JMMC	JMMC-MAN-2720-0002 Revision : 3.0.4 Date : 15/06/2012

JMMC

AMBER Data Reduction Software

Installation Guide

Author : Gilles Duvert Institute : LAOG	Date : SUBMITTED
Approved by : Guillaume Mella Institute : LAOG	Date : APPROVED
Released by : Fabien Malbet Institute : LAOG	Date : RELEASED

CHANGE RECORD

REVISION	DATE	AUTHOR	SECTIONS/PAGES AFFECTED
	REMARKS		
0.1	21/09/2007	Gérard ZINS	All
	First draft		
0.2	24/09/2007	Gérard ZINS	All
	Updated according to comments from Laurence		
0.3	25/09/2007	Gérard ZINS	
	Updated according to comments from Florentin		
0.4	27/09/2007	Gérard ZINS	2.3.2, 2.3.3 & 2.5
	Replaced `--cfistio-dir` and `--fftw-dir` options with `-C` and `-F`		
0.5	04/10/2007	Gérard ZINS	
	Updated according to comments from Fabien		
1.0	04/10/2007	Gérard ZINS	
	Released		
1.1	02/03/2010	Guillaume Mella	
	Updated to prepare the V3.0 release		
2.0	27/07/2010	Gilles Duvert	
	Rewrote everything for version 3 Add troubleshooting section and contact information		
2.1	26/05/2011	Guillaume Mella	2.4.6 b)
	Add some information’s required for emacs users / rlwrap / src installation		
3.0.5	15/06/2012	Guillaume Mella	2.4.6,
	Add information about openmp parallelization and rlwrap Set the version of the document the same as the software one		

TABLE OF CONTENTS

1 Introduction

1.1 Object

This document describes how to install AMBER Data Reduction Software, also known as *amdlib*, which is delivered in either binary or source package. You can get the software from the JMMC web site http://www.jmmc.fr/amberdrs

1.2 Architecture

Amdlib is made of:

- A library of C functions and command-line tools ("core amdlib") that can be used alone to reduce data.
- A more user-friendly interface, written in Yorick language, to ease the use of the core amdlib.
- Sets of Yorick scripts to go beyond the possibilities allowed by the core amdlib alone:
  - Visualization of raw, reduced and calibration data
  - Automated reduction of whole observing nights
  - Absolute calibration using, e.g., night transfer function
  - File browsers, logs, etc…

Thus amdlib uses, and the installation will build, 1) Yorick, 2) the cfitsio and fftw libraries used both by the core amdlib and the yorick plugins, 3) the core amdlib.

1.3 Acknowledgements

The use of amdlib is free, and without any guaranty of any sort. Please follow the acknowledgement policy stated at http://www.jmmc.fr/amberdrs/acknowledgement .

1.4 Reference documents

[1] JMMC-MAN-2720-0001, Revision 3.0, AMBER Data Reduction Software – User Manual

2 Installation

2.1 Supported operating systems

amdlib is supported on the following operating systems:

- Linux
- Mac OS X

WARNING: For MAC users, please configure your mouse with the “right click” option, otherwise you can't get out from the file browser provided with *amdlib*. You adjust this function in the preferences of the X11 application.

WARNING: You can run this software onto one windows machine using VirtualBox, but the installation must be done onto one ext file system because it fails onto one NTFS one.

2.2 About the different ways to install the software

The simplest way to have amdlib up and running is to install the binary package corresponding to your architecture. Give it a try, knowing that uncompatibilies may happen, depending on how old your version of the operating system may be.

The next best solution is to install from the sources, recompiling every package needed (they are all in the source tarball, and a script does it automatically).

Finally, experts in Linux or Mac OS X can build only parts of the distribution, if they already have installed Yorick and/or some libraries needed by amdlib. This kind of partial installation is documented here, but without guarantee.

2.3 Installing from the binary package

An *amdlib* binary package contains everything needed to run. A binary package is specific to a particular hardware/OS platform.

