

## USER GUIDE

# CAN Frame to Channel Conversion Library

Version 1.0.3

The CAN Frame to Channel Conversion Library is a collection of VIs for fast conversion of CAN frames to scaled channel information. You can use the VIs on Windows Vista/XP/2000, Pharlap ETS, and VxWorks. This allows Compact RIO or USB-CAN users to leverage the use of a database file (.ncd or .dbc) to easily and efficiently convert binary frames to readable channels.

## Contents

---

Introduction .....	2
Installation .....	2
LabVIEW (Windows Vista/XP/2000) .....	2
LabVIEW RT (Pharlap ETS and VxWorks) .....	3
Usage .....	4
Limitations .....	6
Multiplexers/Mode-Dependent Messages .....	6
J1939 or Other Higher Level Protocols .....	6
LabVIEW API Description .....	6
Create Message Table.vi .....	7
Close Message Table.vi .....	9
Add New Channel.vi .....	11
Add Selected Channels from File.vi .....	14
Convert Frame to Channels.vi .....	16
Convert Channels to Frame.vi .....	18
Get Number of Channels.vi .....	20
Get Message Size By ID.vi .....	22
Get Channels.vi .....	24
Get Channel Names By ID.vi .....	26
Select Names By ID.vi .....	28

# Introduction

---

This section describes how to install the CAN Frame to Channel Conversion Library for use with LabVIEW (on Windows Vista/XP/2000) and LabVIEW RT (on Pharlap and VxWorks).

## Installation

Begin by extracting the files to any directory. In that directory are five folders: API, Drivers, Error Codes, Examples, and Remote Target Support.

The API directory contains all VIs required to perform frame and channel conversions. **API.vi** lists the VIs on its block diagram.

### LabVIEW (Windows Vista/XP/2000)

Follow these steps to install the CAN Frame to Channel Conversion Library:

1. Close LabVIEW.
2. Place `frchconvlb.dll` from the Drivers folder in the `system32` directory under your Windows installation directory. This directory typically is `C:\WINDOWS\system32`.



**Note** The Drivers folder contains two library files that LabVIEW can call into, `frchconvlb.dll` for Windows 32-bit editions (Vista/XP/2000) and Compact RIO controllers running Pharlap ETS, and `frchconvlb.out` for Compact RIO controllers running VxWorks.

3. Copy the `API.llb` library file and `frchconvlb.mnu` in the `user.lib` directory to your `<LabVIEW>/user.lib` directory. This library contains all VIs required to perform frame to channel conversions.
4. To see descriptions of the library-specific errors encountered, copy `frchconvlb-errors.txt` in the `user.lib\errors` folder to the `<LabVIEW>/user.lib\errors` directory.
5. Restart LabVIEW.

There is now a new functions palette under **User Libraries** with all the VIs from `API.llb`. **API.vi** lists the VIs on its block diagram.

## LabVIEW RT (Pharlap ETS and VxWorks)

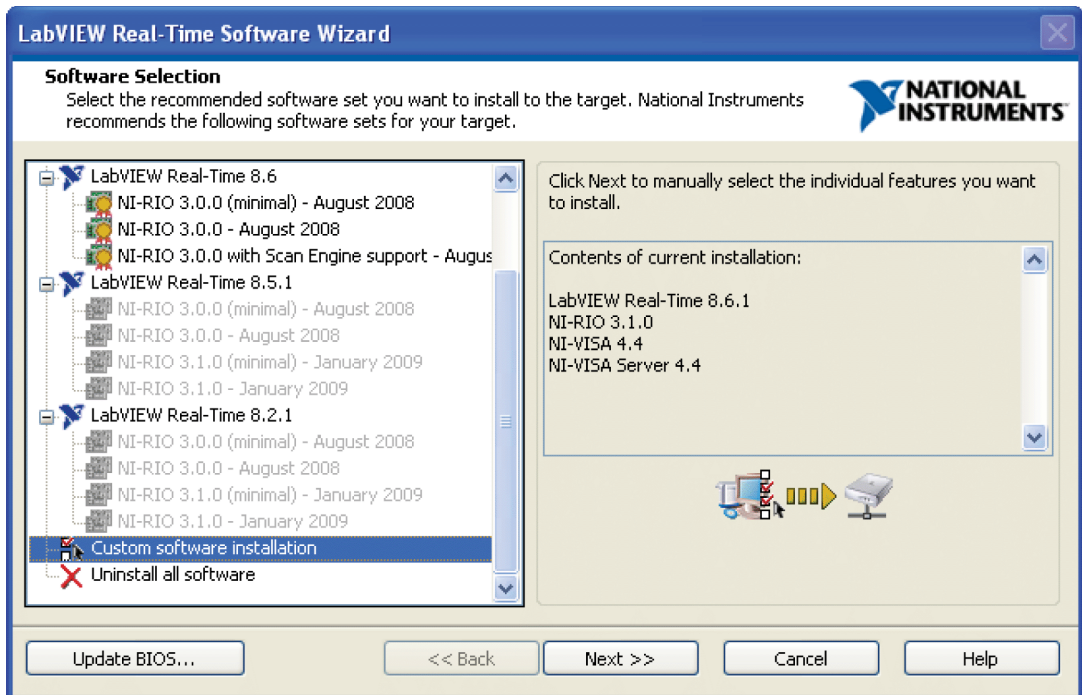
Follow these steps to enable remote target support for the CAN Frame to Channel Conversion Library:

