Lab 4 - RIDE User Manual

Team Orange

Old Dominion University

CS411W

Professor J. Brunelle

April 18, 2023

Version 1

Table of Contents

1. Introduction (Dan Koontz)

R-IDE (ROS - Integrated Development Environment) is a tool that simplifies and improves the development process for ROS applications for all developers. R-IDE exists as an extension available on Visual Studio Code (VSCode) and is available to all ROS developers. These developers may be industry professionals, university students, or robotics hobbyists. The purpose of R-IDE is to create a single workspace with a single window that contains all of the tools commonly needed by ROS developers. This manual will help users create their first projects with R-IDE.

2. Environment Setup (Joshua Peterson)

2.1 Windows

When setting up a ROS environment on windows you may choose between WSL or a VM virtual box with linux installed. You do not need to install both. To install WSL, see section 2.2.1. To install a VM Virtual box, see section 2.1.2.

2.1.1 WSL

Step 1: Install WSL2 on your Windows machine following the official Microsoft WSL installation guide.

Step 2: Install Ubuntu 20.04 for WSL from the Microsoft Store.

Step 3: Launch the WSL Ubuntu terminal and follow the steps in section 2.2 to install ROS on your WSL

2.1.2 VM Virtual Box

Step 1: Download and install VirtualBox for your host OS.

Step 2: Download the Ubuntu 20.04 ISO file.

Step 3: Create a new virtual machine in VirtualBox, selecting "Linux" and "Ubuntu (64-bit)" as the OS type.

Step 4: Allocate at least 2 CPU cores, 4 GB of RAM, and 20 GB of storage.

Step 5: Mount the downloaded Ubuntu 20.04 ISO file as a virtual optical disk and start the virtual machine.

Step 5: Follow the on-screen prompts to install Ubuntu 20.04.

Step 6: Launch the Ubuntu VM and follow the steps in section 2.2 to install ROS on your Ubuntu VM.

2.2 Linux (Ubuntu 20.04) - **Required**

2.2.1 ROS installation - **Required**

Step 1: Follow the official ROS installation guide for ROS Noetic on Ubuntu 20.04.

Step 2: Set up the ROS environment by adding the following lines to your ~/.bashrc file:

```
source /opt/ros/noetic/setup.bash
```

Step 3: Restart your terminal or run the following command:

```
source ~/.bashrc
```

2.2.2 ROS Bridge - **Required**

Step 1: Install the necessary dependencies by running the following commands:

```
sudo apt-get update
sudo apt-get install ros-noetic-rosbridge-server
```

Step 2: ROS bridge is required for RIDE to communicate with ROS. When using RIDE run the following command to start ROS bridge(this will automatically run roscore):

```
roslaunch rosbridge_server rosbridge_websocket.launch
```

2.3 Mac

There are several methods to set up a ROS environment on a Mac, however this document will only describe how to set up the ROS environment using a VM virtual box.

2.3.1 VM Virtual Box

Step 1: Download and install VirtualBox for your host OS.

Step 2: Download the Ubuntu 20.04 ISO file.

Step 3: Create a new virtual machine in VirtualBox, selecting "Linux" and "Ubuntu (64-bit)" as the OS type.

Step 4: Allocate at least 2 CPU cores, 4 GB of RAM, and 20 GB of storage.

Step 5: Mount the downloaded Ubuntu 20.04 ISO file as a virtual optical disk and start the virtual machine.

Step 5: Follow the on-screen prompts to install Ubuntu 20.04.

Step 6: Launch the Ubuntu VM and follow the steps in section 2.2 to install ROS on your Ubuntu VM.

2.4 Visual Studio Code (VSCode) - **Required**

Step 1: Download and install Visual Studio Code for your respective operating system

2.4.1 Microsoft ROS Extension - **Required**

Step 1: Open VSCode

Step 2: Open the Extensions view in VSCode by clicking on the Extensions icon in the “Activity Bar” on the side of the window.

Step 3: Search for "ROS" in the search bar.

Step 4: Click on "Robot Operating System (ROS)" by Microsoft and then click "Install".

2.4.2 RIDE Extension - **Required**

Step 1: Open VSCode

Step 2: Open the Extensions view in VSCode by clicking on the Extensions icon in the “Activity Bar” on the side of the window.

Step 3: Search for "R-IDE" in the search bar.

Step 4: Click on "R-IDE" by “TeamOrangeODU” and then click "Install".

2.3 Catkin Workspace - **Required**

Step 1: Follow the official ROS tutorial to create a new Catkin workspace.

2.3.1 Opening Catkin Workspace in VSCode

Step 1: Open VSCode.

Step 2: Click “file” in the top left corner.

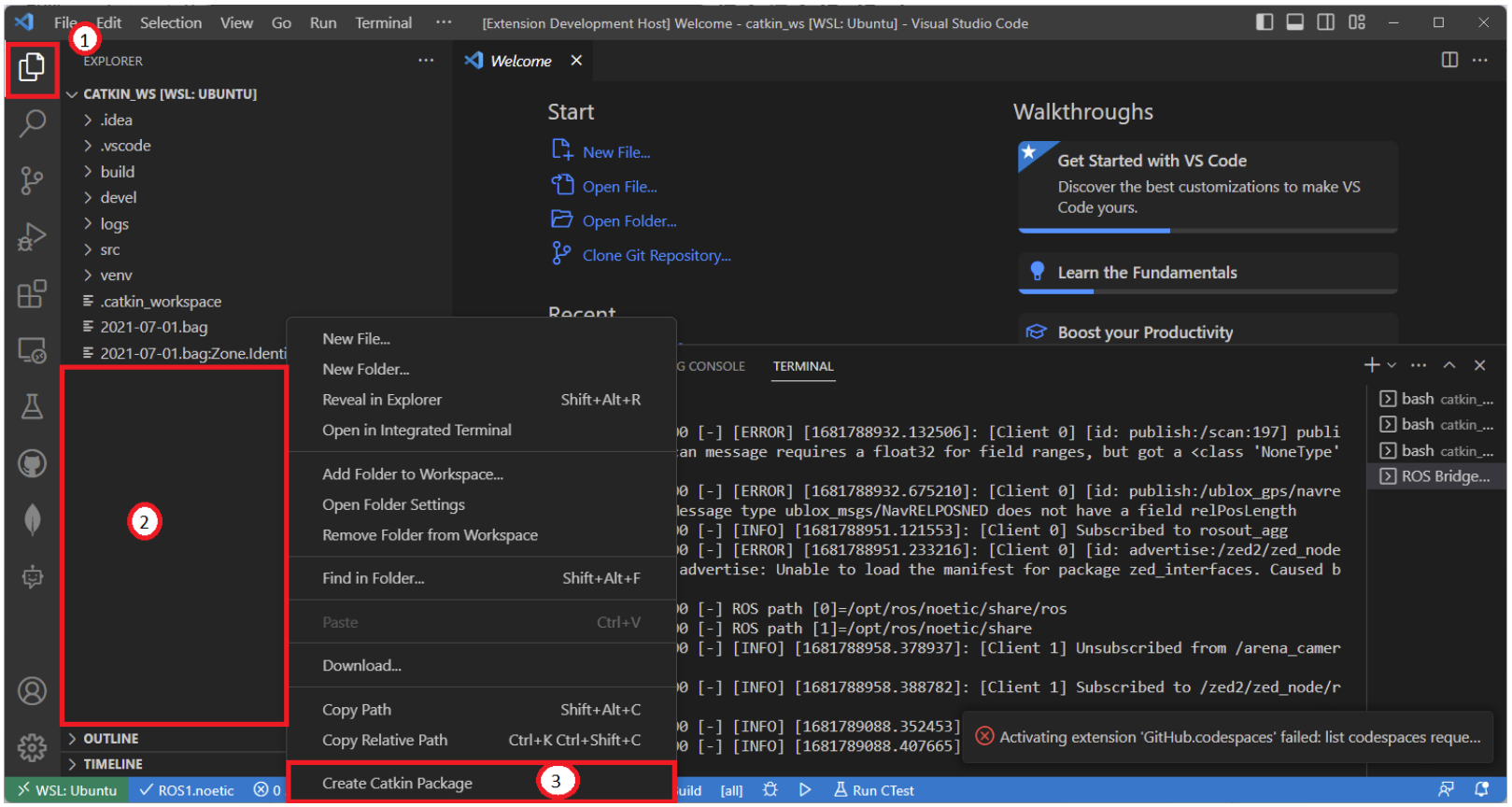
Step 3: Select “Open folder” from the drop down menu

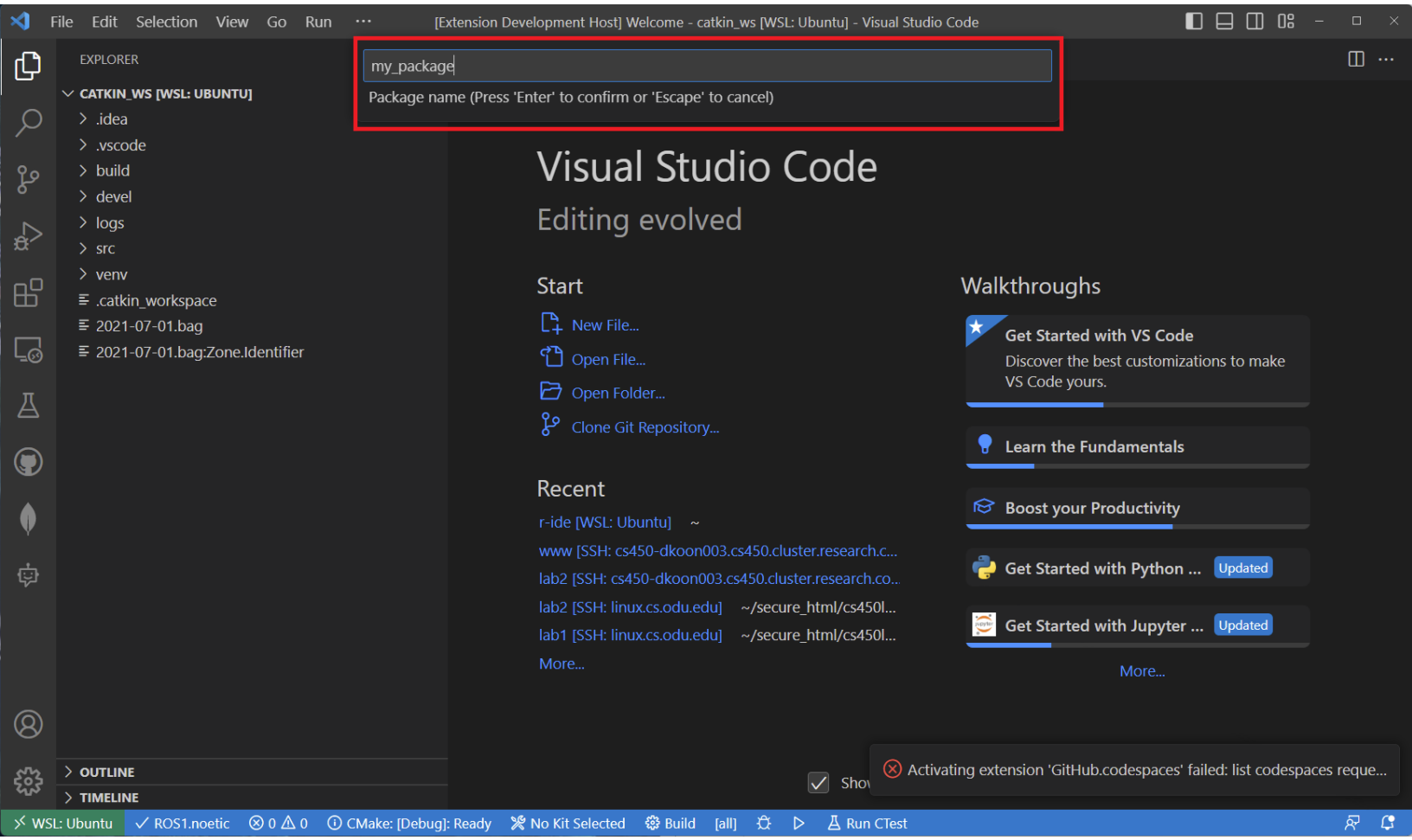
Step 4: Navigate and select the catkin workspace created in step 2.3

3. ROS Package Creation (Dan Koontz)

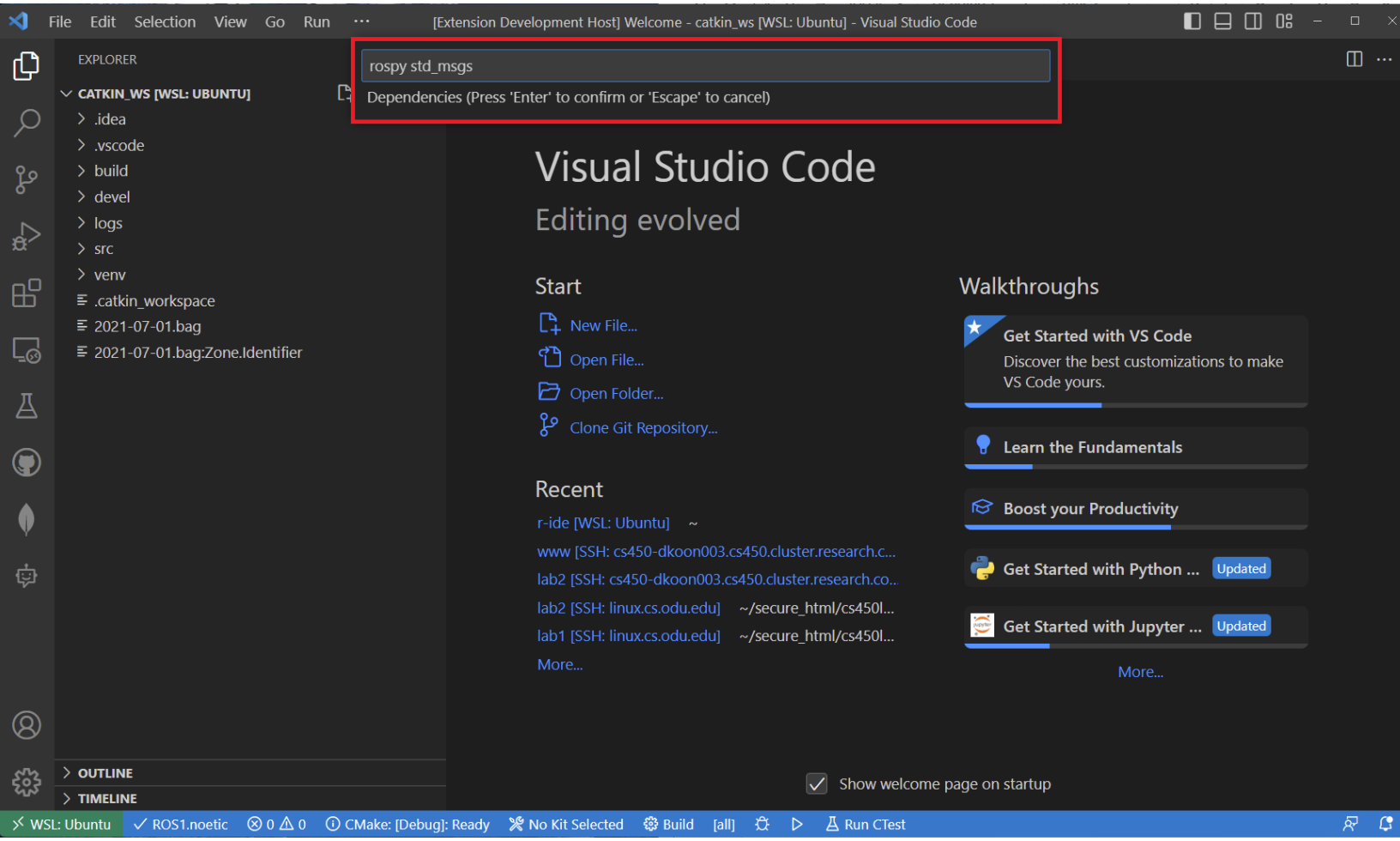
In order to build an application, users need to create a new catkin package.

Step 1. In order to create a new package open up a ROS workspace. To create the new package enter the explorer menu [1] and right click on a directory or empty space [2] and select “Create Catkin Package” [3].

