CMC Driver for Open Protocol Torque Tools
User Manual

by Choose Movement Consulting, LLC

2010001-01-01 REV A.3

Contents

1. General Information

1.1. User Manual Content

This user manual covers the description, system requirements, installation instructions, operating instructions, tutorials, and suggested troubleshooting tips for the CMC Driver for Open Protocol Torque Tools.

1.2. Introduction

1.2.1. General Description

The CMC Driver for Open Protocol Torque Tools software adds support for communicating with torque tools that support Open Protocol to the CMC Driver Framework. The CMC Driver Framework (DF) is a family of Application Programming Interfaces (APIs) that can act as an instrument specific hardware abstraction layer. A torque tool is a class of instrument that performs and/or monitors a tightening operation and returns data associated with the tightening.

References to additional documents for other products are provided in the Referenced Documents section. Both CMC DF documents and manufacturer documents are listed there.

1.2.2. Product Support

For technical support:

- Post to the CMC Driver Framework community located at https://forums.ni.com/t5/CMC-Driver-Framework/gp-p/5394
- Email support@choose-mc.com or torque.support@choose-mc.com

It is encouraged to use the CMC Driver Framework community when possible so that others can learn from your questions.

1.3. System Requirements

The system requirements match the requirements for the version of LabVIEW that the CMC Driver for Open Protocol Torque Tools is being installed for. For more details, see: https://www.ni.com/en-us/support/documentation/supplemental/17/system-requirements-for-labview-development-systems-and-modules.html.

The CMC Driver for Open Protocol Torque Tools requires the software in the following table. With the exception of LabVIEW, the software modules will be installed or upgraded as part of the installation process.

Software Requirement	Minimum Version
LabVIEW	2017 SP1
CMC Driver Framework Core Functions	1.1.1
CMC Driver Framework Open Protocol API	1.0
NI-LabVIEW Command Line Interface x86	2.1
NI-488.2 Runtime	19.5
NI-Serial Runtime	17.5
NI-VISA Runtime	18.0

1.4. End User License Agreement

Copyright © 2016, Choose Movement Consulting, LLC. All rights reserved. This software is part of the CMC Driver Framework. By using this software, you agree to the CMC Driver Framework and CMC Driver End User License Agreement that is found at https://choose-mc.com/pages/legal.

1.5. Applicable Documents

Document Name	Document Number	Revision	Filename
Driver Framework Core User's Manual	1010001-01-01	REV A	DF_Core_Users_Manual.pdf

1.6. Acronyms, Definitions, and Abbreviations

1.6.1. Acronyms and Abbreviations

Acronym	Meaning
CMC	Choose Movement Consulting, LLC
EULA	End User License Agreement
UI	User Interface
VI	Virtual Instrument
API	Application Programming Interface
IO	In/Out
DF	Driver Framework
ID	Identification
MID	Message ID

1.6.2. Definitions

Term	Definition
.instconfig	The file extension used by the Driver Framework to identify files that contain all the necessary information to configure an Instrument.
Configuration File	Refers to an .instconfig

1.7. Products Supported

The CMC Driver for Open Protocol Torque Tools supports communicating with torque tools using the Open Protocol as specified by the Open Protocol specifications (Open Protocol Specification document number 9836441501 1.4 rev6) . This API supports communication with any device that supports Open Protocol communication and the versions of MIDs identified in those specifications.

1.7.1. Devices Used for Testing

Device	Manufacturer	Interface Type
mPro400 Controllers	Cleco/Apex Tool Group	Open Protocol
Power Focus 6000	Atlas Copco	Open Protocol
Simulated Device	Choose Movement Consulting, LLC	Open Protocol
CVI3 Vision	Desoutter	Open Protocol

1.7.2. Open Protocol MID Support

As the Open Protocol is evolving and updating, our intention is to support the full range of MIDs available. The following table lists the currently implemented and supported MIDs. If there is a MID you need that is not currently implemented but available in the target controller, please email torque.support@choose-mc.com and a member of our team will get back to you to determine your requirements and discuss increasing the priority on the specific MID.

Table 1: Supported MIDs

MID Name	MID #	Supported Rev
Communication Start	0001	1
Communication Start Acknowledge	0002	1,2,3
Communication Stop	0003	1
Command Error	0004	1
Command Accepted	0005	1
Application Number Upload Request	0010	1
Application Numbers Upload Reply	0011	1
Application “Selected” Subscribe	0014	1
Application Selected	0015	1
Application Selected Acknowledge	0016	1
Application “Selected” Unsubscribe	0017	1
Select Application	0018	1
Linking Group Numbers Upload Request	0030	1
Linking Group Numbers Upload Reply	0031	1,2
Linking Group Data Upload Request	0032	1
Linking Group Data Upload Reply	0033	1,2,3
Linking Group Info/Selected Subscribe	0034	1
Linking Group/Selected Info	0035	1
Linking Group Info/Selected Acknowledge	0036	1
Linking Group Info/Selected Unsubscribe	0037	1
Select Linking Group	0038	1,2
Linking Group Restart	0039	1,2
Tool Data Upload Request	0040	1
Tool Data Upload	0041	1,2
Disable Tool	0042	1
Enable Tool	0043	1
Vehicle ID Number Upload Subscribe	0051	1
Vehicle ID Number Upload	0052	1
Vehicle ID Number Upload Unsubscribe	0054	1
Last Tightening Result Data Subscribe	0060	1
Last Tightening Result Data Upload Reply	0061	1,2,3,4,5,6,40,41,42,500,998,999
Last Tightening Result Data Acknowledge	0062	1
Last Tightening Result Data Unsubscribe	0063	1
Old Tightening Result Upload Request	0064	1,2,3,4,5,6
Set Time in Torque Controller	0082	1
Multi-spindle result subscribe	0100	1
Multi-spindle result	0101	1
Multi-spindle result acknowledge	0102	1
Multi-spindle result unsubscribe	0103	1
VIN Download Request	0150	1
IO Device Status Request	0214	1
IO Device Status Reply	0215	1,2

MID Name	MID #	Supported Rev
Keep-Alive Message	9999	1

Not all messages will have a corresponding VI in the API. Acknowledge or reply MIDs are integrated with the messages that send or receive the data. For example, Communication Start (MID0001) expects to receive the Communication Start Acknowledge (MID0002), so the wait for MID0002 is built into the **Communication Start.vi**.

2. Software Installation

Your first installation of the CMC Driver Framework should use the **Driver Framework Installer.exe**. After the first time the installer is run, there are additional methods available - described below - for initiating installation of Driver Framework components. The first time through, the support files and runtime engines are installed. Subsequent Driver Framework component installations will use the support files.

2.1. First Time Install

1. Run **Driver Framework Installer.exe**. It is found in the ZIP file download for the CMC Driver Framework. Be sure to expand the ZIP file before running the installer.
2. Follow prompts to select which **Driver Framework** components to install.

2.2. Subsequent Installations

For subsequent installations, you can either use the CMC **Driver Framework Installer.exe** or **NI Package Manager**. See instructions below.

2.2.1. CMC Driver Framework Installer

1. Run **Driver Framework Installer.exe** from the folder you expanded it to.
2. Follow prompts to select which **Driver Framework components** to install.

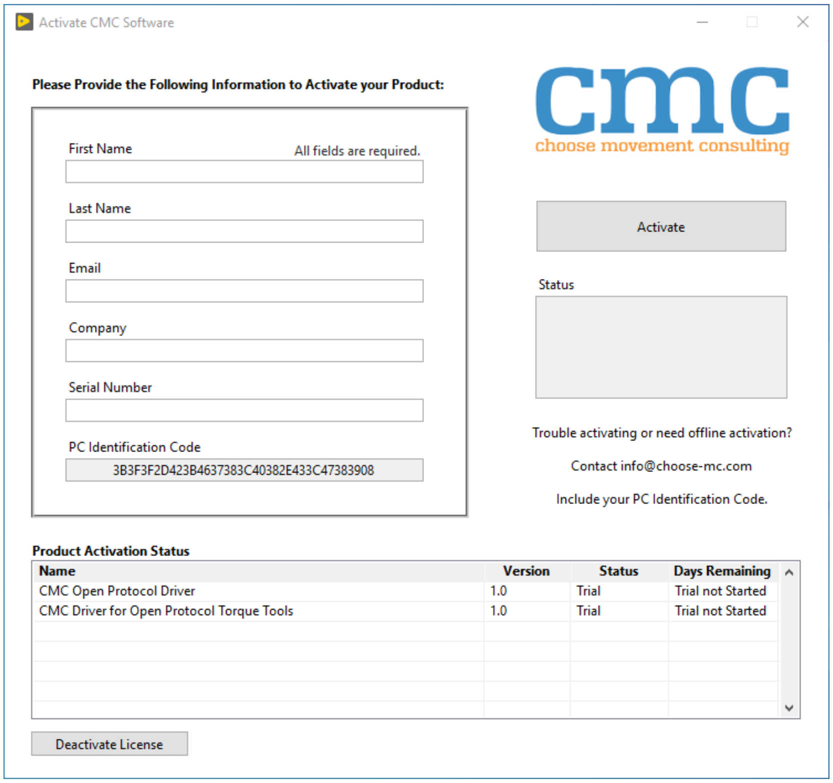
2.2.2. NI Package Manager

1. Click on the **Start Menu** icon.
2. Scroll down and expand the **National Instruments** folder.
3. Select **NI Package Manager**; it might be necessary to run it with elevated privileges depending on your computer's settings.
4. Confirm that you want to allow **NI Package Manager** to make changes to your system.
5. When **NI Package Manager** has finished updating the feeds, select a **tab** as follows:
    a. UPDATES: to install a newer version of an already installed API or Driver.
    b. INSTALLED: to uninstall an already installed API or Driver.
    c. PACKAGES: to install either a new API or Driver, or an API or Driver to a new version of LabVIEW.
6. Select the check box next to each version of an **API** or **Driver** to be installed, updated, or uninstalled and click the button that will initiate your selection.
7. Follow the prompts from **NI Package Manager** to finish the action.

3. Software Activation

The CMC Driver Framework Core software is free and does not require activation. To activate licensed CMC Driver Framework add-ons, like the CMC Driver for Open Protocol Torque Tools, complete the following steps:

1. Open the version of **LabVIEW** the **CMC Driver Framework** was installed to.
2. Go to the **Tools** menu and select **CMC Tools >> License Manager…**



Please Provide the Following Information to Activate your Product:

All fields are required.

First Name

Last Name

Email

Company

Serial Number

PC Identification Code: 3B3F3F2D423B4637383C40382E433C47383908

Trouble activating or need offline activation?

Contact info@choose-mc.com

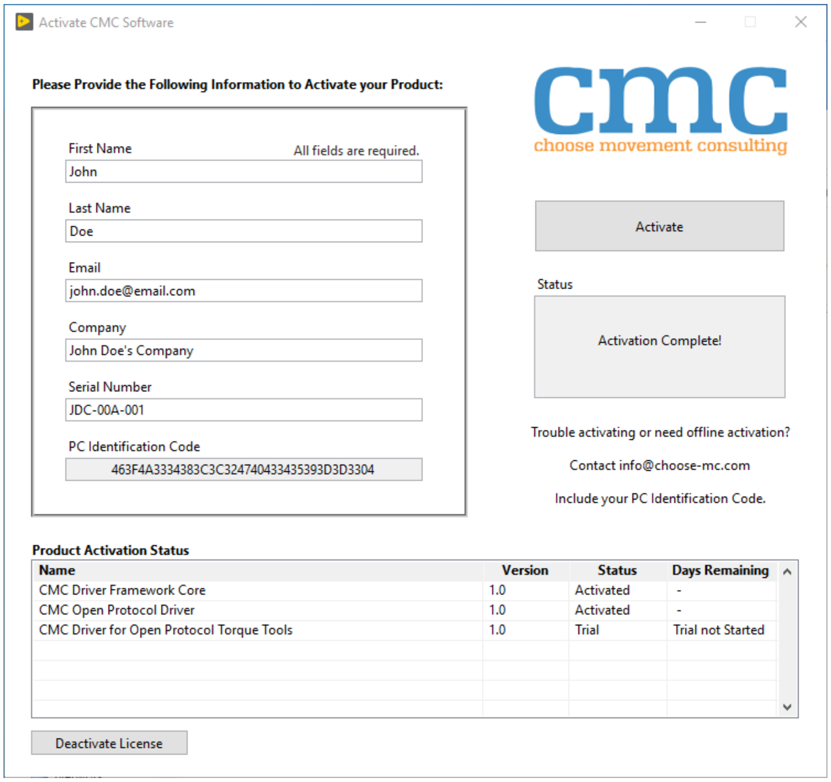
Include your PC Identification Code.

