# Releasing the DikesOvertopping kernel.

This document describes the steps in making a DikesOvertopping release.

# Preliminaries

The target structure for delivery is as follows.

A screenshot of a computer program

Description automatically generated with medium confidence

From TeamCity the following configurations are available. Since some tests seem to be platform and compiler dependent, not all tests are run for all build configurations.

A screenshot of a computer

Description automatically generated

The following table shows the contents of the delivery.



The packaging of the contents from the TeamCity configurations into the delivery has been automated in TeamCity as well: under the configuration with the name *Create delivery*. When deliverables under the columns *Testanalyse* and *Repository* are made available to TeamCity as well, the complete delivery packaging can be automated.

# Preparations

## New version number

Update the version number such that each release has a unique version number.

We use version of the form xx.y.z with the first digits *xx* being the year (modulo 100) of release, the second digit y for a release with changes in the interface or new features (compared to a release within the same year) and the last digit *z* for small updates/bug fixes. An example of a version number would be 23.1.1.

## Select dependencies

DikesOvertopping depends on Fortran-Common-Library and is built using a Fortran compiler.

Ideally a release of DikesOvertopping uses the latest release of Fortran-Common-Library and the newest compiler available on TeamCity.

# Creating the release

The binaries are created on TeamCity in four configurations: windows 32-bit, windows 64-bit, Linux 64-bit with Intel compiler and Linux 64-bit with GNU compiler. The GNU build is only used for performing more checks during compile time in comparison to the Intel compiler, and its artifacts are not released.

The artifacts of the three other builds are copied into the release folder.

When this last step is committed, a tag is created from this commit in Git and the resulting build of the main branch in TeamCity is given a tag and is pinned.