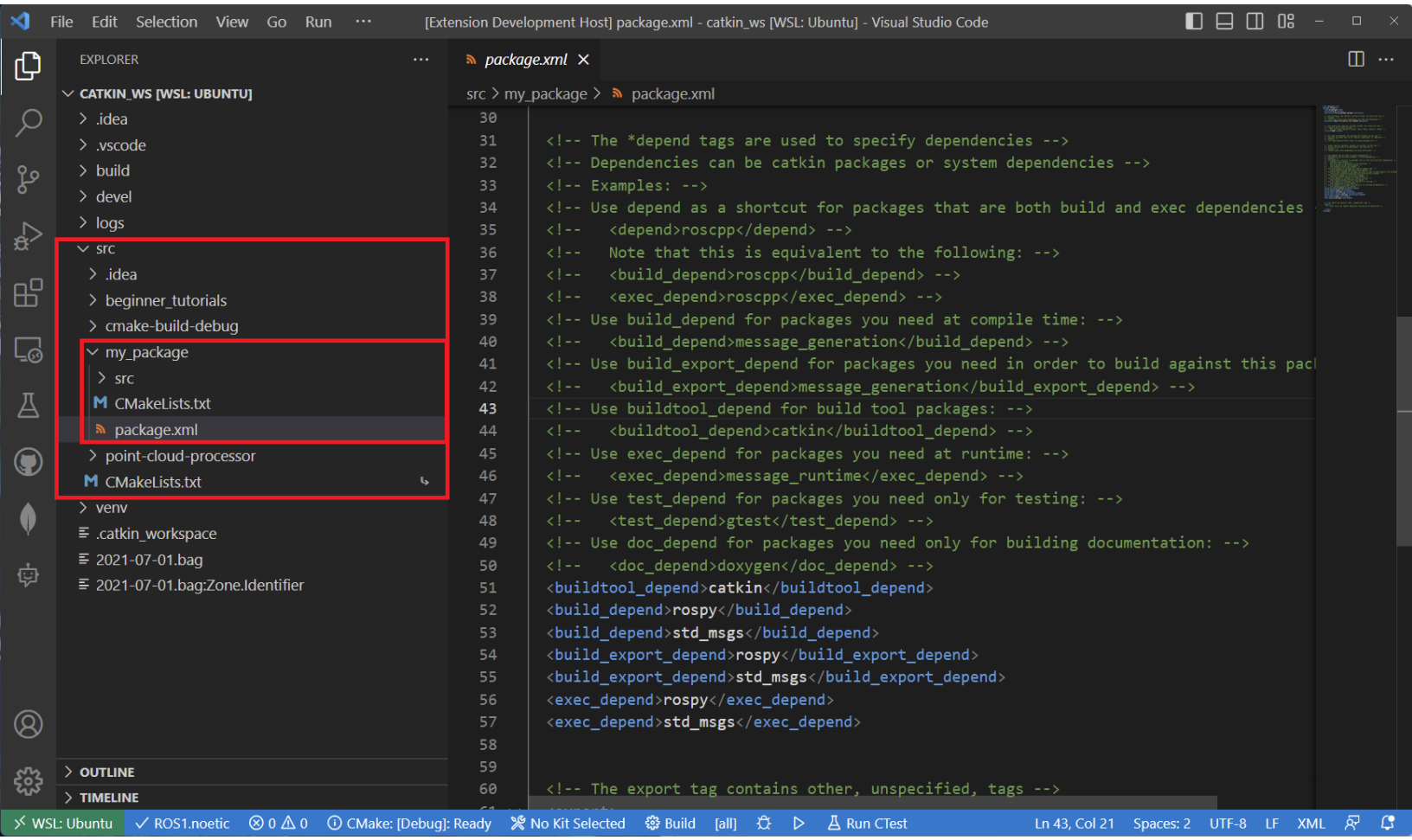




Step 2. In the area indicated by the Red Box: Enter a name for the package in the dialogue that appears.

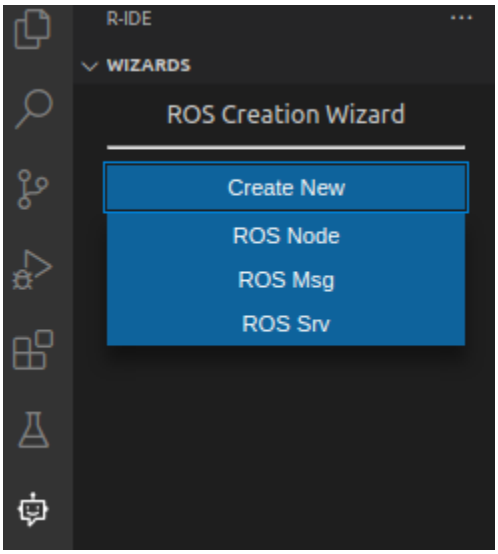


Step 3. In the area indicated by the Red Box: Enter any known dependencies space separated for the package or leave this space blank.

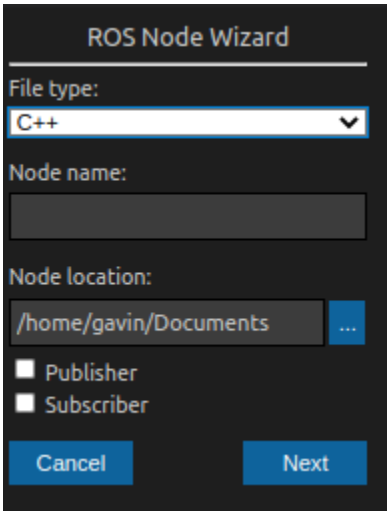


Step 4. The newly created package will now appear under the `src` directory containing the `CMakeLists.txt` and `package.xml` files along with an inner `src` directory. This can be seen in the red box indicated.

4. Creation Wizard (Gavin St. Clair)



Step 1: In the R-IDE Creation Wizard click create new and select either ROS Node, ROS Msg, or ROS Srv.



Step 2: From the R-IDE Creation Wizard select create new and select Ros Node. Click the drop down to select file type, either C++ or python. Type in desired node name and select location. Click the box to choose the node type: publisher, subscriber or both. When finished, click the next button.

Step 2.1: If a user selects ROS Msg or ROS Srv from the Creation Wizard then only options will be given to name the file and select location.

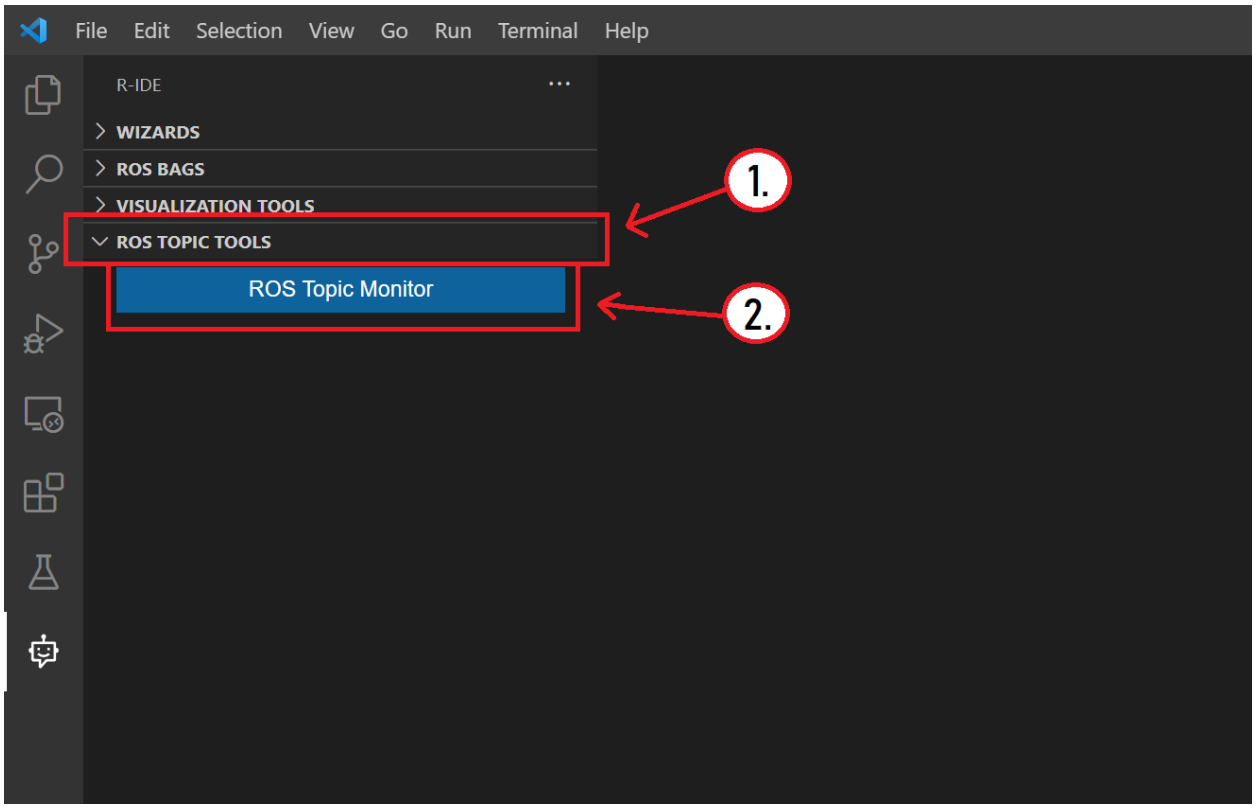
Step 3: There should be a new node file created with the given name and auto populated with the default file snippet depending on your specifications.

5. Topic Monitor (Joshua Peterson)

RIDE's topic monitor is used to subscribe to current topics and view messages in real-time as they are sent between nodes. RIDE's topic monitor also provides a message publisher that gives the developer the ability to send custom messages to subscribed topics.

5.1 Opening RIDE's Topic Monitor

This section describes how to navigate to and open RIDE's topic monitor



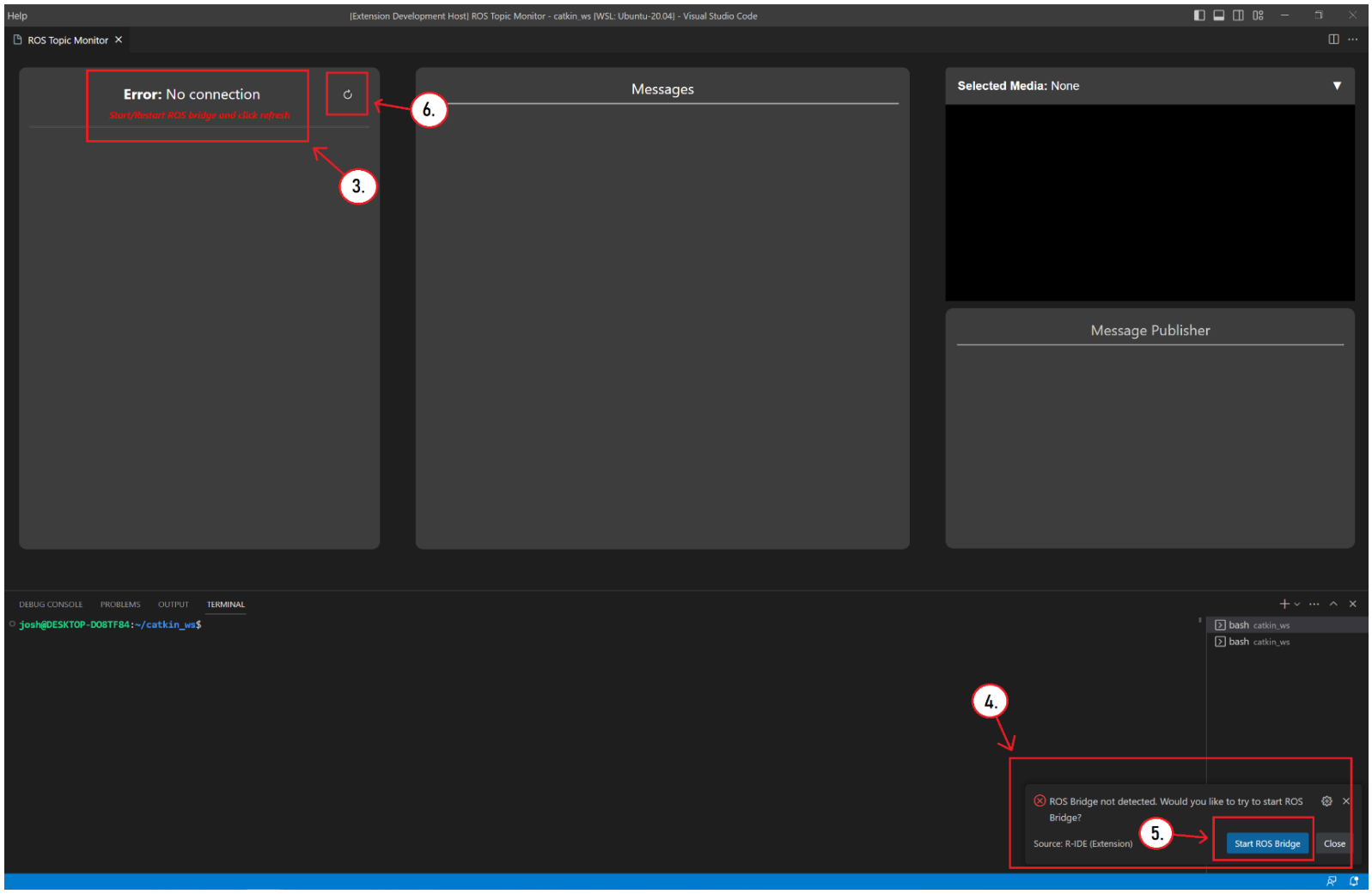
To open RIDE's topic monitor,

Step 1: Click the ROS topic tools button (1) to expand the ROS Topic tools view.

Step 2: After the view is expanded, click the ROS topic monitor button (2) to open the ROS topic monitor.

5.2 Establishing Connection. Describe the ways I use this.

This section describes how to start a connection with ROS bridge via the topic monitor. A ROS bridge connection is required in order for RIDE to connect to your ROS network.



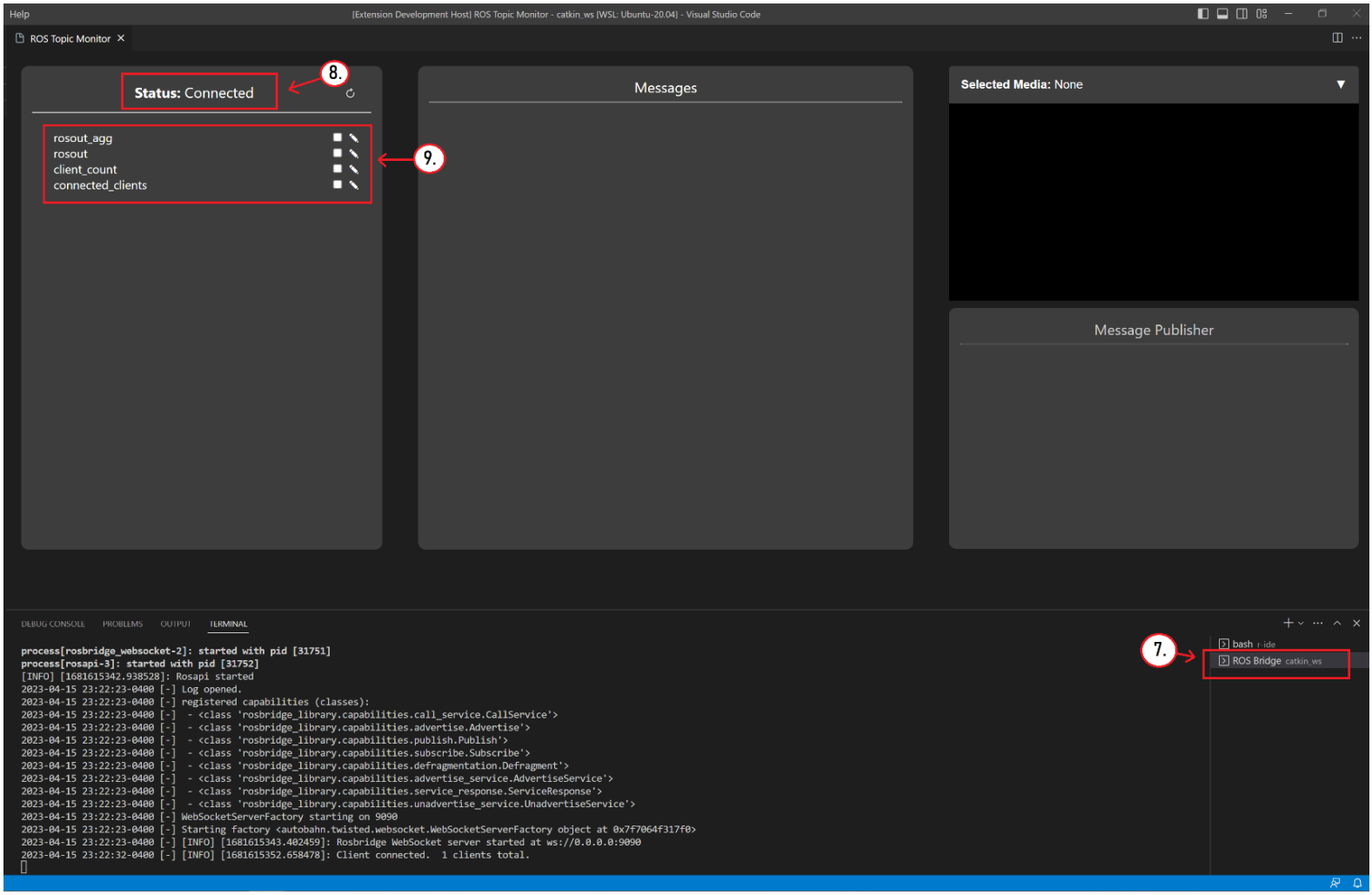
Step 1: If you have not started ROS Bridge yet, the topic monitor will display “Error: No connection” (3).

Step 2: A popup window will display in the bottom right corner (4) with the error message “ROS Bridge not detected, would you like to start ROS Bridge?”.

Step 3: Click start ROS bridge (5) to automatically start ROS Bridge.

Step 4: After ROS Bridge starts, click the refresh button (6) to refresh and display the topics.

If ROS Bridge is not installed, please see section 2.x to install ROS Bridge.

