USER GUIDE

Frame Channel Conversion Library

Version 1.0

The *Frame Channel Conversion Library* is a collection of VIs for fast conversion of CAN frames to scaled channel information. It can be used on Windows 32 bit, Pharlap ETS, and VxWorks. This allows users of 90*xx* CompactRIO devices running VxWorks to perform fast conversions. The library also may be used to perform fast conversions on 900*x* CompactRIO devices running Pharlap ETS, and can handle conversions for USB CAN devices as well.

Contents

NATIONAL INSTRUMENTS™

Introduction

This section describes how to install the *Frame Channel Conversion Library* on Windows, Pharlap ETS, and VxWorks. It also describes the limitations of the library.

Installation

Begin by extracting the files to any directory. In that directory you will find three sub folders: `API`, `Drivers`, and `Examples`.

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

The `Examples` directory contains examples for PharLap ETS, VxWorks, and USB CAN. The Basic examples do not require a CAN module to be installed. The Basic Input Output examples do require CAN hardware. For the cRIO Input Output examples you must use a 2-port NI 985*x* C series module with both ports connected together. Bear in mind that the second port of the NI 985*x* C Series modules needs external power.

The `Drivers` directory contains two library files into which LabVIEW calls:

- `frchconvlib.dll` for Windows 32 bit editions (Vista/XP/2000) and CompactRIO controllers running Pharlap ETS.
- `frchconvlib.out` for CompactRIO controllers running VxWorks.

Windows

To install the driver for Windows, place `frchconvlib.dll` in the `system32` directory under your Windows installation directory; this is typically `C:\WINDOWS\system32\`.

Optionally, you may place `frchconvlib-errors.txt` under the `<ni installation directory>\LabVIEW <X.Y>\user.lib\errors\` directory. This file provides descriptions of library-specific errors if any are returned while your application is running.

Pharlap ETS

Follow the same procedure as the one for Windows. LabVIEW Real-Time downloads all necessary files to your Pharlap ETS target when you run your Real-Time VI.

VxWorks

To install the VxWorks driver, complete the following steps.

1. Access the CompactRIO target through an FTP client. This may be accomplished through the **File Transfer** tool from MAX, or you may use Windows Explorer by entering the following syntax within the address field: `ftp://<IP address of your target>`.

2. Navigate to the `/c/ni-rt/system/` directory and copy `frchconvlib.out` there.

3. To load channel information from a database file (the most common use case), you must additionally place the `.DBC` or `.NCD` file(s) in any directory on your CompactRIO target.

You should now be able to call into the libraries from LabVIEW using the provided VIs.

Usage

Complete these steps to allocate memory and make the library aware of the channel database data.

1. Call **Create Message Table.vi** to allocate memory for the message data.

2. Load the channel database information into the table.

   This is done by using 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. **Add Selected Channels from File.vi** may be used 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, the message table cannot be modified.

Conversion is only possible when all channels have been 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. The array can then be attached to a CAN cluster or frame as the "data" portion.

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

Note These VIs will generate a warning if they cannot find a message ID, so their output should be ignored if a warning is returned.

4. Prepare the data for transmission.

   There are a few VIs that may be used to prepare data for transmission, by providing information about the messages and channels.

   - **Get Number of Channels.vi** provides the user with the number of channels in a particular message.

   - **Get Message Size By ID.vi** helps the user determine the number of bytes used in a message.

   - **Get Channels.vi** retrieves all of the channel names that have been loaded into the message table. This can be a slow process, so it is recommended to call this VI once only, and to re-use the information cluster if needed.

   - **Select Names By ID.vi** provides the user 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 have been loaded.

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

   If you do not call this VI, you will be unable to create other message tables.

