WOVALAB

CML based on DQMH® framework

Antidoc v2.0.0, Oliver Jourdan

Table of Contents

Chapter 1. Project description

The Continuous Measurement and Logging (CML) Delacor Queued Message Handler (DQMH) sample project is a variation of the NI QMH based CML project using DQMH(R) modules instead of separate Message Handle Loops.

Chapter 2. DQMH® modules

This section describes DQMH® module responsibilities and relationships.

2.1. Preamble

A DQMH module is the main component of an architecture based on DQMH® framework. A DQMH module is used to implement a section of the application that has one responsibility.

DQMH® framework defines two different type of DQMH module.

Singleton:

A Singleton DQMH module can have only one instance running at any given time.

Cloneable:

A Cloneable DQMH module can have one or multiple instances running in parallel.

DQMH® framework defines two different ways to carry data throughout the application and with both other DQMH modules and non-DQMH based code.

Request events:

A request is a code that fires an event requesting the DQMH module to do something. Multiple locations in the code can send events to the DQMH module.

Request events are many-to-one.

Requests are usually named using imperative tense.

Broadcast events:

A broadcast is a code that fires an event broadcasting that the DQMH module did something. Multiple Event Structures can register to handle the Broadcast Events.

Broadcast Events are one-to-many.

Broadcasts are usually named using past tense or passive voice.

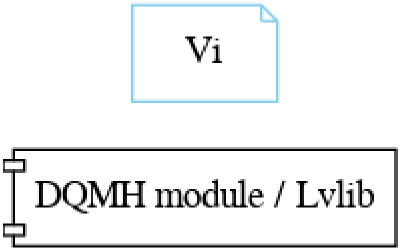
NOTE

Refer to the DQMH® framework official documentation to find more details on how the framework works

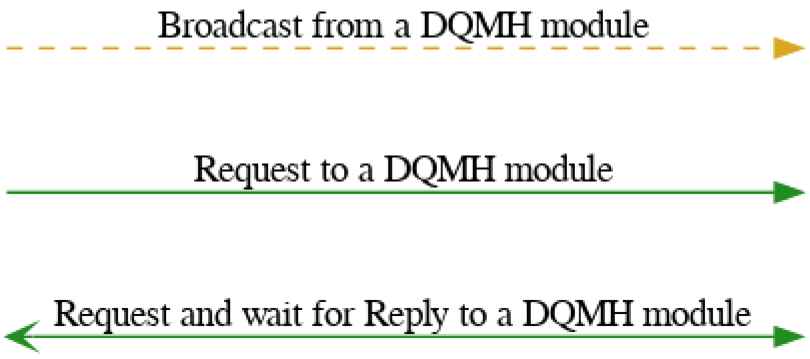
The following section gives you details on the project architecture relying on this framework. It gives you an overview of the modules' interaction and detailed information on each module.

Graphs used in this section have the following legend:

Components:



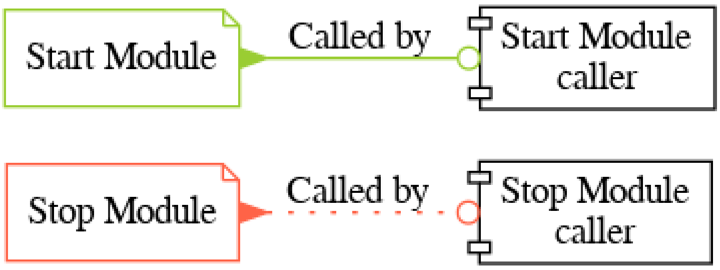
Events:


Broadcast from a DQMH module
Request to a DQMH module
Request and wait for Reply to a DQMH module

NOTE One arrow can represent one or more events between two components

NOTE Request and Request and wait for Reply are represented by only one arrow. If there is no Request and wait for Reply, Request representation is used. Otherwise Request and wait for Reply is used

Start and Stop module callers:



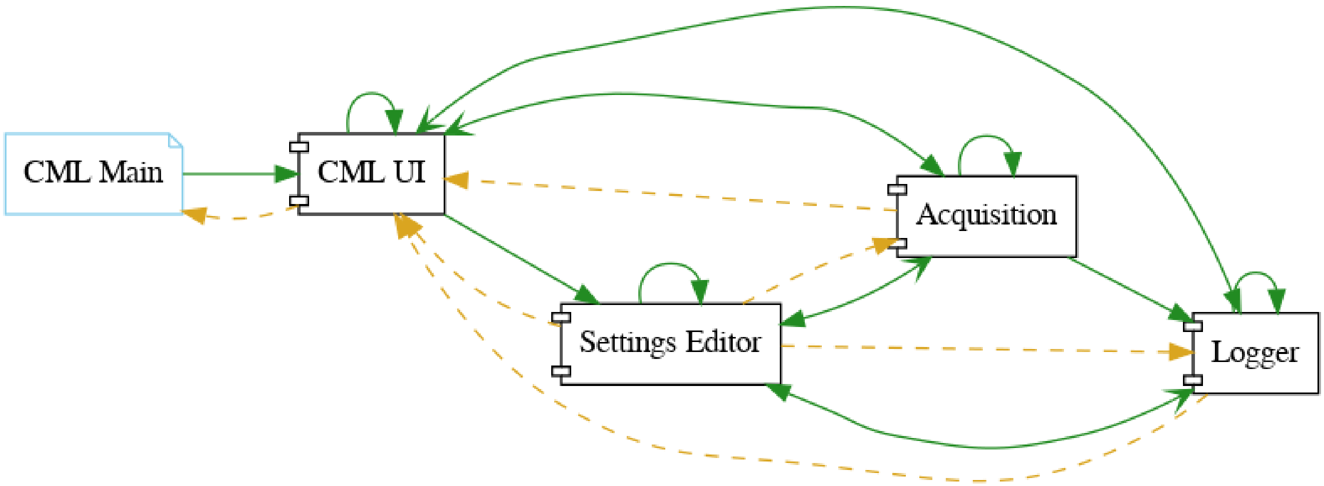
2.2. Modules overview

This project contains 4 singleton modules and 0 cloneable module.

Table 1. Modules list

Singleton	Cloneable
CML UI.lvlib	
Acquisition.lvlib	
Logger.lvlib	
Settings Editor.lvlib	

This graph represents the links between all DQMH modules.