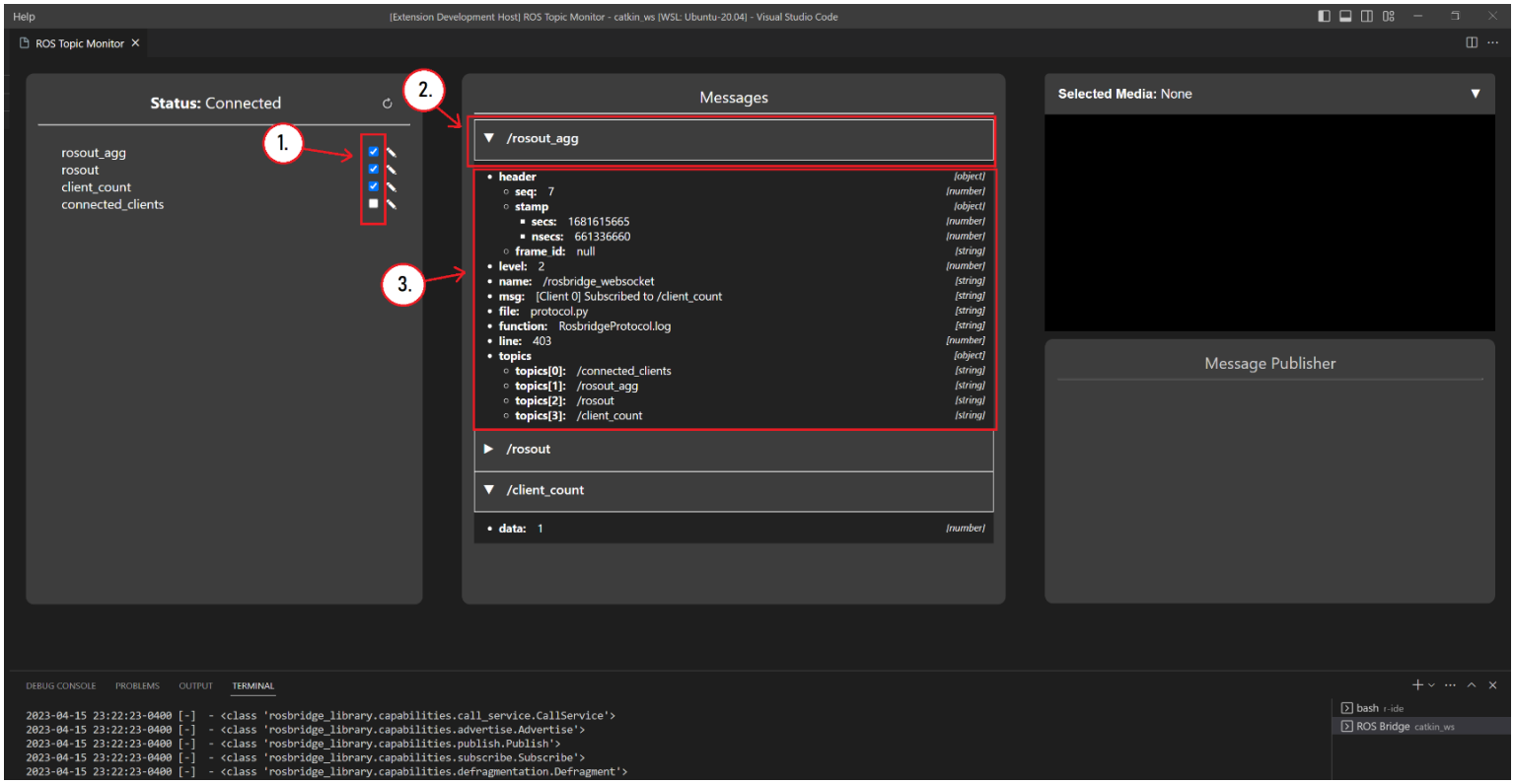


Step 5: After successfully starting ROS bridge (7) and refreshing, you will see “Status: Connected” (8) along with all of the current topics (9).

[This space is intentionally left blank.]

5.3 Topic Subscription

This section describes how to subscribe to current topics and view the messages being sent in real-time.

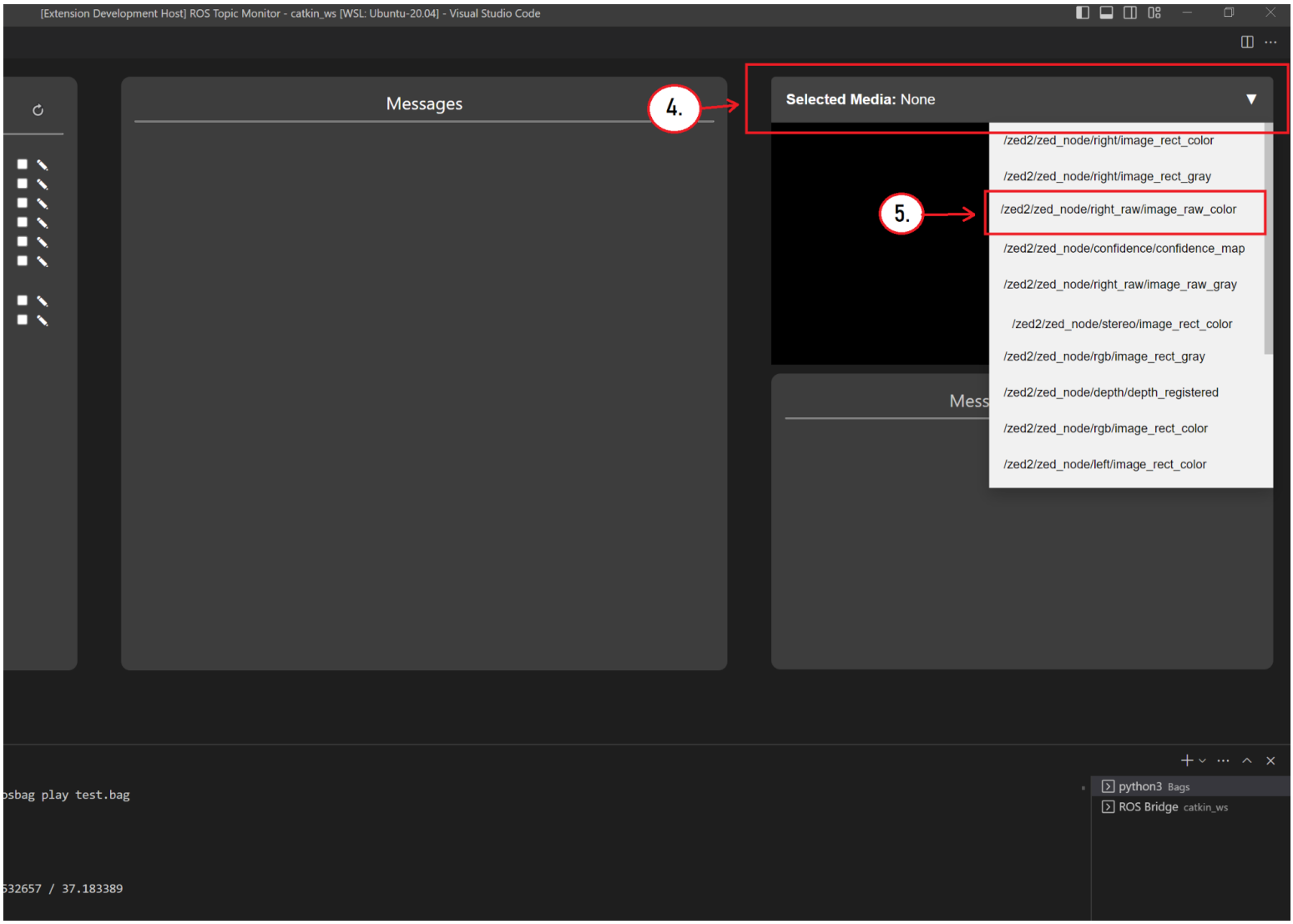


Step 1: To subscribe to a topic, click the topic checkbox (1) next to the topic you would like to subscribe to.

Step 2: You will see the subscribed topic (2) displayed in the messages portion of the screen.

Step 3: Click the subscribed topic (2) to expand the topic and view the topic data (3).

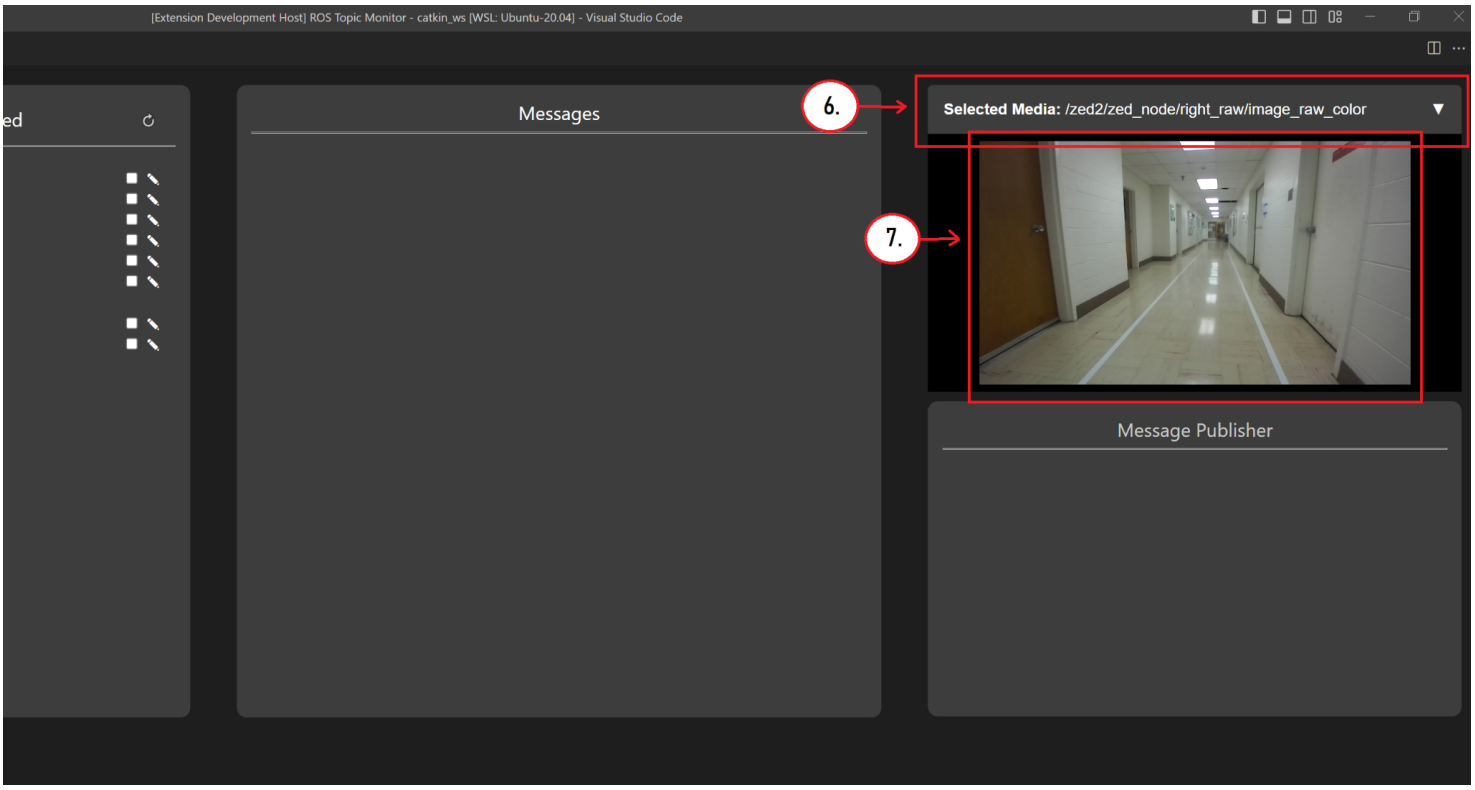
[This space is intentionally left blank.]



Step 4: To subscribe to media topics for camera/sensor imagery, hover your mouse over the selected media (4) drop down. You will see a dropdown menu displayed with all of the camera/sensor imagery topics available for selection.

Step 5: Select a media topic (5).

Note – This example is utilizing a ROS bag that contains several media topics.



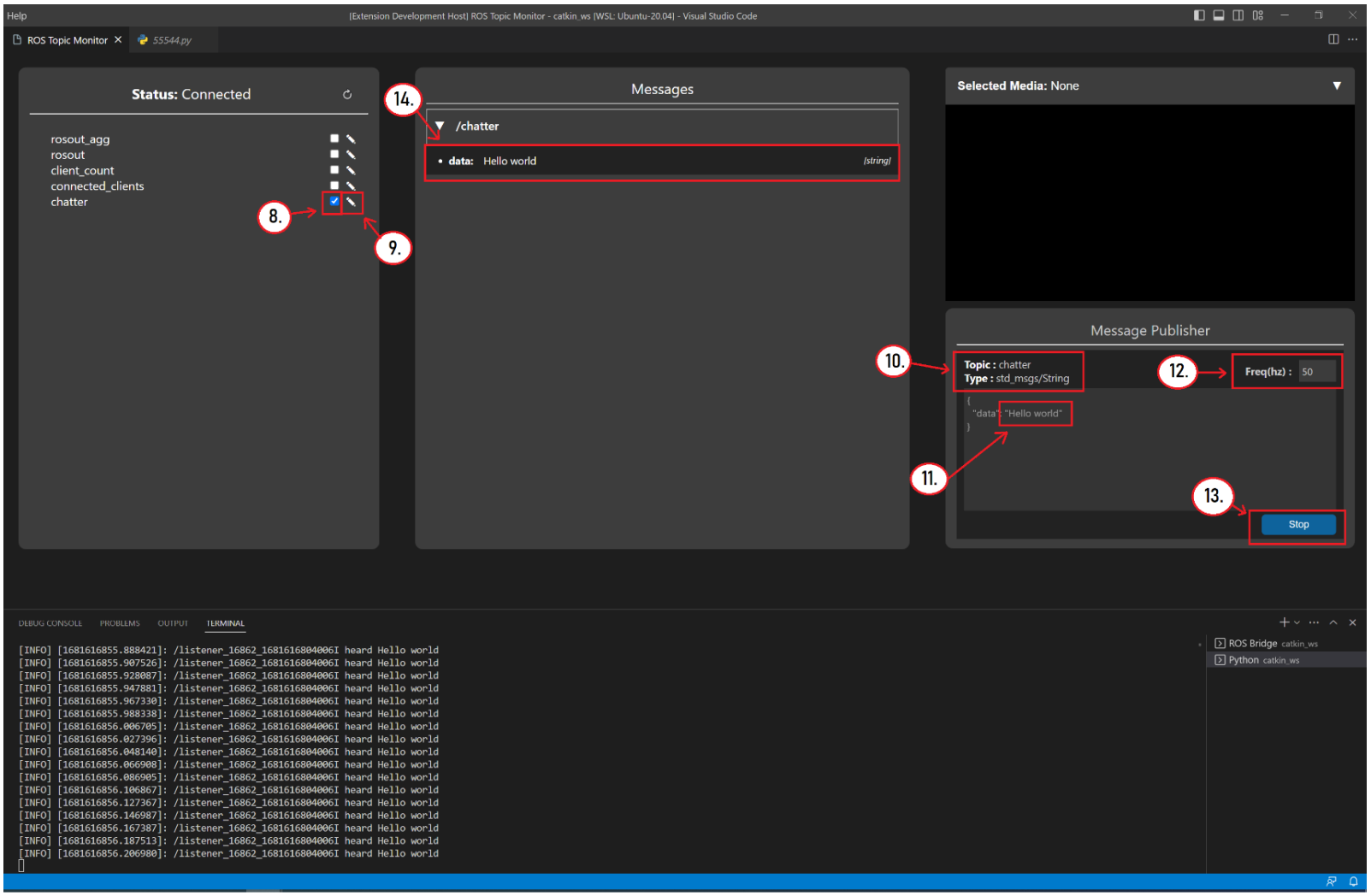
Step 6: After selecting the media topic, you will see your selected media topic (6) displayed in the dropdown menu.

Step 7: The data from your selected media topic will be displayed in the media player (7).

[This space is intentionally left blank.]

5.4 Message Publisher

This section describes how to utilize RIDE's message publisher to send custom messages to any topic that is currently subscribed.



Step 1: To publish messages to a topic, click a topic checkbox (8) to subscribe to a topic.

Step 2: Once subscribed click the pencil icon button (9) to choose the topic you would like to publish messages on.

Step 3: After selecting a topic to publish messages on, you will see the topic and message type (10) displayed in the message publisher, along with a JSON object of the message type.

Step 4: Edit the value section (11) of the JSON object to specify the data you would like to send. In this example we are sending the string “Hello world” to the topic “chatter”.

Step 5: Set the frequency (12) that you would like the messages to be published at and click publish/stop button (13) to start/stop publishing.

