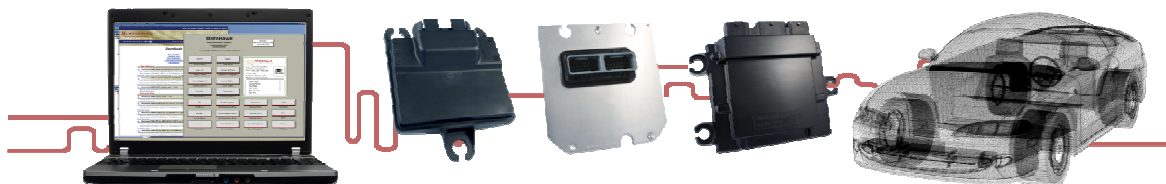




## Documentation

# New Eagle Network Toolbox

User Guide

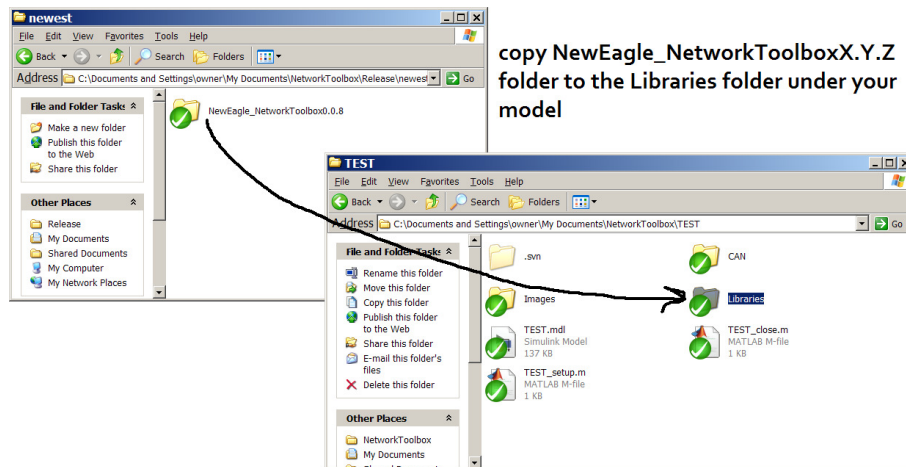


## Introduction

New Eagle's Network Toolbox makes it easy to create CAN input and output blocks in MotoHawk. Instead of manually writing code in MATLAB to handle CAN messaging, Network Toolbox enables you to have useful CAN blocks starting from an industry-standard .dbc file to describe the CAN network. This saves development and debugging time and reduces complexity.

