
SIMDIS SDK Documentation

Release 1.6.0

US Naval Research Laboratory, Code 5773

Sep 11, 2017

1	About	1
2	Installation	3
3	Usage	5
3.1	Building SIMDIS SDK	5

CHAPTER 1

About

The SIMDIS Software Development Kit (SDK) is an open source C++ framework providing functionality to create 3D scenes consisting of objects, whose position and state change with time, that are placed relative to a geographic map. The primary modules provided by the SDK supply functionality to manage object position/state data, update object position/state data as time changes, and render 3D representations of objects within a 3D scene. Secondary modules provide utilities for parsing SIMDIS data files, performing coordinate system conversions, object state vector computations, and data metric calculations.

The intent of the SIMDIS SDK is to provide a C++ framework to be used by third party developers to incorporate functionality similar to that provided by the SIMDIS 3D Visualization and Analysis Toolkit within their own applications. The SIMDIS SDK serves as the basis for development of each new major SIMDIS release.

The SIMDIS SDK was developed by the Visualization Systems Integration branch of the Tactical Electronic Warfare Division at the U.S. Naval Research Laboratory. For more information visit:

<https://simdis.nrl.navy.mil>

CHAPTER 2

Installation

The SIMDIS SDK is provided as a package containing C++ source code for use with Windows and Linux operating systems. Precompiled binaries for use with Microsoft Visual Studio have also been provided for your convenience. See the Installation page for detailed information about supported systems/compilers, third party dependencies, and installation instructions.

The SIMDIS SDK source distribution comes with a number of example programs as well as a sample data set. The example which illustrates the most common usage of the SIMDIS SDK is the Platform Symbology example. Detailed tutorials describing SIMDIS SDK use will be added at a later date.

HTML based API documentation for the SIMDIS SDK source code can be found in the *Doc* subdirectory. This documentation describes the different software components provided by the SDK.

The SIMDIS SDK provides support for loading all 3D model formats supported by [OpenSceneGraph](#) (OpenFlight, 3D Studio, Wavefront OBJ, etc), loading the SIMDIS .db Terrain and Imagery files, and streaming terrain and imagery data from the ReadyMap server product developed by [Pelican Mapping](#) using [osgEarth](#).

The 3D models provided with the SIMDIS SDK sample data package have all been converted to the native Open-Scenegraph .ive binary file format, which embeds the 3D model's geometry data and associated texture files within a single, easy to distribute file.

Building SIMDIS SDK

Overview

The SIMDIS SDK installation instructions will cover the following topics:

- *Building SIMDIS SDK*
 - *Overview*
 - *Supported Systems/Compilers*
 - *Third Party Dependencies*
 - *Binary Installation*
 - *Source Installation*

- * *Installing the Source*
- * *Configuring the Build*
- * *Compiling the Source*
- * *Installing the Results*
- *System Environment Setup*
 - * *Environment Setup for the Sample Data Set*
 - * *Setting the System Search Path*
 - * *Setting up a Project that Uses the SIMDIS SDK*

Supported Systems/Compilers

Operating systems officially supported by the SIMDIS SDK:

- Windows Vista/7/8/10
- Red Hat Enterprise Linux 6/7

Compilers officially supported by the SIMDIS SDK:

- Windows compilers:
 - Microsoft Visual C++ 2010 (VC10)
 - Microsoft Visual C++ 2012 (VC11)
 - Microsoft Visual C++ 2013 (VC12)
 - Microsoft Visual C++ 2015 (VC14)
- Linux compilers:
 - GCC 4.x series

Third Party Dependencies

The SIMDIS SDK depends on the following third party libraries:

- OpenSceneGraph 3.4+
- osgEarth 2.8+
- protobuf 2.6+
- Qt 5.5+
- SQLite 3.8+

Other versions may also work, such as Qt 4.7. But we can only support the configurations that we build against.

Precompiled Windows binaries for all of the SIMDIS SDK dependencies can be obtained from the SIMDIS SDK project download page. Linux users may find that many of the SIMDIS SDK dependencies are available from the package repository for their Linux distribution.

To build the dependencies from source, obtain the source packages from the specified project websites and follow the included build and installation instructions.

Both OpenSceneGraph and osgEarth have additional third party dependencies. More information about these dependencies can be found at the OpenSceneGraph and osgEarth web sites. Linux users will likely find that all of these additional dependencies are available in the package repository for their Linux distribution. For your convenience, precompiled binaries for these dependencies have been included with the precompiled binary packages for Windows.

Binary Installation

If you are compiling the SIMDIS SDK from source, use the instructions in the next section. You may skip this section, proceeding to the next section entitled *Source Installation*.

To install the precompiled SIMDIS SDK binaries available from the SIMDIS SDK project page, simply extract the zip file containing the SIMDIS SDK files to a directory of your choosing. Make note of the location of the SIMDIS SDK installation. The documentation will refer to this directory as `<simdis-sdk-bin-dir>`. Whenever you see the value `<simdis-sdk-bin-dir>`, substitute the name of the directory on your system that contains the installation of the SIMDIS SDK.

