ROS 2 Client Libraries Working Group

Important links

- Governance / Community
- Meeting recordings
- Regular meeting link
    - See ROS Events Calendar for actual invite to add to your calendar https://calendar.google.com/calendar/u/0/embed?src=agf3kajirket8khktupm9go748@group.calendar.google.com

[TEMPLATE] YYYY-MM-DD (UTC)

Agenda

- ***Start meeting recording***
- Attendance
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    -
- PR/issues discussion on managed repositories
    - https://github.com/ros2/rclcpp/pull/2382 Needs to be rebased
    - https://github.com/ros2/rclcpp/pull/2466 required for

Action Items

2024-05-03 (UTC)

Agenda

- Attendance
    -
- Discussions
    - Parameter service inconsistent behaviour
        - https://github.com/ros2/rclcpp/issues/2512
        - Michael Grey and I came to the conclusion that we need to change the behavior aligned with rcl_interfaces.
        - Any thoughts?
        - (btw, i may not be able to make it 8am cz of kids dropoff, probably 8:15-20, sorry)
    - QoS for ROS 2 actions status topic
        - https://github.com/ros2/rcl/issues/1155

        - This is related to https://github.com/ros2/rclcpp/pull/2499
    - Services and actions are not efficient when multiple clients are present https://github.com/ros2/rclcpp/issues/2397
    -

2024-03-29 (UTC)

Agenda

- Attendance
    -
- Need to get a new Google Meet link, since the one generated by Alberto has a hard-stop at 1h.
- We will try to meet every week (Friday 8AM Pacific Time) for the next few weeks since we have a lot of work before the Jazzy code freeze. We will revisit the schedule later.
- Follow ups
    - https://github.com/ros2/rclcpp/pull/2466
- Discussions
    - [Alberto] Thread configuration in rclcpp https://github.com/ros2/rclcpp/pull/2205
    - https://github.com/ros2/rclcpp/pull/2466

https://github.com/ros2/rclcpp/pull/2382/files

2024-03-22 (UTC)

Agenda

- Attendance
    - Alberto Soragna
    - Tomoya Fujita
    - Michael Carroll
    - Janosch Machowinski
    - Fitz
    - Tunna
- Need to get a new Google Meet link, since the one generated by Alberto has a hard-stop at 1h.
- We will try to meet every week (Friday 8AM Pacific Time) for the next few weeks since we have a lot of work before the Jazzy code freeze. We will revisit the schedule later.
- Discussions
    - Utilize rclcpp::WaitSet as part of the executors https://github.com/ros2/rclcpp/pull/2142

        - PR was blocked due to performance issues in the multi-threaded executor. A busy-wait got recently fixed, so we should revisit the performance.
        - The “collection” operation is still slow in the multi-threaded case because of the presence of multiple mutually exclusive callback groups.
        - Design question: should executors be aware of callback groups? If yes, we could have multiple rclcpp::waitset in the multi-threaded executor case and add APIs to implement set operations on them (union, intersection, etc). When “waiting for work” we need to merge the multiple waitsets into a single one.
    - New executors and “experimental” namespace
        - Workflow for the events-executor: design document -> separate repo outside of ros2 org -> merge into rclcpp the required API changes to support the separate repo -> merge events executor into an experimental namespace
        - We still don’t have clear requirements on what does it mean to be in the experimental namespace and how/when something should be moved out of it
        - Proposal for a callback centric executor based on waitset executors: https://github.com/ros2/rclcpp/pull/2405 . This is being rewritten and based on top of the events executor
        - There are some small API changes needed to support that. We should push them before the API code freeze (2 weeks from now)
    - BUG FIX: callback after cancel: https://github.com/ros2/rclcpp/pull/2281
        - The tutorials are wrong and were working only due to the presence of a bug. Fixing the bug may break the code for everyone that copy-pasted the tutorials.
        - The problem has to do with the scope of the actions goal handle.
        - With the PR in the current state, people with the wrong code would just get a (possibly silent) runtime bad behavior (callback not invoked).
        - Proposed to add [[nodiscard]] attribute to have a compile error
    - BUG FIX: race condition in action server: https://github.com/ros2/rclcpp/pull/2250
        - This PR depends on https://github.com/ros2/rclcpp/pull/2420
    - Added TimerInfo to timer callback: https://github.com/ros2/rclcpp/pull/2343
        - This PR depends on https://github.com/ros2/rcl/pull/1113
        - This should be easy to get reviewed and merged
    - Improved rcl_wait for timeout computation https://github.com/ros2/rcl/pull/1135
    - DESIGN DISCUSSION: expand ros idl to support nested arrays of custom type https://github.com/ros2/rosidl/pull/748
        - This PR will enable follow-up work to have better ROS 2 parameters
        - Getting this PR merged is not a priority, but the ROS community should converge about whether it’s a change we eventually want or not.

