OSKAR Installation Guide

1 Introduction

OSKAR can be installed by following the steps described below. A source code archive for Linux, and binary installer packages for macOS and Windows platforms are available to download from:
http://oskar.oerc.ox.ac.uk/

1.1 Linux

1.1.1 Installation

To install the OSKAR package on a Linux system, you will need to compile it from source. Ensure the dependencies have been installed as described in Dependencies (below). Then download the archive and follow the short steps in Building OSKAR.

If using a GPU on Linux, also be sure to install the latest NVIDIA driver from your distribution's repository, or from https://www.geforce.com/drivers

1.1.2 Uninstallation

To remove the OSKAR package on Linux, delete the applications, libraries and headers installed by the `make install` step. By default, these will be placed in:

- Applications: `/usr/local/bin/oskar*`
- Libraries: `/usr/local/lib/liboskar*`
- Headers: `/usr/local/include/oskar/`

1.2 macOS

1.2.1 Installation

To install the OSKAR package on macOS, download and open the disk image (DMG) file and drag the OSKAR.app bundle to your /Applications folder. After installation, double-click the OSKAR.app bundle to launch the GUI and set symbolic links to the applications in /usr/local/bin.

If using a GPU on macOS, also be sure to install the latest NVIDIA CUDA driver from http://www.nvidia.com/object/mac-driver-archive.html

1.2.2 Uninstallation

To remove the OSKAR package on macOS, delete the OSKAR.app bundle from your /Applications folder, and delete symbolic links to the applications by typing `rm -f /usr/local/bin/oskar*` from a terminal prompt.

1.3 Windows

1.3.1 Installation

To install the OSKAR package on 64-bit Windows, download and run the installer executable and follow the on-screen instructions. After installation, the GUI can be launched using the shortcut on the Desktop or Start Menu.

If using a GPU on Windows, also be sure to install the latest NVIDIA driver from
https://www.geforce.com/drivers

1.3.2 Uninstallation

To remove the OSKAR package on Windows, uninstall it from the list of software in Control Panel in the usual way.

2 Dependencies

If GPU acceleration is required, an NVIDIA GPU with CUDA 5.5 or later (and associated NVIDIA CUDA driver) must be installed. If the graphical user interface (GUI) is required, Qt must also be installed. Additionally, the casacore libraries must be present if Measurement Sets are to be exported.

The dependencies are:

- CMake (https://cmake.org), version >= 3.1
- [Optional] NVIDIA CUDA (https://developer.nvidia.com/cuda-downloads), version >= 5.5
- [Optional] Qt 5 (https://www.qt.io)
- [Optional] casacore (https://github.com/casacore/casacore), version >= 1.5.0

3 Building OSKAR

To build OSKAR, open a terminal and give the following commands:

```
$ mkdir build
$ cd build
$ cmake [OPTIONS] ../top/level/source/folder
$ make -j8
```

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.

3.1 Build Options

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

- **-DCUDA_ARCH="<arch>"** (default: all)
  - 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, 3.2, 3.5, 3.7, 5.0, 5.2, 6.0, 6.1, 6.2, 7.0 or ALL
  - ALL is for all Kepler, Maxwell and Pascal architectures (>= 3.0).
  - Multiple architectures can be specified by separating them with semi-colons.
- **-DCMAKE_INSTALL_PREFIX=<path>** (default: /usr/local/)
  - Path prefix used to install OSKAR (with make install)

3.1.1 Advanced Build Options

- **-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 if the CasaCore library 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 if the CasaCore headers in the system include path can't be used to build OSKAR.
- **-DCMAKE_PREFIX_PATH=<path>** (default: None)
  - Specifies a location in which to search for Qt 5. For example, if using Homebrew on macOS, this may need to be set to /usr/local/opt/qt5/
- **-DCMAKE_BUILD_TYPE=<release, debug, or RelWithDebInfo>** (default: release)
  - Build OSKAR with release flags, debug flags, or release flags with debugging symbols.
- **-DFIND_CUDA=ON|OFF** (default: ON)
  - Can be used to tell the build system not to find or link against CUDA.
- **-DNVCC_COMPILER_BINDIR=<path>** (default: None)
  - Specifies a nvcc compiler binary directory override. See nvcc help.
  - Note: This is likely to be needed only on macOS when the version of the compiler picked up by nvcc (which is related to the version of XCode being used) is incompatible with the current version of CUDA.
  - Set this to 'clang' on macOS if using GCC to build the rest of OSKAR.
- **-DFORCE_LIBSTDC++=ON|OFF** (default: OFF)
  - If ON forces the use of libstdc++ with the Clang compiler.
  - Note: Used for controlling linking behaviour when using clang or clang-omp compilers with dependencies which may have been compiled against libstdc++
- **-DBUILD_INFO=ON|OFF** (default: OFF)
  - If ON enables the display of diagnostic build information when running CMake.

3.2 Custom (Non-System) Qt Installations

If Qt 5 cannot be found from the default system paths, make sure to set CMAKE_PREFIX_PATH as described above.

4 Testing the Installation

4.1 Unit Tests

The unit test binaries can be run by typing the following command from the build directory:

```
$ ctest [--verbose]
```

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, your operating system version and your version of OSKAR, to the email address oskar@oerc.ox.ac.uk.

4.2 Running the Example Simulation

With any fresh install of OSKAR, we recommend running the example simulation to establish if a simple simulation behaves as expected.

Revision 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.
6	2014-09-30	Changed default CUDA_ARCH to all.
7	2015-02-28	Removed explicit CFITSIO and CBLAS dependencies. Clarified dependency requirements. Removed MATLAB references.
8	2017-10-30	Removed LAPACK dependencies. Marked CUDA and Qt dependencies as optional. Added note describing how to set a non-standard path for Qt 5. Added options to not find (or link against) CUDA. Added note to describe how to compile for multiple CUDA architectures. Updated required CMake version to 3.1. Added platform-specific guidelines.