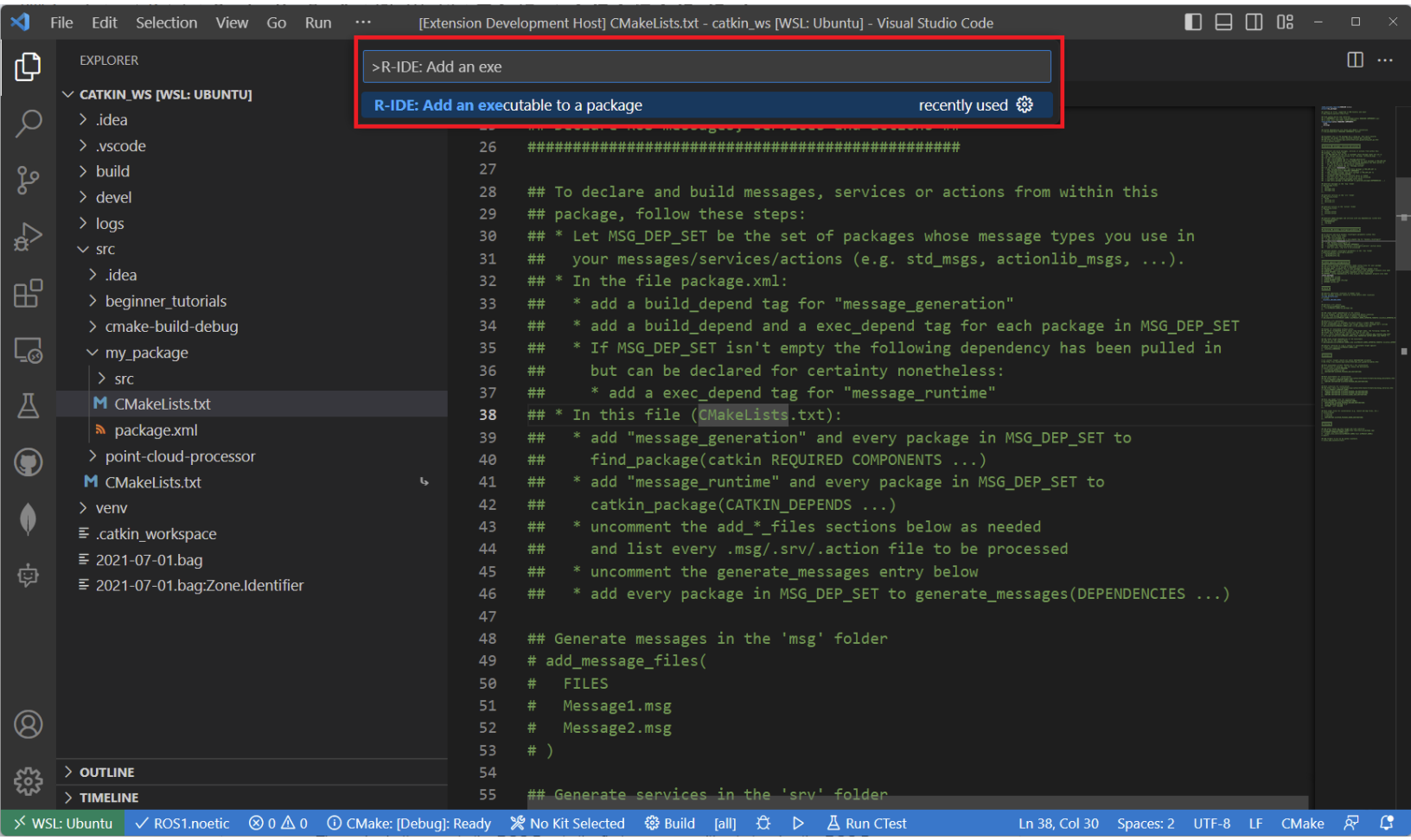
Step 6: You can see the data being published in the subscribed topic (14).

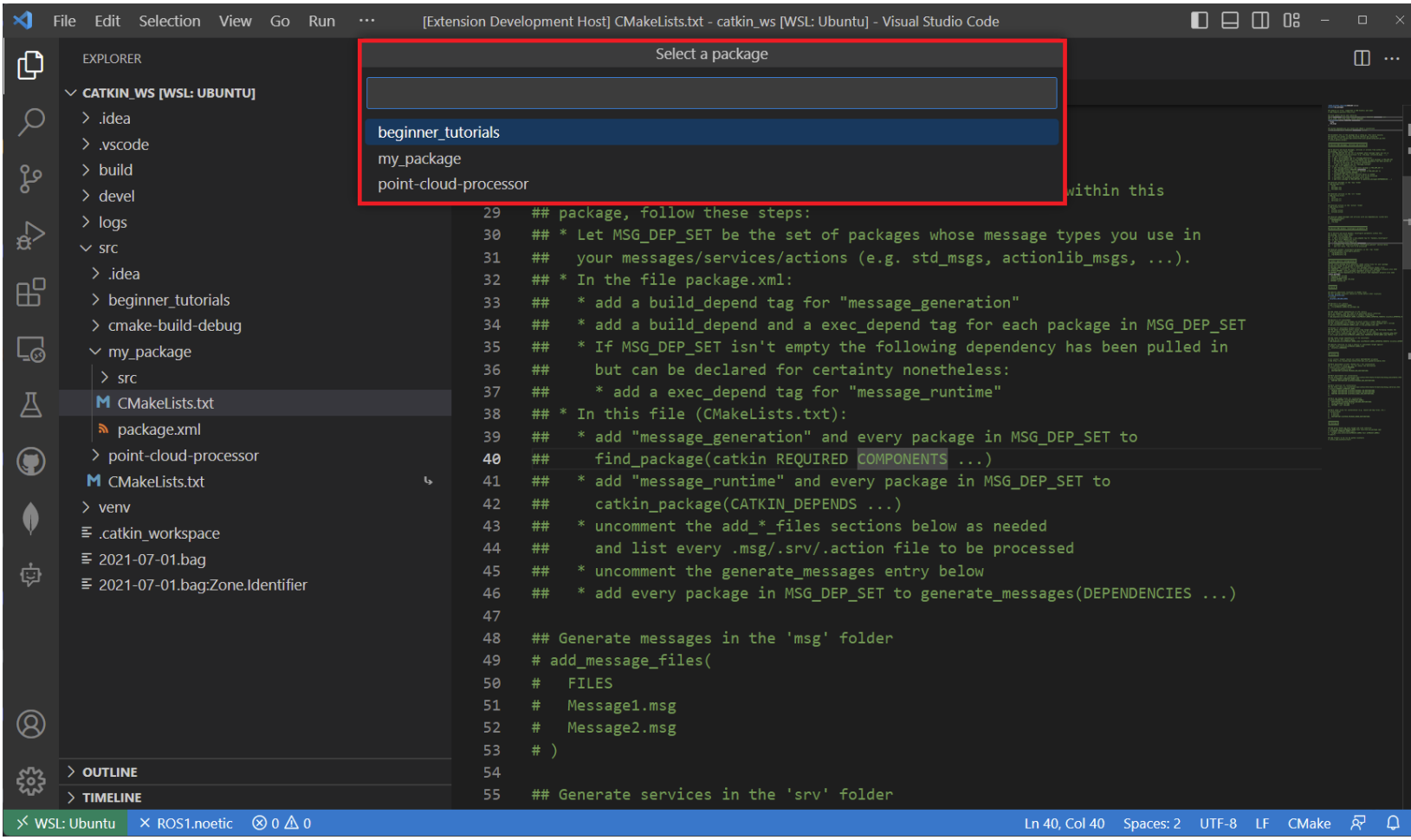
[This space is intentionally left blank.]

6. Create Executable (Dan Koontz)

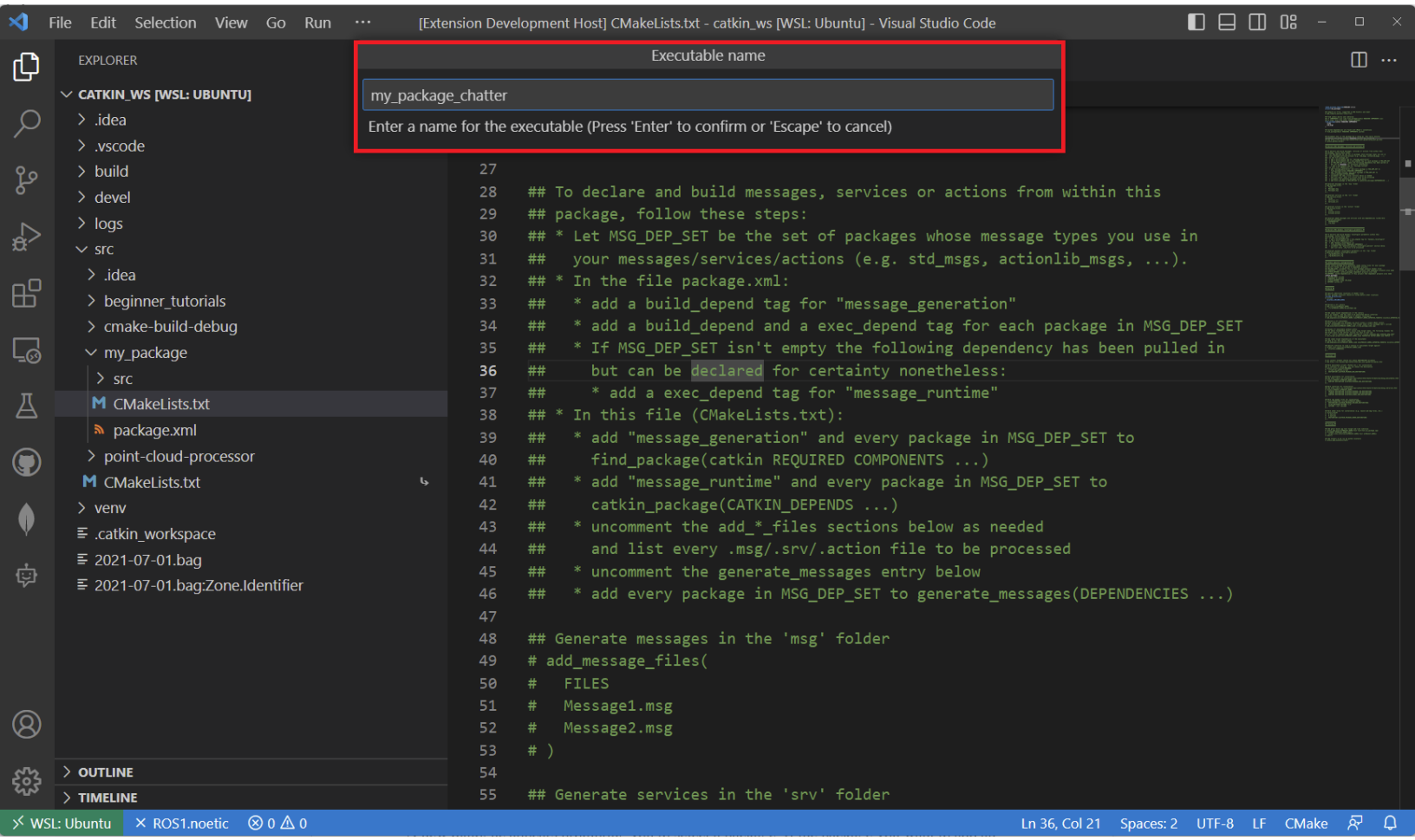
In order to build ROS applications with C++ files, we need to tell CMake to build executables.

Step 1. Enter the Command Palette (Ctrl + Shift + P) and select `R-IDE: Add an executable to the package`.

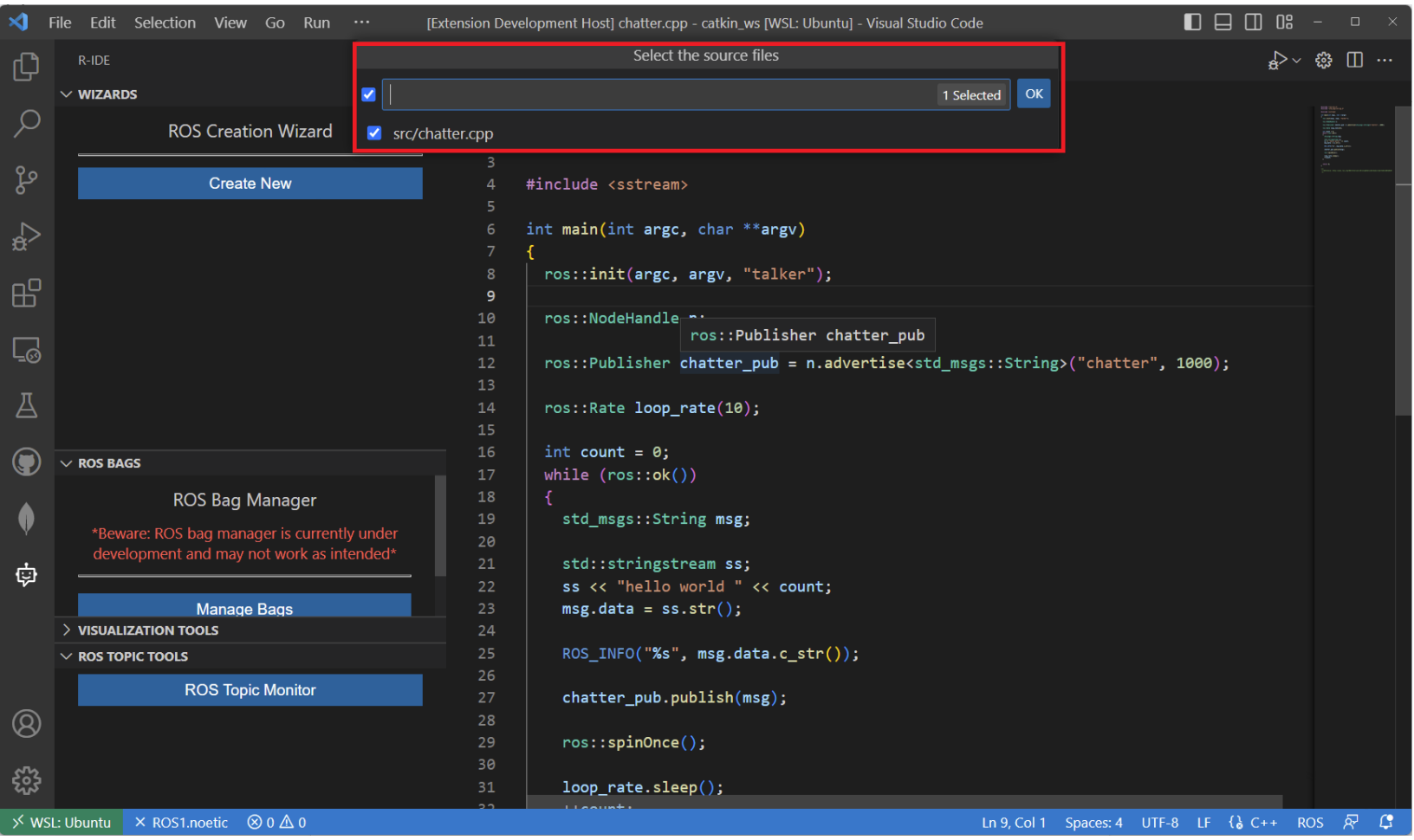




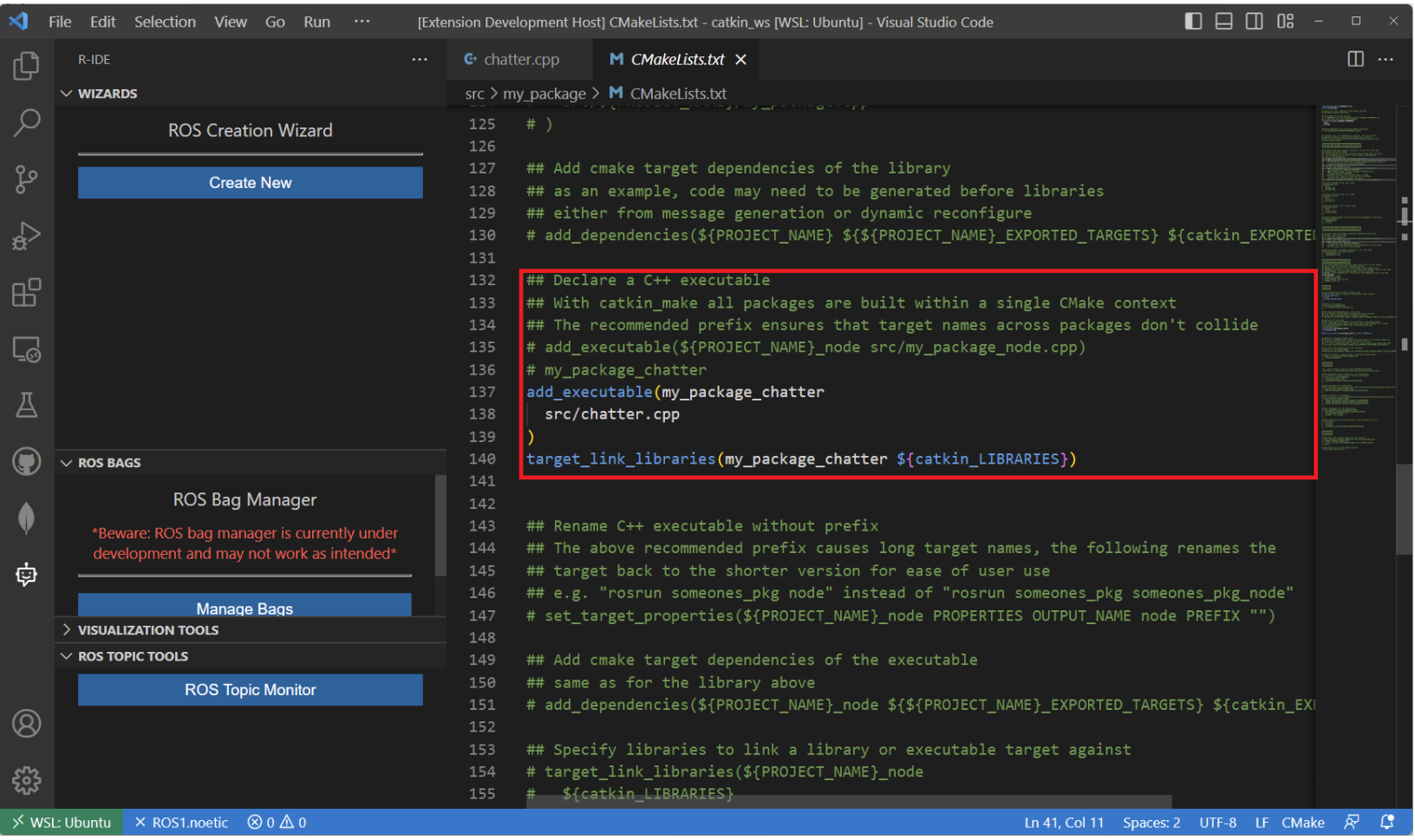
Step 2. In the area indicated by the Red Box: A new dialogue appears prompting you to select a package. If the package you want to add an executable to does not appear, make sure that it is placed in the `catkin_ws/src` directory and contains a `CMakeLists.txt` and `package.xml` file.



