

ROS Toolbox Release Notes



MATLAB® & SIMULINK®



How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

ROS Toolbox Release Notes

© COPYRIGHT 2019–2021 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

R2021a

C++ Code Generation Support for ROS 1: Generate ROS nodes for deploying to target hardware using MATLAB Coder	1-2
ROS Message Structures: Improve performance of ROS messages using structures	1-2
New ROS Functions for Messages: Use new functions for processing messages as structures	1-3
ROS Header Assignment Block: Update fields of ROS Header messages	1-4
ROS Toolstrip Update: Select ROS network types and deploy local or remote nodes with code generation	1-4
ROS 2 Log File: Open and parse ROS 2 bag file	1-4

R2020b

Velodyne ROS Message Reader: Read Velodyne lidar messages from ROS in MATLAB	2-2
ROS Toolbox Interface for ROS Custom Messages add-on added to ROS Toolbox	2-2
ROS Toolbox requires Python 2.7 for starting a ROS Master from MATLAB	2-2
ROS Melodic: Update ROS 1 version support to Melodic Morenia distribution	2-2
Sequence Numbers for Header Messages	2-3

ROS 2 Dashing: Update ROS 2 version support to Dashing Diademata distribution	3-2
--	------------

Network Connection and Exploration: Communicate with ROS and ROS 2 nodes in a network using MATLAB and Simulink	4-2
Multiplatform Support: Access ROS functionality from Windows, Mac, and Linux	4-2
Publishers and Subscribers: Send and receive ROS and ROS 2 messages with MATLAB and Simulink via a ROS network	4-2
Custom Messages: Generate custom messages to use on both ROS and ROS 2 networks based on specified packages	4-2
Log File Playback: Import ROS log files (rosbags) to filter, visualize, and analyze logged data	4-2
Deployment of ROS Nodes: Deploy ROS and ROS 2 nodes to target hardware using Simulink Coder	4-3
ROS Toolbox Support Package for TurtleBot-Based Robots: Connect to TurtleBot hardware	4-3

R2021a

Version: 1.3

New Features

Bug Fixes

C++ Code Generation Support for ROS 1: Generate ROS nodes for deploying to target hardware using MATLAB Coder

Key ROS features now support code generation using MATLAB® Coder™ including:

- `rossubscriber`
- `rospublisher`
- `rosmessage`
- `rosrate`
- `rostime`
- `rosduration`

Many other ROS-related functions also now support code generation. For details about code generation support and limitations, check the **Extended Capabilities** section for each reference page.

For code generation, you must use message structures instead of message objects. For more information, see the ROS Message Structures on page 1-2 release note.

ROS Message Structures: Improve performance of ROS messages using structures

You can now create messages as structures with fields matching the message object properties. Using structures typically improves performance when you create, update, and use ROS messages, but message fields are no longer validated when set.

To use ROS messages as structures, set the 'DataFormat' name-value pair argument when creating your publishers, subscribers, or other ROS objects. Any messages generated from these objects use structures.

```
pub = rospublisher("/scan", "sensor_msgs/LaserScan", "DataFormat", "struct")
msg = rosmessage(pub)
```

```
msg =
```

```
struct with fields:
```

```
MessageType: 'sensor_msgs/LaserScan'
Header: [1x1 struct]
AngleMin: 0
AngleMax: 0
AngleIncrement: 0
TimeIncrement: 0
ScanTime: 0
RangeMin: 0
RangeMax: 0
Ranges: [0x1 single]
Intensities: [0x1 single]
```

You can also create messages as structures directly, but other ROS objects use message objects by default. Make sure to specify the data format as "struct" for the publisher, subscriber, or other ROS objects as well.

```
msg = rosmessage("/scan", "sensor_msgs/LaserScan", "DataFormat", "struct")
pub = rospublisher("/scan", "sensor_msgs/LaserScan", "DataFormat", "struct")
```

These functions support structure messages using the 'DataFormat' name-value argument:

- rosmessage
- rospublisher
- rossubscriber
- rostime
- rostf
- rosduration
- rosvcclient
- rosvccserver
- rosactionclient

For more information, see “Improve Performance of ROS Using Message Structures”.