   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

This version of the *Frame Channel Conversion Library* does not support mode-dependent channels.
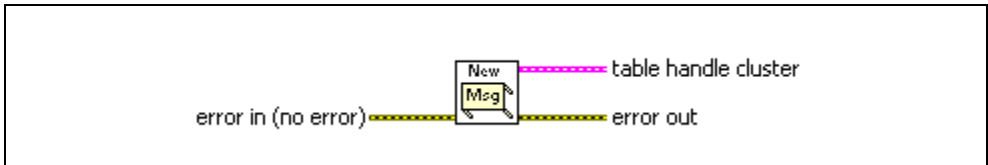
LabVIEW API Description

This section describes the *Frame Channel Conversion Library* API. The following VIs are located within the API directory. **API.vi** contains all of 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.

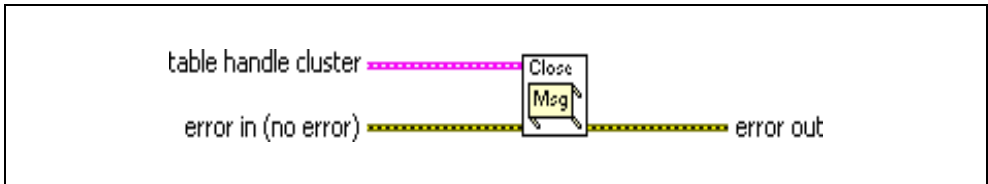**source** identifies the VI where the error occurred.

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.
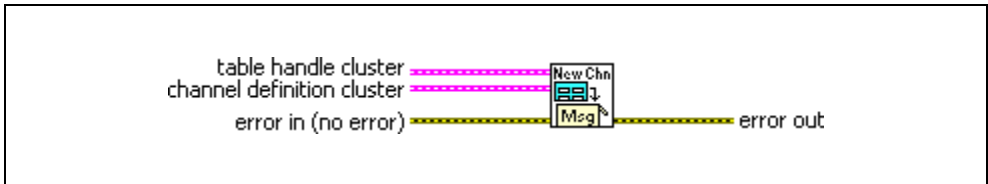
Add New Channel.vi

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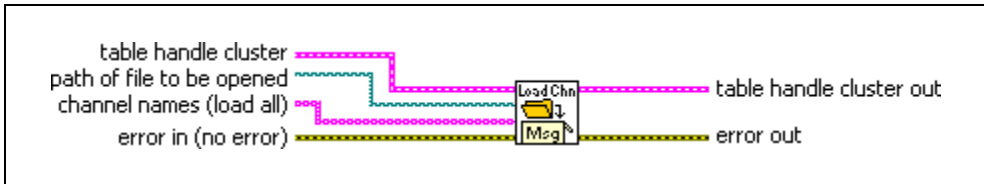
⚠ **Caution**   This VI allocates memory, which can be a slow process. If a conversion or helper VI has been called before this VI on the same message table, it will cause an error.

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 2 formats. If the channel input is left empty, all channels in the file will be loaded.
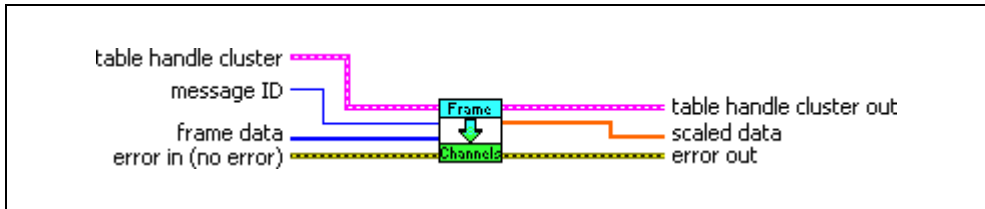
⚠️ **Caution** This VI allocates memory and performs disk access, which can be slow processes. If a conversion or helper VI has been called before this VI on the same message table, it will cause an error.

