Crystal Clear Collaboration

List Mode Format Version 2.0
Getting started and Using GATE with LMF

Luc SIMON
Magalie KRIEGUER
Daniel STRUL
Giovanni SANTIN
Christian MOREL
March 1, 2005
IPHE/UNIL

License

All source codes are covered by the GNU Lesser General Public License (see *LGPL.txt* for text).

Supported platforms

All source codes are implemented in ANSI C.
The LMF library has been successfully compiled, linked and tested on both UNIX and Linux systems. We recommand to use the *gcc* compilers 3.2.2 up to 3.4.2.

List of needed headers

- <stdio.h>
- <stdlib.h>
- <string.h>
- <netinet/in.h>
- "lmf_format.h" (generated by the *configure* script)

Introduction

The LMF library contains tools that implement and exploit the List Mode Format (LMF) developed for the ClearPET project of the Crystal Clear Collaboration. This format allows to store events of the small animal ClearPET demonstrator on an event-by-event basis. This document describes how to install, compile and execute some examples of the LMF library. And, in the lat part, how to generate LMF files from a GATE simulation.

An interfile 3D sinogram builder is implemented within STIR (Software for Tomographic Image Reconstruction). Figure 1 gives an overview of the LMF library.

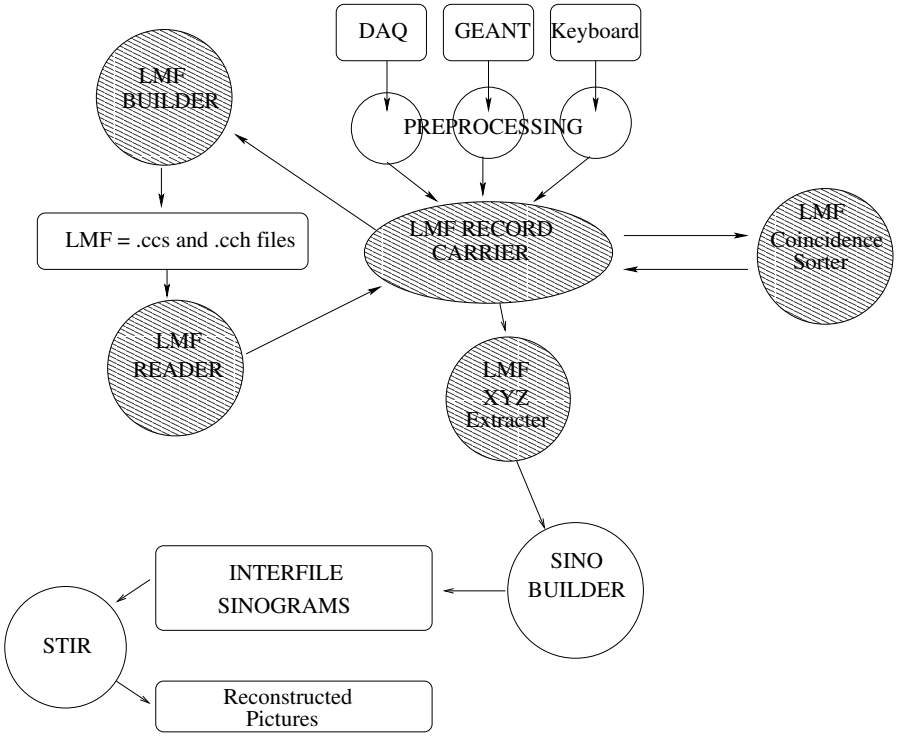

Figure 1: Overview of the LMF library. Dashed objects are included in the LMF library.

The ClearPET LMF is composed of two files: a binary file (*.ccs* extension) and an ASCII file (*.cch* extension). The ASCII file contains information about scanner dimensions and acquisition parameters (date, sizes, ...). See *e.g. LMF/examples/test.cch*.

The binary file contains the records themselves stored in a compact binary format. Actual records implemented in the LMF library are the *event record* (singles or coincidences), the *gate digi record*, and the *count rate record*.