Action Items

2023-01-18 (UTC)

Agenda

- ***No recording captured***
- Attendance
    - Ralph Lange
    - Jacob Hassold
    - Steven! Ragnarök
- Last meeting action items review
    - Deferred
- Community PRs / Governance updates
    - Deferred
- Discussions (add topics here!)
    - Rust lifecycle implementation update from Jacob
        - It’s slow going and somewhat tedious
        - Ralph refers to rcl implementation of states and transitions
        - How are callbacks handled if rcl is handling transitions?
            - rclcpp is using a map, rclrs has a struct
            - in rcl callback info is only available at runtime
            - Trying to get feedback from community is a challenge due to lack of rust experience
            - recommends reviewing nav2 for callback examples
            - lifecycle callback examples from fmi_adapter: https://github.com/boschresearch/fmi_adapter/blob/rolling/fmi_adapter/src/fmi_adapter/FMIAdapterNode.cpp
    - Rust integration work from Open Robotics (REP-2013) on hold while the Infrastructure team helps facilitate separation between OSRC and OSRF infrastructure teams. No current ETA for when this will be picked up.
- PR/issues discussion on managed repositories

Action Items

2022-12-07 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)

    -
- PR/issues discussion on managed repositories
    - ROS 2 Intra-Process improvements https://github.com/ros2/rclcpp/pull/1847
    - ROS 2 monotonic node clock https://github.com/ros2/rclcpp/pull/1982
    -

Action Items

2022-11-09 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    -
- PR/issues discussion on managed repositories

Action Items

2022-10-12 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
- PR/issues discussion on managed repositories
    - https://github.com/ros2/rclcpp/pull/2024 is it necessary that lifecycle services are “owned” by the rcl layer rather than rclcpp ?

Action Items

2022-10-03 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
- Last meeting action items review
- Community PRs / Governance updates
  - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
  - [Geoff] Adding a Simple Action Client/Server
    - On roadmap for ROS I.
    - Nav2 already has them and we can adopt them into core libraries
    - OR will open PR to rclcpp, then we can review to see how to make them more generic
  - [Alberto] When should cleanup actually happen? There is an on_cleanup transition, but it is possible for a node to finalise without going via the transition.
    - [Geoff] It is difficult to have a universal clean-up path because in the case of error it is hard to know what state the node is in - what resources have been successfully acquired and which have not, for example.
  - [Alberto] It would be good to have a reason value to go along with transitions, e.g. an error code.
    - [Geoff] This is theoretically possible, but unclear how much work it would be to implement in rcl and rclcpp.
    - iRobot will try out some ideas and use their results to inform the replacement lifecycle work.
  - Integrating Rust into the buildfarm and supporting rcl-rust on the build farm
- PR/issues discussion on managed repositories

Action Items

2022-09-28 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
  - Geoffrey Biggs, Steven! Ragnarok (Open Robotics)
  - Jacob Hassold (GVSC)
  - Alberto Soragna (iRobot)
  - Tomoya Fujita (Sony)
- Last meeting action items review
- Community PRs / Governance updates

    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    - [Alberto] Intra-process actions: how to handle circular dependencies between intra-process manager and rclcpp_action library? Are actions really an “extension” of the core rather than part of the core?
- PR/issues discussion on managed repositories

Action Items

2022-09-14 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
    - Geoffrey Biggs (Open Robotics)
    - Esteve Fernandez
    - Jacob Hassold (GVSC)
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    - [Esteve and Jacob] What do we need to do to get rclrs into the core ROS 2 libraries?
        - Will make a Discourse post about it, then raise the topic at the TSC.
- PR/issues discussion on managed repositories

