

ros_install

~~~~~创建deb包~~~~~

## 1、安装依赖工具

```
sudo apt-get install python-bloom fakeroot
```

## 2、修改CMakeLists.txt文件，在后面添加install相关内容

举例如下，根据需要打包的具体文件修改添加

python项目：

```
## Install scripts
install(PROGRAMS scripts/talker.py scripts/listener.py
  DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
)
install(FILES
  # myfile1
  # myfile2
  DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}
)
```

c++项目：

```
## Install exec
install(TARGETS cloudplatform_control_node cloudplatform_sdk_control_node visible_cloudplat_node
  infrared_cloudplat_node hostpc_cloudplat_node camera_status_node tianboir_sdk_control_node
  RUNTIME DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
)
## Mark other files for installation (e.g. launch and bag files, etc.)
install(FILES
  launch/cloudplatform_correlation.launch
  launch/robot_camera.urdf
  # myfile1
  # myfile2
  DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}
)
```

## 3、修改package.xml文件

只需要修改该文件中的version字段，例如下面version为2021.2.2

版本号中不要以0开头（除非就是0），不要写成如2021.02.02，否则打包时会报如下警告“WARNING: Package "imu\_383\_driver" does not follow the version conventions. It should not contain leading zeros (unless the number is 0).”：

```
<?xml version="1.0"?>
<package format="2">
  <name>imu_383_driver</name>
  <version>2021.2.2</version>
  <description>The imu_383_driver package</description>
</package>
```

4、执行打包命令(注意16.04和18.04中的ros版本差别)

a、bloom-generate rosdebian --os-name ubuntu --ros-distro melodic

```
chenwei@chenwei-N85-87HP6:~/catkin_dandt_ws/src/imu_383_driver$ bloom-generate rosdebian --os-name ubuntu --ros-distro melodic
==> Generating debs for ubuntu:bionic for package(s) ['imu_383_driver']
No homepage set, defaulting to ''
No historical releaser history, using current maintainer name and email for each versioned changelog entry.
No CHANGELOG.rst found for package 'imu_383_driver'
Package 'imu_383_driver' has dependencies:
Run Dependencies:
  rosdep key      => bionic key
  roscpp           => ['ros-melodic-roscpp']
  rospy            => ['ros-melodic-rospy']
  std_msgs         => ['ros-melodic-std-msgs']
Build and Build Tool Dependencies:
  rosdep key      => bionic key
  roscpp           => ['ros-melodic-roscpp']
  rospy            => ['ros-melodic-rospy']
  std_msgs         => ['ros-melodic-std-msgs']
  catkin           => ['ros-melodic-catkin']
==> Placing templates files in the 'debian' folder.
==> In place processing templates in 'debian' folder.
Expanding 'debian/compat.em' -> 'debian/compat'
Expanding 'debian/changelog.em' -> 'debian/changelog'
Expanding 'debian/rules.em' -> 'debian/rules'
Expanding 'debian/copyright.em' -> 'debian/copyright'
Expanding 'debian/source/format.em' -> 'debian/source/format'
Expanding 'debian/source/options.em' -> 'debian/source/options'
Expanding 'debian/control.em' -> 'debian/control'
chenwei@chenwei-N85-87HP6:~/catkin_dandt_ws/src/imu_383_driver$
```

5、重新打包

a、如果代码有修改，修改代码完成后，修改package.xml中的版本号

b、然后执行catkin\_make --pkg \*\*\*, 编译修改的package

c、删除package的src同级目录下的obj-x86\_64-linux-gnu和debian文件夹

b、fakeroot debian/rules binary

该指令执行完成后可在package同级目录下看到生成的deb包

```
chenwei@chenwei-N85-87HP6:~/catkin_dandt_ws/src$ ls *.deb *.ddeb
ros-melodic-imu-383-driver_0.0.0-0bionic_amd64.deb  ros-melodic-imu-383-driver-dbgshim_0.0.0-0bionic_amd64.ddeb
ros-melodic-imu-383-driver_2021.2.2-0bionic_amd64.deb  ros-melodic-imu-383-driver-dbgshim_2021.2.2-0bionic_amd64.ddeb
chenwei@chenwei-N85-87HP6:~/catkin_dandt_ws/src$
```

5、更新版本重新打包

a、如果代码有修改，修改代码完成后，修改package.xml中的版本号

b、然后执行catkin\_make --pkg \*\*\*, 编译修改的package

c、删除package的src同级目录下的obj-x86\_64-linux-gnu和debian文件夹

