Releasing a new ROS 2 Distribution

Yadunund Vijay - Iron ROS Boss
Chris Lalancette - ROS 2 Technical Lead
Marco Gutierrez - Jazzy ROS Boss

October 19, 2023
Who are we?
Outline

● What is ROS?
● What is a ROS distribution?
● Release infrastructure
● Release process
What is ROS?
What is ROS?

Plumbing + Capabilities + Tools + Ecosystem

Over 1000 pkgs in the ROS ecosystem

Core ROS packages

HTTPS://XRCD.COM/2347/
Variants, Releases, and Target Platforms

REP 2000:

- Release frequency: 12 months alternating between LTS and non-LTS supports.
- Support durations:
  - LTS: 5 years
  - Non-LTS: 1.5 years
- Support levels:
  - Tier 1: Continuously tested with prioritized bug fixes.
  - Tier 2: Periodically tested.
  - Tier 3: Rely on community for testing and bug fixes.
- Target platforms:
  - Support level for combinations of CPU architectures and Operating systems.
Variants, Releases, and Target Platforms

Over 1000 pkgs in the ROS ecosystem

Core ROS packages

REP 2001:

- **core**: ament, launch, rclcpp, rclpy, ros2cli, default rmw implementation
- **base**: core + geometry2, kdl_parser, robot_state_publisher, rosbag2, urdf
- **desktop**: base + rviz, examples, demos
- **perception**: base + image_common, vision_opencv, PCL, ...
- **simulation**: base + ros_gz_bridge + ros_gz_interfaces
- **desktop_full**: desktop + perception, simulation, ros_gz_demos

apt install ros-iron-desktop-full


HTTPS://XKCD.COM/2347/
Variants, Releases, and Target Platforms

Over 1000 pkgs in the ROS ecosystem

Core ROS packages

REP 2001:
- **core**: ament, launch, rclcpp, rclpy, default rmw implementation
- **base**: core + geometry2, kdl_parser, robot_state_publisher, rosbag2, urdf
- **desktop**: core + rviz + examples + demos + introspection tooling
- **perception**: base + image_common, vision_opencv...
- **simulation**: base + ros_gz_bridge + ros_gz_interfaces
- **desktop_full**: desktop + perception, simulation, ros_gz_demos

On release day, the desktop variant is available

apt install ros-iron-desktop-full
New distro creation

New Distro branch created for desktop pkgs

New Distro released

“main”

ROS 2 ROLLING RIDLEY
But why?

- Once released, the API/ABI for core ROS packages will not change throughout the life of the distribution.
  - The Major and Minor version numbers will remain constant while Patch number can be incremented for bug fixes.
- All feature development will target rolling.
- New distros give maintainers the opportunity to make improvements that necessitate API/ABI breaks.
- Community maintained packages do not need to follow such guarantees although encouraged.
Types of release

- **Source release**
  - Tagged version of the `ros2.repos` file

- **Binary release**
  - Generated by binary jobs on the Buildfarm
    - Debians for Ubuntu
    - RPM for RHEL

- **Archive release (pre-built binaries)**
  - Generated by “packaging” job on CI
    - For Tier 1 and Tier 2 supported platforms
      - Ubuntu
      - RHEL
      - Windows

https://github.com/ros2/ros2/releases
Release infrastructure

For detailed talk on how packaging works, watch “The ROS build farm and you: How ROS packages you release become binary packages”
Release infrastructure- ros2.repos

- `<distro>/ros2.repos`: A repos file that clones in all desktop variant packages for a given ROS distro with the correct “source” branch for each repo
  - 106 repos for iron
  - 358 packages

- `<distro>-release/ros2.repos`: Clones in latest release version of each repository

Release infrastructure- bloom

- A python tool to automate release branching and the generation of platform specific source packages, like debian’s src-debs.

- The buildfarm pulls source code from each target branch and builds each package independently.

ros2/rclcpp

ros2-gbp/rclcpp-release
Release infrastructure- rosdistro

- Each distro has a `distribution.yaml` file with entities for each repository released via the ROS Buildfarm (core and community maintained).
  - Source of truth / database

- Specifies
  - Packages released from this repository
  - Source branch for document generation
  - Latest version of package released via the Buildfarm
  - URL of release repository
  - URL of source repository and branch

- Updated via PRs opened automatically when the maintainer blooms a new release.

- Changes to the `distribution.yaml` are automatically polled to trigger binary jobs on the Buildfarm.
Release infrastructure - Buildfarm

https://build.ros.org/
https://github.com/ros2/ros_buildfarm_config
**Release infrastructure - Communication**

- **ROS Discourse**
  - discourse.ros.org
  - Announcements to the community before, during and after release

- **ROS Index**
  - index.ros.org
  - Distro specific information on package versions, documentation

- **ROS 2 Documentation: Rolling**
  - Search docs
  - Release timeline, changelogs, setup instructions, tutorials, guides and more

- **https://build.ros2.org/**
The release process
Rough release steps

1. Prepare
2. Freeze
3. Update Rolling core package binaries
4. Branch
5. Test
6. Release!
7. Post-release!
Prepare: Name release

- Starts in April of previous year
- Follows alphabet order in the form of “<adjective> <turtle species>”
  - e.g. Iron Irwini, Jazzy Jalisco, K-turtle
- Ideas are gathered from the community through a thread on discourse
- List of names are compiled by the ROS Boss
- Names are eliminated:
  - Anything copyrighted (no Ninja Turtles)
  - Offensive
  - Unsuitable
  - Too long (generally longer than 8 characters)
- ROS 2 core team votes on the names
- ROS Boss makes the final selection
- Name is announced on the previous release announcement post on discourse.
Prepare: Develop Release timeline

Release Timeline

November, 2023 - Platform decisions
REP 2000 is updated with the target platforms and major dependency versions.

