

# **HAMPEL SOFTWARE ENGINEERING**

**TEMPLATE**

**HAMPEL SOFTWARE ENGINEERING**

**Version 3.5.0 (2026-05-11)**

# TABLE OF CONTENTS

|  |    |
|--|----|
| 1. Project Description .....             | 2  |
| 1.1. UI Framework .....                  | 2  |
| 1.2. Project-Specific DQMH Modules ..... | 3  |
| 2. State Machines .....                  | 4  |
| 2.1. StateMachine .....                  | 4  |
| 3. Architecture .....                    | 5  |
| 3.1. Overview .....                      | 5  |
| 3.2. Callers .....                       | 6  |
| 3.3. Listeners .....                     | 6  |
| Appendix A: DQMH .....                   | 7  |
| A.1. ActorModule.lvlib .....             | 7  |
| A.2. DBModule.lvlib .....                | 10 |
| A.3. Dummy.lvlib .....                   | 13 |
| A.4. ExampleClone.lvlib .....            | 17 |
| A.5. StateMachine.lvlib .....            | 19 |
| Appendix B: Libraries .....              | 23 |
| B.1. HSE-State-Machine.lvlib .....       | 23 |
| B.2. Project.lvlib .....                 | 24 |
| Appendix C: Classes .....                | 26 |
| C.1. Classes overview .....              | 26 |
| Appendix D: Custom Errors .....          | 27 |
| D.1. Custom errors .....                 | 27 |
| Glossary .....                           | 29 |



Document generated automatically!

This document was created fully automated from the actual LabVIEW Source Code of this project using the [Release Automation Tools](#) of [Hampel Software Engineering](#).

The Release Automation Tools (RAT) help automate the validating, testing, documenting, building, packaging and publishing of your projects. Built-in support for Git lets you trigger our tools from your repository, via GitLab CI/CD or Azure DevOps amongst others.

For a more detailed overview of what these tools do, see <https://rat.hampel-soft.com/>, where you can find information on the available tools, how we automate them using GitLab CI, when the next scheduled webinars are on, and how you can run those tools on your own servers using a commercial license for RAT.

# CHAPTER 1. PROJECT DESCRIPTION

This application template showcases our UI framework for applications with graphical user interfaces. We use this template as a container which loads the project-specific DQMH modules dynamically from a configuration file.

## 1.1. UI FRAMEWORK

The UI framework helps us with:

- reusing UI management code
- project-specific UI layouts
- displaying a fancy splash screen
- populating the Runtime Menu dynamically
- populating a navigation module dynamically
- generating an event log for debugging

...and much more!

The UI framework is built on top of the HSE Libraries and consists of a collection of DQMH modules and a few helper VIs.

### 1.1.1. FRAMEWORK MODULES

These DQMH modules are designed to be reused, hence they are generic and not part of the project-specific code of an application:

- Event Manager
- UI Manager
- Navigation

### 1.1.2. FRAMEWORK VIS

In order to dynamically load DQMH modules and to achieve some of the UI framework functionality, the following VIs are part of the framework, too:

#### STARTUP VI

The /startup.vi is used to run the application. It...

- shows a splash screen
- reads the main configuration (containing the list of modules to load)
- loads the UI Manager and the Event Manager modules
- loads a list of project modules (no static linkage)
- calls the “configure” requests of all the modules
- displays the front panel of the UI Manager module

## PROJECT VIS

These are project-related or project-specific VIs that are supplied by the UI Framework but can or have to be modified for each project.

- /Project/Project.lvlib: Contains the project-specific VIs
- /Project/PROJECT\_InitLogging.vi: Starts the hse-logger
- /Project/PROJECT\_Name-constant.vi: Name of the project
- /Project/PROJECT\_RunTimeMenu.rtm: Run-Time Menu for the application
- /Project/PROJECT\_SplashScreen.vi: Splash Screen
- /Project/PROJECT\_StartupSteps.ctl: List of steps for the startup.vi to execute
- /Project/PROJECT\_UserCredentials.vi: List of users for the built-in login feature

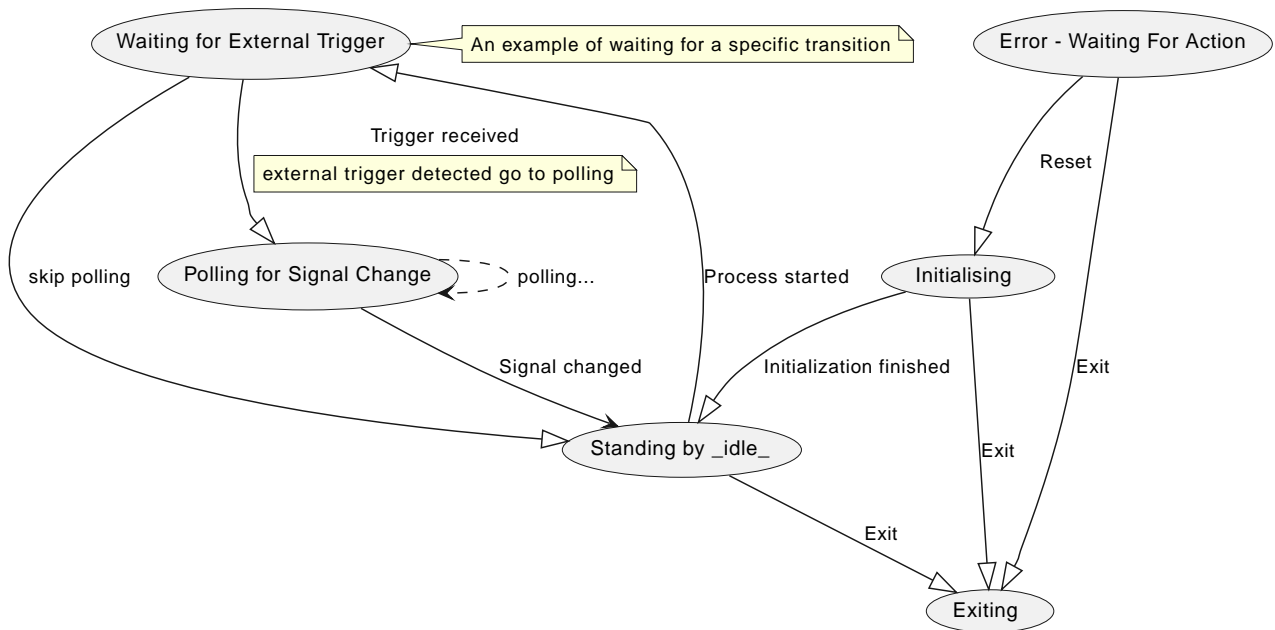
## 1.2. PROJECT-SPECIFIC DQMH MODULES

In order for DQMH modules to be compatible with our application template, these modules need to implement our HSE DQMH flavour.