1. Close Measurement & Automation Explorer (MAX).
2. Copy the contents of the Remote Target Support folder to the <National Instruments>\RT Images folder.

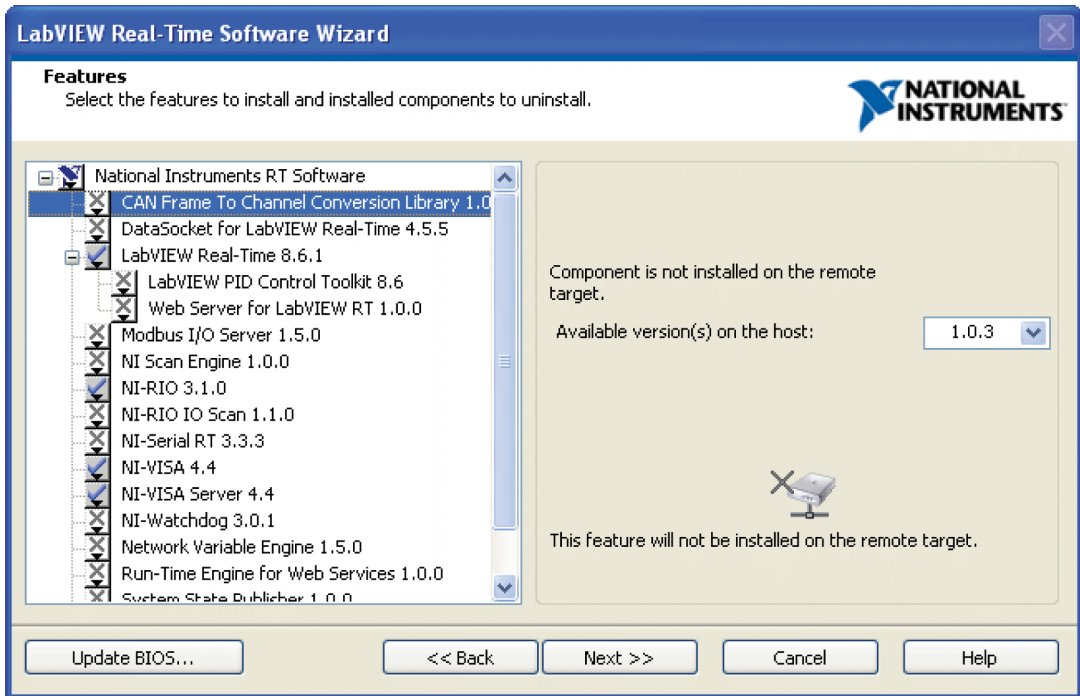


**Note** If you are upgrading from version 1.0.1, you may need to remove the CAN Conversion Library folder from your <National Instruments>\RT Images folder.

3. Start MAX. Right-click **Software** under your RT target and select **Add/Remove Software**.
4. Select **Custom software installation** and click **Next**, as shown below. When a warning appears, click **Yes**.



5. You should see the CAN Frame to Channel Conversion Library, as shown below. Click the **X** and select to install this feature. Click **Next**. When progress is complete and your RT target is rebooted, you can use the CAN Frame to Channel Conversion Library on your RT target.



## Usage



**Note** The library contains several examples to help you get started. These examples are in the examples folder.

Complete these steps to read channels from a database file:

1. Call **Create Message Table.vi** to allocate memory for the message data.
2. Load the channel database information into the table.

