Pyros.

Release 0.4.3

AlexV

May 06, 2018

Contents

This Python package and ROS package interface multiple multi-process systems.

For example you can have a ROS system on one side, with his different nodes, services, etc. and a python multiprocess system (celery, tornado, etc.) on the other side, with also his own nodes, services, etc.

Pyros allows them to communicate, while keeping each other isolated in their own process context.

If you want to know more about how Pyros does this, you can have a look at the *Overview*.

Contents:

CHAPTER 1

Pyros

- If you are a ROS developer and want to use python inside ROS, you can stop reading and go have a look there instead https://github.com/ros/ros_comm
- If you are a Python developer and are curious about how you can plug a robot in your existing System using python, then keep on reading.

Pyros is a multiprocess interface python package.

It is not meant to run by itself, instead it should be used as a client library on one side, connecting to a specific process (ROS node) it spawns inside your multiprocess system (ROS) on the other side, in order to interface with it.

ROS is the main application of pyros, however its abstract base class design aim at developing complete abstraction of such distributed systems and interface with other kinds of multiprocess systems.

This package ultimately doesn't need ROS to be able to run and perform basic selftests on it. When using the ros interface however, your ROS configuration will be dynamically detected by pyros-setup and setup under the hood.

Do not hesitate to get in touch by leaving an issue, if you think Pyros can help interfacing the multiprocess system you need to communicate with.

1.1 Repository structure

This repository has a few main branches:

- master : main branch, python dev workflow, releasing version tags into a pip package.
- indigo-devel : current indigo-based ongoing development. catkin dev workflow.
- indigo : current indigo-based release (ROS pkg - tags attempting to match)
- <ros_distro> : current <ros_distro>-based release (ROS pkg)

Apart from these we follow a feature branching workflow

WARNING: This repository structure is currently being implemented…

1.2 ROS usage

Pyros is a blanket under which all the ROS tricks can remain hidden, and not break your usual python workflows.

ROS needs to be installed on your machine for the rosinterface to work. All system & ROS dependencies need to match pyros expectations when launching Pyros from Python. Pyros currently cannot verify your current system and certify compatibility.

This package includes :

- a rosnode that will be launched in a separate process to maintain isolation.
- a client library to connect to this node and interface with the ROS system.

Standard python packages can then use pyros to access the ROS system via pyros pure python API. That means each package depending on pyros will create their own ROS node. It is intended so that the configuration (what is exposed or not) can be different for each one of them. Be careful in settings where we have multiple similar clients (like web server scaling up, celery multiprocess worker, etc.), only one pyros node is needed per client type…

1.2.1 Why Pyros ?

Historically Pyros spawned from rostful. With multiple multiprocess systems like celery and flask needing to communicate with ROS underneath, it became necessary to implement a generic python interface that could be used from any usual python software, the way a python developer would expect it.

rospy is an interface to python inside ROS, but it also enforces some behaviors that are specific to ROS :

- expect the whole ROS environment is already setup before start (`source workspace/devel/setup.bash`)
- initializing a node is mandatory to communicate on topics (but not services)
- only one node can be initialized once per process
- some global side effects can break things around.

These are not a problem to use python as a language inside a ROS environment, always using rospy. But they make interfacing a ROS system with existing python packages much more tricky than needed, and often the outcome is not quite the expected one :

- Most web servers spawn multiple processes to handle concurrent requests, and an interpreter is loaded in each process, which also has the WSGI application executed for each request. That means, if you use rospy directly you end up with as many nodes as there is web processes (at best). Exactly duplicated nodes which you don't need in your ROS environment. Also these can take time to start, become inconsistent after a while, etc. This is not what a developer working on a robot wants to happen.
- Custom multiprocess software (like celery) have their own expectation and complex setup regarding global state and how processes are managed by the application, between the import from multiple places, multiple launches, and the way memory is managed. Mixing this with the way rospy does things is just a recipe for disaster.

Pyros solves these problems by :

- dynamically loading ROS environment on python import (with pyros-setup).
- not relying on ROS to implement multiprocess communications.
- keeping each multiprocess system implementation isolated in their own process.

1.2.2 Supported distributions currently

- ROS : Indigo

This package can also be used from source as a normal ROS package. Add it to your ROS workspace as a usual repository and everything should work. If it doesn't work from source as you would expect it, please submit an issue.

1.2.3 Coming up soon

- [ ] a pip package is available to be installed from source (works via rosdep).
- [ ] an ubuntu (virtual env embedded) is provided to install pyros on your Ubuntu Trusty to use pyros from binary.

=> If you want to build a binary package depending on pyros, you should rely on the pip dependency mechanism and embed the appropriate version into your deb (virtual env embedded).

1.3 Roadmap

We should do what we can to catch up on IoT platforms design. Although Pyros focus on pure python design, another package depending on it should enable ROS systems to connect as device to these IoT platforms.

Currently Rostful provide HTTP connection to your robot. For example another project could provide MQTT connection to your robot, following amazon choice of using MQTT (OASIS Standard).