```
chenwei@chenwei-N85-87HP6:~/catkin_dandt_ws/src/imu_383_driver$ ls
CMakeLists.txt  debian  include  launch  obj-x86_64-linux-gnu  package.xml  src
chenwei@chenwei-N85-87HP6:~/catkin_dandt_ws/src/imu_383_driver$ rm debian/ -rf
chenwei@chenwei-N85-87HP6:~/catkin_dandt_ws/src/imu_383_driver$ rm obj-x86_64-linux-gnu/ -rf
chenwei@chenwei-N85-87HP6:~/catkin_dandt_ws/src/imu_383_driver$
```

d、重新执行第4步

~~~如果要打包的ros package依赖其他ros package，执行以下步骤~~~

6、打包并安装依赖(例如依赖yidamsg包)

在deb打包本机执行前面1-4步，如果只是有消息定义的包，不需要更改CMakeLists.txt，安装完成后可以在/opt/ros相关目录看到安装文件，如下图：

```
chenwei@chenwei-N85-87HP6:~$ ls /opt/ros/melodic/share/yidmsg/
snake/ package.xml srv/
chenwei@chenwei-N85-87HP6:~$ ls /opt/ros/melodic/share/yidmsg/msg
AHRS_data.msg DemoResult.msg Image.msg manualControlParameters.msg road_dis.msg TaskControlStatus.msg task_status.msg yuntai_preset_control.msg
AlarmMessage.msg Detect_Result.msg Image_Road.msg MonitorPoint.msg RoI_2d.msg TaskDeliver.msg task_stop.msg yuntai_reset_control.msg
battery.msg fly_ctl.msg ImageTest.msg motor_control.msg RoI_2d_position.msg TaskExecuteStatus.msg transfer.msg
captureImage.msg fuzzy_task.msg ImmediatelyTaskStatusWeb.msg nrcar_status.msg RoI_3d_position.msg taskJsonStatus.msg ultrasound.msg
car_status.msg GPS_data.msg InspectedImage.msg onebutton.msg RoI_can.msg task_list.msg UrgencyTask.msg
ControlMode.msg guidao.msg InspectedResult.msg OrigineWeather.msg RouteStatus.msg TaskParameters.msg wall_face_recognition_result.msg
cradle.msg home.msg InspectedResultWeb.msg pointcloud_color.msg run_status.msg weather.msg
cut.msg Image_data.msg LiveImage.msg PreImage.msg start_back_point.msg task_pose.msg Yida_pose.msg
DBTableUpdate.msg Image_ID.msg Log.msg PreImage.msg take_photo.msg TaskRealStatusWeb.msg yuntai_control.msg

chenwei@chenwei-N85-87HP6:~$ ls /opt/ros/melodic/share/yidmsg/srv
CameraChange.srv dataCenterService.srv GoHouse.srv Observe.srv Point_Projection.srv taskControlParameters.srv wall_do_action.srv wall_play_audio.srv
CameraControl.srv Fillrone.srv isinAutoCharger.srv OutHouse.srv RfidScanStatus.srv TaskControl.srv wall_do_patrol.srv wall_task_status.srv
ControlDoor.srv GetBattery.srv Object_Reognition.srv pathPlanningService.srv RobotStop.srv TaskList.srv wall_go_to_position.srv wall_tts_audio.srv

chenwei@chenwei-N85-87HP6:~$ ls /opt/ros/melodic/include/yidmsg/
AHRS_data.h Filrone.h isinAutoCharger.h pointcloud_color.h taskControlParametersRequest.h wall_do_actionResponse.h
AlarmMessage.h FilroneRequest.h isinAutoChargerRequest.h Point_Projection.h taskControlParametersResponse.h wall_do_patrol.h
battery.h fly_ctl.h isinAutoChargerResponse.h Point_ProjectionRequest.h TaskControlRequest.h wall_do_patrolRequest.h
CameraChange.h fuzzy_task.h LiveImage.h Log.h point_status.h TaskControlResponse.h wall_do_patrolResponse.h
CameraChangeRequest.h GetBattery.h manualControlParameters.h PreImage.h point_status.h TaskControlStatus.h wall_face_recognition_result.h
CameraControl.h GetBatteryRequest.h MonitorPoint.h RfidScanStatus.h TaskDeliver.h TaskExecuteStatus.h wall_go_to_position.h
CameraControlRequest.h GetBatteryResponse.h motor_control.h RfidScanStatusRequest.h taskJsonStatus.h wall_go_to_positionResponse.h
CameraControlResponse.h GoHouse.h nrcar_status.h RfidScanStatusResponse.h task_list.h wall_play_audio.h
captureImage.h GoHouseRequest.h Object_Reognition.h road_dis.h TaskList.h wall_play_audioRequest.h
car_status.h GoHouseResponse.h Object_ReognitionRequest.h RobotStop.h TaskListRequest.h wall_play_audioResponse.h
ControlDoor.h Image_data.h Object_ReognitionResponse.h RobotStopRequest.h TaskListResponse.h wall_task_status.h
ControlDoorRequest.h Image_ID.h Observe.h RoI_2d_position.h TaskParameters.h wall_task_statusRequest.h
ControlDoorResponse.h Image_Road.h ObserveRequest.h RoI_2d.h taskPlanStatus.h wall_task_statusResponse.h
ControlMode.h ImageTest.h OutHouseRequest.h RouteStatus.h task_pose.h wall_tts_audio.h
cradle.h ImageTest.h OutHouseResponse.h start_back_point.h task_status.h wall_tts_audioRequest.h
cut.h ImageTest.h OutHouseResponse.h take_photo.h transfer.h wall_tts_audioResponse.h
dataCenterService.h ImageTest.h OutHouseResponse.h take_photo.h transfer.h wall_tts_audioResponse.h
dataCenterServiceRequest.h ImageTest.h OutHouseResponse.h take_photo.h transfer.h wall_tts_audioResponse.h
dataCenterServiceResponse.h ImageTest.h OutHouseResponse.h take_photo.h transfer.h wall_tts_audioResponse.h
DBTableUpdate.h ImageTest.h OutHouseResponse.h take_photo.h transfer.h wall_tts_audioResponse.h
demoResult.h ImageTest.h OutHouseResponse.h take_photo.h transfer.h wall_tts_audioResponse.h
detect_Result.h ImageTest.h OutHouseResponse.h take_photo.h transfer.h wall_tts_audioResponse.h
detect_ResultWeb.h ImageTest.h OutHouseResponse.h take_photo.h transfer.h wall_tts_audioResponse.h
```

