



Minimum language requirements:
C++14
Python 3.7

EOL May 2023

Feature	ROS1	ROS2	Change 1-10
Command Line	roslaunch	ros2 run	2 -New options, some name changes
Platforms	Linux (Mostly)	Linux, Windows, IOS, micros	10
Python	2.x	3.X OOP	2-3
C++	C++ 03	C++ >=11 OOP	2-3
Client Libraries	rospy, roscpp	rclpy, rclcpp	5
Nodes	One node in process	Multiple nodes in a process	2
Topics	rostopic list	ros2 topic list	1

Feature	ROS1	ROS2	Change 1-10
Messages	type/show	Interface - To define format.	2-3
Services	synchronous	asynchronous	3
Actions -goal, feedback, success (Asynchronous)	An add-on	Now Command Line actions possible-	2
Parameters	No Master or Parameter Server	Nodes handle parameters	2
Tools	rqt, wtf, log	Try them – seems OK	
Simulation	Gazebo 7	Gazebo 11	2
Visualization	RVIZ	RVIZ 2	2

<https://roboticsbackend.com/ros1-vs-ros2-practical-overview/#Services>

	Required Support			Recommended Support	
Package	Ubuntu Focal	MacOS**	Windows 10**	Debian Buster	OpenEmbedded**
CMake	3.16.3	3.14.4	3.14.4	3.13.4	3.16.1 / 3.12.2****
EmPY	3.3.2				
Gazebo	11.0.0*	11.0.0	N/A	11.0.0*	N/A
Ignition	Citadel*		N/A	Citadel*	N/A
Ogre	1.10*				N/A
OpenCV	4.2.0	4.2.0	3.4.6*	3.2.0	4.1.0 / 3.2.0****
OpenSSL	1.1.1d	1.1.1f	1.1.1f	1.1.1d	1.1.1d / 1.1.1b****
Poco	1.9.2	1.9.0	1.8.0*	1.9.0	1.9.4
Python	3.8.0	3.8.2	3.8.0	3.7.3	3.8.2 / 3.7.5****
Qt	5.12.5	5.12.3	5.10.0	5.11.3	5.14.1 / 5.12.5****
		Linux only			
PCL	1.10.0	N/A	N/A	1.9.1	1.10.0
RMW DDS Middleware Providers					
Connex DDS	5.3.1			N/A	
Cyclone DDS	0.7.x (Coquette)				
Fast-RTPS	2.0.x				

Index of ROS Enhancement Proposals (REPs)

<https://www.ros.org/reps/rep-0000.html>

<https://www.ros.org/reps/rep-2000.html#foxy-fitzroy-may-2020-may-2023>

Feature	ROS1	ROS2	Change 1-10
Launch Files	In ROS1 roslaunch files are defined in XML	In ROS2 launch files are written in Python which enables to use more complex logic like conditional.	3-4
Package Creation	catkin_create_pkg catkin_make	Python or CMake ament_python ament_cmake	3-4
Source	Source (workspace) ~/catkin_ws/devel/setup.bash	In Workspace source install/setup.bash	1
Package Files	CMakeLists.txt package.xml	package.xml, setup.py, setup.cfg __init__.py (ament_python)	3-5
		CMakeLists.txt, package.xml	
Build	catkin_make	colcon build	2

THE BIG DIFFERENCES

Feature	ROS1	ROS2	Change 1-10
Middleware	Like TCP/IP	Currently all implementations of this interface are based on the DDS standard	10
QoS (Quality of Service)	N/A –A bit line Queue size.	In order for a connection to be made, all of the policies that affect compatibility must be compatible.	8
Real Time	A Patch - Xenomai or RT_PREEMPT	A work in progress	6
Lifecycle of nodes startup	Random unless “Wait” states are used	The life cycle can be used by tools like roslaunch to start a system composed of many components in a deterministic way.	10
Security	A Patch	Will depend on DDS implementation.	10

<https://docs.ros.org/en/rolling/Concepts/About-Quality-of-Service-Settings.html>

<https://roscon.ros.org/2015/presentations/RealtimeROS2.pdf>

<https://docs.ros.org/en/foxy/Tutorials/Real-Time-Programming.html>

To make a real-time computer system, our real-time loop must update periodically to meet deadlines. We can only tolerate a small margin of error on these deadlines (our maximum allowable jitter). To do this, we must avoid nondeterministic operations in the execution path, things like: pagefault events, dynamic memory allocation/deallocation, and synchronization primitives that block indefinitely.

A classic example of a controls problem commonly solved by real-time computing is balancing an [inverted pendulum](#).

ROS 2 Foxy Fitzroy: Setting a new standard for production robot development

<https://aws.amazon.com/blogs/robotics/ros2-foxy-fitzroy-robot-development/>