The library mainly allows to read and write records, and to sort coincidences from a LMF file of singles. As shown on Figure 1, the LMF *record carrier* must

be loaded by using an appropriate *preprocessor*, or by reading LMF files with the LMF *reader* prior to processing LMF data further on. Thus, if your data output are *e.g.* singles, you can build a LMF file of singles, then read it, sort it into coincidences with a chosen coincidence time window, build a new LMF file of coincidences, read this new file, extract the (*x,y,z*) coordinates corresponding to the detector ID, and send those to the *sinogram builder*.

It is recommended to read some documents before using the library:

- `LMF_specifs.pdf`
  : LMF specifications
- `ClearPET_LMF.pdf`
  : LMF implementation
- `LMF_geometry.pdf`
  : LMF implementation : geometry description
- `LMF_GATE_digi_record.pdf`
  : LMF GATE record implementation

1 Install the library

You need nothing but a C compiler to install the LMF library. The source codes are totally ANSI compatible. They have been succesfully compiled on several platforms. The *LMF/* directory can be simply installed locally on your system without any path or *setenv* definitions. You must first untar and unzip the file *LMF.tar.gz*. For this, type:

```
tar xvzf lmf.tar.gz
```

As a result, you can find in *LMF/* 5 directories, a C code (*testForFormat.c*), a configuration script (*configure*), and a makefile (*makefile*).

1.1 Configure

The LMF library needs some standardized variable types, *e.g.* instead of using the *unsigned char* type, it uses *u8*. Since the size of variables differs from one system to another, you have to update the file *includes/lmf_format.h* that contains the type definitions.

To execute the *configure* script, type in *LMF/*:

```
configure
```

This script compiles and executes a source code that generates a new *includes/lmf_format.h* file.
You can check that this file looks like (but may differ):

```
#ifndef __FORMAT__FOR__LMF__
#define __FORMAT__FOR__LMF__
typedef char i8;
typedef unsigned char u8;
typedef short i16;
typedef unsigned short u16;
typedef long i32;
typedef unsigned long u32;
typedef unsigned long long int u64;
typedef long long int i64;
#endif
```

1.2 Compile the LMF library and the examples

In *LMF/*, you can execute the makefile by typing:

```
make clean
make
```

This compiles the source codes and archives *libLMF.a* in *LMF/lib/*. The source codes are in *LMF/src/*, and the header files are in *LMF/includes/*.
To verify that the library has been correctly installed, change directory to *LMF/examples/* where there is another makefile. If you type in this directory:

```
make clean
make
```

you create 5 executables named *EXE_01, EXE_02, EXE_03, EXE_04,* and *EXE_05*. These executables use most of the *libLMF.a* functionalities and allow to understand how one can use the LMF library.

2 Your first steps

In *LMF/examples/* you will find 1 example of LMF files:
test.ccs and *test.cch*.
The files *test.ccs* and *test.cch* have been generated by GATE (Geant4 Application for Tomographic Emission). For example, you can execute *EXE_04*, type:

```
test
```

and then play with the different menu choices.

2.1 Create a LMF file

You can create an artificial *.ccs* file following the instructions of *EXE_01*. The binary output file will be *dummy.ccs*. This latter can be read by *EXE_04*. It uses also the file *dummy.cch*, which is already in the directory.

The files *dummy.ccs* and *dummy.cch* do not have a great interest. Nevertheless, reading *exampleMain_01.c* will help you to learn how to build your own LMF files using your own data from simulation or from real acquisition.
WARNING: After each modification of a source code in *LMF/src/* or *LMF/includes/*, you must recompile *libLMF.a*. In *LMF/*, type *make* if you have modify a **.c* file, and *make clean* followed by *make* if you have modify a **.h* file.
The *LMFbuilder()* function has the following prototype:

```
FILE (*LMFbuilder(struct LMF_ccs_encodingHeader *pEncoH,
  struct LMF_ccs_eventHeader *pEH,
  struct LMF_ccs_countRateHeader *pCRH,
  struct LMF_ccs_gateDigiHeader *pGDH,
  struct LMF_ccs_currentContent *pcC,
  struct LMF_ccs_eventRecord *pER,
  struct LMF_ccs_countRateRecord *pCRR,
  FILE *pfile,
  const i8 *nameOfFile));
```

