Using rosed to edit files in ROS

**Description:** This tutorial shows how to use [rosed](http://wiki.ros.org/rosbash) to make editing easier.  
  
**Tutorial Level:** BEGINNER  
  
**Next Tutorial:**  [Creating a Msg and Srv](http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv) 

**Contents**

1. [Using rosed](http://wiki.ros.org/ROS/Tutorials/UsingRosEd#Using_rosed)
2. [Using rosed with tab completion](http://wiki.ros.org/ROS/Tutorials/UsingRosEd#Using_rosed_with_tab_completion)
3. [Editor](http://wiki.ros.org/ROS/Tutorials/UsingRosEd#Editor)

Using rosed

rosed is part of the [rosbash](http://wiki.ros.org/rosbash) suite. It allows you to directly edit a file within a package by using the package name rather than having to type the entire path to the package.

Usage:

$ rosed [package\_name] [filename]

Example:

$ rosed roscpp Logger.msg

This example demonstrates how you would edit the Logger.msg file within the roscpp package.

If this example doesn't work it's probably because you don't have the vim editor installed. Please refer to [Editor](http://wiki.ros.org/ROS/Tutorials/UsingRosEd#Editor) section. If you don't know how to get out of vim, [click here](http://kb.iu.edu/data/afcz.html).

If the filename is not uniquely defined within the package, a menu will prompt you to choose which of the possible files you want to edit.

Using rosed with tab completion

This way you can easily see and optionally edit all files from a package without knowing its exact name.

Usage:

$ rosed [package\_name] <tab><tab>

Example:

$ rosed roscpp <tab><tab>

* Empty.srv package.xml
* GetLoggers.srv roscpp-msg-extras.cmake
* Logger.msg roscpp-msg-paths.cmake
* SetLoggerLevel.srv roscpp.cmake
* genmsg\_cpp.py roscppConfig-version.cmake
* gensrv\_cpp.py roscppConfig.cmake
* msg\_gen.py

Editor

The default editor for rosed is vim. The more beginner-friendly editor nano is included with the default Ubuntu install. You can use it by editing your ~/.bashrc file to include:

export EDITOR='nano -w'

To set the default editor to emacs you can edit your ~/.bashrc file to include:

export EDITOR='emacs -nw'

***NOTE:*** *changes in .bashrc will only take effect for new terminals. Terminals that are already open will not see the new environmental variable.*

Open a new terminal and see if EDITOR is defined:

$ echo $EDITOR

* nano -w

or

emacs -nw