Product Activation Status

Name	Version	Status	Days Remaining
CMC Open Protocol Driver	1.0	Trial	Trial not Started
CMC Driver for Open Protocol Torque Tools	1.0	Trial	Trial not Started

Figure 1: Activation Window

3. Enter your information in the respective fields. Serial Number is typically sent to the purchaser automatically upon purchase.
4. Click **Activate**.



Please Provide the Following Information to Activate your Product:

All fields are required.

First Name	John
Last Name	Doe
Email	john.doe@email.com
Company	John Doe's Company
Serial Number	JDC-00A-001
PC Identification Code	463F4A3334383C3C3247404334353 93D3D3304

Status

Activation Complete!

Trouble activating or need offline activation?

Contact info@choose-mc.com

Include your PC Identification Code.

Product Activation Status

Name	Version	Status	Days Remaining
CMC Driver Framework Core	1.0	Activated	-
CMC Open Protocol Driver	1.0	Activated	-
CMC Driver for Open Protocol Torque Tools	1.0	Trial	Trial not Started

Figure 2: Software Activated

5. For offline activation, copy the PC Identification Code and send it along with your request to support@choose-mc.com or info@choose-mc.com.

4. Software Component Descriptions

The CMC Driver for Open Protocol Torque Tools is built on top of the CMC Driver Framework Core as well as the CMC Open Protocol API. The API can be found in the LabVIEW functions palette under **CMC >> Torque Tools API** as shown in the figure below.

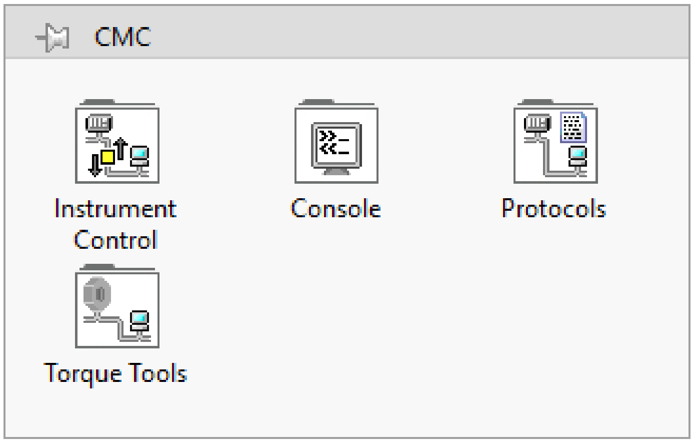


Figure 3: Torque Tools Palette

4.1. CMC Driver Framework Torque Tools Core API

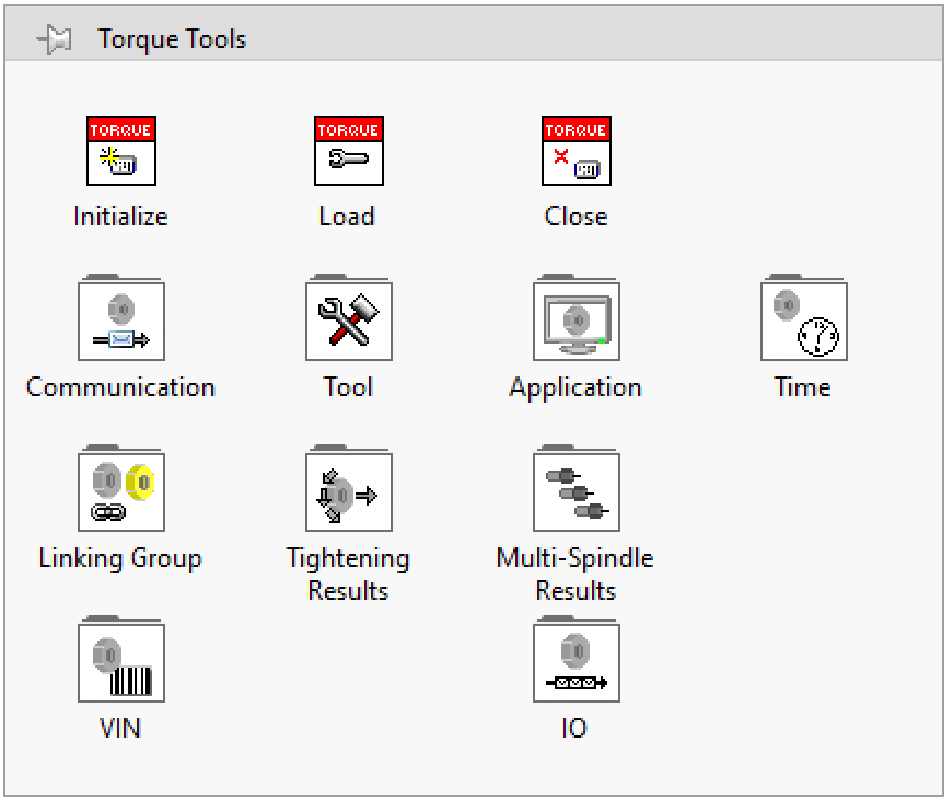


Figure 4: LabVIEW Torque Tools Palette

The CMC Driver for Open Protocol Torque Tools is a set of functions provided on the **CMC >> Torque Tools API** palette for programming an application to interact with torque tools. While the API supports all the described functions, please consult the documentation of a specific device to see whether it supports a specific function, MID or MID revision. We cannot guarantee that all devices will support all MIDs or MID revisions.

Most functions will have the following inputs:

- Console (F):
  If the .instconfig defines the console entries for the device as "selected", this input then defines whether or not the instrument communication for the individual VI is written to the console. The default value is false. This allows for enabling just those messages that are useful for troubleshooting or general information in the console.
- Torque Tools In:
  The Torque Tool object that the function will interact with. This input is required.
- error in (no error):
  This input is the standard error cluster. Most API functions will skip if there is an incoming error. While it is not required, it is good practice to use it to enforce order of operations, among other uses. The default value is no error.
- Timeout (10000 ms):

Defines how long the IO operation will wait, in milliseconds, for a response before timing out. -1 will never timeout. -2 will wait for the duration defined in the .instconfig before timing out. Other negative numbers are undefined. The default value is 10000.

Most functions will have the following outputs:

- Torque Tools Out:
  Returns the Torque Tool object once the VI has finished.
- Raw Returned Message:
  Returns the entirety of the returned response as a string without any interpretation of the data.
- Success?
  Returns a Boolean that indicates whether the command sent was successful.
- error out:
  Returns the error output cluster.

4.1.1. Initialize

Initialize the Torque Tools object for use in an application. The Torque Tool will be added to the console. The **Instrument Name** input must be unique for each instance in the application as that is the name that will be used to track it in the Console.

4.1.2. Load

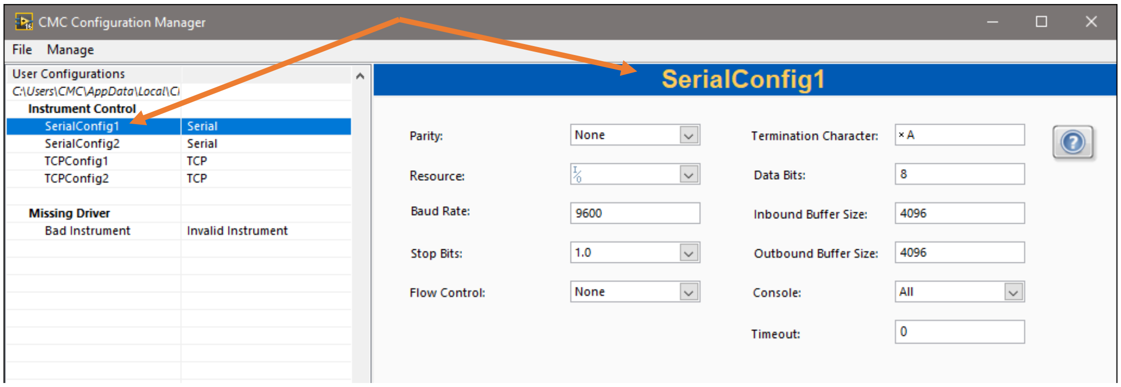
This is a polymorphic VI that will load a configuration from an .instconfig file into a Torque Tool object.

4.1.2.1. Load From Configuration File

This instance takes the path to the .instconfig file that defines the configuration for the Torque Tool object and constructs that object.

4.1.2.2. Load From Configuration Tool

This instance takes the name of the configuration and searches for an .instconfig file that has that name. Once it finds an .instconfig file that defines that name, it constructs that object. The name mentioned here is the name of the configuration as it shows up in the **CMC Configuration Manager**. It is not necessarily the name of the .instconfig file.



User Configurations	
C:\Users\CMC\AppData\Local\C\	
Instrument Control	
SerialConfig1	Serial
SerialConfig2	Serial
TCPConfig1	TCP
TCPConfig2	TCP
Missing Driver	
Bad Instrument	Invalid Instrument

SerialConfig1

Parity: None
Resource:
Baud Rate: 9600
Stop Bits: 1.0
Flow Control: None
Termination Character: × A
Data Bits: 8
Inbound Buffer Size: 4096
Outbound Buffer Size: 4096
Console: All
Timeout: 0

Figure 5. .instconfig Configuration Name

4.1.3. Close

Closes the session to the Torque Tools object and destroys the object. For those objects that use the Instrument Control to communicate with the instrument, this also closes the communication session.

4.1.4. Communication

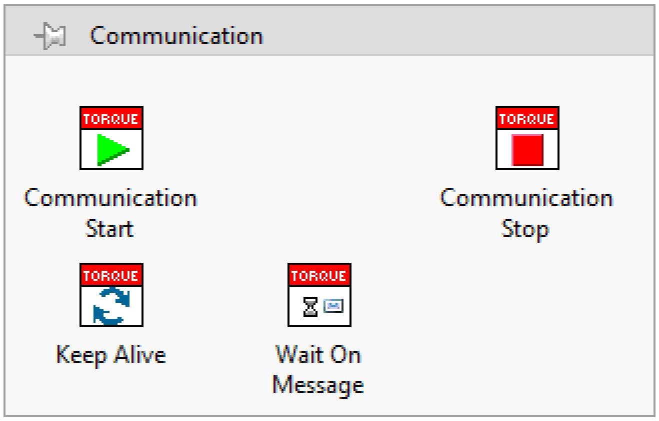


Figure 6: Torque Tools Communication Palette

This subpalette provides access to the functions common to managing the communication between an application and the targeted Torque Tool.

4.1.4.1. Communication Start

Sends the Communication Start message to the Torque Tool and waits for the acknowledgement message from the controller. The Start Acknowledge message contains only the information available in the returned message, based on the requested message version set in the .instconfig file.

4.1.4.2. Keep Alive

Sends the message necessary to instruct the device to keep the communication port open. This will need to be sent based on an interval specified by the manufacturer of the tool.

