

The image features a woman in the center wearing a black Xsens headband and a black long-sleeved shirt with the Xsens logo. She is holding a green bottle. To her left is a white and black humanoid robot, and to her right is a white humanoid robot with orange joints. The background is black with large, stylized orange 'X' shapes.

xsens

13 April 2026

Xsens ROS2 driver integration guide



Our newly developed Xsens ROS2 drivers are now available for download. There are two pathways to complete this integration using two different nodes, both of which have different use cases. This guide covers a basic integration between our Xsens IMU data and a robotics interface.

Our ROS2 driver repository includes an entire ROS2 Workspace with multiple packages. Upon downloading, it will include:

Package	Description
xsens_mvn_ros2	Main package with both ROS 2 nodes, launch files, config, etc.
xsens_mvn_ros2_common	Contains libraries and interfaces for TF broadcasting, topic publishing, and dependencies
xsens_mvn_msgs	Custom ROS 2 message definitions
xsens_mvn_ros2_description	Contains a dynamic, body-scaled URDF generation node
xsens_mvn_ros2_stream	Libraries for UDP socket and MVN protocol handling
xsens_mvn_ros2_xme	Direct hardware connection via XME SDK

Xsens Hardware Setup:

Before starting your ROS2 integration, the first step will always be to set up your Xsens Hardware. To ensure that this is done correctly, please check out our Getting Started Page:

[Movella | Getting started with Motion Capture](#)

Best Practices:

- For the new Xsens Link, use a wired USB-C to USB-C connection to your computer from the Xsens Link Hub. This reduces latency and avoids dropped packets by bypassing the Wi-Fi router connection.
- For the Xsens Link, use a wired Ethernet connection directly to your computer from the Xsens Link bodypack. This reduces latency and avoids dropped packets by bypassing the Wi-Fi router connection.
- Measure body segment dimensions manually in Xsens instead of using default proportions. This improves tracking accuracy, especially in arm and hand movements. Check our [body measurements guide](#) for more information.
- After completing a calibration, walk/move around for 15-20 seconds to warm up the fusion filters before starting data collection.



Choose your path:

Attribute	xsens_stream_node (Streamer)	xsens_xme_node (SDK)
Data Source	Xsens Humanoids Network Streamer (UDP)	XME library embedded, connects to suit on same Linux PC through SDK
Use Case	Operator on Windows/Linux PC with Xsens application	Embedded/standalone on robot or edge device
Requirements	Xsens Humanoids license	Linux SDK license

Please note that the Streamer Node utilizes easy calibration and actor setup in the Xsens GUI. Our native software has an embedded network streamer that simply publishes data packets over UDP protocol. This node receives the UDP packets in the ROS workspace.

The XME Node can be directly embedded into a client's own applications or for robot-native deployment utilizing our Linux SDK.

Compatibility:

- Ubuntu 24.04 LTS
- ROS2 Jazzy Jalisco
 - o How to install: [Ubuntu \(deb packages\) — ROS 2 Documentation: Jazzy documentation](#)

1. Shared Setup

- Clone the repository in your preferred directory

```
git clone <repository-url>
cd xsens_mvn_ros2
```

- Install dependencies

```
sudo rosdep init # only needed once per system
rosdep update
rosdep install -y -i --from-paths $(colcon list --paths-only)
```

- Build the workspace

```
source /opt/ros/jazzy/setup.bash && \

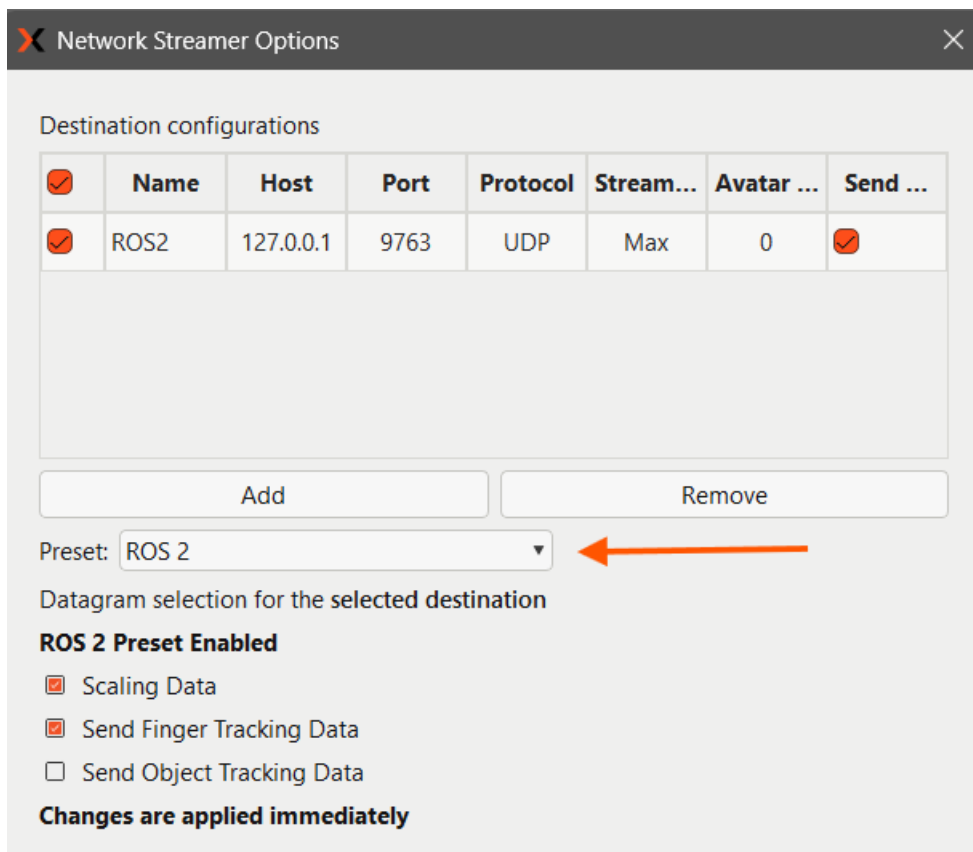
colcon build
```