IoT platforms design :

- Amazon : http://docs.aws.amazon.com/iot/latest/developerguide/protocols.html

1.4 Will NOT do in pyros

- Support for Actions (http://wiki.ros.org/actionlib) in pyros itself.

Actions are built upon multiple ROS topics, and can be interfaced like that, by writing an action client on the other hand (pure python, javascript on top of rostful, etc.), with pyros just forwarding all the required topics.

If this statement happens to be not true, this can be reconsidered. Additionally when interfacing with other systems “outside” of ROS, services are probably what you want, maybe also topics sometimes, but probably not actions. And actions can probably be implemented with just only a couple of services. The problem here is likely the latency + communication rate that will, anyway, be hard to maintain while moving through layers to go outside of ROS.

- Support for Rocon (http://wiki.ros.org/rocon) in pyros itself.

Although early versions of pyros were supporting rocon features (Rapps, interactions, etc.), the maintenance effort required is too high right now, and anyway it is probably better to have a more solid pyros available to allow people to develop packages on top of pyros to match Rocon features from outside of ROS, if the need arises.

=> We probably will not support anything that is not part of the core of ROS. As long as implementation in another python package is possible using pyros.

CHAPTER 2

Overview

Pyros is connecting multiple processes together, from different "multiprocess environments" while assuming the strict minimum about these environments. The main focus is interfacing Python multiprocess environment to ROS multiprocess environment, but we should work towards solving any kind of multiprocess bridging.

A process is supposed to be one program, isolated from all others, so how can you write a program that interface two programs without using one of the underlying mutliprocess system ? It s a chicken and egg problem : to be able to have one set of processes talk to another set of processes, you need to have a multiprocess framework that enable interprocess communication.

Pyros solves that problem by providing his own multiprocess model, that strives to be minimal, yet complete enough to be able to solve any problem that is traditionally solved with imperative or functional paradigms. All you might want to be able to do with a multiprocess software should be doable with Pyros.

2.1 Standing on the shoulder of giants

We are leveraging existing technologies, and concepts :

- Multiprocess is done by python.multiprocessing . Eventually this can be extended to support other concurrency models, following a similar API :
  - Threads (with python.threading) (TO COME)
  - monothread (TO COME : Entities as understood from Entity Component Systems design)
- Communication is handled by pyzmq and can be done via multiple channels :
  - OS pipes
  - network sockets
  - shared memory (TO COME)
- Serialization is done by pickle, with planning to plug any other serialization library that can support the same pickle API:
  - jsonpickle (TO COME)

  - etc.

2.2 Assumptions

We are working based on the assumption that there is a minimal general-purpose distributed computing model, able to solve all traditionally solvable computational problems. While this model is not clearly define theoretically (if you know one, let me know !), a big part of the work is identifying which concepts are present in the many multiprocess framework existing and extract the lowest common denominator from them.

Functional programming can completely abstract the underlying realities of computers for the sake of nice algorithm. While it can be useful in some cases, most real world applications still need to care about how they run, and developers, as humans, naturally think in the "Subject Action Object" form. On a running system, the process is the subject, and therefore abstracting it will remove a concept that is useful for most developer while thinking about his application.

Currently our multiprocessing model is made of :

- Nodes : Think web servers, think one command line process reading from a unix pipe and writing into a file, think stream processing, think dataflow programming.
- Services : think REST API backend, think remote procedure call, think subroutine, synchronous or asynchronous from the client point of view.
- Params : these are here to be able to interface params from ROS. But ultimately there should be some concept of global state, a set of global variables.

While this is not strictly needed (functional languages dont have that), it is widely used, and not having it can make some simple task more complicated than needed. The current plan is to provide some resilient distributed global state by implementing a consensus algorithm like raft in multiprocessing python. Using it in an appliation is not mandatory, and not optimum, but it will be stable and usable from any node, without single point of failure.

- Something else… : There is a concept missing here, that is the complementary of a service.

Something that always broadcast data, where loss is acceptable, and that can reach fast transfer rate. ROS has Topics. But Topics are actually not really useful, since you always need to worry about the endpoints, the connection ends from publisher side or from subscriber side. We need a concept that can interface with topics, yet that doesnt require people to think about connection endpoints. This is probably the type of communication used for stream/dataflow programming, however most implementation do that inside one process and we need to do that between processes.

Note : This will evolve as the software, and our understanding of complex multiprocess systems evolves.

CHAPTER 3

Roadmap

Mid to long-term goals include :

- extract zmp package from Pyros. it is a generic multiprocess framework, and should only be used as a dependency.
- find or implement the raft algorithm in python, to enable distributed global state between multiple processes safely.
- interface with ROS 2
- identify and implement the dataflow/topic style concept.

