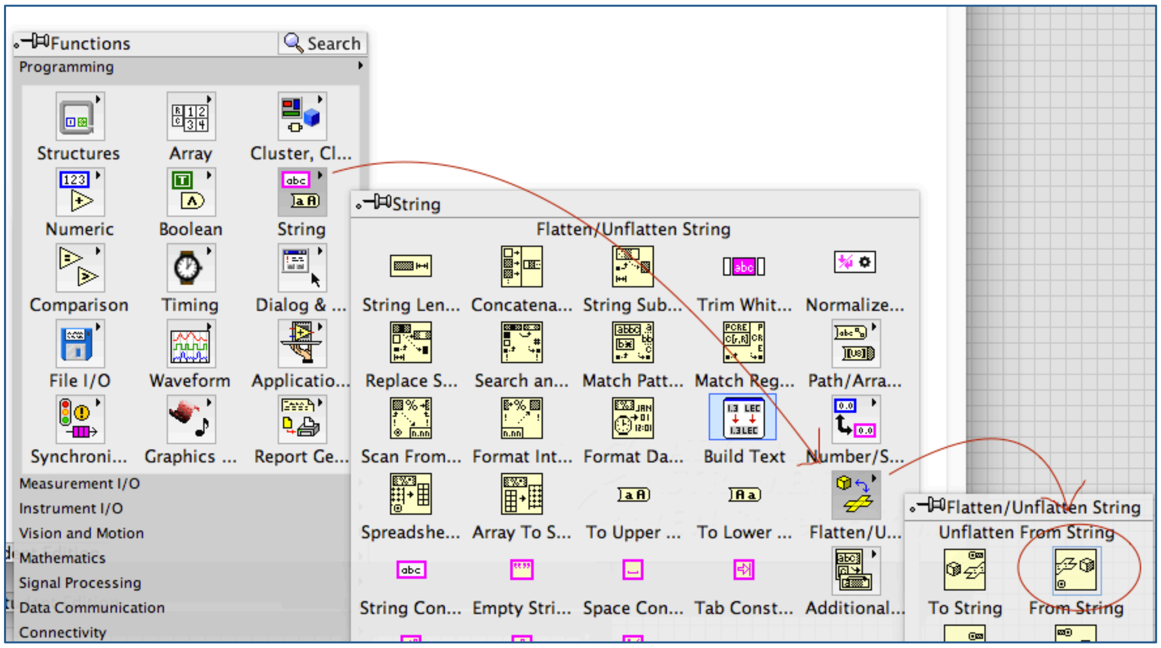
ROS for LabVIEW™ Software Tutorial: Message Parsing

This tutorial will cover how to parse messages in order for them to be understood and executed by a device operating on ROS. Parsing involves unflattening the data string produced by building a message, and taking out the component of the message that matches the desired data type. In this example, we parsed a Boolean. Message parsers are typically used in conjunction with subscribers to read a message returned from a device operating on ROS.

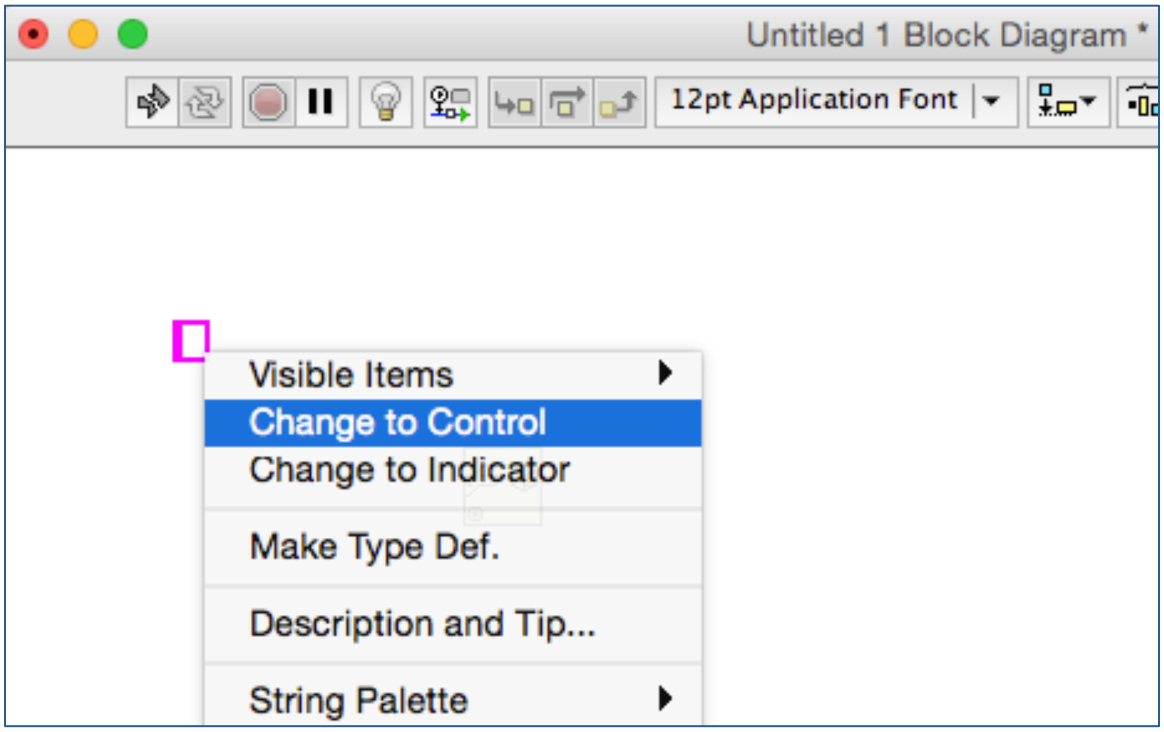
1. Open a new VI in LabVIEW.

