Can I run a bash script using roslaunch?

10

roslaunch bash

I read http://answers.ros.org/question/12327... and http://answers.ros.org/question/33772... questions and I am confused. I see people mention the possibility of running bash scripts via .launch files but I can't find any documentation describing how to do it. Is it even possible?

asked Jan 4 '13
autonomy 415 ●17 ●21 ●25

updated Nov 20 '14
130s 9900 ●184 ●290 ●309
http://www.linkedin.co...

(I'm trying to delay the launch of a gpsd_client node because during startup it scans and usurps other serial ports.)

add a comment

4 Answers

Sort by » oldest newest most voted

18

Yes, This is possible. Place your shell script inside a ROS package. Use chmod to make the script executable. You can then run it from a roslaunch file similar to any node (any executable for that matter):

answered Jan 4 '13
piyushk 2851 ●31 ●49 ●76
http://cs.utexas.edu/~piyushk

```
<node name="foo" pkg="YOUR_ROS_PACKAGE" type="YOUR_SCRIPT_NAME" />
```

link

Comments

Can anyone add to this how this works in the context of catkin? Where should the script go within the node directory, do there need to be CMake commands to install it to a predictable place?
lucasw (Dec 6 '13)

lucasw, have you figured out the answer to your question? if so, could you please post it?
Wesley (Jul 18 '15)

1 I think best practice is to put it in scripts/ alongside python files, and install it the same as a python file would be installed.
lucasw (Jul 21 '15)

I now it's an old thread. lucasw, if you can read this, can you explain me in details what have you done? Thank you!
Marcofon (Nov 9 '16)

5 I think you just need `install(PROGRAMS scripts/foo.sh DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION})`
lucasw (Nov 10 '16)

Thank you for your reply! I tried something similar but it doesn't work. Did you tried and reached your goal? My script it's not a .sh, it has no extension. Thank you again!

EDIT: I tried copying the content of the script in a new script with .sh extension, i'm getting:

```
[test.sh-3] process has died [pid 3316, exit code 1
```

Marcofon (**Nov 10 '16**)

You'll probably need to ask a new question and put your test.sh script and the launch file launching it into it (and link to it from here), though it looks like your script did in fact run when launched, though maybe it didn't do what you want it to?

lucasw (**Nov 10 '16**)

It didn't run at all, it has to do some elaborations to a .txt file and then print results on the terminal. Maybe i have found a workaround, i used the command:

```
system("terminal command for launching the script");
```

It is a c++ command that gives to the terminal the command that you want.

Marcofon (**Nov 11 '16**)

add a comment

15

Better late than never! I came across a similar issue and have implemented a convenient solution. Hopefully many more people can use it, as I find it to be very helpful!

answered Apr 24 '14
koenlek 392 ●10 ●16 ●22

How to use it?

Save the script shown below to a file named "timed_roslaunch.sh". Put it in a scripts/ subfolder of your catkin package. Make it executable using chmod +x. And use it either through command line or a launch file. The scripts help (./timed_roslaunch.sh -h) is shown below:

```
This script can delay the launch of a roslaunch file
Place it in the 'scripts' folder of your catkin package
and make sure that the file is executable (chmod +x timed_roslaunch.sh)

Run it from command line:

Use: ./timed_roslaunch.sh [number of seconds to delay] [rospkg] [roslaunch file] [arguments (optio
nal)]
Or: rosrun [yourpackage] time_roslaunch.sh [number of seconds to delay] [rospkg] [roslaunch file]
[arguments (optional)]
Example: ./timed_roslaunch.sh 2 turtlebot_navigation amcl_demo.launch initial_pose_x:=17.0 initial
_pose_y:=17.0

Or run it from another roslaunch file:

<launch>
<arg name="initial_pose_y" default="17.0" />
  <node pkg="semantic_turtle_test" type="timed_roslaunch.sh"
    args="2 turtlebot_navigation amcl_demo.launch initial_pose_x:=17.0 initial_pose_y:=$(arg initi
al_pose_y)"
    name="timed_roslaunch" output="screen">
```

```
  </node>
</launch>
```