To do this, use either **Add Selected Channels from File.vi** or **Add New Channel.vi**. Be careful with these VIs—they can be slow, and if called while performing conversions, they can have unintended side effects. You can use **Add Selected Channels from File.vi** to load all or only a few channels from a file.

### 3. Perform the conversions.



**Note** Once the data has been loaded and a conversion or helper VI has been called, you cannot modify the message table.

Conversion is possible only when all channels are loaded. **Convert Channels to Frame.vi** is a polymorphic VI that converts an array of scaled channel information into either an array of two U32s or eight U8s. You then can attach the array to a CAN cluster or frame as the “data” portion.

**Convert Frame to Channels.vi** is a polymorphic VI that can convert an array of data (either eight U8s or two U32s) into an array of scaled channel information. The order of scaled information is the same as the order in which the channels were loaded.



**Note** These VIs generate a warning if they cannot find a message ID, so ignore their output if a warning is returned.

### 4. Prepare the data for transmission.

You can use a few VIs to prepare data for transmission, by providing information about the messages and channels.

- **Get Number of Channels.vi** provides the number of channels in a particular message.
- **Get Message Size By ID.vi** helps determine the number of bytes used in a message.
- **Get Channels.vi** retrieves all channel names loaded into the message table. This can be a slow process, so you should call this VI once only and reuse the information cluster if needed.
- **Select Names By ID.vi** provides an array of channel names for a particular message ID. Because it uses a linear search algorithm, it is slow if a large number of messages are loaded.

### 5. Clear the memory used for the table using **Close Message Table.vi**.

Always call this VI regardless of any errors that may have been returned. This VI ignores errors on the input, so it is safe to call even in the event of a failure.

## Limitations

The CAN Frame to Channel Conversion Library is *not* a drop-in replacement for the CAN Channel API, supported on Series2 PCI and PXI NI-CAN devices. The Channel API offers much more functionality, with periodic message support, native database support, and MAX integration. The Channel API is directly integrated in the NI-CAN driver. Moreover, the CAN Frame to Channel Conversion Library currently does not support the following items:

### Multiplexers/Mode-Dependent Messages

Parsing of databases that use these types of messages is not supported. To ensure proper parsing, first use MAX to remove multiplexed and mode-dependant messages, then save the channels as an NCD database.

### J1939 or Other Higher Level Protocols

The CAN Frame to Channel Conversion Library interprets only CAN standard and extended frames/messages. Parsing of DBC databases using other higher level protocols, such as J1939, is not guaranteed to work. As a workaround, first import the DBC database into MAX and save it as an NCD database.

## LabVIEW API Description

---

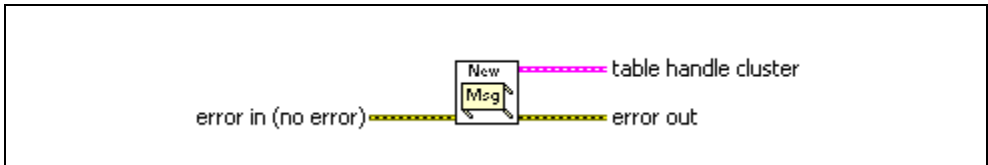
This section describes the CAN Frame to Channel Conversion Library API. The following VIs are in the `API` directory. **API.vi** lists these VIs on its block diagram for convenience.

# Create Message Table.vi

## Purpose

Creates a message table within the library.

## Format



## Inputs



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Outputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is

returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

**Create Message Table.vi** creates a table handle and pre-allocates memory for data conversion. This VI must be called before all other VIs.

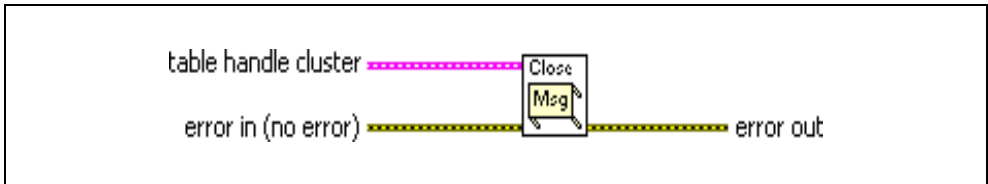


# Close Message Table.vi

## Purpose

Closes a message table created by [Create Message Table.vi](#).