Action Items

2022-08-31 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
    - Geoffrey Biggs (Open Robotics)
    - Alberto Soragna (iRobot)
    - Jacob Hassold (GVSC)
    - Mauro Passerino
    - Tomoya Fujita (Sony)
- Last meeting action items review
- Community PRs / Governance updates

- Discussions (add topics here!)
    - Rclpy: missing API to get action names https://github.com/ros2/ros2cli/blob/rolling/ros2action/ros2action/api/__init__.py#L61
        - https://github.com/ros2/ros2cli/issues/748
        - Seems most appropriate for the rclpy client library, so Alberto will open a PR adding the function.
    - Best practice: passing node interfaces rather than node classes to objects constructor
        - https://github.com/ros2/geometry2/blob/rolling/tf2_ros/include/tf2_ros/transform_listener.h#L89-L104
        - Software engineering best practice would seem to indicate that interfaces should be preferred to passing object references.
        - Alberto will open a PR to TF proposing the change in constructors and follow up in the next meeting with what the response is.
    - How to handle new features in ROS 2 client libraries: user experience vs performance vs code maintenance tradeoffs
        - https://github.com/ros2/rclcpp/issues/1990
        - How do we promote adding new features without increasing the complexity of client libraries?
        - Should we have benchmarks available to allow us to track performance and ensure it does not decrease over time?
        - There are tools available that will measure things like cyclomatic complexity over a set of tests.
        - Will make thread on ROS discourse (Tomoya)
- PR/issues discussion on managed repositories

Action Items

- Tomoya to make a thread on Discourse discussing things like maintaining the complexity level of client libraries, measuring complexity, etc.

2022-07-20 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
    - Ralph Lange (Bosch)
    - Geoffrey Biggs (Open Robotics)
    - Behnood Farshbaf
    - Jacob Hassold (GVSC)
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls

- Discussions (add topics here!)
    - Iron Irwini development has started. Is anyone planning new client library features?
        - Improved executors for rclc. Currently have a single-threaded executor, and a prototype of a threadpool executor.
        - Maybe could do something to produce an abstracted executor, with the specific implementation of the scheduling being interchangeable.
        - Possibly having a 1.0 release of the Rust client library
    - [Geoff] Executor performance - problems, potential solutions, and other possible improvements
        - https://discourse.ros.org/t/reducing-ros-2-cpu-overhead-by-simplifying-the-ros-2-layers/13808/11
        - https://discourse.ros.org/t/singlethreadedexecutor-creates-a-high-cpu-overhead-in-ros-2/10077/16
        - https://www.youtube.com/watch?v=Sz-nllmtcc8&t=109s
- PR/issues discussion on managed repositories

Action Items

2022-07-06 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
    - Geoffrey Biggs (Open Robotics)
    - Jacob Hassold (GVSC)
    - Ralph Lange (Bosch)
    - Alberto Soragna (iRobot)
    - Nikolai Morin
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    - https://github.com/ros2/rclcpp/issues/1955 – Nikolai
    - Are loaned messages meant to be mutable or immutable? If you alter a loaned message in one subscription, will other subscriptions see the modified data?
        - This applies also to intra-process comm where subscriptions can use non-const signatures
        - A use case would be to have multiple subscriptions share the same data pointer (without copies) but then, once everyone else has read the data, one subscription could “take ownership” of it.

- Alberto: IMO this is a complex use-case which would add complexity to the client libraries. Moreover it may be not easy to determine how many people are viewing the data and when they are done with it.

