Note: First create your own ROS repository using the RecommendedRepositoryUsage (/RecommendedRepositoryUsage) page. This tutorial assumes that you have completed the previous tutorials: Once you've created your own ROS package repository, you can document your stacks and packages on ROS.org..

💡 Please ask about problems and questions regarding this tutorial on 🌐 answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Indexing Your ROS Repository for Documentation Generation

Description: There are hundreds of ROS repositories hosted by companies, universities, and individuals around the world. Whether you're releasing one ROS package or hundreds, feel free to start your own repository to share your code with the rest of the community.

Tutorial Level: INTERMEDIATE

This is a tutorial for indexing for documentation only. To make a release into binary builds see the bloom (/bloom) documentation.

Indexing Your ROS package Repository for Documentation Generation

Once you've created your own ROS package repository, you can document your stacks and packages on ROS.org.

1. What you get by adding your package to the indexer

(this image is editable 🌐 here (https://docs.google.com/drawings/d/1aa8tFOi0L-v_dRYQlvrNc7w1SBFeosHQJRzHl3xoZ94/edit?usp=sharing).)

- Your package's wiki page will automatically pull in information from `package.xml` in your package (see the image above).
- (A very good) Side effect; Pulled information helps internet search engines indexing your page, leads to better visibility.
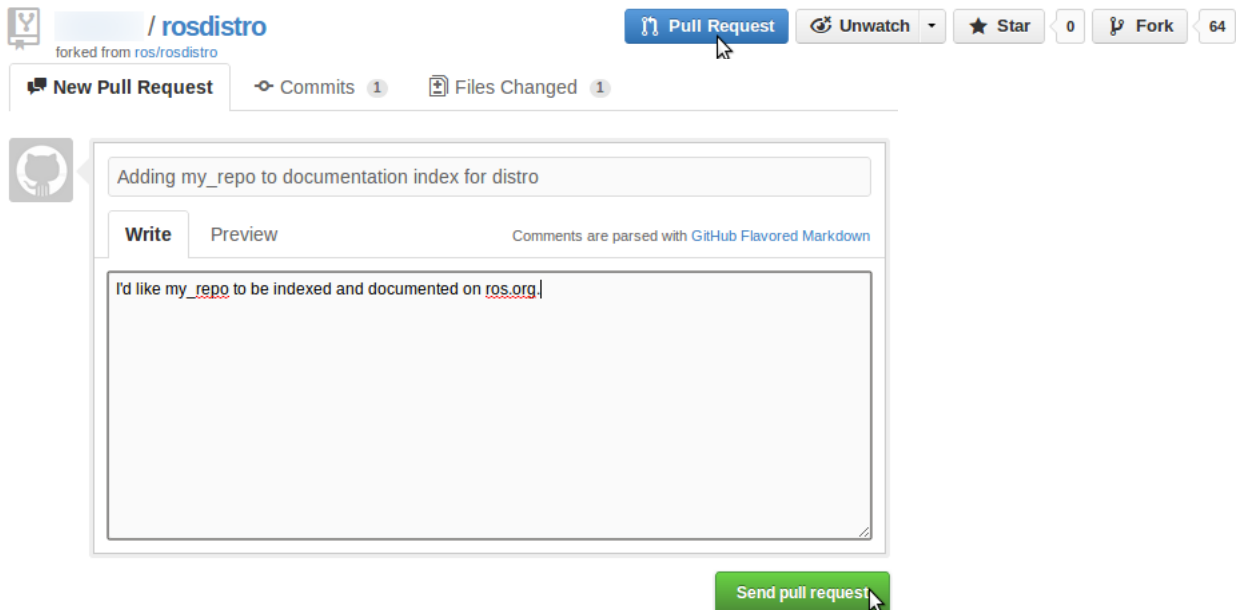
2. First time for a package to be added to the indexer

2.1 Steps to have your repository added to our indexer

- First, fork the following repository on GitHub 🌐https://github.com/ros/rosdistro (https://github.com/ros/rosdistro)

    - More information on forking GitHub repositories: 🌐https://help.github.com/articles/fork-a-repo (https://help.github.com/articles/fork-a-repo)
- Checkout your forked copy of the repository for editing
    - `git clone git@github.com:my_github_name/rosdistro.git`
- In your copy of the rosdistro repository browse to the distribution directory you'd like to update.
    - `cd rosdistro/<DISTRO_NAME>`
- Inside of this directory, edit the distribution.yaml file and add your repository information (while maintaining alphabetic order of the repositories) with a `doc` section.
    - The repository url should point to the upstream repository, not the release repository.
    - Optionally, you repository entry can contain a `depends` value which contains a list of other repository names listed in this file upon which your code (and it's documentation) depend, but which you do not want to document explicitly in the job associated with your repository.
- Run the following script in the root of your rosdistro checkout to make your file comply to rosdistros special formatting rules. It will change the files in place, so if you want to be extra safe you could back up your changes before.
    - `rosdistro_reformat file://$(pwd)/index.yaml`
- Verify your changes before commiting the changes
    - `git diff distribution.yaml`
    - `git commit -m "Adding my_repo_name to documentation index for distro" distribution.yaml`
- Push your changes to your remote GitHub repository `git push origin master`
- Submit a pull request on GitHub to the rosdistro project to add your rosinstall files to the indexer.