## Format



## Inputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Outputs



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is

returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

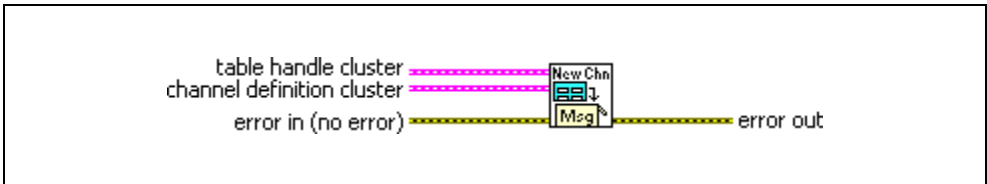
**Close Message Table.vi** closes a message table created by **Create Message Table.vi**. You must call this VI in order to create a new message table. You also must call this VI for memory to be deallocated properly. Otherwise, it will be necessary to shut down LabVIEW to reset. Consider using a stop button to allow your VIs to finish and call this VI.

# Add New Channel.vi

## Purpose

Adds a new user-defined channel to a message table.

## Format



## Inputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**channel definition cluster** is a cluster of information describing the different parameters of a channel required for conversion.



**name** is the name of the channel. This must be unique for the message ID, but not globally.



**units** determines the units that the channel will use.



**start bit** determines the bit that the channel will start with. This number is a value between 0–63.



**number of bits** determines the number of bits that this channel will occupy. This number is a value between 1–64.



**data type** determines the internal data type that represents the scaled channel information. It can be one of three distinctive values: IEEE floating point, Signed Data, or Unsigned Data.



**byte order** determines the byte endianness of the channel data. It will be either Intel or Motorola.



**scaling factor** is the coefficient  $a$  in the linear scaling equation  $a*x + b$ .



**scaling offset** is the constant  $b$  in the linear scaling equation  $a*x + b$ .



**min value** determines the minimum value of the scaled information.



**max value** determines the maximum value of the scaled information.



**message ID** represents the CAN arbitration ID corresponding to the channel. If the message doesn't exist before adding this channel, it will be created.



**number bytes** represents the size of the message to which this channel belongs. If the message doesn't exist before adding this channel, it will be created using this value. Otherwise it will be ignored. Acceptable values range from 1–8 bytes.



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Outputs



**table handle cluster out** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

**Add New Channel.vi** creates a new channel in the message table. If the message ID for the channel does not exist, it will be created when this VI is called. This channel cannot be saved in a database, so ensure that you document the new channel information properly.



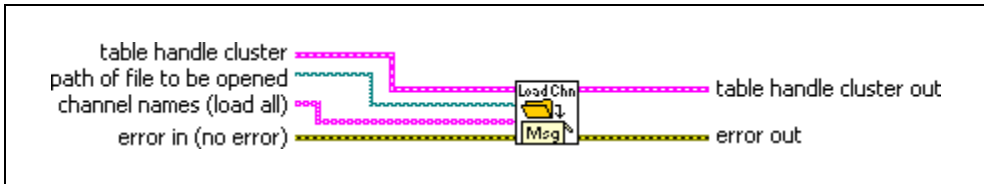
**Caution** This VI allocates memory, which can be a slow process. If a conversion or helper VI has been called before this VI on the same message table, it will cause an error.

# Add Selected Channels from File.vi

## Purpose

Adds channel definitions from a .ncd or .dbc file to a message table.

## Format



## Inputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**path of file to be opened** provides the path to the .dbc or .ncd file to be opened. For VxWorks targets, the path must be on the same device, and all lower case.



**channel names** provides an array of channel name strings to be loaded from the file. This input is optional. If left blank, the VI will load all channels from the file.



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Outputs



**table handle cluster out** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

**Add Selected Channels from File.vi** adds the content of a `.dbc` or `.ncd` file to the message table. The extension must match one of these two formats. If the channel input is left empty, all channels in the file will be loaded.



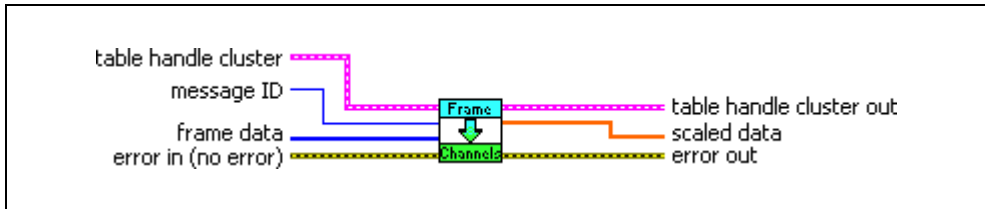
**Caution** This VI allocates memory and performs disk access, which can be slow processes. If a conversion or helper VI has been called before this VI on the same message table, it will cause an error.

