

# LetItB インストールマニュアル

独立行政法人 沖縄科学技術研究基盤整備機構

神経計算ユニット

2008年4月21日 第1版

## 目次

1. はじめに .....	3
2. OS 別インストール作業 .....	4
3. Xerces-C Library インストール .....	5
4. Graph Visualization Library インストール .....	6
5. Gnu Scientific Library インストール .....	6
6. 数値計算用ライブラリ : BLAS/LAPACK インストール .....	7
7. SBML Library インストール .....	8
8. Sundials Library インストール .....	9
9. SBML ODE Solver Library インストール .....	10
10. Qt インストール .....	12
11. Qwt インストール .....	13
12. LetItB インストール .....	14

## 1. はじめに

本マニュアルは、LetItB ソフトウェアを利用するために必要な外部ライブラリおよび LetItB ソフトウェアの本体のソース・インストール方法について記述しております。

本マニュアルの目次順に沿って適宜インストール作業を行って下さい。

### ● 必要ライブラリー一覧（\*下記順番に沿ってインストールすること）

- Xerces-C Library ( xerces-c-2.7.0 or later )
- Graph Visualization Library ( graphviz-2.16.1 or later )
- Gnu Scientific Library ( gsl-1.9 or later )
- 数値計算用ライブラリ : BLAS/LAPACK ( mkl 8.1 or later )
- Systems Biology Markup Language Library ( libsbml-2.3.4 )
- SUite of Nonlinear and Differential/ALgebraic equation Solvers ( sundials-2.1.1 )
- SBML ODE Solver Library ( SBML-odeSolver-1.6.0 )
- Qt ( qt-4.3.1 or later )
- QWT ( qwt-5.0.2 or later )

### ● LetItB ソースプログラム

- LetItB\_ver\_alpha.tar.gz

### [ 推奨環境 ]

\* 現バージョンでのプログラム動作推奨環境は、以下の通りです。

OS : Linux 系 OS (32bit/64bit), Mac OS X

CPU : Intel Pentium 1.30 GHz 相当以上

Mem : 1GB 以上

空き容量 : 100MB 以上

解像度 : SXGA(1280x1024) 以上

### [ 問い合わせ先 ]

本プログラムに関する問い合わせは、下記 e-mail アドレスにご連絡下さい。

沖縄科学技術研究基盤整備機構 神経計算ユニット LetItB-support

e-mail : [nc-admin@oist.jp](mailto:nc-admin@oist.jp)

## 2. OS 別インストール作業

### [ 全 OS 共通 ]

gcc, g++, make 等のプログラムをコンパイルする環境が整っていることが前提です。

### [ Linux OS インストール作業 ]

別章、3. Xerces-C Library インストール～12. LetItB インストールの順番に従い、インストール作業を行って下さい。

### [ Mac OS X インストール作業 ]

#### ① 事前準備

Mac OS X のインストーラ CD を利用し、X11 と Xcode を事前にインストール

#### ② MacPorts インストール

a. 下記、URL にアクセスし、利用 OS のバージョンに合わせて MacPorts をインストール

URL : <http://svn.macports.org/repository/macports/downloads/MacPorts-1.5.0>

( 例 : Mac OS 10.4 の場合は、MacPorts-1.5.0-10.4.dmg をダウンロード )

b. ~/.profile および ~/.bashrc を編集し、コンソールのパスの設定をする

下記、URL を参考にパスを設定

URL : <http://guide.macports.org/>

c. MacPorts のパッケージ情報をアップデート

以下のコマンドを実行

```
# sudo port selfupdate
```

#### ③ 各種ライブラリインストール

「ターミナル」または「xterm」を用い、MacPorts を利用して必要ライブラリ (Xerces-C 、 Expat 、 graphviz 、 grace ) をインストール

具体的にはターミナルから以下のコマンドを入力

```
# sudo port install xercesc
```

```
# sudo port install expat
```

```
# sudo port install graphviz
```

```
# sudo port install grace
```

```
# sudo port install gsl
```

\* 必要に応じて以下の環境変数を設定すること

```
export DYLD_FALLBACK_LIBRARY_PATH=/opt/local/lib:/opt/local/lib/graphviz
:$DYLD_FALLBACK_LIBRARY_PATH
```

- ④ 別章、7. SBML Library インストール～12. LetItB インストールの順番に従い、インストール作業を行って下さい。

### 3. Xerces-C Library インストール

\* Mac OS X の場合は 2 章参照

- ① 開発元からソースを入手 <http://xml.apache.org/xerces-c/download.cgi>

- ② 任意の作業ディレクトリにて圧縮ファイルを解凍

```
#cp xerces-c-current.tar.gz /opt
#tar zxvf ./xerces-c-current.tar.gz
```