You can use **uname -a** to see which is your OS (% represent the shell prompt and is not part of the command):

```
 %  uname -a
Linux localhost 2.6.24.7 #1 SMP Thu Oct 30 14:31:33 EDT 2008 x86 64 Intel(R) Xeon(R)
CPU E5320 @1.86GHz GNU/Linux
```

On Mac OS X (Darwin), use the **sw_vers** command to see which is your OS version:

```
% sw vers
    ProductName:  Mac OS X
    ProductVersion:     10.6.1
    BuildVersion: 10B504
```

Download, from the JMMC web server, the binary package corresponding to your computer (the name is like **amdlib-*<version>*-bin-*<os>*-*<arch>*.tgz**), and unpack it where you want to install it, as shown below for a Linux/i686 version installed in your HOME directory:

```
% cd $HOME/
% tar -xzvf amdlib-3.0-bin-Linux-i686.tgz
    ./amdlib-3.0-bin-Linux-i686/
    ./amdlib-3.0-bin-Linux-i686/bin/
    ...
```

Then start amdlib by calling the program:

```
% <installation directory>/amdlib-<version>-bin-<os>-<arch>/bin/amdlib
```

With the previous example, the command would be:

```
% $HOME/amdlib-3.0-bin-Linux-i686/bin/amdlib
```

That's it! You can test your installation by following the instructions described in section 2.4.6.

It is of course more convenient to avoid typing the long command above. Use an alias:

```
% alias amdlib=$HOME/amdlib-3.0-bin-Linux-i686/bin/amdlib
```

And add this command at the end of your $HOME/.bashrc (or $HOME/.bash_profile on Mac OS X) file to avoid having retyping it each time you log in.

2.4 Installing from the source package

To install from the source package it will be necessary to define a few environment variables and check the prerequisites for compilation.
Since version 3.0.5 the compilation uses openmp to enhance computation speed. If you prefer or can't use this feature, please define NO_PARALLEL option.

2.4.1 Download

The source packages are named according to the following scheme:

```
amdlib-<version>-src.tgz
```

For example, `amdlib-3.0-src.tgz` is the source package containing *amdlib* version 3.0.

Download, from JMMC web server, the source package, and unpack it in a temporary directory, as shown below:

```
% tar -xzvf amdlib-<version>-src.tgz
    ./amdlib-3.0-src/
    ./amdlib-3.0-src/yoco/
    ...
```

2.4.2 Preparation

This step is very important and needs to type a few commands, it cannot be done automatically for you. However you can copy/paste to a terminal the example commands.

When you install amdlib from the sources, or want to recompile any of its components, you must have defined a directory where ***amdlib*** will be installed, and manage to have *an environment variable* called **`INTROOT`** pointing to it. This directory

must not be a preexisting directory (i.e., it must be empty at start and reserved for amdlib uses). Create the directory (lets name it "**INTROOT**" to simplify) set the **INTROOT** environment variable accordingly, and update the relevant searchpaths as shown in the following example:

```
%  mkdir $HOME/INTROOT
```

```
export INTROOT=$HOME/INTROOT
export PATH=$INTROOT/bin:$INTROOT/yorick/bin:$PATH
export LD_LIBRARY_PATH=$INTROOT/lib:$LD_LIBRARY_PATH
```

To ensure that you will always use the new binaries in your bash shell, you need to copy the last 3 lines (above text box) in your `$HOME/.bashrc` file , and type "`hash -r`". Not having the above paths set to the right values will end in failure to compile correctly amdlib and dependences.

NOTE: If you do not use the *yorick* program (see section 2.4.4.1) which is provided in the *amdlib* package, you do not have to add `$INTROOT/yorick/bin` to your path. If you use your own one, it should already present into the PATH variable.