# CHAPTER 2. STATE MACHINES

## 2.1. STATEMACHINE

### 2.1.1. MAIN.VI



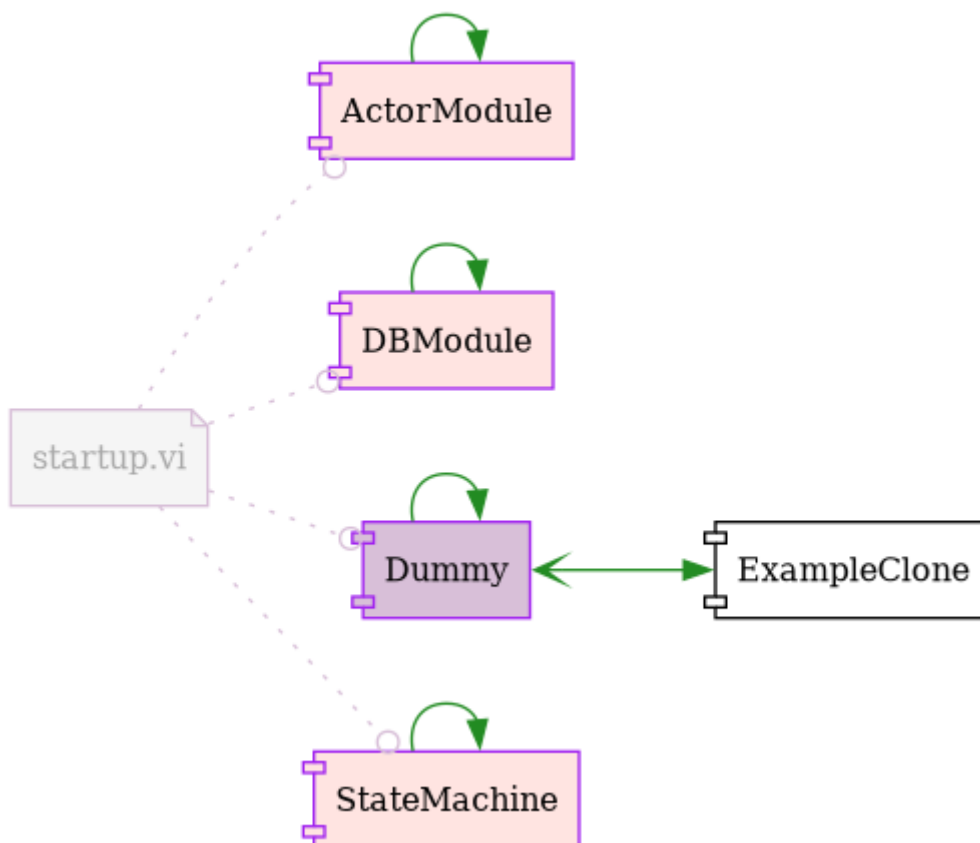
# CHAPTER 3. ARCHITECTURE

## 3.1. OVERVIEW

This project contains 5 DQMH modules.

Table 1. Modules list

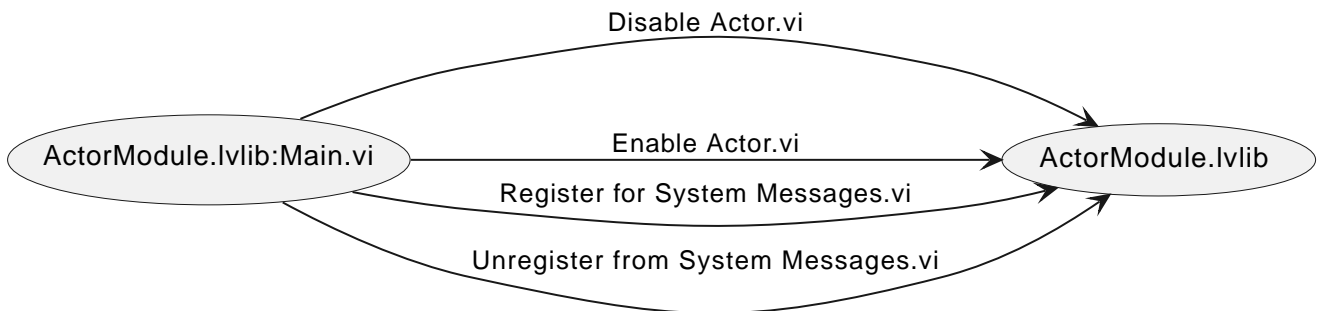
| Module             | Type      | HSE-ified? | Started? | Displayed? |
|--------------------|-----------|------------|----------|------------|
| ActorModule.lvlib  | Singleton | x          | x        |            |
| DBModule.lvlib     | Singleton | x          | x        |            |
| Dummy.lvlib        | Singleton | x          | x        | x          |
| ExampleClone.lvlib | Cloneable |            |          |            |
| StateMachine.lvlib | Singleton | x          | x        |            |



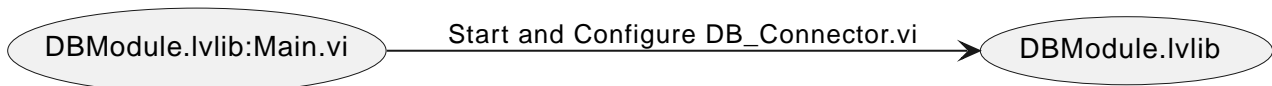
This graph represents the links between all DQMH modules. Purple borders denote HSE-ified modules. A peach background denotes modules being started by the framework as configured in config.ini. A lilac background denotes the module being displayed first (the "startup module").

## 3.2. CALLERS

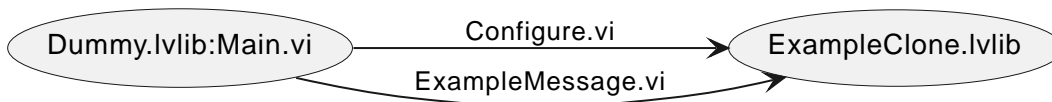
### 3.2.1. ACTORMODULE.LVLIB



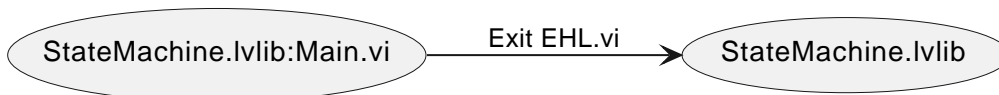
### 3.2.2. DBMODULE.LVLIB



### 3.2.3. EXAMPLECLONE.LVLIB



### 3.2.4. STATEMACHINE.LVLIB



## 3.3. LISTENERS

No elements found.