- Why ROS 2 rclcpp uses a default context? Is this a requirement for the rust library?
  - Need to investigate. It should not be needed for backward compatibility with ROS 1 model, it’s likely it;s there for simplifying setting up an application.
- Rclpy
  - A proper replacement for rospy.sleep is required, rclpy.timer sleep is no replacement due to the "forever block" issue (https://docs.ros2.org/latest/api/rclpy/api/timers.html). Currently time.sleep has to be used even though that might cause issues when using sim_time.
    - File an issue about this so it can be tracked.
  - It would increase ROS2 Python development speed and reduce frustration if Python code changes would not always require rebuilds. Just like it was for ROS1.
    - Using the –symlink-install option to colcon should enable this.
  - Package directory structure (guidelines) are tailored to C++ and not really ideal for Python. E.g.
    - no separation of scripts and src folders for code that is directly called vs. includes/libs (as it was for ROS1)
      - The documentation should be improved for this.
    - the just placeholder filled "resource" directory.
      - The resource directory is necessary due to a shortcoming in Python that doesn’t allow the creation of files during installation, they have to be copied from somewhere. There is no workaround for this.
  - rospy's changes to the Python logger already were not ideal, but rclpy takes this to a whole new level. After a node is closed nothing can be printed anymore from the running Python script, so not just the logger gets manipulated, but whole stdout. This must not happen.
    - Should file an issue about this.
  - The logger can only log strings. I always find myself automatically writing self.get_logger().warn(str(...)), which is a bit annoying. String conversion should probably be done by the logger directly.
  - Lack of custom messages in purely rclpy packages.
    - Some work on running message generation without CMake had been done for Humble but it is not complete yet.
  - There exists no replacement for the ROS1 Python "rosbag" module (import rosbag; with rosbag.Bag...), which made rosbag handling very convenient. I totally understand that this is not a hard requirement

        anymore with the switch to sqlite. But it would help especially new ROS2 users a lot if there would be some inbuilt rosbag2pandas or rosbag2dict converter.
            - Topic for the Tooling WG, not Client Library WG.
            - The rosbag2 libraries feature Python APIs for loading, reading, and recording bags (as well as other functions); the ros2 bag cli is implemented in Python.
- PR/issues discussion on managed repositories

Action Items

2022-06-22 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
    - Ralph Lange - Bosch
    - Alberto Soragna, Mauro Passerino - iRobot
    - Geoffrey Biggs, William Woodall - Open Robotics
    - Jacob Hassold - GVSC
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    - [Geoff] Have started a REP on a new node life cycle. The goal is to get a prototype and REP ready for Iron Irwini.
        - Process for producing the REP shall initially be by making PRs against that fork, then discussing the proposed addition. We can revisit this process if it proves to be unwieldy.
    - [Geoff] tracetools library:
        - Although there is no explicit policy, it would be easier if it were at GitHub:
            - We could run the automated CI jobs against it
            - We could do releases when needed
        - If the maintainers are willing Chris Lalancette is in favour of moving it to GitHub in the ros2 organisation.
        - [Geoff] to follow up with the maintainers of tracetools about whether they would be willing to move it to github/ros2.
    - [Mauro] Purpose of ROS 2 action expiration timeout
        - Geoff: the server needs to keep the goal handle and it may keep around many of them, ready to give results to clients when requested.
          The timeout indicates when a client will not be allowed to request a result

        - anymore. This may be particularly helpful if the resulting data-structure is very heavy and it may be problematic to store many of them in memory.
        - Usage may be unclear, it would be good to have examples/docs to show how to use it
        - In rclcpp a server deletes the goal handle once the client requests the goal, but the handle is kept around until expiration in rcl. Is this a bug?
- PR/issues discussion on managed repositories
    - IPC clients and services: https://github.com/ros2/rclcpp/pull/1847
    - Rcl_guard_condition listener callback: https://github.com/ros2/rcl/pull/966
        - It may be better to not go forward with this. The main goal is to be notified when a ROS timer is paused/resumed (which is currently handled by a rcl guard condition). A better approach is to add explicit on_pause/resume listeners in the timer, keeping the guard condition as simple as possible

Action Items

- [Geoff] to follow up with the maintainers of tracetools about whether they would be willing to move it to github/ros2.

2022-06-08 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
    - Tomoya Fujita - Sony
    - William Woodall - Open Robotics
    - Jacob Hassold - GVSC
    - Behnood Farshbaf
    - Alberto Soragna - iRobot
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    - [Jacob] Problem with building rust client library: message objects are not “found” unless we explicitly clone them in the workspace. This feels wrong and results in colcon warnings about override.
        - [William]: this may be due to the fact that messages are installed via apt-get and they do not include the generated rust version of the messages. If this is the problem, adding messages to the colcon workspace is likely the correct approach and the warnings can be ignored.
        - This can be improved once the rust “IDL generator” is added to the standard ROS 2 packages/build-process. This requires the rust library to be mature enough and stable.

