

MetaMap Source Distribution README

Willie Rogers

November 10, 2011

Contents

| | |
|--|----------|
| 1 Platforms and Prerequisites | 1 |
| 2 Getting the MetaMap Distribution | 2 |
| 3 Extracting the distribution | 2 |
| 4 Installing the distribution | 3 |
| 5 Instructions for compiling MetaMap | 4 |
| 6 Running the compiled MetaMap executable | 5 |
| 7 Running the compiled MetaMap Server executable (mmserver11) | 7 |
| 8 A Note about Dynamic Binaries | 8 |
| 9 Using the Prolog interpreter | 8 |
| 9.1 Running MetaMap in the interpreter | 8 |
| 9.2 Running MMServer in the interpreter | 9 |

1 Platforms and Prerequisites

The MetaMap Source Release currently only runs on the Linux and Mac OS/X 10.6 or greater. In order to build the MetaMap Source release a number of packages are required. These include the following:

MetaMap Full Download The full version of MetaMap must be installed before installing the source code distribution. It can be obtained at the MetaMap webpage: <http://metamap.nlm.nih.gov/#Downloads>.

SICStus Prolog 4.2.0 (32 bit only) available from Swedish Institute of Computer Science (SICS), the Quintus Prolog website: <http://www.sics.se/sicstus/>

GNU C Compiler should already be installed or be available as a package in your distribution if you're using Linux.

Sun Java Runtime Environment (JRE) 1.6 or later Sun's Java JRE 1.6.0 or later is required for use of the MetaMap Source Code Suite. Java is available from the "Developer Resources for Java Technology" website (<http://www.java.com/>)

GNU Binutils should already be installed or be available as a package in your distribution if you're using Linux.

GNU Make should already be installed or be available as a package in your distribution if you're using Linux.

Berkeley DB 4.8.24 should already be installed or be available as a package in your distribution if you're using Linux. Otherwise, you can get it from <http://www.oracle.com/database/berkeley-db/index.html>. The Berkeley DB Release Page is at: <http://www.oracle.com/technology/software/products/berkeley-db/db/index.html>. MetaMap is written to use the Berkeley DB 4.8.24 libraries. It is not expected to run properly using another other version of Berkeley DB, including later versions. This should compiled for 32-bit (use GCC option -m32 in CFLAGS) or use a 32-bit package if available.

Mac OS/X specific prerequisites See file README_macosx.html (online version: http://metamap.nlm.nih.gov/README_macosx.html) for more information on Mac OS/X-specific prerequisites.

2 Getting the MetaMap Distribution

The public MetaMap distribution can be downloaded at the Download Section of the MetaMap website:

<http://metamap.nlm.nih.gov/#Downloads>

The latest version of this document can be downloaded at:

HTML http://metamap.nlm.nih.gov/README_src_2011.html

PDF http://metamap.nlm.nih.gov/README_src_2011.pdf

3 Extracting the distribution

Use the following `tar` command extract the distribution in the same directory where the Public MetaMap distribution was extracted:

```
$ tar xvfj public_mm_linux_2011.tar.bz2
$ tar xvfj public_mm_src_2011.tar.bz2
```

`Tar` will create the distribution directory `public_mm`. Note: The data compression program BZIP2 (available from <http://www.bzip.org>) is required to decompress the distributions. GNU `tar` is preferred, but not required to extract the contents of the distributions.

If you don't have GNU `tar` use the following commands to extract the distributions:

```
$ bzip2 -dc public_mm_linux_2011.tar.bz2 | tar xvf -
$ bzip2 -dc public_mm_src_2011.tar.bz2 | tar xvf -
```

4 Installing the distribution

Connect to the new directory created by extracting the distribution and invoke the install program:

```
$ cd <distribution directory>
$ ./bin/install.sh
```

A sample run of the installation script follows:

```
$ ./bin/install.sh
Enter basedir of installation [/home/piro/projects/public_mm]
```

Basedir is set to /home/piro/projects/public_mm.

The WSD Server requires Sun's Java Runtime Environment (JRE) Sun's Java Developer Kit (JDK) will work as well. if the command: "which" java returns |/usr/local/jdk1.6.0_11/bin/java|, then the JRE resides in |/usr/local/jdk1.6.0_11|.