# Convert Frame to Channels.vi

## Purpose

Converts CAN frame data to scaled channel data.

## Format



## Inputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**message ID** is the CAN arbitration ID corresponding to the channel.



**frame data** is the data from a CAN frame. It may be an array of two U32s or eight U8s.



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.



## Outputs



**table handle cluster out** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**scaled data** is an array of scaled channel data arranged in the same order in which the channels were loaded.



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

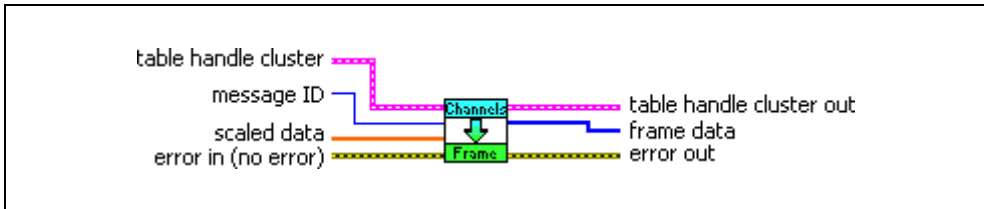
[Convert Frame to Channels.vi](#) is a polymorphic VI that will convert a single frame of data into an array of scaled channel information. The order of the channels will be the same as the order in which the channels were loaded. [Get Channels.vi](#) may help determine the order in which the channels were loaded, and may be used in conjunction with [Select Names By ID.vi](#) to correlate scaled data with channel names.

# Convert Channels to Frame.vi

## Purpose

Converts an array of scaled channel data into an array of CAN frame data.

## Format



## Inputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**message ID** is the CAN arbitration ID corresponding to the channel.



**scaled data** is an array of scaled channel data, arranged in the same order in which the channels were loaded.



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Outputs



**table handle cluster out** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**frame data** is the data from a CAN frame. It may be an array of two U32s or eight U8s.



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

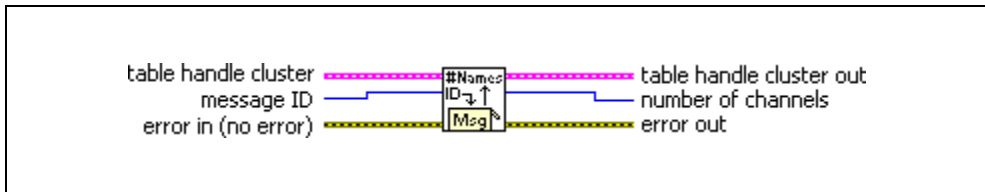
**Convert Channels to Frame.vi** is a polymorphic VI that will convert an array of scaled data into a single frame of data. The order of the array must match the order in which the channels were loaded. **Get Channels.vi** may help determine the order in which the channels were loaded, and may be used in conjunction with **Select Names By ID.vi** to correlate scaled data with channel names.

# Get Number of Channels.vi

## Purpose

Retrieves the number of channels in a particular message.

## Format



## Inputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**message ID** is the CAN arbitration ID corresponding to the channel.



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Outputs



**table handle cluster out** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**number of channels** represents the number of channels in the message with the ID passed in.



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

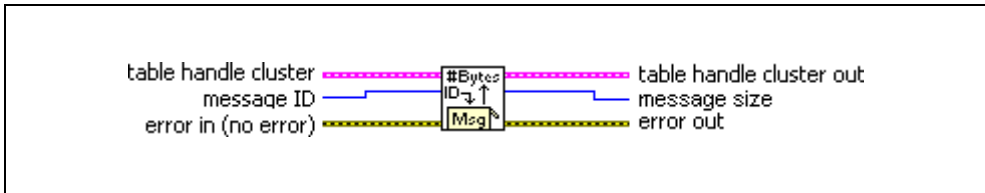
Use [Get Number of Channels.vi](#) to find the number of channels that have been loaded in a particular message.

# Get Message Size By ID.vi

## Purpose

Retrieves the number of bytes in a particular message.

## Format



## Inputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**message ID** is the CAN arbitration ID corresponding to the channel.



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Outputs



**table handle cluster out** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**message size** is the number of bytes in the message with the ID passed in.