By January, 2024 - Rolling platform shift
Build farm is updated with the new platform versions and dependency versions for Jazzy Jalisco.

Mon. April 8, 2024 - Alpha + RMW freeze
Preliminary testing and stabilization of ROS Base packages, and API and feature freeze for RMW provider packages.

Mon. April 15, 2024 - Freeze
API and feature freeze for ROS Base packages in Rolling Ridley. Only bug fix releases should be made after this point. New packages can be released independently.

Mon. April 22, 2024 - Branch
Branch from Rolling Ridley. rosdistro is reopened for Rolling PRs for ROS Base packages. Jazzy development shifts from ros-rolling- to ros-jazzy- packages.

Mon. April 29, 2024 - Beta
Updated releases of ROS Desktop packages available. Call for general testing.

Wed, May 1, 2024 - Kick off of Tutorial Party
Tutorials hosted at https://github.com/osrf/ros2_test_cases are open for community testing.

Mon. May 13, 2024 - Release Candidate
Release Candidate packages are built. Updated releases of ROS Desktop packages available.

Mon. May 20, 2024 - Distro Freeze
Freeze rosdistro. No PRs for Jazzy on the rosdistro repo will be merged (reopens after the release announcement).

Thu. May 23, 2024 - General Availability
Release announcement. rosdistro is reopened for Jazzy PRs.

[1] (1,2,3): The ros_base variant is described in REP 2001 (ros-base).

Prepare: Migrate Rolling onto next Ubuntu

- In even years, migrate to the next Ubuntu (e.g. for 2024, migrate to Ubuntu 24.04)
- When Ubuntu 24.04 alpha packages are available, do the migration (~December/January)
- Now Rolling binary packages will be available targeting next Ubuntu
- CI ([https://ci.ros2.org](https://ci.ros2.org)) also targets next Ubuntu
Prepare: Commission logo/turtle icon

- ~January
- Contact the artist (Joshua Ellingson)
- Give artist the release name and some ideas for logo
- Artist comes up with a few sketches
- ROS 2 core team votes on options
- Artist does final artwork
- Adapt artwork to turtlesim icon
• Announced on discourse
• Typically happens one week before overall freeze (Jazzy: April 8, 2024)
• After RMW freeze, no new features or APIs to the rmw layer or below
  ○ Includes the DDS implementations (e.g. Fast-DDS, CycloneDDS, etc)
  ○ Includes the rmw implementations (e.g. rmw_fastrtps, rmw_cyclonedds, etc)
  ○ Includes the rmw API
- 6 weeks before release (Jazzy: April 15, 2024)
- After freeze, no more API changes or features in core packages
- Bug fixes are still allowed
Update Rolling core package binaries

- Release all changes that made it into Rolling before the freezes
  - Bump and tag source version on rolling
  - Bloom changes
- Ensures all binary packages are up-to-date with the sources
Branch: sources for core repositories

- Branch core sources off of Rolling
  - each core repository (eg. rclcpp) will now have a <distro> branch
- This opens Rolling back up for API/ABI breaking changes
- But keeps <distro> stable for testing and release
- Create a new <distro> ros2.repos file
  - Sources reference the branches created above
  - CI jobs will now use this ros2.repos file
Branch: create new distro binaries

- Run a script which “migrates” Rolling to the new distribution
  - Creates a new rosdistro/distribution.yaml file
  - Creates new release track in the ros2-gbp repositories
  - Creates binaries for the new distribution

- This is why all releases in Rolling must use https://github.com/ros2-gbp for release repository; easiest way to ensure we have permissions to do this branching
Test: interim tarballs

- Tarballs are built at [https://ci.ros2.org/view/packaging/](https://ci.ros2.org/view/packaging/)
- Can be downloaded and run without installing anything additional
- Available for all Tier-1 and Tier-2 platforms:
  - Ubuntu Linux amd64
  - Ubuntu Linux arm64
  - RHEL Linux amd64
  - Windows amd64
Test: announce branch and beta

- Announced on discourse
- Also call for testing during the tutorial party
Build & test: tutorial party

- Core devs and community test out the code
- List of test cases is generated at https://github.com/osrf/ros2_test_cases/
- Make fixes to the core based on testing
  - Bump patch versions and bloom releases
Release!

- Mark distribution as active on
  - rosdistro
  - index.ros.org
  - ros.org
  - docs.ros.org
- Run “sync” job on buildfarm to move packages from ros2-testing into main.
- Create tagged source release on ros2/ros2 and upload tarballs.
- Make the announcement post on discourse
Post release!

● Reflect on release with ROS 2 working group team and garner feedback.
● Ensure docker image is available for new distribution.
● Prepare for Patch release 1 if any fixes are needed right after release.
● Periodically sync packages and fix bugs that may arise along the lifetime of the release.
Conclusion