At first call, this function builds the header of the binary file *.ccs* and writes the first record. Then, at every call, it adds a new record to the binary file.
The use of this function is possible only if you have correctly filled the LMF *record carrier*. This latter is composed by the following structures defined in *includes/structure_LMF.h*

1. struct LMF_ccs_encodingHeader *pEncoH
2. struct LMF_ccs_eventHeader *pEH
3. struct LMF_ccs_countRateHeader *pCRH
4. struct LMF_ccs_gateDigiHeader *pGDH
5. struct LMF_ccs_currentContent *pcC
6. struct LMF_ccs_eventRecord *pER
7. struct LMF_ccs_countRateRecord *pCRR
8. struct LMF_ccs_gateDigiRecord *pGDR

Before any call to *LMFbuilder()*, you must have filled:

- 1
- 2 and/or 3 (and optionally 4 but only if 2 is filled)
- 5
- 6 and/or 7 (and optionally 8, but only if 6 is filled, and pGDR pointer of 6 must point to 8)

2.2 Read a LMF file

Once your LMF files have been built, their reading is quite easy. Look at *exampleMain_04.c* to understand how you can read your files. The functions *LMFcchReader()*, and *LMFreader()* allow to load the LMF *record carrier*. You have to specify as 2nd parameter of *LMFreader()* the processing mode of your records. It must be a string chosen among:

- *countRecords*: counts the different records in a file
- *dump*: displays the records one by one
- *sortCoincidence*: sorts coincidences from a file of singles and generates a couple of LMF files with *_coinci.ccs* and *_coinci.cch* extensions
- *analyseCoinci*: displays coincidence statistics between randoms, trues, and scattered coincidences. This process needs a file containing *gate digi records*.
- *locateIdInScanner*: extract the (*x,y,z*) coordinates of a detector ID, taking into acount the axial and azimuthal position of the scanner
- *treatAndCopy*: copy a couple of LMF files with *_bis.ccs* and *_bis.cch* extensions after having applied a selection of data (threshold cut, deadtime rejection, ...). Some examples have been implemented, but you can create your own data analysis. For this, see *e.g. src/treatEventRecord.c*.

3 Create LMF files from GATE

First, you must create a LMF home directory (ex. : *lmf/*) containing :

- *includes/* directory, containing all the LMF header files (.h)
- *lib/* directory containing libLMF.a

This directory can be the installation directory *lmf/*, but we recommand you to copy it in your */opt/* directory.
Then define the LMF_HOME environement variable as the PATH to this *lmf/* directory:

```
setenv LMF_HOME /opt/lmf
```

Finally you must recompile GATE properly from the main directory of GATE:

```
source env_gate.csh
make clean
make
```

The *env_gate.csh* file takes in account the LMF directory PATH and, and then the *GNUmakefile* compile the gateToLMF class files:

- src/gateToLMF.cc

- src/gateToLMFMessenger.cc
- includes/gateToLMF.hh
- includes/gateToLMFMessenger.hh

Finally you can use the LMF output, with the scriptable interface of GATE. See part 4 for an example of GATE macro to configure your LMF files.

Warning : You can only use the *cylindrical1* GATE system if you want a LMF output. The other systems, like ecat or spectHead do not allow this output. The *cylindrical1* must be defined with the 6 levels geometry hierarchy shown on figure 2. 6 is the maximum number of levels and the words:

- cylindrical1
- rsector
- module
- submodule
- crystal
- layer

are reserved. You can remove the layer level if your PET scanner has no DOI. You can also remove one of the middle levels, like submodule, if you do not need a very divided scanner hierarchy. See the macro example in part 4 that defines the geometry of microPET P4 in the LMF standard.