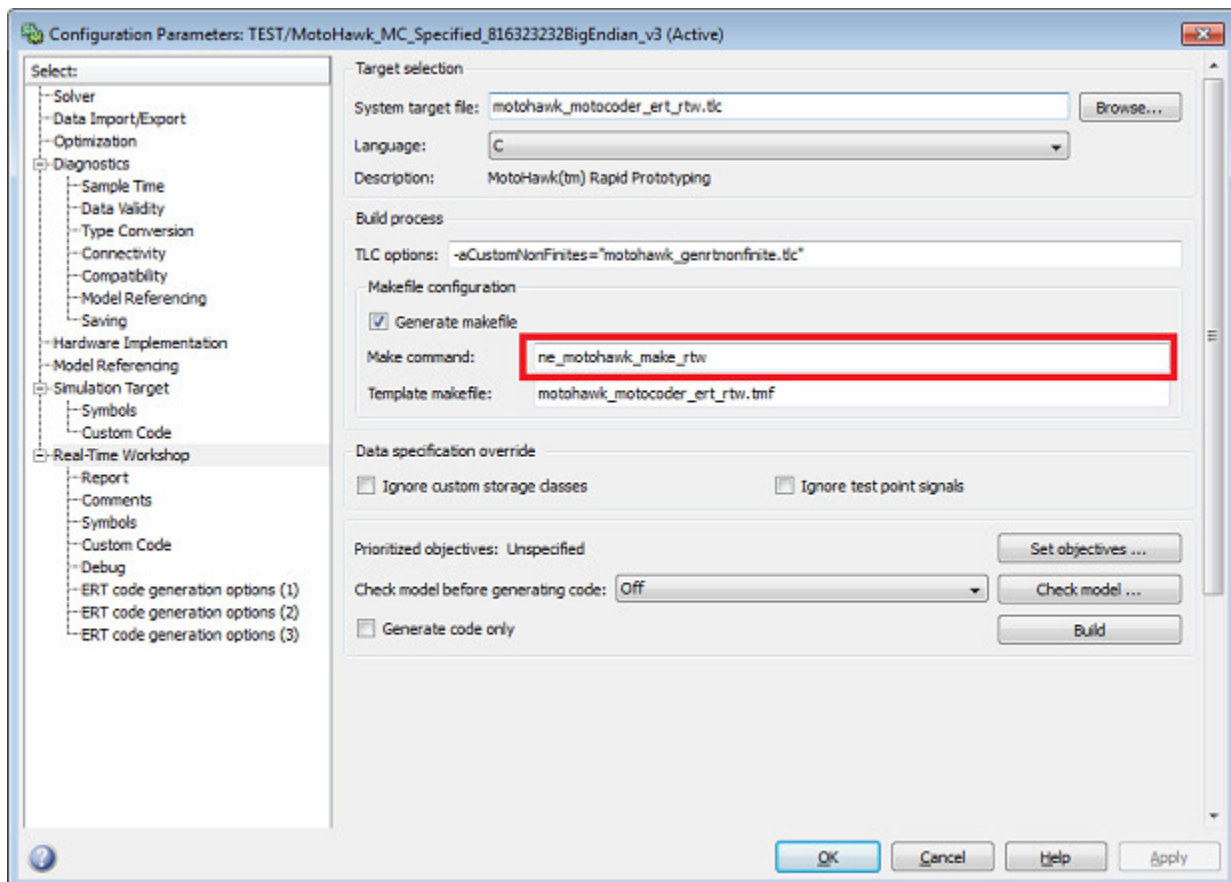
## Installation

To use the Network Toolbox in your MotoHawk projects, just copy the NewEagle\_NetworkToolboxX.Y.Z directory into your Libraries directory under the directory in which your project resides. In order for MATLAB to find the Network Toolbox libraries, make sure your working path is set to your model's directory. You can use the CAN input and output blocks Network Toolbox provides by browsing to New Eagle Network Toolbox in the Simulink Library Browser.

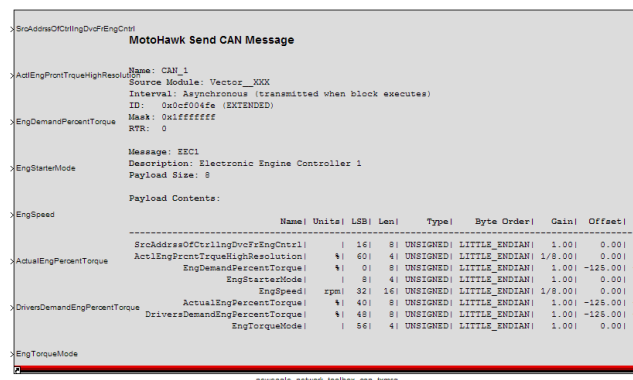


If using the ECM-5554-112 hardware, you will need to update your RTW make command in your model's configuration. To do this in Simulink select Tools->Real Time Workshop->Options...

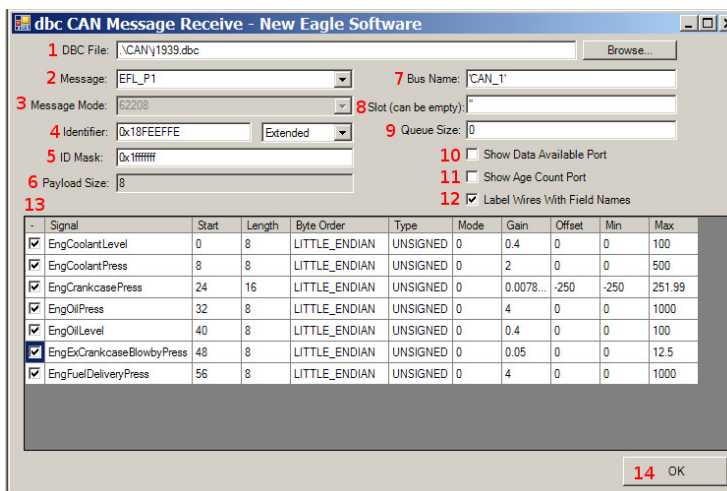
Enter 'ne\_motohawk\_make\_rtw' for the 'Make Command'.




## Usage



Network Toolbox block as it appears within a model.



1. DBC File: You can browse to a .dbc file or enter a path to it. Relative paths will be used where possible to avoid breaking the model if it is copied to a different folder.
2. Message: Defined in the .dbc file. Selecting one message here loads its constituent signals in the signal browser.
3. Message mode: For multi-modal messages, selects the mode that the message will be sent in.
4. Identifier: The CAN bus ID that the message will be sent with.
5. ID Mask: Indicates which bits are relevant for a receive slot. Usually this is 0x1FFFFFFF for an extended CAN ID, or 0x7FF for a standard CAN ID.
6. Payload size: Size of the data to be sent, in bytes.
7. Bus name: CAN bus which the message will be sent or received on.
8. Slot: The slot name can be used to connect this block to a slot that can then be used to dynamically adjust the ID, ID Mask, Payload, and Payload Mask values. In a Send CAN Message block, this text box is the transmit rate in milliseconds. A transmit rate of -1 means asynchronous: transmit whenever the block executes.

- 
9. Queue size: Sets the size of the block's queue which will hold messages until it executes. A queue size of 0 or 1 will only make the most recent message available. In a Send CAN Message block, this text box is the pacing interval in milliseconds, the time delay between sending messages that belong to a multiple message group.
  10. Show data available port: If checked, shows the port on the block that sends the Data Available signal: 1 if a message is available and 0 otherwise. If the queue size is more than 1, then you can wire this port to a Simulink Do-While block and pop messages out of the queue.  
(valid only for Rx)
  11. Show age count port: If checked, shows the port on the block that sends the age of the data. It will increment every time the block executes and no message is available; you can use the output of this port to feed timeout logic.  
(valid only for Rx)
  12. Label wires with field names: If checked, labels wires entering or exiting the block with the names of the signals they carry.
  13. Signal explorer: Shows the signals made available by the .dbc file you have loaded and the message ID you have selected. If the checkbox at left is checked, the port on the block that makes the signal available to source or sink is shown.

## Helpful Hints

If a CAN block from Network Toolbox presents a "File not found" message when you double-click on it, that is just a warning that the `can_example.dbc` file is not in the project directory. You can just dismiss that warning and go on to select another .dbc file.

If a build fails, and it appears that you do not have the DBC dialog on CAN send/receive blocks, you may need to enter 'ne\_can\_postbuild' at the MATLAB command prompt.  
>>ne\_can\_postbuild