Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Note

Compilation of SGeMS

Note no SAND/08/06

Authors Per Røe

Date October 2006

Norsk Regnesentral

Norsk Regnesentral (Norwegian Computing Center, NR) is a private, independent, non-profit foundation established in 1952. NR carries out contract research and development projects in the areas of information and communication technology and applied statistical modeling. The clients are a broad range of industrial, commercial and public service organizations in the national as well as the international market. Our scientific and technical capabilities are further developed in co-operation with The Research Council of Norway and key customers. The results of our projects may take the form of reports, software, prototypes, and short courses. A proof of the confidence and appreciation our clients have for us is given by the fact that most of our new contracts are signed with previous customers.

Norsk Regnesentral
Norwegian Computing Center
Postboks 114, Blindern
NO-0314 Oslo, Norway

Besøksadresse
Office address
Gaustadalléen 23
NO-0373 Oslo, Norway

Telefon · telephone
(+47) 22 85 25 00
Telefaks · telefax
(+47) 22 69 76 60

Internett · internet
www.nr.no
E-post · e-mail
nr@nr.no

Title	**Compilation of SGeMS**
Authors	**Per Røe**
Date	October
Year	2006
Publication number	SAND/08/06

Abstract

This document gives a step-by-step description of how to compile SGeMS. SGeMS is a open-source software package for geostatistical modeling, developed at Stanford University.

Most of the recipe is generally valid, but the part describing the linking with QT describes the setup of QT at NR, and the description on how to link with QT will only work when connected with NRs network.

Keywords	SGeMS
Target group	Internal
Availability	Open
Project number	808002
Research field	Geostatistic
Number of pages	9
© Copyright	Norsk Regnesentral

Contents

1 Preconditions

To be able to compile SGeMS the following software packages and libraries are needed:

1. The SGeMS source code can be downloaded from http://sgems.sourceforge.net/. (http://prdownloads.sourceforge.net/sgems/SGeMS-1.4.1.tgz?download)

2. GsTL can be downloaded from http://gstl.sourceforge.net/ (http://prdownloads.sourceforge.net/gstl/GsTL-1.2.1.tgz?download)

3. QT v.3 for Windows is installed at G:\felles\biblioteker\Qt\Win32\3.3.6. Notice that this is a commercial licensed version of QT. SGeMS requires (at least for SGeMS 1.4.1) QT v.3, while only QT v.4 and newer is available in an open-source version for Windows. The open-source version of QT for Windows also only works with MinGW, not with Visual Studio.

   To link QT with SGeMS, QT has to be compiled dynamically linked. Support for OpenGL and STL is also needed.

   If access to the debug symbols in QT is needed when debugging SGeMS, a debug version of QT should also be installed.

4. Python can be downloaded from http://www.python.org/ (http://www.python.org/ftp/python/2.4.3/python-2.4.3.msi)

5. Coin3D, SimVoleon and SoQT can be downloaded from http://www.coin3d.org/ (http://ftp.coin3d.org/coin/bin/win32/all/Coin-2.4.5-bin-msvc7.zip, http://ftp.coin3d.org/coin/src/all/SoQt-1.3.0.zip, http://ftp.coin3d.org/coin/bin/win32/all/SIMVoleon-v2.0.1-install.exe)

2 Installation

2.1 Python

Run python-2.4.3.msi. If you don't have any special preferences, the standard choices can be used.

2.2 Coin3D

Unzip Coin-X.X.X-bin-msvc7.zip on a suitable place, e.g. C:\Coin.

SimVoleon is installed by running the installation program. Install SimVoleon at the same place as Coin was installed, e.g. C:\Coin.

SoQT is only available as source-code and must be compiled:

1. Unpack SoQt-X-X-X.zip on a suitable place, e.g. C:\tmp.

2. Set a couple of environment variables: (This is done by opening System in the control panel. In the advanced panel press environment variables.)

   - Set QTDIR to G:\felles\biblioteker\Qt\Win32\3.3.6

   - Set COINDIR to the directory where Coin was installed, e.g. C:\Coin.

3. A project file for Visual Studio 2003 is located at \msvc7\soqt1.sln.

4. Change the version number of the qt dll in project properties under Linker->Input->Additional dependencies:

   - Replace qt-mt331.lib with qt-mt336.lib.

5. Compile and build SoQT using the DLL (release) configuration. soqt1.dll will be built and placed in the Coin directory.

2.3 GsTL

Unpack GsTL to the directory that will be used for compilation of SGeMS, e.g. C:\SGeMS.

Since GsTL is a template library, no compilation is necessary.

2.4 SGeMS

Unpack SGeMS to the same directory as where GsTL was placed, e.g. C:\SGeMS.

The following environment variables must be set, in addition to the environment variables set when compiling SoQT:

- PYTHON24_DIR must be set to the directory where Python is installed, e.g. C:\Python24.

- SGEMS_BUILD must be set to the root directory for the SGeMS source code, e.g. C:\SGeMS.

- COIN3D_DIR must be set to the directory where Coin is installed, e.g. C:\Coin.

To be able to compile the debug version, the following environment variable must point to the location of QT, or a debug version of QT:

- Set QTDIR_DBG to G:\felles\biblioteker\Qt\Win32\3.3.6

A project file for Visual Studio 2003 is located in the WinGsTLAppli directory.

Set the configuration to Release.

For all the projects in the workspace, qt-mt335.lib must be replaced with qt-mt336.lib under Linker->Input->Additional dependencies. (If the qt-version was replaced with an environment variable, this would not been necessary. This would save much editing during installation.)

A project file for Visual Studio 2003 is located in the WinGsTLAppli directory.

Add $(SGEMS_BUILD) to the list of additional include directories under C/C++ in project properties.

GsTLAppli\geostat\filtersim_std\partition.cpp must be added as a source file in the GsTLAppli_geostat project.

Compile the solution. (Build Solution)

Copy the plugins directory from C:\SGeMS to the output directory C:\SGeMS\GsTLAppli\lib\win32.

Copy coin2.dll, soqt1.dll and simvoleon2.dll from C:\Coin\bin to C:\SGeMS\GsTLAppli\lib\win32.

Copy qt-mt336.dll from G:\felles\biblioteker\Qt\Win32\3.3.6\bin to C:\tmp\SGeMS\GsTLAppli\lib\win32.

Make sure that GsTLwidgets.dll from the output directory is copied to $(QTDIR)\plugins\designer. This is only needed to be done once per QT installation.

You should now be able to run SGeMS.

3 Debugging

If the previous recipe is followed, compilation of the debug version of SGeMS should only be a matter of recompiling after changing the configuration to Debug, but to be able to run the debug version from Visual Studio the following should be done in addition to copying the same files as above to the debug-output directory:

- Set Working directory under debugging in project properties for GsTLAppli_actions to the debug-output directory, e.g. C:\SGeMS\GsTLAppli\lib\win32\debug.

- If SGeMS already is installed on the computer, you might encounter some problems with the loading of Colormap. To prevent this you can remove the GSTLAPPLIHOME environment variable, but this might prevent the previously installed version of SGeMS from working properly.

- Since the latest version of SimVoleon is compiled with Visual Studio 6, the debug-version is dependent on a version of the standard library that is not present if you only have a newer version of Visual Studio. The solution for this is to link the debug version of SGeMS with the release version of Coin 3D, SimVoleon and SoQT. This is done by replacing coin2d.dll, simvoleon2d.dll and soqt1d.dll with respectively coin2.dll, simvoleon2.dll and soqt1.dll in project properties under Linker-> Input->Additional dependencies for all the projects in the workspace.