    - [Alberto] An action client can cancel a goal only after it has access to a goal handle. This means that a goal can’t be canceled before the server has at least processed the goal request.
        - [William] the goal ID is defined by the server, so the client needs to receive it first. This could be changed, but it’s a large change.
        - Moreover, action servers do not monitor the “liveliness” of a client. It would be nice if the server could check if the client is still alive to an API and handle this case (similarly to if a goal is canceled).
            - This seems like a missing point in the action design. It could be added and implemented as an optional feature.
    - General discussion on error handling:
        - Raising errors in ros callbacks could be try/catched by the executor to avoid bringing down the application. Wrap user callbacks into a function with try catch all. It’s unclear if we want this always turned on.
            - Add node options. This can be helpful for ROS component containers. For example catch exception, log something and unload the node that threw the exception.
        - There are situations where a remote application can send “bad requests” or “bad data” to a robot and cause an exception in rcl/rclcpp. This requires users to build in mechanisms to avoid this (e.g. try catch around the executor spin function).
            - These exceptions may be tricky to handle because users don’t know about them (as they are generated by the ROS libraries and not by the user code)
        - Using rust-like exceptions may improve the situation. This can be a very large change.
        - We don’t have clear documentation on the “list of exceptions” that can be thrown by the ros code.
- PR/issues discussion on managed repositories

2022-05-25 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    -
- PR/issues discussion on managed repositories

Notes

Hi, Geoff is not available today and the meeting isn't starting without him. -> Thanks!

Action Items

2022-05-18 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
  - Geoffrey Biggs, Shane Loretz (Open Robotics)
  - Alberto Soragna (iRobot)
  - Ralph Lange (Bosch)
  - Jacob Hassold (GVSC)
  - Nikolai Morin (rclrs client library)
- Last meeting action items review
  - [Geoff] Talk to Chris Lalancette about moving libstatistics_collector into the ros2 org (or not)
    - Talked to Chris. We came to the conclusion that while libstatistics_collector is a core dependency of rclcpp, it should not be. Instead, the functionality should be available in rcl so that all client libraries receive it. Will propose to Emerson Knapp that they make a PR providing this, at which point the existing libstatistics_collector library can be retired.
  - [Geoff] Talk to Steve about stealing the Nav2 simple action server
    - Nav2 is more than happy for us to take the simple action server.
    - Can we have this in rcl to share it with other client libraries?
      - Maybe doable, although actions rely on rclcpp implementation, but this looks like a C++ specific feature, other client libraries can develop their own "simple syntactic sugar"
      - Action server (and simple action server) use rcl logic for implementing the communication so we can have a C++ server communicate with a rust client (and the same for other client libraries)
- Community PRs / Governance updates
  - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
  - The tracetools library for rclcpp lives on GitLab, despite being a core dependency of rclcpp.
    - https://gitlab.com/ros-tracing/ros2_tracing/-/tree/master/tracetools

        - Geoff to follow up with Chris Lalancette about this.
        - There are mixed opinions on whether core dependencies should be in the ros2 org or not.
        - With rolling, the location of the release repository is more important (it should be in ros2-gpb)
    - Further discussion on roadmap items for I-Turtle
        - Revision of the life cycle for life cycle nodes.
            - We should aim to have an initial implementation in I-Turtle so the community has time to try it and provide feedback before we have to make a decision on starting a deprecation cycle for the existing life cycle.
        - Functional parity between rclpy and rclcpp
            - Ralph Lange to come up with a list of particularly important functions.
        - Remove callback groups from rclpy
            - They add a lot of complexity to the executor
            - They are not necessary in rclpy. Python coroutines can be used to yield to the executor instead.
    - [Alberto] Adding functionalities to ROS 2 parameters:
        - How to detect typos and unused parameters
        - Flag to "require override"
            - Similar to the "override" keyword in C++
        - Support for "NOT_SET" value in yaml file
            - This would give parity between C++ and the YAML.
        - General agreement that both of these would be useful.
        - Alberto Soragna to provide estimates of the feasibility of these two items.
    - [Nikolai]
        - What is the exact meaning of the thread safety comment in rcl?
            - Geoff: wrt to its arguments (or the "primary" argument)
        - Should new client libraries have the same functionality with parameters as rclcpp?
        - Why does rcl_publisher_fini/rcl_subscription_fini take a `rcl_node_t*` when only a `const rcl_node_t*` would be needed?
        - What are the portability goals of ROS 2? (e.g. 32-bit?)
- PR/issues discussion on managed repositories
    -