4.1.4.3. Wait On Message

Checks the communication read buffer for messages sent from the tool that ends with the appropriate termination character and then decodes the header information and returns the data to be decoded by the appropriate VI so any applicable acknowledge messages can be sent. The most common use of the VI is in looking for data sent from a subscription that would have previously been set in the tool.

4.1.4.4. Communication Stop

Sends the Communication Stop message to the Torque Tool and waits for the acknowledge message from the controller. Also closes the communication channel.

4.1.5. Tool

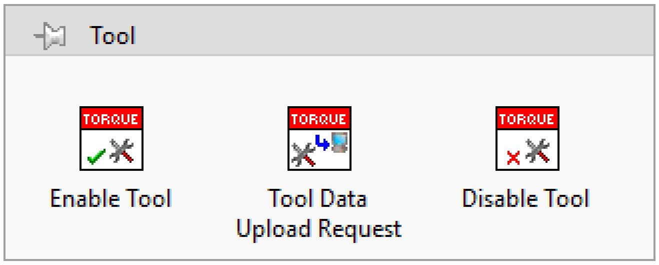


Figure 7: Torque Tools Tool Palette

This subpalette provides access to functions for controlling and getting information on the specific Tool(s) installed on the controller.

4.1.5.1. Enable Tool

Sends the Enable Tool message and waits for the acknowledge message from the controller.

4.1.5.2. Tool Data Upload Request

Requests the Tool Data Upload information from the tool and waits for the response message from the controller.

4.1.5.3. Disable Tool

Sends the Disable Tool message and waits for the acknowledge message.

4.1.6. Application

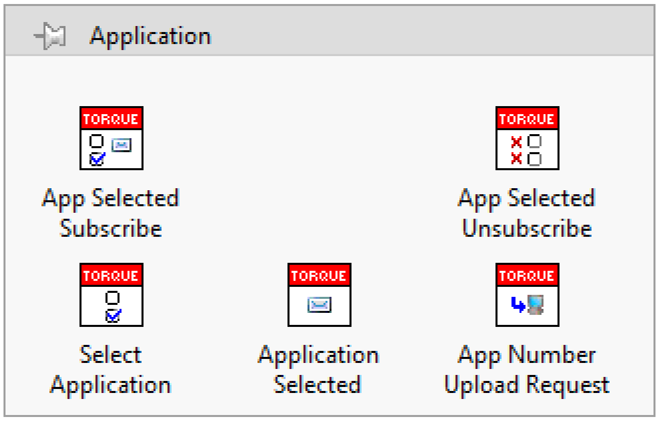


Figure 8: Torque Tools Application Palette

This subpalette provides access to functions for interacting with the applications loaded on the controller which get run using the tool.

4.1.6.1. App Selected Subscribe

Subscribes to the Application Selected message and waits for the acknowledge message from the controller.

4.1.6.2. Select Application

Sends the command to select the specified application to make it the active application and waits for the acknowledge message.

4.1.6.3. Application Selected

Decodes the Application Selected message returned by the Wait On Message VI (see Section 4.1.4.3). If the supplied Message String input is not an Application Selected message the VI will return an error.

The Application Selected message is sent from the controller when the active application is changed on the controller and a subscription is active.

4.1.6.4. App Number Upload Request

Requests a list of all applications configured on the torque tool's controller and waits for the returned message containing the list of applications.

4.1.6.5. App Selected Unsubscribe

Unsubscribes from the Application Selected message and waits for the acknowledge message from the controller.

4.1.7. Time

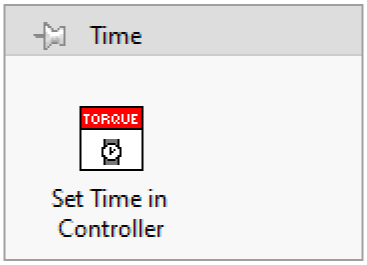


Figure 9: Torque Tools Time Palette

4.1.7.1. Set Time in Controller

Reads the current time of the computer and sends it to the controller to set as the current time on the controller. Used to synchronize the timestamps on results as closely as possible.

4.1.8. Linking Group

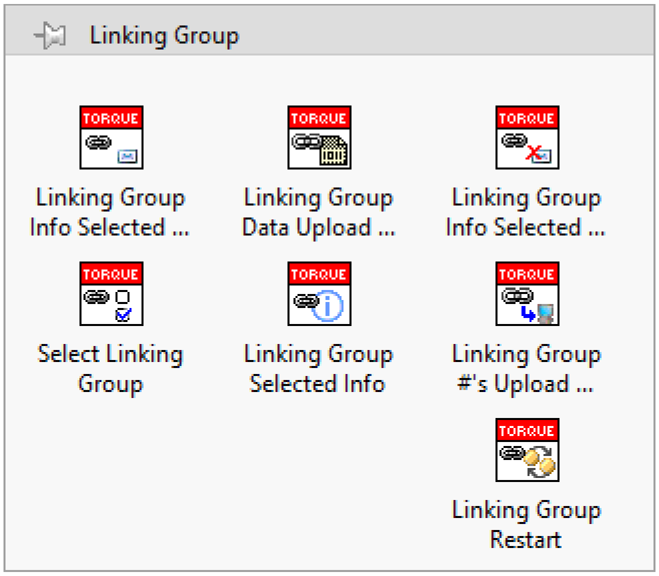

Linking Group
TORQUE
Linking Group Info Selected ...
TORQUE
Linking Group Data Upload ...
TORQUE
Linking Group Info Selected ...
TORQUE
Select Linking Group
TORQUE
Linking Group Selected Info
TORQUE
Linking Group #'s Upload ...
TORQUE
Linking Group Restart

Figure 10: Torque Tools Linking Group Palette

4.1.8.1. Linking Group Info Selected Subscribe

Subscribes to the Linking Group Info Selected message and waits for the acknowledge message from the controller.

4.1.8.2. Linking Group Data Upload Request

Requests the Linking Group Data for a specific Linking Group and waits for the return message containing data. Returned information contains the name of the linking group and the application numbers that are a part of the linking group. See the documentation for the MID for all the data returned and the available MID revisions.

4.1.8.3. Linking Group Info Selected Unsubscribe

Unsubscribes from the Linking Group Info Selected message and waits for the acknowledge message from the controller.

4.1.8.4. Select Linking Group

Set the specified linking group as the active linking group and waits for the acknowledge message.

4.1.8.5. Linking Group Selected Info

Decodes the Linking Group Info message returned by the Wait On Message VI (see Section 4.1.4.3). If the supplied Message String input is not a Linking Group Selected message the VI will return an error.

The Linking Group Info message is sent from the controller when the Linking Group is changed on the controller and a subscription is active.

4.1.8.6. *Linking Group #s Upload Request*

Requests a list of the linking groups present on the controller and waits for the returned message containing the list of Linking Groups.

4.1.8.7. *Linking Group Restart*

Restart the current linking group. See the documentation for your specific torque controller to understand how specific controllers respond to this command.

4.1.9. Tightening Results

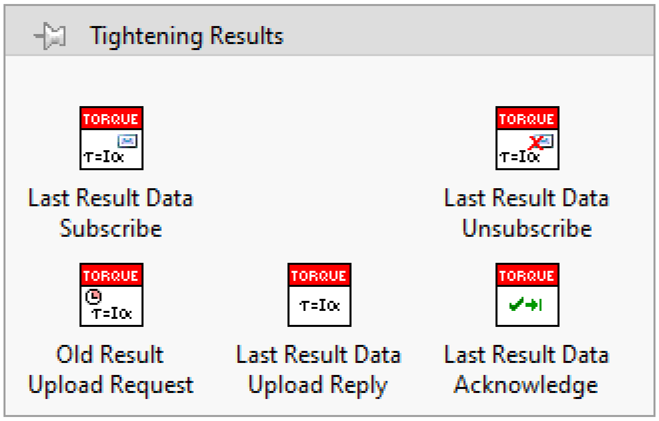

Tightening Results
TORQUE
T=I∝
Last Result Data Subscribe
TORQUE
T=I∝
Last Result Data Unsubscribe
TORQUE
T=I∝
Old Result Upload Request
TORQUE
T=I∝
Last Result Data Upload Reply
TORQUE
Last Result Data Acknowledge

Figure 11: Torque Tools Tightening Results Palette

4.1.9.1. *Last Result Data Subscribe*

Subscribes to the Last Result Data message and waits for the acknowledge message from the controller.

4.1.9.2. *Last Result Data Unsubscribe*

Unsubscribes from the Last Result Data message and waits for the acknowledge message from the controller.

4.1.9.3. *Last Result Data Upload Reply*

Decodes the Last Result Data message returned by the Wait On Message VI (see Section 4.1.4.3). If the supplied Message String input is not a Last Result Data message the VI will return an error. After decoding, acknowledge message is sent to the controller.

The Last Result Data message is sent from the controller when a tightening has been completed by the torque tool and a subscription to the message is active.

4.1.9.4. *Last Result Data Acknowledge*

Acknowledge receipt of the Last Tightening Result data from the controller.

4.1.9.5. *Old Result Upload Request*

Requests an old tightening result be sent from the tightening data stored in the torque controller. A Tightening ID of 0 will return the most recent tightening data.

4.1.10. Multi-Spindle Results

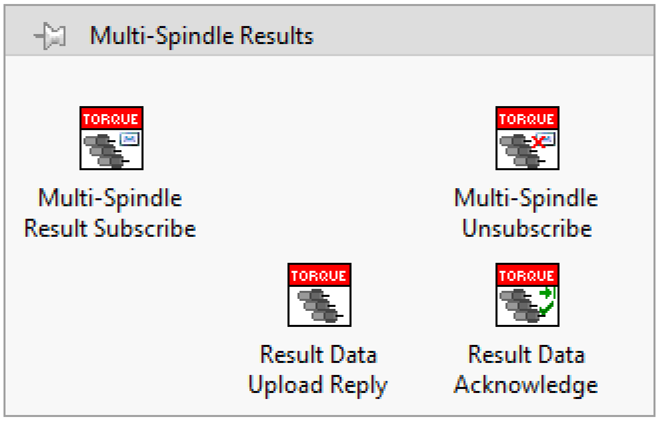


Figure 12: Torque Tools Multi-Spindle Results Palette

4.1.10.1. Multi-Spindle Result Subscribe

Subscribes to the Multi-Spindle Result message and waits for the acknowledge message from the controller.

4.1.10.2. Multi-Spindle Result Unsubscribe

Unsubscribes from the Multi-Spindle Result message and waits for the acknowledge message from the controller.

4.1.10.3. Multi-Spindle Result Upload Reply.

Decodes the Multi-Spindle Result message returned by the Wait On Message VI (see Section 4.1.4.3). If the supplied Message String input is not a Multi-Spindle Result message the VI will return an error. After decoding, acknowledge message is sent to the controller.

The Multi-Spindle Result message is sent from the controller when a multi-spindle tightening has been completed by the torque tool and a subscription to the message is active.

4.1.10.4. Multi-Spindle Result Data Acknowledge

Acknowledge receipt of the Last Tightening Result data from the controller.

4.1.11. IO

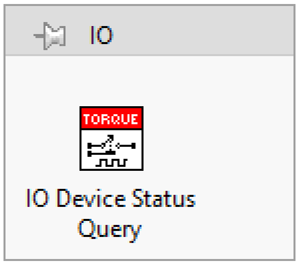


Figure 13: Torque Tools IO Palette

4.1.11.1. *IO Device Status Query*

Requests the status of the Relay and Digital devices connected to the torque controller based on the Device Number specified as an input. Waits for the information to be returned from the controller.