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

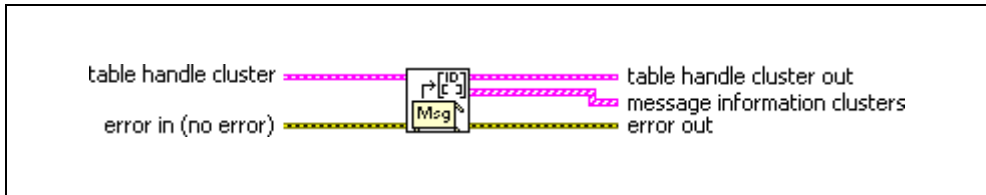
Use [Get Message Size By ID.vi](#) to find the number of bytes in the message with the given ID.

# Get Channels.vi

## Purpose

Retrieves all channel names that have been loaded in the message table, organized by ID.

## Format



## Inputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Outputs



**table handle cluster out** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**message information clusters** represents an array of message information clusters.



**message ID** is the CAN arbitration ID corresponding to the channel.





**message size** represents the number of bytes in the message with the ID passed in.



**channel names** is an array of channel name strings.



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

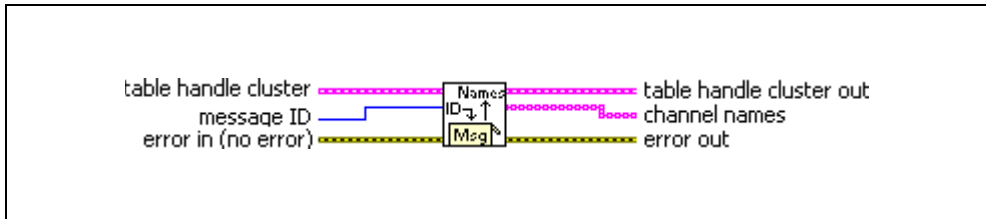
**Get Channels.vi** will retrieve the names of all channels loaded by the driver, clustered with the ID. This can slow the process if called repeatedly. Instead, call it just once and pass the clusters repeatedly to **Select Names By ID.vi**.

# Get Channel Names By ID.vi

## Purpose

Retrieves an array of channel names for a particular message ID from the message table.

## Format



## Inputs



**table handle cluster** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**message ID** is the CAN arbitration ID corresponding to the channel.



**error in** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the **error in** cluster in **error out**.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Outputs



**table handle cluster out** is a cluster containing a table handle as well as pre-allocated space to pass to function calls. You should not modify the arrays contained in this cluster.



**channel names** represents an array of channel name strings.



**error out** describes error conditions. If the **error in** cluster indicated an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.



**status** is True if an error occurred.



**code** is the error code number identifying an error. A value of 0 means success. A negative value means error: VI did not execute the intended operation. A positive value means warning: VI executed intended operation, but an informational warning is returned. For a description of the **code**, wire the error cluster to a LabVIEW error-handling VI, such as the Simple Error Handler.



**source** identifies the VI where the error occurred.

## Description

[Get Channel Names By ID.vi](#) retrieves the names of channels based on the message ID to which they belong. This is a slower option than using [Select Names By ID.vi](#).



**Caution** This VI allocates memory, which can be a slow process.

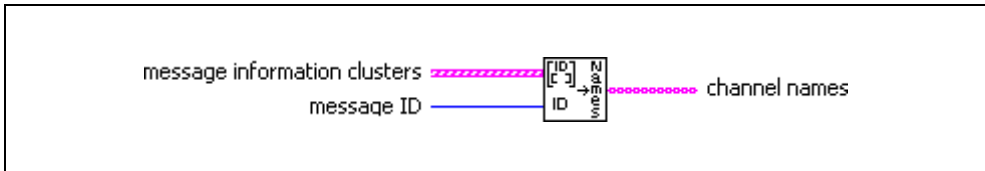
# Select Names By ID.vi

---

## Purpose

Retrieves an array of channel names for a particular message ID from the clusters created by [Get Channels.vi](#).

## Format



## Inputs



**message information clusters** represents an array of message information clusters.



**message size** represents the number of bytes in the message with the ID passed in.



**channel names** represents an array of channel name strings.



**message ID** is the CAN arbitration ID corresponding to the channel.

## Outputs



**channel names** represents an array of channel name strings.

## Description

**Select Names By ID.vi** extracts the names of channels based on the message ID to which they belong. This is a faster way to get either the name or number of channels for a particular message ID because it uses pre-allocated memory from [Get Channels.vi](#). It uses a linear search algorithm, but it is sorted initially by message ID, so it is possible to perform a binary search on the ID.