New ROS Functions for Messages: Use new functions for processing messages as structures

Functions that operate on specialized ROS messages are now available to support the use of message structures as inputs. These new functions replace the existing object functions of message objects, and support ROS and ROS 2 message structures as inputs instead of message objects.

The object functions will be removed in a future release.

Message Types	Object Function Name	New Function Name
Image	readImage	rosReadImage
CompressedImage	writeImage	rosWriteImage
LaserScan	readCartesian	rosReadCartesian
	readScanAngles	rosReadScanAngles
	lidarScan	rosReadLidarScan
	plot	rosPlot
PointCloud2	apply	rosApplyTransform
	readXYZ	rosReadXYZ
	readRGB	rosReadRGB
	readAllFieldNames	rosReadAllFieldNames
	readField	rosReadField
	scatter3	rosPlot
Quaternion	readQuaternion	rosReadQuaternion

Message Types	Object Function Name	New Function Name
OccupancyGrid	readBinaryOccupancyGrid readOccupancyGrid writeBinaryOccupancyGrid writeOccupancyGrid	rosReadBinaryOccupancyGrid rosReadOccupancyGrid rosWriteBinaryOccupancyGrid rosWriteOccupancyGrid
Octomap	readOccupancyMap3D	rosReadOccupancyMap3D
PointStamped PoseStamped QuaternionStamped Vector3Stamped TransformStamped	apply	rosApplyTransform
All messages	showdetails	rosShowDetails

ROS Header Assignment Block: Update fields of ROS Header messages

The Header Assignment block enables you to update the fields of a ROS Header message in Simulink®. You can specify the frame ID manually, set the time stamp based on the current ROS time, and use custom Header field names for your specific ROS application.

ROS Toolstrip Update: Select ROS network types and deploy local or remote nodes with code generation

The **ROS** tab on the Simulink Toolstrip now enables you to switch between ROS and ROS 2 board implementations and specify a ROS device for deploying ROS nodes from Simulink.

To add the **ROS** tab to the Simulink Toolstrip, open the **APPS** tab and click the down arrow. Under **Control Systems**, select **Robot Operating System (ROS)** to add the tab to your model.

The **ROS Network** list allows you to choose between ROS, Raspberry Pi™ ROS, or ROS 2. This sets the **Hardware board** and other configuration parameters appropriately.

The **Deploy to** list enables you to select **Localhost** or **Remote Device**. You can also set your remote device parameters by selecting **Manage Remote Device**.

ROS 2 Log File: Open and parse ROS 2 bag file

Use the `ros2bag` object to open, parse, and store all messages from the ROS 2 bag file. You can retrieve the messages from the `ros2bag` object using the `readMessages` function.

R2020b

Version: 1.2

New Features

Bug Fixes

Compatibility Considerations

Velodyne ROS Message Reader: Read Velodyne lidar messages from ROS in MATLAB

The `velodyneROSMessageReader` object reads ROS messages from a rosbag or ROS network and loads the point cloud data into MATLAB as `pointCloud` objects. The `velodyneROSMessageReader` object supports the 'VLP16', 'VLP32C', 'HDL32E', and 'HDL64E' device models.

ROS Toolbox Interface for ROS Custom Messages add-on added to ROS Toolbox

The ROS Toolbox Interface for ROS Custom Messages add-on is now part of the ROS Toolbox installation. You no longer need to use the **Add-On Explorer** to install the `rosgenmsg` and `ros2genmsg` functions.

Also, you no longer need to restart MATLAB after adding custom message definitions.

To use custom messages, follow these steps:

- Use the `rosgenmsg` or `ros2genmsg` function with the path to the parent folder of your custom message packages as the input argument.
- Add the custom message folder to the MATLAB:

```
addpath('<folderpath>/matlab_msg_gen_ros1/<arch>/install/m')  
  
savepath
```

- Refresh all message class definitions, which requires clearing the workspace:

```
clear classes  
  
rehash toolboxcache
```

No MATLAB restart is required to see the updated ROS messages.