7、配置依赖(例如依赖yidmsg包)

a、在~/./ros/目录下创建rosdep.yaml，该文件内容如下：

格式如下（operating_system:前后各有一个空格）：

```
rosdep_name:
operating_system: package1 package2
```

```
chenwei@chenwei-N85-87HP6:~$ cat .ros/rosdep.yaml
yidmsg:
ubuntu: ros-melodic-yidmsg
chenwei@chenwei-N85-87HP6:~$
```

b、将该文件连接到rosdep索引中，即在/etc/ros/rosdep/sources.list.d/目录下添加一个名为50-my-default.list文件，文件内容如下，其中，file指向的是以上编写的rosdep.yaml路径：

```
chenwei@chenwei-N85-87HP6:~$ cat /etc/ros/rosdep/sources.list.d/50-my-default.list
yaml file:///home/chenwei/.ros/rosdep.yaml
chenwei@chenwei-N85-87HP6:~$
```

c、执行rosdep update更新索引，将自定义消息加载到ros软件列表：

```
chenwei@chenwei-N85-87HP6:/etc/ros/rosdep/sources.list.d$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Hit file:///home/chenwei/.ros/rosdep.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Add distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Skip end-of-life distro "groovy"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Add distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/chenwei/.ros/rosdep/sources.cache
chenwei@chenwei-N85-87HP6:/etc/ros/rosdep/sources.list.d$
```