# APPENDIX A: DQMH

DQMH modules documentation

## A.1. ACTORMODULE.LVLIB

**Type:** Singleton

**Responsibility:** No description found (add content in DQMH module lplib description)

### A.1.1. MODULE RELATIONSHIP

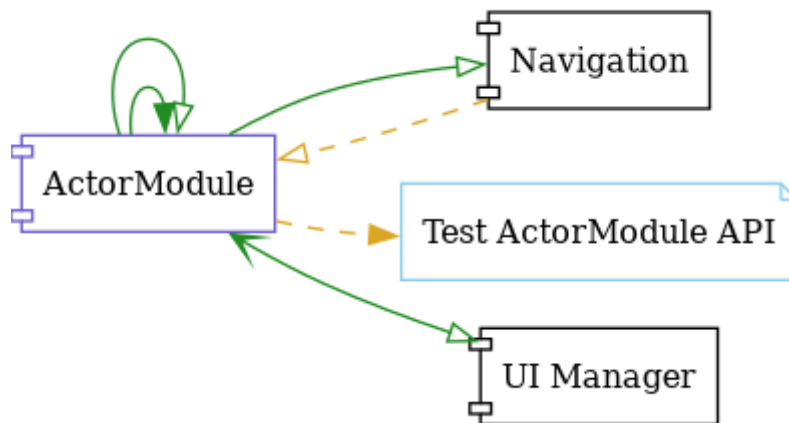


Table 2. Requests callers

| Request Name                    | Callers                   |
|---------------------------------|---------------------------|
| Configure                       |                           |
| Disable Actor                   | ActorModule.lvlib:Main.vi |
| Enable Actor                    | ActorModule.lvlib:Main.vi |
| Get Module Execution Status     |                           |
| Hide Panel                      |                           |
| Prepare                         |                           |
| Register for System Messages    | ActorModule.lvlib:Main.vi |
| Request UI Display              |                           |
| Show Diagram                    |                           |
| Show Panel                      |                           |
| Unregister from System Messages | ActorModule.lvlib:Main.vi |

Table 3. Broadcasts Listeners

| Broadcast Name  | Listeners               |
|-----------------|-------------------------|
| Error Reported  | Test ActorModule API.vi |
| Module Did Init | Test ActorModule API.vi |

| Broadcast Name                 | Listeners               |
|--------------------------------|-------------------------|
| Module Did Stop                | Test ActorModule API.vi |
| Status Updated                 | Test ActorModule API.vi |
| System Message                 | Test ActorModule API.vi |
| Update Module Execution Status | Test ActorModule API.vi |

*Table 4. Used requests*

| Module            | Requests   |
|-------------------|--|
| ActorModule.lvlib | Disable Actor.vi<br>Enable Actor.vi<br>Register for System Messages.vi<br>Unregister from System Messages.vi |
| Navigation.lvlib  | Get Module Execution Status.vi<br>Update Navigation.vi   |
| UI Manager.lvlib  | Get Module Execution Status.vi<br>Remove VI from Subpanel by VI Ref.vi                                       |

*Table 5. Registered broadcast*


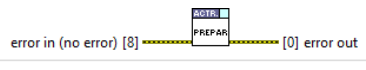








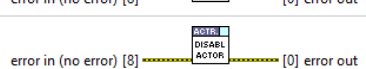


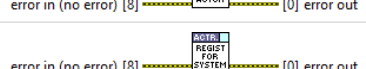


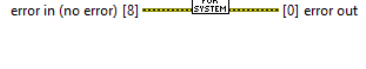






| Module           | Broadcasts             |
|------------------|------------------------|
| Navigation.lvlib | Sub-Control Pressed.vi |


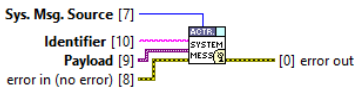

## A.1.2. EVENT LIST







The **default events** are not listed.



*Table 6. Events*

| Name                            | Type  | Connector pane  | Description   | S. | R.  | I. |
|---------------------------------|---|---|---|----|---|----|
| Prepare                         |  | <br>error in (no error) [8] --- [0] error out  | Prepare (the UI of the) module for display  |    |  |    |
| Configure                       |  | <br>error in (no error) [8] --- [0] error out  | Triggers the auto-configuration of the module   |    |  |    |
| Enable Actor                    |  | <br>error in (no error) [8] --- [0] error out  | Activates the helper loop timeout structure   |    |  |    |
| Disable Actor                   |  | <br>error in (no error) [8] --- [0] error out  | Stops the timeout polling of the helper loop  |    |  |    |
| Register for System Messages    |  | <br>error in (no error) [8] --- [0] error out  | Registers for other modules' System Message broadcast events  |    |  |    |
| Unregister from System Messages |  | <br>error in (no error) [8] --- [0] error out  | Unregisters from other modules' System Message broadcast events   |    |  |    |
| Request UI Display              |  | <br>Scenario [10] --- [3] Accept?<br>error in (no error) [8] --- [0] error out<br>wait for reply (T) [6] --- [0] error out | Requests the module to display its UI as specified in <b>Scenario</b> : "managed"  in the UI Manager's subpanel "stand-alone"  as a separate window |    |  |    |

| Name           | Type  | Connector pane  | Description  | S. | R. | I.  |
|----------------|---|---|--|----|----|---|
| System Message |  |  | <p>Broadcasts a "System Message" consisting of an Identifier (string), a Payload (variant) and a Source (enum).</p> <p><b>Source</b> values are: - "module": A broadcast from within a module itself - "application": A broadcast generated from outside a module - "network": A broadcast sent/received via network</p> |    |    |  |