Step 3. In the area indicated by the Red Box: Give the new executable a name. The package name is provided as a prefix.



Step 4. In the area indicated by the Red Box: The source files that appear in the package will present as a list. Use the menu to select the source files that make up the executable. If a source file is missing, make sure that it appears within the package's directory.

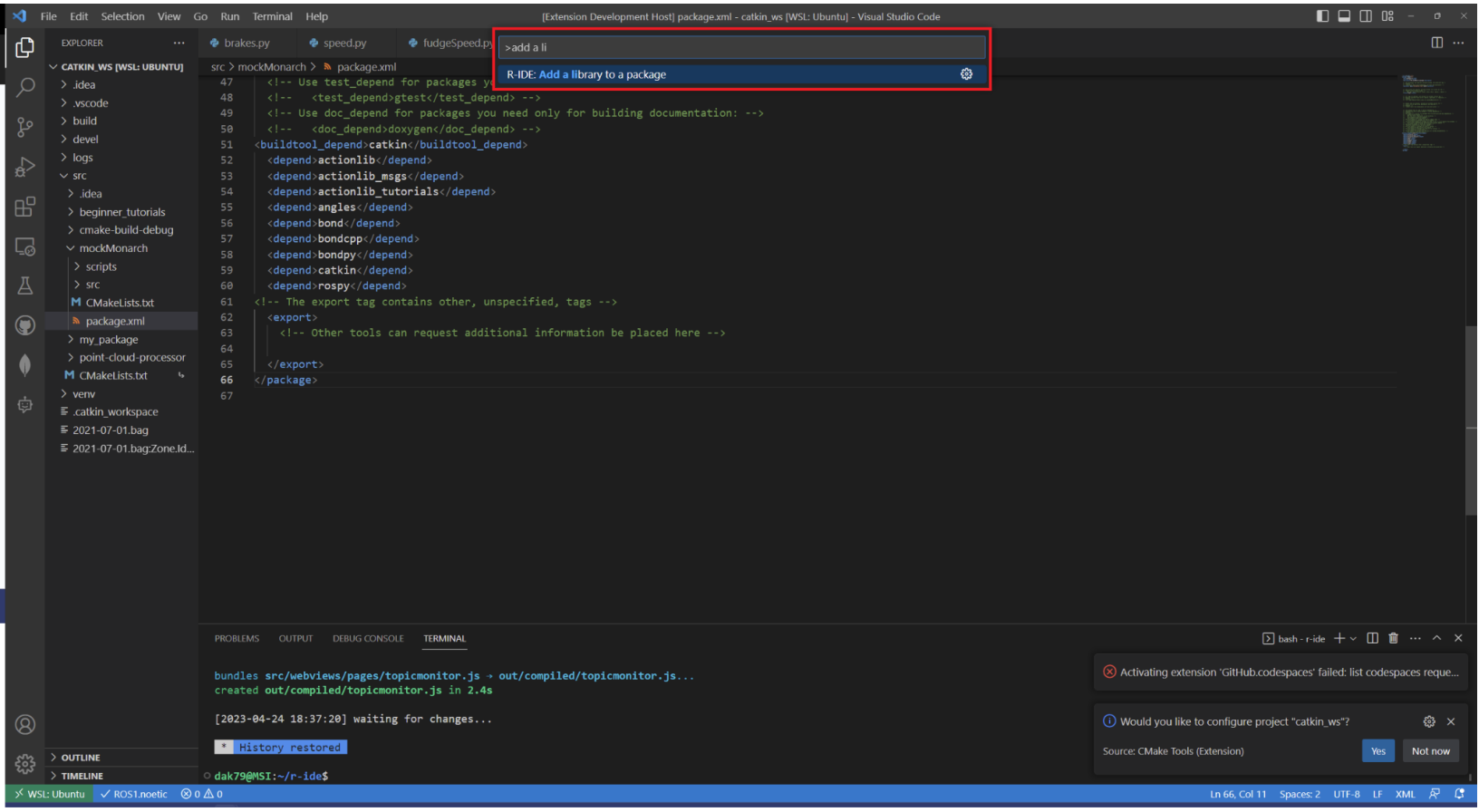


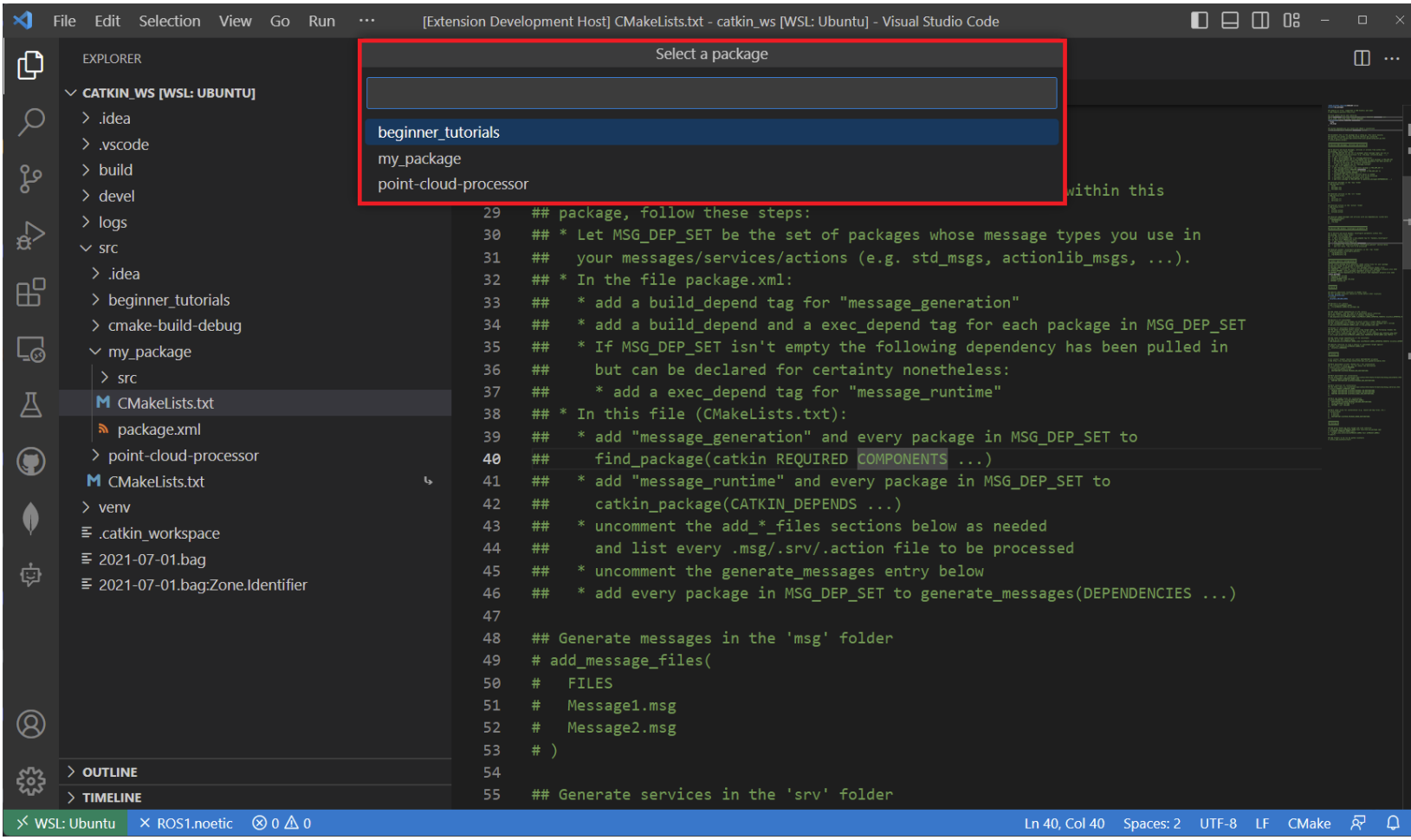
Step 5. In the area indicated by the Red Box: The executable will now appear in the `CMakeLists.txt` file under declaring an executable.

7. Create Library (Dan Koontz)

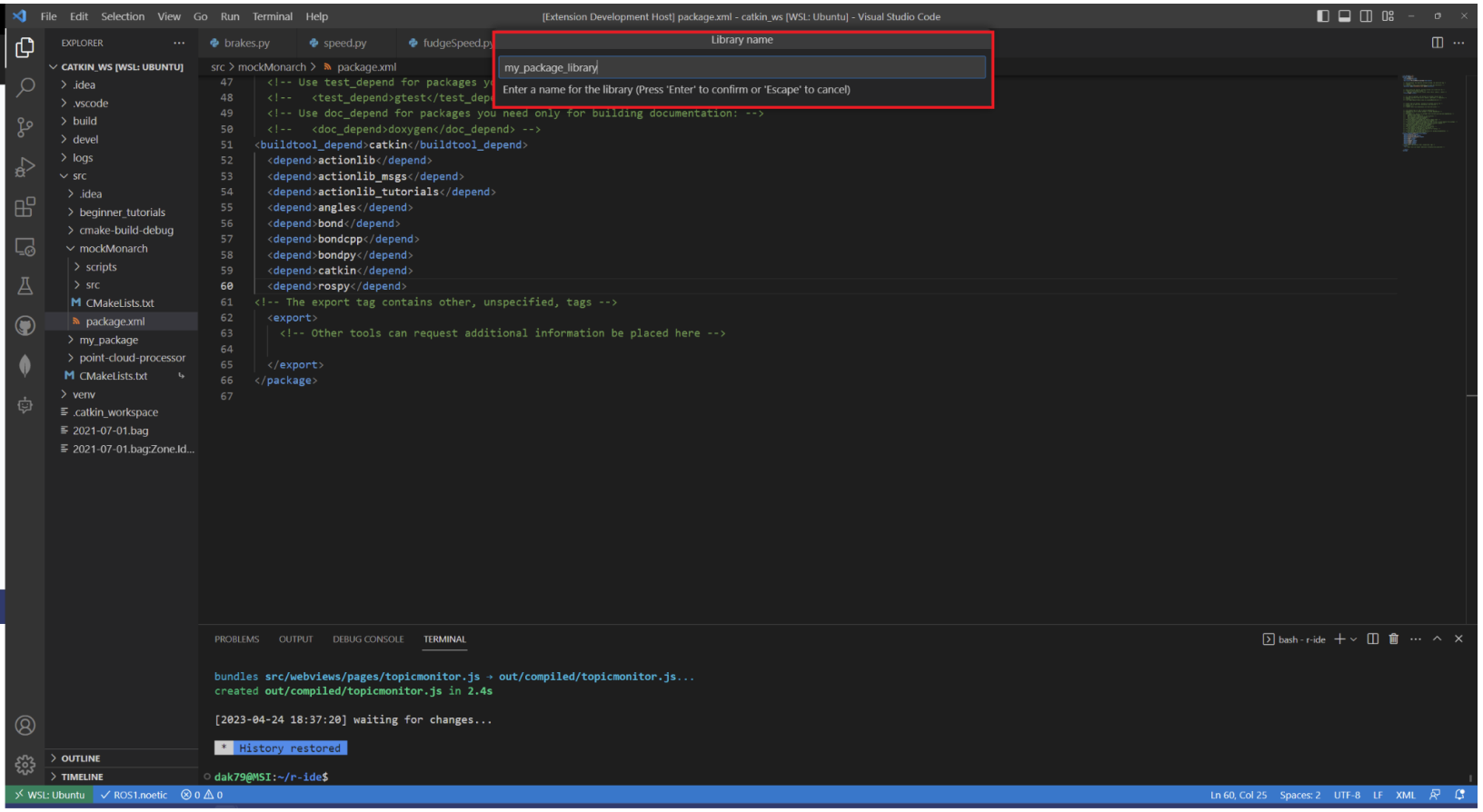
In order to create libraries out of C++ files, we need to tell CMake to build a library in the `CMakeLists.txt` file.

Step 1. Enter the Command Palette (Ctrl + Shift + P) and select `R-IDE: Add a library to the package`.

