mloc Command Files

Eric Bergman
Global Seismological Services
June 9, 2013

This document describes the basic structure of command files for use with the multiple event relocation program *mloc*.

Background

Control over the processing of mloc is managed by commands, of which there are about 70. Only a small number of commands are always required for a run. Commands may be entered interactively or read from a text file containing the commands and arguments. In practice we use both approaches, starting with a basic command file that handles commands that are mainly repeated from run to run, and issuing a small number of commands interactively to launch the run. Indeed, a command file cannot be read except via an interactive command (cfil). It is common to edit the command file slightly between runs as well, but most of the information (notably the event definitions) in a command file stays constant from run to run.

Command files can be given any name, but standard practice is to make a new command file for each run and name it after the run ID, e.g., “cluster1.2.cfil”. The filename suffix “cfil” is standard, but it is not required.

Processing a Command File

When mloc is launched there are several interactive steps that occur first:

- Giving a name for the run
- Giving the name of the subdirectory where the related files are stored
- Specifying if data flags are to be honored

After this, the program lists all the available commands and asks for interactive command input. The normal procedure is to tell mloc to process the command file that has been prepared for this run:

```
cfil cluster1.2.cfil
```

During processing of the command file (in the main program mloc.f90) the user may be asked for some additional input, or warning messages may be issued. The processing of commands is handled by the subroutines in mloc_commands.f90. When the command file is finished the program asks for more interactive input. The relocation does not begin until the user issues the “run” command.

Command File Structure

Command files consist of two primary sections. The first section contains commands that control the procedures that will be used for the run, affecting all events, but it does not define the actual events that will be relocated. The second section defines the events and allows the user to issue commands which are event-specific. Some commands can be issued in either section and have a different action, depending on whether they are issued before any events have been defined.

Defaults

It is possible to make a run of mloc with no commands except the ones that define events, because all parameters that control how the relocation is done have defaults. The defaults are defined in the main program mloc.f90. In this case the relocation would be done using teleseismic P arrivals and the ak135 travel time model. Data would be weighted inversely to their uncertainty and phase re-identification would be done. All four hypocentral parameters would be free. Only a basic output plot would be made, as well as the core output files.

Defining Events

Events are defined with a minimum of three commands, for example:

```
memb
even 20090807.1859.26
inpu 20090807.1859.26.mnf
```

The “memb” (MEMBer) command starts the definition of a new event. The “even” (EVENt) command provides an identifying name for the event that will appear in various parts of the output files. It is normally derived from the filename of the datafile itself, which is supplied in the “inpu” (INPUt) command. Additional commands could be given and they would apply only to the current event until another “memb” command is encountered.

Except for the very smallest clusters, assembly of the event definition section would be very tedious to do by hand, so it is normally done by one of several utilities, especially ones that search through a concatenated set of event files (i.e., a bulletin) and extract individual events for a cluster.

Terminating a Command File

No special steps are required to terminate a command file. It normally ends with the definition of the last event.