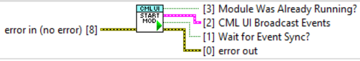
2.3. CML UI.lvlib

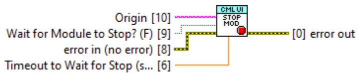
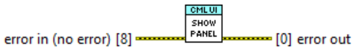
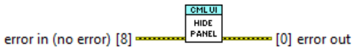
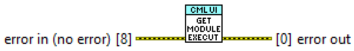
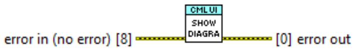
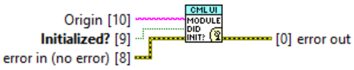
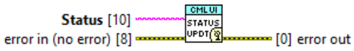
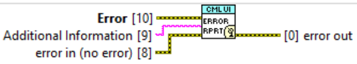
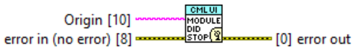
Type: Singleton

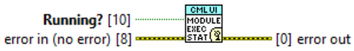
Responsibility: This module is the user interface of the Continuous Measurement and Logging application.

2.3.1. Event list

Table 2. Events

Name	Type	Connector pane	Description	S.	R.	I.
Start Module		 CML UI START MOD error in (no error) [8] [3] Module Was Already Running? [2] CML UI Broadcast Events [1] Wait for Event Sync? [0] error out	Launches the Module Main.vi.			

Name	Type	Connector pane	Description	S.	R.	I.
Stop Module		 Origin [10] Wait for Module to Stop? (F) [9] error in (no error) [8] Timeout to Wait for Stop (s... [6] CML UI STOP MOD [0] error out	Send the Stop request to the Module's Main.vi. If <b>Wait for Module to Stop?</b> is TRUE, this VI will wait until the module main VI stops, and will timeout at the <b>Timeout to Wait for Stop</b> value. This value defaults to "-1", which means the VI will not timeout, and will always wait until the module main VI stops before completing execution. Note: The <b>Timeout to Wait for Stop</b> value is ignored if 'Wait for Module to Stop?' is set to FALSE.			
Show Panel	○→	 error in (no error) [8] CML UI SHOW PANEL [0] error out	Send the Show Panel request to the Module's Main.vi.			
Hide Panel	○→	 error in (no error) [8] CML UI HIDE PANEL [0] error out	Send the Hide Panel request to the Module's Main.vi.			
Get Module Execution Status	○→	 error in (no error) [8] CML UI GET MODULE EXECUT [0] error out	Fire the Get Module Execution Status request.			
Show Diagram	○→	 error in (no error) [8] CML UI SHOW DIAGRA [0] error out	This VI tells the Module to show its block diagram to facilitate troubleshooting (add probes, breakpoints, highlight execution, etc).			
Module Did Init	📡	 Origin [10] Initialized? [9] error in (no error) [8] CML UI MODULE DID INIT? [0] error out	Send the Module Did Init event to any VI registered to listen to this module's broadcast events.			
Status Updated	📡	 Status [10] error in (no error) [8] CML UI STATUS UPDT [0] error out	Send the Status Updated event to any VI registered to listen to events from the owning module.			
Error Reported	📡	 Error [10] Additional Information [9] error in (no error) [8] CML UI ERROR RPRT [0] error out	Send the Error Reported event to any VI registered to listen to events from the owning module.			
Module Did Stop	📡	 Origin [10] error in (no error) [8] CML UI MODULE DID STOP [0] error out	Send the Module Did Stop event to any VI registered to listen to this module's broadcast events.			

Name	Type	Connector pane	Description	S.	R.	I.
Update Module Execution Status		 Running? [10] error in (no error) [8] CML UI MODULE EXEC STAT [0] error out	Broadcast event to specify whether or not the module is running.			

Type: ▯ Request | ▯ Request and Wait for Reply | ▯ Broadcast

Scope: ▯ Protected | ▯ Community

Reentrancy: ▯ Preallocated reentrancy | ▯ Shared reentrancy

Inlining: ▯ Inlined

2.3.2. Module relationship

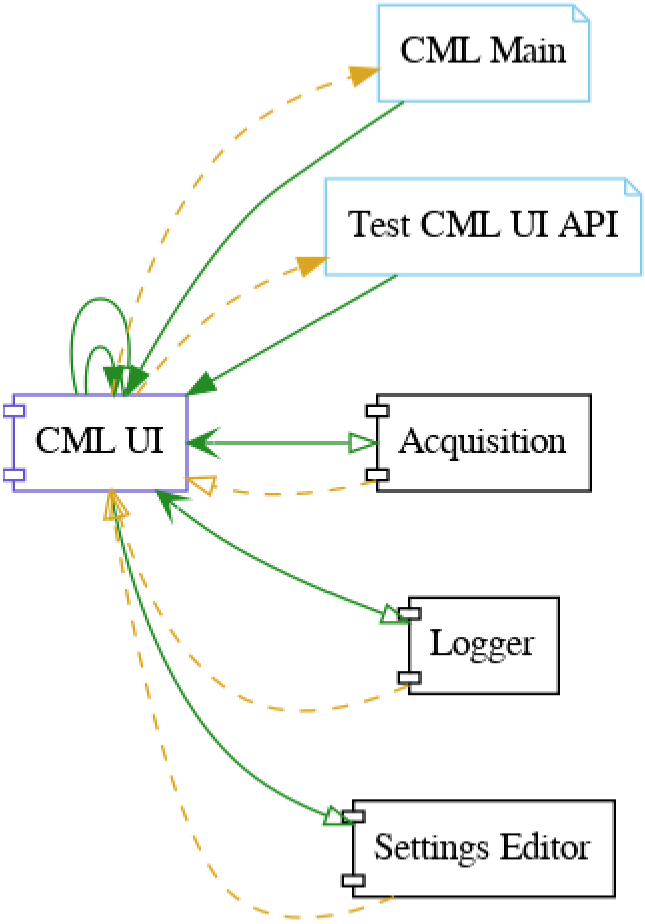


Table 3. Requests callers

Request Name	Callers
Get Module Execution Status	CML UI.lvlib:Obtain Broadcast Events for Registration.vi CML UI.lvlib:Start Module.vi
Hide Panel	Test CML UI API.vi

Request Name	Callers
Show Diagram	Test CML UI API.vi
Show Panel	CML Main.vi Test CML UI API.vi

Table 4. Broadcasts Listeners

Broadcast Name	Listeners
Error Reported	Test CML UI API.vi CML Main.vi
Module Did Init	Test CML UI API.vi CML Main.vi
Module Did Stop	Test CML UI API.vi CML Main.vi
Status Updated	Test CML UI API.vi CML Main.vi
Update Module Execution Status	Test CML UI API.vi CML Main.vi

