Overview Repositories **Discussions** Projects Packages People

[Proposal] Splitting debug tools and non-runtime tools from universe repositories #4067

Unanswered **mitsudome-r** asked this question in **Ideas**

mitsudome-r on Dec 19, 2023 Maintainer

Currently, we have many packages under Autoware.Universe which are not meant to be run at runtime and only used for debugging and evaluation. This makes the repository huge and also makes people who is not interested in such tools to wait for longer built time.

Also, we have packages that are hosted by third party members which are not visible by the users.
examples: calibration tool, mapora

I would like to collect any ideas which makes the management of such packages easier.

Proposal

- Create a repositories which collects packages for non-runtime tools or packages that are mainly for core developers
- Add a tools.repos under Autoware repository and put links to such packages
  This way, normal users don't have to clone and wait for build for the packages that are mainly used by developers.

I am open to any opinions or ideas so please feel free to comment in the threads.

↑ 1

Category

Ideas

Labels

None yet

8 participants

3 comments · 9 replies

Oldest Newest Top

xmfcx on Dec 19, 2023 Maintainer

Related:

- Move drivers into separate .repos file #3366

↑ 1

2 replies

xmfcx on Dec 19, 2023 Maintainer

As we discussed in the Software WG meeting, I suggest:

- Let's create an `autoware_tools` repository.
    - Move all existing universe packages to there.
- Move this new repo and any other non-runtime-dependent repos to `extra-packages.repos` file mentioned in Move drivers into separate .repos file.

❤️ 1

TakaHoribe on Dec 20, 2023 Maintainer

I agree with creating a repository on Autoware Foundation and managing the code separately using a `.repos` file.

mitsudome-r on Jan 9 Maintainer Author edited

I have created an empty autoware_tools repository.
I'm going to copy some of the workflows and PR templates from Autoware Universe so that we can run build checks and other CIs after moving some packages to the new repo.

Also, here are some of the candidate packages that we could move to autoware_tools repository from autoware.universe:

- accel_brake_map_calibrator
- bag_time_manager_rviz_plugin
- control_performance_analysis
- control_validator
- dummy_diag_publisher
- dummy_infrastructure
- ~~dummy_perception_publisher~~ (required for using planning simulator tutorial demo)
- fake_test_node
- goal_distance_calculator
- heatmap_visualizer
- image_diagnostics
- joy_controller
- kinematic_evaluator
- lanelet2_map_preprocessor
- localization_error_monitor
- localization_evaluator
- object_velocity_splitter
- path_distance_calculator
- planning_debug_tools
- planning_evaluator
- planning_test_utils

- planning_topic_converter
- ~~planning_validator~~ (Needed to publish diagnostic information at runtime)
- pose_instability_detector
- rtc_manager_rviz_plugin
- rtc_replayer
- simulator_compatibility_test
- static_centerline_optimizer
- steer_offset_estimator
- tier4_adapi_rviz_plugin
- tier4_automatic_goal_rviz_plugin
- tier4_calibration_rviz_plugin
- tier4_camera_view_rviz_plugin
- tier4_control_rviz_plugin
- tier4_datetime_rviz_plugin
- tier4_debug_rviz_plugin
- tier4_debug_tools
- tier4_localization_rviz_plugin
- tier4_logging_level_configure_rviz_plugin
- tier4_perception_rviz_plugin
- tier4_planning_rviz_plugin
- tier4_screen_capture_rviz_plugin
- tier4_simulated_clock_rviz_plugin
- tier4_state_rviz_plugin
- tier4_system_rviz_plugin
- tier4_target_object_type_rviz_plugin
- tier4_traffic_light_rviz_plugin
- tier4_vehicle_rviz_plugin
- traffic_light_recognition_marker_publisher
- traffic_light_visualization

I judged mainly from the impression I get from the package names so there could be some packages that should be kept in autoware.universe. Even if the package is only used for debug purpose, I think we can still keep packages in Autoware.Universe if it is used frequently by most users. Also, moving rviz_plugin packages could be controversial so I'm would like to have any feedback.

