Catkin

Catkin is a collection of cmake macros and associated python code used to build some parts of ROS

For a good background read the doc at htt: <http://catkin-tools.readthedocs.org/en/latest/>

# Catkin command line tools

Quick overview of useful catkin commands:

catkin clean --build --devel

just clean a single package:

catkin clean PKGNAME

Building ros packages

catkin build

Setting and unsetting CMake options:

catkin config --cmake-args -DENABLE\_CORBA=ON -DCORBA\_IMPLEMENTATION=OMNIORB

catkin config --no-cmake-args

Create “Debug” and “Release” profiles and then build them in independent build and devel spaces:

catkin config --profile debug -x \_debug --cmake-args -DCMAKE\_BUILD\_TYPE=Debug

catkin config --profile release -x \_release --cmake-args -DCMAKE\_BUILD\_TYPE=Release

catkin build --profile debug

catkin build --profile release

build specific packages (however this will build dependency packages)

build PKG [PKG ...]

**Initialize catkin workspace**

source /opt/ros/indigo/setup.bash # Source ROS indigo to use Catkin

mkdir -p /tmp/quickstart\_ws/src # Make a new workspace and source space

cd /tmp/quickstart\_ws # Navigate to the workspace root

catkin init

**Build one catkin project**

If you’re only interested in building a single package in a workspace, you can also use the --no-deps option along with a package name. This will skip all of the package’s dependencies, build the given package, and then exit.

catkin build PKG --no-deps # Build PKG only

**Use some but not all of the processor cores**

You can control the number of build jobs. Typically a job controller is used and all the cores are assigned make files, which halts GUI interaction.

You can control the maximum number of packages allowed to build in parallel by using the -p or --parallel-packages option and you can change the number of make jobs available with the -j or --jobs option.

To disable the job server, you can use the --no-jobserver option.

catkin build PKG --no-deps -p 7 # Build PKG only, and use only 7 or 8 cores

**Building With Warnings**

It can sometimes be useful to compile with additional warnings enabled across your whole catkin workspace. To achieve this, use a command similar to this:

$ catkin build -v --cmake-args -DCMAKE\_C\_FLAGS="-Wall -W -Wno-unused-parameter"

This command passes the -DCMAKE\_C\_FLAGS=... argument to all invocations of cmake.

# Adding gtest in ROS for unit testing

Mon 21 Nov 2016 04:47:50 PM EST

<http://catkin-tools.readthedocs.io/en/latest/verbs/catkin_build.html> has a section on using catkin with gtest.

## Version Info

This is the version information at the time of this documentation (changes often).

ROS VERSION: Indigo

CATKIN VERSION: 0.4.2 Using Python 2.7.6 (default, Jun 22 2015, 17:58:13) [GCC 4.8.2]

UBUNTU VERSION: cat /etc/lsb-release

DISTRIBID=Ubuntu

DISTRIBRELEASE=14.04

DISTRIBCODENAME=trusty

DISTRIBDESCRIPTION="Ubuntu 14.04.5 LTS"

## Summary of Commands

General build all packages with DEBUG flag:

catkin build -DCMAKEBUILDTYPE=Debug

Build all packages test

catkin runtests nistrobotsnc

Run all package tests:

catkin test

Run package nist\_robotsnc tests

catkin test nist\_robotsnc

Check for failing tests: run executable built by catking, in our case, it is located at:

/usr/local/michalos/nistfanuc*ws/devel/lib/nist*robotsnc/conversionTest

You can embed "std::cout <<" output to be displayed in addition to the CTest program. Running generates a lot of console output, as shown below:

> /usr/local/michalos/nistfanuc\_ws/devel/lib/nist\_robotsnc/conversionTest

[==========] Running 4 tests from 1 test case.

[----------] Global test environment set-up.

[----------] 4 tests from TFEigenConversions

[ RUN ] TFEigenConversions.tf\_eigen\_vector

tf\_eigen\_vector test

[ OK ] TFEigenConversions.tf\_eigen\_vector (0 ms)

[ RUN ] TFEigenConversions.tf\_eigen\_quaternion

tf\_eigen\_quaternion test

[ OK ] TFEigenConversions.tf\_eigen\_quaternion (0 ms)

[ RUN ] TFEigenConversions.tf\_eigen\_transform

tf\_eigen\_transform test1

[ OK ] TFEigenConversions.tf\_eigen\_transform (0 ms)

[ RUN ] TFEigenConversions.eigen\_tf\_transform2

tf\_eigen\_transform2 test

/usr/local/michalos/nistfanuc\_ws/src/nist\_robotsnc/test/conversiontests.cpp:108: Failure

The difference between t1.getBasis()[i][j] and affine.matrix()(i,j) is 0.84449695716742035, which exceeds 1e-6, where

t1.getBasis()[i][j] evaluates to -0.84449695716742035,

affine.matrix()(i,j) evaluates to -6.8950132430539161e-310, and

1e-6 evaluates to 9.9999999999999995e-07.

[ FAILED ] TFEigenConversions.eigen\_tf\_transform2 (1 ms)

[----------] 4 tests from TFEigenConversions (1 ms total)

[----------] Global test environment tear-down

[==========] 4 tests from 1 test case ran. (1 ms total)

[ PASSED ] 3 tests.

[ FAILED ] 1 test, listed below:

[ FAILED ] TFEigenConversions.eigen\_tf\_transform2

1 FAILED TEST

## Building C++ gtest in ROS

Several Steps are involved in :

1. Create ./test/conversiontests.cpp
2. Add to CMakeLists.txt catkin*add*gtest(conversionTest test/conversiontests.cpp)
3. catkin provides the run*tests target which runs all the tests. You can tab complete run*tests\_ to get a target for each test you have registered which only runs that test.

catkin runtests

1. Run tests (cd to the test executable built with catkin runtests) and execute:

> /usr/local/michalos/nistfanuc\_ws/devel/lib/nist\_robotsnc/conversionTest

1. Unit test output:

michalos@woodsy:nistfanuc\_ws> cd /usr/local/michalos/nistfanuc\_ws/devel/lib/nist\_robotsnc

michalos@woodsy:nist\_robotsnc> ./conversionTest

[==========] Running 4 tests from 1 test case.

[----------] Global test environment set-up.

[----------] 4 tests from TFEigenConversions

[ RUN ] TFEigenConversions.tf\_eigen\_vector

[ OK ] TFEigenConversions.tf\_eigen\_vector (0 ms)

[ RUN ] TFEigenConversions.tf\_eigen\_quaternion

conversionTest:

. . .

### Sample Problem: Assertion failed in Conversions.h!

michalos@woodsy:nistfanuc\_ws> cd /usr/local/michalos/nistfanuc\_ws/devel/lib/nist\_robotsnc

michalos@woodsy:nist\_robotsnc> ./conversionTest

[==========] Running 4 tests from 1 test case.

[----------] Global test environment set-up.

[----------] 4 tests from TFEigenConversions

[ RUN ] TFEigenConversions.tf\_eigen\_vector

[ OK ] TFEigenConversions.tf\_eigen\_vector (0 ms)

[ RUN ] TFEigenConversions.tf\_eigen\_quaternion

conversionTest: /usr/local/michalos/nistfanuc\_ws/src/nist\_robotsnc/include/nist\_robotsnc/Conversions.h:71: Convert: Assertion `0' failed.

Aborted (core dumped)