**Type:**  → Request |  → Request and Wait for Reply |  → Broadcast

**Scope:**  → Protected |  → Community

**Reentrancy:**  → Preallocated reentrancy |  → Shared reentrancy

**Inlining:**  → Inlined

### A.1.3. MODULE START/STOP CALLS

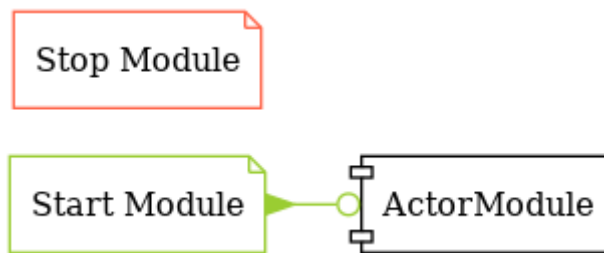


Table 7. Start and Stop module callers

| Function     | Callers                          |
|--------------|----------------------------------|
| Start Module | ActorModule.lvlib:Load Module.vi |
| Stop Module  |                                  |

### A.1.4. MODULE HELPER LOOPS

Table 8. Helper Loops Found

| Name                | Helper Loop Type | Events Handled   |
|---------------------|------------------|--|
| Helper (Actor) Loop | Custom           | Timeout<br><br>ActorModule Request Events.Enable Actor<br><br>ActorModule Request Events.Disable Actor<br><br>ActorModule Request Events.Register for System Messages<br><br>UI.System Message<br><br>ActorModule Request Events.Unregister from System Messages<br><br>ActorModule Request Events.Stop Module |

## A.1.5. MODULE CUSTOM ERRORS



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

Module ActorModule.lvlib use the following custom errors:

Table 9. 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.   |
| Request and Wait for Reply Timeout | 403686 | %s  |

## A.1.6. MODULE CONSTANT VIS

Table 10. Constant VIs Found

| VI Name                    | Data Type | Value       |
|----------------------------|-----------|-------------|
| Module Name—constant.vi    | String    | ActorModule |
| Module Timeout—constant.vi | I32       | 5000        |

## A.2. DBMODULE.LVLIB

**Type:** Singleton

**Responsibility:** No description found (add content in DQMH module lvlib description)

### A.2.1. MODULE RELATIONSHIP

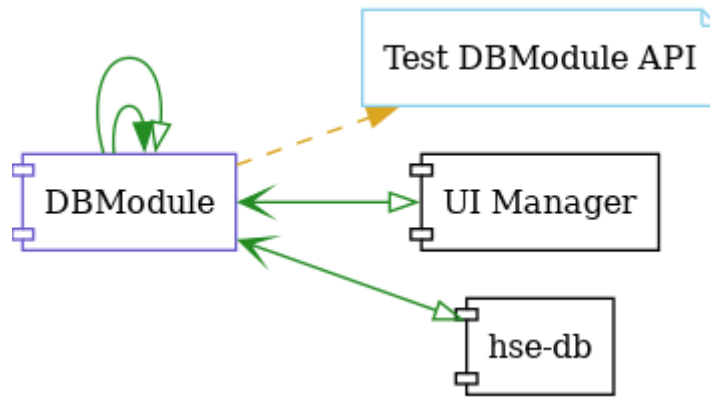


Table 11. Requests callers

| Request Name                     | Callers                |
|----------------------------------|------------------------|
| Configure                        |                        |
| Get Module Execution Status      |                        |
| Hide Panel                       |                        |
| Prepare                          |                        |
| Request UI Display               |                        |
| Show Diagram                     |                        |
| Show Panel                       |                        |
| Start and Configure DB_Connector | DBModule.lvlib:Main.vi |

Table 12. Broadcasts Listeners

| Broadcast Name                 | Listeners            |
|--------------------------------|----------------------|
| Error Reported                 | Test DBModule API.vi |
| Module Did Init                | Test DBModule API.vi |
| Module Did Stop                | Test DBModule API.vi |
| Status Updated                 | Test DBModule API.vi |
| System Message                 | Test DBModule API.vi |
| Update Module Execution Status | Test DBModule API.vi |

Table 13. Used requests

| Module           | Requests   |
|------------------|--|
| DBModule.lvlib   | Start and Configure DB_Connector.vi  |
| UI Manager.lvlib | Remove VI from Subpanel by VI Ref.vi   |
| hse-db.lvlib     | DB_CONNECTOR.lvlib:Configure.vi<br>DB_CONNECTOR.lvlib:Get DB-Type.vi<br>DB_CONNECTOR.lvlib:Query.vi (5)<br>DB_CONNECTOR.lvlib:Stop Module.vi |

Table 14. Registered broadcast


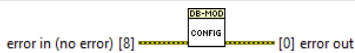


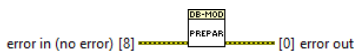










| Module | Broadcasts |
|--------|------------|
| ☒—☒    | ☒—☒        |

## A.2.2. EVENT LIST





The **default events** are not listed.

Table 15. Events

| Name                             | Type  | Connector pane  | Description   | S. | R.  | I. |
|----------------------------------|---|---|---|----|---|----|
| Configure                        |  |  | Triggers the auto-configuration of the module.  |    |  |    |
| Prepare                          |  |  | Prepare (the UI of the) module for display.   |    |  |    |
| Request UI Display               |  |  | Requests the module to display its UI as specified in <b>Scenario</b> : "managed" <input type="checkbox"/> in the UI Manager's subpanel "stand-alone" <input type="checkbox"/> as a separate window |    |  |    |
| Start and Configure DB_Connector |  |  | starts and configures the DB_Connector Module   |    |  |    |
| System Message                   |  |  | Broadcast a system generic message.   |    |  |    |

**Type:**  → Request |  → Request and Wait for Reply |  → Broadcast

**Scope:**  → Protected |  → Community

**Reentrancy:**  → Preallocated reentrancy |  → Shared reentrancy

**Inlining:**  → Inlined

## A.2.3. MODULE START/STOP CALLS

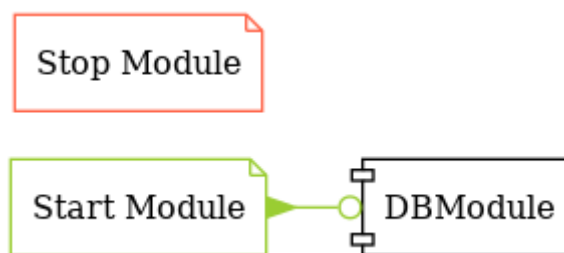


Table 16. Start and Stop module callers

| Function     | Callers                       |
|--------------|-------------------------------|
| Start Module | DBModule.lvlib:Load Module.vi |
| Stop Module  |                               |