Table 5. Used requests

Module	Requests
Acquisition.lvlib	Calibrate DAQ.vi Get Module Execution Status.vi Start Acquiring.vi Stop Acquiring.vi Stop Module.vi (2)
CML UI.lvlib	Stop Module.vi
Logger.lvlib	Get Module Execution Status.vi Initialize File.vi Stop Logging.vi Stop Module.vi (2)
Settings Editor.lvlib	Get Module Execution Status.vi Show Panel.vi Stop Module.vi (2)

Table 6. Registered broadcast

Module	Broadcasts
Acquisition.lvlib	Acquisition Started.vi Acquisition Stopped.vi Data Updated.vi Device Calibrated.vi Error Reported.vi Module Did Init.vi Status Updated.vi Update Module Execution Status.vi
Logger.lvlib	Error Reported.vi Module Did Init.vi Status Updated.vi Update Module Execution Status.vi
Settings Editor.lvlib	Error Reported.vi Module Did Init.vi Status Updated.vi Update Module Execution Status.vi

2.3.3. Module Start/Stop calls

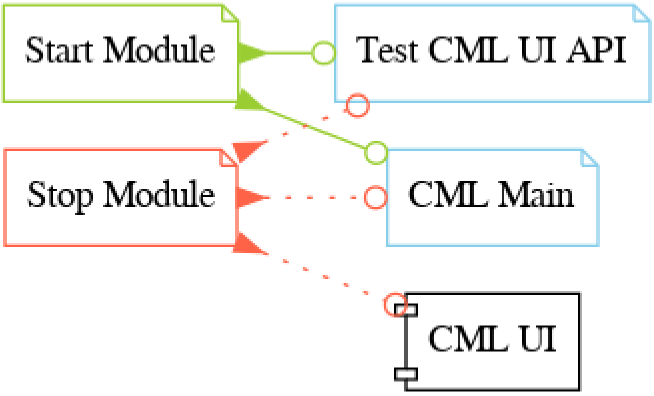


Table 7. Start and Stop module callers

Function	Callers
Start Module	CML Main.vi Test CML UI API.vi
Stop Module	CML UI.lvlib:Handle Exit.vi CML Main.vi Test CML UI API.vi

2.3.4. Module custom errors

TIP Custom errors are added to the module via vi named `*--error.vi`.

Module CML UI.lvlib use the following custom errors:

Table 8. Custom errors

Name	Code	Description
Module Not Running	403681	%s Module is not running.
Module Not Stopped	403682	The Stop Module VI for the %s module timed out while waiting for the module main VI to stop. The module main VI may still be running.
Module Not Synced	403683	%s Module was unable to synchronize events.

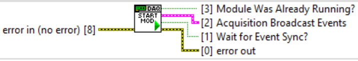
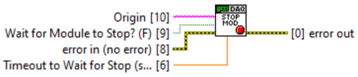
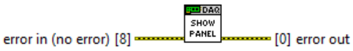
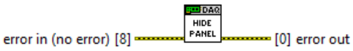
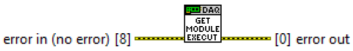
2.4. Acquisition.lvlib

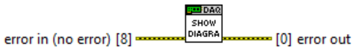
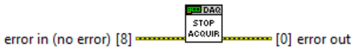
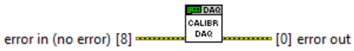
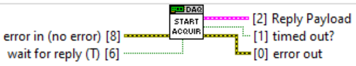
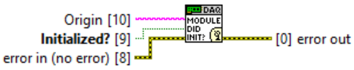
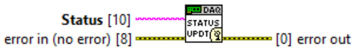
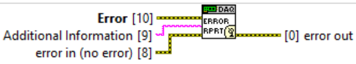
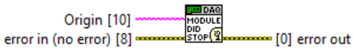
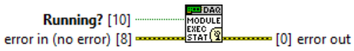
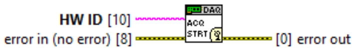
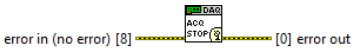
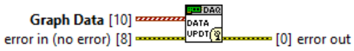
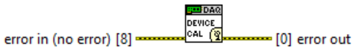
Type: Singleton

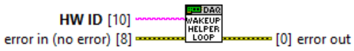
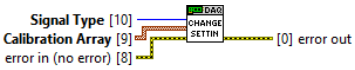
Responsibility: This module handle the continuous data acquisition.

2.4.1. Event list

Table 9. Events

Name	Type	Connector pane	Description	S.	R.	I.
Start Module		 error in (no error) [8] DAQ START MOD [3] Module Was Already Running? [2] Acquisition Broadcast Events [1] Wait for Event Sync? [0] error out	Launches the Module Main.vi.			
Stop Module		 Origin [10] Wait for Module to Stop? (F) [9] error in (no error) [8] Timeout to Wait for Stop (s... [6] DAQ STOP MOD [0] error out	Send the Stop request to the Module's Main.vi. If <b>Wait for Module to Stop?</b> is TRUE, this VI will wait until the module main VI stops, and will timeout at the <b>Timeout to Wait for Stop</b> value. This value defaults to "-1", which means the VI will not timeout, and will always wait until the module main VI stops before completing execution. Note: The <b>Timeout to Wait for Stop</b> value is ignored if 'Wait for Module to Stop?' is set to FALSE.			
Show Panel	○→	 error in (no error) [8] DAQ SHOW PANEL [0] error out	Send the Show Panel request to the Module's Main.vi.			
Hide Panel	○→	 error in (no error) [8] DAQ HIDE PANEL [0] error out	Send the Hide Panel request to the Module's Main.vi.			
Get Module Execution Status	○→	 error in (no error) [8] DAQ GET MODULE EXECUT [0] error out	Fire the Get Module Execution Status request.			