Notes

Action Items

- [Geoff] Find out what the policy is on where core dependencies of rclcpp should live. Re: tracetools library

- [Ralph] Provide a list of functionalities that are particularly important to have in rclpy that are not currently there but are in rclcpp
- [Alberto] Provide estimates of the feasibility and implementation workload for the proposed additions to parameters

2022-04-27 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
    - William Woodall (Open Robotics)
    - Alberto Soragna, Mauro Passerino (iRobot)
    - Tomoya Fujita (Sony)
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    - Time to start thinking about roadmap items for I-Turtle!
        - Move rclcpp statistics library to rcl
            - It currently depends on a C++ library
            - It needs access to message data
            - Engage with authors to understand challenges/scope of this work
        - Change verbosity level at runtime
            - Does this feature already exist? No, there-s a custom implementation for demos http://docs.ros.org/en/rolling/Tutorials/Logging-and-logger-configuration.html#logger-level-configuration-externally
            - Can use a parameter or a service
        - Advanced developer documentation
    - [Alberto] ROS 2 parameters discussion
        - When to use parameters vs service + topic? Why are they not used for lifecycle nodes?
            - Parameters were not available when lifecycle was implemented
            - Parameters come with overhead. This is a big problem as many ROS 2 design decisions are driven by limitation of the DDS
        - Add support for private configuration parameters?
            - Allow to have ROS 2 parameters that are not exposed by the parameters server
    - Explicit support for hidden topics/services/parameters
        - Print a warning if you subscribe to hidden topic; Print a warning if you try to read hidden parameter
        - This should be an opt-in feature. It could be a parameter, but same scalability limitations mentioned above. It could be an option in the node constructor.

- PR/issues discussion on managed repositories

Notes

Action Items

2022-04-13 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
  - Ralph Lange (Bosch)
  - Stefan Hoffmann (Schiller Automatisierungstechnik GmbH)
  - Jeremy Adams
  - Geoffrey Biggs, Shane Loretz (Open Robotics)
  - Nikolai Morin
- Last meeting action items review
- Community PRs / Governance updates
  - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
  - Triage
    - https://github.com/ros2/rclcpp/pull/1847
    - https://github.com/ros2/rclcpp/pull/1891
  -
- PR/issues discussion on managed repositories
  - https://answers.ros.org/question/398651/ros2-per-topic-intra-process-communications-setup/

Notes

Action Items

2022-03-24 (UTC)

Agenda

- ***Start meeting recording***
- Attendance

    - Geoffrey Biggs, William Woodall - Open Robotics
    - Jacob Hassold - DCS
    - Ralph Lange - Bosch
    - Tomoya Fujita - Sony
    - Alberto Soragna, Mauro Passerino - iRobot
    - Stefan Hoffmann - Schiller Automatisierungstechnik GmbH
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/community/pulls
- Discussions (add topics here!)
    - Governance information has been added. Please PR if you wish to see changes.
    - [Geoff] Moving the libstatistics_collector package to the ros2 org
        - Emerson Knapp has requested this due to it being a dependency of rclcpp and the tooling WG not wanting to maintain a core dependency
            - This package is currently in tooling because it was maintained by tooling WG
    - [Alberto] Discussion on the verbosity of ROS 2 client libraries APIs and its effects on code quality and application development.
        - There is a lot of boilerplate code required when using the rclcpp API, e.g. the amount of code required to use parameters.
        - Some of the verbosity comes from C++ being a verbose language, some comes from safety mechanisms (e.g. validating parameter input YAMLs), and some comes from the API for parameters not yet being polished as much as they could be.
        - Create a “this annoyed me” pain-points list that the WG can discuss.
        - We should pull the simple action server out of Nav2 and into its own package or into rclcpp.
    - Topic statistics appear to only be implemented in rclcpp. Is this correct?
        - If so, this is a candidate for pushing down to rcl, possibly as an librcl_statistics.
    - Triage for Humble
        - https://github.com/ros2/rclcpp/pull/1561
        - https://github.com/ros2/rclcpp/pull/1847
        - https://github.com/ros2/rclcpp/pull/1892
        - https://github.com/ros2/rclcpp/pull/1891
- PR/issues discussion on managed repositories
    - [Mauro] intra-process optimization for clients and services https://github.com/ros2/rclcpp/pull/1847
        - This is transparent to allow data to be sent to both intraprocess receivers and network receivers at the same time, but it does have the same double delivery problem if the underlying middleware doesn’t support it. Although this is less of a problem than for topics if services are used in the recommended way of one service provider for the service.
        - This PR needs to address the double delivery problem appropriately.

