

C++ UI design for NAG Optimization Modelling Suite Users' guide

Contents

- [1. Introduction](#)
- [2. Installation](#)
- [3. Building and Running](#)
- [4. Doxygen Documentation](#)
- [5. Writing Programs](#)
- [6. Contacts](#)

1. Introduction

This document is essential reading for every user of the C++ User Interface for the NAG Optimization Modelling suite. It provides information on installation and usage of this particular Interface. For all questions concerning the installation or usage of the NAG Library or dco/c++, which the Interface relies on, it is recommended to read the corresponding Users'/Installer's Notes.

2. Installation

Install and activate the correct versions of NAG Library ([nll6i285bl](#)) and dco/c++ ([dcl6i37ngl_v370](#)) according to corresponding Installer's notes found on the linked pages. (Activation keys have been included with the Interface, under /license/nag.key.)

After verifying that the softwares have been successfully installed, acquire the Interface from the [Gitlab repository](#) either by downloading it directly from the link, or by running the following:

```
# git clone https://git.rwth-aachen.de/tranmankhang1705/  
# nag-optimization-modelling-suite-ui.git
```

3. Building and Running

The location in which the acquired directory is placed will thus be referred to as `${REPO_DIR}`. In an arbitrary directory, create a new directory 'build' and move there:

```
# mkdir build
# cd build
```

Run `cmake` to create the necessary make files, changing `${HOME}`, if an alternative installation directory for the NAG Library and `dco/c++` was chosen beforehand:

```
# cmake ${REPO_DIR} -DNAG_dco_cpp_DIR=${HOME}/NAG/dcl6i37n9l_v370/
# -DNAG_Library_DIR=${HOME}/NAG/nll6i285bl/
```

To build, run:

```
# make
```

To run the examples, case studies, and the code tests, run:

```
# ctest
```

It is possible to build individual examples, case studies, and code tests by setting a target:

```
# make examples.xxx
# make tests.xxx
# make caseStudies.xxx
```

to compile an arbitrary program from the `/examples`, `/tests`, or `/caseStudies` folder respectively.

To build and execute an example (in this case, `e04kf`), run the following:

```
# make examples.e04kf
# ./examples/examples.e04kf
```

To build and execute your own programs, create a sub directory with an arbitrary `${SUB_DIR_NAME}` within `/examples` and place your programs there. Thus, run:

```
# make examples.${SUB_DIR_NAME}
# ./examples/examples.${SUB_DIR_NAME}
```

Note: A new sub directory is to be created for every program, as they can only contain one program each.

4. Doxygen Documentation

The Doxygen documentation contains comprehensive information about every class and function of the Interface. To generate the Doxygen documentation, run the following commands:

```
# cd ${REPO_DIR}/include
# doxygen -g Doxyfile
# doxygen Doxyfile
```

Doxygen documentation in the form of .html and .tex file are to found within /include/html and /include/latex respectively.

5. Writing programs

Before starting to write your own programs, it is recommended to first look at the examples found within /examples, which are basic self-explanatory usages of the three optimisation solvers present in the Interface.

Alternatively, look inside /caseStudies to see the Interface in work on problems more complex.

For any questions concerning the classes and functions, the Doxygen documentation is to be consulted.

When solving problems with **non-linear constraint functions**, special attention is to be made. The dimension of the output array in such a function must be manually specified. For example, for $g : \mathbb{R}^2 \rightarrow \mathbb{R}^1$, the following is to be written:

```
auto constraint_problem = [] (auto const &x, auto &y)
{
    // manually specified
    y.resize(1);
    y[0] = 12.0 * x[0] + 11.9 * x[1]
        - 1.645 * sqrt(0.28 * pow(x[0], 2) + 0.19 * pow(x[1], 2));
};
```

To better understand this footnote, check out /examples/e04st as well as /caseStudies/st_XXX.

6. Contacts

For any question not covered in the guide, please contact one of our group members. We would be delighted to be able to assist you.

Members of group 6:

Tran, Man Khang at khang.tran@rwth-aachen.de

Feldman, Maksim at maksim.feldman@rwth-aachen.de

Valiyev, Ziya at ziya.valiyev@rwth-aachen.de

Korkin, Konstantin at konstantin.korkin@rwth-aachen.de

Huang, Yifei at yifei.huang@rwth-aachen.de

Supervisor:

Lotz, Johannes at lotz@stce.rwth-aachen.de