4.1.12. VIN Telegrams

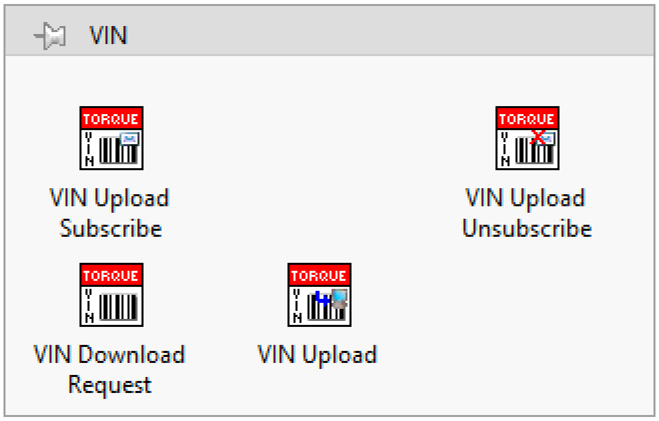


Figure 14: Torque Tools VIN Palette:

4.1.12.1. *VIN Download Request*

Sets a new VIN number in the torque controller to be the active VIN number and waits for the acknowledge message.

4.1.12.2. *VIN Upload Subscribe*

Send a subscription to when a new VIN number is set as the active VIN number in the torque tool.

4.1.12.3. *VIN Upload*

Decodes the message string containing the VIN Upload information while subscribed to new VIN number uploads in the torque tool.

4.1.12.4. *VIN Upload Unsubscribe*

Unsubscribes from receiving new VIN number set as the active VIN number in the torque tool.

5. Setup Tutorial

5.1. Controller Setup

It is necessary to verify that the torque controller is set up to communicate via Open Protocol. The exact steps and settings needed to complete the setup will be different for each controller. Please consult the documentation for your specific controller for how to complete the generic steps outlined below.

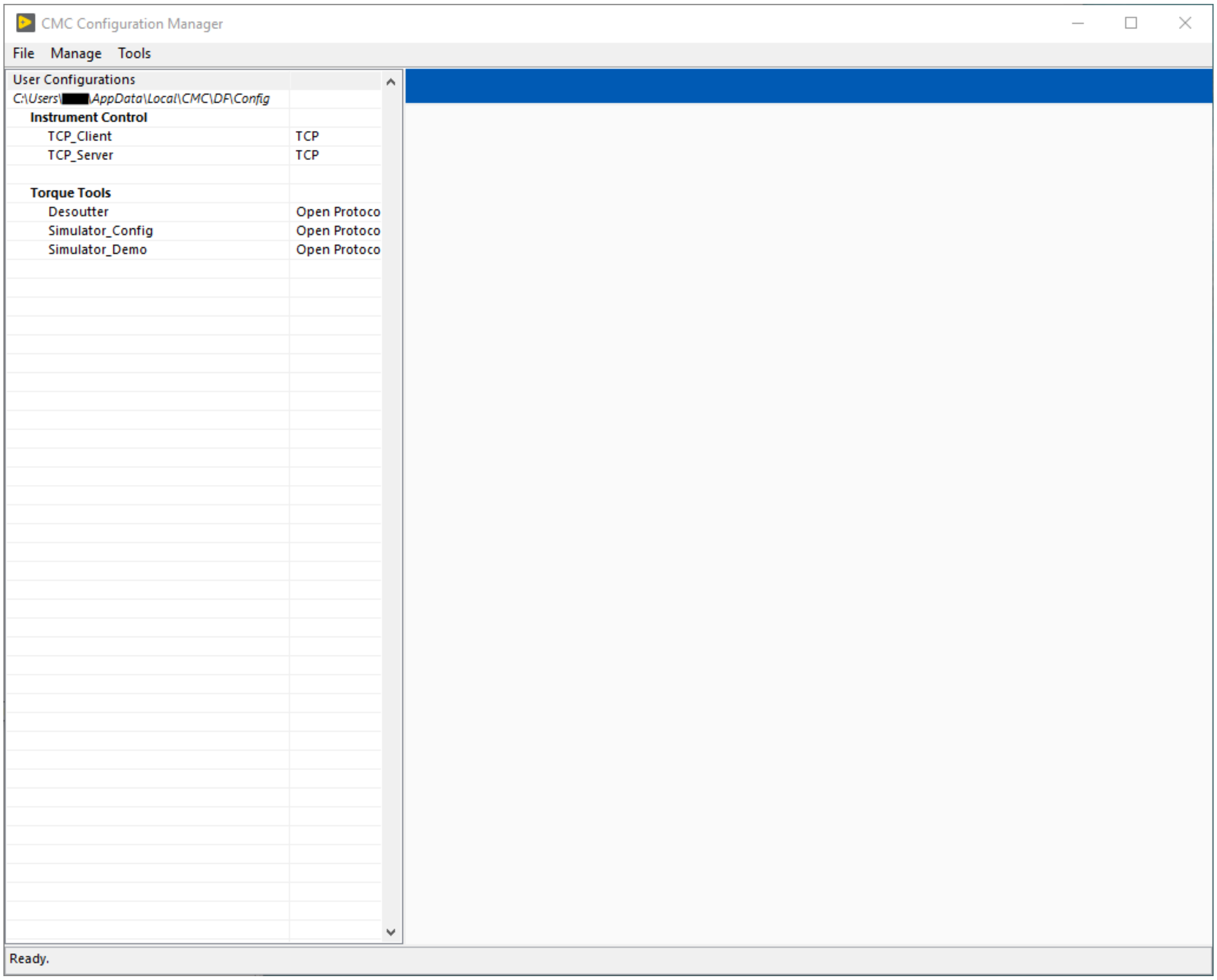
1. Make sure **Open Protocol** is set as the active communication protocol.
2. Set the **IP address** of both the computer running the application and the torque controller so they are compatible with each other.

3. Set the ethernet port for **Open Protocol** communication on the torque controller to the desired value, check the manufacturers documentation for the default or preferred port.
4. Save the settings to the torque controller.
5. Allow the controller to restart if it is required.

5.2. Controller *.instconfig File Setup

The Torque Tools API ships with an example .instconfig files that enables a user to run the example without being connected to a Torque Tool controller. To configure an .instconfig file for use with a torque tool, follow the steps listed below referencing the Driver Framework Core User's Manual.

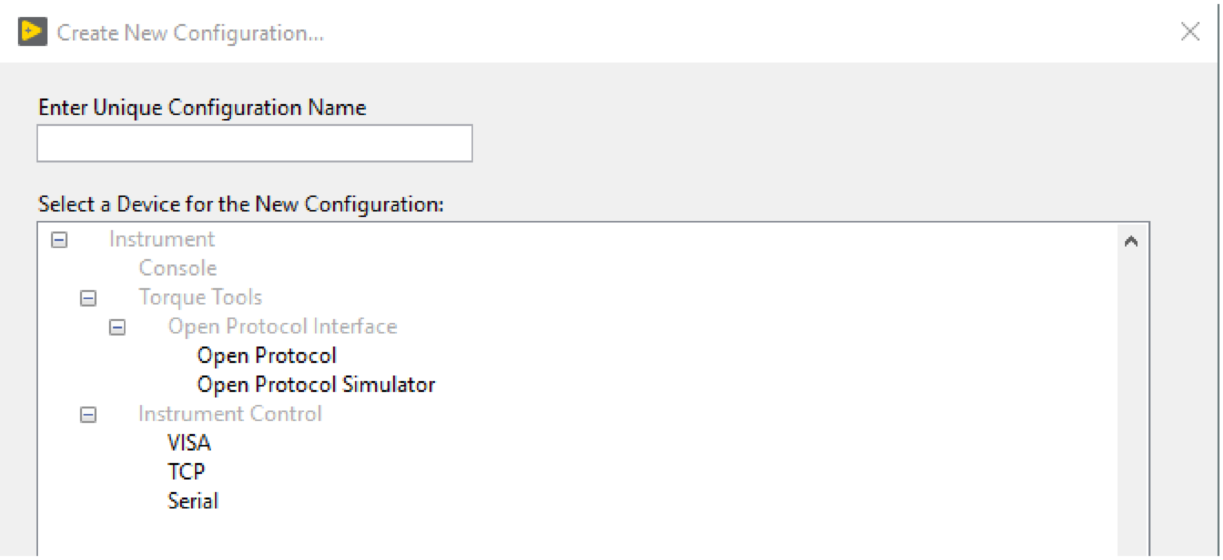
1. Open the version of **LabVIEW** for which the Torque Tools API was installed.
2. From the **LabVIEW Tools Menu** select **CMC Tools** >> **Configuration Manager**…
3. If this is the first time the **CMC Configuration Manager** has been opened the list of configuration files will look like the image below.



User Configurations	
C:\Users\[redacted]\AppData\Local\CMC\DF\Config	
Instrument Control	
TCP_Client	TCP
TCP_Server	TCP
Torque Tools	
Desoutter	Open Protoco
Simulator_Config	Open Protoco
Simulator_Demo	Open Protoco

Figure 15. Instrument Configuration List in the CMC Configuration Manager.

4. From the **File** menu, select **Create New**
5. Enter a unique name into the **Enter Unique Configuration** Name control
6. Select the **Open Protocol** item in the **Hardware configuration** selection list



Create New Configuration...

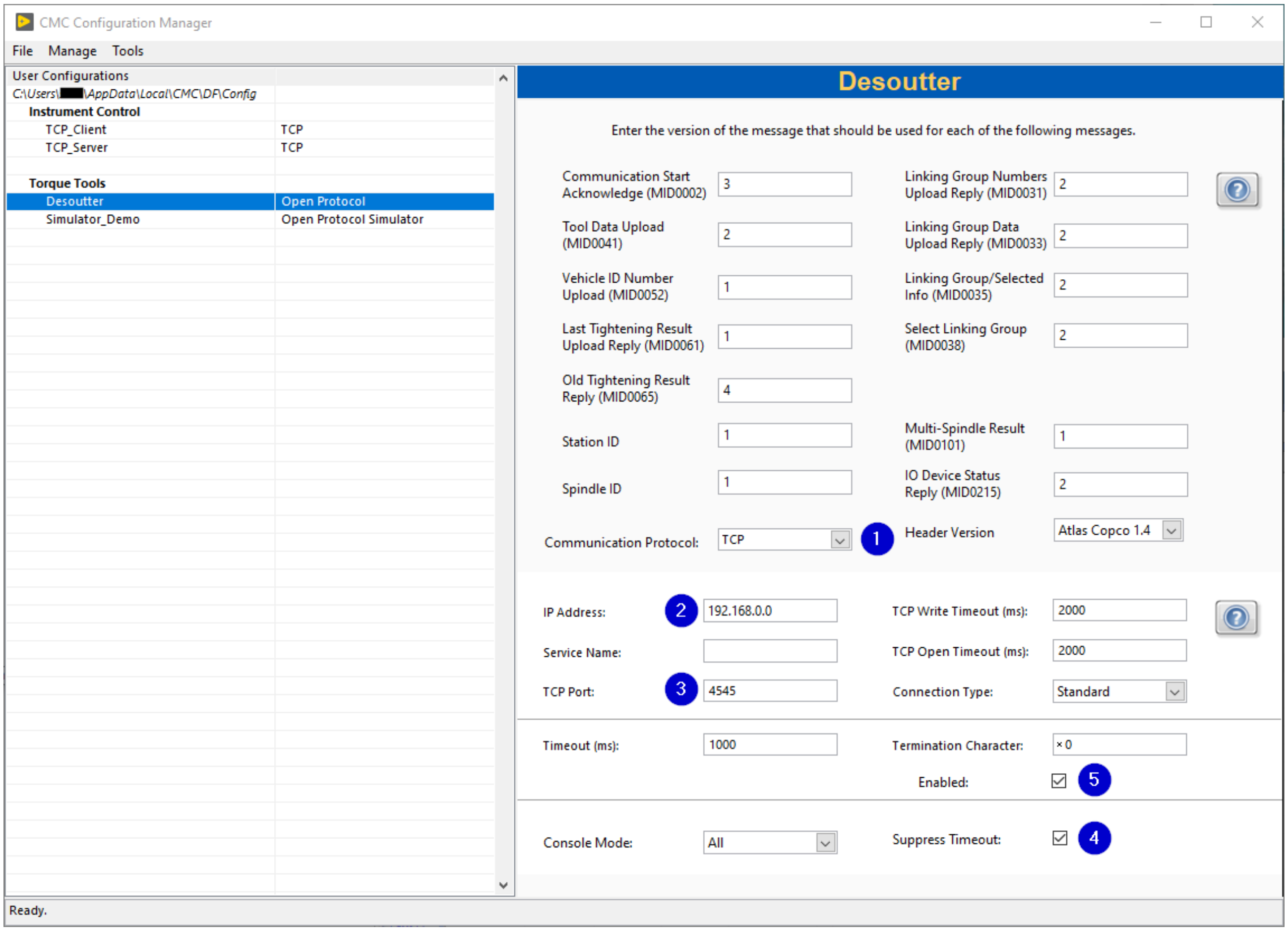
Enter Unique Configuration Name

Select a Device for the New Configuration:

- Instrument
  - Console
  - Torque Tools
    - Open Protocol Interface
      - Open Protocol
      - Open Protocol Simulator
  - Instrument Control
    - VISA
    - TCP
    - Serial

Figure 16. Creating a New Configuration.

