OSKAR Installation Guide

Version history:

Revision	Date	Modification
1	2012-04-23	Creation.
2	2012-06-13	Added description of new CMAKE build flags specifying the location of custom library search paths for LAPACK, CBLAS and CFITSIO. Added note on using custom (non-system) Qt4 installations.
3	2012-06-19	Added description of new CMAKE flags for specifying the location of CPPUNIT and CasaCore added in OSKAR-2.0.3-beta.
4	2013-04-12	Added description of MATLAB_ROOT CMAKE variable.
5	2013-11-14	Updated for version 2.3.0. Removed CppUnit dependency. Unit tests now use the Google testing framework, which is included in the OSKAR source tree.

1 Introduction

This document describes how to obtain, build and install the OSKAR simulation package. The current release is available as a source code distribution, targeted at Linux and Mac OS X operating systems. Microsoft Windows installs may be possible, but are not currently supported.

2 Obtaining OSKAR

The current release of OSKAR can be downloaded as a compressed zip archive from the following link:

- http://www.oerc.ox.ac.uk/~ska/oskar/OSKAR-Source.zip

Older versions (if available) are listed by version at:

http://www.oerc.ox.ac.uk/~ska/oskar

3 Dependencies

OSKAR depends on a number of other libraries. The main components of OSKAR require an NVIDIA GPU with CUDA 4.0 (and associated NVIDIA driver) or later and Qt 4 (version 4.6 or later) to be installed on the target system, and also CBLAS and LAPACK for full functionality. Additionally, the CFITSIO and casacore libraries must be present if FITS images and Measurement Sets are to be exported. The MATLAB interface will be built if MATLAB R2010b or later is found.

The unit tests require the make use of Google's C++ unit testing framework (https://code.google.com/p/googletest) which is bundled with OSKAR source code.

The OSKAR CMake build system will compile and install as many components as possible according to the following dependencies being satisfied:

- NVIDIA CUDA (http://developer.nvidia.com/cuda-downloads), version >= 4.0
  - Required for: liboskar, liboskar_apps, Applications, MATLAB interface
  - (Make sure you install the CUDA toolkit.)
- Qt4 (http://qt.nokia.com), version >= 4.6
  - Required for: liboskar_apps, Applications
  - Optional for: MATLAB interface (for manipulation of OSKAR settings files)
- CBLAS (http://www.netlib.org/blas)
  - Optional for: liboskar (to enable the use of extended sources)
  - Available as part of the GSL and ATLAS packages.
- LAPACK (http://www.netlib.org/lapack)
  - Optional for: liboskar (to enable the use of extended sources)
- CFITSIO (http://heasarc.gsfc.nasa.gov/fitsio)
  - Required for: liboskar_fits
  - Optional for: liboskar_apps, Applications (for FITS file export)
- casacore (http://code.google.com/p/casacore), version >= 1.2.0
  - Required for: liboskar_ms
  - Optional for: liboskar_apps, Application (for Measurement Sets export)

- MATLAB (http://www.mathworks.co.uk/products/matlab), version >= R2010b
    - Required for: Matlab interface
- CMake (www.cmake.org), version >= 2.8
    - Required for: Makefile generation

3.1 Ubuntu Packages

If you are running a recent version of the Ubuntu Linux distribution (>= 10.04), then you can simply download and install the following packages from the main distribution repository:

- libqt4-dev
- libatlas-base-dev
- liblapack-dev
- libcfitsio3-dev
- g++
- cmake

However, note that you will need to install CUDA from the NVIDIA downloads page, as described in Section 3. Note that you will also need the NVIDIA driver for your particular model of NVIDIA GPU.

4 Building OSKAR

Once the archive has been uncompressed, OSKAR can be built by issuing the following commands:

```
$ mkdir build
$ cd build
$ cmake [OPTIONS, SEE §4.1] ../top/level/source/folder [1]
$ make
```

OSKAR can then be installed with:

```
$ make install
```

You may need to run this command as root, unless you change the default install location as described in the next section.

[1] This is the folder that contains the top level CMakeLists.txt file, which will be in the root of the folder that you unzipped or checked out.