Name	Type	Connector pane	Description	S.	R.	I.
Show Diagram	○→	 DAQ SHOW DIAGRA error in (no error) [8] [0] error out	This VI tells the Module to show its block diagram to facilitate troubleshooting (add probes, breakpoints, highlight execution, etc).			
Stop Acquiring	○→	 DAQ STOP ACQUIR error in (no error) [8] [0] error out	Requests the Acquisition Module to stop acquiring			
Calibrate DAQ	○→	 DAQ CALIBR DAQ error in (no error) [8] [0] error out	Add calibration routine, and linear slope and offset to the acquisition helper loop.			
Start Acquiring	○→ ⟲	 DAQ START ACQUIR error in (no error) [8] wait for reply (T) [6] [2] Reply Payload [1] timed out? [0] error out	<b>Note</b>: This VI was renamed by the DQMH Rename Event utility. Make sure the VI Description is updated to reflect the new event name, then delete this comment. Requests Acquisition Module to start acquiring.			
Module Did Init	📡	 DAQ MODULE DID INIT? Origin [10] Initialized? [9] error in (no error) [8] [0] error out	Send the Module Did Init event to any VI registered to listen to this module's broadcast events.			
Status Updated	📡	 DAQ STATUS UPDT Status [10] error in (no error) [8] [0] error out	Send the Status Updated event to any VI registered to listen to events from the owning module.			
Error Reported	📡	 DAQ ERROR RPRT Error [10] Additional Information [9] error in (no error) [8] [0] error out	Send the Error Reported event to any VI registered to listen to events from the owning module.			
Module Did Stop	📡	 DAQ MODULE DID STOP Origin [10] error in (no error) [8] [0] error out	Send the Module Did Stop event to any VI registered to listen to this module's broadcast events.			
Update Module Execution Status	📡	 DAQ MODULE EXEC STAT Running? [10] error in (no error) [8] [0] error out	Broadcast event to specify whether or not the module is running.			
Acquisition Started	📡	 DAQ ACQ STRT HW ID [10] error in (no error) [8] [0] error out	Broadcasts that the Acquisition Module started acquiring			
Acquisition Stopped	📡		Broadcasts that the Acquisition Module has stopped acquiring data.			
Data Updated	📡	 DAQ DATA UPDT Graph Data [10] error in (no error) [8] [0] error out	Broadcasts the latest data acquired			
Device Calibrated	📡	 DAQ DEVICE CAL error in (no error) [8] [0] error out	Broadcasts that the Acquisition module calibrated the device.			

Name	Type	Connector pane	Description	S.	R.	I.
Wakeup Helper Loop	o→	 HW ID [10] error in (no error) [8] DAQ WAKEUP HELPER LOOP [0] error out	This is a private request to wake up the Helper Loop and start acquiring.			
Change Settings	o→	 Signal Type [10] Calibration Array [9] error in (no error) [8] DAQ CHANGE SETTIN [0] error out	Fire an event to change the hardware settings in the helper loop. This is a private request event that should only be fired from within the Acquisition Module. If this request was public, other modules could call it and the Settings Editor would not get the notification that the values changed.			

Type: o→ ▯ Request | o→ ▯ Request and Wait for Reply | ▯ Broadcast

Scope: ▯ Protected | ▯ Community

Reentrancy: P ▯ Preallocated reentrancy | S ▯ Shared reentrancy

Inlining: ▯ Inlined

2.4.2. Module relationship

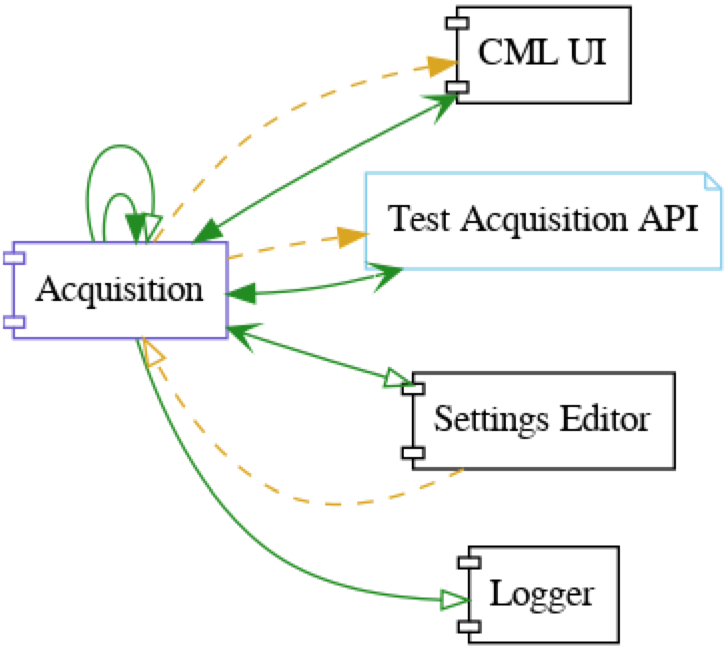


Table 10. Requests callers

Request Name	Callers
Calibrate DAQ	CML UI.lvlib:Main.vi Test Acquisition API.vi
Change Settings	Acquisition.lvlib:Main.vi

Request Name	Callers
Get Module Execution Status	Acquisition.lvlib:Obtain Broadcast Events for Registration.vi Acquisition.lvlib:Start Module.vi
Hide Panel	Test Acquisition API.vi
Show Diagram	Test Acquisition API.vi
Show Panel	Test Acquisition API.vi
Start Acquiring	CML UI.lvlib:Main.vi Test Acquisition API.vi
Stop Acquiring	CML UI.lvlib:Main.vi Test Acquisition API.vi
Wakeup Helper Loop	Acquisition.lvlib:Main.vi

Table 11. Broadcasts Listeners

Broadcast Name	Listeners
Acquisition Started	CML UI.lvlib:Main.vi Test Acquisition API.vi
Acquisition Stopped	CML UI.lvlib:Main.vi Test Acquisition API.vi
Data Updated	CML UI.lvlib:Main.vi Test Acquisition API.vi
Device Calibrated	CML UI.lvlib:Main.vi Test Acquisition API.vi
Error Reported	CML UI.lvlib:Main.vi Test Acquisition API.vi
Module Did Init	CML UI.lvlib:Main.vi Test Acquisition API.vi
Module Did Stop	Test Acquisition API.vi
Status Updated	CML UI.lvlib:Main.vi Test Acquisition API.vi
Update Module Execution Status	CML UI.lvlib:Main.vi Test Acquisition API.vi

Table 12. Used requests

Module	Requests
Acquisition.lvlib	Change Settings.vi Stop Module.vi Wakeup Helper Loop.vi
Logger.lvlib	Log Data.vi

Module	Requests
Settings Editor.lvlib	Get Module Execution Status.vi Stop Module.vi (2) Update Application Settings.vi

Table 13. Registered broadcast

Module	Broadcasts
Settings Editor.lvlib	Application Settings Updated.vi

2.4.3. Module Start/Stop calls

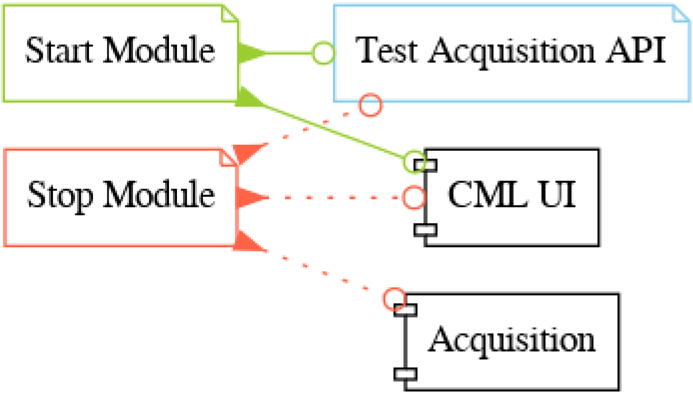


Table 14. Start and Stop module callers

Function	Callers
Start Module	CML UI.lvlib:Main.vi Test Acquisition API.vi
Stop Module	CML UI.lvlib:Main.vi Acquisition.lvlib:Handle Exit.vi Test Acquisition API.vi

2.4.4. Module custom errors

TIP Custom errors are added to the module via vi named `*--error.vi`.