7. Click **OK**
8. Select a location to save the new .instconfig file (the default location is `C:\Users\<user name>\AppData\Local\CMC\DF\Config)` and click **OK**
9. The **CMC Configuration Manager** will automatically switch to editing the newly created configuration
10. Make sure the **Communication Protocol** is set to **TCP.** (1) in the figure below
11. Set the **IP Address** (2) to the **IP address** collected from the controller
12. Set the **TCP Port** (3) to the **Port** number collected from the controller
13. Make sure that the **Service Name** (just below (2)) is empty.
    a. *Most torque tool controllers are not configured with service names and will not connect if there is a service name specified.*



User Configurations
C:\Users\[redacted]\AppData\Local\CMC\DF\Config

Instrument Control	
TCP_Client	TCP
TCP_Server	TCP
Torque Tools	
Desoutter	Open Protocol
Simulator_Demo	Open Protocol Simulator

Desoutter

Enter the version of the message that should be used for each of the following messages.

Communication Start Acknowledge (MID0002)	3	Linking Group Numbers Upload Reply (MID0031)	2
Tool Data Upload (MID0041)	2	Linking Group Data Upload Reply (MID0033)	2
Vehicle ID Number Upload (MID0052)	1	Linking Group/Selected Info (MID0035)	2
Last Tightening Result Upload Reply (MID0061)	1	Select Linking Group (MID0038)	2
Old Tightening Result Reply (MID0065)	4		
Station ID	1	Multi-Spindle Result (MID0101)	1
Spindle ID	1	IO Device Status Reply (MID0215)	2
Communication Protocol:	TCP	Header Version	Atlas Copco 1.4
IP Address:	192.168.0.0	TCP Write Timeout (ms):	2000
Service Name:		TCP Open Timeout (ms):	2000
TCP Port:	4545	Connection Type:	Standard
Timeout (ms):	1000	Termination Character:	\x0
		Enabled:	☑
Console Mode:	All	Suppress Timeout:	☑

Figure 17. Open Protocol Torque Tool Configuration.

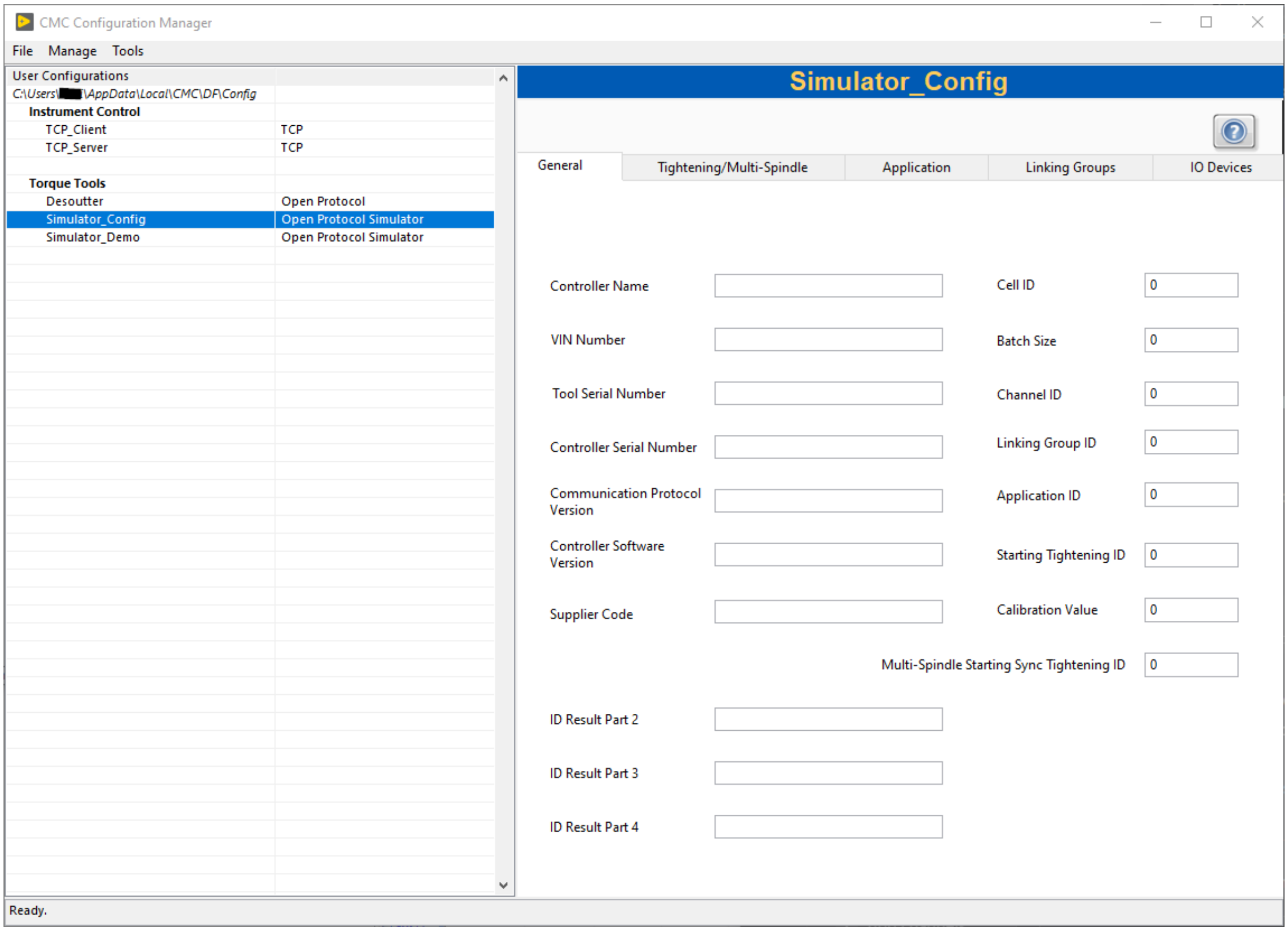
14. Enable the Termination Character of "0x00" by checking the **Enable** box under the **Termination Character** input. (5)
15. If running the example with this .instconfig file, consider checking the **Suppress Timeout** dialog box as it will clear the periodic timeout error that will happen when waiting for a subscription. More detailed information on the **Suppress Timeout** can be found in the **Driver Framework Core User's Manual**. (4)
16. Select **Save** from the **File** menu
17. After the file is saved, you may exit the **CMC Configuration Manager** or create another .instconfig file if desired

5.3. Simulated *.instconfig File Setup

Configuring a new Simulated Device is similar to creating a configuration for a physical device, but a lot more information can be specified. The information provided in each of the tabs represents a field of data that can returned by a controller for the MIDs available. Sometimes the information is used in multiple messages. Descriptions of the fields are provided in further subsections of this description. Please refer to the manuals and documentation for your specific device to determine if that information is relevant to your device. The following instructions walk through the process of creating a new Simulated Device from scratch.

1. Follow steps 1-9 from Section 5.2, substituting the selection of the **Open Protocol Simulator** device in step 6.
2. Make changes to the information for each of the tabs (detailed in the following sub sections) of the user interface.
3. Select **Save** from the **File** menu.
4. After the file is saved you may exit the **CMC Configuration Manager** or create another .instconfig file if desired.

5.3.1. General Configuration Tab



User Configurations

C:\Users\[redacted]\AppData\Local\CMC\DF\Config

Instrument Control	
TCP_Client	TCP
TCP_Server	TCP
Torque Tools	
Desoutter	Open Protocol
Simulator_Config	Open Protocol Simulator
Simulator_Demo	Open Protocol Simulator

Simulator_Config

Controller Name		Cell ID	0
VIN Number		Batch Size	0
Tool Serial Number		Channel ID	0
Controller Serial Number		Linking Group ID	0
Communication Protocol Version		Application ID	0
Controller Software Version		Starting Tightening ID	0
Supplier Code		Calibration Value	0
		Multi-Spindle Starting Sync Tightening ID	0
ID Result Part 2			
ID Result Part 3			
ID Result Part 4			

Ready.

Figure 18: Simulated Device General Configuration

5.3.1.1. Controller Name

The controller name is a 25-character long name assigned to the controller. The name can be any number of characters up to 25, the length sent from the controller to the application software is always exactly 25, so the simulator software takes care of formatting the return string to be exactly 25 characters. When a new value is entered, the configuration will truncate if the length is too long and pad with spaces before the string if there are fewer than 25 characters.

5.3.1.2. VIN Number

The VIN (Vehicle Identification Number) is the default VIN to use when returning results. The VIN number is allowable up to 25 characters long. The VIN number gets updated when “MID0050” is sent with a new VIN number. When a new value is entered, the configuration will truncate if the length is too long and pad with spaces before the string if there are fewer than 25 characters.

5.3.1.3. Tool Serial Number

The Tool Serial Number is 14 ASCII characters that represent the serial number of the tool used in the tightening result returned. It may also be returned by other messages but always represents the connected tools serial number. This is not modifiable by any MID. When a new value is entered, the configuration will truncate if the length is too long and pad with spaces before the string if there are fewer than 14 characters.

5.3.1.4. Controller Serial Number

The Controller Serial Number is 10-character ASCII representation of the serial number of the controller. It may also be returned by other messages but always represents the connected controller’s serial number. This is not modifiable by any MID. When a new value is entered, the configuration will truncate if the length is too long and pad with spaces before the string if there are fewer than 10 characters.

5.3.1.5. Communication Protocol Version

The Communication Protocol Version is a 19-character ASCII representation of the currently installed version of the protocol. When a new value is entered, the configuration will truncate if the length is too long and pad with spaces before the string if there are fewer than 19 characters.

5.3.1.6. Controller Software Version

The Controller Software Version is a 19-character ASCII representation of the currently installed software or firmware installed on the controller. When a new value is entered, the configuration will truncate if the length is too long and pad with spaces before the string if there are fewer than 19 characters.

5.3.1.7. Supplier Code

The Supplier Code is a 3-character abbreviated version of the manufacturer of the torque tool.

5.3.1.8. ID Result Part 2

The ID Result Part 2 is an additional identification option that can be used to tie multiple ID numbers together in a results file. The ID Result number is allowable up to 25 characters long. Depending on the device specific implementation, the ID Result can get updated when “MID0150” is sent with a new VIN number. When a new value is entered, the configuration will truncate if the length is too long and pad with spaces before the string if there are fewer than 25 characters.