CHAPTER 4

Pyros

4.1 baseinterface

4.2 mockinterface

4.3 rosinterface

CHAPTER 5

ZMP

CHAPTER 6

Changelog

6.1 0.4.3 (2018-04-18)

- Disabling integration tests in python for now. [AlexV]
- Removing ros interface as test requirement as there is no python package for it just yet. [AlexV]
- Adding ros interface to testing requirements on ROS. [AlexV]
- Moving tests outside of the package, to ease test dependency maintenance. [AlexV]
- Adding pyup config. [AlexV]
- Fixing tox config. [AlexV]
- Improving testing by adding requirements matching rosdistro packages versions. [AlexV]
- Fixing badge links. [AlexV]

6.2 0.4.2 (2018-04-17)

- V0.4.2. [AlexV]
- Fixing pyros_common exceptions. [AlexV]
- Removing QC badge. [AlexV]

6.3 0.4.1 (2017-05-08)

- V0.4.1. [alexv]
- Adding pyros.server package to setup.py. [alexv]
- Improving imports to rely on pyros_setup only for ros server test. [alexv]

6.4 0.4.0 (2017-05-08)

- V0.4.0. [alexv]
- Removing catkin build files (should be in release repo). [alexv]
- Package.xml -> 0.3.2. [alexv]
- Fixing setup and examples using upcoming version of dependencies. [AlexV]
- Added dynamic setup API on client. [AlexV]
- Fixing usage of deprecated logging.warn. [AlexV]
- Adding examples doc. [AlexV]
- Fixes package names for tests to pass locally. [AlexV]
- Moving context server in pyros main package from pyros-common. [AlexV]
- Fixing package dependency names for new namespaced interface ROS packages. [AlexV]
- Moving pyros.interfaces to pyros_interfaces in pyros_common repo. now using namespace pyros_interfaces as dependency. [AlexV]
- WIP restructuring config and readding ctx_server here… [alexv]
- Fixing tests with new structure. [alexv]
- Moved a lot of things to pyros-common. [alexv]
- Fixing tests. [alexv]
- WIP : main with click. [alexv]
- Extracting ros dependent part and restructure. [alexv]
- Fixing doc to build without catkin. [alexv]
- Fixing import of BaseInterface. [AlexV]
- Fixing interface inheritance on the way to splitting mock and ros… [alexv]
- Moving mock interface outside of ROS, to not depend on ROS for mocks. [alexv]
- Adding rospublish command. [alexv]
- Mutating into a pure python package. adding rosdevelop setup.py command. [alexv]

6.5 0.3.2 (2017-03-21)

- V0.3.2. [alexv]
- Changing api for catkin_pip 0.2. [AlexV]
- Now relying on catkin_pip >0.2. [alexv]

6.6 0.3.1 (2017-01-13)

- V0.3.1. [alexv]
- Disabling cache test on jade since rocon_python_comms not available yet. [alexv]

- Being less strict about mock depend. [alexv]
- Fixing optional dependency for travis tests. [alexv]
- Small comments for tests. [alexv]
- Skipping travis test of rpm & debian branches. [alexv]
- Removing old gone six submodule. [alexv]
- Changed dependency to tblib third party released package to allow build for any ROS platform. [alexv]
- Making python-mock a full dependency (used in package, for transitivity). commenting tblib, might not be needed. [alexv]

6.7 0.3.0 (2016-11-14)

- Fixed rospkg version. [alexv]
- V0.3.0. [alexv]
- Changing branch to master for all readme badges. [alexv]
- Small file rename to match new name from pub and sub. [alexv]
- Adding api subpackage to setup.py. [AlexV]
- Moving all ros api calls into subpackage to make patching easier. cleaned up imports. [AlexV]
- First version of rospy safe API. [AlexV]
- Fixing node interface after pub/sub mixup. simplified stringpub and string sub tests. [alexv]
- Fixing tests filenames. cleanup debug prints. [alexv]
- Fixing cross logic between pubs ans subs. [alexv]
- Improving subscriber / publisher inter interfacing… added timeout to connect a pub/sub. [alexv]
- Now properly inverting (publisher_if is subscriber and vice versa) [alexv]
- Fixing broken update loop of ros_interface. [alexv]
- Skipping failing rosinterface test for now. passing fine independently, more investigation needed… [alexv]
- Separating pubs and subs. needs pyros_test 0.0.6 for tests. [alexv]
- Disabling cache builds on kinetic for now. [AlexV]

  rocon_python_comms is not available on kinetic yet.
- Forcing install of debs for travis script. [alexv]
- Now running same tests for python flow or installed catkin build. [alexv]
- Improving travis build to test with cache as well… change version_eq to version_gte since buidfarm doesnt handle version_eq properly (yet?). [alexv]
- Importing contextmanager from contextlib instead of decorator. [alexv]
- Moving mockinterface into rosinterface.mock since design follows ROS concepts. fixed all tests. bumped pyros minor version to 0.3.0 because of all the changes… [alexv]
- Merged testRosInterfaceNoCache and testRosInterfaceCache. fixed all issues. [alexv]
- Fixed tests without cache. [alexv]