Module Acquisition.lvlib use the following custom errors:

Table 15. Custom errors

Name	Code	Description
Module Not Running	403681	%s Module is not running.
Module Not Stopped	403682	The Stop Module VI for the %s module timed out while waiting for the module main VI to stop. The module main VI may still be running.
Module Not Synced	403683	%s Module was unable to synchronize events.

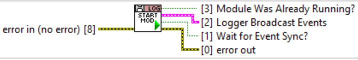
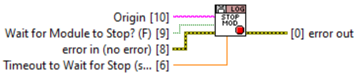
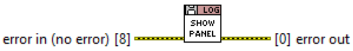
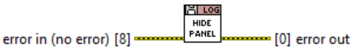
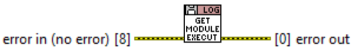
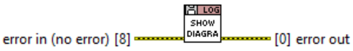
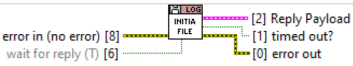
2.5. Logger.lvlib

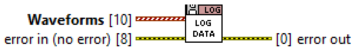
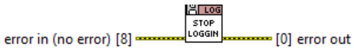
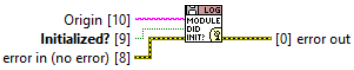
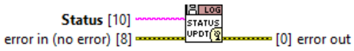
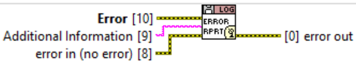
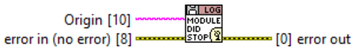
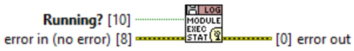
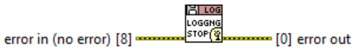
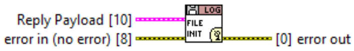
Type: Singleton

Responsibility: This module handle data logging in file.

2.5.1. Event list

Table 16. Events

Name	Type	Connector pane	Description	S.	R.	I.
Start Module		 LOG START MOD error in (no error) [8] [3] Module Was Already Running? [2] Logger Broadcast Events [1] Wait for Event Sync? [0] error out	Launches the Module Main.vi.			
Stop Module		 LOG STOP MOD Origin [10] Wait for Module to Stop? (F) [9] error in (no error) [8] Timeout to Wait for Stop (s... [6] [0] error out	Send the Stop request to the Module's Main.vi. If <b>Wait for Module to Stop?</b> is TRUE, this VI will wait until the module main VI stops, and will timeout at the <b>Timeout to Wait for Stop</b> value. This value defaults to "-1", which means the VI will not timeout, and will always wait until the module main VI stops before completing execution. Note: The <b>Timeout to Wait for Stop</b> value is ignored if 'Wait for Module to Stop?' is set to FALSE.			
Show Panel	○→	 LOG SHOW PANEL error in (no error) [8] [0] error out	Send the Show Panel request to the Module's Main.vi.			
Hide Panel	○→	 LOG HIDE PANEL error in (no error) [8] [0] error out	Send the Hide Panel request to the Module's Main.vi.			
Get Module Execution Status	○→	 LOG GET MODULE EXECUT error in (no error) [8] [0] error out	Fire the Get Module Execution Status request.			
Show Diagram	○→	 LOG SHOW DIAGRA error in (no error) [8] [0] error out	This VI tells the Module to show its block diagram to facilitate troubleshooting (add probes, breakpoints, highlight execution, etc).			
Initialize File	○→ ↺	 LOG INITIA FILE error in (no error) [8] wait for reply (T) [6] [2] Reply Payload [1] timed out? [0] error out	Request the Logger to initialize the file and report when the Logger has initialized the file.			

Name	Type	Connector pane	Description	S.	R.	I.
Log Data	○→	 Waveforms [10] error in (no error) [8] LOG LOG DATA [0] error out	<b>Note</b>: This VI was renamed by the DQMH Rename Event utility. Make sure the VI Description is updated to reflect the new event name, then delete this comment. Requests Logger module to log data to file			
Stop Logging	○→	 error in (no error) [8] LOG STOP LOGGIN [0] error out	Requests Logger module to stop logging and the module broadcasts when the logging has stopped.			
Module Did Init	📡	 Origin [10] Initialized? [9] error in (no error) [8] LOG MODULE DID INIT? [0] error out	Send the Module Did Init event to any VI registered to listen to this module's broadcast events.			
Status Updated	📡	 Status [10] error in (no error) [8] LOG STATUS UPDT [0] error out	Send the Status Updated event to any VI registered to listen to events from the owning module.			
Error Reported	📡	 Error [10] Additional Information [9] error in (no error) [8] LOG ERROR RPRT [0] error out	Send the Error Reported event to any VI registered to listen to events from the owning module.			
Module Did Stop	📡	 Origin [10] error in (no error) [8] LOG MODULE DID STOP [0] error out	Send the Module Did Stop event to any VI registered to listen to this module's broadcast events.			
Update Module Execution Status	📡	 Running? [10] error in (no error) [8] LOG MODULE EXEC STAT [0] error out	Broadcast event to specify whether or not the module is running.			
Logging Stopped	📡	 error in (no error) [8] LOG LOGGNG STOP [0] error out	Requests Logger module to stop logging and the module broadcasts when the logging has stopped.			
File Initialized	📡	 Reply Payload [10] error in (no error) [8] LOG FILE INIT [0] error out	Request the Logger to initialize the file and report when the Logger has initialized the file.			