8、使用第三方库文件

要打的包中有使用第三方动态库文件的，可能会提示无版本信息，因为其不是使用dpkg安装，所以需要修改rules文件。

到需要打包的目录中，找到debian/rules，打开之后找到override_dh_shlibdeps，在最后加上如下选项：

--dpkg-shlibdeps-params=--ignore-missing-info

```
49 override_dh_shlibdeps:
50 stat# In case we're installing to a non-standard location, look for a setup.sh.h
51 roll# In the install tree that was dropped by catkin, and source it. It will e.h
52 roll# set things like CMAKE_PREFIX_PATH, PKG_CONFIG_PATH, and PYTHONPATH.
53 roll# if [ -f "/opt/ros/melodic/setup.sh" ]; then . "/opt/ros/melodic/setup.sh"; fi && \
54 roll# dh_shlibdeps -l$(CURDIR)/debian/ros-melodic-yd-cloudplatform//opt/ros/melodic/lib/ --dpkg-shlibdeps-params=--ignore-missing-info
```

~~~~~制作安装第三方动态库及配置环境变量的可执行程序~~~~~

1、将需要的第三方库文件打包成.tar.gz文件

```
tar -czvf setup_20210205_lib.tar.gz thirdparty_lib/
```

2、将sh执行脚本与第一步中的压缩文件制作成.run可执行程序

```
cat ros_install_lib.sh setup_20210205_lib.tar.gz > ros_install_lib_20210205.run
```

执行脚本如下，红框中为压缩文件名称：

```
chenwei@chenwei-M85-87HP6:~/notebook/ros_install$ cat ros_install_lib.sh
#!/bin/bash

# Only root can install
if [ $(id | grep root) ] > /dev/null; then
    echo "Installed"
else
    echo "You must be root to install these SDK package."
    exit 1
fi

CDIR=$(pwd)
INSTALLDIR=$CDIR/ros_needed_lib
dir_root=$(echo $CDIR | cut -d "/" -f2)
echo $dir_root
dir_users=$(echo $CDIR | cut -d "/" -f3)
echo $dir_users
if [ -d $INSTALLDIR ]; then
    rm -rf $INSTALLDIR
fi
mkdir $INSTALLDIR
export INSTALLDIR

# Find tar.gz line
ARCHIVE=$(awk '(/^_ARCHIVE_FOLLOWS_/) { print NR + 1; exit 0; }' $0)

# Use file
tail -n +$ARCHIVE $0 > $INSTALLDIR/setup_20210205_lib.tar.gz
cd $INSTALLDIR
tar -zxvf setup_20210205_lib.tar.gz

dir_all="/usr/local/lib/thirdparty_lib"
dir_hk="/usr/local/lib/thirdparty_lib/HCTNetSDKCom"
dir_ffmpeg="/usr/local/lib/thirdparty_lib/ffmpeg_lib"
dir_tb="/usr/local/lib/thirdparty_lib/tianbot"
cp thirdparty_lib /usr/local/lib -rf
echo "export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$dir_all:$dir_hk:$dir_ffmpeg:$dir_tb" >> /home/$dir_user/.bashrc
echo " " >> /home/$dir_user/.bashrc
echo " " >> /home/$dir_user/.bashrc

rm $INSTALLDIR/setup_20210205_lib.tar.gz
rm -rf $INSTALLDIR/thirdparty_lib
rm -rf $INSTALLDIR

echo " "
echo "***** Installation successful ! *****"
echo " "
exit 0
__ARCHIVE_FOLLOWS__
chenwei@chenwei-M85-87HP6:~/notebook/ros_install$
```

~~~~~安装及更新deb包~~~~~

1、在新系统安装deb

```
sudo dpkg -i *.deb
```

```
chenwei@chenwei-ThinkPad-E420:~$ sudo dpkg -i ros-melodic-imu-383-driver_0.0.0-0bionic_amd64.deb
[sudo] chenwei 的密码:
正在选中未选择的软件包 ros-melodic-imu-383-driver。
(正在读取数据库 ... 系统当前共安装有 335806 个文件和目录。)
正准备解包 ros-melodic-imu-383-driver_0.0.0-0bionic_amd64.deb ...
正在解包 ros-melodic-imu-383-driver (0.0.0-0bionic) ...
正在设置 ros-melodic-imu-383-driver (0.0.0-0bionic) ...
```