- Basic usecase now working again with cache. needs lots of cleanup… [alexv]
- Continuing changes in rosinterface, splitting service, topics and params interface pools now rosinterface tests all passing. [alexv]
- Splitting baseinterface to simplify things. fixed mockinterface and tests. [alexv]
- Various cleanups. [alexv]
- Improved profiling script. [alexv]
- Comments. [alexv]
- Fixing bwcompat issues. dropping shutdown behavior fix for now. [alexv]
- Improved management of interface topics and reference counting. still broken for multiprocess because shutdown is not working properly. [alexv]
- Fixing params and services removal with cache diff input. improved topics interface creation and cleanup. [alexv]
- Speeding up topic interfacing. [alexv]
- Fixed logic for removing transients on difference update. now forwarding exception if param type not found small test improvements. [alexv]
- Fixing param behavior in ros_interface and added unit tests. [alexv]
- Fixing hybrid usecase of devel catkin workspace without ROS setup. [alexv]
- Adding python-tblib as a ros dependency. [alexv]
- Now using ros-shadow-fixed for testing with latest dependencies. [AlexV]
- Making the travis_checks script switch to his own dir on startup. [alexv]
- Fixing envvars checks for travis. made travis_checks.bash script executable. [alexv]
- Fixing typos. [alexv]
- Now travis tests with docker and on kinetic. [alexv]
- Improving first dynamic ROS import to ros_interface. improved logging. some test clean up since we use python testing framework now. [alexv]
- Fixing rospkg version. fixing setup.py commands for release flow. [alexv]

6.8 0.2.0 (2016-09-01)

- V0.2.0. [alexv]
- Preparing release flow. cosmetics. [alexv]
- Now fails with explanation if ConnectionCacheProxy not available in rocon_python_comms. [alexv]
- Moving on with step by step rostesting and partial python testing, because of process conflicts. [alexv]
- Making travis nose tests more verbose. [alexv]
- Increased dependent version of pyros_setup. attempt fixing travis. [alexv]
- Changed config behavior. now using pyros-setup default config. getting rid of complex default+override behavior for import config. improved logger. improved setup.py commands. [alexv]
- Importing pyros_setup only when imports from ros_interface failed. [alexv]

- Created deprecated decorator as util in pyros until we find better solution. [alexv]
- Fixing dependency on pyzmp with strict version. removed useless env values for travis. [alexv]
- Improved main init to import dependencies from python or from ROS packages. fixed check for unicode strings. started implementing CATKIN_PIP_NO_DEPS for testing. reviewing dependencies version. [alexv]
- Improved travis test scripts from pyros-setup scripts. improved setup.py with publish method fixed python3 issues on pyros_client. [alexv]
- Moved some dependencies out of pyros_setup, to not require pyros_setup if using ROS environment as usual. [alexv]
- Describing improved repository structure. [alexv]
- Improving release script. [AlexV]

6.9 0.0.9 (2016-08-25)

- Disabled pyrosROS test hanging on jenkins sometimes. [alexv]
- Releasing 0.0.9 for gopher benevolent. [alexv]
- Removing old gone six submodule. [alexv]

6.10 0.1.0 (2016-07-08)

- Regenerating full changelog. [AlexV]
- Rosdep dependency is likely redundant with pypi package. [AlexV]
- Fixing python mock version to be compatibel with trusty. [AlexV]
- Revert “dropping installspace build. no ros-indigo deb package will be created. requirements are too high for trusty : six >= 1.9” [AlexV]

  This reverts commit 64a0688e6706424c3c9a3742f776fcb73e833fff.
- Revert “downloading six >=1.9 for tests, ignoring system version” [AlexV]

  This reverts commit 946bf8df10ae50fcef8b77114521fcb861b31a56.
- Dropping installspace build. no ros-indigo deb package will be created. requirements are too high for trusty : six >= 1.9. [AlexV]
- Downloading six >=1.9 for tests, ignoring system version. [AlexV]
- Adding pypi mock dependency. [AlexV]
- Generating changelog in preparation for version 0.1.0. [alexv]
- Reducing ros python dependencies since we now rely on catkin_pip. [alexv]
- Cleanup debug log. [alexv]
- Moved debug logging to special logdebug file to reduce terminal logspam. [alexv]
- Fixing tests. [alexv]
- Disabling some test to prevent catkin test hanging… but test pass when run without –with-xunit. probably a nose issue. [alexv]

- Fix adding available services. quick fix on early topics detection to avoid dropping topic interface just after creation. now comparing local topic connection counter with global topic connection counter instead of always assuming 1. improved logging. [alexv]
- Fixed checking for available transients. now doesnt have to be a dict, just an iterable. [alexv]
- Now storing endpoints for topics in order to accurately detect lost topics when we get only endpoints diff from cache. WIP. some tests breaking now. [alexv]
- WIP. attempting to fix diff behavior with cache in corner cases when things changing fast on the system. [alexv]
- Changing static method used from class to class method used from self. [alexv]
- Now using diff optimisation in connection_cache. [alexv]
- Getting pyzmp 0.0.11 via dependencies to hopefully fix travis. not using requirements any longer since we dont have extra dependencies and catkin_pip_setup does install the package in dev mode. [alexv]
- Fixing node behaviors with recent pyzmp. [alexv]
- Reviewing how we use zmp nodes and improving tests… WIP. [alexv]
- Fix adding available services. improved logging. [alexv]