↑ 2

7 replies

Show 2 previous replies

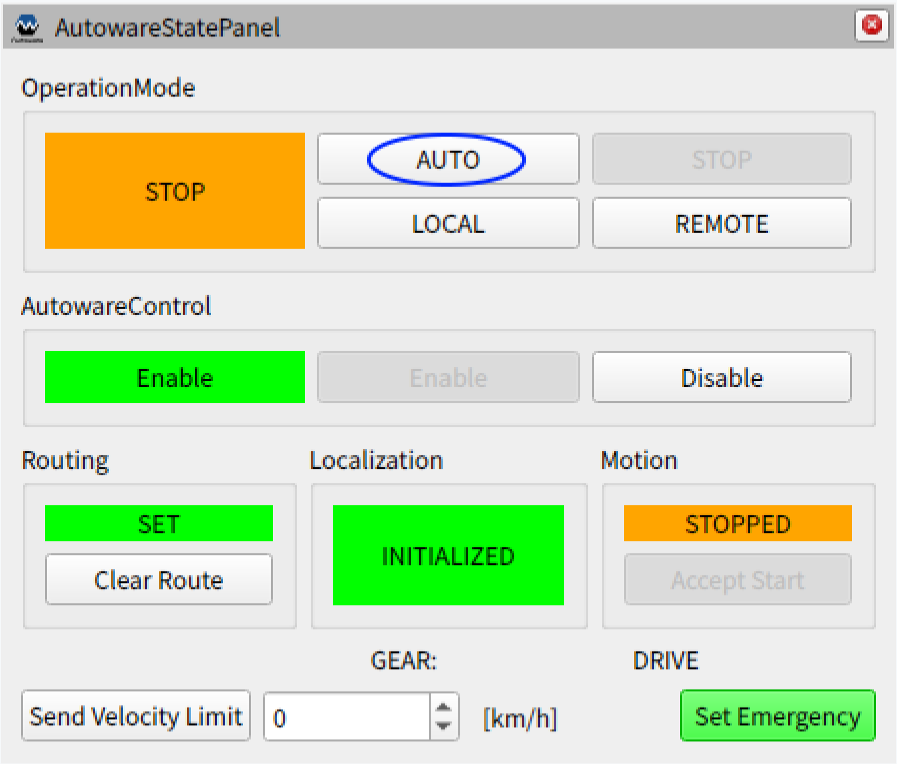
mitsudome-r on Jan 10 Maintainer Author

I just listed rviz_plugins blindly, but I think the one's that are used by default in planning_simulator tutorial and rosbag replay tutorial could stay in Autoware.Universe.

👍 1

isamu-takagi on Jan 10 Maintainer

The `tier4_state_rviz_plugin` (I plan to merge it with the `tier4_adapi_rviz_plugin`) provides the operation mode button. This is the only GUI that starts autonomous driving, so I recommend putting it in the default rviz config. If you are concerned about the inclusion of plugins that you do not use, I can also extract only the parts that are essential for autonomous driving to another package.



asa-naki on Jan 10 Collaborator

I think `diagnostic_converter` should also be moved to autoware_tools repository from autoware.universe.
I think this package is unused in planning_simulator tutorial and rosbag replay tutorial.

scepter914 on Jan 11 Collaborator

The package of `object_velocity_splitter` uses for perception in Autoware.

SakodaShintaro on Jan 25 Collaborator

@mitsudome-r
Sorry for noticing it late.
The following three packages are related to runtime diagnostics, so I would like to leave them in autoware.universe.

- localization_error_monitor
- pose_instability_detector
- steer_offset_estimator

doganulus on Jan 14 Collaborator

> Currently, we have many packages under Autoware.Universe which are not meant to be run at runtime and only used for debugging and evaluation. This makes the repository huge and also makes people who are not interested in such tools to wait for longer built time.

I agree with this statement and the proposed split of the Universe repo.

Further, I think this can be a good time to start discussing the concept of *Autoware package collections*, which are repositories containing Autoware packages and conforming to some standard rules and conventions. This generalizes the solution of splitting and allowing third parties to keep their Autoware packages in their own spaces or in separate repositories.

This first amounts to defining a standardized repository layout for such collections.

↑ 1

0 replies