NOTE: If you are using *tcsh* shell please edit `.cshrc` and replace the **export NAME=value** forms by **setenv NAME value**. To ensure that you will use the new binaries, execute the **rehash** command.
NOTE: Mac Os X users have to use `$HOME/.bash_profile` instead of `.bashrc`

A script will be used for the installation. By default the script installs the *yorick, cfitsio* and *fftw* libraries and programs as well, but you can interactively prevent the installation of a library already installed. The following section deals with the prerequistes if you have one or more of these already installed and want to make a "custom" installation. If not, you can go directly to section 2.4.3.

2.4.3 Default installation from the sources

The default command to compile and install amdlib is using the `install.sh` script with 'yes' by default, as shown below:

```
% cd amdlib-<version>-src
% ./install.sh -y
Installation directory is /home/test/amdlib
Installing AMBER-DRS on Linux/x86 64...
Installing tools/librairies for Yorick
Installation directory is /home/test/amdlib
Installing cfitsio library V3240...
Uncompressing ...
...
```

The script installs *amdlib* and all required software (if you do not want that, see 2.4.4); i.e. *yorick* program, *fftw* library and *cfitsio* library.

2.4.4 Prerequisites for a "custom" installation.

This section only concerns people who have one of the following software installed on their computer and who want to use it instead of the one provided in the *amdlib* package.

- *yorick* program, version **2.1** or later

- *fftw* library, version **3**
- *cfitsio* library, version **2.49** or later
- *openmp library* , present on most recent O.S.

If it is not your case, you can go directly to either section 2.3 for installing binary package or to section 2.4.3 for installing source package.

2.4.4.1 Yorick

amdlib needs at least version **Yorick 1.6.02**. If there is an old version (e.g. 1.5) installed on your computer, the installation will fail. Please uninstall the old version or (at least) remove it from your path, and use the one provided in *amdlib* package.

The yorick program **must be in your path**, and yorick sub-directories **must be writable**. To check this, execute the following commands:

```
% yorick
Copyright (c) 2005.  The Regents of the University of
California.
All rights reserved.  Yorick 2.1.01 ready.  For help type
'help'
> system, "ls -l " + Y_SITE
total 20
drwxr-xr-x  2 user group 4096 Feb 15 21:30 doc
drwxr-xr-x  2 user group 4096 Feb 21 20:01 g
drwxr-xr-x  2 user group 4096 Feb 25 11:26 i
drwxr-xr-x  2 user group 4096 Feb 15 21:30 i-start
drwxr-xr-x  2 user group 4096 Feb 24 16:03 i0
```

And verify the access permissions of the listed directories.

2.4.4.2 The *fftw* library

amdlib requires the *fftw* library version **3, in “development” mode**. Please install the right version (ex: “`urpmi libfftw-devel`”) or use the one provided in *amdlib* package. If the *fftw* library is not installed in a standard directory (i.e. `/usr` or `/usr/local`), use the `-F` option when installing.

2.4.4.3 The *cfitsio* library

amdlib requires the *cfitsio* library version **2.49 in “development” mode** (or later versions) (ex: “`urpmi libcfitsio-devel`”) . If you use an older version, the installation will fail. Please update your installation version or use the one provided in *amdlib* package. If the *cfitsio* library is not installed in a standard directory (i.e. `/usr` or `/usr/local`), use `-C` option when installing.

2.4.4.4 Additional tips to use installed libraries

Some OS install includes and librairies files in some specific locations. You can declare some environment variable to help the makefile compile amdlib. For this, please have a look into the Make.cfg file which give you the format of EXTRA_INC_PATH and EXTRA_LIB_PATH

2.4.5 Custom installation from the sources

If the script install.sh is called without the '-y' option, the installation of *yorick*, *fftw* and *cfitsio* is done interactively and the script asks for confirmation.
If one of the *fftw* and/or *cfitsio* library is already installed on your computer, `-F` and `-C` options can be used to specify the installation directories (see sections 2.4.4.22.4.4.2 and 2.4.4.3).