Type: ○→ ▯ Request | ⇄ ▯ Request and Wait for Reply | 📡 ▯ Broadcast

Scope: 🔑 ▯ Protected | 🔑 ▯ Community

Reentrancy: 🄿 ▯ Preallocated reentrancy | 🅂 ▯ Shared reentrancy

Inlining: ⇉ ▯ Inlined

2.5.2. Module relationship

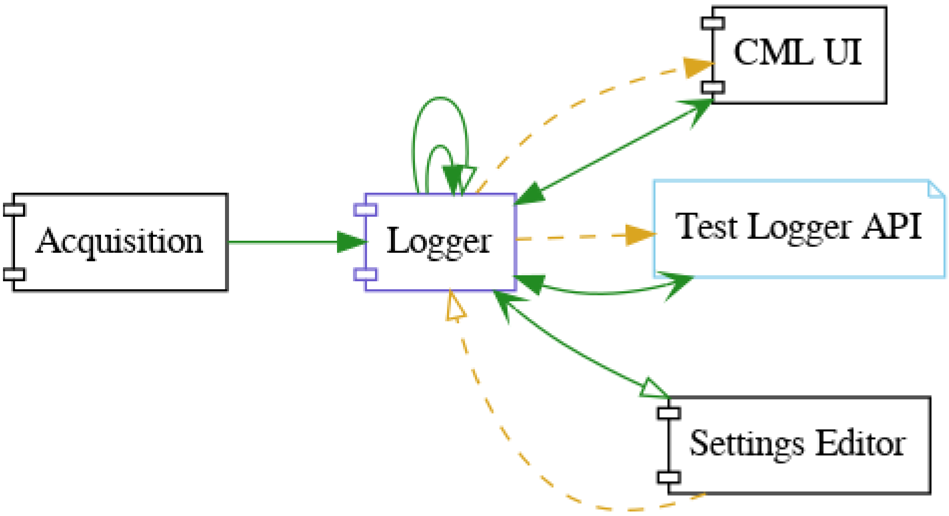


Table 17. Requests callers

Request Name	Callers
Get Module Execution Status	Logger.lvlib:Obtain Broadcast Events for Registration.vi Logger.lvlib:Start Module.vi
Hide Panel	Test Logger API.vi
Initialize File	CML UI.lvlib:Main.vi Test Logger API.vi
Log Data	Acquisition.lvlib:Main.vi Test Logger API.vi
Show Diagram	Test Logger API.vi
Show Panel	Test Logger API.vi
Stop Logging	CML UI.lvlib:Main.vi Test Logger API.vi

Table 18. Broadcasts Listeners

Broadcast Name	Listeners
Error Reported	CML UI.lvlib:Main.vi Test Logger API.vi
File Initialized	Test Logger API.vi
Logging Stopped	Test Logger API.vi
Module Did Init	CML UI.lvlib:Main.vi Test Logger API.vi
Module Did Stop	Test Logger API.vi
Status Updated	CML UI.lvlib:Main.vi Test Logger API.vi

Broadcast Name	Listeners
Update Module Execution Status	CML UI.lvlib:Main.vi Test Logger API.vi

Table 19. Used requests

Module	Requests
Logger.lvlib	Stop Module.vi
Settings Editor.lvlib	Get Module Execution Status.vi Stop Module.vi (2) Update Application Settings.vi

Table 20. Registered broadcast

Module	Broadcasts
Settings Editor.lvlib	Application Settings Updated.vi

2.5.3. Module Start/Stop calls

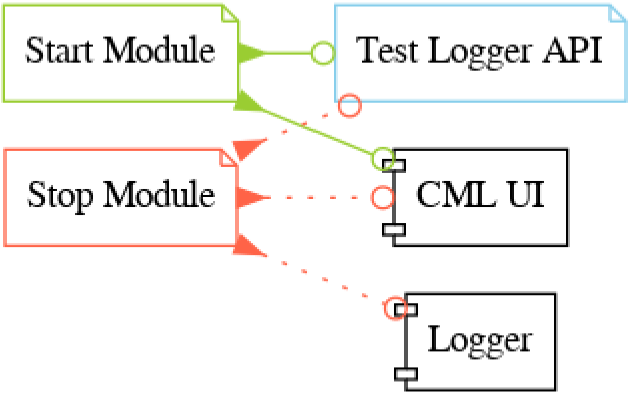


Table 21. Start and Stop module callers

Function	Callers
Start Module	CML UI.lvlib:Main.vi Test Logger API.vi
Stop Module	CML UI.lvlib:Main.vi Logger.lvlib:Handle Exit.vi Test Logger API.vi

2.5.4. Module custom errors

TIP Custom errors are added to the module via vi named `*--error.vi`.

Module Logger.lvlib use the following custom errors:

Table 22. Custom errors

Name	Code	Description
Module Not Running	403681	%s Module is not running.
Module Not Stopped	403682	The Stop Module VI for the Logger module timed out while waiting for the module main VI to stop. The module main VI may still be running.
Module Not Synced	403683	%s Module was unable to synchronize events.

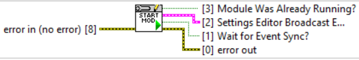
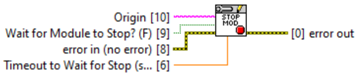
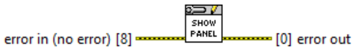
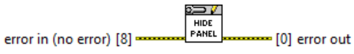
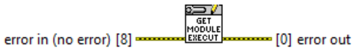
2.6. Settings Editor.lvlib

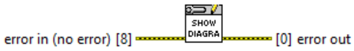
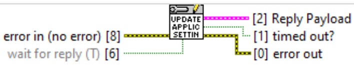
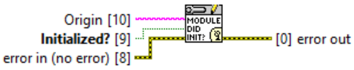
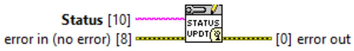
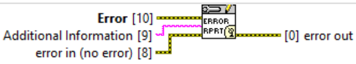
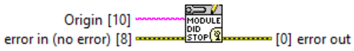
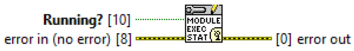
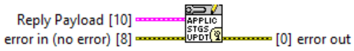
Type: Singleton

Responsibility: This module is an ui that allow user to change application settings.

2.6.1. Event list

Table 23. Events

Name	Type	Connector pane	Description	S.	R.	I.
Start Module		 error in (no error) [8] START MOD [3] Module Was Already Running? [2] Settings Editor Broadcast E... [1] Wait for Event Sync? [0] error out	Launches the Module Main.vi.			
Stop Module		 Origin [10] Wait for Module to Stop? (F) [9] error in (no error) [8] Timeout to Wait for Stop (s... [6] STOP MOD [0] error out	Send the Stop request to the Module's Main.vi. If <b>Wait for Module to Stop?</b> is TRUE, this VI will wait until the module main VI stops, and will timeout at the <b>Timeout to Wait for Stop</b> value. This value defaults to "-1", which means the VI will not timeout, and will always wait until the module main VI stops before completing execution. Note: The <b>Timeout to Wait for Stop</b> value is ignored if 'Wait for Module to Stop?' is set to FALSE.			
Show Panel	○→		Send the Show Panel request to the Module's Main.vi.			
Hide Panel	○→		Send the Hide Panel request to the Module's Main.vi.			
Get Module Execution Status	○→	 error in (no error) [8] GET MODULE EXECUT [0] error out	Fire the Get Module Execution Status request.			

