

A Gentle Introduction to ROS

Front matter

Jason M. O’Kane

Jason M. O’Kane
University of South Carolina
Department of Computer Science and Engineering
315 Main Street
Columbia, SC 29208

<http://www.cse.sc.edu/~jokane>

©2014, Jason Matthew O’Kane. All rights reserved.

This is version 2.1.3 (5776342), generated on April 20, 2016.

Typeset by the author using \LaTeX and `memoir.cls`.

ISBN 978-14-92143-23-9

A Gentle Introduction to ROS

Jason M. O’Kane

Jason M. O’Kane
University of South Carolina
Department of Computer Science and Engineering
315 Main Street
Columbia, SC 29208

<http://www.cse.sc.edu/~jokane>

©2014, Jason Matthew O’Kane. All rights reserved.

This is version 2.1.3 (5776342), generated on April 20, 2016.

Typeset by the author using \LaTeX and `memoir.cls`.

ISBN 978-14-92143-23-9

Contents in Brief

Contents in Brief	iii
Contents	v
1 Introduction	1
<i>In which we introduce ROS, describe how it can be useful, and preview the remainder of the book.</i>	
2 Getting started	11
<i>In which we install ROS, introduce some basic ROS concepts, and interact with a working ROS system.</i>	
3 Writing ROS programs	39
<i>In which we write ROS programs to publish and subscribe to messages.</i>	
4 Log messages	61
<i>In which we generate and view log messages.</i>	
5 Graph resource names	77
<i>In which we learn how ROS resolves the names of nodes, topics, parameters, and services.</i>	

6 Launch files	83
<i>In which we configure and run many nodes at once using launch files.</i>	
7 Parameters	105
<i>In which we configure nodes using parameters.</i>	
8 Services	117
<i>In which we call services and respond to service requests.</i>	
9 Recording and replaying messages	133
<i>In which we use bag files to record and replay messages.</i>	
10 Conclusion	141
<i>In which we preview some additional topics.</i>	
Index	145

Contents

Contents in Brief	iii
Contents	v
1 Introduction	1
1.1 Why ROS?	1
Distributed computation	2
Software reuse	2
Rapid testing	3
ROS is not	4
1.2 What to expect	4
Chapters and dependencies	5
Intended audience	5
1.3 Conventions	7
1.4 Getting more information	7
Distributions	8
Build systems	9
1.5 Looking forward	9
2 Getting started	11
2.1 Installing ROS	11

	Adding the ROS repository	11
	Installing the package authentication key	12
	Downloading the package lists	12
	Installing the ROS packages	13
	Installing turtlesim	13
	Setting up rosdep systemwide	13
2.2	Configuring your account	14
	Setting up rosdep in a user account	14
	Setting environment variables	14
2.3	A minimal example using turtlesim	15
	Starting turtlesim	16
2.4	Packages	17
	Listing and locating packages	17
	Inspecting a package	18
2.5	The master	20
2.6	Nodes	21
	Starting nodes	21
	Listing nodes	22
	Inspecting a node	23
	Killing a node	23
2.7	Topics and messages	24
	2.7.1 Viewing the graph	24
	2.7.2 Messages and message types	27
	Listing topics	27
	Echoing messages	28
	Measuring publication rates	28
	Inspecting a topic	28
	Inspecting a message type	30
	Publishing messages from the command line	31
	Understanding message type names	33
2.8	A larger example	34
	2.8.1 Communication via topics is many-to-many.	35
	2.8.2 Nodes are loosely coupled.	36
2.9	Checking for problems	37
2.10	Looking forward	37
3	Writing ROS programs	39
	3.1 Creating a workspace and a package	39

