

Création du package : Localisation

1. Création du workspace

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace
$ cd ~/catkin_ws/
$ catkin_make
$ source devel/setup.bash
```

2. Création d'un package

```
catkin_create_pkg beginner_tutorials std_msgs rospy roscpp
```

Dans cet exemple, On crée le package `beginner_tutorials` grâce à la commande « `catkin_create_pkg` ». On y précise qu'il dépend de :

- `std_msgs`
- `rospy` = utilisation de python
- `roscpp` = utilisation de C++

Dans le cas du projet de localisation, il faudra utiliser :

```
catkin_create_pkg localisation roscpp
```

Dans le cas général, la création de package avec catkin est de la forme :

```
# This is an example, do not try to run this
# catkin_create_pkg <package_name> [depend1] [depend2] [depend3]
```

3. Construction du package

Ouvrir un terminal dans le workspace catkin et utiliser la commande :

```
$ catkin_make
```

Dans le cas où le code source est à un endroit différent, supposé être `my_src`, alors appeler `catkin_make` comme ceci :

```
$ catkin_make --source my_src
```

Par la suite, on vérifie qu'il y a bien le dossier `beginner_tutorials/` (donc notre package) et le fichier `CMakeLists.txt` dans le chemin `catkin_ws/src` :

```
$ cd ~/catkin_ws/  
$ ls src  
beginner_tutorials/  CMakeLists.txt@
```

Dans notre cas, nous aurons :

```
$ cd ~/catkin_ws/  
$ ls src  
localisation/  CMakeLists.txt
```

Il faut alors à nouveau utiliser la commande :

```
$ catkin_make
```

4. Utilisation du programme en C++ et configuration du CMakeLists.txt

Le package localisation a été créé, mais pour le moment il ne contient aucun exécutable (node). Il faut donc ajouter le programme localisation.cpp dans le chemin : ~/catkin_ws/src/localisation/src

Afin de compiler le programme de localisation il faut paramétrer le fichier CMakeLists.txt sous le chemin ~/catkin_ws/src/localisation/

Cela doit être fait dans la partie « Build » du fichier.

```
#####  
## Build ##  
#####  
  
## Specify additional locations of header files  
## Your package locations should be listed before other locations  
include_directories(  
  # include  
  ${catkin_INCLUDE_DIRS}  
  ${ALSA_LIBRARIES}  
  ${ALSA_CFLAGS_OTHER}  
  ${ALSA_INCLUDE_DIRS}  
)  
  
## Declare a C++ library  
# add_library(${PROJECT_NAME}  
#   src/${PROJECT_NAME}/localisation.cpp  
# )  
  
## Add cmake target dependencies of the library  
## as an example, code may need to be generated before libraries  
## either from message generation or dynamic reconfigure  
# add_dependencies(${PROJECT_NAME} ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})  
  
## Declare a C++ executable  
## With catkin_make all packages are built within a single CMake context  
## The recommended prefix ensures that target names across packages don't collide  
add_executable(${localisation}_loca_exe src/Localisation_v2.0.0.cpp)  
  
target_link_libraries (_loca_exe ${ALSA_LIBRARIES})  
target_compile_options(_loca_exe PUBLIC ${ALSA_CFLAGS_OTHER})  
target_include_directories(_loca_exe PUBLIC ${ALSA_INCLUDE_DIRS})
```

En ajoutant cela, l'exécutable _loca_exe sera créé à partir du fichier source Localisation_v2.0.0.cpp. On y précise aussi les dépendances pour ALSA.

Une fois tout cela réalisé, il ne reste plus qu'à utiliser catkin_make à nouveau, dans le répertoire catkin_ws :

```
~/catkin_ws$ catkin_make
```

5. Exécution du programme

Lancer ros avec la commande roscore :

```
$ roscore
... logging to /home/quentin/.ros/log/cd6566ac-a965-11ea-b485-0800270dbaf5/
roslaunch-quentin-VirtualBox-7157.log

Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://quentin-VirtualBox:41047/
ros_comm version 1.12.14

SUMMARY
=====

PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.14

NODES
auto-starting new master
process[master]: started with pid [7167]
ROS_MASTER_URI=http://quentin-VirtualBox:11311/

setting /run_id to cd6566ac-a965-11ea-b485-0800270dbaf5
process[rosout-1]: started with pid [7180]
started core service [/rosout]
```

Puis dans un autre terminal (en laissant le terminal roscore ouvert), d'abord préciser la source avant de lancer l'exécution du node `_loca_exe` venant du package localisation :

```
$ source ~/catkin_ws/devel/setup.bash
$ rosrun localisation _loca_exe
```

Le programme doit alors démarrer.