- ③ 任意のディレクトリ名に rename

```
#mv xerces-c-src_2_7_0 xersec-c-2.7.0
```

- ④ 環境変数の設定

```
# export XERCESCROOT=/opt/xerces-c-2.7.0
```

- ⑤ Install (root で実行)

```
# cd /opt/xerces-c-2.7.0/src/xercesc
```

(以下 1 行にて記入)

```
# ./runConfigure -plinux -gcc -xg++ -minmem -nsocket -tnative -rpthread
# gmake
# gmake install
```

[補足] sample のビルド(必要に応じて)

```
# cd /opt/xersec-c-2.7.0/samples
# ./runConfigure -plinux -gcc -xg++
# gmake
```

\* 必要に応じて以下の環境変数を設定すること

```
export XERCESCROOT=/opt/xerces-c-2.7.0
export LD_LIBRARY_PATH=/opt/xersec-c-2.7.0/lib:$ LD_LIBRARY_PATH
```

## 4. Graph Visualization Library インストール

\*Mac OS X の場合は2章参照

① 開発元からソースを入手 <http://www.graphviz.org/>

② 任意の作業ディレクトリにて圧縮ファイルを解凍

```
#tar zxvf graphviz-2.16.1.tar.gz
```

③ インストール(root で実行)

```
# cd /<解凍先>/graphviz-2.16.1
```

```
# ./configure --prefix=/opt/graphviz-2.16.1
```

```
# make
```

```
# make install
```

\*必要に応じて以下の環境変数を設定すること

```
export LD_LIBRARY_PATH=/opt/graphviz-2.16.1/lib:$LD_LIBRARY_PATH
```

## 5. Gnu Scientific Library インストール

\*Mac OS X の場合は2章参照

① 開発元からソースを入手 <http://www.gnu.org/software/gsl/>

② 任意の作業ディレクトリにて圧縮ファイルを解凍

```
#tar zxvf gsl-1.9.tar.gz
```

③ 解凍されたディレクトリ内に移動

④ インストール(root で実行)

```
# cd /<解凍先>/gsl-1.9
```

```
# ./configure --prefix=/opt/gsl-1.9
```

```
# make
```

```
# make install
```

\*必要に応じて以下の環境変数を設定すること

```
export LD_LIBRARY_PATH=/opt/gsl-1.9/lib:$LD_LIBRARY_PATH
```

## 6. 数値計算用ライブラリ : BLAS/LAPACK インストール

\*Mac OS X の場合は、インストール作業は不要

Intel 社の HP より Math Kernel Library (Linux 非商用版)を、個別にダウンロード、ライセンスファイルの入手をお願いいたします。

<http://www.intel.com/cd/software/products/asmo-na/eng/219771.htm>

① 数値計算用ライブラリをダウンロード

\*例 Intel 社の Math Kernel Library を利用 l\_mkl\_p\_8.1.014.tgz

② 任意の作業ディレクトリにて圧縮ファイルを解凍

```
#tar zxvf l_mkl_p_8.1.014.tgz
```

③ 解凍されたディレクトリ内に移動

④ インストール(root で実行)

```
# cd /<解凍先>/l_mkl_p_8.1.014
```

```
# ./install.sh
```

\* インストーラの指示に従いインストール

\* 以下参考としてインストーラの流れを記述

Setp 1 : 1 を選択

Setp 2 : 2 を選択

Setp 3 : ライセンスファイルのパス(フルパス)を入力 XXX.lic

Setp 4 : 1 を選択

Setp 5 : /tmp/mkl を入力

Setp 6 : /opt/intel/mkl/8.1 を入力

\*必要に応じて以下の環境変数を設定すること

```
export LD_LIBRARY_PATH=/opt/intel/mkl/8.1/lib/32:$LD_LIBRARY_PATH
```

## 7. SBML Library インストール

\*事前に Xerces-C がインストールされている事

\*SBML ライブラリは libsbml-2.3.4(version 2.3.4)を利用すること

- ① 開発元からソースを入手
- ② 任意の作業ディレクトリにて圧縮ファイルを解凍

```
#unzip libsbml-2.3.4.zip
```

- ③ 解凍されたディレクトリ内に移動
- ④ マイナーバグの修正

```
/<解凍先>/libsbml-2.3.4/src/util/StringBuffer.c
```

```
151 行目の StringBuffer_appendNumber 関数内にある size = 16 を  
size = 32 に修正
```

```
/<解凍先>/docs/src/Makefile.in
```

```
79 行目の formatted=../formatted を formatted=../formatted/formatted  
に修正
```

