

6 ratings:  **3.66** out of 5 [Rate this Document](#)

NI-CAN Channel and Frame API

Primary Software: Driver Software>>NI-CAN

Primary Software Version: 2.6.1

Primary Software Fixed Version: N/A

Secondary Software: N/A

Hardware: Industrial Communications>>CAN

Problem:

I have noticed that there are two different APIs to use for CAN communication. Which one should I use for my application?

Solution:

The choice of whether to use the Channel API or the Frame API is dependent upon both the needs of your application and your personal preference. The Channel API is favored by many because it provides access to the CAN network in simple physical units that are easy to use. Using this API, the NI-CAN software converts the raw data of the CAN messages to or from the physical units of each channel in the channel configuration. However, because the Frame API provides low-level access to the CAN network, there are certain scenarios where it is recommended over the Channel API. Here are some examples of times when it would be better to use the Frame API:

You are using an application developed with NI-CAN version 1.6 or earlier.

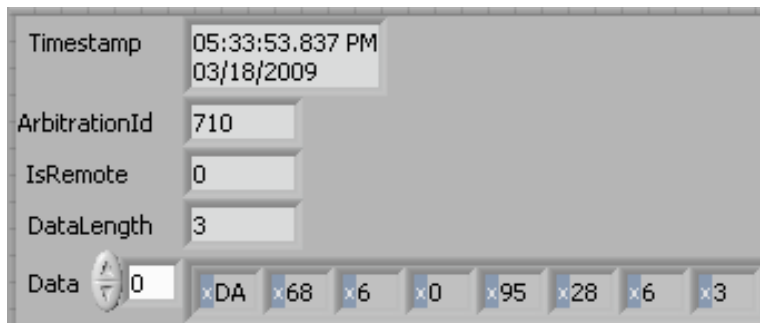
You need to implement a command/response protocol with your device, where you write a command to a device and the device then sends a response.

You need to use remote frames with your device. The Channel API does not support this feature.

You are synchronizing the CAN communication with data acquisition from a DAQ card. The Frame API provides lower-level RTSI features than those provided with the Channel API, and therefore is better for advanced synchronization.

You are using one of the NI USB-847x CAN Interfaces. The USB-CAN products do not support Channel API or CAN Objects. You can still use Frame API but there are some limitations on the functions available for USB-CAN. For a list of these functions, see the Related Links section below.

When receiving messages using the Frame API, you receive the data as an array of unsigned 8 bit characters. Since there are up to 8 bytes of information per message, many messages have multiple channels of information. This allows a single CAN frame to have multiple pieces of information. The Channel API will allow you to automatically parse each channel from a message and scale that channel as necessary.



This image of the Frame API shows the 8 bytes of information. There is no indication of which bytes are paired, which order the bytes should go in, or any scaling information. When configuring the CAN channels in MAX, all of this information can be entered to automatically format the data into the format desired by the user.

CAN Channel Properties

Channel Name:

Start Bit: No. of Bits:

Byte Order:

Data Type:

Scaling Factor:

Scaling Offset:

Minimum Value:

Maximum Value:

Default Value:

Unit:

Comment:

Serial Number (0x7FF):

	7	6	5	4	3	2	1	0
0	6	5	4	3	2	1	0	0
1								
2								
3								
4								
5								
6								
7								

OK Cancel Help

When using the Channel API in LabVIEW, you need to specify that you would like to read the value of Data3 and the read will automatically return the value of the last 12 bits scaled and offset.

NOTE: You cannot use both the Channel API and the Frame API on the same CAN interface at the same time. If one application is running using the Frame API on CAN0, then you cannot run another application using the Channel API also on CAN0. If you have a 2-port CAN card, you can use the separate applications on the different interfaces, or you can run the applications at different times.

Related Links:

[NI-CAN Hardware and Software Manual](#)

[KnowledgeBase 3ECCNE7T: How Do I Synchronize DAQmx with CAN in LabVIEW?](#)

[KnowledgeBase 3LH7HD4B: NI-CAN Self-Reception and Filtering](#)

[KnowledgeBase 46JCHSJL: Which NI-CAN API function calls are supported with my NI USB-847x device?](#)

Attachments:

Report Date: 01/24/2005

Last Updated: 02/20/2012

Document ID: 3HNHF5HS

Your Feedback! ☐ Poor ☐ Excellent

Yes No