## A.2.4. MODULE HELPER LOOPS



No Helper Loops Found

## A.2.5. MODULE CUSTOM ERRORS



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

Module DBModule.lvlib use the following custom errors:

Table 17. 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.   |
| Request and Wait for Reply Timeout | 403686 | %s  |

## A.2.6. MODULE CONSTANT VIS

Table 18. Constant VIs Found

| VI Name                    | Data Type | Value    |
|----------------------------|-----------|----------|
| Module Name—constant.vi    | String    | DBModule |
| Module Timeout—constant.vi | I32       | 5000     |

## A.3. DUMMY.LVLIB

**Type:** Singleton

**Responsibility:** No description found (add content in DQMH module lvlib description)

### A.3.1. MODULE RELATIONSHIP

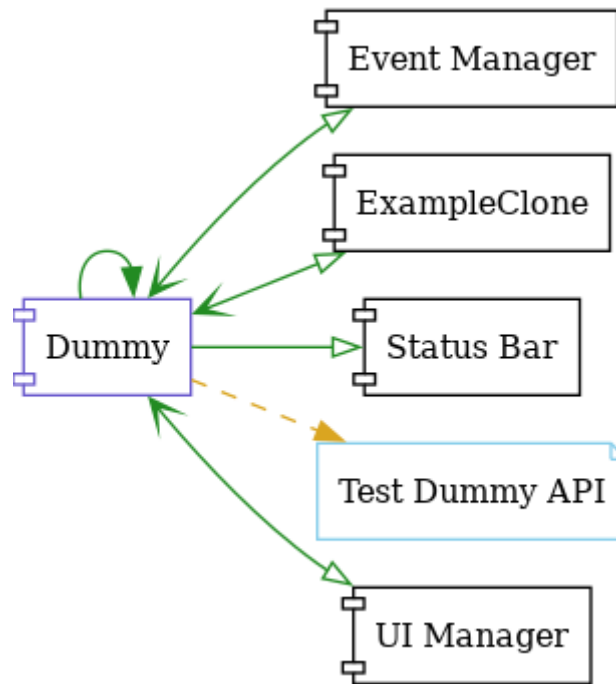


Table 19. Requests callers

| Request Name                | Callers |
|-----------------------------|---------|
| Configure                   |         |
| Get Module Execution Status |         |
| Hide Panel                  |         |
| Prepare                     |         |
| Register for UI Login       |         |
| Request UI Display          |         |
| Show Diagram                |         |
| Show Panel                  |         |

Table 20. Broadcasts Listeners

| Broadcast Name                 | Listeners         |
|--------------------------------|-------------------|
| Error Reported                 | Test Dummy API.vi |
| Module Did Init                | Test Dummy API.vi |
| Module Did Stop                | Test Dummy API.vi |
| Status Updated                 | Test Dummy API.vi |
| System Message                 | Test Dummy API.vi |
| Update Module Execution Status | Test Dummy API.vi |

Table 21. Used requests

| Module              | Requests       |
|---------------------|----------------|
| Event Manager.lvlib | Set Modules.vi |

| Module             | Requests   |
|--------------------|--|
| ExampleClone.lvlib | Configure.vi<br>ExampleMessage.vi<br>Request UI Display.vi<br>Show Diagram.vi<br>Stop Module.vi (2)  |
| Status Bar.lvlib   | Update Status Bar.vi   |
| UI Manager.lvlib   | Get CLI Arguments.vi<br>Get Module Execution Status.vi<br>Remove VI from Subpanel by VI Ref.vi<br>Set Modules.vi<br>Set Run-Time Menu Visibility.vi (2)<br>Set Window Closeable State.vi (2) |

Table 22. Registered broadcast

| Module | Broadcasts |
|--------|------------|
| ☒—☒    | ☒—☒        |

### A.3.2. EVENT LIST



The **default events** are not listed.

Table 23. Events

| Name                  | Type                       | Connector pane | Description   | S. | R. | I. |
|-----------------------|----------------------------|----------------|---|----|----|----|
| Configure             | Request                    |                | Self-configuration (triggered automatically from main.vi)   |    |    | ☒  |
| Prepare               | Request                    |                | Prepare (the UI of the) module for display  |    |    | ☒  |
| Register for UI Login | Request                    |                | Registers for the UI module's "Login" broadcast event   |    |    | ☒  |
| Request UI Display    | Request and Wait for Reply |                | Requests the module to display its UI as specified in <b>Scenario</b> : "managed" ☒ in the UI Manager's subpanel "stand-alone" ☒ as a separate window |    |    | ☒  |
| System Message        | Broadcast                  |                | <b>Note:</b> This VI was modified by the <b>Validate DQMH Module</b> tool to have a broadcast event glyph overlay on its icon.                        |    |    | ☒  |

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

**Scope:** Protected | Community

**Reentrancy:** Preallocated reentrancy | Shared reentrancy

**Inlining:** Inlined

### A.3.3. MODULE START/STOP CALLS

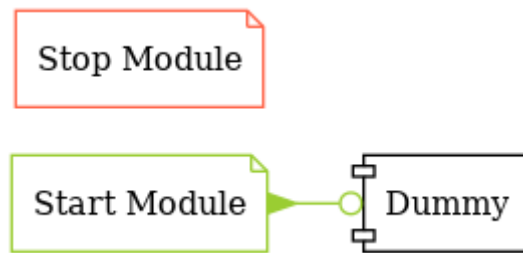


Table 24. Start and Stop module callers

| Function     | Callers                    |
|--------------|----------------------------|
| Start Module | Dummy.lvlib:Load Module.vi |
| Stop Module  |                            |

### A.3.4. MODULE HELPER LOOPS



No Helper Loops Found

### A.3.5. MODULE CUSTOM ERRORS



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

Module Dummy.lvlib use the following custom errors:

Table 25. 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.   |
| Request and Wait for Reply Timeout | 403686 | %s  |
| My first custom error              | 500987 | My first custom error   |
| This is an error with input        | 500988 | This is an error with input %s  |
| Error ring                         | 500989 | Error ring on block diagram   |

### A.3.6. MODULE CONSTANT VIS

Table 26. Constant VIs Found

| VI Name                    | Data Type | Value |
|----------------------------|-----------|-------|
| Module Name—constant.vi    | String    | Dummy |
| Module Timeout—constant.vi | I32       | 5000  |