2. Path A: Xsens_Stream_Node

This node allows you to continue utilizing the Xsens software and GUI. A simplified workflow is as follows:

Open Xsens → Calibrate Suit → Turn on Network Streamer → Launch ROS Driver → Verify ROS Topics

Once the hardware is set up and you have calibrated your suit, please open the Network Streamer (Options → Network Streamer). Then, select the ROS2 preset.



Ensure the UDP port is assigned to **9763**. This is the default port the node is listening to. If necessary, this preset can be changed within the Streamer Node parameters.

Launch Driver:

```
ros2 launch xsens_mvn_ros2 xsens_stream.launch.py
```

Configure:

```
src/pkgs/xsens_mvn_ros2_stream/config/xsens_stream_node.yaml
```

Linux OS:

The simplest way to complete this setup is to utilize the Linux version of our Xsens Humanoids software. This would allow you to run a ROS workspace alongside our software on the same computer, without needing to dual boot or run a Linux Kernel.

To successfully download the Linux version of our software, please follow these instructions:

<https://base.xsens.com/s/article/Xsens-Linux-Install>

Windows OS:

Our software was originally built with Windows OS in mind. However, working on a Windows computer can make it difficult to stream to a ROS2 node operating on a Linux OS.

It is possible to complete this pipeline using a dual-boot system or a Linux Kernel. For example, you can use Windows Subsystem for Linux (WSL) to run ROS on a Windows machine.

Check the guides (Guide 1; Guide 2) or follow the steps below.

How to set up WSL for ROS:

- In Windows Powershell use the command "wsl -- install"
- Install WSL: wsl.exe --list --online
- Install Ubuntu 20.04: wsl.exe--install Ubuntu-20.04
- Set up your user (note: password entry is invisible)
- Install ROS Jazzy using the ROS-Wiki guide
- Choose ROS-BASE (desktop version) to access Rviz
- Install GIT library and catkin_tools following the Setup Guide. Pay specific attention to the "Requirements" section.

Known challenge: UDP streaming doesn't work out of the box. MVN Network Streamer cannot send UDP packets directly into the WSL instance. This blocks live data flow to ROS inside WSL.

Workaround for UDP forwarding

- Install Nmap or an alternative UDP forwarder
- Activate your UDP forwarder through a PowerShell terminal, communicating from your desired UDP port to your WSL IP address. Please ensure to check your WSL2 IP each time you restart WSL, as it can change.

3. Path B: Xsens_Xme_Node

This node allows for direct interfacing with our SDK, giving control over the motion capture and data management to the customer. It does require access to our SDK License (XME-DEV).

With the lack of our normal Xsens GUI for immediate feedback, this node comes with diagnostic commands. These commands allow you to check up on some of the basic status levels of the suit. To monitor, use command:

```
ros2 topic echo /diagnostics
```



It is important to note that the use of our SDK requires a customer's own development and coding. It is a compilation of various scripts and examples that can be used to create a customer's own application. For more information on the SDK, please visit this article:

[Xsens SDK Download and reference documentation](#)

Once you download the SDK from our website, you can investigate our typical workflow and full SDK documentation here:

C:\Program Files\Xsens\MVN SDK 2026.0\Documentation\html

From there, select one of the Microsoft Edge HTML documents. For example, "class_xme_callback".

Within Command Prompt, you will load your subject's measurements, perform a calibration, and begin a recording.

Launch:

```
ros2 launch xsens_mvn_ros2 xsens_xme.launch.py
```

Configure:

```
src/pkgs/xsens\_mvn\_ros2\_xme/config/xsens\_xme\_node.yaml
```

```
config/body\_dimensions.yaml
```

For more information regarding these two nodes, their parameters, the SDK configurations, default settings, and diagnostic commands, please check out the [README documents](#).

4. Visualization/Verification

This package contains a URDF Publisher Node. This provides dynamic URDF generation that is scaled to the body measurements of the Xsens Avatar.

RViz and the URDF publisher are automatically included in the python launch files. You can use these visualization tools to verify the pipeline when `launch\_description:=true`.

Configure:

```
src/pkgs/xsens_mvn_ros2_description/config/xsens_urdf_publisher_node.yaml
```

Additionally, you can run the following tests to verify the installation of the nodes:

```
```bash
Run tests
source install/setup.bash
colcon test
colcon test-result --verbose
```

```
Check that nodes are available
ros2 pkg executables xsens_mvn_ros2_stream
ros2 pkg executables xsens_mvn_ros2_xme
ros2 pkg executables xsens_mvn_ros2_description
\\`
```

Software	Windows	Linux*
<b>Operating System</b>	Windows 10 (64 bit) Xsens Software 2023.0 and older Windows 11 (64 bit) Xsens Software 2023.2 and up	Ubuntu 22.04 and 24.04
<b>Internet</b>	An active internet connection is required for <a href="#">license</a> activation	An active internet connection is required for <a href="#">license</a> activation
<b>Network</b>	<a href="#">Network requirements</a> for connection with hardware.	<a href="#">Network requirements</a> for connection with hardware.

**\*NOTE:** Older versions of Ubuntu, as well as Debian and Red Hat, have not been tested and are not supported.