Where does your distribution of Sun's JRE reside?
Enter home path of JRE (JDK) [/usr/local/jdk1.6.0_11]:

Using /usr/local/jdk1.6.0_11 for JAVA_HOME.

```
/home/piro/projects/public_mm/WSD_Server/config/disambServer.cfg generated
/home/piro/projects/public_mm/WSD_Server/config/log4j.properties generated
Setting up bin directory scripts:
/home/piro/projects/public_mm/bin/make_all generated.
/home/piro/projects/public_mm/bin/metamap11 generated.
/home/piro/projects/public_mm/bin/SKRenv generated.
/home/piro/projects/public_mm/bin/SKRmake generated.
/home/piro/projects/public_mm/bin/skrmedpostctl generated.
/home/piro/projects/public_mm/bin/SKRrun generated.
/home/piro/projects/public_mm/bin/uninstall.sh generated.
/home/piro/projects/public_mm/bin/wsdserverctl generated.
Setting up test suite:
/home/piro/projects/public_mm/TestSuite/buildTestSuite11.sh generated.
/home/piro/projects/public_mm/TestSuite/runTest_2011.sh generated.
Checking for required datafiles
Checking for optional datafiles (WSD)
Public MetaMap Install complete.
Running MetaMap source development environment setup...
```

MetaMap Source Install

```
Public MetaMap Basedir: /home/piro/projects/public_mm
```

```
Where does your SICStus Prolog installation reside?
```

```
Enter base dir of SICStus Prolog installation [/usr/local/SICStus/sp-4.2.0-x86-linux-glibc2.5]:
```

```
Where does your GNU C compiler reside?
```

```
Enter base dir of GNU C Compiler [/usr]:
```

```
Where does your Berkeley DB 4.8.24 installation reside?
```

```
Enter base dir of Berkeley DB 4.8.24 installation [/usr]:
```

```
/usr/local/BerkeleyDB.4.8.24-m32
```

```
Berkeley DB 4.5 sharable library found.
```

```
libdb.so sharable library found.
```

```
Setting up bin directory SKRenv script:
```

```
/home/piro/projects/public_mm/bin/SKRenv generated.
```

```
Source Install Settings:
```

```
SICStus basedir: /usr/local/SICStus/sp-4.2.0-x86-linux-glibc2.5
```

```
GCC basedir: /usr
```

```
GCC compiler: /usr/bin/gcc
```

```
Berkeley DB basedir: /usr/local/BerkeleyDB.4.8.24-m32
```

```
Public MetaMap Source Install complete.
```

```
Public MetaMap Install Settings:
```

```
Public MetaMap basedir: /home/piro/projects/public_mm
```

```
Public MetaMap Program Dir: /home/piro/projects/public_mm/bin
```

```
Java Home dir: /usr/local/jdk1.6.0_11
```

```
$
```

5 Instructions for compiling MetaMap

Add the `$(BASEDIR)/public_mm/bin` directory to your program path, where `BASEDIR` is where you placed the MetaMap public distribution/

To do this in `tcsh` or `csh` do the following:

```
$ set path = ( ${BASEDIR}/public_mm/bin $path )
```

If you are using `bash` or `sh`, use the following command:

```
PATH=${BASEDIR}/public_mm/bin:$PATH
export PATH
```

or source the init file `public_mm/setup.csh` or `public_mm/setup.sh`:

```
$ source setup.csh
```

or if using bash or sh

```
$ . setup.sh
```

Then move to the \$BASEDIR/public_mm directory and type:

```
$ cd $BASEDIR/public_mm/src
$ make
... various messages and warnings ...
```

If there are no errors, then go to the \$BASEDIR/public_mm/src and there should be two files: `metamap11.BINARY.<platform>` and `metamap11.sav` where `platform` is name of platform the binary is compiled for. Also, in the directory `$BASEDIR/public_mm/src/mmserver` there should be `mmserver11.BINARY.<platform>` and `mmserver11.sav`.

If you wish to use a different name for binary and savefile you can specify the name by setting the `APPNAME` variable when invoking `make`:

```
$ cd $BASEDIR/public_mm/src
$ make APPNAME=metamapcustom
```

Similarly, after running the initial compilation above, you can move to the `MMSERVER` directory and specify the name of the MetaMap server binary and savefile:

```
$ cd $BASEDIR/public_mm/src/mmserver
$ make APPNAME=mmserverlocal
```

6 Running the compiled MetaMap executable

Be sure the SKR/Medpost Tagger server is running before attempting to run the compiled MetaMap program. Also, if you want Word Sense Disambiguation be sure the WSD Server is running. (see "Starting the MetaMap Servers" in the MetaMap README file, <http://metamap.nlm.nih.gov/README.html>)

You can run the compiled executable from the `$BASEDIR/public_mm/src` using the command:

```
$ SKRrun.11 ./metamap11.BINARY.Linux <metamap arguments>
```

