to the nearest kilometer (or more), to ensure correct reading by a Fortran formatted read statement.

No information on the statistical level of the uncertainties of origin time, depth, or epicenter is provided in the MNF format because there is so little standardization at present. Such values are commonly interpreted as  $\pm 1 \sigma$  for origin time and depth, but confidence ellipses are usually calculated at 90% or 95% confidence levels.



The “GTCU” field carries a four-character code relating to calibration status (I prefer this term to “ground truth” level). It could be the GTX formulation (e.g., Bondar et al., (2004)), but I have developed the GTCU nomenclature to provide much more detailed information on the subject of what hypocentral parameters are considered to be calibrated (i.e., thought to be bias-free). The GTCU nomenclature is documented fully elsewhere.

The placement of a traditional (integer) “orid” value within the 18-character field is optional, but it is probably best to right-justify it (i.e., in columns 112:121) to facilitate reading as an integer. *mloc* writes the cluster name and series number as a character string that is left-justified in the field, beginning at column 104.

### **Depth Record (“D” in column 1)**

Column	Description
<b>1</b>	<b>Line format flag “D”</b>
3	Usage flag, optional
<b>5:9</b>	<b>Depth (f5.1)</b>
11:11	Depth code, optional (a1)
13:17	Plus depth uncertainty, optional (f5.1)
19:23	Minus depth uncertainty, optional (f5.1)
25:121	Authorship and comments, optional (a97)

Note: Both depth uncertainties are provided as positive numbers. “Plus” depth uncertainty is on the deeper side; “Minus” uncertainty is shallower, and therefore should not be greater than the focal depth in absolute value. If only one depth uncertainty is encountered, it should be interpreted as a symmetric uncertainty. It is important to put the decimal place into the depth field, even if precision is only to the nearest kilometer (or more), to ensure correct reading by a Fortran formatted read statement. The “depth code” in column 11 is an optional character flag that informs *mloc* about the nature of the depth constraint. Standard values are:

Depth Code	Meaning
c	Cluster default depth
d	Teleseismic depth phases
e	Engineered (man-made explosion)
f	Fault model (InSAR, GPS, etc)
l	Local-distance readings
m	<i>mloc</i> solution with free depth
n	Near-source station readings
r	Relocation outside <i>mloc</i> with free depth
u	Unknown
w	Waveform analysis

Although the use of depth codes is optional (blank is processed without problems), it is recommended to use them whenever possible, because certain procedures in *mloc* make use of these codes, for example, in determining the starting depth for an event.

The depth code “1” indicates a constraint on focal depth from readings at local distance (but still direct, crustal P and S phases), beyond the distance range normally considered appropriate for depth control (for which the “n” flag is provided). This is an advanced topic that requires careful analysis in *mloc*.

### **Magnitude Record ("M" in column 1)**

Column	Description
<b>1</b>	<b>Line format flag “M”</b>
3	Usage flag, optional (a1)
<b>5:8</b>	<b>Magnitude (f4.2)</b>
10:14	Magnitude scale, optional (a5)
16:110	Author and comments, optional (a95)
112:121	Magnitude ID, optional (a10)

Note: *mloc* recognizes a magnitude record in which the usage flag is “=” as the preferred magnitude for this event. *mloc* uses only the first two characters of magnitude type. Magnitude ID would normally be the orid from a **hypocenter record**; it should be right-justified in the field.

### **Phase Reading Record ("P" in column 1)**

Column	Description
<b>1</b>	<b>Line format flag “P”</b>
3	Usage flag, optional (a1)
<b>5:10</b>	<b>Station code (a6)</b>
12:17	Epicentral distance, optional (f6.2)
19:21	Azimuth, event to station, optional (i3)
23:23	Prevent phase re-identification flag, optional (“!”)
24:31	Input phase name, optional (a8)
<b>33:36</b>	<b>Arrival time year (i4)</b>
<b>38:39</b>	<b>Arrival time month (i2)</b>
<b>41:42</b>	<b>Arrival time day (i2)</b>
<b>44:45</b>	<b>Arrival time hour (i2)</b>
<b>47:48</b>	<b>Arrival time minute (i2)</b>
<b>50:55</b>	<b>Arrival time seconds (f6.3)</b>
57:58	Reading Precision, optional (i2)
60:64	TT residual, optional (f5.1)
66:73	Original phase name, optional (a8)
75:79	Agency, optional (a5)
81:88	Deployment or network, optional (a8)

90:94	Station, optional (a5)
96:97	Location, optional (a2)
99:101	Channel, optional (a3)
103:110	Author, optional (a8)
112:121	Arrival ID, optional (a10)

Note: It is helpful, but not required, to place a “dot” between the ADSLC fields. The usage flag carries the variable “fcode” in *mloc*. Common values of fcode are:

Value	Meaning
x	outlier, do not use
d	duplicate reading, do not use
m	missing station coordinates, do not use
p	phase that is unknown or not used
s	reading that is being skipped

Readings marked with ‘s’ are skipped according to an explicit command (SKIP command) that has been given in *mloc* to specify a combination of station-phase-authorship to be skipped. Wildcards are supported so that, for example, all instances of a certain station can be skipped, or only specific phase readings by a specific author.

### **Comment Record ("#" in column 1)**

Column	Description
<b>1</b>	<b>Line format flag “#”</b>
2:121	Comment, optional (a120)

### **Stop Record ("S" in column 1)**

Column	Description
<b>1</b>	<b>Line format flag “S”</b>

Note: Only the “S” in column 1 is required, but for better readability it is useful to write “STOP” in columns 1:4.

### **End of File Record ("EOF" in columns 1-3)**

Column	Description
<b>1:3</b>	<b>Line format flag “EOF”</b>