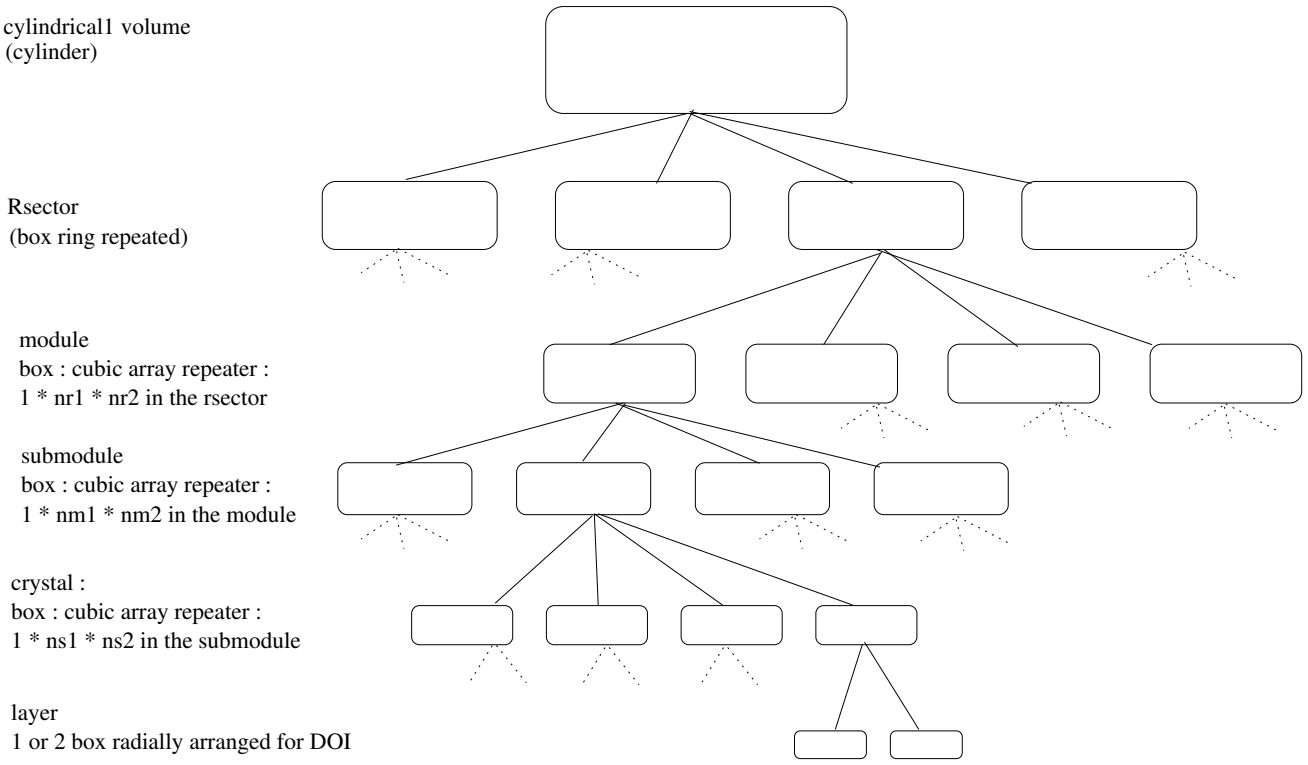

Figure 2: cylindrical1: a 6 levels geometry for GATE that allows LMF output.