## A.4. EXAMPLECLONE.LVLIB

**Type:** Cloneable

**Responsibility:** No description found (add content in DQMH module lplib description)

### A.4.1. MODULE RELATIONSHIP

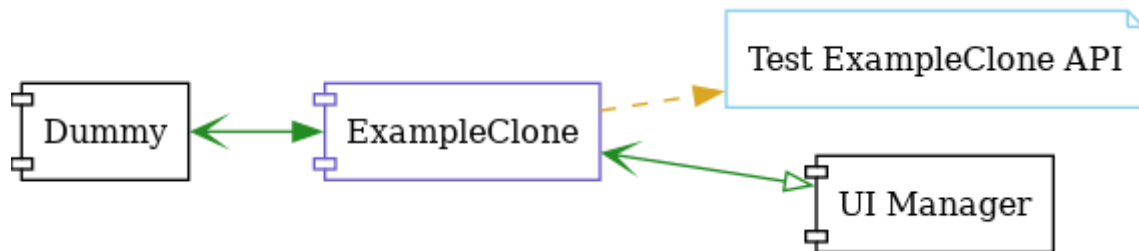


Table 27. Requests callers

| Request Name       | Callers            |
|--------------------|--------------------|
| Configure          | Dummy.lvlb:Main.vi |
| ExampleMessage     | Dummy.lvlb:Main.vi |
| Hide Panel         |                    |
| Prepare            |                    |
| Request UI Display | Dummy.lvlb:Main.vi |
| Show Diagram       | Dummy.lvlb:Main.vi |
| Show Panel         |                    |

Table 28. Broadcasts Listeners

| Broadcast Name                 | Listeners                |
|--------------------------------|--------------------------|
| Error Reported                 | Test ExampleClone API.vi |
| Module Did Init                | Test ExampleClone API.vi |
| Module Did Stop                | Test ExampleClone API.vi |
| Status Updated                 | Test ExampleClone API.vi |
| System Message                 | Test ExampleClone API.vi |
| Update Module Execution Status | Test ExampleClone API.vi |

Table 29. Used requests

| Module          | Requests                             |
|-----------------|--------------------------------------|
| UI Manager.lvlb | Remove VI from Subpanel by VI Ref.vi |

Table 30. Registered broadcast


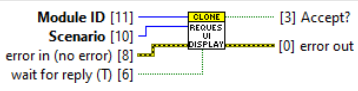


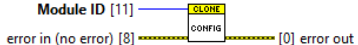


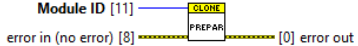


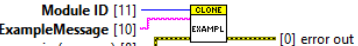


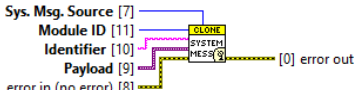

| Module | Broadcasts |
|--------|------------|
| ☒—☒    | ☒—☒        |

### A.4.2. EVENT LIST



The default events are not listed.

Table 31. Events

| Name               | Type   | Connector pane  | Description  | S. | R.   | I. |
|--------------------|--|---|--|----|--|----|
| Request UI Display |   |  <p>Module ID [11] [3] Accept?<br/>           Scenario [10] [0] error out<br/>           error in (no error) [8]<br/>           wait for reply (T) [6]</p>   | Requests another module to display its UI in the given <b>Display Scenario</b> : The "Stand-Alone" scenario asks the module to open its front panel as a new windows. The "Managed" scenario asks the module to broadcast a request for display which the UI Manager then processes, showing the module in its subpanel. |    |   |    |
| Configure          |   |  <p>Module ID [11] [0] error out<br/>           error in (no error) [8]</p>  | Triggers the auto-configuration of the module  |    |   |    |
| Prepare            |   |  <p>Module ID [11] [0] error out<br/>           error in (no error) [8]</p>  | Prepares the module UI (front panel) for display   |    |   |    |
| ExampleMessage     |   |  <p>Module ID [11] [0] error out<br/>           ExampleMessage [10] [0] error out<br/>           error in (no error) [8]</p>   | Send a message to the module that is displayed on its front panel  |    |   |    |
| System Message     |  |  <p>Sys. Msg. Source [7] [0] error out<br/>           Module ID [11] [0] error out<br/>           Identifier [10] [0] error out<br/>           Payload [9] [0] error out<br/>           error in (no error) [8]</p> | No description found (add content in vi description)   |    |  |    |

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

Scope:  → Protected |  → Community

Reentrancy:  → Preallocated reentrancy |  → Shared reentrancy

Inlining:  → Inlined

### A.4.3. MODULE START/STOP CALLS

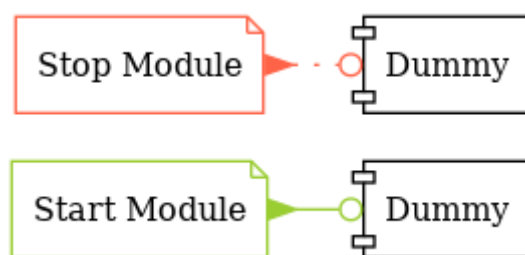


Table 32. Start and Stop module callers

| Function     | Callers             |
|--------------|---------------------|
| Start Module | Dummy.lvlib:Main.vi |
| Stop Module  | Dummy.lvlib:Main.vi |

### A.4.4. MODULE HELPER LOOPS



No Helper Loops Found

## A.4.5. MODULE CUSTOM ERRORS



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

Module ExampleClone.lvlib use the following custom errors:

Table 33. Custom errors

| Name                               | Code   | Description  |
|------------------------------------|--------|--|
| Module Running as Singleton        | 403680 | The "%s" module is currently running as singleton, but the Start Module VI was called with 'Run as Singleton' specified as FALSE.  |
| 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.  |
| Module Not Running                 | 403684 | Not a single instance of "%s" Module running.  |
| Module Running as Cloneable        | 403685 | The "%s" module is currently running as cloneable, but the Start Module VI was called with 'Run as Singleton' specified as TRUE.   |
| Request and Wait for Reply Timeout | 403686 | %s   |
| Master Reference Not Closed        | 403687 | The "%s" module cannot be run as singleton because the Master Reference is still open from a prior run as cloneable. If you plan on running this module as both singleton and cloneable, consider changing your Main VI to wire a value of TRUE to the 'Close Master Reference' input of Init Module.vi. |