Name	Type	Connector pane	Description	S.	R.	I.
Show Diagram	○→		This VI tells the Module to show its block diagram to facilitate troubleshooting (add probes, breakpoints, highlight execution, etc).			
Update Application Settings	○→↻	 error in (no error) [8] wait for reply (T) [6] UPDATE APPLIC SETTIN [2] Reply Payload [1] timed out? [0] error out	Request Settings Editor to return the latest settings. The Settings Editor can send the applications settings as a reply or via broadcast.			
Module Did Init	📡	 Origin [10] Initialized? [9] error in (no error) [8] MODULE DID INIT? [0] error out	Send the Module Did Init event to any VI registered to listen to this module's broadcast events.			
Status Updated	📡	 Status [10] error in (no error) [8] STATUS UPDT [0] error out	Send the Status Updated event to any VI registered to listen to events from the owning module.			
Error Reported	📡	 Error [10] Additional Information [9] error in (no error) [8] ERROR RPRT [0] error out	Send the Error Reported event to any VI registered to listen to events from the owning module.			
Module Did Stop	📡	 Origin [10] error in (no error) [8] MODULE DID STOP [0] error out	Send the Module Did Stop event to any VI registered to listen to this module's broadcast events.			
Update Module Execution Status	📡	 Running? [10] error in (no error) [8] MODULE EXEC STAT [0] error out	Broadcast event to specify whether or not the module is running.			
Application Settings Updated	📡	 Reply Payload [10] error in (no error) [8] APPLIC STGS UPDT [0] error out	Request Settings Editor to return the latest settings. The Settings Editor can send the applications settings as a reply or via broadcast.			

Type: ○→ ▯ Request | ○→↻ ▯ Request and Wait for Reply | 📡 ▯ Broadcast

Scope: 🔑 ▯ Protected | 🔑 ▯ Community

Reentrancy: P ▯ Preallocated reentrancy | S ▯ Shared reentrancy

Inlining: ⇉ ▯ Inlined

2.6.2. Module relationship

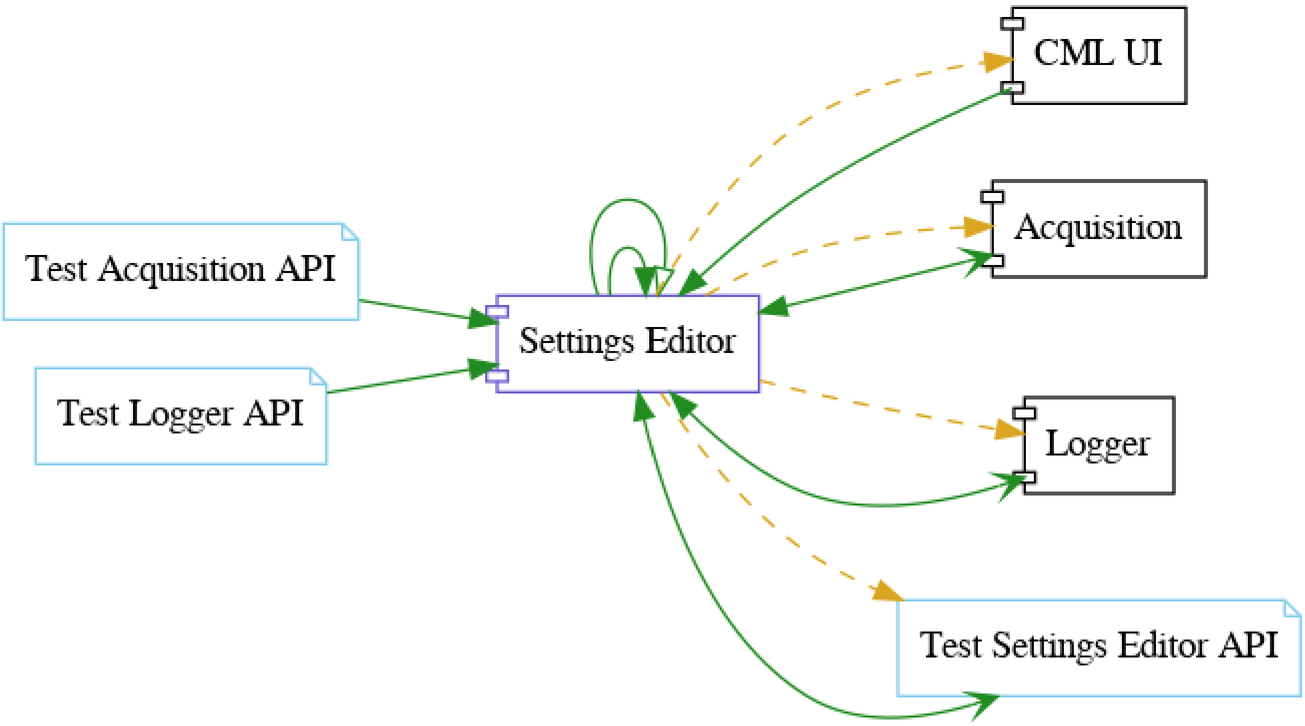


Table 24. Requests callers

Request Name	Callers
Get Module Execution Status	Settings Editor.lvlib:Obtain Broadcast Events for Registration.vi Settings Editor.lvlib:Start Module.vi
Hide Panel	Test Settings Editor API.vi
Show Diagram	Test Settings Editor API.vi
Show Panel	CML UI.lvlib:Main.vi Test Acquisition API.vi Test Logger API.vi Test Settings Editor API.vi
Update Application Settings	Acquisition.lvlib:Main.vi Logger.lvlib:Main.vi Test Settings Editor API.vi

Table 25. Broadcasts Listeners

Broadcast Name	Listeners
Application Settings Updated	Acquisition.lvlib:Main.vi Logger.lvlib:Main.vi Test Settings Editor API.vi
Error Reported	CML UI.lvlib:Main.vi Test Settings Editor API.vi
Module Did Init	CML UI.lvlib:Main.vi Test Settings Editor API.vi