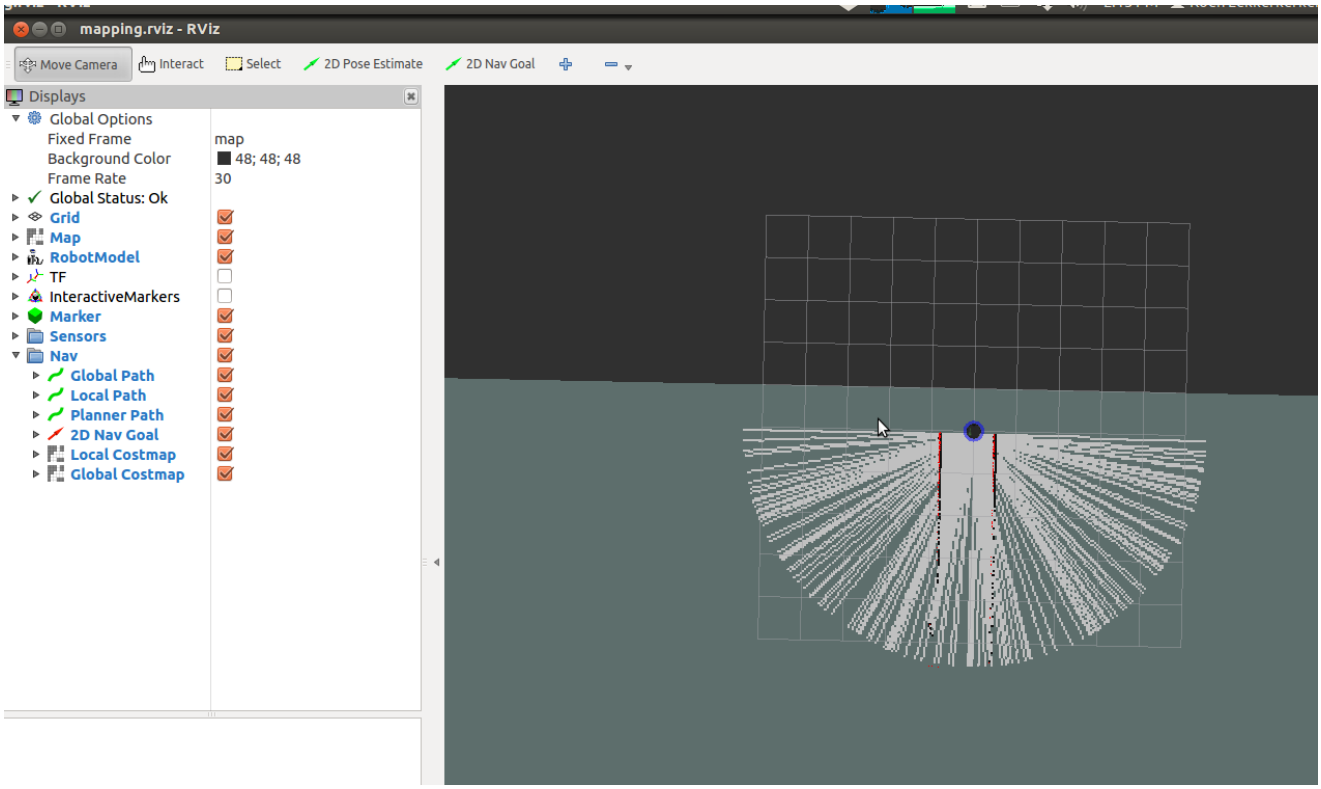
The script or use this link.

```
#!/bin/bash
#
# Script to delay the launch of a roslaunch file
#
# Koen Lekkerkerker
# Thu 24 Apr 2014
#
# Use: ./timed_roslaunch.sh [number of seconds to delay] [rospkg] [roslaunch file]
#

function showHelp(){
    echo
    echo "This script can delay the launch of a roslaunch file"
    echo "Place it in the 'scripts' folder of your catkin package"
    echo "and make sure that the file is executable (chmod +x timed_roslaunch.sh)"
    echo
    echo "Run it from command line:"
    echo
    echo "Use: ./timed_roslaunch.sh [number of seconds to delay] [rospkg] [roslaunch file] [argume
nts (optional)]"
    echo "Or: rosrun [yourpackage] time_roslaunch.sh [number of seconds to delay] [rospkg] [roslau
nch file] [arguments (optional)]"
    echo "Example: ./timed_roslaunch.sh 2 turtlebot_navigation amcl_demo.launch initial_pose_x:=1
7.0 initial_pose_y:=17.0"
    echo
    echo "Or run it from another roslaunch file:"
    echo
    echo '<launch>'
    echo '  <arg name="initial_pose_y" default="17.0" />'
    echo '  <node pkg="semantic_turtle_test" type="timed_roslaunch.sh"'
    echo '    args="2 turtlebot_navigation amcl_demo.launch initial_pose_x:=17.0 initial_pose_y:=
$(arg initial_pose_y)"'
    echo '    name="timed_roslaunch" output="screen">'
    echo '  </node>'
    echo '</launch>'
}

if [ "$1" = "-h" ]; then
    showHelp
else
    echo "start wait for $1 seconds"
    sleep $1
    echo "end wait for $1 seconds"
    shift
        echo "now running 'roslaunch $@'"
    roslaunch $@
fi
```

