

版本管理

一、版本号生成配置

二、在代码中使用版本号

一、版本号生成配置

1. 将以下代码添加到自己工程的 CMakeLists.txt 里面

```
1 #####
2 ##          auto git version set          ##
3 #####
4 find_package(Git REQUIRED)
5 execute_process(
6     WORKING_DIRECTORY ${CMAKE_CURRENT_SOURCE_DIR}
7     COMMAND ${GIT_EXECUTABLE} describe --tags --dirty --long
8     TIMEOUT 5
9     RESULT_VARIABLE exit_code
10    OUTPUT_VARIABLE GIT_VERSION
11    OUTPUT_STRIP_TRAILING_WHITESPACE
12 )
13
14 if(NOT ${exit_code} EQUAL 0)
15     message(WARNING "git describe failed, please check the git tag set is ok.")
16     set(GIT_VERSION "0.0.0")
17 endif()
18
19 configure_file(
20     ${CMAKE_CURRENT_SOURCE_DIR}/include/version.h.in
21     ${CMAKE_CURRENT_SOURCE_DIR}/include/version.h
22     @ONLY
23 )
24
25 message(STATUS "building from git tag ${GIT_VERSION}")
```

2. 在 include 文件夹下创建 version.h.in 的文件，并加入以下内容

C | 复制代码

```
1 #ifndef VERSION_H
2 #define VERSION_H
3
4 #define VERSION "@GIT_VERSION@"
5
6 #endif
```

3. 在自己的仓库中，将 version.h 添加到 .gitignore 中
4. 自己的仓库中至少需要一个 tag，这里为了规范，全部命名为 x.y.z 的形式，例如 1.1.0。参照下面命令打 tag

Shell | 复制代码

```
1 git tag -a x.y.z <commit-id>
2 # 例如:
3 git tag -a 1.0.0 a55e5bd0b2fea15e0f0aaeb75a6a6b3d499f5624
```

二、在代码中使用版本号

示例代码如下

C++ | 复制代码

```
1 #include <ros/ros.h>
2 #include "version.h"
3
4 int main(int argc, char const *argv[])
5 {
6     ROS_INFO_STREAM("current version is: " << VERSION);
7     return 0;
8 }
```