Broadcast Name	Listeners
Module Did Stop	Test Settings Editor API.vi
Status Updated	CML UI.lvlib:Main.vi Test Settings Editor API.vi
Update Module Execution Status	CML UI.lvlib:Main.vi Test Settings Editor API.vi

Table 26. Used requests

Module	Requests
Settings Editor.lvlib	Stop Module.vi

Table 27. Registered broadcast

Module	Broadcasts
—	—

2.6.3. Module Start/Stop calls

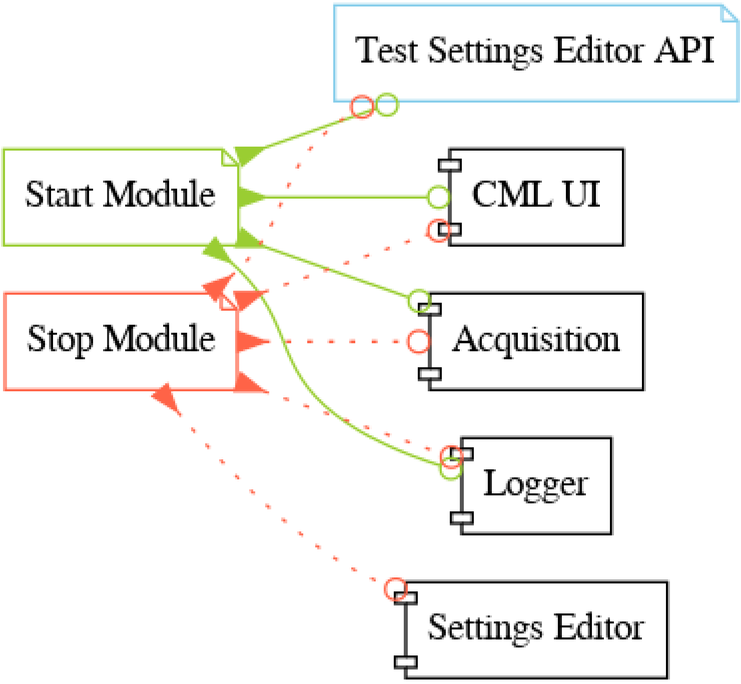


Table 28. Start and Stop module callers

Function	Callers
Start Module	CML UI.lvlib:Main.vi Acquisition.lvlib:Main.vi Logger.lvlib:Main.vi Test Settings Editor API.vi

Function	Callers
Stop Module	CML UI.lvlib:Main.vi Settings Editor.lvlib:Handle Exit.vi Acquisition.lvlib:Main.vi Logger.lvlib:Main.vi Test Settings Editor API.vi

2.6.4. Module custom errors

TIP Custom errors are added to the module via vi named `*--error.vi`.

Module Settings Editor.lvlib use the following custom errors:

Table 29. Custom errors

Name	Code	Description
Module Not Running	403681	%s Module is not running.
Module Not Stopped	403682	The Stop Module VI for the %s module timed out while waiting for the module main VI to stop. The module main VI may still be running.
Module Not Synced	403683	%s Module was unable to synchronize events.

Chapter 3. Libraries

This section describes the libraries contained in the project.

3.1. CML Shared.lvlib

Responsibility: This library gathers all the resource shared among the different part of the code.

Version: 1.0.0.0

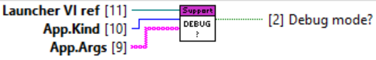
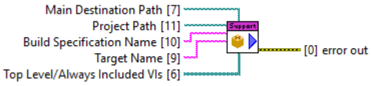
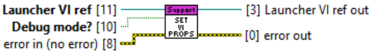
This library has no functions set to non private scope.

3.2. Launcher Support.lvlib

Responsibility: This library gathers functions used to build and laucnh the application.

Version: 1.0.0.0

Table 30. Functions (non private scope only)

Name	Connector pane	Description	S.	R.	I.
Determine if Running in Debug Mode	 Launcher VI ref [11] App.Kind [10] App.Args [9] Support DEBUG ? [2] Debug mode?	The launcher VI is meant to be run as a headless launcher for the module main VI. The launcher VI can be used as a debugging tool. This VI determines if the launcher VI is running as a debugger by parsing the command line arguments or checking if the VI is running in development mode and sets the Launcher VI properties accordingly.			
Pre-Build Action	 Main Destination Path [7] Project Path [11] Build Specification Name [10] Target Name [9] Top Level/Always Included VIs [6] Support [0] error out	This Pre-Build Action VI sets the Debug mode to false to ensure the top level VI has the headless properties at build time.			
Set VI Properties for Debugging Mode	 Launcher VI ref [11] Debug mode? [10] error in (no error) [8] Support SET VI PROPS [3] Launcher VI ref out [0] error out	This VI sets the properties of the VI Launcher to either headless or debugging mode.			

Scope: 🔑 ▯ Protected | 🔑 ▯ Community

Reentrancy: P ▯ Preallocated reentrancy | S ▯ Shared reentrancy

Inlining: ⇉ ▯ Inlined

Chapter 4. Custom errors

TIP Custom errors are added via vi named `*--error.vi`.

Table 31. Custom errors

Name	Code	Description	Owned by
Module Not Running	403681	%s Module is not running.	CML UI.lvlib Acquisition.lvlib Logger.lvlib Settings Editor.lvlib
Module Not Stopped	403682	The Stop Module VI for the %s module timed out while waiting for the module main VI to stop. The module main VI may still be running.	CML UI.lvlib Acquisition.lvlib Settings Editor.lvlib
Module Not Stopped	403682	The Stop Module VI for the Logger module timed out while waiting for the module main VI to stop. The module main VI may still be running.	Logger.lvlib
Module Not Synced	403683	%s Module was unable to synchronize events.	CML UI.lvlib Acquisition.lvlib Logger.lvlib Settings Editor.lvlib

Chapter 5. Legal Information

5.1. Document creation

This document has been generated using the following tools.

5.1.1. Antidoc

Project website: Antidoc

Maintainer website: Wovalab

BSD 3-Clause License

Copyright © 2019, Wovalab, All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5.1.2. Asciidoc for LabVIEW™

Project website: Asciidoc toolkit

Maintainer website: Wovalab

BSD 3-Clause License

Copyright © 2019, Wovalab, All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided

5.1.3. Graph Builder

Project website: Graph Builder

BSD 3-Clause License

THE POSSIBILITY OF SUCH DAMAGE.

5.1.4. classy Diagram Viewer

Project website: classy Diagram Viewer

5.2. Product used in the project

The documented project has been developed with the following products.

5.2.1. DQMH®

Find more details on DQMH® Consortium website