5.3.1.9. ID Result Part 3

The ID Result Part 3 is an additional identification option that can be used to tie multiple ID numbers together in a results file. The ID Result number is allowable up to 25 characters long. Depending on the device specific implementation, the ID Result can get updated when "MID0150" is sent with a new VIN number. When a new value is entered, the configuration will truncate if the length is too long and pad with spaces before the string if there are fewer than 25 characters.

5.3.1.10. IID Result Part 4

The ID Result Part 4 is an additional identification option that can be used to tie multiple ID numbers together in a results file. The ID Result number is allowable up to 25 characters long. Depending on the device specific implementation, the ID Result can get updated when "MID0150" is sent with a new VIN number. When a new value is entered, the configuration will truncate if the length is too long and pad with spaces before the string if there are fewer than 25 characters.

5.3.1.11. Cell ID

Cell ID is a 4-digit ASCII number ('0…9') representing the cluster number or cell ID as defined by the manufacturer of the device you are simulating. Maximum allowable entry is 9999.

5.3.1.12. Batch Size

The Batch Size is a 2-digit ASCII number ('0…9') that defines the number of rundowns that a parameter set or application is expecting when run. The Batch Size for a specific application can be updated using "MID0019". Currently there is no effect on the randomly generated data from a tightening subscription. Maximum allowable entry is 99.

5.3.1.13. Channel ID

Specifies the 2-digit ASCII number in the range of 0-99. The context of the Channel ID can change depending on the MID. Currently the simulator maintains just a single Channel ID in the data.

5.3.1.14. Linking Group ID

Specifies the 2-digit ASCII number of the active Linking Group/Job. When a new value is entered, the configuration checks to see if the newly specified Linking Group is present. If the Linking Group is not present in the Linking Group tab, then a dialog will pop up with the warning and set the value back to the previous value.

5.3.1.15. Application ID

Specifies the 3-digit ASCII number of the active Application/Parameter Set. Values entered here that are not in the Application array of the Application tab will be added at the end of the Application List array. Maximum allowable entry is 999.

5.3.1.16. Starting Tightening ID

Specifies the number to start at when returning Tightening IDs. Tightening IDs are 10-digit ASCII numbers ('0…9') that represent a unique ID given to each tightening completed. The simulator increments the tightening ID after each tightening result message sent from the simulator. Currently the individual tightening results returned are not retained in memory. The maximum allowable entry is 4294967295.

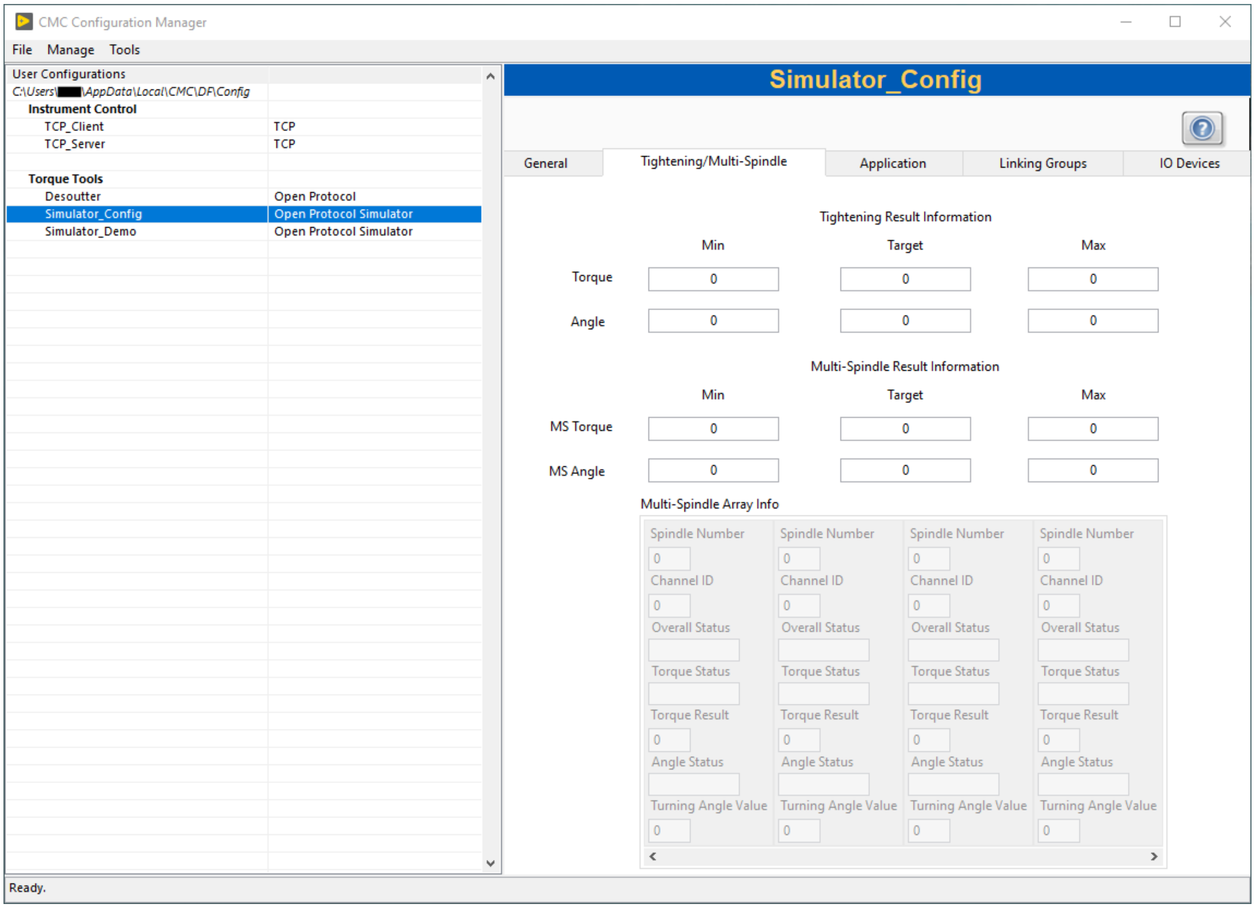
5.3.1.17. Calibration Value

The calibration value associated with a specific tool. Currently the calibration value is returned as the calibration value for any tools that are considered connected to the controller.

5.3.1.18. Multi-Spindle Starting Sync Tightening ID

Specifies the number to start at when returning Multi-Spindle Sync Tightening IDs. Sync Tightening IDs are 5-digit ASCII numbers (‘0…9’) that represent a unique ID given to each sync tightening completed. The simulator increments the tightening ID after each sync tightening result message sent from the simulator. Currently the individual sync tightening results returned are not retained in memory.

5.3.1. Tightening/Multi-Spindle Configuration Tab



User Configurations	
C:\Users\[redacted]\AppData\Local\CMC\DF\Config	
Instrument Control	
TCP_Client	TCP
TCP_Server	TCP
Torque Tools	
Desoutter	Open Protocol
Simulator_Config	Open Protocol Simulator
Simulator_Demo	Open Protocol Simulator

Tightening Result Information

	Min	Target	Max
Torque	0	0	0
Angle	0	0	0

Multi-Spindle Result Information

	Min	Target	Max
MS Torque	0	0	0
MS Angle	0	0	0

Multi-Spindle Array Info

Spindle Number	Spindle Number	Spindle Number	Spindle Number
0	0	0	0
Channel ID	Channel ID	Channel ID	Channel ID
0	0	0	0
Overall Status	Overall Status	Overall Status	Overall Status
Torque Status	Torque Status	Torque Status	Torque Status
Torque Result	Torque Result	Torque Result	Torque Result
0	0	0	0
Angle Status	Angle Status	Angle Status	Angle Status
Turning Angle Value	Turning Angle Value	Turning Angle Value	Turning Angle Value
0	0	0	0

Figure 19: Simulated Device Tightening Configuration

5.3.1.1. Torque Min

Specifies the minimum value that the simulator can return for a torque result. This is used specifically when the simulator is generating random data to return as a tightening result. The range of allowable values is 0 - 9999.99.

5.3.1.2. Torque Target

Specifies the nominal value that the simulator is targeting as the ideal returned torque result. This is used specifically when the simulator is generating random data to return as a tightening result. The range of allowable values is 0 - 9999.99.

5.3.1.3. Torque Max

Specifies the maximum value that the simulator can return for a torque result. This is used specifically when the simulator is generating random data to return as a tightening result. The range of allowable values is 0 - 9999.99.

5.3.1.4. Angle Min

Specifies the minimum value that the simulator can return for an angle result. This is used specifically when the simulator is generating random data to return as the angle portion of a tightening result. The range of allowable values is 0 - 99999.

5.3.1.5. Angle Target

Specifies the nominal value that the simulator is targeting as the ideal returned angle result. This is used specifically when the simulator is generating random data to return as the angle portion of a tightening result. The range of allowable values is 0 - 99999.

5.3.1.6. Angle Max

Specifies the maximum value that the simulator can return for an angle result. This is used specifically when the simulator is generating random data to return as the angle portion of a tightening result. The range of allowable values is 0 - 99999.

5.3.1.7. MS Torque Min

Specifies the minimum value that the simulator can return for a multi-spindle torque result. This is used specifically when the simulator is generating random data to return as a multi-spindle tightening result. The limit is used as the low limit for the random number generator for each of the spindles in the array. The range of allowable values is 0 - 9999.99.

5.3.1.8. MS Torque Target

Specifies the nominal value that the simulator is targeting as the ideal returned torque of the multi-spindle result. This is used specifically when the simulator is generating random data to return as a multi-spindle tightening result. The target is the same, but a different random number will be generated for each spindle in the array. The range of allowable values is 0 - 9999.99.

5.3.1.9. MS Torque Max

Specifies the maximum value that the simulator can return for a multi-spindle torque result. This is used specifically when the simulator is generating random data to return as a multi-spindle tightening result. The limit is used as the high limit for the random number generator for each of the spindles in the array. The range of allowable values is 0 - 9999.99.

5.3.1.10. MS Angle Min

Specifies the minimum value that the simulator can return for a multi-spindle angle result. This is used specifically when the simulator is generating random data to return as the multi-spindle angle portion of a multi-spindle tightening result. The limit is used as the low limit for the random number generator for each of the spindles in the array. The range of allowable values is 0 - 99999.

5.3.1.11. MS Angle Target

Specifies the nominal value that the simulator is targeting as the ideal returned multi-spindle angle result. This is used specifically when the simulator is generating random data to return as the angle portion of a multi-spindle tightening result. The target is the same, but a different random number will be generated for each spindle in the array. The range of allowable values is 0 - 99999.