  Conflicts: pyros/baseinterface/baseinterface.py pyros/rosinterface/ros_interface.py
- Fixed checking for available transients. now doesnt have to be a dict, just an iterable. [alexv]

  Conflicts: pyros/rosinterface/ros_interface.py
- Next TODO. first step to simplification. [alexv]
- Removed useless None in get(smthg, None) [alexv]
- Added interface cache tests to run by default. reverted debug long timeouts. [alexv]
- Finished manual merging of connection_cache_diff_callback. fixed all RosInterfaceCache tests, but code really need refactoring… [alexv]
- More changes from connection_cache_diff_callback branch. only ros_interface.py changes are left todo. [alexv]
- Starting manual merge of connection_cache_diff_callback branch. [alexv]
- Fixes for connection cache with diff optimisation. added pubsub wait for confirm from cache, but deleted pubsub report deleted before confirmation from cache. Not sure if it is the right choice, but extra care is needed when deleting… [alexv]
- Fix tests for RosInterface especially with cache (but no diff optim) [alexv]
- Adding yujin underlay as we need it for connectioncache message format. [alexv]
- Fixing path to current workspace. [alexv]
- Renaming catkin_pure_python to catkin_pip. [alexv]
- Updating for catkin_pure_python 0.1.0. [AlexV]
- Fixing various minor python issues. [AlexV]
- Fixed service and topic type introspection. [alexv]
- Fixing definitions to match new topic class structure. [alexv]
- Fixing rostest call of testService.py. [alexv]
- Locking version numbers for pyros-setup and pyros-test dependencies. [alexv]
- Todo comments. py3 compat. [alexv]

- Removed duplicated import. [AlexV]
- Not installing pyros-setup from ROS package. pyros-setup should be useful only if run without ROS (directly from pip). [AlexV]
- Fixing self tests. now using pyros_setup pip package. [alexv]
- Adding nosemain for self test. [alexv]
- Now using pyzmp package dependency instead of internal zmp sources. removed submodules. [alexv]
- Now travis check python and ros workflows. [AlexV]
- Moving to package v2. [alexv]
- Replacing obsolete navi/semantic_locations by new /rocon/semantics/locations. [alexv]
- Moved pyros and zmp sources, otherwise pyros was not find through egg link. [alexv]
- Added version. fixed tests in cmakelists. added default config file, removed useless testing config. added entry point for selftests. added requirements devel dependency to pyros-setup. [alexv]
- Cleaning up rosinterface __init__. now doing ros setup only in child node process, dynamically. parent process is isolated. [alexv]
- Cleaning up imports and fixing tests. [alexv]
- Refactored to add configuration at module, package and user levels. implified pyros-setup configuration from rosinterface. reviewed separation between node and interface to isolate all ros setup in child process. now doing ROS message conversion internally in rosinterface service and topic classes. fixed most tests. now uses six to improve python3 compatibility. [alexv]
- Starting to adapt to new configuration from pyros-setup. [alexv]
- Now using catkin_pure_python. [alexv]
- Add Gitter badge. [The Gitter Badger]
- Cosmetics, comments and small fixes… [alexv]
- Readme regarding IoT. [alexv]
- Cosmetics. [alexv]
- Changing reinit method to a setup service. now reinitialize rosinterface everytime the list of services or topic passed by the user changes. refactor the base interface to hold local copy of system state. fix all tests. [alexv]
- Added missing rosservice dependency. [alexv]
- Fixing package dependencies for catkin. [alexv]
- Fixing catkin build. [alexv]
- Removing unused ROS service specifications. [alexv]
- Improved exception handling. adding mock client to make unittests easy. cosmetics. [alexv]
- Improved Readme. [AlexV]
- Removing dynamic_reconfigure. [alexv]
- Removed rocon feature. cleanup. [alexv]
- Exposing servicecall timeout exception. cosmetics. [alexv]
- Warn -> info when it's not meant to be alarming to the users. [Daniel Stonier]
- Fixing log warn -> info for startup args. [alexv]

