Bridging Your Transitions from ROS 1 to ROS 2

Mabel Zhang
Open Robotics
WHO, WHAT, WHY, W...
Overview

**WHO** cares
You run on ROS 1 now.
You(r boss) want to move to ROS 2.

**WHAT** is the bridge
Talks between ROS 1 and ROS 2

**WHY** would I bother
Stuff don’t break while you port incrementally to ROS 2
Disclaimer:
I didn’t write the bridge.
(I’m as newb as you.)
Outline

Built-in types
Matching custom types (automatically associated)
Non-matching custom types (NOT automatically associated)

All code available: github.com/mabelzhang/ros1_bridge_sandbox
HOW
Setup

A ROS 1 Robot (e.g. VRX)  
ROS 1 Melodic  
ROS 2 Dashing

All code available: github.com/mabelzhang/ros1_bridge_sandbox

Dockerfile
BASICS: BUILT-IN TYPES
Use Case

If you only ever use built-in types. No custom types.

std_msgs
sensor_msgs
geometry_msgs
std_srvs
...

All code available: github.com/mabelzhang/ros1_bridge_sandbox
Built-in Types: Work Out of the Box

Install bridge:
$ sudo apt install ros-dashing-ros1-bridge

See all built-in types automatically bridged:
$ ros2 run rosl1_bridge dynamic_bridge --print-pairs

Run bridge:
$ ros2 run rosl1_bridge dynamic_bridge --bridge-all-topics

All code available: github.com/mabelzhang/rosl1_bridge_sandbox
Demo: Built-in Types

ROS 1

sensor data

ROS 2

All code available: github.com/mabelzhang/ros1_bridge_sandbox
All code available: [github.com/mabelzhang/ros1_bridge_sandbox](https://github.com/mabelzhang/ros1_bridge_sandbox)
Demo: Built-in Types

ROS 1

command

ROS 2

All code available: github.com/mabelzhang/ros1_bridge_sandbox
ROS 1

bridge

ROS 2
INTERMEDIATE: CUSTOM TYPES, AUTO-ASSOCIATED
Use Case

If you define custom types, e.g.

```
JointCommand.msg:
float64 position
```

All code available: github.com/mabelzhang/ros1.bridge_sandbox
Custom Types: Automatic association

Need to **recompile bridge**.

Easiest: automatic association between ROS 1 & 2 types

**Conditions:**

- **Package names** end in `_msgs` in ROS 1, `_msgs / _interfaces` in ROS 2 AND
- **Message names** match AND
- **Fields** match

All code available: [github.com/mabelzhang/ros1_bridge_sandbox](github.com/mabelzhang/ros1_bridge_sandbox)
Custom Types: Compiling the Bridge

Key: Separate workspaces and separate shells, for ROS 1, ROS 2, and bridge

- ros1_bridge_ws/src/ros1_bridge
- ros1_msgs_ws/src/bridge_msgs
- ros2_msgs_ws/src/bridge_msgs

Compile and source each workspace in separate shells.

All code available: github.com/mabelzhang/ros1_bridge_sandbox
Custom Types: Compiling the Bridge

Shell 1: Build ROS 1 messages
$ . /opt/ros/melodic/setup.bash
$ catkin_make_isolated --install
$ . install_isolated/setup.bash

Shell 2: Build ROS 2 messages
$ . /opt/ros/dashing/setup.bash
$ colcon build --packages-select bridge_msgs
$ . install/local_setup.bash

All code available: github.com/mabelzhang/ros1_bridge_sandbox
Custom Types: Compiling the Bridge

Shell 3: Build bridge

$ . /opt/ros/melodic/setup.bash
$ . /opt/ros/dashing/setup.bash
$ . ../ros1_bridge_sandbox/ros1_msgs_ws/install_isolated/setup.bash
$ . ../ros1_bridge_sandbox/ros2_msgs_ws/install/local_setup.bash
$ colcon build --packages-select ros1_bridge --cmake-force-configure
$ . install/local_setup.bash

Verify custom messages are bridged:

$ ros2 run ros1_bridge dynamic_bridge --print-pairs | grep bridge_msgs
- 'bridge_msgs/JointCommand' (ROS 2) <=> 'bridge_msgs/JointCommand' (ROS 1)

Instructions Available

All code available: github.com/mabelzhang/ros1_bridge_sandbox
Demo: Custom Type

ROS 1

custom msg

ROS 2

All code available: github.com/mabelzhang/ros1.bridge.sandbox
bridge

ROS 1
pub
custom

ROS 2
custom
Demo: Custom Type

ROS 1

custom msg

ROS 2

All code available: github.com/mabelzhang/ros1_bridge_sandbox
bridge

ROS 1
sub
custom

ROS 2
pub
custom
Part 3/3

IF YOU MUST: CUSTOM MAPPINGS, NOT AUTO-ASSOCIATED
Use Case

If message types cannot have matching names and fields, e.g. /rosout:

```
rosgraph_msgs/Log.msg:
Header header
byte level
string name
string msg
string file
string function
uint32 line
string[] topics
```

```
rcl_interfaces/Log.msg:
builtin_interfaces/Time
stamp
uint8 level
string name
string msg
string file
string function
uint32 line
```

All code available: github.com/mabelzhang/ros1_bridge_sandbox
Define Custom Mapping in YAML

```yaml
rcl_interfaces/mapping_rules.yaml:

ros1_package_name: 'rosgraph_msgs'
ros1_message_name: 'Log'
ros2_package_name: 'rcl_interfaces'
ros2_message_name: 'Log'
fields_1_to_2:
  header.stamp: 'stamp'
  level: 'level'
  name: 'name'
  msg: 'msg'
  file: 'file'
  function: 'function'
  line: 'line'
```

All code available: github.com/mabelzhang/ros1_bridge_sandbox
Define Custom Mapping in YAML

```
rcl_interfaces/CMakeLists.txt:
install(FILES mapping_rules.yaml DESTINATION share/${PROJECT_NAME})
```

```
rcl_interfaces/package.xml:
<export>
  <ros1_bridge mapping_rules="mapping_rules.yaml"/>
</export>
```

Recompile package; recompile bridge as before.

All code available: github.com/mabelzhang/ros1_bridge_sandbox
Demo: Custom YAML Mapping

All code available: github.com/mabelzhang/ros1_bridge_sandbox
bridge

ROS 1

sub

ROS 2
custom
YAML

ROS 2
pub
A FEW NOTES
A Few Notes

`echo` alone does not suffice. Write a subscriber.

Use `--bridge-all-topics` to guarantee bridging.

If cannot get dynamic bridge to work, try static bridge `simple_bridge.cpp` in `ros1_bridge`.

Migration guide from ROS 1 to ROS 2
https://index.ros.org/doc/ros2/Contributing/Migration-Guide/

All code available: github.com/mabelzhang/ros1_bridge_sandbox
Thank you.