ROS Toolbox requires Python 2.7 for starting a ROS Master from MATLAB

When connecting to a ROS network, you must have Python version 2.7. Download Python from the homepage and set the version using the `pyenv` function.

Compatibility Considerations

To connect to a ROS network using the `rosinit` function, set your Python version using the `pyenv` function. The selected version of Python will be used for all subsequent MATLAB operations, and is persistent across sessions.

ROS Melodic: Update ROS 1 version support to Melodic Morenia distribution

Starting this release, the ROS Toolbox supports the ROS Melodic Morenia distribution. A ROS distribution is a stable set of ROS packages that define the supported platforms, message definitions,

and new features for the release. In previous releases, the ROS Toolbox supported the Indigo Igloo distribution.

Sequence Numbers for Header Messages

Compatibility Considerations

Previously, when manually setting the `Seq` property of a ROS Header message (`std_msgs/Header`), a subscriber that receives this message would keep the given sequence number (from the published message) and not increment the value.

In R2020b, the subscriber always increments the sequence value whenever receiving a new message by adding one to the `Seq` property.

R2020a

Version: 1.1

Bug Fixes

ROS 2 Dashing: Update ROS 2 version support to Dashing Diademata distribution

Starting this release, the ROS Toolbox supports the Dashing Diademata distribution. A ROS distribution is a stable set of ROS packages that define the supported platforms, message definitions, and new features for the release. In previous releases, the ROS Toolbox supported the Bouncy Bolson distribution.

R2019b

Version: 1.0

New Features

Network Connection and Exploration: Communicate with ROS and ROS 2 nodes in a network using MATLAB and Simulink

Connect to ROS and ROS 2 to prototype robotics applications and access robotics hardware or simulators over a ROS network. You can create your own ROS network using MATLAB or connect to an existing ROS network. To set up a ROS network, start by calling `rosinit`. For ROS 2 networks, see `ros2node`.

For more information, see [Network Connection and Exploration](#).

Multiplatform Support: Access ROS functionality from Windows, Mac, and Linux

The ROS Toolbox enables you to connect to and run ROS and ROS 2 networks on Windows®, Mac, and Linux platforms.

Publishers and Subscribers: Send and receive ROS and ROS 2 messages with MATLAB and Simulink via a ROS network

ROS shares information using messages. Messages are a simple data structure for sharing data. To receive, or subscribe to, a message, use `rossubscriber` or `ros2subscriber`. To send, or publish, a message, use `rospublisher` or `ros2publisher`. For an example of sending and receiving messages, see [Exchange Data with ROS Publishers and Subscribers](#) or [Exchange Data with ROS 2 Publishers and Subscribers](#).

For more information, see [Publishers and Subscribers](#).

Custom Messages: Generate custom messages to use on both ROS and ROS 2 networks based on specified packages

You can create your own ROS custom messages and use them in MATLAB and Simulink with ROS networks to transmit information. For ROS custom messages, use `rosAddons` to install the necessary addon, and then use the `rosgenmsg` function. To learn the requirements for generating custom messages, see [ROS Custom Message Support](#). For ROS 2, use `ros2genmsg` with your custom message packages and see the [ROS 2 Custom Message Support example](#).

Log File Playback: Import ROS log files (rosbags) to filter, visualize, and analyze logged data

ROS topics are stored in log files called rosbags. You can access and filter information from rosbags in MATLAB. For an example of working with rosbags, see [Work with rosbag Logfiles](#).

You can access transformations between coordinate systems as ROS topics and use them to transform data in MATLAB. For more information, see [Access the tf Transformation Tree in ROS](#).

For more information, see [ROS Log Files and Transformations](#)

Deployment of ROS Nodes: Deploy ROS and ROS 2 nodes to target hardware using Simulink Coder

For examples that generate code for standalone ROS nodes, see:

- [Generate a Standalone ROS Node from Simulink®](#)
- [Generate a Standalone ROS 2 Node from Simulink®](#)

ROS Toolbox Support Package for TurtleBot-Based Robots: Connect to TurtleBot hardware

For more information, see [ROS Toolbox Support Package for TurtleBot -Based Robots](#).