- Fixme comments. [alexv]
- Adding simple test to assert rospy potentially strange behaviors. separating cache and non cache tests. catching connection_cache proxy init timeout, showing error and disabling. [alexv]
- Adding custom manager argument in basenode, and making shutdown possible override more obvious. [alexv]
- ZMP : services and node advertisement now done in context managers. Node now support using custom context manager when starting in another process. cosmetics. [alexv]
- Improving base support to pass diff instead of query full state everytime. now with callback called from connection cache proxy to only process list if change happens. [alexv]
- Fixing reinit to be delayed if ros interface not ready yet. [alexv]
- Fixing pyrosROS test with latest pyros_test. [alexv]
- Adding pyrosRos test to catkin tests. [alexv]
- Reiniting connection cache if dynamic_reconfigure disable/enable it. [alexv]
- Using enable_cache in dynamic_reconfigure to be able to dynamically switch if needed. [alexv]
- Fixed populating empty message instance. comments. [alexv]
- Adding missing rosnode as test dependency. [AlexV]
- Disabling roconinterface dynamic import. [AlexV]
- Moving more nodes to pyros-test. [AlexV]
- Moving nodes to pyros-test. skipping tests if connection_cache not found. [AlexV]
- Better error message if tests are run from python without pyros-test installed in ROS env. [AlexV]
- Using pyros_cfg and fix import in rocont interface, to run nosetests from python venv. [AlexV]
- Added generated code for dynamic_reconfigure. [AlexV]
- Adding requirements, fixing setup.py for setuptools. [AlexV]
- Now allowing to delay the import of rosinterface subpackage and passing base_path to find ROS environment dynamically. [alexv]
- Using ros-shadow-fixed for travis. [AlexV]
- Cleaning up comments. [alexv]
- Adding option to enable cache or not from rosparams. [alexv]
- Ros_interface now using topics and service types from cacche if available, otherwise query one by one when needed. making sure cache process is started and stopped during the test to avoid scary harmless warnings. [alexv]
- Improving tests. [alexv]
- Using silent fallback for connectioncache proxy. [alexv]
- Fixing dependencies in package.xml. [alexv]
- Pyros now dependein on pyros_setup and pyros_test for tests. [alexv]
- Pyros now depending on pyros_setup. [alexv]
- Expose_transients_regex now relying on _transient_change_detect directly. small refactor to allow transient updates only with ROS system state differences. fixing mockinterface to call reinit only after setting up mock Added first connection_cache subscriber implementation to avoid pinging the master too often. WIP. [alexv]

6.11 0.0.8 (2016-01-25)

- Doing zmp tests one by one to workaround nose hanging bug with option –with-xunit. [alexv]
- Making service and param new style classes. [alexv]
- Fixing throttling to reinitialize last_update in basenode. [alexv]
- Fixing a few quantifiedcode issues… [alexv]
- ZMP node now passing timedelta to update. Pyros nodes now have a throttled_update method to control when heavy computation will be executed (potentially not every update) [alexv]
- Displaying name of ROS node in log when starting up. [alexv]
- Mentioning dropping actions support in changelog. [alexv]
- Overhauled documentation. [alexv]
- Cosmetics. [alexv]
- Exposing pyros service exceptions for import. [alexv]
- Adding node with mute publisher for tests. [alexv]
- Fixing basic test nodes return message type. cosmetics. [alexv]
- Reviewing README. [alexv]
- Changelog for 0.1.0. cosmetics. [alexv]
- Migrated % string formating. [Cody]
- Fixing badges after rename. [alexv]
- Avoid mutable default arguments. [Cody]
- Made namedtuple fields optional like for protobuf protocol. [alexv]
- Fixing zmp tests with namedtuple protocol. [alexv]
- Fixing catkin cmakelists after test rename. [alexv]
- Making client exceptions also PyrosExceptions. [alexv]
- Begining of implementation of slowservice node for test. not included in tests yet. [alexv]
- Removed useless hack in travis cmds, fixed typo. [alexv]
- Trying quick hack to fix travis build. [alexv]
- Adding status message when creating linksto access catkin generated python modules. [alexv]
- Adding zmp tests to catkin cmakelists. [alexv]
- Added dummy file to fix catkin install. [alexv]
- Small install and deps fixes. [alexv]
- Simplifying traceback response code in node. [alexv]
- Fixing unusable traceback usecase in zmp. [alexv]
- Cosmetics. adding basemsg unused yet. [alexv]
- Moving exception to base package, as they should be usable by the client of this package. [alexv]
- Making pyros exceptions pickleable. minor fixes to ensure exception propagation. [alexv]
- Comments. [alexv]