Source Installation

If you do not plan on building the SIMDIS SDK from source, and instead plan to use one of the precompiled binary versions available from the SIMDIS SDK project page, you may skip this section and proceed to the next section entitled *System Environment Setup*.

Installing the Source

You must first extract the SIMDIS SDK from the zip file obtained from the SIMDIS SDK project page to a directory of your choosing. Make note of the location of the SIMDIS SDK. The documentation will refer to this directory as `<simdis-sdk-src-dir>`. Whenever you see the value `<simdis-sdk-src-dir>`, substitute the name of the directory on your system that contains the SIMDIS SDK source code.

Configuring the Build

The SIMDIS SDK makes use of the CMake Cross Platform Make build system for generating platform specific build files. If you do not already have CMake installed on your system, you must obtain and install it. CMake can be obtained from the [CMake project web page](#).

You will use the `cmake-gui` application to configure the SIMDIS SDK for building. Start `cmake-gui` and specify the source and build directories. You should specify `<simdis-sdk-src-dir>` as the source directory. You may also specify `<simdis-sdk-src-dir>` as the build directory, but it is recommended that you keep the build directory separate from the source directory, using a directory such as `<simdis-sdk-src-dir>/build`.

Now click the ‘Configure’ button and select your target build system. CMake will generate a list of variables describing the build properties, such as the locations of the 3rd party dependencies, and display them in a list. Many of 3rd party dependency locations will be initially marked as `NOTFOUND`. Specify the correct paths to the 3rd party dependencies and click ‘Configure’ again, then click Generate. You are now ready to build the SIMDIS SDK source code.

NOTE: If you have installed all of the third party dependencies in the same directory, the SIMDIS SDK CMake configuration should locate the files. You do not have to specify the location of each 3rd party library independently.

NOTE: You can specify the installation location for the SIMDIS SDK with the `CMAKE_INSTALL_PREFIX` variable.

Compiling the Source

To compile the source, simply start the build using the build system that you specified in the previous section. Microsoft Visual Studio users will find solution files for the SIMDIS SDK in the build directory specified to CMake. Make users will find a make file in the build directory specified to CMake.

Installing the Results

The CMake project files include support for installing the SIMDIS SDK files generated by the build process. Microsoft Visual Studio users can run the install process by building the `INSTALL` project that appears in the solution under the `CMakePredefinedTargets` folder. make users can run the install process by invoking the `make install` command.

Make note of the location of the SIMDIS SDK installation. The documentation will refer to this directory as `<simdis-sdk-bin-dir>`. Whenever you see this value `<simdis-sdk-bin-dir>`, substitute the name of the directory on your system that contains the installation of the SIMDIS SDK.

System Environment Setup

Environment Setup for the Sample Data Set

The example programs provided with the SIMDIS SDK require a sample data set for proper operation. This data set can be obtained from the SIMDIS SDK project page. Once you have downloaded the zip file containing the sample data set, you should extract it to a directory of your choosing. After the files have been extracted, you must create an environment variable named `SIMDIS_SDK_FILE_PATH`.

whose value is set to the path to the directory containing the sample data. Instructions for setting environment variables on Windows can be found [here](#).

Setting the System Search Path

The system library and program search paths must be setup to find the libraries and programs installed for the SIMDIS SDK. Windows users need to add `<simdis-sdk-bin-dir>/bin` to the `PATH` environment variable, so Windows knows where to look for the SIMDIS SDK DLLs and example programs. Instructions for setting environment variables on Windows can be found [here](#).

Linux users will need to add `<simdis-sdk-bin-dir>/bin` to their `PATH` environment variable to run the SIMDIS SDK examples from the command line without having to specify the full path to the example executables. The directory `<simdis-sdk-bin-dir>/lib` will need to be added to the system's library search path so that the system knows where to find the SIMDIS SDK shared libraries. This can be done with `RPATH`, `LD_LIBRARY_PATH`, or `ldconfig`. Some notes on shared libraries can be found [here](#).

NOTE: Use of `LD_LIBRARY_PATH` is considered bad practice/dangerous.

NOTE: If you only want to run the SIMDIS SDK example programs, you should not need to setup the library search path.

Setting up a Project that Uses the SIMDIS SDK

Projects using the SIMDIS SDK must add `<simdis-sdk-bin-dir>/include` to their include path and `<simdis-sdk-bin-dir>/lib` to their library path. Microsoft Visual Studio users can set these values on a project by project basis by specifying the appropriate values in the project settings, or can set these values globally by adding them to the "Include files" and "Library files" lists found in the "VC++ Directories" section of the "Projects

and Solutions” section of the Options dialog. The Options dialog is accessed through the Microsoft Visual Studio Tools menu.