\*以下、g++ での make 時にエラーが出た場合に修正

```
/<解凍先 >/libsbml-2.3.4/src/sbml/SBMLSchemaInputSource.h
```

```
79 行目 :
```

```
(修正前) BinInputStream* SBMLSchemaInputSource::makeStream () const;
```

```
(修正後) BinInputStream* makeStream () const;
```

```
関連 URL : http://www.sbml.org/forums/index.php?t=tree&goto=3837&rid=0
```

- ⑤ インストール(root で実行)

```
# cd /<解凍先>/libsbml-2.3.4
```

( Linux OS の場合 )

```
# ./configure --prefix=/opt/libsbml-2.3.4 --with-xerces=/opt/xerces-c-2.7.0
```

( Mac OS X の場合 )

```
# ./configure --prefix=/opt/libsbml-2.3.4 --with-xerces=/opt/local
```

```
# make
```

```
# make install
```

- ⑥ ドキュメントのインストール(省略可)

```
# make docs
```

```
# make install-docs
```



\* 必要に応じて以下の環境変数を設定すること

(Linux OS の場合)

```
export LD_LIBRARY_PATH=/opt/libsbml-2.3.4/lib:$LD_LIBRARY_PATH
```

(Mac OS X の場合)

```
export DYLD_FALLBACK_LIBRARY_PATH=/opt/libsbml-2.3.4/lib
:$DYLD_FALLBACK_LIBRARY_PATH
```

## 8. Sundials Library インストール

\*sundials ライブラリは sundials-2.1.1(version 2.1.1)を利用すること

① 開発元からソースを入手 <http://www.llnl.gov/CASC/sundials/>

② 任意の作業ディレクトリにて圧縮ファイルを解凍

```
#tar zxvf sundials-2.1.1.tar.gz
```

③ sundials 用の include ディレクトリと lib ディレクトリを作成

```
#mkdir -p /opt/sundials/include
```

```
#mkdir -p /opt/sundials/lib
```

④ インストール(root で実行)

```
# cd /<解凍先>/sundials
```

```
# ./configure --prefix=/opt/sundials --enable-shared
```

```
# make
```

```
# make install
```

\* 必要に応じて以下の環境変数を設定すること

(Linux OS の場合)

```
export LD_LIBRARY_PATH=/opt/sundials/lib:$LD_LIBRARY_PATH
```

(Mac OS X の場合)

```
export DYLD_FALLBACK_LIBRARY_PATH=/opt/sundials/lib
:$DYLD_FALLBACK_LIBRARY_PATH
```

## 9. SBML ODE Solver Library インストール

\* 事前に libsbml-2.3.4, sundials-2.1.1, graphviz がインストールされている事

① 開発元からソースを入手 <http://www.tbi.univie.ac.at/~raim/odeSolver/>

② 任意の作業ディレクトリにて圧縮ファイルを解凍

```
#tar zxvf SBML_odeSolver-1.6.0.tar.gz
```

③ 解凍されたディレクトリ内に移動

④ マイナーバグの修正

```
/<解凍先>/src/sbmlsolver/sbml.h
```

68 行目

```
(修正前) extern "C" {
```

```
(修正後) }
```

⑤ (Mac OS X の場合のみ)

パッチ (isSetValue.patch) を適用する

```
# cp isSetValue.patch SBML_odeSolver-1.6.0/
```

```
# cd SBML_odeSolver-1.6.0/src/
```

```
# patch -p1 < ../isSetValue.patch
```

環境変数の設定

```
# export LDFLAGS='-L/opt/local/lib -lgraph'
```

⑥ インストール (root で実行)

\* どちらの OS の場合でも 1 行で記入すること

```
# cd /<解凍先>/SBML_odeSolver-1.6.0
```

(Linux OS の場合)

```
# ./configure --prefix=/opt/SBML_odeSolver-1.6.0
```

```
--with-libsbml=/opt/libsbml-2.3.4 --with-sundials=/opt/sundials
```

```
--with-grace=/usr/X11R6/lib/xmgrace --with-graphviz=/opt/graphviz-2.16.1
```

(Mac OS X の場合)

```
# ./configure --prefix=/opt/SBML_odeSolver-1.6.0
```

```
--with-libsbml=/opt/libsbml-2.3.4 --with-sundials=/opt/sundials --with-grace=/opt/local
```

```
--with-graphviz=/opt/local/lib/graphviz
```

--- ⑥ インストールの続き---

```
# make
# make install
```

\*必要に応じて以下の環境変数を設定すること

(Linux OS の場合)

```
export LD_LIBRARY_PATH=/opt/SBML_odeSolver-1.6.0/lib:$LD_LIBRARY_PATH
```

(Mac OS X の場合)

```
export DYLD_FALLBACK_LIBRARY_PATH =/opt/SBML_odeSolver-1.6.0/lib
:$ DYLD_FALLBACK_LIBRARY_PATH
```