The synopsis of the `install.sh` script is:

```
install.sh [-h] [-y] [-C <dir>] [-F <dir>] [-l logfile]
  -h            print this help.
  -y            assume yes to all questions
  -C <dir>      specify directory where cfitsio library is installed
  -F <dir>      specify directory where fftw library is installed
```

2.4.6 Starting amdlib

After a successful installation from sources, amdlib is in your PATH and to start it, just execute *amdlib* or launch *yorick* by typing the following command line:

```
% yorick -i amdlib.i
Copyright (c) 2005.  The Regents of the University of California.
All rights reserved.  Yorick 2.1.01 ready.  For help type 'help'
…
--==< amdlib, version 3.0 >==--
Type "amdlib" if you want some help

\\\|||||||||||||||||||||||||||||||||||||||||||||||||||||||///
---                                                       ---
---       PLEASE ACKNOWLEDGE THE USE OF AMBER DATA,       ---
---          REDUCTION SOFTWARE FROM THE JMMC,            ---
---           FOR EACH USE IN A PUBLICATION.              ---
---      http://www.jmmc.fr/amberdrs/acknowledgement      ---
---                                                       ---
///|||||||||||||||||||||||||||||||||||||||||||||||||||||||\\\
```

If your setup is ok, you should also be able to launch amdlib using following yorick command:

```
> #include "amdlib.i"
```

To get a quick help, type:

```
> help, amdlib
```

To quit *yorick*, type:

```
> quit
```

It is recommended to use *amdlib* through either *rlterm/rlwrap* utility or *emacs* as explained on http://yorick.sourceforge.net/environ.php web page.

a) *Rlterm / Rlwrap* utility
rlterm or *rlwrap* offer you command recall and command-line editing capabilities.

Linux users may install *rlwrap* or *rlterm* from their common package manager tool. Mac users may copy the prepared binaries into one *bin* directory present in the PATH, or copy rlwrap program in the directory where amdlib command leaves (use *%which amdlib* to find it). Binaries are available from http://jmmc.fr/~swmgr/rlwrap

amdlib command will try to invoke one of these commands. Use arrow keys to explore the command history:

```
    % amdlib
     amdlib: Starting rlterm yorick -i amdlib.i

    Copyright (c) 2005.  The Regents of the University
      ...
```

b) *emacs*

To prepare the configuration of emacs for using amdlib, run `amdlib -e` command to get instructions as in the following example:

```
% /your/path/to/amdlib-3…/bin/amdlib –e
To use the amdlib package associated to this command into emacs:

Cut and paste next lines into '/Users/mella/.bash profile'
--8<--------------------------------------------------------------------------------
export INTROOT=/Users/mella/INTROOT
export PATH=$INTROOT/bin:$INTROOT/yorick/bin:$PATH
export LD LIBRARY PATH=$INTROOT/lib:$LD LIBRARY PATH
--8<--------------------------------------------------------------------------------

Cut and paste next lines into '/Users/mella/.emacs'
--8<--------------------------------------------------------------------------------
(if (not (featurep 'yorick)) (load "/Users/mella/INTROOT/yorick/bin/yorick.el" nil
t))
(setq yorick-executable-name "/Users/mella/INTROOT/yorick/bin/yorick")
--8<--------------------------------------------------------------------------------
```

Use one of the three option to launch *yorick* in an *emacs* session:

1. `emacs -f yorick` in a shell terminal to both launch *emacs* and *yorick*
2. `<alt>-x` and then type `"yorick"` in a running *emacs*
3. `<ctrl>-c <ctrl>-s` to save and include directly a *yorick* script file you are editing under *emacs*.

2.5 Troubleshooting

You can request help by contacting the JMMC user support
jmmc-user-support@ujf-grenoble.fr
http://www.jmmc.fr/support.htm

Please attach the full output of the “amdlib –c “:

```
    % /your/path/to/amdlib-3…/bin/amdlib -c
```

You are invited to increase the log level of your report using following commands:

```
    > amdlibSetPreference, “logEnable”, 1
    > amdlibSetPreference, "logLevel", "debug"
```