可以在opt/ros相关目录下看到安装文件

```
chenwei@chenwei-ThinkPad-E420:~$ ls /opt/ros/melodic/lib/imu_383_driver/
imu_383_driver
chenwei@chenwei-ThinkPad-E420:~$ ls /opt/ros/melodic/share/imu_383_driver/
cmake imu_383_driver.launch package.xml
chenwei@chenwei-ThinkPad-E420:~$
```

2、卸载旧版本，安装新版本

```
sudo dpkg --purge ***-dbgshym
sudo dpkg --purge ***
sudo dpkg -i ***.deb(新版本deb文件)
```

```
chenwei@chenwei-ThinkPad-E420:~$ sudo dpkg --purge ros-melodic-imu-383-driver-dbgshym
(正在读取数据库 ... 系统当前共安装有 338406 个文件和目录。)
正在卸载 ros-melodic-imu-383-driver-dbgshym (0.0.0-0bionic) ...
chenwei@chenwei-ThinkPad-E420:~$ sudo dpkg --purge ros-melodic-imu-383-driver
(正在读取数据库 ... 系统当前共安装有 338403 个文件和目录。)
正在卸载 ros-melodic-imu-383-driver (0.0.0-0bionic) ...
chenwei@chenwei-ThinkPad-E420:~$ ls /opt/ros/melodic/lib/imu_383_driver/
ls: 无法访问 '/opt/ros/melodic/lib/imu_383_driver/': 没有那个文件或目录
chenwei@chenwei-ThinkPad-E420:~$ ls /opt/ros/melodic/share/imu_383_driver/
ls: 无法访问 '/opt/ros/melodic/share/imu_383_driver/': 没有那个文件或目录
chenwei@chenwei-ThinkPad-E420:~$ sudo dpkg -i ros-melodic-imu-383-driver_2021.2.2-0bionic_amd64.deb
正在选中未选择的软件包 ros-melodic-imu-383-driver。
(正在读取数据库 ... 系统当前共安装有 338393 个文件和目录。)
正准备解包 ros-melodic-imu-383-driver_2021.2.2-0bionic_amd64.deb ...
正在解包 ros-melodic-imu-383-driver (2021.2.2-0bionic) ...
正在设置 ros-melodic-imu-383-driver (2021.2.2-0bionic) ...
chenwei@chenwei-ThinkPad-E420:~$ ls /opt/ros/melodic/lib/imu_383_driver/
imu_383_driver
chenwei@chenwei-ThinkPad-E420:~$ ls /opt/ros/melodic/share/imu_383_driver/
cnafe imu_383_driver.launch package.xml
chenwei@chenwei-ThinkPad-E420:~$
```

~~~~~安装第三方动态库~~~~~

- 1、将.run文件拷贝到 /home/用户名 目录下，给.run文件赋予可执行权限

```
sudo chmod 777 ros_install_lib_20210205.run
```

- 2、使用root用户，执行.run可执行程序，安装成功后退出root权限

```
sudo su
./ros_install_lib_20210205.run
#看到***** Installation successful ! *****说明安装完成
exit
```

```
thirdparty_lib/libcrypto.so
thirdparty_lib/libcnet50k_log_Switch.xml
thirdparty_lib/libffmepg_lib/libavfilter.so.7
thirdparty_lib/libffmepg_lib/libavutil.so.56
thirdparty_lib/libffmepg_lib/libbwsresample.so.3
thirdparty_lib/libffmepg_lib/libbavcodec.so.58
thirdparty_lib/libffmepg_lib/libbavformat.so.58
thirdparty_lib/libffmepg_lib/libbwscale.so.5
thirdparty_lib/libffmepg_lib/libbavdevice.so.58
thirdparty_lib/libffmepg_lib/libx264.so.157
thirdparty_lib/libSuperRender.so
***** Installation successful ! *****
root@chenwei-NB5-B7HP6:/home/chenwei/notebook/ros_install#
```