2. Drag "Unflatten from String" to the block diagram.



3. Drag a string constant to either side of "Unflatten to String." Left click on each and change the string on the

left to a control and the string on the right to an indicator. Name the control "msg in" and the indicator "msg out."

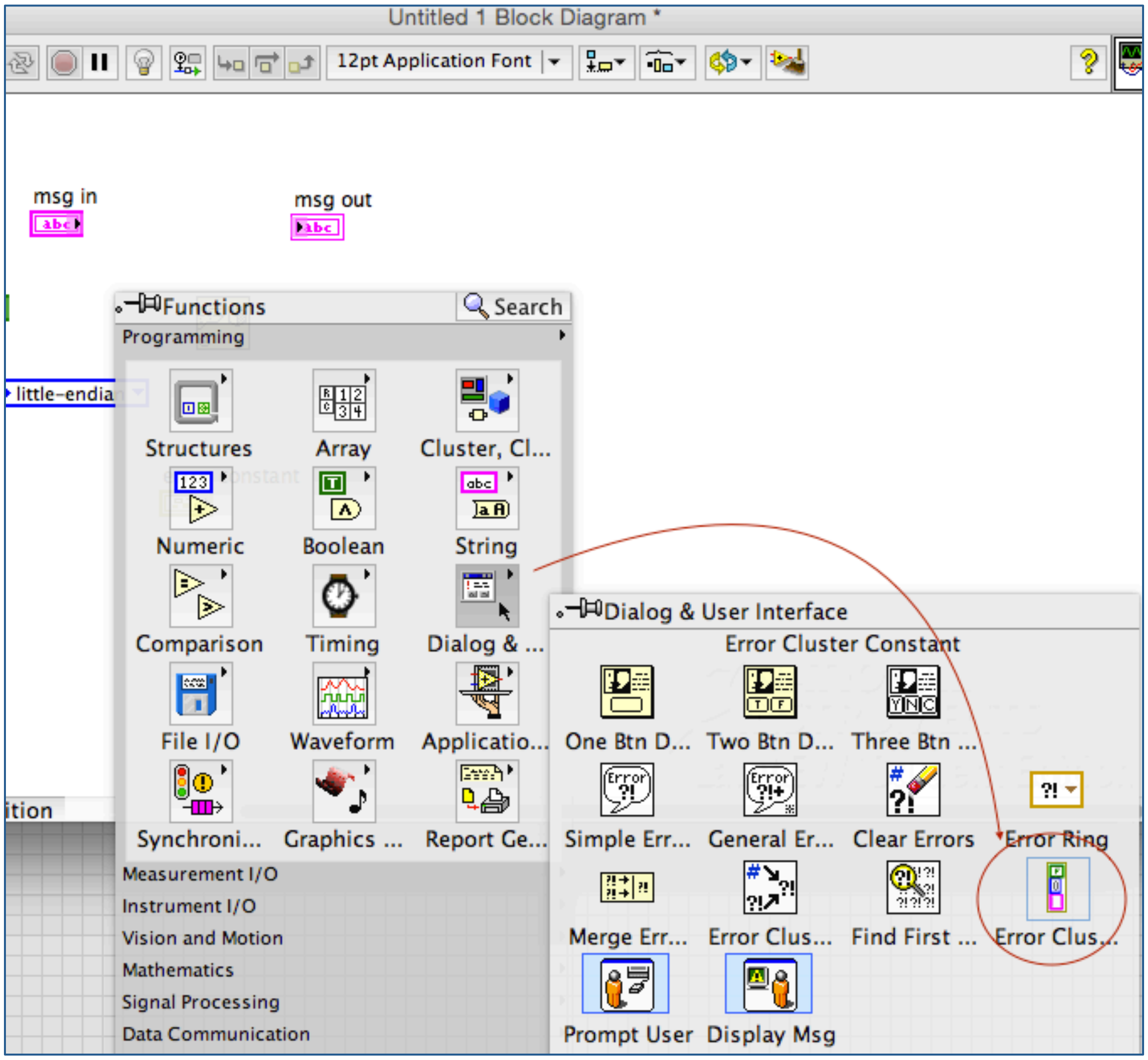


4. Drag a constant of whichever data type you want to parse to the left of Unflatten from String. Left click on it and change it to Control. Label this "type." Drag another constant of the same type to the right of Unflatten from String. Change this to Indicator.

5. Drag an enum constant to the block diagram. Left click and select "Edit Items…" to add the same items as you added to the enum constant in the message builder. You can also simply copy and paste this from the message builder.

6. Drag an error cluster to the block diagram, to the left of Unflatten from String. Left click on the cluster to

change it to Control and uncheck View as Icon. Label this "error in"