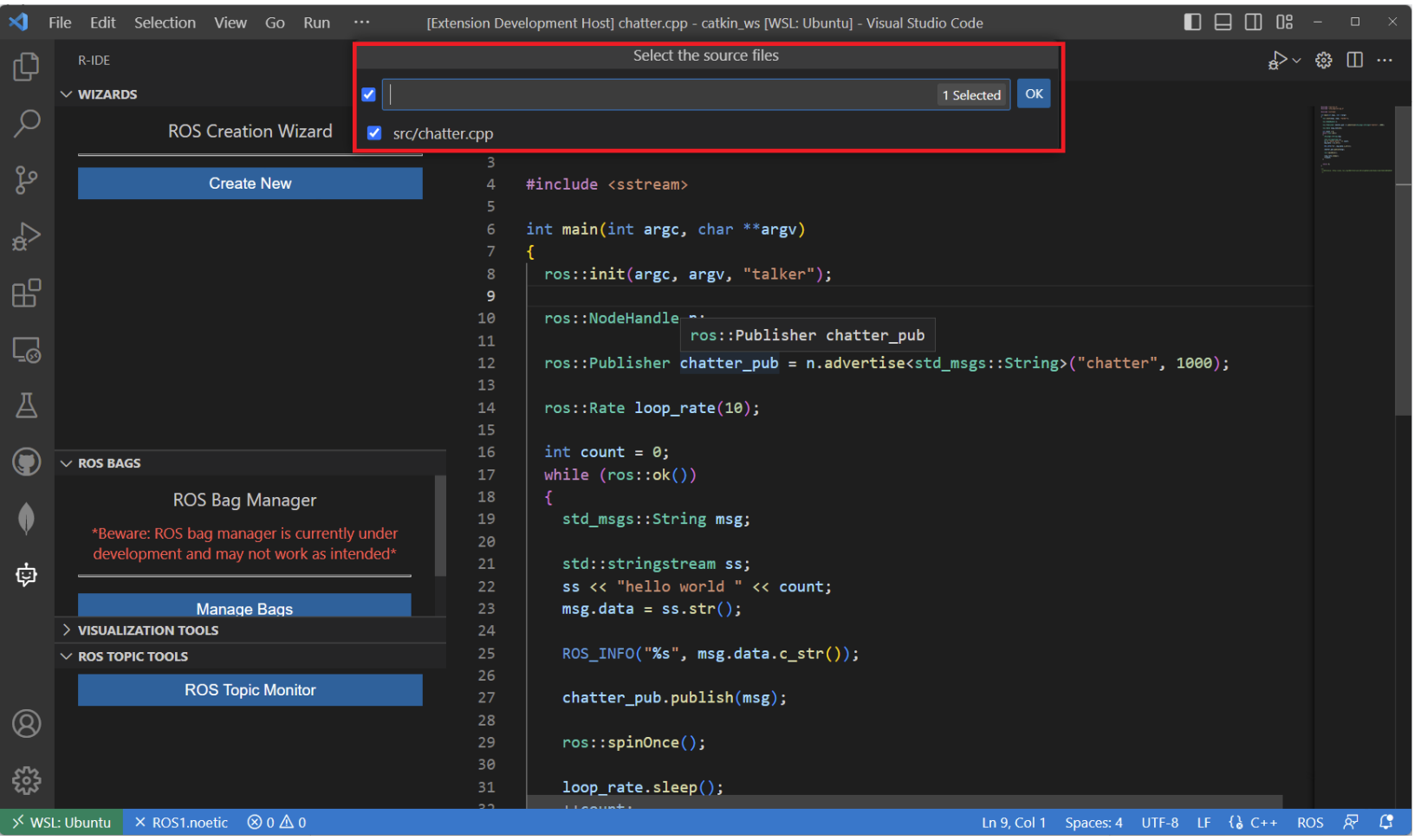




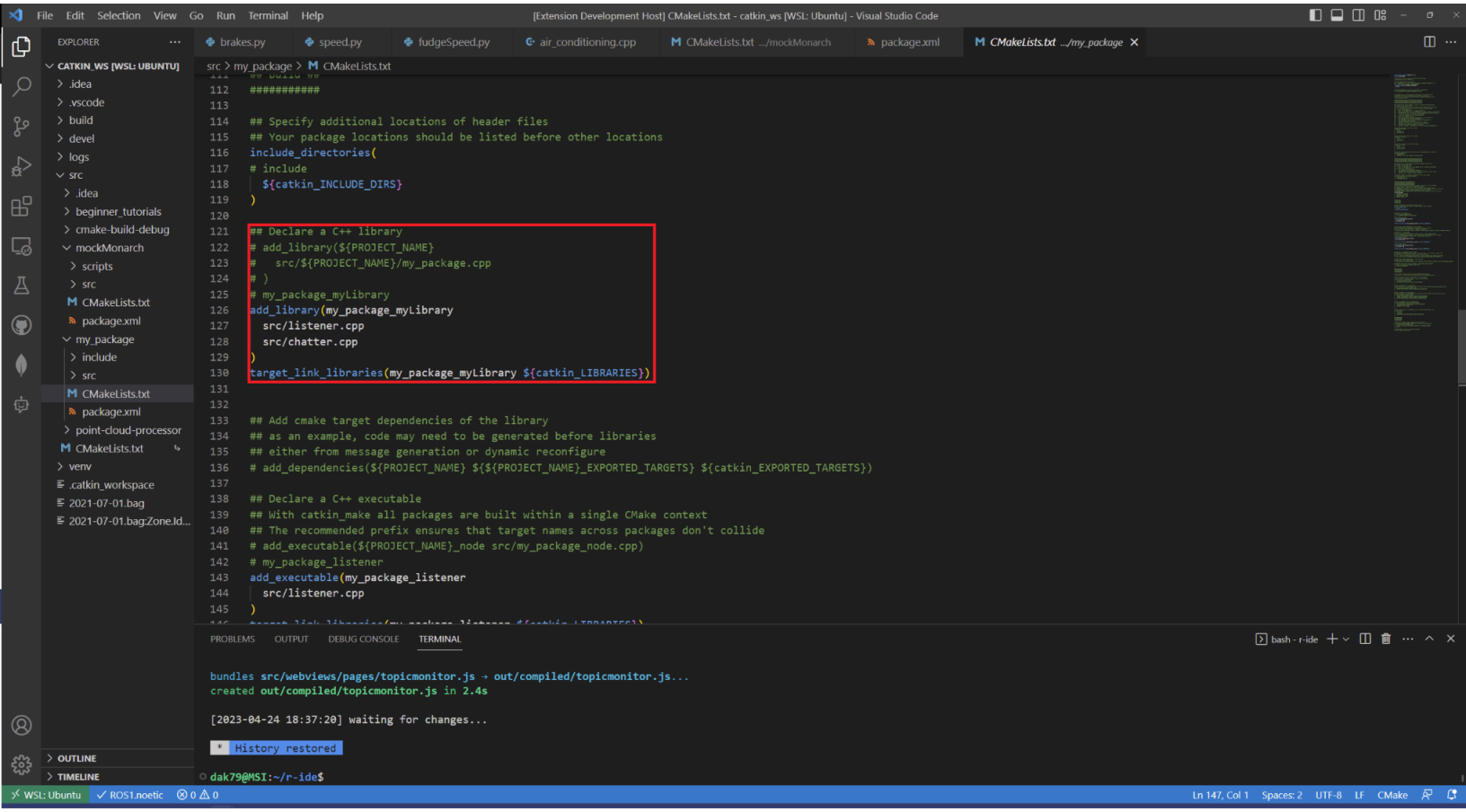
Step 2. In the area indicated by the Red Box: A new dialogue appears prompting you to select a package. If the package you want to add a library to does not appear, make sure that it is placed in the `catkin_ws/src` directory and contains a `CMakeLists.txt` and `package.xml` file.



Step 3. In the area indicated by the Red Box: Give the new library a name. The package name is provided as a prefix.



Step 4. In the area indicated by the Red Box: The source files that appear in the package will present as a list. Use the menu to select the source files that make up the library. If a source file is missing, make sure that it appears within the package's directory.

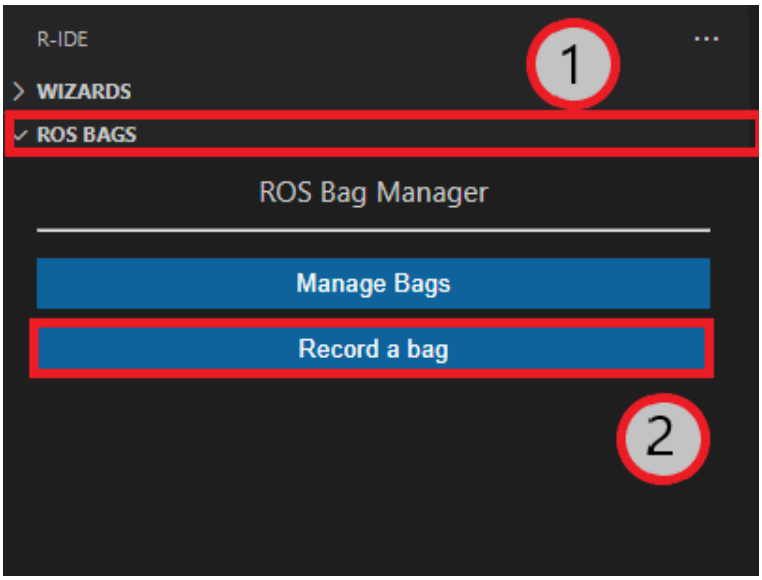


Step 5. In the area indicated by the Red Box: The library will now appear in the `CMakeLists.txt` file under declaring a library.

8. ROS Bag Manager (Dominik Soos)

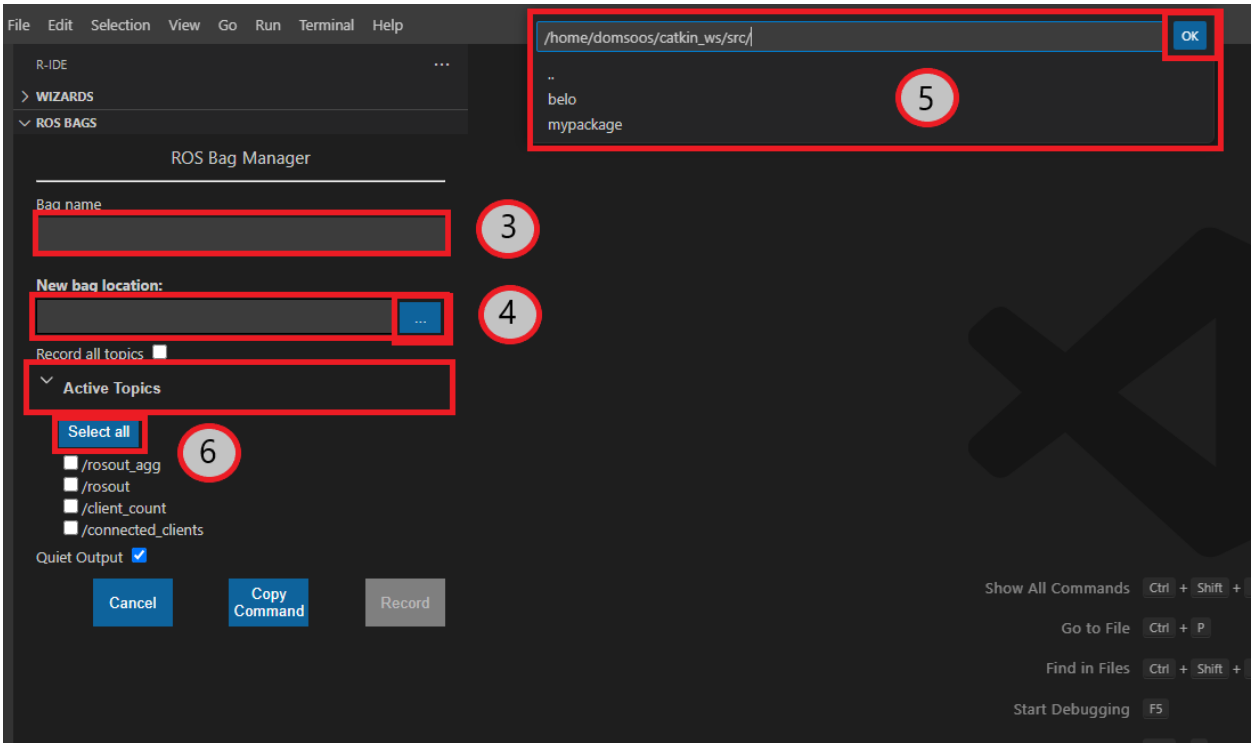
After successfully establishing connection in section 5.2.
If the connection is not established, click on connecting the ROS Bridge

8.1 Recording

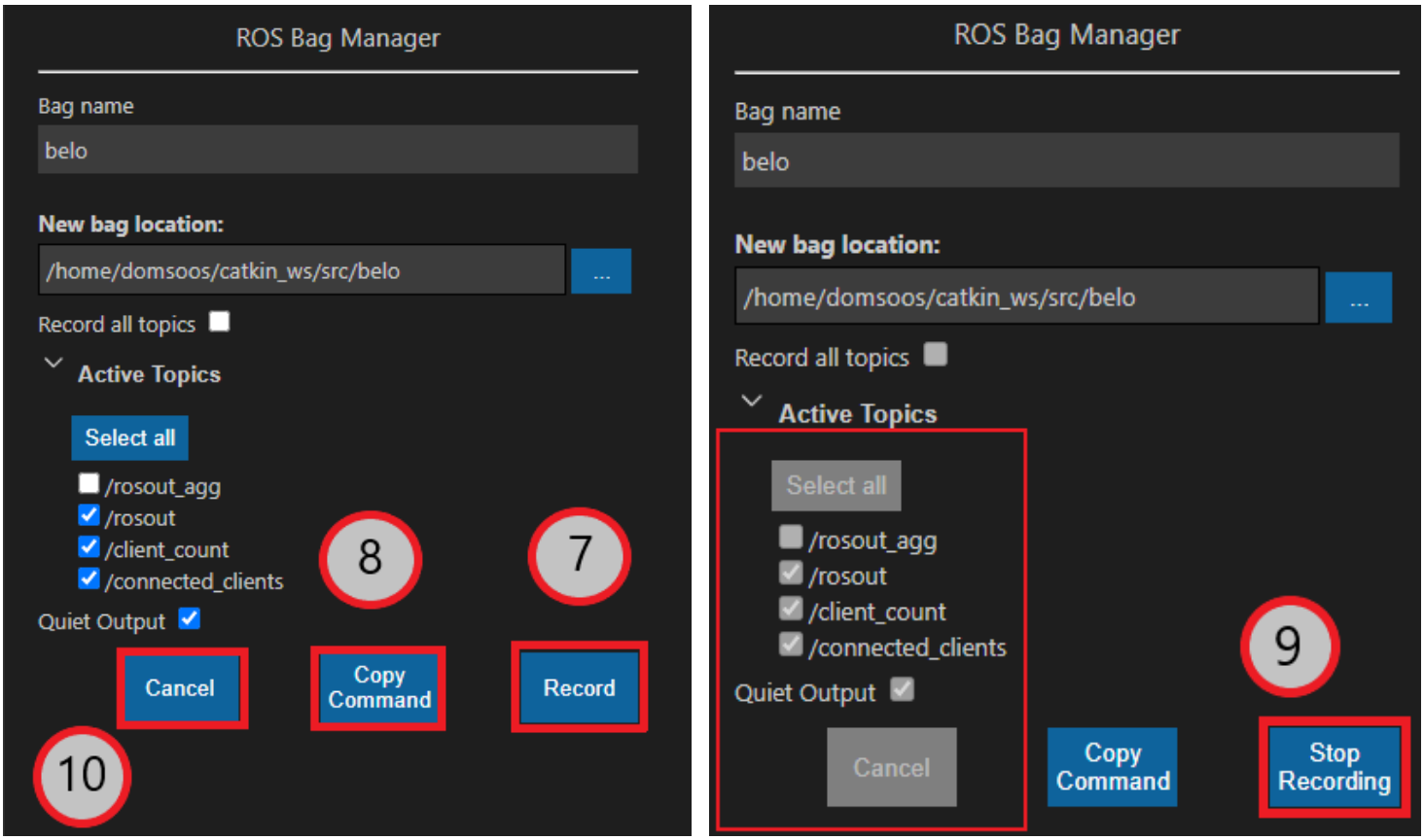


Step 1:
Open the ROS Bag Manager by clicking on ROS Bags expandable.

Step 2:
Click on Record a Bag tab to open the dialog for recording the currently executing ROS Topics to a new ROS Bag.

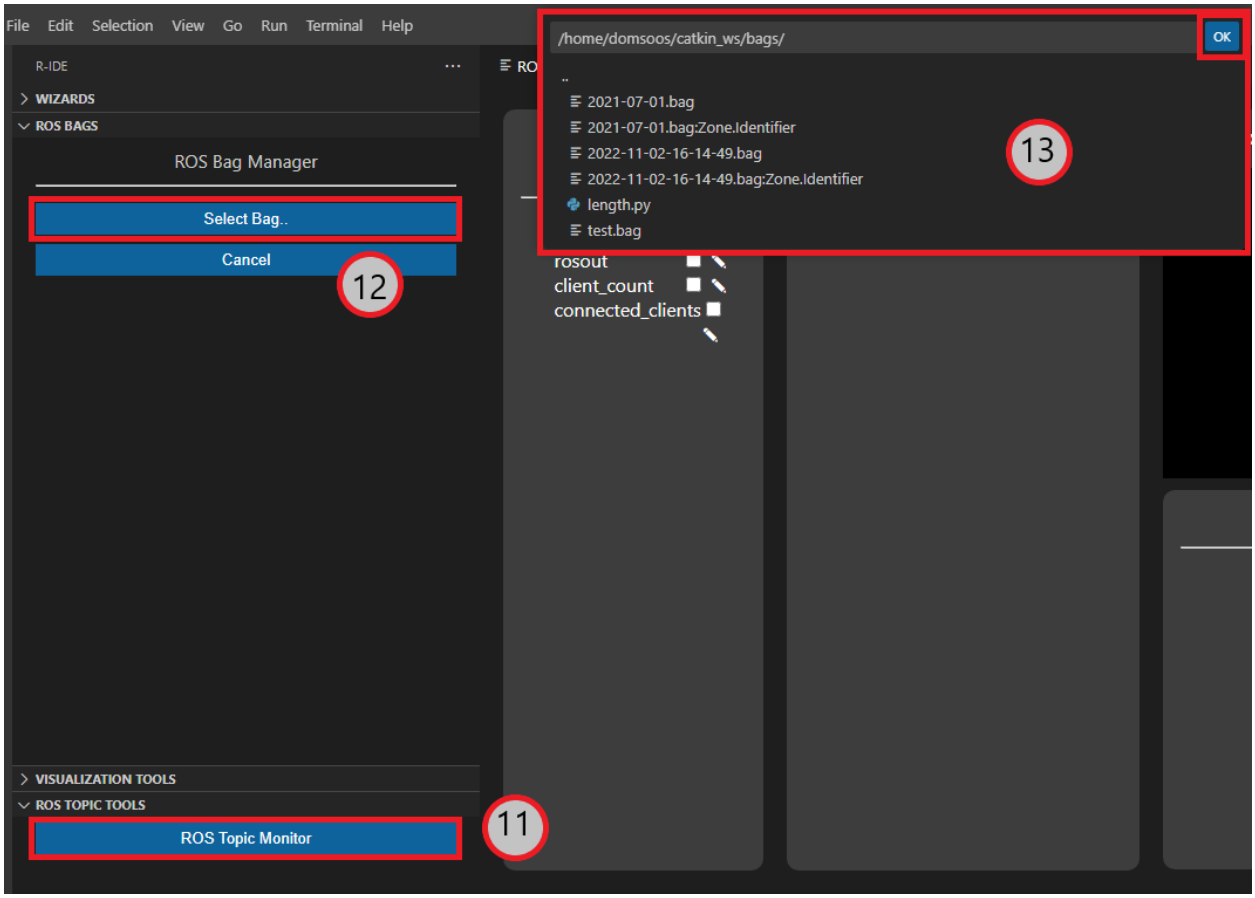


Step 3: The name of the ROS Bag may be specified but it is not mandatory. It is optional since ROS uses the date as default value for the name.
Step 4: The location for the recorded ROS Bag to be saved has to be specified in the dialog. It is a mandatory requirement for the Record button to show up.
Step 5: Once the location has been specified, press OK
Step 6: Select the desired ROS Topics to be recorded. There is an option to select all currently active ROS Topics.