For example:

```
$ SKRenv.11 ./metamap11.BINARY.Linux
./metamap11.BINARY.Linux
Berkeley DB databases (USAbase 2011AA strict model) are open.
Static variants will come from table varsan in /home/piro/projects/public_mm/DB/DB.USAbase.2011
Derivational Variants: Adj/noun ONLY.
Accessing lexicon /home/piro/projects/public_mm/lexicon/data/lexiconStatic2011.
```

Variant generation mode: static.

metamap11.BINARY.Linux (2011AA)

Control options: NONE

|: obstructive sleep apnea <-- Input by the user.
|:

Processing 00000000.tx.1: obstructive sleep apnea
Established connection to Tagger Server on localhost.
Processing 00000000.tx.1: obstructive sleep apnea

Phrase: "obstructive sleep apnea"

Meta Candidates (9):

- 1000 Obstructive sleep apnoea (Sleep Apnea, Obstructive) [Disease or Syndrome]
- 901 Apnea, Sleep (Sleep Apnea Syndromes) [Disease or Syndrome]
- 827 APNOEA (Apnea) [Finding,Pathologic Function]
- 827 Sleep [Organism Function]
- 827 Obstructive (Obstructed) [Functional Concept]
- 827 Apnea (Apnea Adverse Event) [Finding]
- 793 Sleeping (Asleep) [Finding]
- 755 Sleepy [Finding]
- 727 Sleeplessness [Sign or Symptom]

Meta Mapping (1000):

- 1000 Obstructive sleep apnoea (Sleep Apnea, Obstructive) [Disease or Syndrome]

|: heart attack and lung disease <-- Input by the user.
|:

Processing 00000000.tx.1: heart attack and lung disease
Established connection to Tagger Server on localhost.
Processing 00000000.tx.1: heart attack and lung disease

Phrase: "heart attack"

Meta Candidates (7):

- 1000 Heart attack (Myocardial Infarction) [Disease or Syndrome]
- 861 Heart [Body Part, Organ, or Organ Component]
- 861 Attack, NOS (Onset of illness) [Finding]
- 861 attack (Attack behavior) [Social Behavior]
- 861 Heart (Entire heart) [Body Part, Organ, or Organ Component]
- 861 Attack (Observation of attack) [Finding]
- 827 attacked (Assault) [Injury or Poisoning]

Meta Mapping (1000):

- 1000 Heart attack (Myocardial Infarction) [Disease or Syndrome]

Phrase: "and"

Phrase: "lung disease"

Meta Candidates (14):

```

1000 Lung Disease (Lung diseases) [Disease or Syndrome]
  972 Pulmonary Disease (Physicians - Pulmonary Disease) [Professional or Occupational Group]
  861 Disease [Disease or Syndrome]
  805 MAL (MAL gene) [Gene or Genome]
  805 MAL (MKL1 gene) [Gene or Genome]
  805 MAL (TIRAP gene) [Gene or Genome]
  805 MALS (MALARIA, MILD, SUSCEPTIBILITY TO) [Finding]
  694 Lung [Body Part, Organ, or Organ Component]
  694 Lung (Entire lung) [Body Part, Organ, or Organ Component]
  638 Pulmonary (Pulmonary:-:Point in time:^Patient:-) [Clinical Attribute]
  638 Pulmonary (Pulmonary (qualifier value)) [Qualitative Concept]
  601 Pneumonia [Disease or Syndrome]
  601 Pulmonary Artery (Pulmonary artery structure) [Body Part, Organ, or Organ Component]
  601 Pulmonary artery (Entire pulmonary artery) [Body Part, Organ, or Organ Component]

```

Meta Mapping (1000):

```

1000 Lung Disease (Lung diseases) [Disease or Syndrome]
|: ^D
$

```

7 Running the compiled MetaMap Server executable (mmserver11)

The MetaMap Java API distribution must be installed to use the Java Client library.

As with the compiled MetaMap executable, be sure the SKR/Medpost Tagger server is running before attempting to run MetaMap. Also, if you want Word Sense Disambiguation be sure the WSD Server is running. (see "Starting the MetaMap Servers" in the MetaMap README file, <http://metamap.nlm.nih.gov/README.html>)

go to the \$BASEDIR/public_mm/src and link the binary and save file in the \$BASEDIR/public_mm/src/mmserver directory in the \$BASEDIR/public_mm/src directory. THIS is done to allow the MetaMap server to use the sicstus runtime environment that is present in the \$BASEDIR/public_mm/src directory:

```

$ cd $BASEDIR/public_mm/src
$ ln mmserver/mmserver11.* ./