## A.4.6. MODULE CONSTANT VIS

Table 34. Constant VIs Found

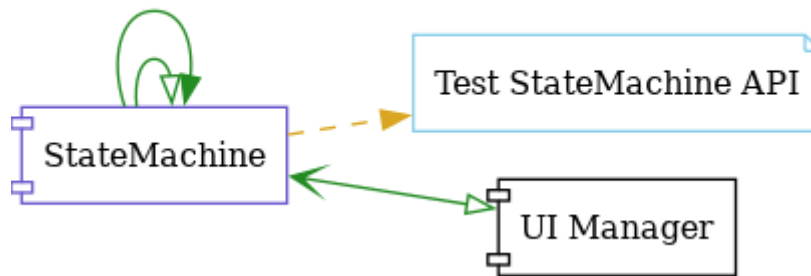
| VI Name                    | Data Type            | Value                   |
|----------------------------|----------------------|-------------------------|
| Module Name—constant.vi    | ["String", "String"] | [" %d", "ExampleClone"] |
| Module Timeout—constant.vi | I32                  | 5000                    |

## A.5. STATEMACHINE.LVLIB

**Type:** Singleton

**Responsibility:** No description found (add content in DQMH module lvlib description)

### A.5.1. MODULE RELATIONSHIP


*Table 35. Requests callers*

| Request Name                | Callers                    |
|-----------------------------|----------------------------|
| Configure                   |                            |
| DEMO Inject Transition      |                            |
| Exit EHL                    | StateMachine.lvlib:Main.vi |
| Get Module Execution Status |                            |
| Hide Panel                  |                            |
| Prepare                     |                            |
| Request UI Display          |                            |
| Show Diagram                |                            |
| Show Panel                  |                            |

*Table 36. Broadcasts Listeners*

| Broadcast Name                 | Listeners                |
|--------------------------------|--------------------------|
| DEMO Updated UI State          | Test StateMachine API.vi |
| Error Reported                 | Test StateMachine API.vi |
| Module Did Init                | Test StateMachine API.vi |
| Module Did Stop                | Test StateMachine API.vi |
| Status Updated                 | Test StateMachine API.vi |
| System Message                 | Test StateMachine API.vi |
| Update Module Execution Status | Test StateMachine API.vi |

*Table 37. Used requests*

| Module             | Requests   |
|--------------------|--|
| StateMachine.lvlib | Exit EHL.vi  |
| UI Manager.lvlib   | Remove VI from Subpanel by VI Ref.vi<br>Set Window Closeable State.vi (2)<br>Shutdown Application.vi |

*Table 38. Registered broadcast*

| Module | Broadcasts |
|--------|------------|
| ☒—☒    | ☒—☒        |

## A.5.2. EVENT LIST



The **default events** are not listed.

Table 39. Events

| Name                   | Type | Connector pane | Description   | S. | R. | I. |
|------------------------|------|----------------|---|----|----|----|
| Configure              |      |                | Triggers the auto-configuration of the module.  |    |    |    |
| Prepare                |      |                | Prepares the module for displaying its front panel.   |    |    |    |
| Request UI Display     |      |                | Requests the module to display its UI as specified in <b>Scenario</b> : "managed"  in the UI Manager's subpanel "stand-alone"  as a separate window   |    |    |    |
| DEMO Inject Transition |      |                | Triggers a transition via the public API.<br><br>This request serves the purpose of testing/show-casing the state machine. By design, transitions should not be triggered from outside the DQMH module (i.e. via the public API)  |    |    |    |
| System Message         |      |                | Broadcasts a "System Message" consisting of an Identifier (string), a Payload (variant) and a Source (enum).<br><br><b>Source</b> values are: - "module": A broadcast from within a module itself - "application": A broadcast generated from outside a module - "network": A broadcast sent/received via network |    |    |    |
| DEMO Updated UI State  |      |                | Broadcasts whenever the "Update UI State" private request was executed after a state change   |    |    |    |
| Exit EHL               |      |                | this event will stop the event handling loop  |    |    |    |

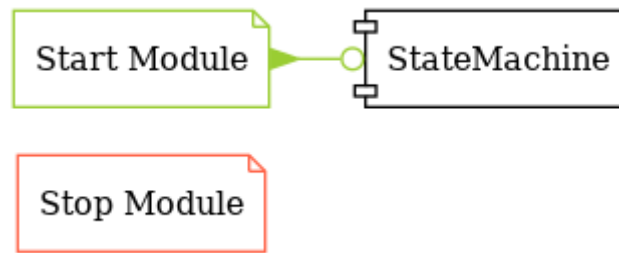
**Type:** → Request | → Request and Wait for Reply | → Broadcast

**Scope:** → Protected | → Community

**Reentrancy:** → Preallocated reentrancy | → Shared reentrancy

**Inlining:** → Inlined

## A.5.3. MODULE START/STOP CALLS


*Table 40. Start and Stop module callers*

| Function     | Callers                           |
|--------------|-----------------------------------|
| Start Module | StateMachine.lvlib:Load Module.vi |
| Stop Module  |                                   |

## A.5.4. MODULE HELPER LOOPS

*Table 41. Helper Loops Found*

| Name               | Helper Loop Type | Events Handled           |
|--------------------|------------------|--------------------------|
| State Machine Loop | Custom           | No Event Structure Found |

## A.5.5. MODULE CUSTOM ERRORS



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

Module StateMachine.lvlib use the following custom errors:

*Table 42. 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.   |
| Request and Wait for Reply Timeout | 403686 | %s  |

## A.5.6. MODULE CONSTANT VIS

*Table 43. Constant VIs Found*

| VI Name                    | Data Type | Value        |
|----------------------------|-----------|--------------|
| Module Name—constant.vi    | String    | StateMachine |
| Module Timeout—constant.vi | I32       | 5000         |