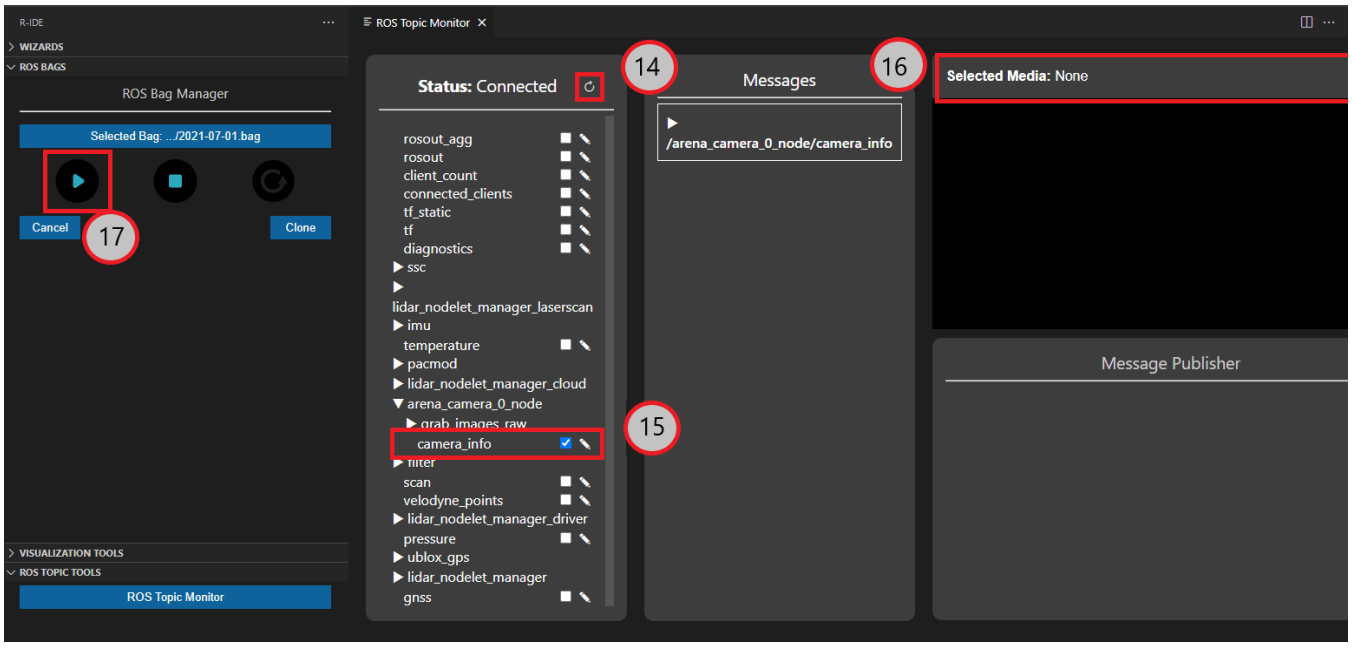


Step 7: Click on Record to start recording the bag.
Step 8: Alternatively, the user can copy the command, modify and execute it as needed.
Step 9: Click the Stop Recording button to stop the recording and save the file. Notice that adding and removing topics are disabled while recording a ROS Bag.
Step 10: After clicking Stop Recording, the user may cancel to play the recorded ROS Bag.

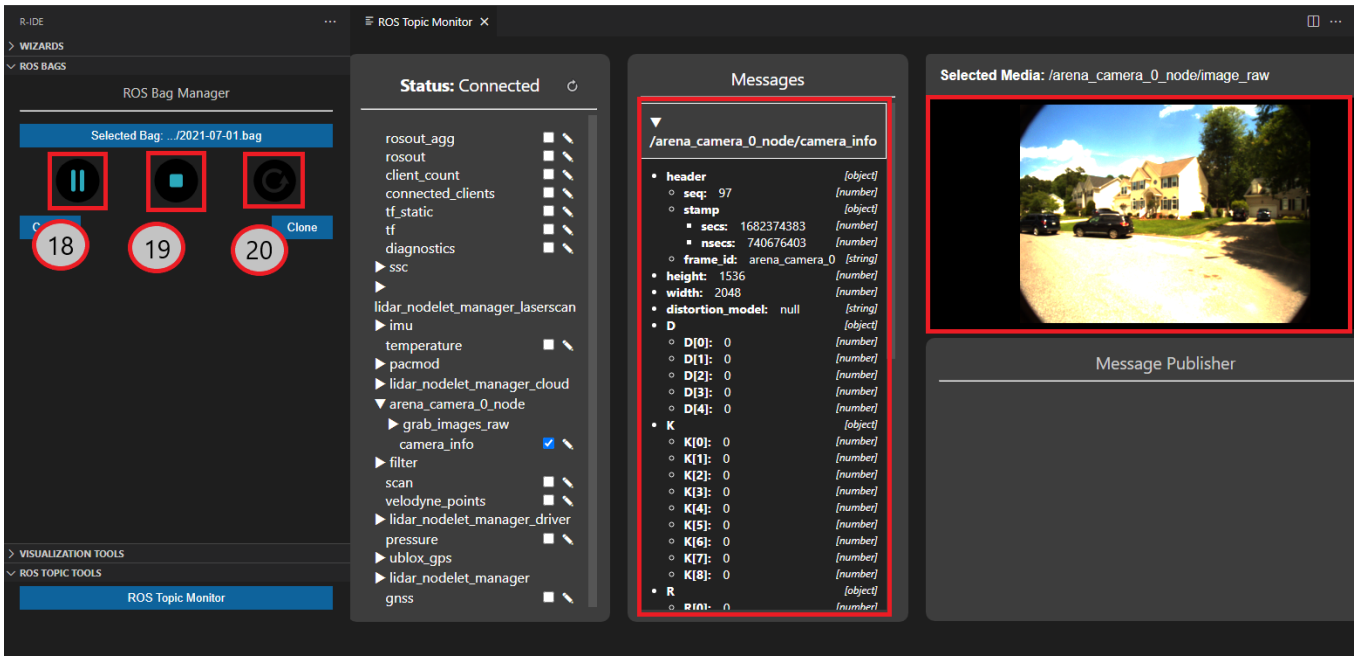
8.2 Playback



Step 11: Click on the ROS Topic Monitor to view the active topics
Step 12: Select the desired ROS Bag to play in the catkin workspace by:
Step 13: Specify the path to the ROS Bag and click OK.



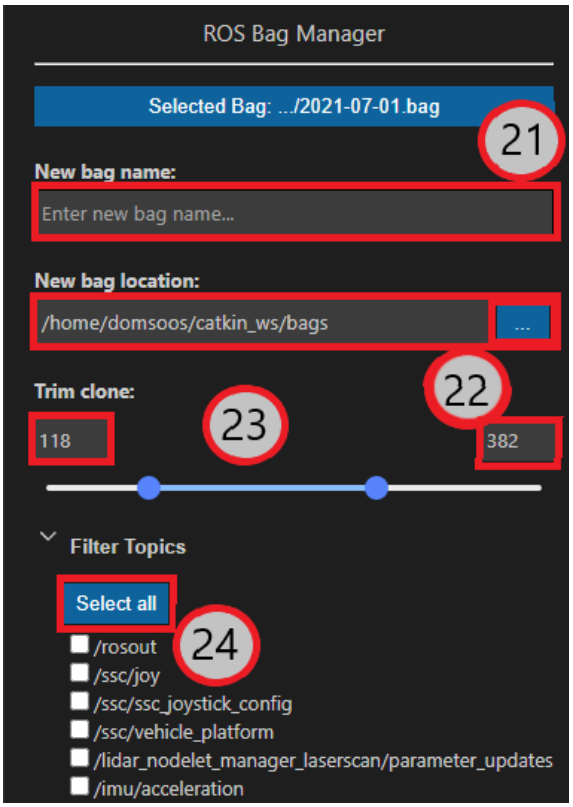
Step 14: Refresh the ROS Topic Monitor, so that the updated ROs Topics are displayed from the selected ROS Bag
Step 15: Look for the camera_info topic and check the box to be subscribed for that topic
Step 16: Select the media file if the user would like to display media, as in section 5.3.
Step 17: Press the play button to start playing the bag.

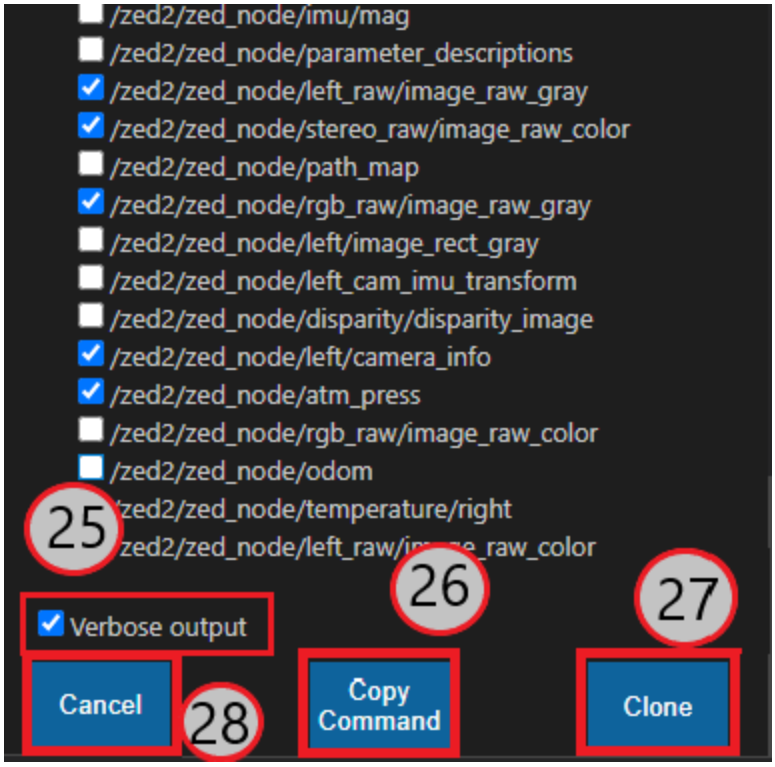


Users have the option to pause, stop and replay the currently playing ROS Bag:
Step 18: Pause the ROS Bag
Step 19: The stop button pauses and resets the ROS Bag to the beginning of the bag.
Step 20: The replay button resets the ROS Bag to the first sequence without stopping the ROS Bag. On the right side, the user can view the incoming messages for the media stream displayed on the far right.

8.3 Cloning

The user has the ability to clone a selected ROS Bag. If the user would like to select a new ROS Bag, complete Steps 11-13 and click on the Clone button.





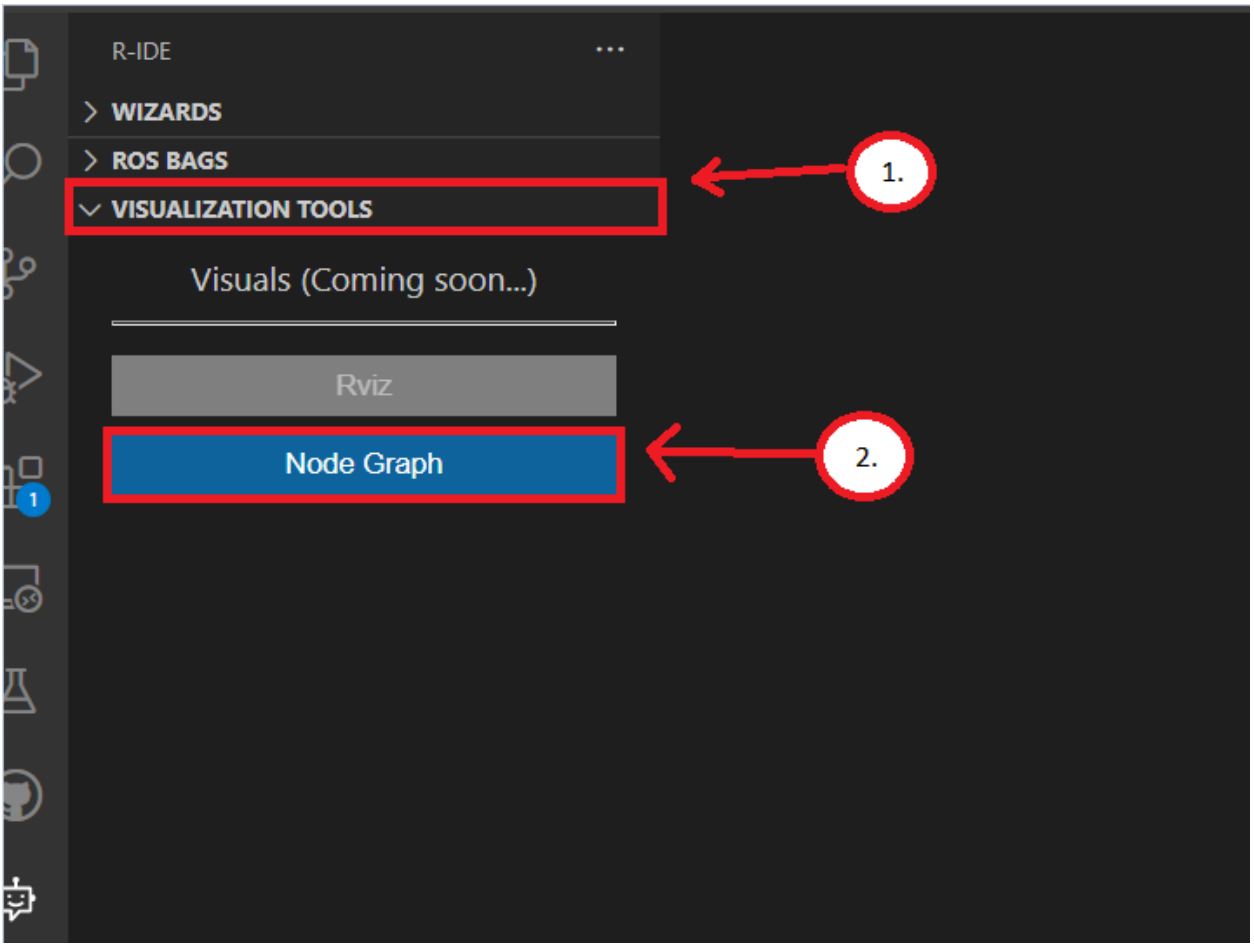
Step 21: Enter the name of the Bag to be saved
Step 22: Specify the location to clone the bag in
Step 23: The user can trim clone a ROS Bag by selecting an interval from the bag to trim using the slider. Initially it is specified to the start and end time of the bag.
Step 24: The user has the ability to select the topics to clone, by either selecting them manually or by selecting the ‘Select all’ option to select all the topics in the selected ROS Bag.

As before, If the user did not specify a name to the bag, then the default time will be assigned to the name.

Step 25: We can clone so that it use verbose output
Step 26: Copy command and execute it ourselves in a new terminal
Step 27: Click the clone button to clone the ROS Bag with the specified features
Step 28: Cancel the entire cloning process

9. Node Graph (Justin Tymkin)

9.1 Open node graph

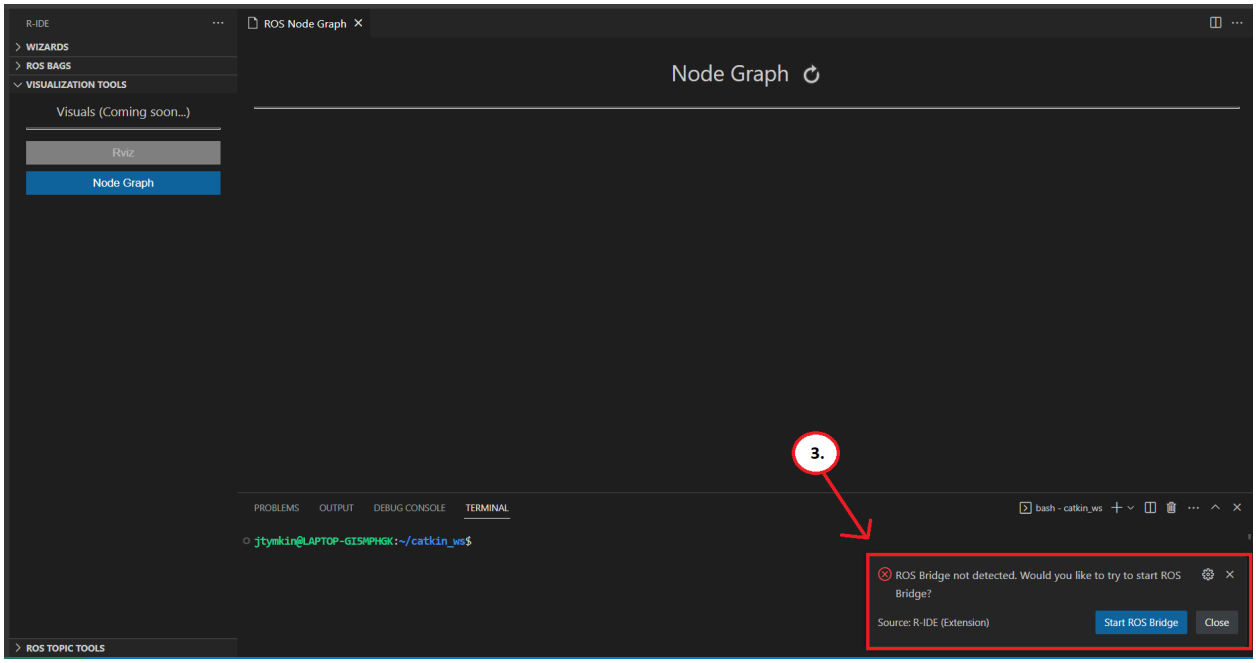


To use the Node Graph,

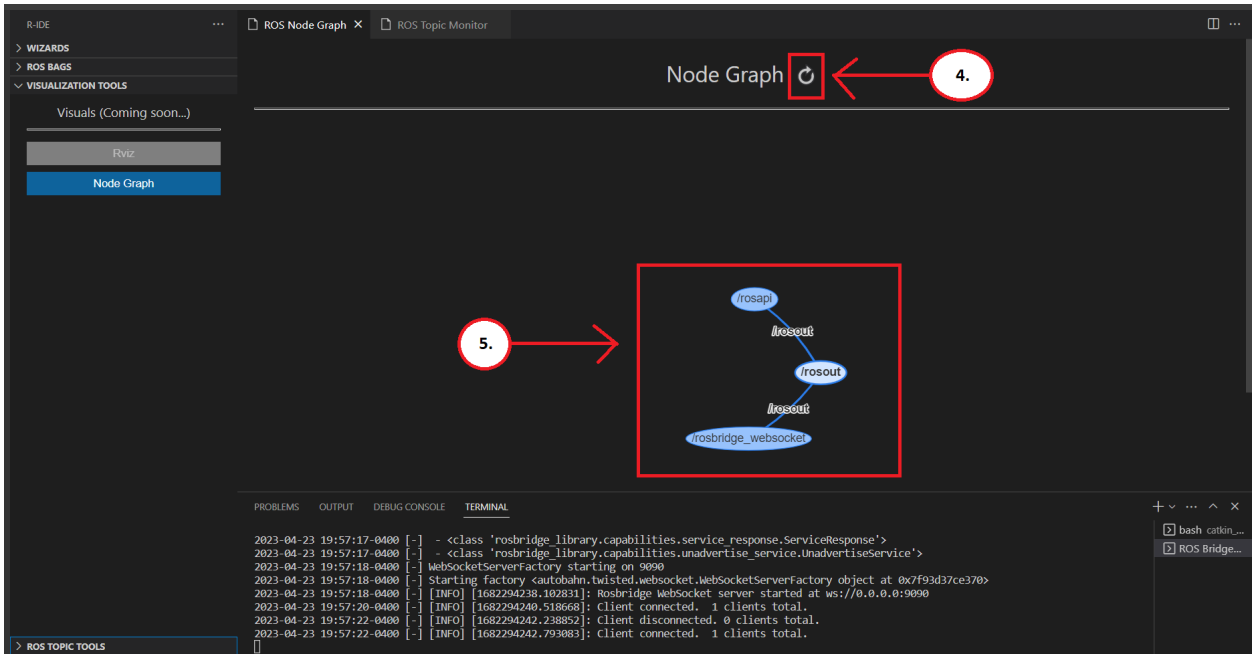
Step 1: Click the Visualization tools button (1) to expand the Visualization tools view.