\* make 時のエラー対策

I. malloc.h を include できないというエラーメッセージが現れる

```
# sudo ln -s /usr/include/malloc/malloc.h /usr/include/malloc.h
```

を実行する。

II. arith.h を include できないというエラーメッセージが現れる

```
# sudo emacs /opt/local/include/graphviz/geom.h
```

などとしてエディタで geom.h 24 行目付近の #include "arith.h" をコメントアウトする。

III. -Unwind\_Resume を解決できないというエラーメッセージが現れる

```
# emacs odeSolver/Makefile
```

などとしてエディタで、67 行目付近を修正する。

(修正前) CCLD = \$(CC)

(修正後) CCLD = g++

## 10. Qt インストール

\* Open Source 版を利用すること

① 開発元からソースを入手 <http://www.trolltech.com/developer/downloads/qt/x11>

② 圧縮ファイルを解凍

```
#tar zxvf ./qt-x11-opensource-src-4.3.1.tar.gz
```

③ 任意のディレクトリ名に rename

```
#mv qt-x11-opensource-src-4.3.1 qt-4.3.1
```

④ インストール(root で実行)

```
# cd /<解凍先>/qt-4.3.1
```

```
# ./configure -prefix /opt/qt-4.3.1
```

```
# make
```

```
# make install
```

\* make および make install は非常に時間が掛かります。

⑤ インストール後、環境変数を設定

( Linux OS の場合 )

```
export QTDIR=/opt/qt-4.3.1
```

```
export PATH=$QTDIR/bin:$PATH
```

```
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
```

( Mac OS X の場合 )

```
export QTDIR=/opt/qt-4.3.1
```

```
export PATH=$QTDIR/bin:$PATH
```

```
export DYLD_LIBRARY_PATH=$QTDIR/lib:$DYLD_LIBRARY_PATH
```

[確認]

```
> qmake -version
```

バージョン情報が出力されると OK

[ Qt コンパイルの基本的な流れ]

```
> qmake -project
```

```
> qmake
```

```
> make
```

## 11. Qwt インストール

\* 事前に Qt がインストールされている事

- ① 開発元からソースを入手 <http://qwt.sourceforge.net/>
- ② 任意の作業ディレクトリにて圧縮ファイルを解凍

```
# unzip ./qwt-5.0.2.tar.gz
```

\* デフォルトの場合 /usr/local/qwt-5.0.2 内にインストールされる。変更したい場合は、

qwt-5.0.2/qwtconfig.pri の

```
unix {
```

```
    INSTALLBASE = ** ここを変更 (例 /opt/qwt-5.0.2)
```

```
}
```

- ③ インストール(root で実行)

```
# cd /<解凍先>/qwt-5.0.2
```

```
# qmake
```

```
# make
```

```
# make install
```

\* 必要に応じて以下の環境変数を設定すること

(Linux OS の場合)

```
export LD_LIBRARY_PATH=/usr/local/qwt/lib:$LD_LIBRARY_PATH
```

(Mac OS X の場合)

```
export DYLD_LIBRARY_PATH=/opt/qwt-5.0.2/lib:$DYLD_LIBRARY_PATH
```

[補足] example の make (必要に応じて)

```
> cd /<解凍先>/qwt-5.0.1/example
```

```
> qmake
```

```
> make
```

参考 URL : <http://qwt.sourceforge.net/qwtinstall.html>

## 12. LetItB インストール

\* 事前に全ての必要ライブラリがインストールされている事

① ソースを入手 `LetItB_ver_alpha.tar.gz`

② 任意の作業ディレクトリにて圧縮ファイルを解凍

```
# tar zxvf LetItB_ver_alpha.tar.gz
```

③ 必要ライブラリのパス設定(ファイルの編集)

```
LetItB_ver_alpha/src/create_makefile.sh
```

```
create_makefile.sh ファイル内にて、
```

```
該当する INCLUDEPATH, LIBS(ライブラリパス)を記入
```

④ コンパイル作業

```
# ./create_makefile.sh
```

```
# make
```

\* `LetItB_ver_alpha` ディレクトリに”LetItB”の実行ファイルが生成されていると終了

⑤ (Mac OS X の場合のみ) 必要に応じてリンクファイルを作成

```
# ln -s /<解凍先>/LetItB_ver_alpha/LetItB.app/Contents/MacOS/LetItB
```

```
/<解凍先>/LetItB_ver_alpha/LetItB
```

\* 1行で記入

### [実行方法]

```
> ./LetItB
```

\* 実行時に引数は無し

### [利用方法]

別途ユーザマニュアルをご覧ください。

以上