rosparam

Package Links
Code API
Tutorials
FAQ
Changelog
Change List
Reviews

Dependencies (2)
Used by (3)
Jenkins jobs (10)

Package Summary

Released | Continuous Integration: 1371 / 1372 | Documented

rosparam contains the rosparam command-line tool for getting and setting ROS Parameters on the Parameter Server using YAML-encoded files. It also contains an experimental library for using YAML with the Parameter Server. This library is intended for internal use only. rosparam can be invoked within a roslaunch file.

- Maintainer status: maintained
- Maintainer: Dirk Thomas <dthomas AT osrfoundation DOT org>
- Author: Ken Conley
- License: BSD
- Source: git https://github.com/ros/ros_comm.git (branch: melodic-devel)

Contents
1. YAML Format
2. roslaunch API
3. rosparam command-line tool
    1. rosparam list
    2. rosparam get
    3. rosparam set
    4. rosparam delete
    5. rosparam dump
    6. rosparam load
4. Roadmap
5. See Also

1. YAML Format

YAML is a lightweight markup language that supports all parameter types. For more on YAML, see the YAML 1.2 spec. For tips on entering YAML in at the command-line, please see the YAML command line guide.

rosparam uses a 1-to-1 correspondence between Parameter Server types and YAML types. For example:

```
string: 'foo'
integer: 1234
float: 1234.5
boolean: true
list: [1.0, mixed list]
dictionary: {a: b, c: d}
```

YAML dictionaries can occur as the argument to the load and set commands of rosparam. Dictionaries in this context are interpreted differently from namespace dictionaries that are set as the value of a parameter (e.g. from a C++ or Python node). Instead of *replacing* a parameter namespace, dictionaries *in rosparam* are unpacked into individual parameters to be set on the Parameter Server. Thus, rosparam dictionaries can be thought of as *adding* new values to a parameter namespace.

There are also special converters for angle radian/degree representations. Any Python-legal mathematical expression can be used with the radian value, with pi used to represent pi.

```
angle1: rad(2*pi)
angle2: deg(180)
```

or,

```
angle1: !degrees 181.0
angle2: !radians 3.14169
```

In either case, the angle value is converted to radians (float).

Wiki
Distributions
ROS/Installation
ROS/Tutorials
RecentChanges
rosparam

Page
Immutable Page
Info
Attachments
More Actions:

User
Login

2. roslaunch API

The <rosparam> tag enables the use of the rosparam tool for loading and dumping parameters encoded in YAML files. The <rosparam> tag can be put inside of a <node> tag, in which case the parameter is treated like a private name.

3. rosparam command-line tool

The rosparam tool enables command-line setting and getting of parameters as well as loading and dumping Parameter Server state to a file. The currently supported commands are:

```
rosparam set    set parameter
rosparam get    get parameter
rosparam load   load parameters from file
rosparam dump   dump parameters to file
rosparam delete delete parameter
rosparam list   list parameter names
```

Command-line arguments to rosparam obey the ROS_NAMESPACE environment variable (see Environment Variables). Parameter names that are not globally specified are resolved with respect to ROS_NAMESPACE.

NOTE: get and dump are essentially the same command, as are set and load, with the only difference being whether or not a file is used.

3.1 rosparam list

list

list all parameter names.

```
$ rosparam list
```

list <namespace>

list all parameters in a particular namespace.

```
$ rosparam list /namespace
```

3.2 rosparam get

get <parameter-name>

Get a parameter value.

```
$ rosparam get parameter_name
```

-p

Pretty-print output. *WARNING: this is not YAML-safe.*

-v

Show verbose output.

3.3 rosparam set

set <parameter-name> [parameter-value]

Set a parameter to a value. parameter-value is required unless --textfile or --binfile are specified.

```
$ rosparam set parameter_name value
```

NOTE: if parameter value is a dictionary, this will *add* to the currently set values.

Examples:

Setting a list with one as a string, integer, and float:

```
$ rosparam set /foo "['1', 1, 1.0]"
```

Setting an entire namespace of parameters using a YAML dictionary:

```
$ rosparam set /gains "p: 1.0
i: 1.0
d: 1.0"
```

-v

Show verbose output.

-t <text_file>, --textfile <text_file> **New in Diamondback**

Set parameter to contents of text file.

-b <binary_file>, --binfile <binary_file> **New in Diamondback**

Set parameter to contents of binary file. Parameter Server will store value as an XML-RPC Binary type (Base 64 encoded).

3.4 rosparam delete

delete <parameter-name>

Delete a parameter value.

```
$ rosparam delete parameter_name
```

-v

Show verbose output.

3.5 rosparam dump

dump <file>

Dump the YAML-formatted contents of the Parameter Server to a file.

```
$ rosparam dump dump.yaml
```

dump <file> <namespace>

Dump only the the parameters in the specified namespace.

```
$ rosparam dump dump.yaml /namespace
```

-v

Verbose output. e.g.:

```
$ rosparam dump -v gains.yaml /gains
dumping namespace [/gains] to file [gains.yaml]
/gains/i=1.0
/gains/p=1.0
/gains/d=1.0
```

3.6 rosparam load

load <yaml-file> [namespace]

Load parameters from a YAML file into the specified [namespace] (defaults to /). NOTE: this will *add* to the currently set values.

```
$ rosparam load dump.yaml
```

-v

Show verbose output.

4. Roadmap

rosparam is a stable command-line tool within the ROS core toolchain. No major feature development is currently scheduled for this tool.

5. See Also

- roscpp parameter syntax
- rospy parameter syntax