

Experience in using ROS2 under Windows

N. Oakes and L. Hetherton

*CSIRO Data61, Australia
Email: nerolie.oakes@data61.csiro.au*

Abstract: ROS (Robot Operating System) is a software development kit for robotics applications (<https://www.ros.org>). Originally conceived in 2007, it was first released as an open-source (Apache 2.0) package distribution by the Open Source Robotics Foundation in 2013 (<https://www.theconstructsim.com/history-ros>). ROS quickly became popular, and there were many third-party open-source modules released. However, by 2015, it was becoming apparent that ROS had some fundamental deficiencies in areas such as security and robustness that made it unsuitable for use in commercial applications. This was becoming a deterrent to its use in industrial applications. As a result, there was a concerted effort to develop ROS version 2 (ROS2), which addressed these problems. The first distribution release of ROS2 was made in 2017. ROS2 is complex and is designed to work closely with computer operating systems and libraries. It is continually being upgraded and understandably can only be tested under a limited set of options. In 2021, under the auspices of its Future Digital Manufacturing initiative, CSIRO started investigating replacing and expanding existing software with ROS2 modules, some of which were for projects delivered into Windows-based software ecosystems.

In attempting to employ ROS2 on a specific Windows-based software project, we found that while the ROS project had extensive documentation and third-party modules already developed in the areas of interest, much of the documentation still referred to incompatible ROS modules and most of the third-party modules had been developed under Ubuntu (Linux). This led to a very frustrating development process, where ROS2 documentation often contained links to incompatible ROS documentation; fixes were made to modules for problems under Ubuntu that were untested on (and incompatible with) Windows or alternatively, fixes were made to branches but these were not passed through to the released versions for months. When the project commenced in 2021, the current release version was ROS “Galactic”. We encountered difficulties during the installation process, in running the tutorials and in using third-party modules referenced in the tutorials. Some of the problems were anticipated in the installation instructions, but most were solved only through a time-consuming combination of online search and trial-and-error.

Examples of installation complications are: an outdated version of Python needed to be installed at a particular location; an outdated version of VC++ (2013) needed to be installed; an unavailable version of OpenSSL was required (another was found that did work); and a particular dummy directory needed to be created before installation could proceed. The extensive tutorials suffered from lack of testing under different environments resulting in listed environment variables being incorrect for a binary installation, and another dummy directory needing to be created before the tutorials would run. Similarly, the instructions for configuring communication between nodes under the default Cyclone Data Distribution Service (DDS) on different computers was incomplete, and it was only possible to communicate using a configuration file that explicitly listed the networked computers as peers – an undesirable outcome for the project in question.

The project was expected to make use of many third-party modules, however, the ROS2 tutorials often led to incompatible ROS packages. If compatible packages were located, configuration files might contain changes that addressed problems under Ubuntu but created problems under Windows. These had often been identified and fixed in source branches but not passed through to the distributed binary package. When the latest source code was compiled, the environmental setup failed. A partially successful (but far from ideal) compromise was reached by pasting patched files from branches on to the original released binary installation.

Our experience was that using ROS2 under Windows led to a frustrating level of issues. A key challenge for the ROS2 community, if ROS2 adoption is to grow unhindered, is to better support this ubiquitous operating system on which many research and commercial ventures depend. Doing so appears to be within the remit of the recently launched Australian initiative ROSella (<http://rosella.org>), and this would be a welcome step.

Keywords: *ROS2, robotics*