rosservice

Wiki
Distributions
ROS/Installation
ROS/Tutorials
RecentChanges
rosservice

Page
Immutable Page
Info
Attachments
More Actions:

User
Login

Package Links
Code API
Tutorials
FAQ
Changelog
Change List
Reviews

Dependencies (6)
Used by (10)
Jenkins jobs (10)

Package Summary

Released | Continuous Integration: 1371 / 1372 | Documented

rosservice contains the rosservice command-line tool for listing and querying ROS Services. It also contains a Python library for retrieving information about Services and dynamically invoking them. The Python library is experimental and is for internal-use only.

- 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. See also

The rossrv tool provides information about srv files.

2. rosservice command-line tool

The rosservice command implements a variety of commands that let you discover which services are currently online from which nodes and further drill down to get specific information about a service, such as its type, URI, and arguments. You can also call a service directly from the command line.

The currently supported commands are:

```
rosservice call call the service with the provided args
rosservice find find services by service type
rosservice info print information about service
rosservice list list active services
rosservice type print service type
rosservice uri  print service ROSRPC uri
```

2.1 rosservice args

args <service-name>

Print the arguments to a service.

```
$ rosservice args /service_name
```

2.2 rosservice call

call <service-name> [service-args]

Call a service from the command line.

```
$ rosservice call /service_name service-args
```

e.g.

```
$ rosservice call /add_two_ints 1 2
```

--wait (*New in ROS 1.1*)

Wait until service is advertised before calling.

2.2.1 YAML syntax

Please see YAML command line for a detailed description and examples of how to specify service arguments to call.

2.2.2 Negative numbers

Please see YAML command line for a detailed description and examples of how to call rosservice with negative-number arguments.

2.3 rosservice find

find <service-type>

Display all services of a particular type.

```
$ rosservice find rospy_tutorials/AddTwoInts
```

2.4 rosservice list

list

List all the services that are currently available.

```
$ rosservice list
```

list <namespace> (ROS 0.11)

List all services in the specified namespace.

```
$ rosservice list /rosout
```

-n

Include the name of the node that implements the service.

```
$ rosservice list -n
```

2.5 rosservice info (ROS 0.11)

info <service-name>

Print information about specified service.

```
$ rosservice info /rosout
```

2.6 rosservice node

node <service-name>

Display the name of the node that provides a particular service.

```
$ rosservice node /service_name
```

2.7 rosservice type

type <service-name>

Display the type of a service.

```
$ rosservice type /service_name
```

This is useful in combination with rossrv. For example, to show the srv file that defines a service:

```
$ rossrv show `rosservice type /service_name`
```

or with a shell pipe:

```
$ rosservice type add_two_ints | rossrv show
int64 a
int64 b
---
int64 sum
```

2.8 rosservice uri

uri <service-name>

Display the URI of a service. This is useful, for example, if you wish to know what address a service is using.

```
$ rosservice uri /service_name
```

3. Roadmap

rosservice is a stable command-line tool within the ROS core toolchain. It's currently feature set is not expected expand much. Currently, the only major feature planned is the ability to use YAML text files as well as piped YAML input with the rosservice call command. This feature is currently not scheduled.