4.1 Build Options

When running the 'cmake' command a number of build options can be specified. These are as listed below.

- **-DCUDA_ARCH**=<arch> (default: 2.0)
  - Sets the target architecture for the compilation of CUDA device code
  - <arch> must be one of either: 1.3, 2.0, 2.1, 3.0, 3.5 or ALL
  - *Note: 1.1 (or even 1.0) architecture may also work if running in single precision, although this architecture isn't officially supported by OSKAR*

- **-DCMAKE_INSTALL_PREFIX**=<path> (default: /usr/local/)
  - Prefix install path used when installing OSKAR (with make install)

4.1.1 Advanced Build Options

- **-DCMAKE_BUILD_TYPE**=<release, debug, or relwithdebinfo> (default: release)
  - Build OSKAR with release flags, debug flags, or release flags with debugging symbols.

- **-DLAPACK_LIB_DIR**=<path> (default: searches the system library paths)
  - Specifies a custom path in which to look for the LAPACK library (liblapack.so).
  - *Note: This should only be used in special cases, where the version of LAPACK installed in the system library path can't be used to build OSKAR.*

- **-DCBLAS_LIB_DIR**=<path> (default: searches the system library paths)
  - Specifies a custom path in which to look for the CBLAS library (ATLAS: libcblas.so; GSL: libgslcblas.so)
  - *Note: This should only be used in special cases, where the version of CBLAS installed in the system library path can't be used to build OSKAR.*
- **-DCFITSIO_LIB_DIR**=<path> (default: searches the system library paths)
  - Specifies a custom path in which to look for the CFITSIO library (libcfitsio.so).
  - *Note: This should only be used in special cases, where the version of CFITSIO installed in the system library path can't be used to build OSKAR.*

- **-DCFITSIO_INC_DIR**=<path> (default: searches the system include paths)
  - Specifies a custom path in which to look for the CFITSIO library headers (fitsio.h and others)
  - *Note: This should only be used in special cases, where the version of CFITSIO headers installed in system include path can't be used to build OSKAR.*

- **-DCASACORE_LIB_DIR**=<path> (default: searches the system library paths)
  - Specifies a custom path in which to look for the CasaCore libraries (libcasa_ms.so and others).
  - *Note: This should only be used in special cases, where the version of* CasaCore *installed in the system library path can't be used to build OSKAR.*

- **-DCASACORE_INC_DIR**=<path> (default: searches the system include paths)
    - Specifies a custom path in which to look for the CasaCore library headers. This is the path to the top level casacore include folder.
    - *Note: This should only be used in special cases, where the version of* CasaCore *headers installed in system include path can't be used to build OSKAR.*

- **-DMATLAB_ROOT**=<path> (default: searches common system install locations)
    - Specifies a custom root path, in which to search for MATLAB libraries and library headers.
      Note: This should only be used in special cases, where MATLAB is *installed in a non-standard location, or, where several versions of MATLAB are installed, and a particular version is favored.*

4.2 Environment Variables

The following environment variables affect the CMake build system:

- **MATLAB_ROOT**
    - The MATLAB root install directory used by CMake when searching for the MATLAB dependency. This directory should contain the MATLAB 'extern' directory. (Note that this variable should not normally need to be set, but can be used if the default CMake macros fail to find MATLAB on your system.)

4.3 Custom (Non-System) Qt4 Installations

When searching for a valid Qt4 installation, the OSKAR CMake build system queries the qmake binary in order to determine the location of the relevant libraries and headers. Therefore, all that is required to use a specific version of Qt4 is to add the location of the desired qmake binary to the beginning of the system search path.

5 Testing the Installation

5.1 Unit Tests

When building OSKAR from source, a number of unit test binaries will be created. The tests can be run by typing the following command from the build directory:

```
$ make test
```

All the unit tests should pass. If any fail, please report this by copying the terminal output and sending it, together with a description of the hardware in your machine and your version of OSKAR, to the email address oskar@oerc.ox.ac.uk.

5.2 Running the Example Simulation

It is also advised, with any fresh install of OSKAR, to run the example simulation described in the documentation found at http://www.oerc.ox.ac.uk/~ska/oskar to establish if a simple simulation behaves as expected.