

ROS Filesystem Concepts (/ROS/Concepts): *Packages (/Packages) | Metapackages (/Metapackages) | Manifest (/Manifest) | msg (/msg) | srv*

For rosbuilt, see: rosbuilt/srv (/rosbuilt/srv)

Contents

1. Command-line Tools
2. Service Description Specification
3. Building .srv Files
4. Client Library Support

ROS uses a simplified service description language ("srv") for describing ROS service (/Services) types. This builds directly upon the ROS msg format (/msg) to enable *request/response* communication between nodes (/Nodes). Service descriptions are stored in .srv files in the srv/ subdirectory of a package (/Packages). You should familiarize yourself with the msg format (/msg) before attempting to write .srv files.

Service descriptions are referred to using package resource names (/Names). For example, the file robot_srvs/srv/SetJointCmd.srv is commonly referred to as robot_srvs/SetJointCmd.

1. Command-line Tools

rossrv (/rossrv) prints out service descriptions, packages that contain .srv files, and can find source files that use a service type.

2. Service Description Specification

A service description file consists of a *request* and a *response* msg type (/msg), separated by ' - - - '. Any two .msg files concatenated together with a ' - - - ' are a legal service description.

Here is a very simple example of a service that takes in a string and returns a string:

```
string str
---
string str
```

We can of course get much more complicated (if you want to refer to a message from the same package you must not mention the package name):

```
#request constants
int8 F00=1
int8 BAR=2
#request fields
int8 foobar
another_pkg/AnotherMessage msg
---
#response constants
uint32 SECRET=123456
#response fields
another_pkg/YetAnotherMessage val
CustomMessageDefinedInThisPackage value
uint32 an_integer
```

You **cannot** embed another `.srv` inside of a `.srv`.

3. Building .srv Files

See: [catkin/CMakeLists.txt#msgs_srvs_actions \(/catkin/CMakeLists.txt#msgs_srvs_actions\)](#)

4. Client Library Support

In Python, the generated Python service type file (e.g. `foo_srvs.srv.Foo`) provides nearly all the information you might want about a `.srv` file. You can examine the `__slots__` and `_slot_types` and other fields to introspect information about the request and reply Messages. For *advanced* users, the `roslib.srvs` module in the `roslib (/roslib)` Package provides basic support for parsing `.srv` files in Python. This library is only recommended for those implementing service generators.

Except where otherwise noted, the ROS wiki is
licensed under the

Wiki: `srv` (last edited 2017-01-07 16:54:04 by IsaacSaito (/IsaacSaito))

Creative Commons Attribution 3.0

(<http://creativecommons.org/licenses/by/3.0/>) | Find us on Google+

(<https://plus.google.com/113789706402978299308>)

Brought to you by:  Open Source Robotics Foundation

(<http://www.osrfoundation.org>)