	Creating a workspace	39
	Creating a package	40
	Editing the manifest	41
3.2	Hello, ROS!	41
3.2.1	A simple program	41
3.2.2	Compiling the Hello program	44
	Declaring dependencies	44
	Declaring an executable	45
	Building the workspace	45
	Sourcing setup.bash	46
3.2.3	Executing the hello program	47
3.3	A publisher program	47
3.3.1	Publishing messages	49
	Including the message type declaration	49
	Creating a publisher object	49
	Creating and filling in the message object	51
	Publishing the message	51
	Formatting the output	52
3.3.2	The publishing loop	52
	Checking for node shutdown	52
	Controlling the publishing rate	53
3.3.3	Compiling pubvel	54
	Declaring message type dependencies	54
3.3.4	Executing pubvel	54
3.4	A subscriber program	55
	Writing a callback function	55
	Creating a subscriber object	57
	Giving ROS control	59
3.4.1	Compiling and executing subpose	60
3.5	Looking forward	60
4	Log messages	61
4.1	Severity levels	61
4.2	An example program	62
4.3	Generating log messages	62
	Generating simple log messages	62
	Generating one-time log messages	65
	Generating throttled log messages	65

4.4	Viewing log messages	67
4.4.1	Console	68
	Formatting console messages	68
4.4.2	Messages on rosout	69
4.4.3	Log files	72
	Finding the run id	72
	Checking and purging log files	72
4.5	Enabling and disabling log messages	73
	Setting the logger level from the command line	74
	Setting the logger level from a GUI	75
	Setting the logger level from C++ code	75
4.6	Looking forward	76
5	Graph resource names	77
5.1	Global names	77
5.2	Relative names	78
	Resolving relative names	79
	Setting the default namespace	79
	Understanding the purpose of relative names	80
5.3	Private names	80
5.4	Anonymous names	81
5.5	Looking forward	82
6	Launch files	83
6.1	Using launch files	83
	Executing launch files	83
	Requesting verbosity	85
	Ending a launched session	85
6.2	Creating launch files	86
6.2.1	Where to place launch files	86
6.2.2	Basic ingredients	86
	Inserting the root element	86
	Launching nodes	87
	Finding node log files	88
	Directing output to the console	88
	Requesting respawning	89
	Requiring nodes	89
	Launching nodes in their own windows	90

6.3	Launching nodes inside a namespace	91
6.4	Remapping names	93
6.4.1	Creating remappings	93
6.4.2	Reversing a turtle	95
6.5	Other launch file elements	97
6.5.1	Including other files	97
6.5.2	Launch arguments	99
	Declaring arguments	100
	Assigning argument values	101
	Accessing argument values	101
	Sending argument values to included launch files	101
6.5.3	Creating groups	102
6.6	Looking forward	104
7	Parameters	105
7.1	Accessing parameters from the command line	105
	Listing parameters	105
	Querying parameters	106
	Setting parameters	107
	Creating and loading parameter files	107
7.2	Example: Parameters in turtlesim	108
	Reading the background color	109
	Setting the background color	109
7.3	Accessing parameters from C++	110
7.4	Setting parameters in launch files	113
	Setting parameters	113
	Setting private parameters	113
	Reading parameters from a file	114
7.5	Looking forward	115
8	Services	117
8.1	Terminology for services	117
8.2	Finding and calling services from the command line	118
	Listing all services	118
	Listing services by node	119
	Finding the node offering a service	120
	Finding the data type of a service	120
	Inspecting service data types	121

Calling services from the command line	122
8.3 A client program	123
Declaring the request and response types	123
Creating a client object	123
Creating request and response objects	125
Calling the service	125
Declaring a dependency	127
8.4 A server program	127
Writing a service callback	127
Creating a server object	129
Giving ROS control	130
8.4.1 Running and improving the server program	130
8.5 Looking ahead	131
9 Recording and replaying messages	133
9.1 Recording and replaying bag files	133
Recording bag files	133
Replaying bag files	134
Inspecting bag files	134
9.2 Example: A bag of squares	135
Drawing squares	135
Recording a bag of squares	135
Replaying the bag of squares	136
9.3 Bags in launch files	139
9.4 Looking forward	140
10 Conclusion	141
10.1 What next?	141
Running ROS over a network	141
Writing cleaner programs	142
Visualizing data with rviz	142
Creating message and service types	142
Managing coordinate frames with tf	142
Simulating with Gazebo	143
10.2 Looking forward	143
Index	145