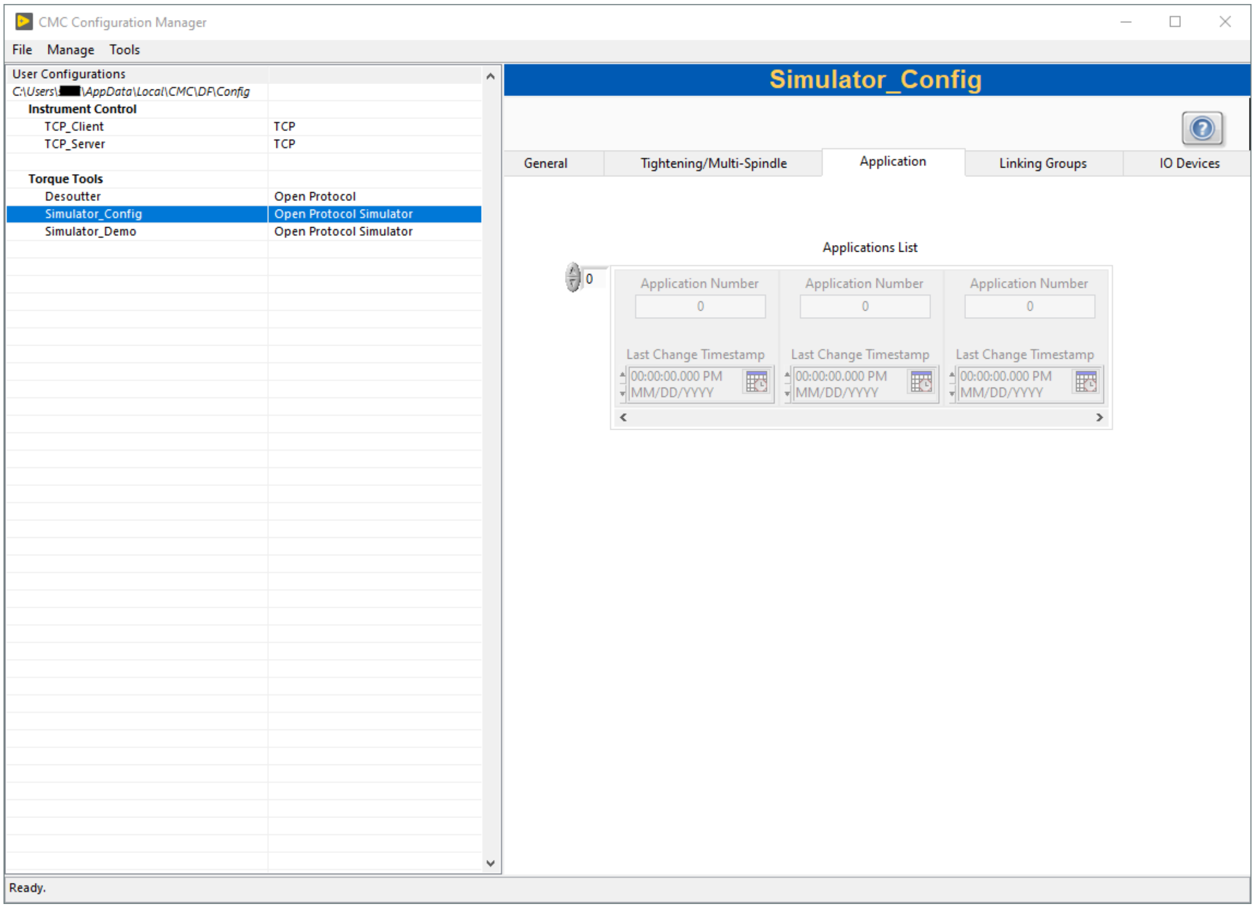
5.3.1.12. MS Angle Max

Specifies the maximum value that the simulator can return for a multi-spindle angle result. This is used specifically when the simulator is generating random data to return as the angle portion of a multi-spindle tightening result. The limit is used as the high limit for the random number generator for each of the spindles in the array. The range of allowable values is 0 - 99999.

5.3.1.13. Multi-Spindle Array Info

The multi-spindle array info is used to specify the number of spindles in the multi-spindle results. It is important to fill out the Spindle Number (a unique number in each array for each spindle 01-99) and the Channel ID of the spindle (a number in each member of the array 01-20). The remaining values are generated randomly each time a multi-spindle result is required, including the status entries, so whatever is entered in this array will be overwritten for those values.

5.3.2. Application Configuration Tab



User Configurations	
C:\Users\[redacted]\AppData\Local\CMC\DF\Config	
Instrument Control	
TCP_Client	TCP
TCP_Server	TCP
Torque Tools	
Desoutter	Open Protocol
Simulator_Config	Open Protocol Simulator
Simulator_Demo	Open Protocol Simulator

Simulator_Config

Applications List

0

Application Number	Application Number	Application Number
0	0	0
Last Change Timestamp	Last Change Timestamp	Last Change Timestamp
00:00:00.000 PM MM/DD/YYYY	00:00:00.000 PM MM/DD/YYYY	00:00:00.000 PM MM/DD/YYYY

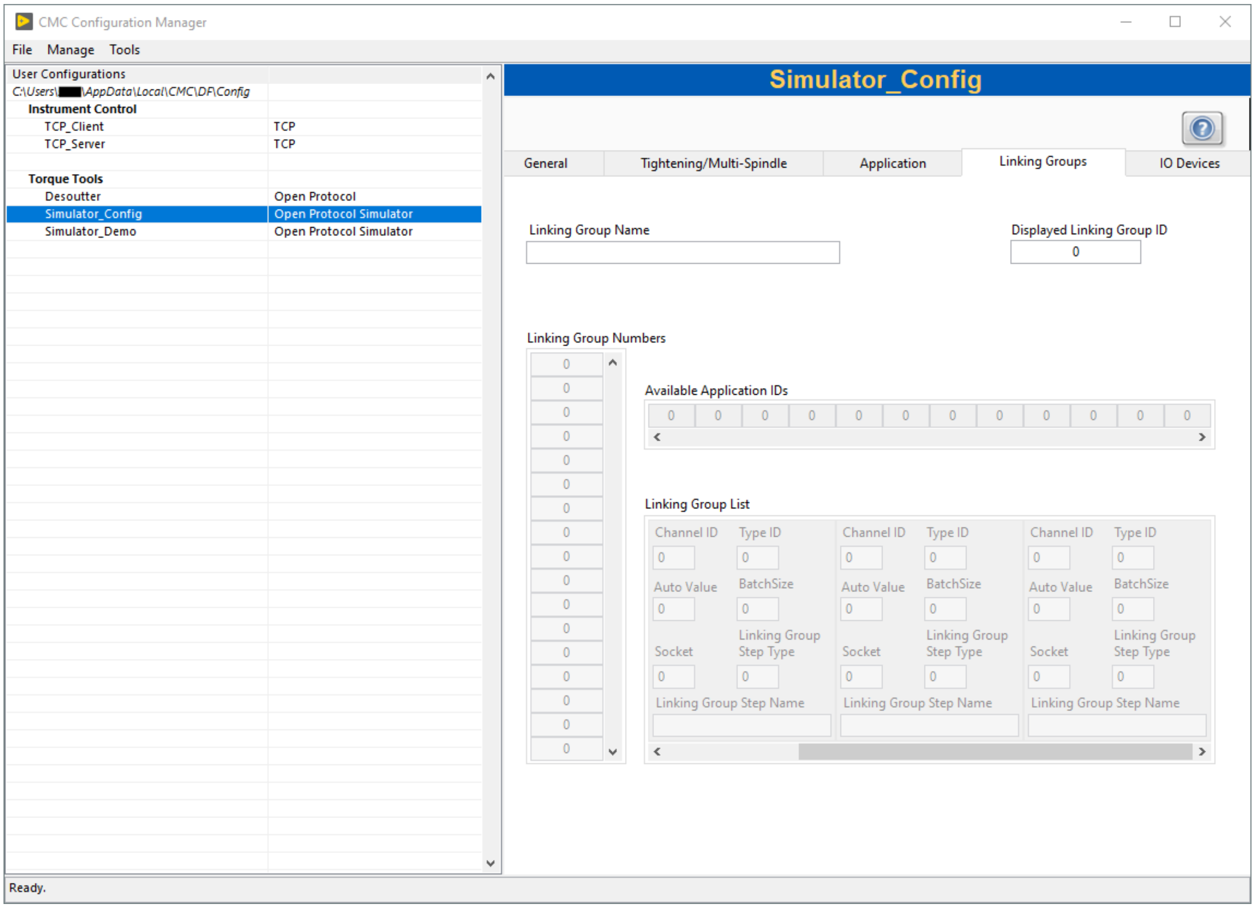
Ready.

Figure 20: Simulated Device Application Configuration

5.3.2.1. Application List

Specifies the application numbers that are installed on the simulated controller. The Last Change Timestamp will be updated when an array element is created new. If the number is later changed, or a different time is desired, the time will need to be modified manually. Valid application numbers are in the range 000-999. When applications are added to the array of applications the Available Application IDs indicator is updated on the Linking Groups configuration page.

5.3.3. Linking Groups Configuration Tab



User Configurations	
C:\Users\[redacted]\AppData\Local\CMC\DF\Config	
Instrument Control	
TCP_Client	TCP
TCP_Server	TCP
Torque Tools	
Desoutter	Open Protocol
Simulator_Config	Open Protocol Simulator
Simulator_Demo	Open Protocol Simulator

Simulator_Config

Linking Group Name:

Displayed Linking Group ID: 0

Linking Group Numbers: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Available Application IDs: 0 0 0 0 0 0 0 0 0 0 0 0

Linking Group List

Channel ID	Type ID	Channel ID	Type ID	Channel ID	Type ID
0	0	0	0	0	0
Auto Value	BatchSize	Auto Value	BatchSize	Auto Value	BatchSize
0	0	0	0	0	0
Socket	Linking Group Step Type	Socket	Linking Group Step Type	Socket	Linking Group Step Type
0	0	0	0	0	0
Linking Group Step Name		Linking Group Step Name		Linking Group Step Name	

Ready.

Figure 21: Simulated Device Linking Group Configuration

5.3.3.1. Linking Group Numbers

A list of the linking groups available on the simulated controller. Add a Linking Group number to the controller by adding a new number into the Linking Group Numbers array. Linking Groups can be removed by right clicking and selecting **Delete Element**. Valid linking group numbers are 00-99.

5.3.3.2. Displayed Liking Group ID

To change the displayed Linking Group, change this control to be one of the numbers in the array of Linking Group Numbers. When the number is entered, that linking group now becomes the active Linking Group on the controller. The name will be updated into the Linking Group Name and any applications that are a part of the Linking Group List will be displayed there.

5.3.3.3. Linking Group Name

This is a 25-character ASCII label that can be set for each linking group but is not necessary to set.

5.3.3.4. Available Application IDs

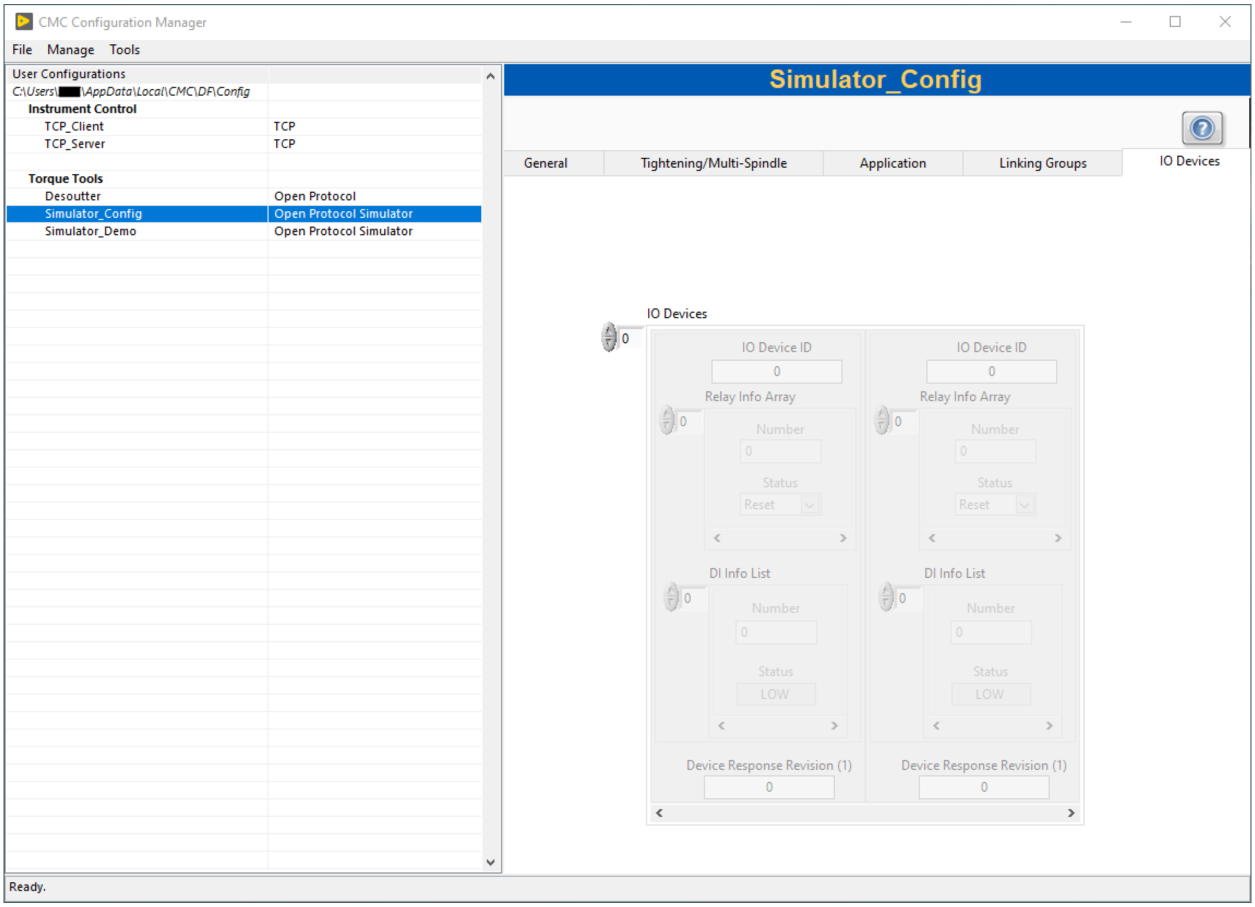
Shows a list of the available application ID numbers. These are the applications that are available to add to the Linking Group List. You can add an application ID to the Linking Group List by right clicking on the desired application **ID** and selecting "Add To Linking Group" and a new entry will be added to the Linking Group List.