Now why did I need it?

My problem was that when I have launch script to launch a fairly large project. Gazebo simulates a world with a simulated turtlebot and gmapping is used for mapping it. It seems that gazebo returns laser measurements before the whole world is spawn. The robot is in a corridor, but as the walls are spawned later than the laser, the first laser measurements give a full free circle. As gmapping already processed these measurements, gmapping incorrectly mapped this circle as free space.

See screenshot:



link

Comments

That script looks useful. However, wouldn't subsequent laser readings be consumed by gmapping, causing it to update the map with the obstacles/walls?

ceverett (**Apr 24 '14**)

yes and no: the walls are indeed mapped after the world has spawned. But, the fact that space behind these walls is considered free, will affect gmappings performance slightly when it compares measuremnts & expectations (theoretically). In practice, gmapping is robust enough to overcome this.

koenlek (**Apr 24 '14**)

The core of the problem in my opinion lies with Gazebo, that starts up the robots sensors before the world has spawn. Which does not make sense...

koenlek (**Apr 24 '14**)

If I do not misunderstand, it seems that you could try setting the Gazebo simulator to be paused when it starts in the launch file, and you could start the simulation with the start button in the GUI of the simulator after you are sure that everything is correctly spawned.

Mike Gao (**May 12 '14**)

1 couldn't you also just separate into two launch files: 1st one launches world, 2nd one launches robot+sensor

nbanyk (**Sep 18 '14**)

add a comment

0

I also stumbled across this problem and implemented a generic solution. You might want to have a look at the aliencontrol ROS package.

answered Oct 25 '18
aschaefer 31 ●3 ●4

This package contains a node that starts any external application. Almost more importantly, the node also takes care of a clean shutdown, no matter whether the external application or the containing ROS node is terminated.

It is run like so (if you wanted to start `your_script.sh`):

```
rosrun aliencontrol aliencontrol /path/to/your_script.sh
```

If you want to embed it in your launch script, add:

```
<node pkg="aliencontrol" type="aliencontrol" name="aliencontrol">
    <param name="cmd" value="/path/to/your_script.sh"/>
</node>
```

link

add a comment

0

See this source code http://www.ros.org/wiki/webots_run for an example of running an executable from a ROS package.

answered Jan 7 '13
dbworth 1083 ●28 ●43 ●51

link

Comments

+1; For those who may have missed the nuance here -- The advantage of doing it this way is that you are actually creating a ROS Node for your executable, and in this way you can tell the process is executing, and possibly control it.
rlinsalata (**Sep 19 '15**)

I did not understand what is that. Webots replaces roslaunch ? The executable is a webots ? Where does this link talk about executable ?
Piero-31 (**Jul 19 '16**)

add a comment

Your Answer

Please start posting anonymously - your entry will be published after you log in or create a new account.

Add Answer

Question Tools

Follow

5 followers

subscribe to rss feed

Stats

Asked:	**Jan 4 '13**
Seen:	**31,971 times**
Last updated:	**Oct 24 '18**

Related questions

- Why does my shell script (web server) node suddenly not work?
- Shell bash used in a roslaunch[SOLVED]
- Melodic upgrade errors on Ubuntu 18.04.1 LTS
- How do I start a node with an anonymous name with roslaunch?
- what are the differences between "name" and "type" in roslaunch?
- How to launch a roslaunch in Qt with a terminal (GUI) ?
- How to read rosparam in a launch file
- ROS Stream prints spurious characters to log
- roslaunch dies with exit code -11 after it stops receiving topic from bag file
- Can't run rtabmap from roslaunch