

ROS.NET Unity for Multiplatform Applications

Jordan Allspaw
Carlos Ibarra Lopez
Eric McCann



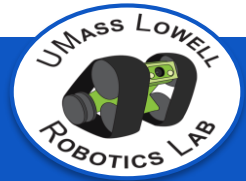
What is ROS.NET?

- A series of C# projects that allow a MANAGED .NET application to communicate traditional ROS nodes



Standard ROS Features

- Publishers & Subscribers
- Parameters
- Services
- Custom Messages



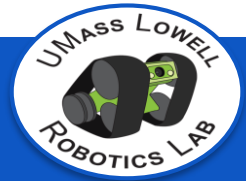
Tools/Samples

- Compressed Image View
- Point Cloud and LaserScan Visualization
- URDF / Robot Model
- Dynamic Reconfigure
- TF Tools



ROS.NET Syntax

```
4 using Ros_CSharp;
5
6 public class samplepublisher : MonoBehaviour {
7     public ROSCore rosmaster;
8     private Publisher<Messages.std_msgs.String> pub;
9
10
11     void Start () {
12         NodeHandle nh = rosmaster.getNodeHandle();
13         pub = nh.advertise<Messages.std_msgs.String>("/chatter", 10);
14     }
15
16     void Update () {
17         Messages.std_msgs.String msg = new Messages.std_msgs.String();
18         msg.data = "Welcome to ROSCON!";
19         pub.publish(msg);
20     }
21 }
```



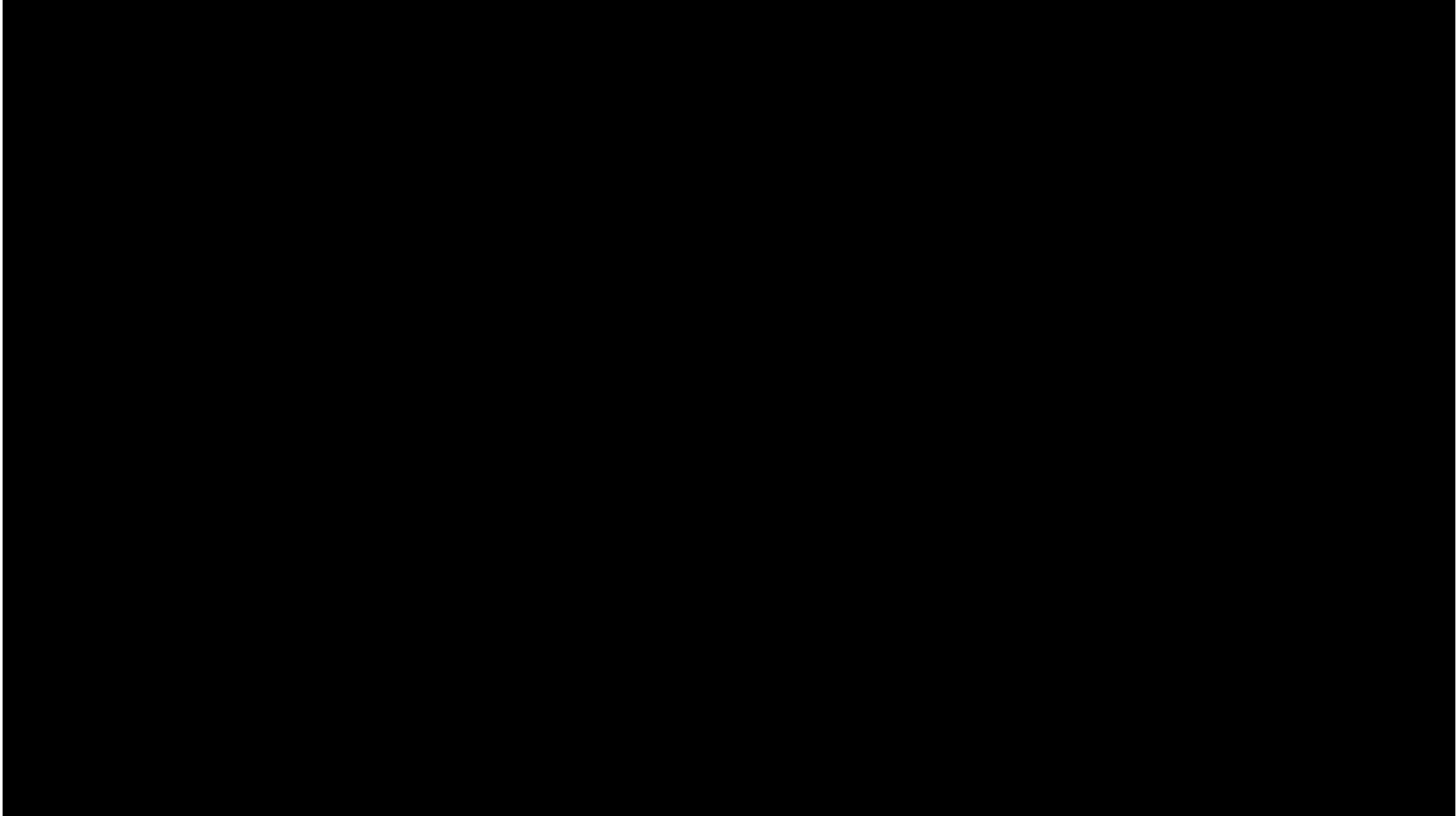
Why use Unity (and Windows)

- Some devices only support Windows
- Support for Windows, Linux, Mac OS, Android, IOS
- Quickly utilize 3D graphic capabilities



Sample Usage with HTC Vive





UMass Lowell Robotics Lab
robotics.cs.uml.edu

University of Massachusetts Lowell
www.uml.edu



Thanks!

<https://github.com/uml-robotics/ROS.NET>

https://github.com/uml-robotics/ROS.NET_Unity