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.
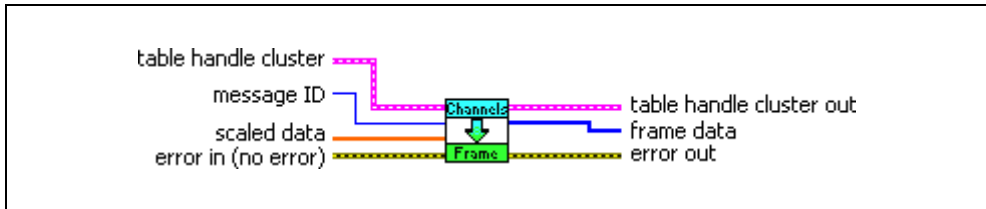
Description

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

table handle cluster
message ID
scaled data
error in (no error)
Channels → Frame
table handle cluster out
frame data
error out

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.
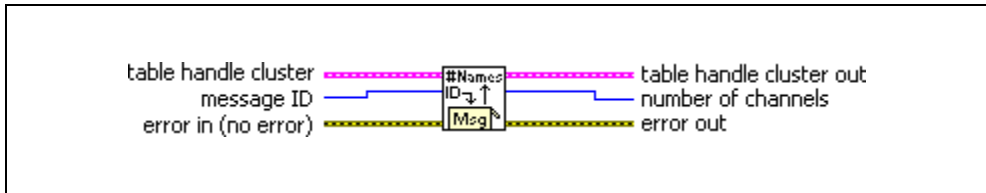
Description

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.

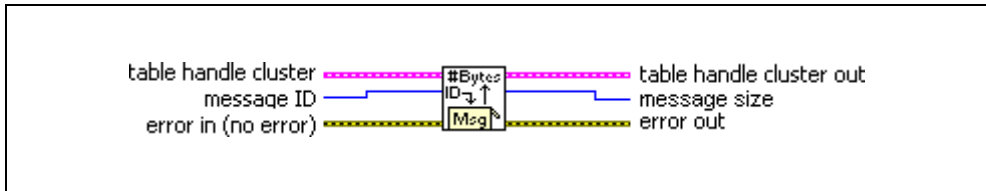**source** identifies the VI where the error occurred.

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.
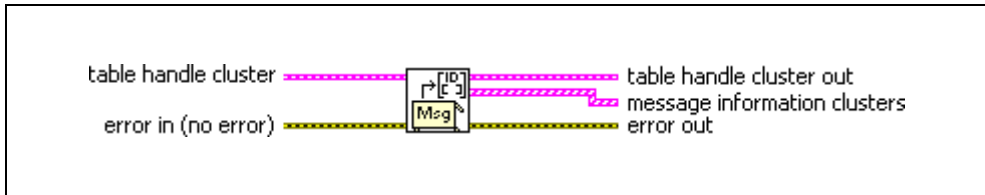
Description

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.

[abc] **channel names** is an array of channel name strings.

[error out icon] **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.

[TF] **status** is True if an error occurred.

[I32] **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.

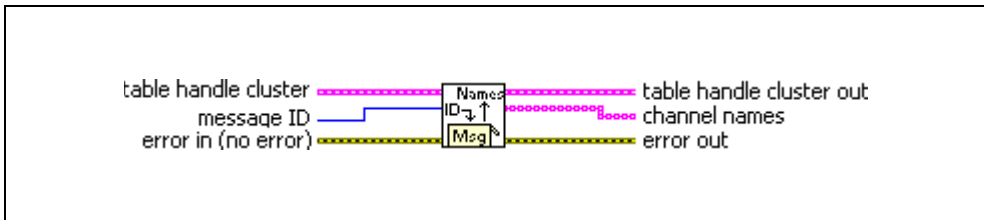[abc] **source** identifies the VI where the error occurred.

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.

TF	**status** is True if an error occurred.
I32	**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.
abc	**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**.

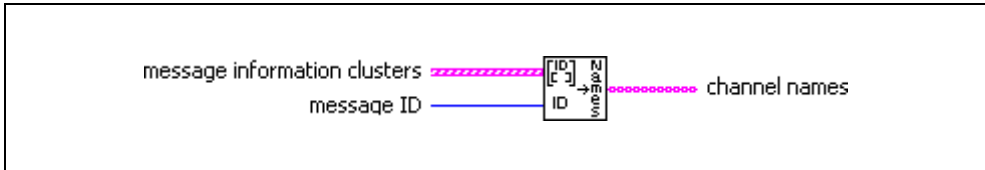⚠ **Caution** This VI allocates memory, which can be a slow process.

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.