- Ros_setup now use of install workspace optional. fixes problems running nodes (which needs message types) from nosetests. [alexv]
- Added cleanup methods for transients. it comes in handy sometime (for ROS topics for example). [alexv]
- Pretty print dynamic reconfigure request. [alexv]
- Cleanup debug logging. [alexv]
- Adding logic on name was not a good idea. breaks underlying systems relaying on node name like params for ROS. [alexv]
- Removing name from argv, catching keyboard interrupt from pyros ros node. cosmetics. [alexv]
- Increasing default timeouts for listing services call form pyros client. [alexv]
- Fixed multiprocess mutli pyros conflict issues with topics with well known rosparam. now enforcing first part of node name. cosmetics. [alexv]
- Removed useless logging. [alexv]
- Adding basetopic and fixed topic detection in rosinterface. zmp service now excepting on timeout. [alexv]
- Fixed exceptions handling and transfer. fixed serialization of services and topic classes for ROSinterface. [alexv]
- Now reraise when transient type resolving or transient instance building fails. added reinit methods to list of node service to be able to change configuration without restarting the node (usecase : dynamic reconfigure) added option to PyrosROS node to start without dynamic reconfigure (useful for tests and explicit reinit) added some PyrosROS tests to check dynamic exposing of topics. cleaned up old rostful definitions. cosmetics. [alexv]
- Cleaning up old action-related code. fixed mores tests. [alexv]
- Fixing how to get topics and services list. commented some useless services (interactions, ationcs, etc.). [alexv]
- Changing version number to 0.1.0. preparing for minor release. [alexv]
- Refactoring ros emulated setup. [alexv]
- Improving and fixing rosinterface tests. still too many failures with rostest. [alexv]
- Fixing tests for Pyros client, and fixed Pyros client discovery logic. cosmetics. [alexv]
- Making RosInterface a child of BaseInterface and getting all Topic and test services to pass. cosmetics. [alexv]
- Improved test structure for rostest and nose to collaborate… [alexv]
- WIP. reorganising tests, moved inside package, nose import makes it easy. still having problems with rostest. [alexv]
- Fixing testTopic for rostest and nose. cosmetics. [alexv]
- Finishing python package rename. [alexv]
- Separated rospy / py trick from test. [alexv]
- Fixing testRosInterface rostest to be runnable from python directly, and debuggable in IDE, by emulating ROS setup in testfile. [alexv]
- Implemented functional API, abstract base interface class, mockinterface tests. [alexv]
- Moving and fixing tests. [alexv]
- Changing ros package name after repository rename. [alexv]
- Fixing setup.py for recent catkin. [alexv]
- Protecting rospy from unicode args list. [alexv]

- Implemented transferring exception information via protobuf msg. readding tblib as dependency required for trusty. [alexv]
- WIP. starting to change message to be able to just not send the traceback if tblib not found. [alexv]
- Restructuring code and fixing all tests to run with new zmp-based implementation. [alexv]
- Now able to use bound methods as services. [alexv]
- Adding python-tblib as catkin dependency. [alexv]
- Useful todo comments. [alexv]
- Now using pickle is enough for serialization. getting rid of extra dill and funcsig dependencies. [alexv]
- Not transmitting function signature anymore. not needed for python style function matching. [alexv]
- Added cloudpickle in possible serializer comments. [alexv]
- Now forwarding all exceptions in service call on node fixed all zmp tests. [alexv]
- Fixing all zmp tests since we changed request into args and kwargs. [alexv]
- Starting to use dill for serializing functions and params. [alexv]
- Adding comments with more serialization lib candidates… [alexv]
- WIP. looking for a way to enforce arguments type when calling a service, and parsing properly when returning an error upon exception. [alexv]
- Getting message to work for both protobuf and pickle. Now we need to choose between tblib and dill for exception serialization. [alexv]
- Adding dill as dependency. [alexv]
- Multiprocess simple framework as separate zmp package. [alexv]
- Comments. [alexv]
- Transferring exceptions between processes. [alexv]
- Fixing all service tests and deadlock gone. [alexv]
- Improved service and node tests. still deadlock sometimes… [alexv]
- Multiprocess service testing okay for discover. [alexv]
- WIP. starting to use zmq for messaging. simpler than other alternatives. [alexv]
- WIP implementing service. [alexv]
- WIP adding mockframework a multiprocess communication framework. [alexv]
- Adding mockparam. [alexv]
- Adding code health badge. [alexv]
- Adding requirements badge. [alexv]
- Adding code quality badge. [alexv]
- Adding echo tests for mocktopic and mockservice. [alexv]
- Renaming populate / extract commands. [alexv]
- Setting up custom message type and tests for mock interface. [alexv]
- Fixing mockmessage and test. [alexv]
- Improving mockmessage and tests. [alexv]

- Started to build a mock interface, using python types as messages. This should help more accurate testing with mock. [alexv]
- Adding six submodule. tblib might need it. otherwise it might come in useful anyway. [alexv]
- Adding tblib to be able to transfer exception between processes. [alexv]
- Fixing travis badge. [alexv]
- Adding travis badge. [alexv]
- Starting travis integration for autotest. [alexv]
- Adding rostopic as a test_depend. [alexv]
- Fixes to make this node work again with rostful cosmetics and cleanups. [alexv]
- First implementation to expose params to python the same way as we do for topics and services. [alexv]

6.12 0.0.7 (2015-10-12)

- 0.0.7. [alexv]
- Adding log to show rostful node process finishing. [alexv]
- Change message content check to accept empty dicts. [Michal Staniaszek]
- Fixing corner cases when passing None as message content. invalid and should not work. [alexv]
- Fixing tests. and changed api a little. [alexv]
- Removing useless fancy checks to force disabling rocon when set to false. updated rapp_watcher not working anymore. [AlexV]
- Rocon_std_msgs changed from PlatformInfo.uri to MasterInfo.rocon_uri. [AlexV]
- Send empty dicts instead of none from client. [Michal Staniaszek]
- Service and topic exceptions caught and messages displayed. [Michal Staniaszek]
- Fleshed out topic and service info tuples. [Michal Staniaszek]
- Can check for rocon interface, get interactions. [Michal Staniaszek]
- Listing functions for client, corresponding mock and node functions. [Michal Staniaszek]
- Now passing stop_event as an argument to the spinner. cosmetics. [alexv]
- Fix when running actual rostfulnode. [alexv]
- Now running rostful_node in an separate process to avoid problems because of rospy.init_node tricks. [alexv]
- Cosmetics. [alexv]
- Improving how to launch rostest test. fixed hanging nosetest. hooking up new test to catkin. [alexv]
- Force-delete for services, test for removal crash on expose. [Michal Staniaszek]