4 Example of a GATE macro with LMF output

The following example shows a typical GATE macro that generates the two LMF files: .ccs and .cch. This macro is available on the private GATE webpages with the documentation.

```
# Please read the commented lines of this example.

# No verbosity
/control/verbose 0

#     W O R L D
/gate/world/geometry/setXLength 40 cm
/gate/world/geometry/setYLength 40. cm
/gate/world/geometry/setZLength 40. cm

#     M O U S E
/gate/world/daughters/name mouse
/gate/world/daughters/insert cylinder
```

```
/gate/mouse/setMaterial Water
/gate/mouse/vis/setColor red
/gate/mouse/geometry/setRmax 18.5 mm
/gate/mouse/geometry/setRmin 0. mm
/gate/mouse/geometry/setHeight 68. mm

#     C Y L I N D R I C A L
/gate/world/daughters/name cylindrical1
/gate/world/daughters/insert cylinder
/gate/cylindrical1/setMaterial Water
/gate/cylindrical1/geometry/setRmax 145 mm
/gate/cylindrical1/geometry/setRmin 130 mm
/gate/cylindrical1/geometry/setHeight 80 mm
/gate/cylindrical1/vis/forceWireframe

#     R S E C T O R
/gate/cylindrical1/daughters/name rsector
/gate/cylindrical1/daughters/insert box
/gate/rsector/placement/setTranslation 135 0 0 mm
/gate/rsector/geometry/setXLength 10. mm
/gate/rsector/geometry/setYLength 19. mm
/gate/rsector/geometry/setZLength 76.6 mm
/gate/rsector/setMaterial Water
/gate/rsector/vis/forceWireframe

#      M O D U L E
/gate/rsector/daughters/name module
/gate/rsector/daughters/insert box
/gate/module/geometry/setXLength 10. mm
/gate/module/geometry/setYLength 19. mm
/gate/module/geometry/setZLength 19. mm
/gate/module/setMaterial Water
/gate/module/vis/forceWireframe
/gate/module/vis/setColor gray

# C R Y S T A L
/gate/module/daughters/name crystal
/gate/module/daughters/insert box
/gate/crystal/geometry/setXLength 10. mm
/gate/crystal/geometry/setYLength 2.2 mm
/gate/crystal/geometry/setZLength 2.2 mm
/gate/crystal/setMaterial Water
/gate/crystal/vis/forceWireframe
/gate/crystal/vis/setColor magenta
```

```
#  L A Y E R
/gate/crystal/daughters/name LSO
/gate/crystal/daughters/insert box
/gate/LSO/geometry/setXLength 10. mm
/gate/LSO/geometry/setYLength 2.2 mm
/gate/LSO/geometry/setZLength 2.2 mm
/gate/LSO/placement/setTranslation 0 0 0 mm
/gate/LSO/setMaterial LSO
/gate/LSO/vis/setColor yellow

# R E P E A T    C R Y S T A L
/gate/crystal/repeaters/insert cubicArray
/gate/crystal/cubicArray/setRepeatNumberX 1
/gate/crystal/cubicArray/setRepeatNumberY 8
/gate/crystal/cubicArray/setRepeatNumberZ 8
/gate/crystal/cubicArray/setRepeatVector 10. 2.4 2.4 mm

# R E P E A T    M O D U L E
/gate/module/repeaters/insert cubicArray
/gate/module/cubicArray/setRepeatNumberZ 4
/gate/module/cubicArray/setRepeatVector 0. 0. 19.2 mm

# R E P E A T    R S E C T O R
/gate/rsector/repeaters/insert ring
/gate/rsector/ring/setRepeatNumber 42

# A T T A C H    S Y S T E M
/gate/systems/cylindrical1/rsector/attach rsector
/gate/systems/cylindrical1/module/attach module
/gate/systems/cylindrical1/crystal/attach crystal
/gate/systems/cylindrical1/layer0/attach LSO

# A T T A C H    L A Y E R  SD
/gate/LSO/attachCrystalSD
/gate/mouse/attachPhantomSD

# D I G I T I Z E R
/gate/digitizer/convertor/verbose 0
/gate/digitizer/modules/insert adder
/gate/digitizer/adder/verbose 0
/gate/digitizer/modules/insert readout
/gate/digitizer/readout/verbose 0
/gate/digitizer/readout/setDepth 1
```

```
/gate/digitizer/modules/insert blurring
/gate/digitizer/blurring/setEnergyOfReference 511. keV
/gate/digitizer/blurring/setResolution 0.26
/gate/digitizer/blurring/verbose 0
/gate/digitizer/modules/insert thresholder