# APPENDIX B: LIBRARIES

Misc. reuse libraries


## B.1. HSE-STATE-MACHINE.LVLIB

**Responsibility:** No description found (add content in lplib description)

**Version:** 1.0.0.0

### B.1.1. FUNCTIONS

Table 44. Functions (non private scope only)

| Name                              | Connector pane  | Description   | S. | R. | I.  |
|-----------------------------------|---|---|----|----|---|
| SM Create Event                   | [3] Req. Transition Event<br>error in (no error) [8] [0] error out  | Creates the local user event used for requesting a transition.  |    |    |   |
| SM Destroy Event                  | Req. Transition Event [11]<br>error in (no error) [8] [0] error out   | Destroys the local user event used for requesting a transition.   |    |    |   |
| SM Request Transition             | Req. Transition [7]<br>Transition [11]<br>Data [10]<br>error in (no error) [8] [0] error out  | Sends a Transition to the State Machine which will react dependent on it's state.   |    |    |   |
| SM Polling - Leave State if FALSE | Boolean Condition [11] [3] Polling Action   | <b>Keep polling</b> if the boolean condition is TRUE  |    |    |   |
| SM Polling - Leave State if TRUE  | Boolean Condition [11] [3] Polling Action   | <b>Keep polling</b> if the boolean condition is TRUE  |    |    |   |
| SM DEMO                           | SM HSE<br>DEMO  | Demonstrates the use of the HSE-State Machine and can be used as a Copy and Paste Template.   |    |    |   |
| SM Process Exception Transition   | Exception State [7]<br>Exception Transition [11]<br>Data [10]<br>error in (no error) [8]<br>Message Queue In [6]<br>[3] State<br>[2] Exception<br>[0] error out | If an error occurs, go to the "Erroring" state and send an update to the UI   |    |    |   |
| SM Update UI State                | Message Queue In [11]<br>State [10]<br>error in (no error) [8]<br>[3] Message Queue Out<br>[0] error out  | Propagate state change to UI.<br><br>HSE: This is only a place holder, in DQMH modules this would use the Message Queue. Implement your own communication mechanism here. |    |    |   |
| SM Wait for Transition            | Event Registration Refnum [11]<br>error in (no error) [8]<br>[3] Transition<br>[2] Data<br>[0] error out  | Waits for a Transition to occur in the duration of Timeout.   |    |    |  |

Scope:  → Protected |  → Community

Reentrancy:  → Preallocated reentrancy |  → Shared reentrancy

Inlining:  → Inlined

## B.1.2. LIBRARY CONSTANT VIS



No Constant VIs Found




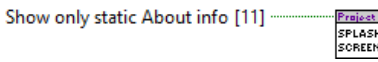

## B.2. PROJECT.LVLIB

**Responsibility:** No description found (add content in lvlib description)

**Version:** 1.0.0.0

### B.2.1. FUNCTIONS

Table 45. Functions (non private scope only)

| Name                                    | Connector pane  | Description  | S. | R. | I. |
|---|---|--|----|----|----|
| PROJECT_Evaluate Default Menu Selection |    | No description found (add content in vi description)   |    |    |    |
| PROJECT_InitLogging                     |    | Initiate the logging functions.  |    |    |    |
| PROJECT_Layout-VI                       |  | <p>This is the default layout VI for the HSE UI Framework. It features three areas: - header (contains the "SP_Header" subpanel) - navigation (contains the "SP_Navigation" subpanel) - content (contains the "SP_Content" subpanel)</p> <p>This VI is loaded by the UI Manager module, the subpanels are then populated with DQMH modules as configured in the UI Manager's configuration file.</p> |    |    |    |
| PROJECT_SplashScreen                    |  | No description found (add content in vi description)   |    |    |    |
| PROJECT_User Credentials                |  | No description found (add content in vi description)   |    |    |    |

Scope:  → Protected |  → Community

Reentrancy:  → Preallocated reentrancy |  → Shared reentrancy

Inlining:  → Inlined

### B.2.2. LIBRARY CONSTANT VIS

Table 46. Constant VIs Found

| VI Name                                | Data Type | Value      |
|--|-----------|------------|
| PROJECT_ApplicationVersion—constant.vi | String    | unreleased |
| PROJECT_Name—constant.vi               | String    | TEMPLATE   |



| VI Name                                | Data Type | Value |
|--|-----------|-------|
| PROJECT_StaticDependencies—constant.vi |           |       |

# APPENDIX C: CLASSES

LabVIEW Classes

## C.1. CLASSES OVERVIEW

This project contains 0 classe and 0 interface.

*Table 47. Classes list*

|  |
|--|
|  |
|--|

# APPENDIX D: CUSTOM ERRORS

List of Custom Error VIs

## D.1. CUSTOM ERRORS



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

Table 48. Custom errors

| Name                               | Code   | Description  | Owned by   |
|------------------------------------|--------|--|--|
| Module Running as Singleton        | 403680 | The "%s" module is currently running as singleton, but the Start Module VI was called with 'Run as Singleton' specified as FALSE.  | ExampleClone.lvlib   |
| Module Not Running                 | 403681 | %s Module is not running.  | ActorModule.lvlib<br>DBModule.lvlib<br>Dummy.lvlib<br>StateMachine.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.  | ActorModule.lvlib<br>DBModule.lvlib<br>Dummy.lvlib<br>ExampleClone.lvlib<br>StateMachine.lvlib |
| Module Not Synced                  | 403683 | %s Module was unable to synchronize events.  | ActorModule.lvlib<br>DBModule.lvlib<br>Dummy.lvlib<br>ExampleClone.lvlib<br>StateMachine.lvlib |
| Module Not Running                 | 403684 | Not a single instance of "%s" Module running.  | ExampleClone.lvlib   |
| Module Running as Cloneable        | 403685 | The "%s" module is currently running as cloneable, but the Start Module VI was called with 'Run as Singleton' specified as TRUE.   | ExampleClone.lvlib   |
| Request and Wait for Reply Timeout | 403686 | %s   | ActorModule.lvlib<br>DBModule.lvlib<br>Dummy.lvlib<br>ExampleClone.lvlib<br>StateMachine.lvlib |
| Master Reference Not Closed        | 403687 | The "%s" module cannot be run as singleton because the Master Reference is still open from a prior run as cloneable. If you plan on running this module as both singleton and cloneable, consider changing your Main VI to wire a value of TRUE to the 'Close Master Reference' input of Init Module.vi. | ExampleClone.lvlib   |
| My first custom error              | 500987 | My first custom error  | Dummy.lvlib  |
| This is an error with input        | 500988 | This is an error with input %s   | Dummy.lvlib  |

**TEMPLATE**

---

| <b>Name</b> | <b>Code</b> | <b>Description</b>          | <b>Owned by</b> |
|-------------|-------------|-----------------------------|-----------------|
| Error ring  | 500989      | Error ring on block diagram | Dummy.lvlib     |

# GLOSSARY

The rat-documentr tool facilitates the following LabVIEW-related tools and libraries:

- Antidoc by Wovalab
- AsciiDoc Toolkit by Wovalab
- Graph Builder by C. Gambini
- Classy by T. Boylston
- DQMH® by the DQMH Consortium

Furthermore, it relies on the following tools and libraries:

- Ruby
- AsciiDoctor
- Java
- GraphViz
- PlantUML
- Kroki.io