  Test service nodes added
- Fix crash when reconfigure removes topics, started on unit tests. [Michal Staniaszek]
- Fixing removing from dictionary topic_args. [alexv]
- Stopped removal of slashes from front of topics. [Michal Staniaszek]
- Fixed regex and add/remove issues with topics and services. [Michal Staniaszek]

- Fixed topic deletion, multiple calls to add. [Michal Staniaszek]

  The interface now tracks how many calls have been made to the add function and ensures that topics are not prematurely deleted from the list. Actions also have a similar thing going on, but not sure if it works since they are unused. Services are unchanged.

  Ensured uniqueness of topics and services being passed into the system using sets.

  Removed unnecessary ws_name code.

  Issue #27.

- Fix *_waiting list usage, service loss no longer permanent. [Michal Staniaszek]

  The lists *_waiting now contain topics, services or actions which we are expecting, but do not currently exist. Once it comes into existence, we remove it from this list.

  When services disconnect, their loss is no longer permanent. This had to do with the services being removed and not added to the waiting list.

  Fixes issue #21.

- Full regex, fixed reconfigure crash. [Michal Staniaszek]

  Can now use full regex in topic or service strings to match incoming strings.

  Fixed crash when dynamic reconfigure receives an invalid string

- Strings with no match characters don't add unwanted topics. [Michal Staniaszek]

  Regex fixed with beginning and end of line expected, previously would allow a match anywhere in the string.

  Issue #17.

- Removed separate lists for match strings. [Michal Staniaszek]
- Remove printing, unnecessary adding to _args arrays. [Michal Staniaszek]
- Adding wildcard * for exposing topics or services. [Michal Staniaszek]

  Implementation should be such that other match characters can be easily added if necessary.

  Fixes issue #17.

- Added TODO. [alexv]
- Added exception catching for when rocon interface is not available. [Michal Staniaszek]
- Added important technical TODO. [alexv]
- Fixing bad merge. [alexv]
- Fixing unitests after merge. [AlexV]
- Quick fix to keep disappeared topics around, waiting, in case they come back up… [alexv]
- Turning off consume/noloss behavior. should not be the default. should be in parameter another way to expose topics. [AlexV]
- Allowing to call a service without any request. same as empty request. [AlexV]
- Keeping topics alive even after they disappear, until all messages have been read… WIP. [AlexV]
- Preparing for release 0.0.6. setup also possible without catkin. [AlexV]
- Changing rostful node design to match mock design. [AlexV]
- Fixing RostfulCtx with new Mock design. added unittest file. [AlexV]
- Improved interface of rostful client. added unit tests for rostfulClient. [AlexV]

- Improved interface of rostful mock, now async_spin return the pipe connection. added more unit tests for rostful mock. [AlexV]
- Added rostful mock object (useful if no ROS found). improved structure and added small unit test. [AlexV]
- Changing cfg file name to fix install. [AlexV]
- Comments TODO to remember to fix hack. [AlexV]
- Tentative fix of cfg… comments. [AlexV]
- Adding python futures as dependency. [AlexV]
- Commenting out icon image. no cache home on robot. need to find a new strategy. [AlexV]
- Removed useless broken services. [AlexV]
- Fixing catkin_make install with dynamic reconfigure. [AlexV]
- Adding bloom release in release process to sync with pypi release. [AlexV]
- Fixes for release and cosmetics. [AlexV]
- Preparing pypi release. [AlexV]
- Improving rostful node API. Adding rostful pipe client and python pipe protocol. removed redundant ros services. [AlexV]
- Simplifying rapp start and stop by using rapp_watcher methods. [AlexV]
- Now starting and stopping rapp. still ugly. [AlexV]
- Fixes to get rocon features to work again. [AlexV]

6.13 0.0.3 (2015-07-01)

- Preparing pypi release. small fix. [AlexV]
- Adding helper services to access Rosful node from a different process. Hacky, working around a limitation of rospy (cannot publish on a topic created in a different process for some reason…). Proper design would be to call directly the python method (work with services - node_init not needed) [AlexV]
- Small cleanup. [AlexV]
- Adding context manager for rospy.init_node and rospy.signal_shutdown. No ROS signal handlers anymore. Cleanup properly done when program interrupted. [AlexV]
- Playing with signal handlers… [AlexV]
- Improved test. but topic interface not symmetric. needs to deeply test message conversion. [AlexV]
- Small fixes and first working test to plug on existing topic. [AlexV]
- Adding first copy from rostful. splitting repo in 2. [AlexV]
- Initial commit. [AlexV]

CHAPTER 7

Indices and tables

- genindex
- modindex