5.3.3.5. Linking Group List

The list of application IDs and Channel IDs that make up the specified Linking Group. Additional applications can be added by right clicking on the desired **Available Application ID** or by typing in an available application **ID** into the **App ID** entry of an empty array element. It is possible to add more than one entry for an application ID.

Channel ID is constrained to 00-99; Type ID (Application) is constrained to 000-999; Auto Value is 1 or 0; BatchSize is constrained to 00-99; Socket is constrained to 00-99; Linking Group Step Name is constrained to 0-25 ASCII characters; Linking Group Step Type is constrained to 00-99.

5.3.4. IO Devices Configuration Tab



User Configurations
C:\Users\[redacted]\AppData\Local\CMC\DF\Config

Instrument Control	
TCP_Client	TCP
TCP_Server	TCP
Torque Tools	
Desoutter	Open Protocol
Simulator_Config	Open Protocol Simulator
Simulator_Demo	Open Protocol Simulator

IO Device ID: 0 — Relay Info Array: Number 0, Status Reset — DI Info List: Number 0, Status LOW — Device Response Revision (1): 0

IO Device ID: 0 — Relay Info Array: Number 0, Status Reset — DI Info List: Number 0, Status LOW — Device Response Revision (1): 0

Ready.

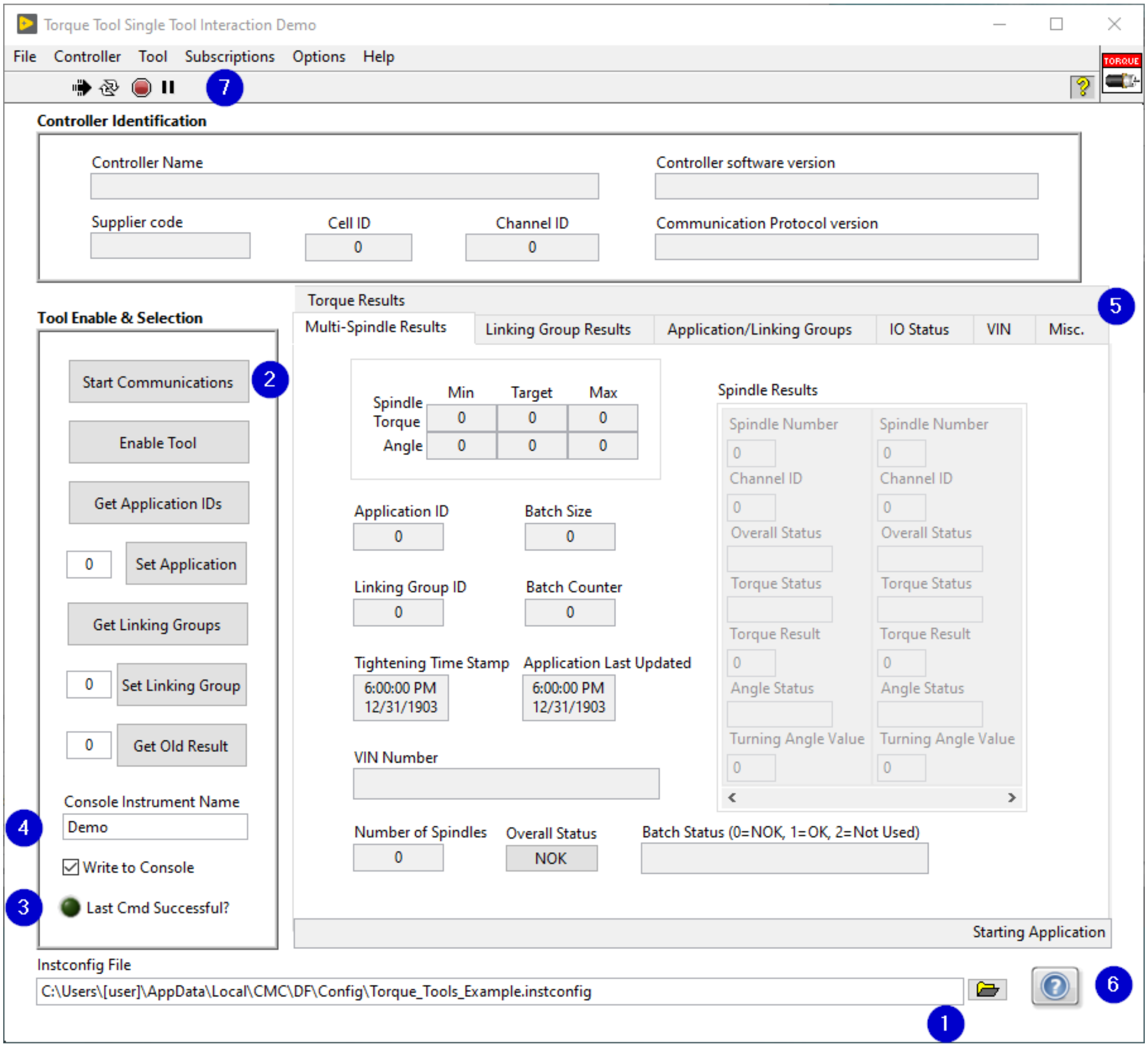
Figure 22: Simulated Device IO Device Configuration

5.3.4.1. IO Devices

An array that simulates IO devices connected to the controller. The device information consists of the IO Device ID, the Relay Info Array and a DI Info List. The Relay Info Array specifies the outputs of the designated Device ID. Each relay has a Number and then a Status of either Set or Reset. The DI Info List returns the Status (Low or High) of the array of inputs specified by Number. DI statuses are generated randomly each time the MID is called to query them.

6. Example

The example that ships with the Torque Tools API is one that allows for interacting with a single torque tool. The example can be run using the **Open Protocol Simulator** instrument or an actual Torque Tool.



Controller Identification

Controller Name

Controller software version

Supplier code

Cell ID 0

Channel ID 0

Communication Protocol version

Tool Enable & Selection

0 Set Application

0 Set Linking Group

0 Get Old Result

Console Instrument Name Demo

☑ Write to Console

● Last Cmd Successful?

	Min	Target	Max
Spindle Torque	0	0	0
Angle	0	0	0

Application ID 0

Batch Size 0

Linking Group ID 0

Batch Counter 0

Tightening Time Stamp 6:00:00 PM 12/31/1903

Application Last Updated 6:00:00 PM 12/31/1903

VIN Number

Spindle Results

Spindle Number 0	Spindle Number 0
Channel ID 0	Channel ID 0
Overall Status	Overall Status
Torque Status	Torque Status
Torque Result 0	Torque Result 0
Angle Status	Angle Status
Turning Angle Value 0	Turning Angle Value 0

Number of Spindles 0

Overall Status NOK

Batch Status (0=NOK, 1=OK, 2=Not Used)

Starting Application

Instconfig File

C:\Users\[user]\AppData\Local\CMC\DF\Config\Torque_Tools_Example.instconfig

Figure 23: Torque Tools Example Front Panel

Instructions for Interfacing to a Torque Tool:

1. Configure the **Torque Tool** hardware to use Open Protocol. Setup the hardware and .instconfig file per Section 5.1.
    a. If using a simulated device, skip to the next step.
2. Use the **CMC Configuration Manager** to create an .instconfig file for the torque tool. Detailed instructions are available in the Configuration Manager application or by selecting **Torque Tools Manual** from the help menu. You can also see Section 5.2 above.
3. Select the .instconfig file you created (Item 1 of *Figure 23*). If left blank, a dialog box will be launched after starting the VI to select an .instconfig file.
4. Select a new name if desired for the **Console Instrument Name** (Item 4 of *Figure 23*).
5. Start/Run the **VI**.
6. Click **Start Communications** to start a session with the tool (Item 2 of *Figure 23*).
7. You will know if a command was successful by looking at the **Last Command Successful? Indicator** (Item 3 of *Figure 23*)

To Subscribe to a Torque Tool event:

1. Complete the steps above to start a communication session.
2. Select the appropriate selection from the **Subscriptions** menu (Item 7 of *Figure 23*).
3. Test the subscription by performing the action using the **Torque Tool**.
    a. If you are using a Simulated Device, there is a background process that is randomly generating messages for subscriptions. Testing is best accomplished by selecting the **Last Tightening Result Data** or the **Multi-Spindle Result Data** from the **Subscriptions** menu. Those are both more likely to change sooner than the other subscriptions.

To See a Torque Result:

1. Complete the steps above selecting **Last Tightening Result Data** from the **Subscriptions** menu.
2. Perform a torque.
3. View the torque results on the **Torque Results** tab.
4. The most recent torque result can be requested by entering “0” in the control next to and then pressing the **Get Old Result** button.

Help is available in the Demo by pressing the **?** button in the lower right-hand corner of the application (Item 6 in *Figure 23*).

Different Views are available by selecting the appropriate tab (Item 5 of *Figure 23*). A brief description of each tab is listed below:

- Torque Results – Displays the information returned from a single tightening result (MID0061)
- Multi-Spidle Results – Displays the information returned from a multi-spindle tightening result (MID0101)
- Linking Group Results – Displays the information returned as part of the Linking Group Info message (MID0035)

- Application/Linking Groups – Displays the Applications and Linking Groups available on the connected torque controller. Applications are returned when requested as MID0010. Linking Groups are returned when requested as part of MID0030.
    - Applications can be requested by pressing the **Get Application IDs** button on the front panel.
    - Linking Group numbers can be requested by pressing the **Get Linking Groups** button on the front panel.
- IO Status – Displays the current status of the IO attached to the controller (MID0215)
- VIN – Allows the user to send a new VIN identifier as a part of “MID0150”. Enter the new VIN in the **VIN for Upload** string control (limited to 25 characters) and press the **Upload VIN** button.
- Misc. – Shows any errors returned by the system, as well as the last raw message received from the torque controller.

7. Troubleshooting Recommendations

See the CMC Driver Framework group on the NI forums for technical support and general discussion. We encourage all of our users to post questions here for the benefit of everyone.
https://forums.ni.com/t5/CMC-Driver-Framework/gp-p/5394

If you cannot post your issue publicly, then contact us at support@choose-mc.com and we can assist you directly.