/gate/digitizer/thresholder/setThreshold 250. keV
/gate/digitizer/modules/insert upholder
/gate/digitizer/upholder/setUphold 750. keV
/gate/digitizer/thresholder/verbose 0


# C O I N C I    S O R T E R
/gate/digitizer/coincidence/setWindow 10. ns
/geometry/test/run
/gate/systems/cylindrical1/describe



# I N A C T I V E    C O M P T O N
#/gate/physics/gamma/selectCompton inactive

# C U T   X ,  D E L T A   A N D   E L E C T R O N
/gate/physics/setXRayCut 1 GeV
/gate/physics/setDeltaRayCut 1 GeV
/gate/physics/setElectronCut 1 km



# I N I T I A L I Z E
/gate/systems/cylindrical1/verbose 0
/gate/geometry/enableAutoUpdate
/run/initialize


# V E R B O S I T Y
/control/verbose 0
/grdm/verbose 0
/run/verbose 0
/event/verbose 0
/tracking/verbose 0
/gate/application/verbose 0
/gate/generator/verbose 0
/gate/stacking/verbose 0
/gate/event/verbose 0
```

```
/gate/source/verbose 0


# S O U R C E
/gate/source/addSource twogamma
/gate/source/twogamma/setActivity 1000. becquerel
/gate/source/twogamma/setType backtoback
/gate/source/twogamma/gps/centre 0. 0. 0. cm
/gate/source/twogamma/gps/particle gamma
/gate/source/twogamma/gps/energytype Mono
/gate/source/twogamma/gps/monoenergy 0.511 MeV
/gate/source/twogamma/gps/type Volume
/gate/source/twogamma/gps/shape Cylinder
/gate/source/twogamma/gps/radius 18.5 mm
/gate/source/twogamma/gps/halfz 34.0 mm
/gate/source/twogamma/gps/confine mouse_P
/gate/source/twogamma/gps/angtype iso
/gate/source/twogamma/gps/mintheta    0. deg
/gate/source/twogamma/gps/maxtheta  180. deg
/gate/source/twogamma/gps/minphi      0. deg
/gate/source/twogamma/gps/maxphi    360. deg
/gate/source/list



# O U T P U T
/gate/output/ascii/setOutFileHitsFlag 1
/gate/output/ascii/setOutFileSingleDigiFlag 1
/gate/output/ascii/setOutFileCoincDigiFlag 1



##############################################
#
#  L M F    O U T P U T
#
##############################################

##############################################

# Set the LMF files name
# Here the output files will be myFirst.ccs
# and myFirst .cch
/gate/output/lmf1/setLMFFileName myFirst
```

```
# Coincidence Bool = 0 -> Singles LMF file
/gate/output/lmf1/setCoincidenceBool 0

# Store the detectorID (1) or not (0)
/gate/output/lmf1/setDetectorIDBool 1
# Store the energy (1) or not (0)
/gate/output/lmf1/setEnergyBool 1

# Store the axial position if the PET has
# a translation movement (1) or not (0)
/gate/output/lmf1/setGantryAxialPosBool 0


# Store the angular position if the PET has
# a rotation movement (1) or not (0)
/gate/output/lmf1/setGantryAngularPosBool 0

# Store these reserved values like that:
/gate/output/lmf1/setSourcePosBool 0
/gate/output/lmf1/setNeighbourBool 0
/gate/output/lmf1/setNeighbourhoodOrder 0


# Store extra informations, that are not available
# in a real world (1) or not (0)
/gate/output/lmf1/setGateDigiBool 1
# if 0 the following values are ignored

# Store the compton number (1) or not (0)
/gate/output/lmf1/setComptonBool 1

# Store the sourceID (1) or not (0)
/gate/output/lmf1/setSourceIDBool 0

# Store the source XYZ position (1) or not (0)
/gate/output/lmf1/setSourceXYZPosBool 0

# Store the real XYZ position (1) or not (0)
/gate/output/lmf1/setGlobalXYZPosBool 0

# Store the eventID (1) or not (0)
/gate/output/lmf1/setEventIDBool 1
```

```
# Store the runID (1) or not (0)
/gate/output/lmf1/setRunIDBool 1

###############################################



# S T A R T
/gate/application/setTimeSlice     0.1  s
/gate/application/setTimeStart     0.   s
/gate/application/setTimeStop      0.1  s
/gate/application/startDAQ
```

Contents