7. Wire the type, msg in, error in, and enum into Unflatten from String. Type should wire to terminal "type," msg in to "binary string," error in to "error in," and enum to "byte order."

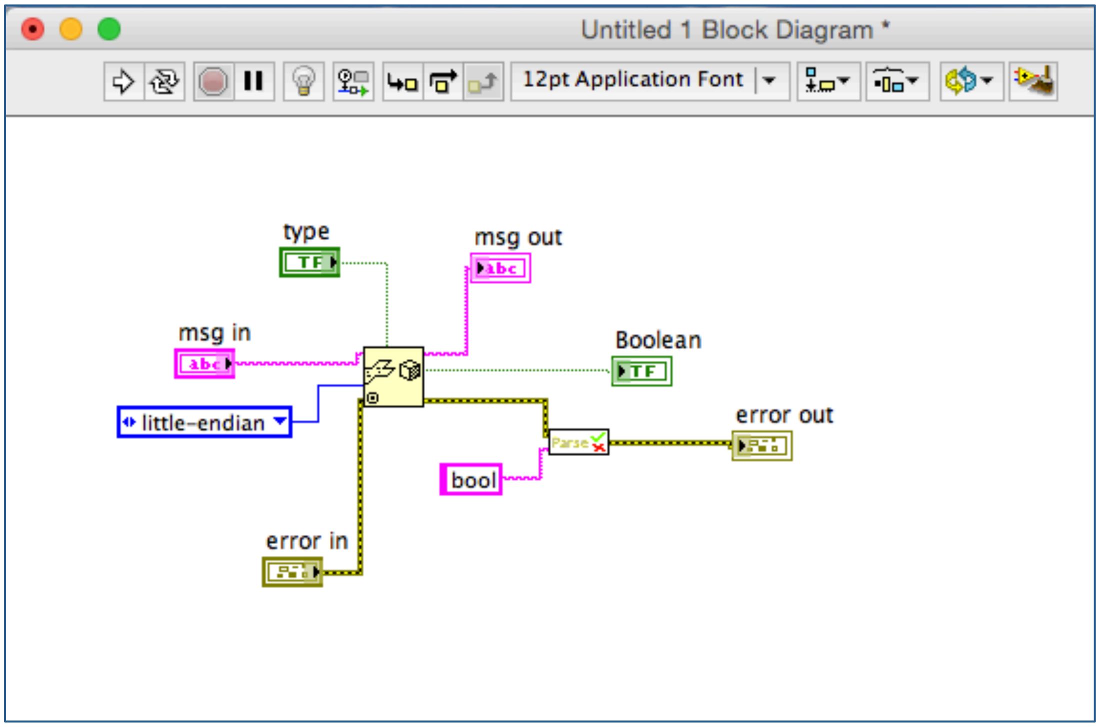
8. Wire the output from the terminal "rest of binary string" to msg out and the output from "value" to the Indicator of whatever data type you are parsing.

9. To the far right of Unflatten from String, drag a parseError sub, found under messageParsing in the ROSforLV menu under user libraries.

10. Drag another error cluster to the right of parseError. Change this to Indicator and label it “error out.” Wire the output from parseError into error out.

11. Drag a string constant to the left of parseError. Into the string type “bool.” Wire this into the input terminal “type” of parseError.

12. Wire the output from terminal “error out” from Unflatten from String into “error in” on parseError to complete your message parser.


Untitled 1 Block Diagram *
12pt Application Font
type
msg out
msg in
Boolean
little-endian
error out
bool
error in