            - Currently if the client finds a server in the same process, the request will only go to that server and it won't go over the network even if other servers exist.
    - [Tomoya] ContentFilteredTopic API https://github.com/ros2/rclcpp/pull/1561 requesting review (https://github.com/ros2/rcl/pull/894)

Notes

Action Items

- [Geoff] Talk to Chris Lalancette about moving libstatistics_collector into the ros2 org (or not)
- [Geoff] Talk to Steve about stealing the Nav2 simple action server
- [Geoff] Start a pain points list somewhere, somehow

2022-03-02 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
    - Geoffrey Biggs, Shane Loretz, William Woodall (Open Robotics)
    - Alberto Soragna, Mauro Passerino (iRobot)
    - Ralph Lange (Bosch)
    - Jacob Hassold (DCS)
- Last meeting action items review
- Community PRs / Governance updates
    - https://github.com/ros2-client-libraries-wg/
- Discussions (add topics here!)
    - Kick-off of life cycle redesign
        - https://github.com/ros2/rclcpp/pull/1836
        - It should be possible to iterate on implementation without need to rewrite from scratch
        - We should design the behavior when an entity is disabled
            - Lifecycle design https://design.ros2.org/articles/node_lifecycle.html says that entities shouldn't receive execution time, which is vague. We could have configurations although it may complicate things.
            - Acting at the DDS level would be the best from performance point of view and it would give visibility to clients, but it may be difficult to implement and rely on features from the middleware implementations
            - For safety reasons, a client needs to know if the service is sleeping/dead (e.g. it should be notified)

      - Nodes already send a notification when they change lifecycle state. Using this probably require to have all nodes as lifecycle nodes
    - Clients should be able to realize when something is wrong in the server, but currently they only see it as a timeout. This would require changes in all the layers
      - Probably we can’t rely on this notification in all situations: for example if the server is not spinning or the executor is blocked by something else, the request will not be processed even if the entity is active
      - A disabled server could “ignore” request and then process them when it awakes, treating them as failed and notifying the client
    - As a system integrator it is helpful to know if a deactivated entity is receiving data while being inactive
    - Idea: have a small ack message sent from the server to the client for the various stages: request received, processing request, etc.
      - This is probably hard to implement and it would require work with the middleware vendors
  - Follow-up on read-only parameters
    - See original issue: https://github.com/ros2/rclcpp/issues/1762
    - Plugins can’t use read-only parameters without a way to undeclare them - if a node unloads a plugin any read-only parameters created by that plugin will be left behind.
    - Making read-only parameters mutable-by-some breaks the semantic that other nodes should know that if a parameter is immutable, they only need to read it once.
    - Perhaps we need to separate “immutable-to-all” and “read-only to external entities” concepts
      - Adding this would require API changes and changes at the ROS message level
  - [Alberto] Callback groups and single thread executors. How to signal that two callback groups can’t be used in the same executor?
  - [Jacob] Easier discussions/pooling resources between the Client Libraries WG and Rust WG (still working on building/linking after changes to `rcl` includes)
- PR/issues discussion on managed repositories
- Create notes template for next meeting