Step 2: After the view is expanded, click the Node Graph button (2) to open the Node Graph.

9.2 ROS bridge

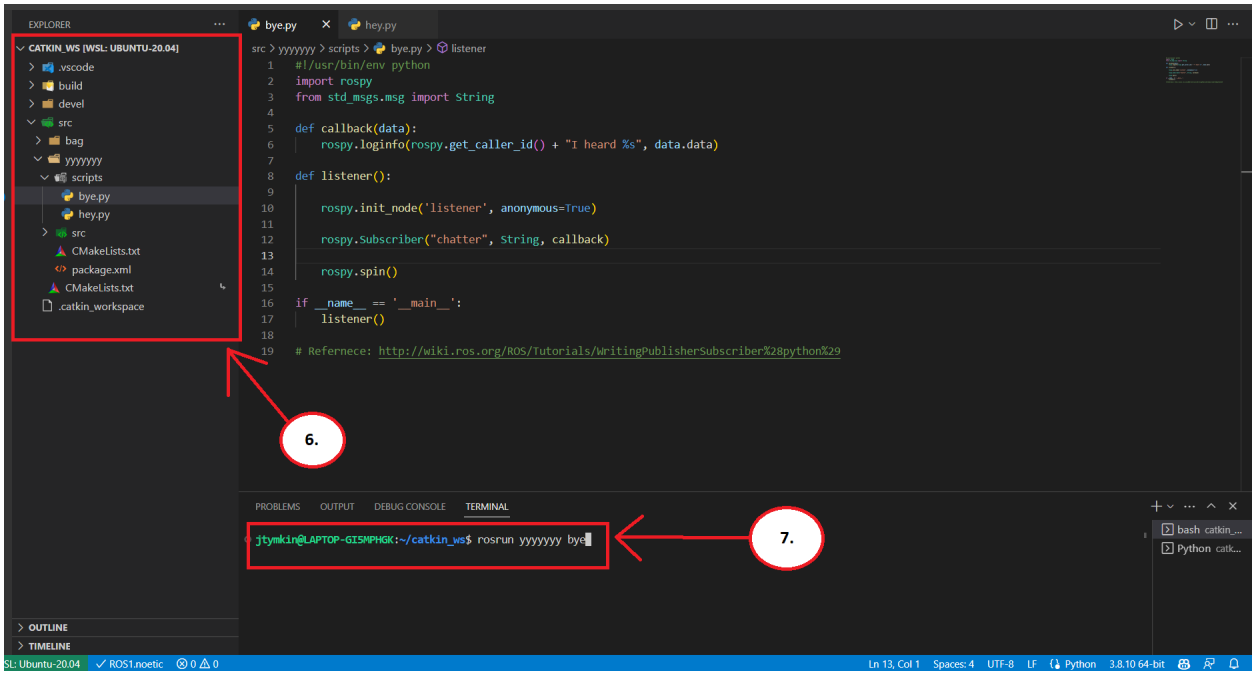


Step 3. After selecting the Node Graph you'll be asked to connect to ROS bridge (3) to establish communication with ROS, select start ROS bridge.



Step 4. After connecting to the ROS bridge, click the refresh button (4), The Node Graph should now appear (5) showing communication from rosout to the ROS API and the ROS bridge websocket.

9.3 create nodes

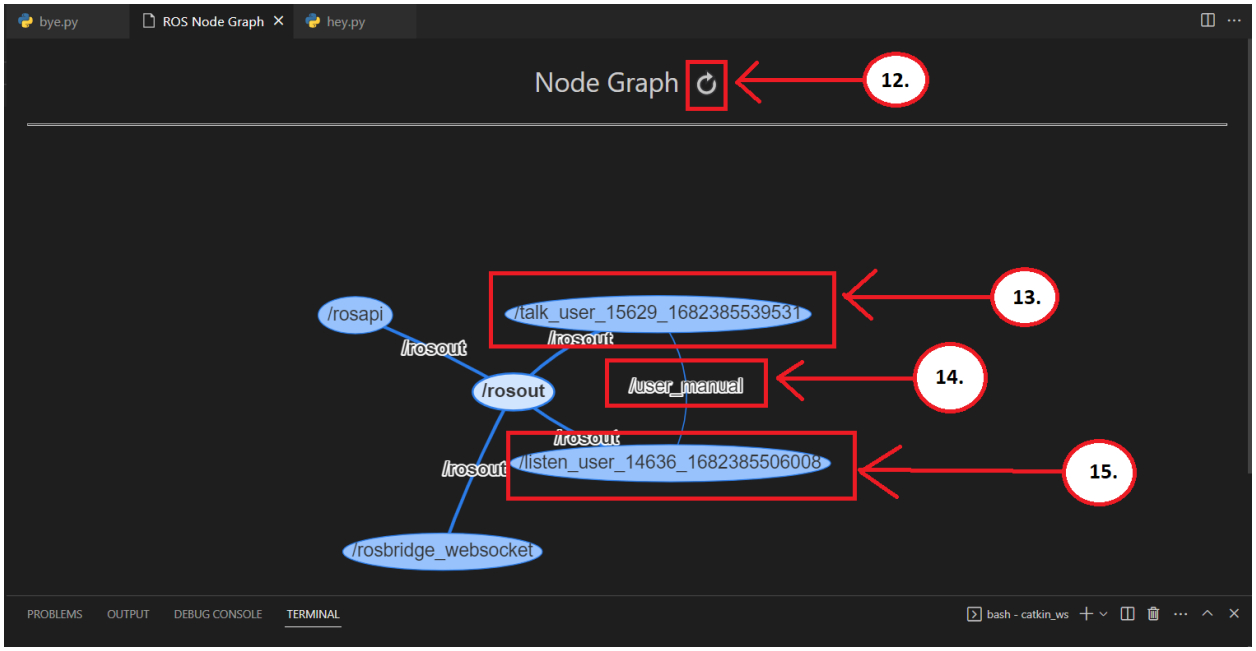


Step 5. To create a talker or listener node (6), follow the appropriate steps (illustrated in section 4 creation wizard).

Step 6. Once your node is created, execute the node in the terminal (7).

Step 7. After executing the node, take note of the name of the topic (8) being used for communication between nodes in your publisher function and the name of your publisher node (9) in the initialize node function. The topic should be the same for the publisher and listener node.

Step 8: Similarly for the listener node, after executing the node take note of the name of the topic (10) being used for communication between nodes in the subscriber function and the name of your listener node (11) in the initialize node function. The topic should be the same for the publisher and listener node.

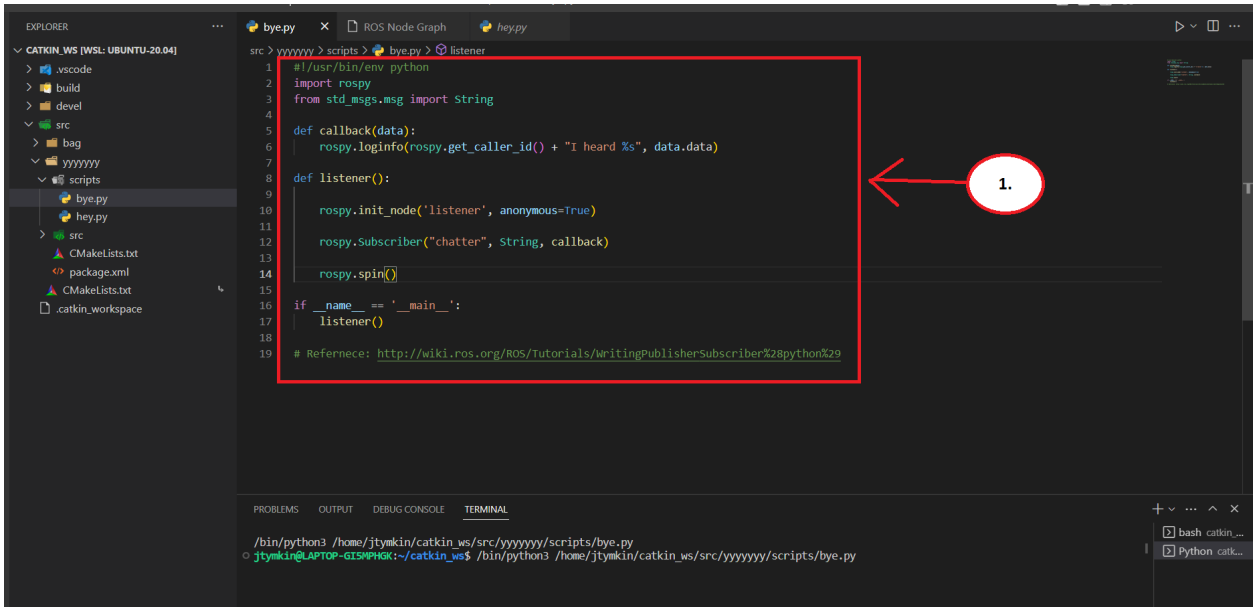


Step 9: Refresh the graph again (12) after executing the node to confirm that it is active and communicating properly with the rosout.

Step 10:. Ensure that all nodes are active and communicating directly with each other, your publisher node (13) and topic (14) should be the same from step 7. Your listener node (15) and topic (14) should be the same from step 8. If you need to create additional talker or listener nodes to communicate with a previously created node, repeat the necessary steps.

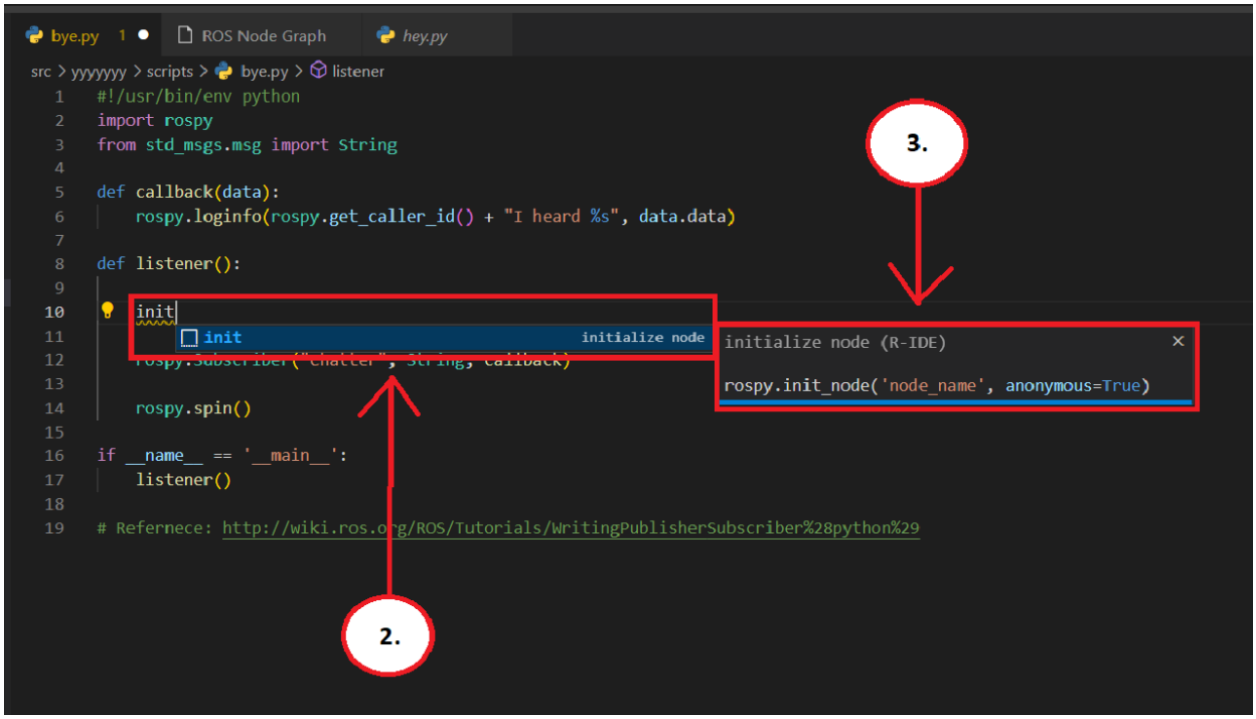
10. Snippets (Justin Tymkin)

10.1 Code template

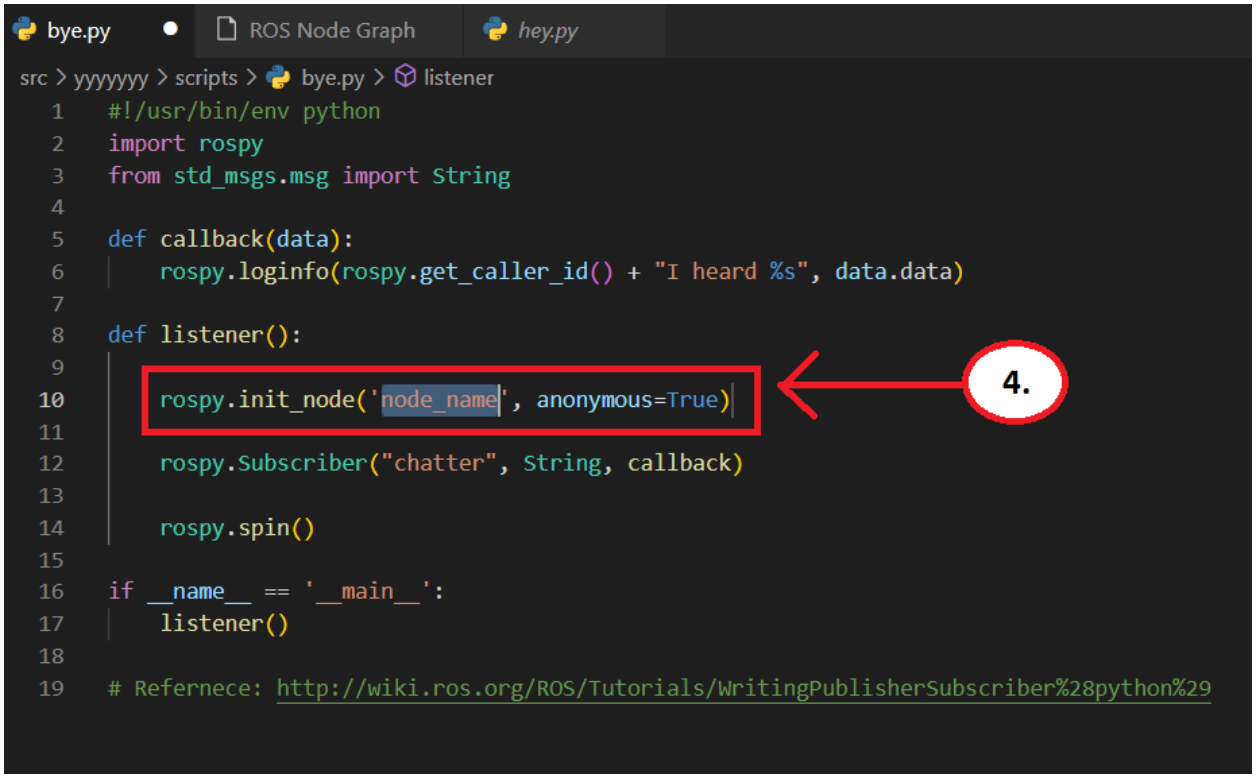


Step 1. When using the creation wizard in section 4 to create a node, the resulting file will be pre-populated with a code template (1) from the ROS wiki that serves as an example of what a node might look like.

10.2 Code snippet



Step 2. Once you have created your node using the wizard, you can then use code snippets to create additional nodes. By making a call (2) to the desired snippet, you can quickly insert a block of code. This shortcut will be labeled with the name of the code snippet (3), which you can use to easily locate and reference it.



Step 3. Selecting the code snippet name will then create a generic block of code for the user to tab through (4), defining the necessary variables to customize the code for their specific use case. All snippets can be found in a catalog here:

https://github.com/domsoos/r-ide/blob/main/snippetCatalogue.md