Develop ROS™ applications with Visual Studio Code and Azure

Gunter Logemann, Senior Software Engineer
Our ROS for Windows Journey
Our ROS Journey

- **Jun 2018**: ROSCON 2018
- **Sep 2018**: Experimental Release of ROS on Windows
- **May 2019**: Microsoft Build 2019
- **Jun 2018**: General Availability of ROS on Windows
- **Oct 2019**: ROSCON 2019
- **Jul 2019**: Developer flow using VS Code and Azure
- **Oct 2019**: Preview of VS Code extension for ROS

---

**Images:**
- ROS on Windows 10 IoT
- Autonomous System
Azure IoT Hub Connector ROS node
Azure Kinect ROS Driver
Windows ML Tracker ROS node
Azure IOT_Hub connector

Azure IoT Hub Connector ROS node

These samples demonstrate how to use the various features of Microsoft Azure IoT Hub service to relay telemetry messages or dynamic reconfiguration commands in ROS nodes.

- **IOT Hub connector sample in C++**
- **IOT Hub connector sample in Python**
- **Dynamic reconfiguration sample**
Azure Kinect ROS Node

Azure Kinect ROS Driver

- A PointCloud2, optionally colored using the color camera
- Raw color, depth and infrared Images, including CameraInfo messages containing calibration information
- Rectified depth Images
- Rectified color Images
- The IMU sensor stream
- A TF2 model
Windows ML Node

Windows ML Tracker ROS node

- ROS node which uses Windows Machine Learning (WinML) to track people (or other objects) in camera frames. Uses camera frames from the OpenCV based cv_camera node at /cv_camera/image_raw
- Outputs an image with the ID boxes for tracked objects overlayed on the source frame at /tracked_objects/image
279 ROS packages enabled on Windows

~1,800 downloads of ROS Melodic monthly

From 6 Continents across the world

Over 14,000 installs of ROS extension since July
Visual Studio Code extension for ROS
Visual Studio Code

Free, cross-platform, open source
Fast and lightweight
Rich extension ecosystem

“In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents claiming to use it.” - Wikipedia
ROS Extension

- Automatic ROS workspace activation.
- Allows starting, stopping and viewing the ROS system status.
- Automatically discover build tasks.
- 3D preview URDF and XACRO files.
- Debug ROS nodes (C++ or Python) by attaching to the process or from the ROS launch.
- And more
URDF Preview

A tool built on top of Robot Web Tools
Visualize URDF\XACRO files in seconds
ROS2 Support

- Workspace Discovery
- ROS2 Daemon Start\Stop
- ROS2 Status Monitor
- Colcon Build Tool Integration
Getting Started
Pre-requisites

Install ROS/ROS2 (Melodic and Dashing) on Windows (http://aka.ms/ros)

Install C\C++ Toolchain (depending on your platform)
   GNU GCC on Linux\MacOS
   Visual C++ Workload on Windows

Install Visual Studio Code & ROS Extension
ROS on Azure with Windows VM

Look for “ROS” on aka.ms/azure/quickstart

This template creates a Windows VM and installs the ROS into it using the CustomScript extension.
ROS on Azure with Windows VM

Let’s walk through the template
What’s next?

Add debug support for ROS2 Launch system
  Plus ROS and ROS2 coexistence environment
More improvement for cross-platform development
More visualization tooling
End-to-end DevOps workflow
Call to Action

Try it out aka.ms/ros/vscode
Give us feedback on github.com/ms-iot/vscode-ros
Learn more about ROS on Windows IoT and Azure aka.ms/ros
Demo: A simple ROS application