Notes

Action Items

2022-02-16 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
  - Geoffrey Biggs, Shane Loretz (Open Robotics)
  - Alberto Soragna
  - Ralph Lange (Bosch)
  - Alejandro Mosteo
  - Esteve Fernandez
  - Tomoya Fujita
  - Stefan Hoffmann
- Last meeting action items review
- Community PRs / Governance updates
  - https://github.com/ros-clientlibs-wg/community/pulls
  - https://docs.ros.org/en/rolling/Governance.html#client-libraries
- Discussions (add topics here!)
  - Invite Google group for getting meeting invitations is now available.
  - Working group repository has been set up.
  - Rust Client Library build issues
    - Best way to coordinate efforts and pool knowledge between working groups
  - Lifecycle design and currently open-issues
    - The current design and, in particular, the implementation of lifecycle nodes does not appear to be complete. E.g. There are no lifecycle services or lifecycle actions.
    - https://github.com/ros2/rclcpp/pull/1836 <- A proposed implementation for lifecycle services
    - If when a sub is inactive you buffer requests, there's the problem of how many you buffer and what you do when you wake up with all the potentially old data
    - Servers should return a "failure" notifying the client that they are inactive. How to do that?
However, even if a server is active, there are situations where it does not respond (e.g. not spinning in an executor).
    - The same applies to subscriptions. Even if a publisher is present in the system, it does not imply that it's producing data (it may be inactive) but we don't have any notification mechanism here.
    - It's a valid situation to have a server inactive and a client active. For example a remote machine trying to communicate with a server on an inactive robot
    - There is QoS to control how old data is (deadline?), not fully implemented for servers

        - Example: SLAM algorithm subscribes to data and starts building pose graph but it does not publish until warm-up completed. This probably requires application-level lifecycle management, out of scope for lifecycle ROS entities?.
        - The Lifecycle design and implementation in ROS 2 appears to be incomplete/inconsistent from a fundamental concept level, up to the implementation (missing services, etc.).
        - We have two options:
            - Continue with the current concept and figure out how to fit services and actions into it
            - Rethink the life cycle concept from the ground up, then provide an implementation that is an alternative to the existing lifecycle library. Eventually deprecate the existing one.
        - Error conditions in the life cycle state machine are also an issue. It’s unclear how the error state should be used.
        - Bosch and others have done some work on node management
            - https://github.com/micro-ROS/system_modes/tree/master/system_modes
            - https://github.com/ros-planning/navigation2/tree/main/nav2_lifecycle_manager
            - https://hexdocs.pm/elixir/Supervisor.html
        - Should an orchestrator that manages life cycle nodes be part of launch or a separate entity?
- PR/issues discussion on managed repositories
    - https://github.com/ros2/rclcpp/pull/1850
        - Should “read-only” mean read-only for others, or for the owner as well?
        - Should a node be able to undeclare a “read-only” parameter?
        - The original design for read-only parameters had an important use case that a parameter could be declared in a YAML file and then never modified by anything.
        - https://github.com/ros2/rclcpp/blob/025cd5ccc8202a52f7c7a3f037d8faf46f7dc3f3/rclcpp/include/rclcpp/node_interfaces/node_parameters_interface.hpp#L210-L217
- Create notes template for next meeting

Notes

Action Items

- (Geoff) Fix link to repository on the governance page
- (Geoff) Kick off a new design document on re-thinking the life cycle design

2022-02-02 (UTC)

Agenda

- ***Start meeting recording***
- Attendance
  - Geoffrey Biggs, Shane Loretz, William Woodall (Open Robotics)
  - Jacob Hassold (DCS)
  - Ralph Lange (Bosch)
  - Esteve Fernandez (ros2-rust, ros2-java, ros2-dotnet)
- Introduction to the working group
- Discussions (add topics here!)
  - Working group governance information
  - Regular meeting time
    - The current meeting time will be used for now (3pm UTC, every two weeks on Wednesday)
  - Best method for coordinating work between multiple client libraries
    - Currently, Rust client library is having difficulties building due to a change in rcl's install locations
    - Not sure of best location for documentation for creation of new client libraries
  - The ROS 2 documentation could do with more client library documentation, e.g. about executors and how to use them effectively, how to use lifecycle nodes, etc. Much of the information on design.ros2.org could be used as a base.
  - A Matrix chat may be desirable in the future once we have more members. For now, communication shall be via the meetings and Discourse.
  - https://github.com/ros-tooling/libstatistics_collector ownership and future
    - this is a direct dependency of rclcpp (only), and is not really very much code
    - I propose repo moves to ros2 org, or code gets absorbed into rclcpp directly (or rcpputils) (or rcl/rcutils for possibility of use by rclpy and other client libs)
    - *More information is needed on the reason for moving to the ros2 org. Is it a maintenance load problem?*
- PR/issues discussion on managed repositories
  - https://github.com/ros2/examples/pull/328
  - https://github.com/ros2/examples/pull/331

Notes

Action Items

- (Geoff) Follow up with Emerson Knapp regarding moving the libstatistics_collector package to the ros2 org.