```

You can then run the compiled executable from the \$BASEDIR/public_mm/src directory using the command:

```

$ SKRrun.11 ./mmserver11.BINARY.Linux
Server options: [port(8066),accepted_hosts(['127.0.0.1'])]
Berkeley DB databases (USAbase 2011AA strict model) are open.
Static variants will come from table varsan in /home/piro/projects/public_mm/DB/DB.USAbase.2011
Derivational Variants: Adj/noun ONLY.
Accessing lexicon /home/piro/projects/public_mm/lexicon/data/lexiconStatic2011.
Variant generation mode: static.
port:8066

```

You should then be able to run the Java API client library normally; you can test the server using the test client `public_mm/bin/testapi.sh`.

8 A Note about Dynamic Binaries

The dynamic binaries produced by `linkSICStus` require the several sharable libraries created by `make_all` and the SICStus runtime environment.

The files `db_access.so`, `qp_lexicon.so`, `nls_signal.so`, and `qp_morph.so` in `public_mm/bin` are required by the MetaMap and MMServer binaries to run properly. On, Mac OS/X the corresponding files are `db_access.bundle`, `qp_lexicon.bundle`, `nls_signal.bundle`, and `qp_morph.bundle`. And on Windows NT and 7 the corresponding files are `db_access.dll`, `qp_lexicon.dll`, `nls_signal.dll`, and `qp_morph.dll`.

If you modify files in `public_mm/src/db`, `public_mm/src/lib`, `public_mm/src/lexicon/lexicon`, or `public_mm/src/lexicon/morph` you will need to copy the new versions of the corresponding files in `public_mm/src` to `public_mm/bin` or the changes made will not take effect.

Also needed are the files comprising the SICStus runtime environment which are provided with the main and binary distributions in the directory tree in `public_mm/bin/sp-4.2.0`; you can also create this environment using the instructions in the SICStus Prolog 4 User's Manual: "6.7.2.1 Runtime Systems on UNIX Target Machines" (<http://www.sics.se/sicstus/docs/latest4/html/sicstus.html/Runtime-Systems-on-UNIX-Target-Machines.html#Runtime-Systems-on-UNIX-Target-Machi>)

9 Using the Prolog interpreter

9.1 Running MetaMap in the interpreter

```
$ cd public_mm
$ source setup.sh (or setup.csh in C-shell or tcsh)
$ cd src
$ SKRenv.11 sicstus -f -l sicstus.ini <--- Note that this is SKRenv.11 and not SKRrun.11!!
...
```

The predicate "go/0" will start MetaMap:

```
go.
Berkeley DB databases (normal 10 strict model) are open.
Static variants will come from table varsan in /home/piro/projects/public_mm/DB/DB.normal.10.st
Derivational Variants: Adj/noun ONLY.
Accessing lexicon /home/piro/projects/public_mm/lexicon/data/lexiconStatic2011.
Variant generation mode: static.
```

```
sicstus (2011)
```

Control options:


```
|: cold
|:
Established connection to Tagger Server on localhost.
Processing 00000000.tx.1: cold
```

```
Phrase: "cold"
Meta Candidates (4):
  1000 Cold (Cold Temperature) [Natural Phenomenon or Process]
  1000 Cold (Common Cold) [Disease or Syndrome]
  1000 Cold (Cold Sensation) [Physiologic Function]
  966 Colds (Colds homeopathic medication) [Pharmacologic Substance]
Meta Mapping (1000):
  1000 Cold (Cold Sensation) [Physiologic Function]
Meta Mapping (1000):
  1000 Cold (Cold Temperature) [Natural Phenomenon or Process]
Meta Mapping (1000):
  1000 Cold (Common Cold) [Disease or Syndrome]
|: ^D
$
```

9.2 Running MMServer in the interpreter

```
$ cd public_mm
$ source setup.sh (or setup.csh in C-shell or tcsh)
$ cd src/mmserver
$ SKRenv.11 sicstus -f -l ../sicstus.ini <--- Note that this is SKRenv.11 and not SKRrun.11!!
...
```

The predicate “main/0” will start the MetaMap Server:

```
| ?- main.
Server options: [port(8066),accepted_hosts(['127.0.0.1'])]
Berkeley DB databases (USAbase 2011AA strict model) are open.
Static variants will come from table varsan in /nfsvol/nls3aux18/DB/DB.USAbase.2011AA.strict.
Derivational Variants: Adj/noun ONLY.
Accessing lexicon /rhome/wjrogers/specialist/SKR/src/lexicon/data/lexiconStatic2011.
Variant generation mode: static.
port:8066
```