    - Concrete examples of a pull request: 🌐1 (https://github.com/ros/rosdistro/pull/13688/files), 🌐2 (https://github.com/ros/rosdistro/pull/13665/files)

2.2 Create a wiki page for your package in very short steps

1. Create a wiki.ros.org account if necessary and log on.
2. Enter into the wiki the url of the page you want to create.
    - e.g. "`http://wiki.ros.org/your_package`" for "your_package".
3. The page opened above will say the page does not exist, select the appropriate Template from the list to create your page.
    - For a package, choose "`PackageTemplate`".
4. We also encourage you to create usage documentation, tutorials, reviews, or whatever other content you want. See the guide to writing package documentation (/PackageDocumentation)

That's it! Once your pull request is serviced, documentation for your rosinstall files will be generated the next time the documentation jobs run.

3. Pull in changes from upstream

To pull in changes from github.com/ros/rosdistro, you need to add another remote

`git remote add upstream https://github.com/ros/rosdistro.git`

Now to pull changes other people have made and ROS has commited to github:

`git pull upstream`

Git will pull changes from upstream and automatically merge them into your current branch. This single step could won't notify you of conflicts between your local repo and upstream. You may have to identify a branch when using this: `git pull upstream master`

An alternative is a two step process:

`git fetch upstream`

`git merge upstream/master`

This two step process will prompt you if there are any conflicts between your local repo and upstream.

Check for the indexer to successfully index your ROS package

For every meta/package/stack you add an document job is created on the Jenkins server: 🌐 http://build.ros.org/ (http://build.ros.org/) (this might take a while after adding)

Simply enter the meta/package/stack name in the search box to find corresponding jobs. On the job page the status, failure message and full console output can be inspected.

Documenting Your ROS package Repository on ROS.org

Once your repository has been indexed, overnight the documentation jobs are recreated and run.

Creating a wiki page for your package is already explained above. To add documentation there, the following guidelines help you get started with documenting on the wiki:

- Wiki Style Guide (/StyleGuide)
- Guide to Writing Package Documentation (/PackageDocumentation)
- Guide to Writing Tutorials (/WritingTutorials)
- Guide to ROS.org Wiki Macros (/WikiMacros)

Some exemplars of good wiki documentation:

- tf (/tf)
- carrot_planner (/carrot_planner)
- actionlib (/actionlib)
- ground_station (/ground_station)
- velodyne_driver (/velodyne_driver)

Announce your package

Being indexed on `ros.org`, your package has far more visibility than ever before among the global community. Some people may find your package through Google or 🌐 ROS wiki's package search (http://www.ros.org/browse/list.php). It's always better, however, to let people know your work (see also 🌐 a discussion about how announcement helps (https://discourse.ros.org/t/new-software-release-announcements/1203)).

To do so simply make a new post on 🌐 "`general`" category on discourse.ros.org (http://discourse.ros.org/c/general).

If you have a notable update for a package that is already released, announcement for it is also appreciated.

1. Template for a new package announcement

If you wonder what to include in the announcement, you could add:

- Package name.
- Brief description of the package (this can be a copy from your package.xml. Don't forget to add description in your package.xml everyone btw!).
- Link to the package page on wiki.ros.org (explained earlier in this same wiki (/rosdistro/Tutorials/Indexing%20Your%20ROS%20Repository%20for%20Documentation%20Generation#Create_a_wiki_page_for_your_package_in_very_short_steps)
    - Link to the repository works too, although creating wiki page is strongly encouraged.
- Link to the tutorial of the package if exists.
- Acknowledgement for the funding, notable contributers if anything.

You can find 🌐 many previous concrete examples (https://www.google.com/search?client=ubuntu&channel=fs&q=site:lists.ros.org+(announce+OR+announcement+OR+announcing)&ie=utf-8&oe=utf-8).

Wiki: rosdistro/Tutorials/Indexing Your ROS Repository for Documentation Generation (última edición 2017-09-14 04:32:08 efectuada por ScottOBrien (/ScottOBrien))

Brought to you by: Open Source Robotics Foundation

(http://www.osrfoundation.org)